OMRON

Vision Sensor

FH/FZ5 Series

Vision System

User's Manual

FH-1□□□
FH-3□□□

FZ5-L35□
FZ5-6□□
FZ5-11□□

SYSMAC
always in control

Z340-E1-07

Introduction

Thank you for purchasing the FH/FZ5.
This manual provides information regarding functions, performance and operating methods that are required for using the FH/FZ5.
When using the FH/FZ5, be sure to observe the following:

- The FH/FZ5 must be operated by personnel knowledgeable in electrical engineering.
- To ensure correct use, please read this manual thoroughly to deepen your understanding of the product.
- Please keep this manual in a safe place so that it can be referred to whenever necessary.

FH/FZ5 Manual Configuration

The following table gives the manual configuration of the FH/FZ5.

Cat. No.	Manual name	Contents	Main applications
2285550-0	FH Image Processing System Instruction Manual	Provides FH-series specifications, dimensions, part names, I/O information, mounting information, and wiring information.	Checking I/O and other specifications and performing installation and wiring
9524422-4 (FZ5-6□□/11□□)	FH5 Image Processing System Instruction Manual	Provides FH5-series specifications, dimensions, part names, I/O information, installation information, and wiring information.	
9910002-2 (FZ5-L3□□)			
Z340 (This manual)	Vision System FH/FZ5 Series User's Manual	Describes the software functions, settings, and operations for using FH/FH5-series Vision Sensors.	Any application other than the above applications and communications
Z341	Vision System FH/FZ5 Series Processing Item Function Reference Manual	Describes the functions, settings, and operations of the processing items that you can use with the FH/FH5-series Vision Sensors.	Checking information on processing items when designing or manipulating measurement flows Use this manual together with the *User's Manual*.
Z342	Vision System FH/FZ5 Series User's Manual for Communications Settings	Describes the functions, settings, and communications methods for communicating between FH/FH5-series Vision Sensors and external devices (e.g., a PLC). The following communications protocols are included. Parallel, PLC Link, EtherNet/IP, EtherCAT, and Non-procedure	Checking information on communications functions
Z343	Vision System FH Series Operation Manual for Sysmac Studio	Describes the operating procedures for setting up and operating FH-series Vision Sensors from the Sysmac Studio FH Tools.	Communicating with an NJ-series Controller via EtherCAT communications

Conventions Used in This Manual

Symbols

The symbols used in this manual have the following meanings.

IMPORTANT Indicates relevant operational precautions that must be followed.

Note Indicates operation-related suggestions from OMRON.

Use of Quotation Marks and Brackets

In this manual, menus and other items are indicated as follows.

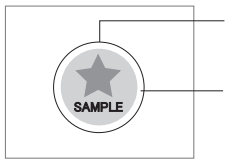
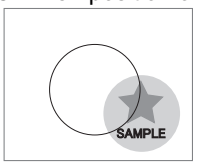
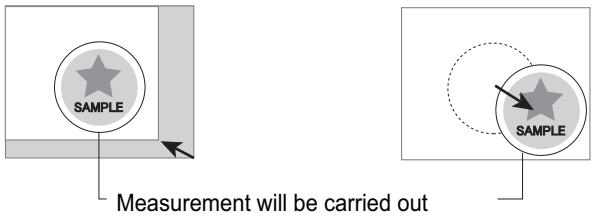
[ ] Menu Indicates the menu names or processing items shown in the menu bar.

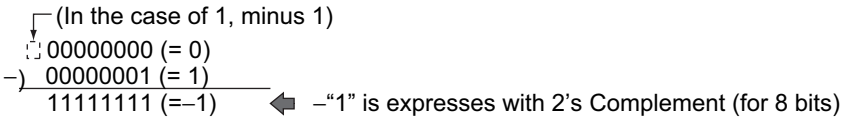
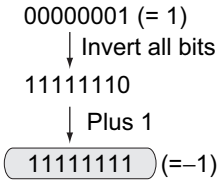
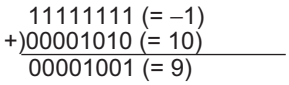
" " Item name Indicates the item names displayed on the screen.

Definitions of Basic Terms

Term	Definition
measurement flow (abbreviated as “flow”)	A continuous flow of measurement processing. A measurement flow consists of a scene created from a combination of processing items.
measurement processing	Executing processing items for inspections and measurements.
processing item	Any of the individual items for vision inspections that are partitioned and packaged so that they can be flexibly combined. These include the Search, Position Compensation, and Fine Matching items. Processing items can be classified for image input ([Input image]), inspection/measurement ([Measurement]), image correction ([Compensate image]), inspection/measurement support ([Support measurement]), process branching ([Branch]), results external output ([Output result]), resulting image display ([Display result]), etc. You can freely classify processing items to handle a wide range of applications. A scene (i.e., a unit for changing the measurement flow) is created by registering the processing items as units.
scene	A unit for changing the measurement flow that consists of a combination of processing items. “Scene” is used because of the correspondence to the scene (i.e., type of measurement object and inspection contents) where measurements are performed. A scene is created for each measurement or measurement contents. You can easily achieve a changeover simply by changing the scene when the measurement object or inspection content changes. Normally you can set up to 128 scenes. If you need more than 128 scenes, you can separate them into different groups or use the Conversion Scene Group Data Tool to create a scene group that contains over 128 scenes.
processing unit (abbreviated as “unit”)	A processing item that is registered in a scene. Numbers are assigned to processing units in order from the top and they are executed in that order. Processing items are registered for the processing units to create a scene (i.e., a unit for changing the measurement flow).
measurement trigger	A trigger for executing measurements. With a parallel interface, the STEP signal or command 00 (Continuous Measurement) is used. With a serial interface, an Execute One Measurement or a Start Continuous Measurement command is used.
test measurement	A measurement that is performed to manually test (check) measurements under the conditions that are set in the currently displayed scene. Test measurements can be executed on an Adjustment Window. Processing is completed inside the Controller and the measurement results are not normally output on an external interface. However, you can select [Output] in [Test measurement] to output the measurement results after executing measurements.
single measurement	A measurement that is executed only once in synchronization with the trigger input.
continuous measurement	Measurements executed continuously and automatically without a trigger input as soon as each measurement is completed.*1 *1: There is actually a small delay between completing a measurement and starting the next measurement. “Continuous measurement” is used as opposed to “single measurement.”

Term		Definition
operation modes	Single-line High-speed Mode	A high-speed mode that processes the measurement flow for the first trigger and then processes the measurement flow in parallel for the second trigger to achieve a high-speed trigger input interval. It is used together with the multi-input function.
	Multi-line Random-trigger Mode	A trigger mode that allows you to independently processing multiple measurement flows. With traditional image processing, two or more triggers cannot be acknowledged at the same time. In Multi-line Random-trigger Mode, you can randomly input multiple triggers into one Controller to independently process multiple scenes in parallel.
	Non-stop Adjustment Mode	A mode that allows you to adjust the flow and set parameters while performing measurements. The enables adjustments without stopping the line or stopping inspections.
	High-speed Logging Mode	A logging mode that allows complete parallel processing of measurements and logging. Traditionally, logging was not possible while processing measurements. Either measurements or logging had to be given priority and the other one had to wait. With this mode, you can save the measurement images in external memory without affecting the transaction time.
parallel processing (an option for any of the above operation modes)		Parallel processing splits part of the measurement flow into two or more tasks, and processes each task in parallel to shorten the transaction time. Processing items for parallel processing are used so that the user can specify the required parallel processing.
multi-input function		A function that is used to consecutively and quickly input images. It allows the next STEP signal to be acknowledged as soon as the image input processing is completed. There is no need to wait for measurement processing to be completed. You can check whether image input processing has been completed with the status of the READY signal. Even if the READY signal is ON when measurement processing is being executed, the next STEP signal can be acknowledged.

Term	Definition
Position compensation	When the location and direction of measured objects are not fixed, the positional deviation between reference position and current position is calculated and measurement is performed after correcting. Please select processing items that are appropriate to the measurement object from processing items that are related to position compensation. ● Reference position Measurement area and objects to be measured are correctly aligned.  ● When position of object to be measured is deflected  When position deflection correction is set in advance: Measurement will be carried out after moving the image for a corresponding deflection and returning to the reference position. Measurement will be carried out after moving the Measurement area for a corresponding deflection. 
Intelligent camera (with lighting function)	Cameras with a dome-shaped light can also be controlled with the controller. This is beneficial when the effects of ambient light are to be avoided and when it is desirable to shorten the lighting setting time. For details, see Reference: ▶ Intelligent Camera (with Lighting) (p.73).
Reference position	The point that is always the reference. If the location of the registered model is different from the reference position, the setting should be changed in [Ref. setting].
Model	The image pattern that serves as the inspection target. Characteristics portions are extracted from images of the object and registered as model registration.

Term	Definition
2's complement	Binary numbers are generally used to represent negative numbers. Negative numbers are expressed by "Inverting all bits of a positive number and adding 1 to the result". (Example) "−1" is expressed as 2's complement "−1" can be calculated by "0-1".  There are methods for simple calculation without performing this kind of computation. For instance, "Negative number = inverting all bits of a positive number and then adding 1 to the result".  The first digit is used to judge whether the number is positive or negative. • When 0: Positive number (or 0) • When 1: Negative number The advantage of two's complement numbers is that positive and negative numbers can be used as is in calculations. (Example) When −1+10=9 

Contents

MEMO

Terms and Conditions Agreement

Read and understand this Manual

Please read and understand this catalog before purchasing the products. Please consult your OMRON representative if you have any questions or comments.

Warranty, Limitations of Liability

● Warranties

Exclusive Warranty

Omron's exclusive warranty is that the Products will be free from defects in materials and workmanship for a period of twelve months from the date of sale by Omron (or such other period expressed in writing by Omron). Omron disclaims all other warranties, express or implied.

Limitations

OMRON MAKES NO WARRANTY OR REPRESENTATION, EXPRESS OR IMPLIED, ABOUT NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE OF THE PRODUCTS. BUYER ACKNOWLEDGES THAT IT ALONE HAS DETERMINED THAT THE PRODUCTS WILL SUITABLY MEET THE REQUIREMENTS OF THEIR INTENDED USE.
Omron further disclaims all warranties and responsibility of any type for claims or expenses based on infringement by the Products or otherwise of any intellectual property right.

Buyer Remedy

Omron’s sole obligation hereunder shall be, at Omron’s election, to (i) replace (in the form originally shipped with Buyer responsible for labor charges for removal or replacement thereof) the non-complying Product, (ii) repair the non-complying Product, or (iii) repay or credit Buyer an amount equal to the purchase price of the non-complying Product; provided that in no event shall Omron be responsible for warranty, repair, indemnity or any other claims or expenses regarding the Products unless Omron’s analysis confirms that the Products were properly handled, stored, installed and maintained and not subject to contamination, abuse, misuse or inappropriate modification. Return of any Products by Buyer must be approved in writing by Omron before shipment. Omron Companies shall not be liable for the suitability or unsuitability or the results from the use of Products in combination with any electrical or electronic components, circuits, system assemblies or any other materials or substances or environments. Any advice, recommendations or information given orally or in writing, are not to be construed as an amendment or addition to the above warranty.

See http://www.omron.com/global/ or contact your Omron representative for published information.

● Limitation on Liability; Etc

OMRON COMPANIES SHALL NOT BE LIABLE FOR SPECIAL, INDIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, LOSS OF PROFITS OR PRODUCTION OR COMMERCIAL LOSS IN ANY WAY CONNECTED WITH THE PRODUCTS, WHETHER SUCH CLAIM IS BASED IN CONTRACT, WARRANTY, NEGLIGENCE OR STRICT LIABILITY.

Further, in no event shall liability of Omron Companies exceed the individual price of the Product on which liability is asserted.

Application Considerations Warranties

Suitability of Use

Omron Companies shall not be responsible for conformity with any standards, codes or regulations which apply to the combination of the Product in the Buyer's application or use of the Product. At Buyer's request, Omron will provide applicable third party certification documents identifying ratings and limitations of use which apply to the Product. This information by itself is not sufficient for a complete determination of the suitability of the Product in combination with the end product, machine, system, or other application or use. Buyer shall be solely responsible for determining appropriateness of the particular Product with respect to Buyer's application, product or system. Buyer shall take application responsibility in all cases.

NEVER USE THE PRODUCT FOR AN APPLICATION INVOLVING SERIOUS RISK TO LIFE OR PROPERTY WITHOUT ENSURING THAT THE SYSTEM AS A WHOLE HAS BEEN DESIGNED TO ADDRESS THE RISKS, AND THAT THE OMRON PRODUCT(S) IS PROPERLY RATED AND INSTALLED FOR THE INTENDED USE WITHIN THE OVERALL EQUIPMENT OR SYSTEM.

Programmable Products

Omron Companies shall not be responsible for the user's programming of a programmable Product, or any consequence thereof.

Disclaimers

Performance Data

Data presented in Omron Company websites, catalogs and other materials is provided as a guide for the user in determining suitability and does not constitute a warranty. It may represent the result of Omron's test conditions, and the user must correlate it to actual application requirements. Actual performance is subject to the Omron's Warranty and Limitations of Liability.

Change in Specifications

Product specifications and accessories may be changed at any time based on improvements and other reasons. It is our practice to change part numbers when published ratings or features are changed, or when significant construction changes are made. However, some specifications of the Product may be changed without any notice. When in doubt, special part numbers may be assigned to fix or establish key specifications for your application. Please consult with your Omron's representative at any time to confirm actual specifications of purchased Product.

Errors and Omissions

Information presented by Omron Companies has been checked and is believed to be accurate; however, no responsibility is assumed for clerical, typographical or proofreading errors or omissions.

Meanings of Signal Words

● Symbols and the meanings for safety precautions described in this manual.

In order for the product to be used safely, the following indications are used in this book to draw your attention to the cautions. The cautions with the indications describe the important contents for safety.

WARNING	Indicates a potentially hazardous situation which, if not avoided, will result in minor or moderate injury, or may result in serious injury or death. Additionally there may be significant property damage.

CAUTION	Indicates a potentially hazardous situation which, if not avoided, may result in minor or moderate injury or in property damage.

● Meanings of Alert Symbols

The following alert symbols are used in this manual.

	Indicates general prohibitions for which there is no specific symbol.
	Indicates the possibility of electric shock under specific conditions.
	Indicates the possibility of explosion under specific conditions.
	Indicates the possibility of laser radiation.
	Indicates the possibility of injury by high temperature under specific conditions.

● Alert statements in this Manual

The following alert statements apply to the products in this manual. Each alert statement also appears at the locations needed in this manual to attact your attention.

WARNING	
This product must be used according to the instruction manual. Failure to observe this may result in impairment of functions and performance of the product	
This product is not designed or rated for ensuring safety of persons. Do not use it for such purposes.	
Never connect the AC power supply with this product. When the AC power supply is connected, it causes the electric shock and a fire.	
A lithium battery is built into the Controller and may occasionally combust, explode, or burn if not treated properly. Dispose of the Controller as industrial waste, and never disassemble, apply pressure that would deform, heat to 100°C or higher, or incinerate the Controller.	
Since camera that can be connected with this product emits a visible light that may have an adverse effect on the eyes, do not stare directly into the light emitted from the LED. If a specular object is used, take care not to allow reflected light enter your eyes.	
Do not touch the terminals while the power supply is ON. Doing so may result in electrical shock.	
Please take external safety measures so that the system as a whole should be on the safe side even if a failure of a sensor controller or an error due to an external factor occurred. An abnormal operation may result in serious accident.	
Please take fail-safe measures on your side in preparation for an abnormal signal due to signal conductor disconnection and/or momentary power interruption. An abnormal operation may result in a serious accident.	

CAUTION	
Danger of burns. Do not touch the case while the LED is ON or just after power is turned OFF, since it remains extremely hot.	

Precautions for Safe Use

● Installation Environment

- Do not use the product in areas where flammable or explosive gases are present.
- Install the product so that air can flow freely through its cooling vents.
- Clean the vent hole and discharge opening to prevent dust or particles from blocking them. Blocked cooling vents or discharge opening of the fan increasing heat inside, causing malfunction of the product.
- Do not install the product close to high-voltage devices and power devices in order to secure the safety of operation and maintenance.
- Make sure to tighten all installation screws securely.

● Power Supply and Wiring

- Make sure to use the product with the power supply voltage specified by this manual.
- Use the specified wire size (AWG10 to 16).
- Keep the power supply wires as short as possible (Max.2m).
- Use a DC power supply with safety measures against high-voltage spikes(safety extra low-voltage circuits on the secondary side).
- Do the following confirmations again before turning on the power supply.
  - Is the voltage and polarity of the power supply correct? (24VDC)
  - Is not the load of the output signal short-circuited?
  - Is the load current of the output signal appropriate?
  - Is not the mistake found in wiring?
  - Is the voltage and polarity of the encoder power (ENC0 VDD / ENC0 GND / ENC1 VDD / ENC1 GND) supply? (5VDC)

● Ground

- The power supply circuit of the FH sensor controller is insulated from the internal circuit.
- Be sure to use a base to install the camera connected with the FH Sensor Controller. Since the exclosure of the camera main body made of metals is short-circuited with the internal circuit, the internal circuit might be short-circuited with FG if no base is used, so that failures or malfunctions may be caused.
- Perform Class D-class grounding (with a grounding resistance of 100Ω or less).
- Keep the ground line as short as possible by setting the grounding point as close as possible.
- Ground the FH Sensor Controller independently. If sharing the ground line with other devices or connecting it with a building beam, the controller might be adversely effected.
- Check wiring again before turning on the Controller.

● Other

- Use only the camera and cables designed specifically for the product. Use of other products may result in malfunction or damage of the product.
- Always turn OFF the Controller's power before connecting or disconnecting a camera or cable. Connecting the cable with power supplied may result in damage of the camera or peripheral devices.
- For the cable that is flexed repeatedly, use the robotic cable type (flexing resistance type) to prevent damages.
- Do not apply torsion stress to the cable. It may damage the cable.
- Secure the minimum bending radius of the cable. Otherwise the cable may be damaged.
- Do not attempt to dismantle, repair, or modify the product.
- Should you notice any abnormalities, immediately stop use, turn OFF the power supply, and contact your OMRON representative.
- The FH Sensor Controller and camera case are hot while power is supplied or directly after the controller is turned off. Do not touch the case.
- Be sure to dispose of the product as industrial waste.

- Do not drop, impose excessive vibration or shock on the product. Doing so may result in malfunction or burning.
- Since a lithium battery is incorporated, there is a rare case when you are seriously injured due to firing or blowout.

Precautions for Correct Use

● Installation and Storage Sites

Install and store the product in a location that meets the following conditions:

- Surrounding temperature of 0 to 50°C (–20 to +65°C in storage)
- No rapid changes in temperature (place where dew does not form)
- Relative humidity of between 35 to 85 %
- No presence of corrosive or flammable gases
- Place free of dust, salts and iron particles
- Place free of vibration and shock
- Place out of direct sunlight
- Place where it will not come into contact with water, oils or chemicals

● Orientation of Product

- To keep proper ventilation, install the main unit only the direction as Instruction sheet and manual so that the ventilation holes are not blocked.

● Ambient Temperature

- To keep proper air flow, keep the top of the Controller 50 mm or more apart from other devices. Install the controller with a clearance of 30 mm on the right, left side, and 15 mm for rear planes.
- Do not install the product immediately above significant heat sources, such as heaters, transformers, or large-capacity resistors.
- Do not let the ambient temperature exceed 50°C (122°F).
- Provide a forced-air fan cooling or air conditioning if the ambient temperature is near 50°C (122°F) so that the ambient temperature never exceeds 50°C (122°F).

● Noise Resistance

- Do not install the product in a cabinet containing high-voltage equipment.
- Do not install the product within 200 mm of power cables.

● Component Installation and Handling

- Touching Signal Lines
  To prevent damage from static electricity, use a wrist strap or another device for preventing electrostatic discharges when touching terminals or signal lines in connectors.
- Handling a USB Memory/SD card (Reference: ▶ Using External Memory Devices (FH and FZ5-11□□ Only) (p.256))
  To remove a USB memory or SD card, make sure that data is not being read or written to it.
- For USB memory, the LED flashes while data is being read or written, so make sure that it is lit steadily before removing the memory.
- For SD card, the SD BUSY LED flashes while data is being read or written, so make sure that it is turned OFF before removing the memory.
- Turning OFF the Power
  Do not turn OFF the power while a message is being displayed indicating that processing is being performed. Data in memory will be corrupted, and the product may not operate correctly the next time it is started.

● Maintenance

- Turn OFF the power and take safety precautions before conducting inspections. Electrical shock can result from attempting safety inspections with the power turned ON.
- Clean the lens with a lens-cleaning cloth or air brush.
- Lightly wipe off dirt with a soft cloth.
- Dirt on the CCD must be removed using an air brush.
- Do not use thinners or benzene.

● Communication with High-order Device

- After confirming that this product is started up, communicate with the high-order device. When this product has started up, an indefinite signal may be output from the high-order interface. To avoid this problem, clear the receiving buffer of your device at initial operations.

● Fail-Safe Measures

- If you wish to operate a stage and/or a robot using a measurement result from a sensor controller(e.g. axis movement amount output based on calibration/alignment measurement),always take safety measures so that the measurement result should be checked by the stage/robot if it is within the range of movement of the stage/robot before operation.
- On a sensor controller side, supplementarily use operations and branches of the sensor controller to configure a check flow such as “data should not be externally provided if the data is in a range from -XXXXX to XXXXX “ based on the stage/robot’s range of movement.

MEMO

Before Operation

This section describes the basic flow and preparations that are necessary to start operation.

Basic Mechanism for Measurements

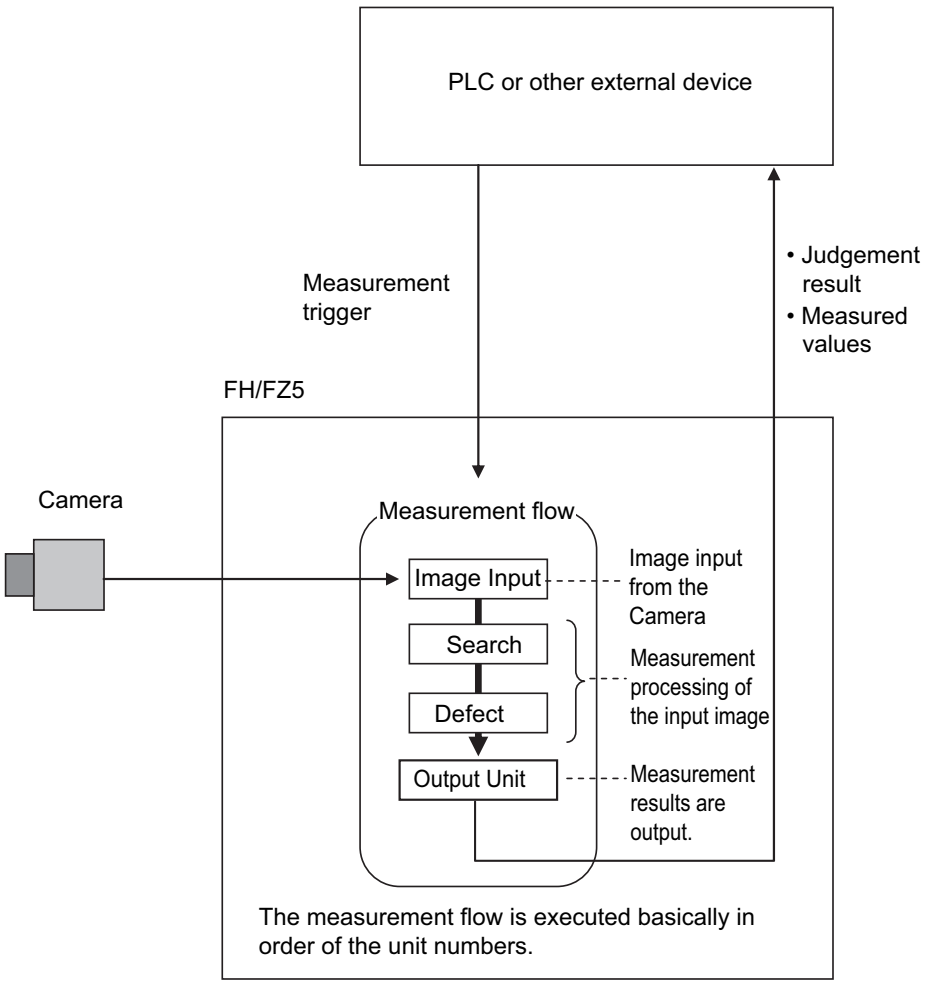
An FH/FZ5-series Sensor Controller uses pre-built packages that contain all the processing tasks (for image input, measurement processing, displays, outputs, etc.) that are required for vision inspections.
You arrange these packaged processes in order of execution of the vision inspection.
An FH/FZ5-series Controller executes vision inspections according to user-created flows.

Note

In the FH/FZ5, a flow that contains packaged processes that are arranged in order of execution of processing items and image processing is called a measurement flow.
Processing items and measurement flows can have more than one setting. You can switch the setting based on the scene to inspect. (Reference: ▶ Setting Scenes (Measurement Flows) (p.47))

Concept of Measurement Processing

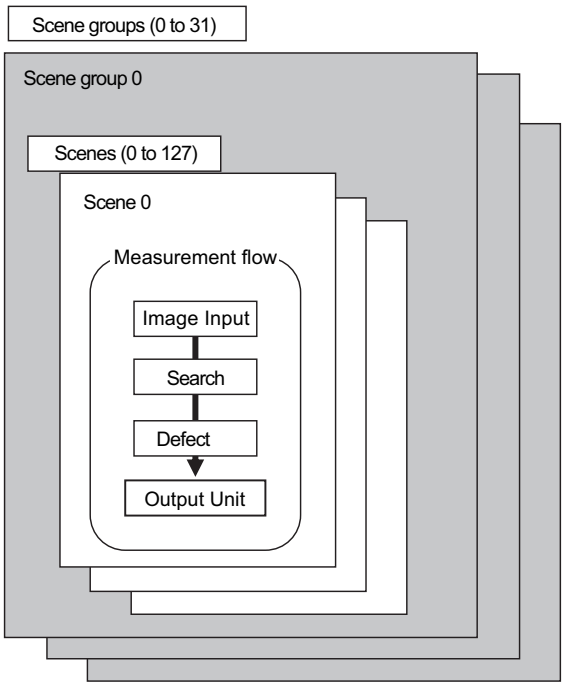
When the FH/FZ5 receives a measurement trigger from the PLC or other external device, the image input from a Camera, measurement processing, and output of measurement results (e.g., OK/NG judgement results) are executed in the order that those processing items are registered in the measurement flow.



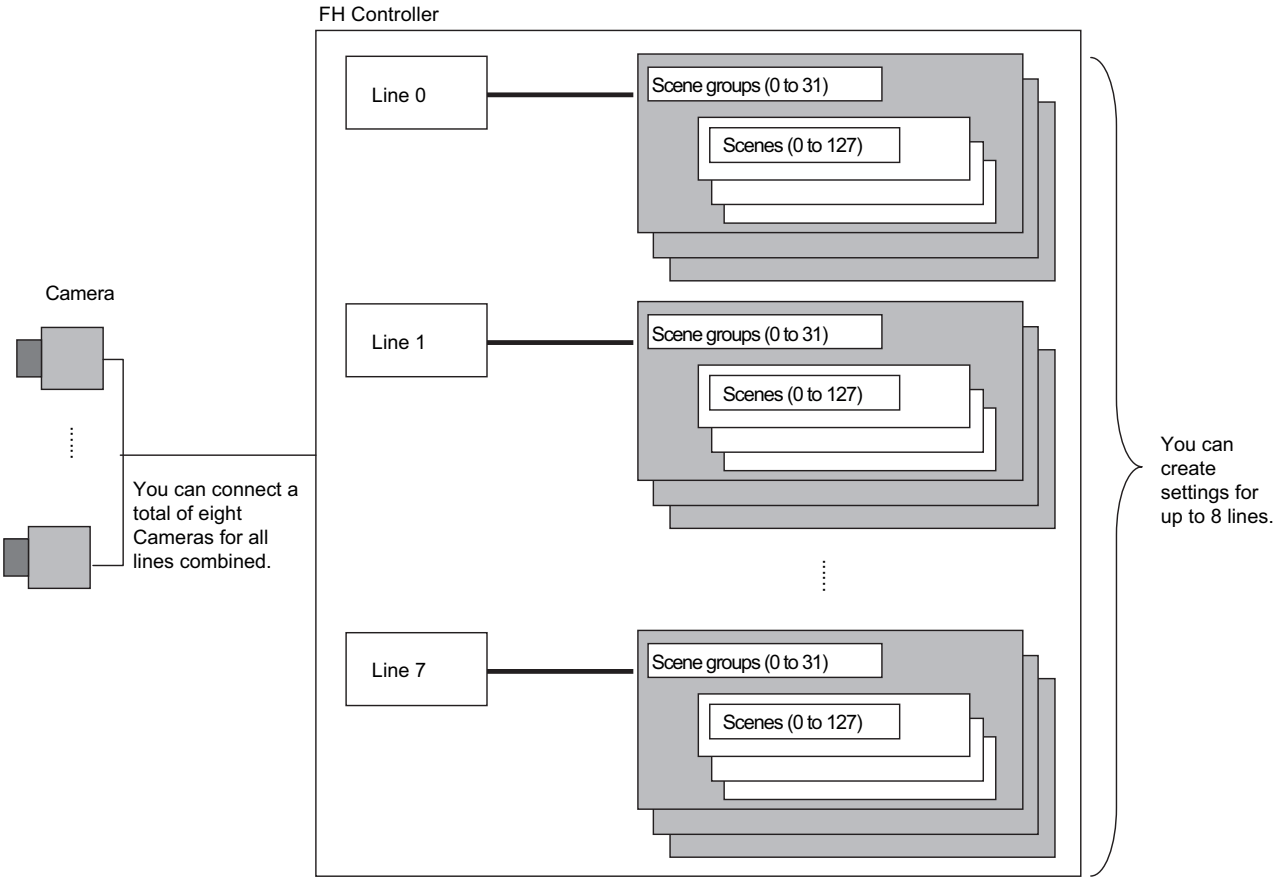
- In the measurement flow, you can change the processing to execute based on the inspection results or input conditions of the vision inspection.
- You can use macro processing to execute pre-packaged processing items and functions in the FH/FZ5 to create original programs. This allows you to create original measurement processes, display processing, input and output processing, and settings dialog boxes that are custom-tailored to your application.

Support for a Variety of Scenes and Measurement Lines

You can have more than one setting for a measurement flow, depending on the inspections. This is called a scene and you can set up to 128 scenes. (Reference: ▶ What Is a Scene? (p.48))
You can also combine more than one scene into a scene group. Management of scenes by scene groups increases the number of scenes that you can handle. (Reference: ▶ What Is a Scene Group? (p.51))



If you have more than one measurement line, the FH allows you to selectively execute up to eight measurement lines. This allows you to maintain settings for the scenes and scene groups for every line from 0 to 7.
You can connect up to eight Cameras to the FH, (four for the FZ5), and selectively control each Camera.
(Reference: ▶ Multi-line Random-trigger Mode (p.273))



Flow of Application

The following table shows the flow for using the FH/FZ5.

Procedure	Description	Reference
Preparations	Installation and Wiring	FH Instruction Manual FZ5 Instruction Manual
	↓	
	Turning ON Power	FH Instruction Manual FZ5 Instruction Manual
	↓	
	Language Selection in Dialog Box (only when the Sensor Controller is started for the first time)	Reference: ▶ Selecting the Language: [Language Setting] (p.292)
	↓	
	Main Window (Layout 0) Display	Reference: ▶ Adjustment Windows and Run Windows (p.68)
	↓	
	Camera Adjustments (Display the settings dialog box for a Camera Image Input processing item.)	Reference: ▶ Preparing the Controller and Cameras (p.70)
	↓	
	Select [Tool] – [System settings], and then under [Startup setting], set the settings for [Basic], [Communication], and [Operation mode].	Reference: ▶ Changing the System Environment (p.263)
	↓	
	Click the [Data save] button, and then select [Function] – [System restart].	Reference: ▶ Saving Settings before Turning OFF the Power and Restarting (p.43) Reference: ▶ Loading Settings Data from the Controller RAM Disk or an External Memory Device to the Sensor Controller (p.261)
	↓	
	Select [Tool] – [System settings], and then set the settings for [Camera], [Communication] and [Other].	Reference: ▶ Changing the System Environment (p.263)
	↓	
	Click the [Data save] button, and then select [Function] – [System restart].	Reference: ▶ Saving Settings before Turning OFF the Power and Restarting (p.43) Reference: ▶ Loading Settings Data from the Controller RAM Disk or an External Memory Device to the Sensor Controller (p.261)

Procedure	Description	Reference
↓		
Scene Editing	In the Main Window (layout 0), edit the measurement flow. • Register processing items. • Set the properties for each processing item.	Reference: ▶ Setting Scenes (Measurement Flows) (p.47)
	↓	
	Click the [Data save] button.	Reference: ▶ Saving Settings before Turning OFF the Power and Restarting (p.43) Reference: ▶ Loading Settings Data from the Controller RAM Disk or an External Memory Device to the Sensor Controller (p.261)
↓		
Testing	Execute test measurements. (In the Main Window (layout 0), click the [Measure] button.)	Reference: ▶ Performing Test Measurement/Starting Operation (p.67)
	↓	
	Adjust the parameters for each processing item.	Vision System FH/FZ5 Series Processing Item Function Reference Manual (Cat. No. Z341)
	↓	
	Click the [Data save] button.	Reference: ▶ Saving Settings before Turning OFF the Power and Restarting (p.43) Reference: ▶ Loading Settings Data from the Controller RAM Disk or an External Memory Device to the Sensor Controller (p.261)
↓		
Measuring (Operation)	In the Main Window (layout 0), click the [Switch layout] button, and then select [Main Window (Layout 1)].	Reference: ▶ Performing Test Measurement/Starting Operation (p.67)
	↓	
	In the Main Window (layout 1), check the communications with the PLC.	Vision System FH/FZ5 Series User's Manual for Communications Settings (Cat. No. Z342)
	↓	
	In the Main Window (layout 1), execute commands from the PLC, such as measurement trigger commands.	Vision System FH/FZ5 Series User's Manual for Communications Settings (Cat. No. Z342)
↓		
Management and Analysis	Save and analyze measurement data and images.	Reference: ▶ Useful Functions for Operation (p.109) Reference: ▶ Using Tools (p.131)

Types of Windows

There are 9 windows (with layout numbers 0 to 8) that you can use to set and operate an FH/FZ5.

Layout	Application
Layout 0	Layout 0 is the adjustment window by default. (Reference: ▶ Main Window (Layout 0): Adjustment Window (Default) (p.68))
Layout 1	Layout 1 is a run window by default. (Reference: ▶ Main Window (Layout 1): Run Window (Default) (p.69))
Layout 2 to layout 8	Layouts 2 to 8 are for user-defined purposes and are created as they are needed. (Reference: ▶ Arranging the Window Display (p.92)) For example, if more than one line is being controlled, you can set and use a separate window for each line. (Reference: ▶ Multi-line Random-trigger Mode (p.273))

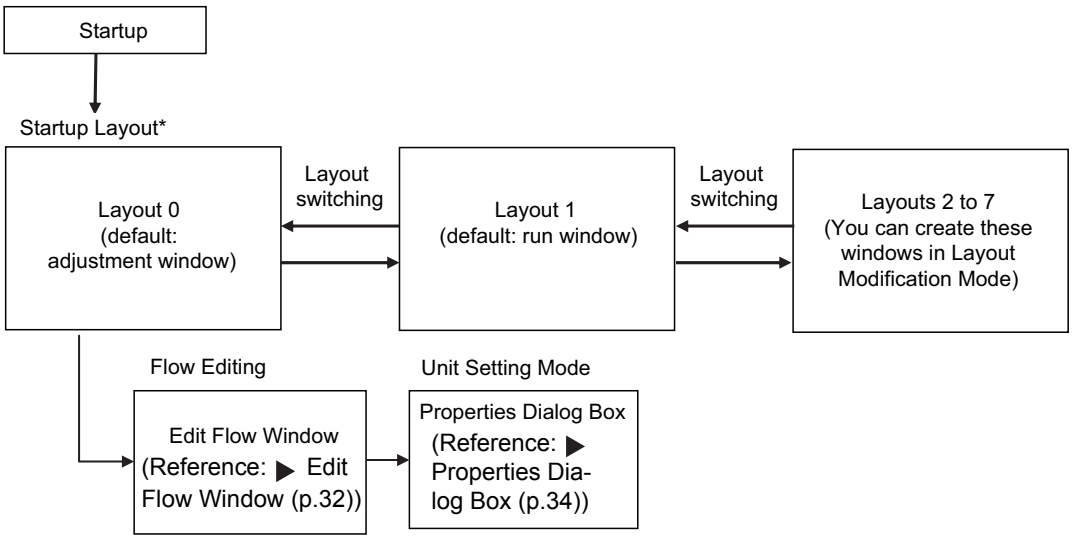
Note

If necessary, you can change layout 0 and layout 1 from the default adjustment window and run window and use them for other purposes.

IMPORTANT

If the operation mode is set to Multi-line Random-trigger Mode, you must create a layout for each line. (Reference: ▶ Multi-line Random-trigger Mode (p.273))

Overview of Window Navigation



* Select as follows: [System settings] − [Startup setting] − [Basic] tab − [Startup layout].
Default: [Layout 0] (both local and remote)

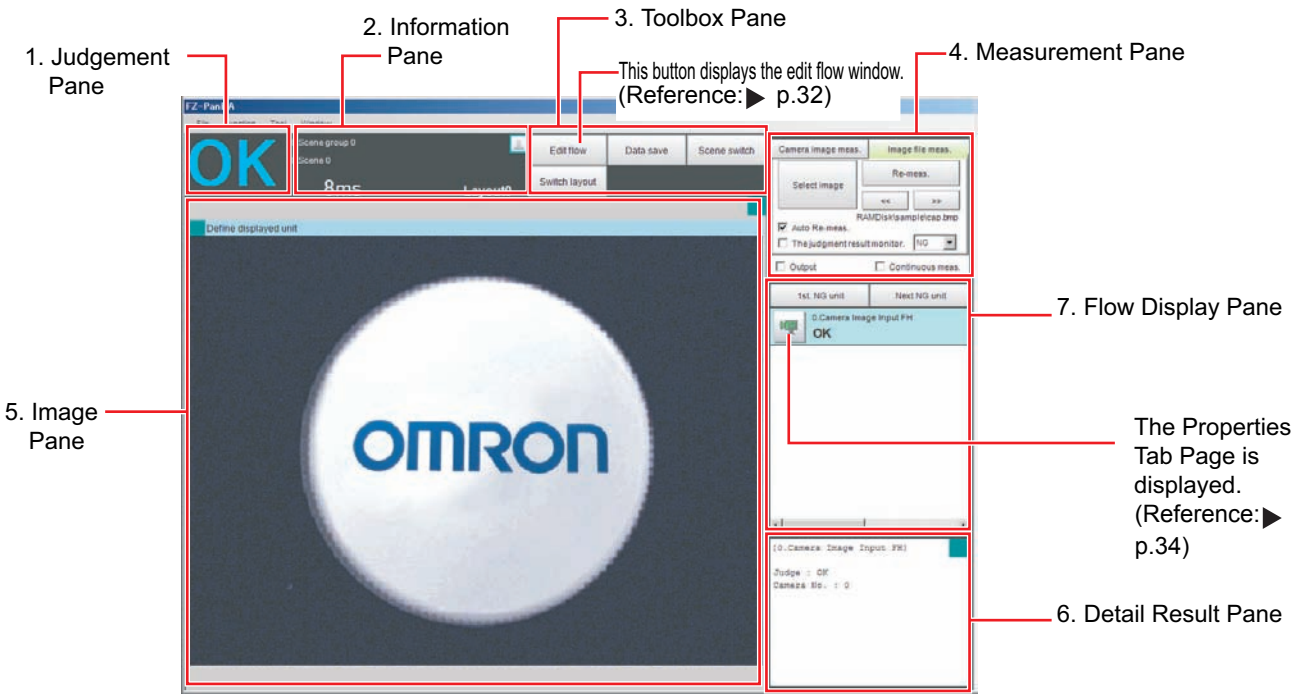
Main Window (Layout 0): Adjustment Window (Default)

You use this window to set up a measurement process and perform test measurements to confirm whether the measurements are processed as intended.
Based on the results of the test measurement, move to the property settings for each processing unit and make adjustments.
When the Main Window (layout 0) is being displayed, measurement results are not output and the RUN signal remains OFF. (Layout modification can be used to change the layout.)
You can manually execute or stop test measurements.
If the measurement results are stable, switch to the Main Window (layout 1) and perform measurements.

Note

Use the following procedure to change to Main Window (layout 1).
Select [Switch layout] from the [Function] menu (default layout).

Names of the Parts of the Main Window (Layout 0): Adjustment Window (Default)



1. Judgement Pane

- **Overall Judgement Result**
  The overall judgment (OK/NG) for the scene is displayed.
  This is the overall judgement result for the entire scene. If any processing unit that is reflected in the overall judgement returns NG, the overall result is NG.
  *1: The judgement results for individual processing units are displayed in the Detail Result Pane.

2. Information Pane

- **Layout:**
  The number of the currently display layout is displayed.
- **Processing time:**
  The time required for the measurement process is displayed.
- **Scene group name and scene name:**
  The scene group number and the scene number of the currently displayed scene are displayed.

3. Toolbox Pane

- **[Edit flow]:**
  This button starts the edit flow window that is used to edit flows. (Reference: ▶ Edit Flow Window (p.32))
- **[Data save]:**
  This saves settings data to the flash memory in the Controller. If you make any changes to the settings, make sure to click this button to save the settings.
- **[Switch scene]:**
  This button switches between scene groups and scenes:
- **[Switch layout]:**
  This button switches the layout number.

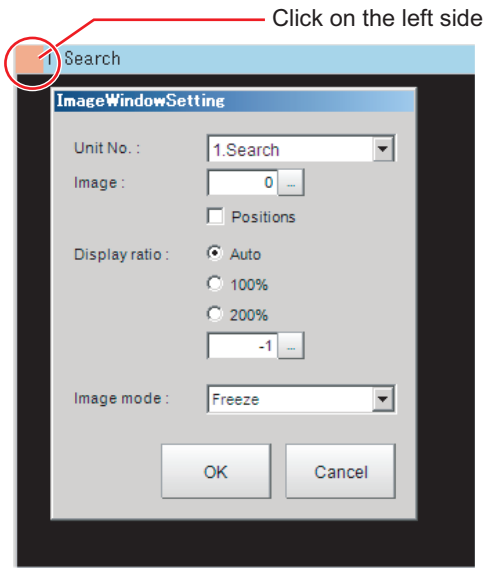
4. Measurement Pane

- **Camera image meas.:**
  This button performs a test measurement on the Camera image.
- **Image file meas.:**
  This button re-measures saved images.
- **Output:**
  Select this option to output the measurement results of test measurements that are performed in the Adjustment Window. Do not select this option if you need to perform test measurements on the Sensor Controller but do not need to output the results externally.
  This setting menu is used for temporary setting changes while the Main Window is displayed. When the scene or layout is changed, the "Output" setting in the Layout setup is applied without saving the "Output" setting in the Measurement Pane. Use as appropriate for your application.
  (Reference: ▶ Setting the Behavior of Output Signals for Each Layout (Layout Settings) (p.90))
- **Continuous meas.:**
  Select this option to continuously perform test measurements in the Adjustment Window.
  Select the [Continuous meas.] option and click [Measure] to repeatedly execute measurements.

5. Image Pane

This pane shows the measured image.
It also displays the selected processing unit name or the [Define displayed unit].

- Click on the left side of the processing unit name to display the Properties Dialog Box for the Image Pane.



On this dialog box, you can change the image mode and other contents that are displayed in the Image Display Pane.
(Reference: ▶ Changing the Image Mode and Other Display Contents (p.92))

- Click the upper corner of the Image Display Pane to display the [Add Image Window] button and [Alignment] button. This allows you to display more than one pane side by side.
(Reference: ▶ Arranging Window Elements (Layout Modification) (p.78))

6. Detail Result Pane

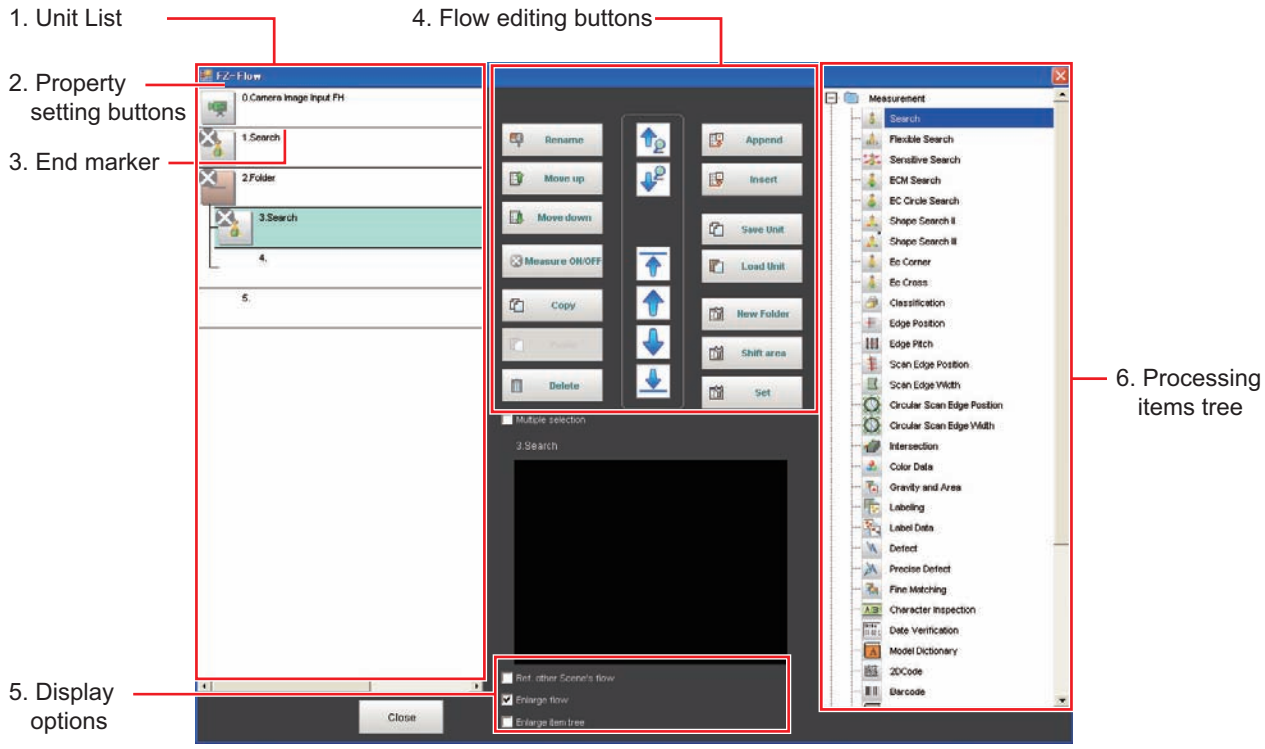
This pane displays the results of test measurements.

7. Flow Display Pane

This pane displays the contents of the measurement process (the processing that is set in the measurement flow).
Click the icon of any processing item to display the Properties Dialog Box to set processing item parameters and other settings. (Reference: ▶ Properties Dialog Box (p.34))
You can use the following buttons to move to the next processing unit with an NG judgement result.
[1st NG unit]: Moves to the first processing unit with an NG judgement.
[Next NG unit]: Moves to the next processing unit with an NG judgement.

Edit Flow Window

The edit flow window is used to create measurement flows. You use the edit buttons to arrange processing units in a scene, or to add or delete units.
The right side shows the processing items that can be used as parts in the measurement flow. The left side shows the scene as the measurement flow.
When a measurement trigger is input, processing is executed in sequence starting from the top of the measurement flow.
Click the button for any processing item that is set in the measurement flow, or click the [Set] button to open the Properties Dialog Box.



1. Unit List

The processing units included in the flow are displayed in a list.
You can create a flow for a scene by adding processing items to the unit list.

2. Property Setting Buttons

The Properties Dialog Box is displayed and allows you to make detailed settings.

3. End Marker

This marker indicates the end of the flow.

4. Flow Editing Buttons

You can use these buttons to change the order of processing units within the scene or to delete processing units.
(Reference: ▶ Editing Processing Units in Scenes (p.53))

5. Display Options

- **Enlarge flow**
  When this option is selected, the Unit list flow (1) is displayed with large icons.
- **Enlarge item tree**
  When this option is selected, the processing item tree (6) is displayed with large icons.
- **Ref. other Scene's flow**
  When this option is selected, other scene flows within the same scene group can be referred to.

6. Processing Items Tree

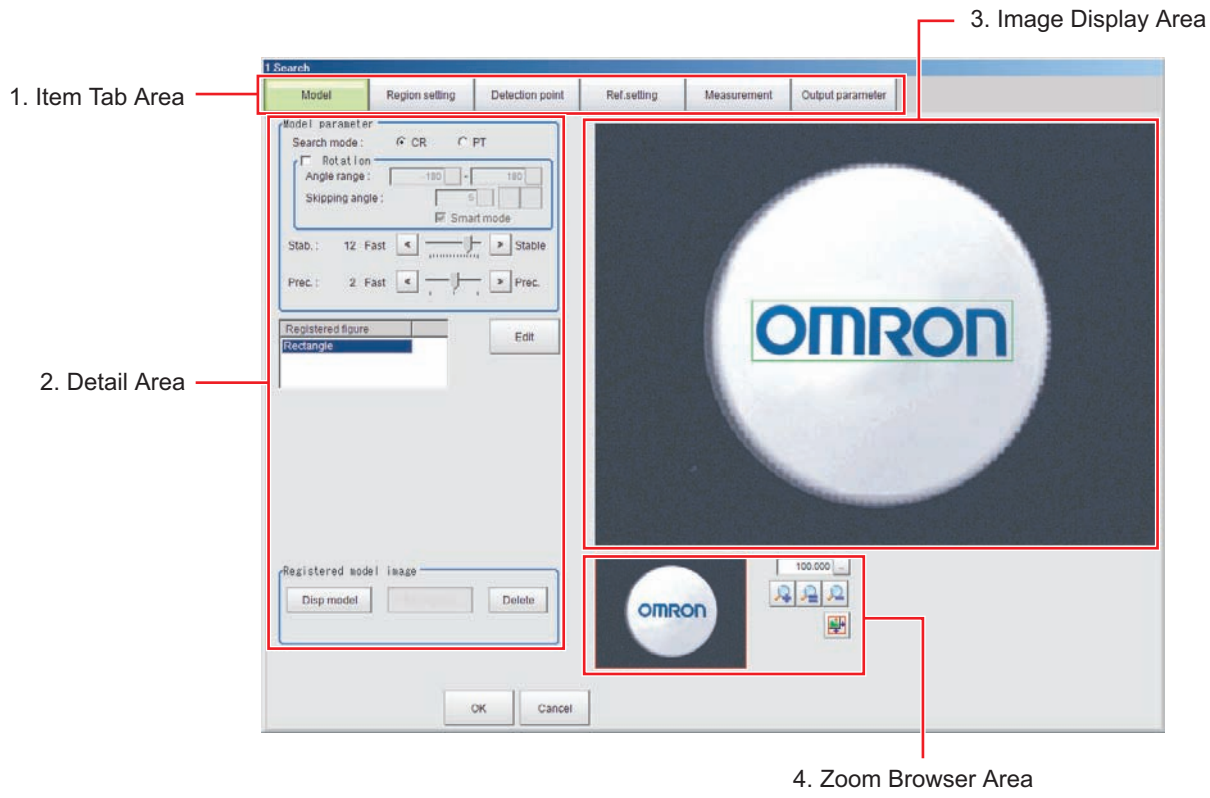
This area allows you to select an item to add to the flow. The items are displayed in tree format that is organized by type.

Click the [+] button for each item to display the lower level items. Click the [-] button for each item to hide the lower level items.

When [Ref.other Scene's flow] is selected, the scene select box and other scene flows are displayed.

Properties Dialog Box

This dialog box is used to set measurement parameters, judgement conditions, and other conditions for processing items that are registered as processing units in the measurement flow.



1. Item Tab Area

The parameters for the processing unit that is currently being set are displayed. Settings are performed starting with the item on the left.

2. Detail Area

This area is used to set detail items.

3. Image Display Area

This area shows Camera images, figures, positions, and other graphic information.

4. Zoom Browser Area

Use this area to zoom in and out from the displayed image.

Main Window (Layout 1): Run Window (Default)

This window is used to perform actual measurements. The measurement results are output through the communications interfaces.
If the Controller is ready to perform measurements, the RUN signal turns ON.

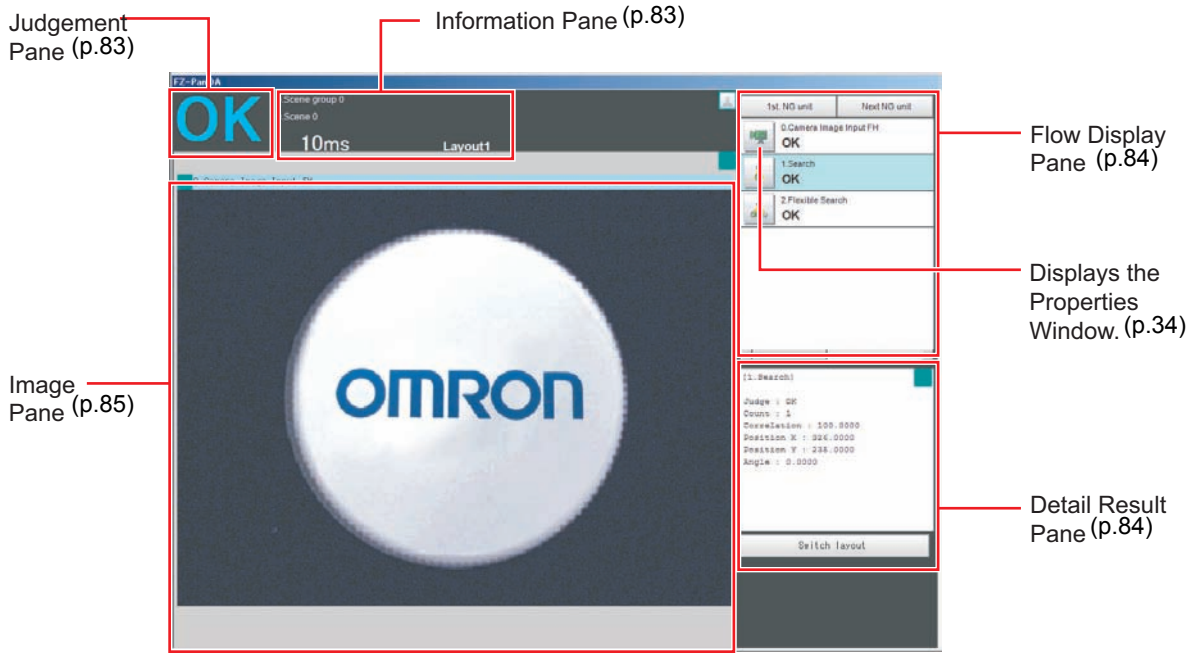
When this window is displayed, the FH/FZ5 is ready to perform measurements.
If a measurement trigger is input from an external device, the FH/FZ5 begins execution of measurement processing according to the contents and sequence that is set in the measurement flow.

Note

Use the following procedure to change to Main Window (layout 0).
Select [Switch layout] from the [Function] menu (default layout).

Names of the Parts of the Main Window (Layout 1): Run Window (Default)

The functions of each display window are the same as for the Main Window (layout 0).
The Toolbox Pane and Measurement Pane that are displayed on the Main Window (layout 0) are not displayed.
(Layout modification can be used to change the layout.)



Menu List

The menu for the Main Window is described below.

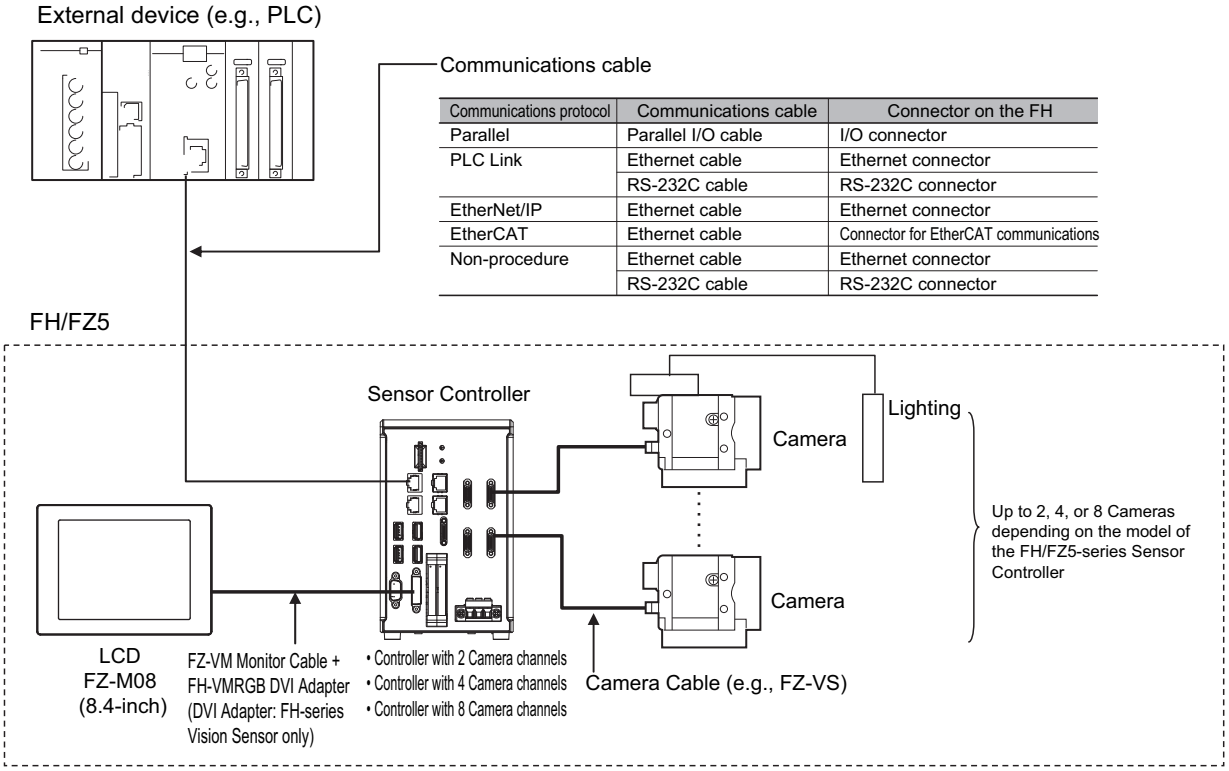
Main menu	Command	Description	Reference
File	End	Exits the FH/FZ5 System. After you exit the system, restart the system or turn OFF the power supply.	---
Function	Measure	Executes a test measurement (single or continuous measurement), or a re-measurement.	Reference: ▶ Executing Test Measurements (p.74) Reference: ▶ Remeasuring Saved Images (p.109)
	Scene switch	This button switches between scene groups and scenes:	Reference: ▶ Changing the Scene or Scene Group (p.60)
	Scene maintenance	Used to copy, delete, and rename scenes and scene groups.	Reference: ▶ Editing Scenes (p.62) Reference: ▶ Editing Scene Groups (p.64)
	Edit flow	Used to edit measurement flows.	Reference: ▶ Editing Processing Units in Scenes (p.53)
	Switch layout	Changes the layout number to display.	Reference: ▶ Arranging Window Elements (Layout Modification) (p.78)
	Clear measurement	Clears the measurement data.	Reference: ▶ Clearing Measurement Results (p.122)
	Clear logging image	Clears the images that are logged in the Controller.	Reference: ▶ Clearing Saved Images (p.123)
	Screen capture	Captures the screen that is being displayed.	Reference: ▶ Capturing Screen Images (p.123)
	Saving last logging image	Logs the most recent input images that are being displayed.	Reference: ▶ Logging Measurement Values and Measurement Images (p.113)
	Data save	Saves settings data to the flash memory in the FH/FZ5.	Reference: ▶ Saving Settings before Turning OFF the Power and Restarting (p.43) Reference: ▶ Saving and Loading Data (p.253)
	Save to file	Saves the settings data, logged images, and logged data to a file. It also copies files between memory devices.	Reference: ▶ Saving and Loading Data (p.253)
	Load from file	Loads a settings file.	
	System initialization	Restores the Controller to the default settings.	Reference: ▶ Initializing the Controller (p.45)
	System restart	Restarts the Controller.	Reference: ▶ Saving Settings before Turning OFF the Power and Restarting (p.43) Reference: ▶ Saving and Loading Data (p.253)

Main menu	Command	Description	Reference
Function	Transfer data	Transfers (copies) the settings for line 0 to line 1, and then changes to the Non-stop Adjustment Mode.	Reference: ▶ Non-stop Adjustment Mode (p.279)
	Non-stop data transfer	Transfers (copies) the settings for line 1 to line 0 in Non-stop Adjustment Mode without affecting measurements.	
	Operation log	Saves the operation log of the Controller to a special file.	Reference: ▶ Using the Operation Log (p.125)
	System information	Displays the Controller's model and software version.	Reference: ▶ Checking System Information: [System Information] (p.300)
Tool	NG analyzer	Used to analyze NG causes based on image data.	Reference: ▶ Using NG Analyzer (p.132)
	System settings	Sets settings that are related to the system environment of the Controller. Includes settings such as the startup settings, Camera settings, and communications settings.	Reference: ▶ Changing the System Environment (p.263)
	Security settings	Used to set accounts and limit access to the Controller.	Reference: ▶ Using Accounts (p.150)
	User data tool	Sets the initial value for user data that can be shared across scenes and across scene groups.	Reference: ▶ Using the User Data Tool (p.138)
	Settings download and upload tools	Used to download and upload a list of set values for scene data as a CSV file.	Reference: ▶ Outputting a List of Scene Data Set Values (p.143)
	Image file save	Saves logged images and image files that are saved in the Controller.	Reference: ▶ Saving Image Files to a RAM Disk or External Memory Device (p.146)
	Registered image Manager	Used to register the most recent input image, a logged image that is saved in the Controller, or an image file and load it as a measurement image.	Reference: ▶ Using the Registered Image Manager (p.148)
	Customize I/O command	Creates a user-defined communications unit (i.e., a custom communications command) for the FH/FZ5 to execute in response to an input from the PLC or other external device.	Reference: ▶ Using Communication Command Macro (p.164)
	Flow Viewer	Shows the flow of processing for the measurement flow that is currently being edited.	Reference: ▶ Flow Viewer (p.165)
	Calibration support tool	Graphically displays the relationship of the positions of the Camera coordinates and the actual coordinates.	Reference: ▶ Calibration Support Tool (p.167)
	Update standard position tool	Used to set or change the reference position for more than one processing unit that you specify in the measurement flow.	Reference: ▶ Updating the Reference Position (p.137)
	Conversion scene group data tool	Creates a scene group that has more than 129 scenes.	Reference: ▶ Creating Scene Group Data for More Than 128 Scenes (p.140)
	Custom dialog tool	Used to create customized dialog boxes for setting processing items.	Reference: ▶ Custom Dialogs (p.94)
	Custom dialog	Displays a dialog box that is created with the custom dialog tool.	
	Scene Group Saving Destination Settings	Changes the location for saving scene group data to the sensor controller's memory or external storage.	Reference: ▶ Saving Scene Groups in External Memory (p.168)
Window	Layout modification	Used to change any layout from layout 0 to layout 8.	Reference: ▶ Arranging the Window Display (p.92)
	Layout setup	Sets whether to use the RUN signal output and external outputs for layouts 0 to 8.	

System Configuration

The FH/FZ5 is a Vision Sensor that uses a controller to process measurements of objects that are imaged with a Camera.
You connect an LCD for operations and monitoring, and various Cameras to the FH/FZ5-series Sensor Controller.
You connect external devices, such as a PLC or a computer, through a parallel, Ethernet, or RS-232C cable.

You can connect up to eight Cameras, depending on the model of the Controller.
To measure more than one line with a single Sensor Controller, you assign the Camera for the measurements to each line beforehand, and switch between Cameras during the measurement flow.



Communications protocol	Communications cable	Connector on the FH
Parallel	Parallel I/O cable	I/O connector
PLC Link	Ethernet cable	Ethernet connector
	RS-232C cable	RS-232C connector
EtherNet/IP	Ethernet cable	Ethernet connector
EtherCAT	Ethernet cable	Connector for EtherCAT communications
Non-procedure	Ethernet cable	Ethernet connector
	RS-232C cable	RS-232C connector

*1. FZ-MEM2G or FZ-MEM8G USB Memory is sold separately.

Basic Configuration Models

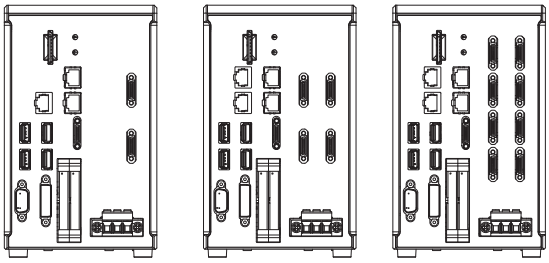
FH Basic Configuration

* Items indicated with an asterisk are dedicated items, and cannot be substituted.

* Sensor controller

The Controller performs the image processing specified by the user settings and outputs the measurement results.

Camera 2ch type	Camera 4ch type	Camera 8ch type
FH-3050	FH-3050-10	FH-3050-20
FH-1050	FH-1050-10	FH-1050-20



* Camera cable

Camera cable
FZ-VS (2m, 5m,10m, min. bending radius: 69mm)
Bend resistant camera cable
FZ-VSB (2m, 5m, 10m min. bending radius: 69mm)
Right-angle camera cable
FZ-VSL (2m,5m,10m min. bending radius: 69mm)
Long-distance camera cable
FZ-VS2 (15m min. bending radius: 93mm)
Long-distance Right-angle camera cable
FZ-VSL2 (15m min. bending radius: 93mm)

* Camera

Detects workpieces as images.
Standalone camera
FZ-SC/FZ-S/
FZ-SC2M/FZ-S2M/
FZ-SFC/FZ-SF/
FZ-SPC/FZ-SP/
FZ-SHC/FZ-SH
FH-SC/SM
FH-SC02/SM02
FH-SC04/SM04
FZ-SC5M2/FZ-S5M2
FH-SC12/SM12

Intelligent compact camera
FZ-SQ010F/FZ-SQ050F/
FZ-SQ100F/FZ-SQ100N

Intelligent camera
FZ-SLC15/FZ-SLC100
Automatic focus camera
FZ-SZC15/FZ-SZC100

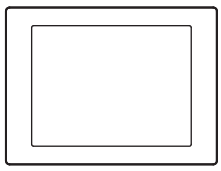
Camera with Lighting

Lighting controller
FLV-TCC1/FLV-TCC4/FL-TCC1

* LCD monitor

Use the monitor to check images and display the condition-setting menus.

FZ-M08 (8.4-inch)



* Monitor cable

FZ-VM (2m, 5m, min. bending radius: 75mm)

* Monitor conversion cable

FH-VMRGB

Peripheral Device

* USB memory
FZ-MEM2G

Power Supply

The power supply connected to FH Sensor Controller varies depending on the number of connected cameras and types for various

FZ5 Basic Configuration

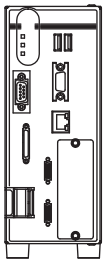
FZ5-L35□

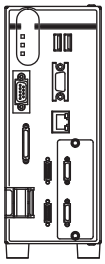
* Items indicated with an asterisk are dedicated items, and cannot be substituted.

* Sensor Controller

The Controller performs the image processing specified by the user settings and outputs the measurement results.

Camera 2ch type	Camera 4ch type
FZ5-L350	FZ5-L350-10
FZ5-L355	FZ5-L355-10





* Camera cable

Camera cable
FZ-VS (2m, 5m,10m, min. bending radius: 69mm)
Bend resistant camera cable
FZ-VSB (2m, 5m, 10m min. bending radius: 69mm)
Right-angle camera cable
FZ-VSL (2m,5m,10m min. bending radius: 69mm)
Long-distance camera cable
FZ-VS2 (15m min. bending radius: 93mm)
Long-distance Right-angle camera cable
FZ-VSL2 (15m min. bending radius: 93mm)

* Camera

Detects workpieces as images.

Standalone camera
FZ-SC/FZ-S/
FZ-SC2M/FZ-S2M/
FZ-SC5M2/FZ-S5M2/
FZ-SFC/FZ-SF/
FZ-SPC/FZ-SP/
FZ-SHC/FZ-SH

Intelligent compact camera
FZ-SQ010F/FZ-SQ050F/
FZ-SQ100F/FZ-SQ100N

Intelligent camera
FZ-SLC15/FZ-SLC100

Automatic focus camera
FZ-SZC15/FZ-SZC100

Camera with Lighting

Lighting controller
FLV-TCC1/FLV-TCC4/FL-TCC1

Peripheral Device

* USB memory
FZ-MEM2G
FZ-MEM8G

Power Supply

Recommended Model
By OMRON Corporation
S8VS-12024 (FZ5-L35□)
S8VS-18024 (FZ5-L35□-10)

Input Device

Mouse, track ball
(Commercially available USB devices)

FZ5-6□□/11□□

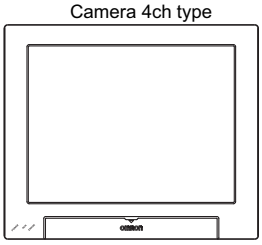
* Items indicated with an asterisk are dedicated items, and cannot be substituted.

*
Sensor controller

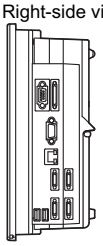
The Controller performs the image processing specified by the user settings and outputs the measurement results.

Camera 2ch type	Camera 4ch type
FZ5-110□	FZ5-110□-10
FZ5-60□	FZ5-60□-10

Camera 4ch type



Right-side view



Touch pen (standard accessory)

*
Camera cable

Camera cable
FZ-VS (2m, 5m,10m, min. bending radius: 69mm)
Bend resistant camera cable
FZ-VSB (2m, 5m, 10m min. bending radius: 69mm)
Right-angle camera cable
FZ-VSL (2m,5m,10m min. bending radius: 69mm)
Long-distance camera cable
FZ-VS2 (15m min. bending radius: 93mm)
Long-distance Right-angle camera cable
FZ-VSL2 (15m min. bending radius: 93mm)

*
Camera

Detects workpieces as images.

Standalone camera
FZ-SC/FZ-S/
FZ-SC2M/FZ-S2M/
FZ-SFC/FZ-SF/
FZ-SPC/FZ-SP/
FZ-SHC/FZ-SH
FZ-SC5M2/FZ-S5M2
Intelligent compact camera
FZ-SQ010F/FZ-SQ050F/
FZ-SQ100F/FZ-SQ100N
Intelligent camera
FZ-SLC15/FZ-SLC100
Automatic focus camera
FZ-SZC15/FZ-SZC100

Camera with Lighting

Lighting controller
FLV-TCC1/FLV-TCC4/FL-TCC1

Peripheral Device

* USB memory
FZ-MEM2G
FZ-MEM8G

Power Supply

Recommended Model by OMRON:
S8VS-18024

Power Supply

Mouse, track ball
(Commercially available USB devices)

Functional Comparison between the FH-series and FZ5-series Controllers

The following table shows the differences between the FH and FZ5.

Controller model / Function	FH Series			FZ5 Series			
	FH-□050-20	**FH-□050-10**	**FH-□050**	**FZ5-L35□-10 FZ5-6□□-10**	**FZ5-L35□ FZ5-6□□**	**FZ5-11□□-10**	**FZ5-11□□**
Number of Camera connections	8[*2]	4	2	4	2	4	2
Operation mode[*1]	OK			---		OK	
Number of scenes (number of line process changes)	128[*3]			128[*3]			
Input interface							
EtherCAT	OK			---			
External memory							
USB memory	OK			OK			
SD memory card	OK			---			
Sysmac Studio support	OK			---			
Menu languages	Japanese, English, Simplified Chinese, Traditional Chinese, Korean, German, French, Spanish, and Italian			Japanese, English, Simplified Chinese, and Traditional Chinese[*4]		Japanese, English, Simplified Chinese, Traditional Chinese, Korean, German, French, Spanish, and Italian	

*1: With the quad-core CPU that is installed, you can set different operation modes to meet different applications.
Reference: ▶ Setting the Operation Mode (p.268)

*2: When using the FH-SC12/FH-SM12, you can connect up to four cameras. Do not connect more than five cameras when using only the FH-SC12/FH-SM12. When connecting the FH-SC12/FH-SM12 for use with another camera model, you can connect up to four FH-SC12/FH-SM12 cameras and also connect up to four cameras of the other camera model, for a total of eight connected cameras.

*3: The Conversion Scene Group Data Tool allows you to create a scene group of more than 129 scenes.

*4: Nine languages are selectable in the [Language setting] dialog box under [System Settings], however, only Japanese, English, Simplified Chinese, and Traditional Chinese can be used. If a language other than these four languages is selected, the menus will appear in English. The menus will also appear in English if system data with set languages other than these four languages is loaded.

Saving Settings before Turning OFF the Power and Restarting

Before you turn OFF the power supply to the Controller, save the data that you have set to the flash memory in the Controller.
The settings data in the flash memory of the Controller are loaded everytime the Controller is restarted, so make sure to use the [Data save] command to save the settings data to the flash memory in the Controller. Then, always restart the Controller. The settings data will not take effect unless you restart the Controller. If you do not save the data to the flash memory in the Controller, changes to the settings will be lost when you turn OFF the power supply.

1 **In the Main Window (layout 0), click the [Data save] button in the Toolbox Pane to save the settings data.**

2 **Turn OFF the power supply to the Controller and exit the system.**

Note

- Data That Is Saved
  Scene data and system data are saved in the Controller. Logged images and data that are saved in the RAM disk are not saved. Perform any of the following procedures to keep this data.
  - Copying Data Saved in the RAM disk to External Memory
    Reference: ▶ Saving Settings Data to the Controller RAM Disk or an External Memory Device (p.258)
  - Changing the Save Destination for Logged Data to External Memory
    Reference: ▶ Logging Measurement Values and Measurement Images (p.113)
- Using the Scene Group Function
  The scene data that is set for scene group 0 is saved in the Controller for models FZ5-L3□□ and FZ5-6□□. The scene data from scene groups 1 to 31 is saved to the external memory and overwrites any previously saved data. (For the FH or FZ5-11□□, all data is saved in the Controller by default.)

Restarting the Controller: [System Restart]

You can restart the Controller. Before restarting, back up any required data, such as scene data and system data.
Reference: ▶ Saving Settings Data to the Flash Memory and Restarting (p.257)

1 **Select [System restart] from the [Function] menu.**

The [System restart] dialog box is displayed.

2 **Click [OK].**

The Controller is restarted.
The settings data that was saved in flash memory is enabled.

IMPORTANT

- Immediately after starting up the FH/FZ5 Sensor Controller and immediately after changing scenes, there will be no input image. No input image is processed as the same color image as in the factory default state.
- If you open the Properties Dialog Box before inputting an image, click the Cancel button to close the dialog box. Pressing the OK button in the dialog box will change the setting to the same color camera setting as the factory default setting.

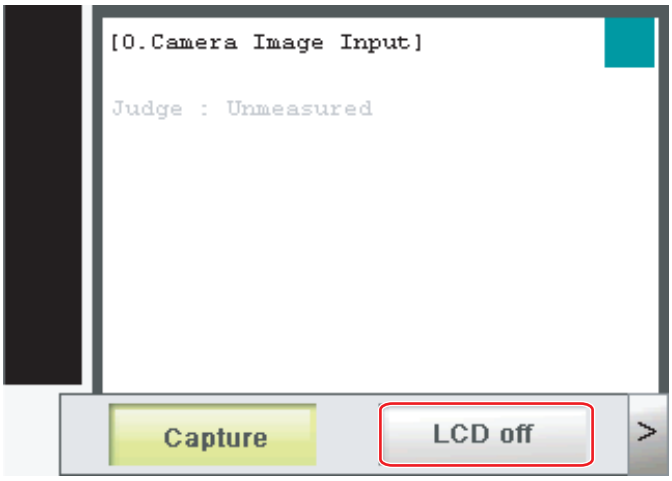
(Reference: ▶ The measurement NG (image mismatch) error will result when connecting a monochrome camera (p.832))

Turning OFF the LCD

(This function is supported only by the FZ5-600/1100-series LCD-integrated Controllers.)
You can turn OFF the LCD without turning OFF the Controller.

1 **Click the [LCD off] button that is located on the Measurement Manager Bar in the lower right corner of the window.**

A confirmation message is displayed.



2 **Click [OK].**

Power to the LCD is turned OFF.

Restoring Power to the LCD

(This function is supported only by the FZ5-600/1100-series LCD-integrated Controllers.)
Click the bottom edge of the LCD.
After a while, the LCD power will turn ON.

Initializing the Controller

You can restore all settings in the Controller to their default settings.

Initializing the Controller: [System initialization]

You can restore the Controller to the default settings. Before you initialize the Controller, back up any required data, such as scene data and system data.
Reference: ▶ Saving and Loading Data (p.253)

1 **Select [System initialization] from the [Function] menu.**

The [System initialization] dialog box is displayed.

2 **Click [Execute].**

A confirmation dialog box is displayed.

3 **Click [Yes].**

The Controller is initialized and restarted.

MEMO

Setting Scenes (Measurement Flows)

A measurement flow consist of a series of combined processing items and is called a scene. This section explains how to create and edit scenes.

What Is a Scene?

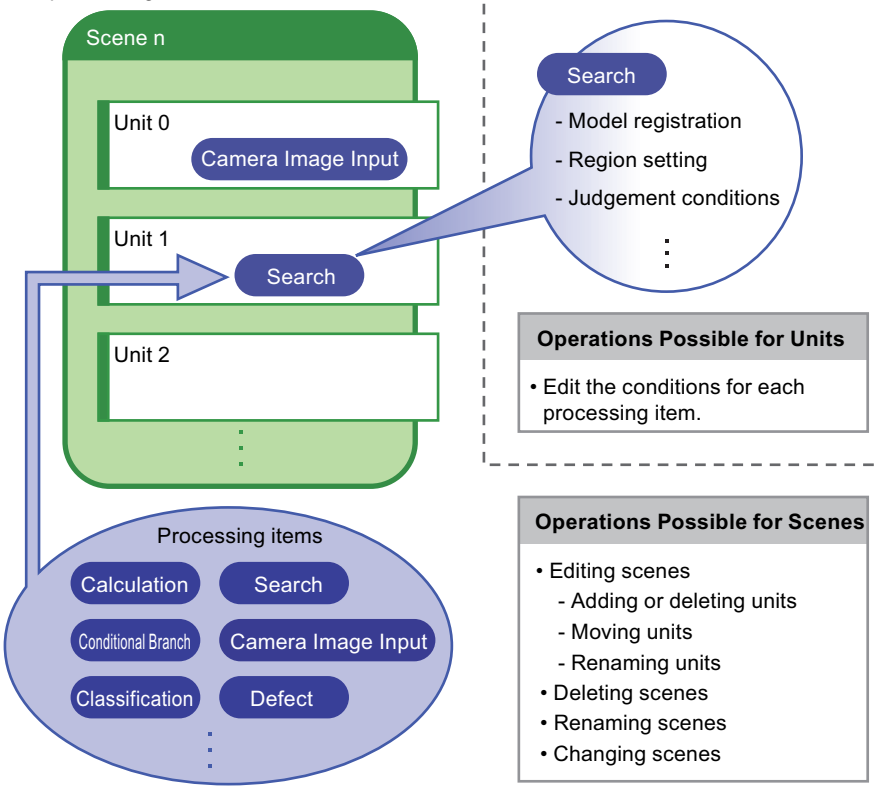
Processing items for use with various measurement objects and measurement objectives are provided in the Sensor Controller. By combining and executing these processing items, measurement adapted to the purpose can be implemented. A combination of processing items is called a scene. Scenes can be easily created by combining processing items that are suited to the measurement purpose from the list of processing items that are provided.

Editing Scenes

Set a processing item for each unit in the flow.

Editing Units

Set the conditions for each unit that forms the flow.



Changing Processes Using Scenes

You can create more than one scene. For example, if you create scenes for each measurement object, such as scene 0 to inspect label ABC and scene 1 to inspect label XYZ, you can change the process smoothly just by changing the scene whenever the measurement object or measurement objective has changed.

Reference: ▶ Changing the Scene or Scene Group (p.60)

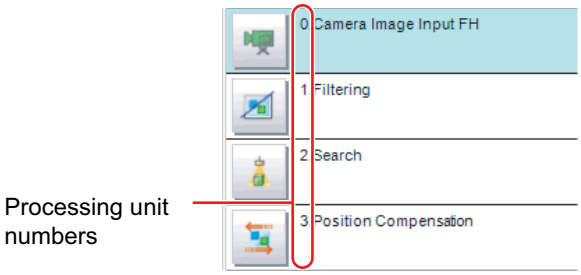
Normally you can set up to 128 scenes. If you require more than 128 scenes, you can create scene groups to easily manage them.

The Using the Conversion Scene Group Data Tool (p.140) allows you to create a scene group that has more than 129 scenes.

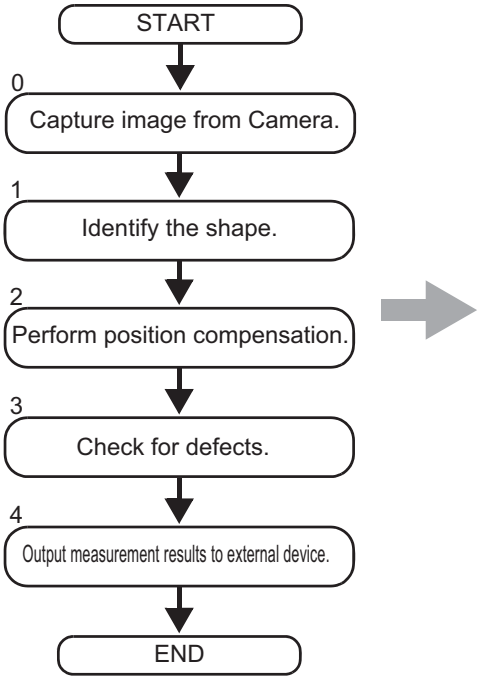
Reference: ▶ What Is a Scene Group? (p.51)

Example of a Scene

The processing items registered to a scene are called processing units. In the edit flow window where you create scenes, select the processing items required for measurement and add them to the flow. The number at the top of the processing unit is called the unit number. When the measurement trigger is input, processing is executed in the order of the processing unit numbers.



Example: Normal Measurement

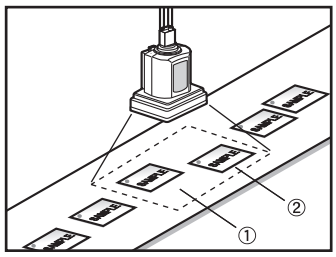


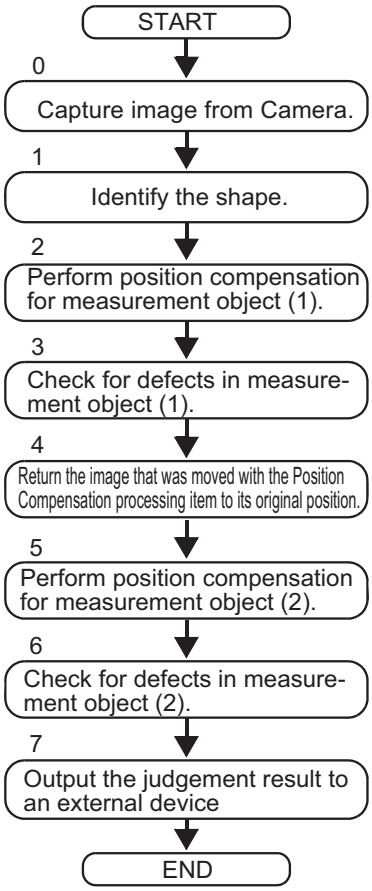
Visual Representation

Note

By default, a [Camera Image Input] or [Camera Image Input FH] processing item is set for unit 0.

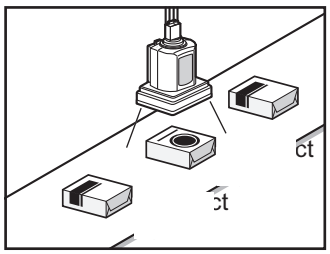
Example: Adding Position Compensation for Two Measurement Objects in the Same Field of View

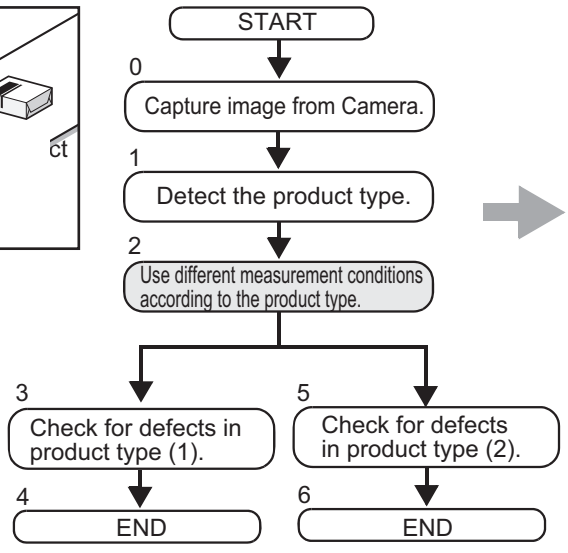




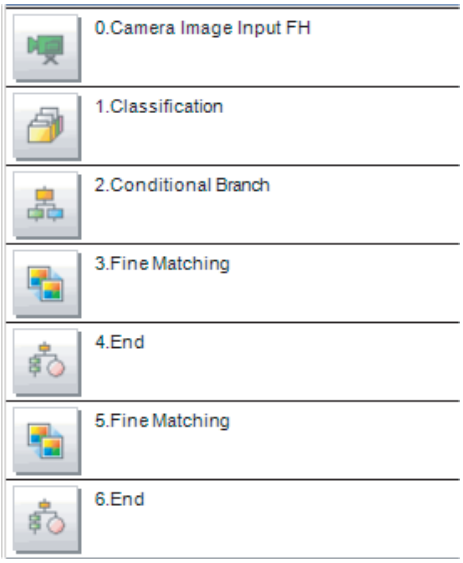
Visual Representation

Example: Determining the Product Type from the Image and Branching Later Inspection Conditions According to the Product Type (Branch Processing)



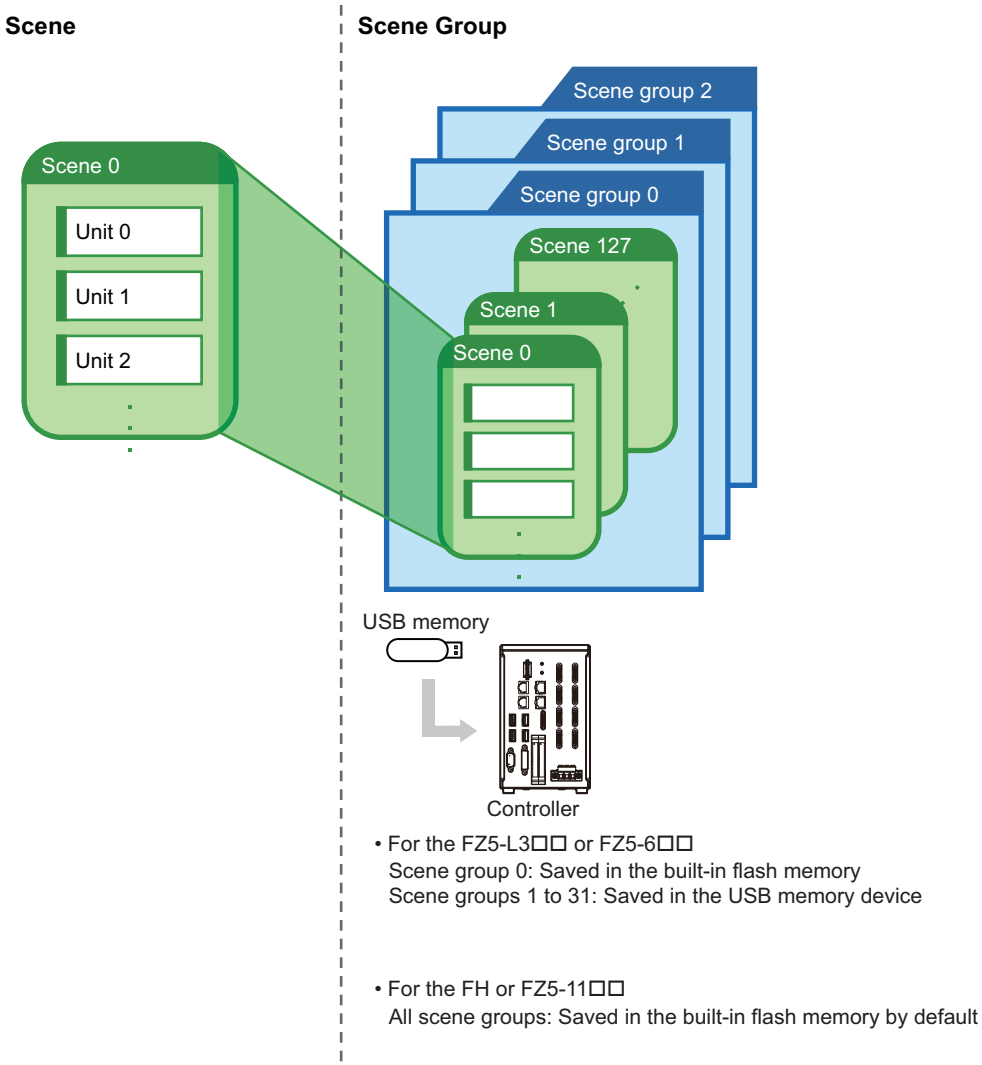


Visual Representation



What Is a Scene Group?

A scene group is a collection of up to 128 separate scenes. Creating scene groups is convenient to increase the number of scenes and to manage scenes according to categories.
For the FZ5-L3□□ or FZ5-6□□, a USB memory is required to create scene groups. Scene group 0 is saved in the Controller, while scene groups 1 to 31 are saved in USB memory. (For the FH or FZ5-11□□, all data is saved in the Controller.)



Note

- Up to 128 scenes can be handled as one scene group, and up to 32 scene groups can be set. You can therefore use 128 scenes × 32 scene groups, or 4,096 scenes.
- For the FZ5-L3□□ and FZ5-6□□, the drive name of the USB memory device must be named “USBDisk” for it to be used as the destination for scene groups 1 to 31. The names “USBDisk2” to “USBDisk4” cannot be used for saving. When other USB memory devices are already inserted for the USB disk, perform operations only after removing all USB memory devices other than the one in which the scene group data is stored.
- If the external memory capacity is insufficient for the data size, 4,096 scenes cannot be set. The scene data size depends on the settings.
- The data size that can be set (available data memory) can be checked from the system menu.
  Reference: ▶ Checking System Information: [System Information] (p.300)

Creating a Scene

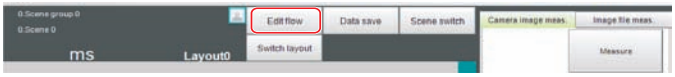
This section explains how to add processing units to a scene.

1 **In the Main Window (layout 0), display the scene to edit.**

Reference: ▶ Changing the Scene or Scene Group (p.60)

2 **Click the [Edit flow] button.**

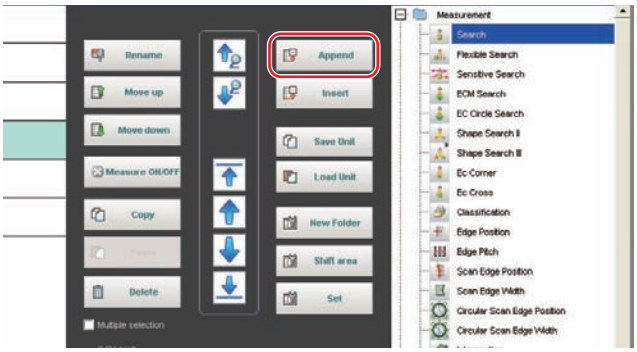
The edit flow window is displayed.



3 **Select the processing item to add from the processing item tree.**

4 **Click the [Append] button.**

The selected processing item is added at the bottom of the unit list (flow).



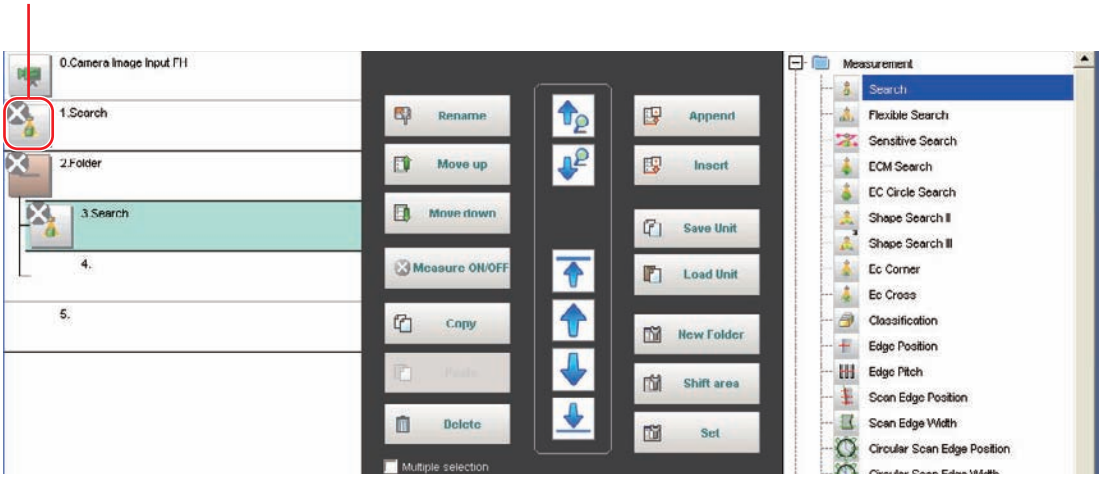
5 **Add another processing unit if necessary. Repeat the steps after 3.**

Note

- Limitations on Settings
  The number of image input processing items that can be used is limited.
  Reference: ▶ About Limits on the Number of Image Input Processing Items Used (p.858)

6 **Either click the icon of the processing unit to be set or click the [Set] button.**

Property setting buttons



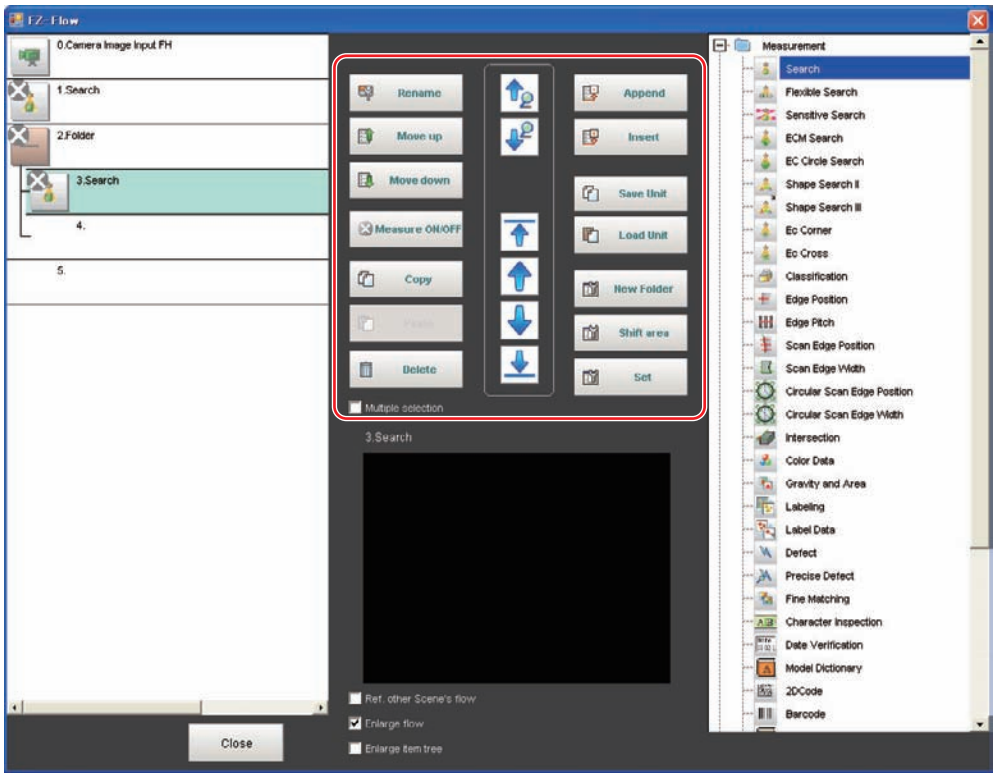
The Properties Tab Page is displayed. Set detailed conditions. The displayed contents depend on the processing item.

7 **Set the conditions.**

The displayed contents depend on the processing item.

Editing Processing Units in Scenes

You use the edit buttons in the edit flow window to arrange processing units in a scene, or to delete processing units.



- **Searching for a Processing Unit ( ) ( )**
  Searches the measurement flow for the processing item that is selected in the processing item tree.

- **Selecting a Processing Unit ( ) ( ) ( ) ( )**
  Automatically selects the processing unit at the very top or bottom of the unit list, or a processing unit at any position in between.

- **Adding a Processing Units at a Specified Position (Append) (Insert)**
  Adds or inserts a processing unit at the bottom position of the scene or another specified position.

- **Moving a Processing Unit (Move up)(Move down)**
  Moves a processing unit within a scene to change the processing order.

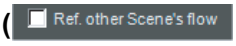
- **Copying and Pasting Processing Units (Copy) (Paste)**
  Copies and pastes a processing unit while maintaining the same settings.

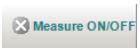
- **Saving Processing Unit Data to a File (Save Unit)**
  Saves the data for a selected processing unit to a file.

- **Loading Processing Unit Data (Load Unit)**
  Loads the data for a processing unit from a file.

- **Deleting a Processing Unit (Delete)**
  Deletes a processing unit from the scene.

- **Renaming Processing Units (**  **)**
  Renames a processing unit in the scene. Unit names must begin with a character other than ° (semi-voiced sound symbol) and ゛ (voiced sound symbol). Also, unit names cannot consist of only a single-byte number, only a "+", or only a ".".
- **Setting the Details of a Processing Unit (**  **)**
  Sets the properties of a processing unit in the scene.
- **Shift Area (**  **)**
  Changes the related figure data in one batch.
- **New Folder (**  **)**
  Used when multiple processing units are managed as a group.
- **Handling Multiple Processing Units (**  **)**
  Used when processing units are copied or deleted together. Click the [Multiple selection] button to display the check boxes for the processing items. The selected processing units can be manipulated as a group.

- **Referring to Other Scene Flows (**  **)**
  Units in other scenes can be referred to and added to the current scene flow.
- **Selecting a scene to refer to displays the flow for that scene.**

- **Switching processing unit measurement execution ON/OFF (**  **)**
  You can switch off the processing unit within a scene so that measurement is not executed for individual processing units. The processing unit that has been switched off cannot execute measurement until switched on. Pressing the button toggles the measurement execution setting between ON and OFF.

IMPORTANT

Make sure to enable Measure for the image input processing unit for unit 0 before use.

Note

- If a processing unit is inserted, the numbers for the subsequent processing units increase by one. With processing items related to results output or branch control, the numbers for processing units set as references also automatically increase by one.
- If a button other than [Paste] is clicked after pasting a processing unit, continued pasting of the processing item cannot be performed.
- If a processing unit is deleted, the numbers for the subsequent processing units decrease by one. With processing items related to results output or branch control, the numbers for processing units set as references also automatically decrease by one.
- To hide a specific processing unit in the flow on an adjustment window or run window, insert a "*" (single byte) at the beginning of the processing unit name.
- The number of units that can be set in a flow depends on the available data memory.
  With the FH or FZ5-11□□, a warning message is displayed when the amount of available application memory falls below 1 GB. Adjust the inspection flow so that the available memory is more than 1 GB.
  Reference: ▶ Checking System Information: [System Information] (p.300)
- When copying and pasting a conditional branch processing unit or other processing unit that refers to another processing units, the branch destination of the pasted processing unit is initialized to [End processing]. When pasting whole processing units within a folder, a parallelization processing unit, or a parallelized task processing unit, references from inside the pasted block to outside the block are initialized to [End processing].
  When copying and pasting a calculation processing unit or other processing unit that refers to another processing unit, the content of the expression of the pasted processing unit becomes #ERR. When pasting whole processing units within a folder, a parallelization processing unit, or a parallelized task processing unit, references from inside the pasted block to outside the block become #ERR.
- Turning measurement execution ON/OFF for a folder, a parallelization processing unit or a parallelized task processing unit turns measurement execution ON/OFF for all processing units within the folder, the parallelization processing unit or the parallelized task processing unit.
- When [ON] is set for [Parallel Execute] in [Operation mode setting] in the system settings, turning the measurement execution setting ON/OFF for a processing unit may greatly change the processing time. In this case, use parallelization processing items to parallelize the measurement flow instead of using automatic parallel processing.
- When measurement execution has been set to OFF for a processing unit that is a conditional branch destination, processing is executed starting with the first processing unit set to ON after the branch destination processing unit.

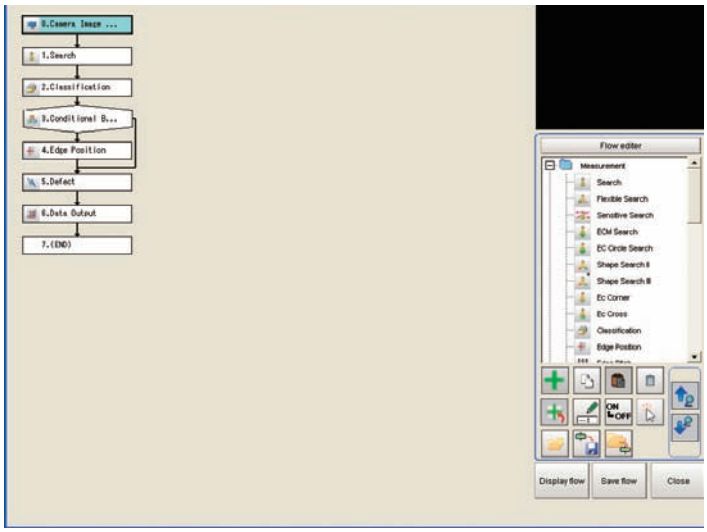
Displaying and Checking Processing Branches in a Scene

To see how a branch affects the processing flow after you edit a scene in the edit flow window, start the Flow Viewer.
Select [Flow Viewer] from the [Tool] menu.

Overview

The Flow Viewer shows the flow of processing for the measurement flow that is currently being edited.
This makes it easy to understand the relationship of branches and folders in the measurement flow.
The Flow Viewer allows you to edit the measurement flow, and also allows you to edit, rename, and save processing units in the same way as with the edit flow window.
You can save the flow of processing for the currently displayed measurement flow as an image file.

- You can also edit the measurement flow.
- You can edit the unit properties in the same way as with the edit flow window.
- You can save the flow of processing for the currently displayed measurement flow as an image in JPEG format.
- You can also rename and save processing units.

In Edit Flow Window	In Flow Viewer
	

Window Configuration

The window configuration for the Flow Viewer is shown below.

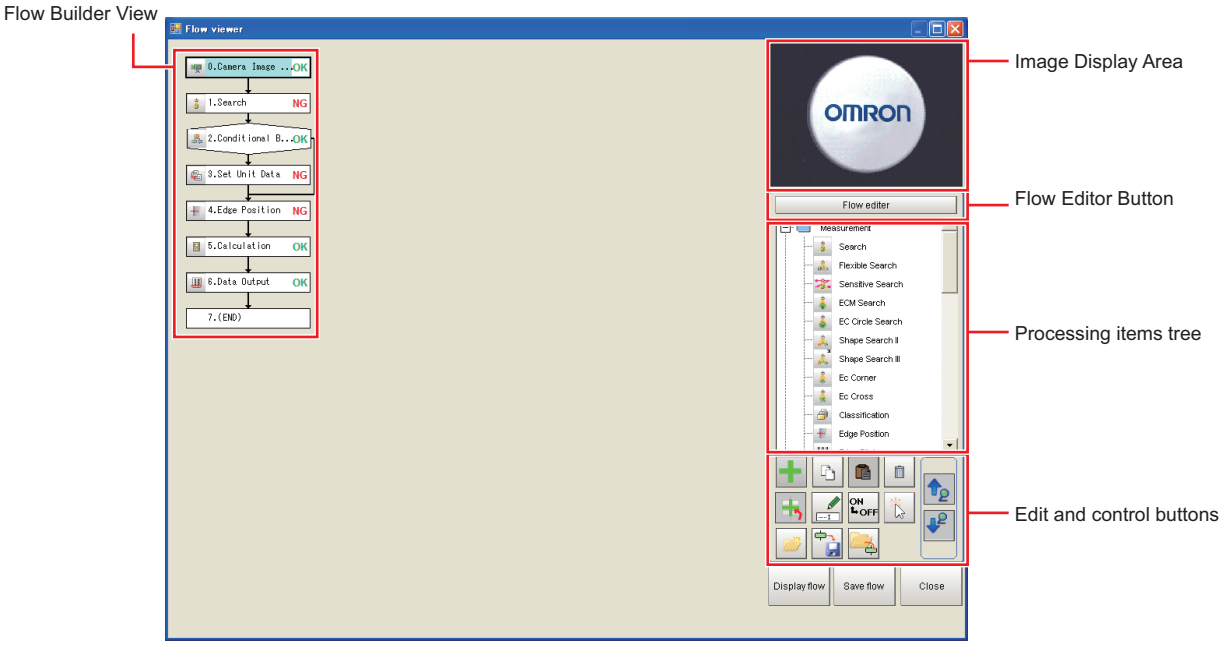


Image Display Area

This area shows Camera images, figures, positions, and other graphic information.
The measurement image is updated each time a unit is selected.

Flow Editor Button

This button starts the edit flow window.

Tree of Processing Items

This area allows you to select an item to add to the flow. The items are displayed in tree format that is organized by type.

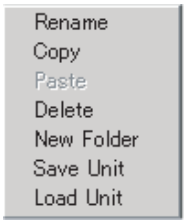
Flow Editing Functions

Repositioning a Unit within a Flow

On the Flow Builder, drag the unit to the desired position.
To insert a unit before another unit, drag it onto the desired unit.

Right-click Menu

The following popup menu is displayed when you right-click on a unit in the flow.



The menu commands perform the same operations as the Edit and Control Buttons.

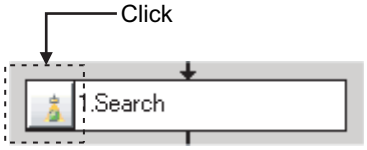
Edit and Control Buttons

Icon button	Function	Description
	Add to End of Flow	Adds the selected item in the item tree to the end of the flow as a new unit.
	Add before Current Unit	Adds the selected item in the item tree to the position directly before the selected unit in the flow as a new unit.
	New Folder	Creates a folder directly before the selected unit in the flow. Drag units to edit the contents of the folder.
	Copy	Copies the selected unit in the flow.
	Paste	Pastes the copied unit to the position directly before the selected unit in the flow.
	Delete	Deletes the selected unit from the flow.
	Rename	Renames the selected unit in the flow.
	Save Unit	Saves the settings for the selected unit in the flow to a file (with the extension .unit).

Icon button	Function	Description
	Load Unit	Loads the settings for a processing unit from a file (extension .unit) and places it directly before the selected unit in the flow.
	Measurement ON/OFF	Used to turn off measurement execution for individual in-scene processing units so that measurement is not executed.
	Movement mode	Enables the flow display position to be moved by drag operations instead of by using the scroll bar.
	Search Up	Searches for the selected item in the item tree upward from the selected position in the flow.
	Search Down	Searches for the selected item on the item tree in the downward direction from the selected position in the flow.
	Display Entire Flow	Displays the entire flow in the center of the view. The [Save] button outputs the same image as the [Save flow] button.
	Save Flow Image	Outputs the entire flow as an image file (extension .jpg).
	Exit Flow Viewer	Closes the Flow Viewer.

Changing the Properties of Units in a Flow

On the Flow Builder, click the icon to the left of the unit for which to change the properties. The property dialog box for the unit is displayed.



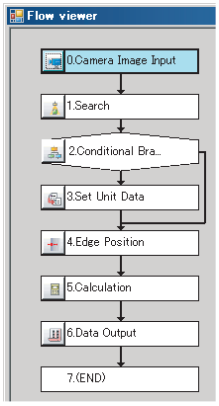
Examples of Branch and Folder Views

Examples:

Edit Flow Window	Flow Viewer	Edit Flow Window	Flow Viewer

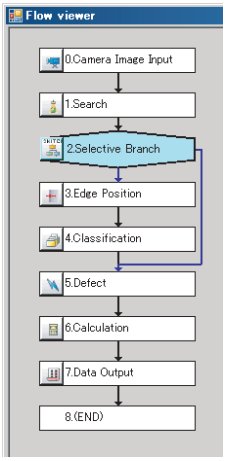
Conditional Branch View

⇒

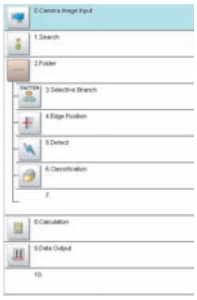


Selective Branch

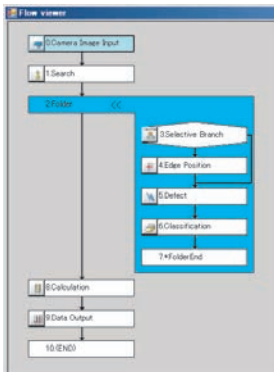
⇒



Folder View



⇒



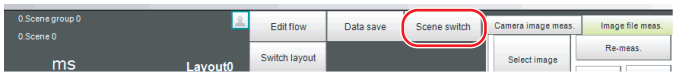
Changing the Scene or Scene Group

You can change the scene to perform a process change. The default display is scene 0 when the power supply is turned ON. You can create many scenes (scenes 0 to 127).
Also, if you use scene groups, you can set up to 4,096 scenes.
You can also send commands from an external device to change the scene group or scene.
Reference: ▶ *Vision System FH/FZ5 Series User's Manual for Communications Settings* (Cat No. Z342).

Changing the Scene

1 **In the Main Window (layout 0), click the [Scene switch] button in the Toolbox Pane.**

The [Scene switch] dialog box is displayed.

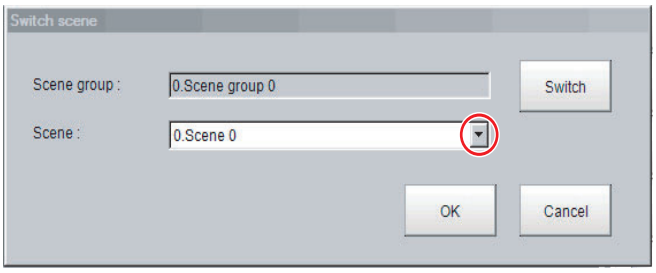


Note

- An alternative operation is to select [Switch scene] from the [Function] menu.

2 **Click [▼] to select the scene to change to.**

To change the scene group, click [Switch], and then click [▼] in the displayed dialog box to select the scene group to change to.



3 **Click [OK].**

The scene changes.

Changing Scene Groups

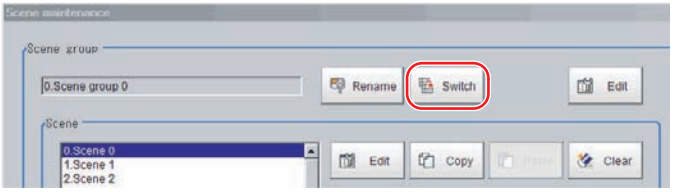
Use the following procedure to change to the scene group in which the scene to be edited is stored.

1 **In the Main Window (layout 0), select [Scene maintenance] from the [Function] menu.**

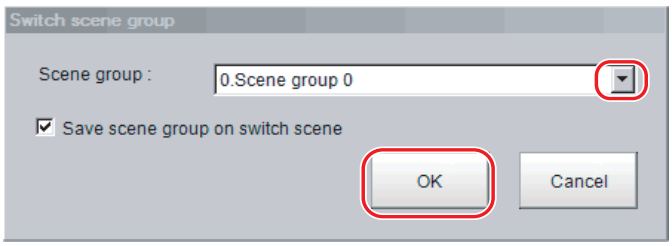
The [Scene maintenance] dialog box is displayed.

2 **Click the [Switch] button in the Scene Group Area.**

The [Scene group switch] dialog box is displayed.



3 **Change to the scene group to edit.**



1 **Click the [▼] button to select the scene group to edit.**

2 Select whether the scene group should be saved when changing to another scene group.

Parameter	Set value [Factory default]	Description
Save scene group on switch scene	[Selected]	When the scene group is changed, the data of the scene group before changing is saved.
	Not selected	The scene group data is not saved when you change to another scene group. Therefore, you can change the scene group faster.

Note

- The setting for whether to save a scene group when changing is linked to the settings of the [Measure setting] dialog box.
  Reference: ▶ Setting the Conditions That Are Related to Operation during Measurement (p.267)

3 Click [OK].

The scene group is changed and the [Scene maintenance] dialog box is displayed again.

IMPORTANT

- If you select [Save scene group on switch scene], data may be lost if the power supply is turned OFF while changing the scene group. Make sure that the power supply is not turned OFF while the scene group is being changed.
- If the available external memory is not sufficient to save data when changing the scene group, the data will be initialized if the power supply is turned OFF because the scene group data in the external memory is temporarily cleared during the saving process. To avoid this problem, increase the available external memory without turning OFF the power supply or reduce the scene group data size, and save the data to the external memory again.
- Immediately after starting up the FH/FZ5 Sensor Controller and immediately after changing scenes, there will be no input image. No input image is processed as the same color image as in the factory default state.
- If you open the Properties Dialog Box before inputting an image, click the Cancel button to close the dialog box. Pressing the OK button in the dialog box will change the setting to the same color camera setting as the factory default setting.

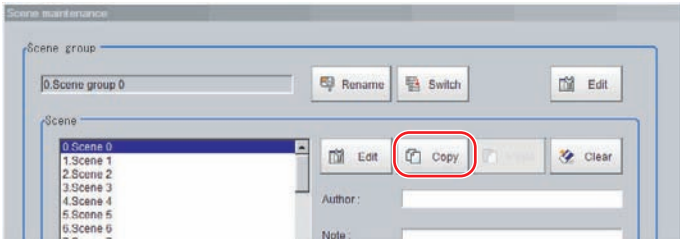
(Reference: ▶ The measurement NG (image mismatch) error will result when connecting a monochrome camera (p.832))

Copying Scenes

Use the following procedure to copy and paste a scene within a scene group.
This is convenient for reusing a scene and changing only the required parts.

1 **In the Main Window (layout 0), select [Scene maintenance] from the [Function] menu.**
The [Scene maintenance] dialog box is displayed.

2 **In the scene list, click the scene to copy, and then click [Copy].**



3 **In the scene list, click the scene where the copy is to be made, and then click [Paste].**
An overwrite confirmation dialog box is displayed.

4 **Click [Yes].**
The scene selected as the destination is overwritten with the copied scene data.

5 **Click the [Close] button.**

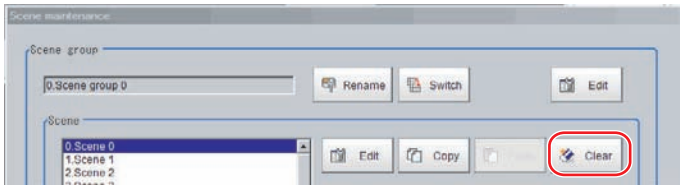
Deleting Scenes

Use the following procedure to delete the settings and return to the default values. Use the following procedure to initialize the measurement contents for a scene.

1 **In the Main Window (layout 0), select [Scene maintenance] from the [Function] menu.**
The [Scene maintenance] dialog box is displayed.

2 **Click the scene to be deleted from the scene list.**

3 **Click the [Clear] button.**
A confirmation message is displayed.



4 **Click [Yes].**
The scene data is deleted.

5 **Click the [Close] button.**

Renaming a Scene and Adding a Description

A user description can be added to each scene. This is convenient for making settings more easily understandable when managing many scenes.

1 **In the Main Window (layout 0), select [Scene maintenance] from the [Function] menu.**

The [Scene maintenance] dialog box is displayed.

2 **Click the scene to be renamed from the scene list.**

3 **Click the [Edit] button.**

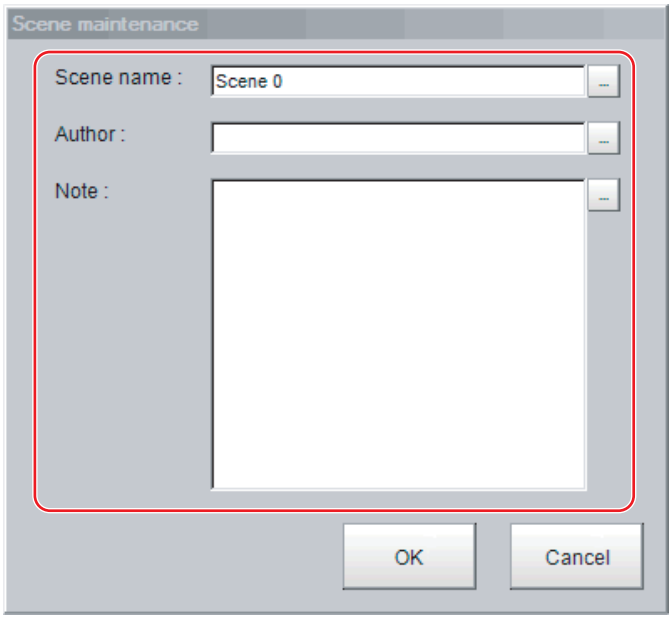
4 **Set the [Scene name], [Author], and [Note].**

1 **Click the [· · ·] button for each item.**

The on-screen keyboard is displayed.

2 **Set the names and a note.**

The [Scene name] and [Author] cannot be longer than 15 characters, and the [Note] cannot be longer than 255 characters.
You cannot use ° (semi-voiced sound symbol) and ゛ (voiced sound symbol) in the [Scene name].



Note

- When entering the [Note], enter a line-break after 32 single-byte characters or 17 double-byte characters. Without line breaks, the display of character strings will be truncated.

5 **Click [OK].**

6 **Click the [Close] button.**

Editing Scene Groups

Copying or deleting can be done by scene group and scene groups can be renamed.

Note

For the FZ5-L3□□ or FZ5-6□□, make sure to check that a USB memory device has been inserted before performing this operation.

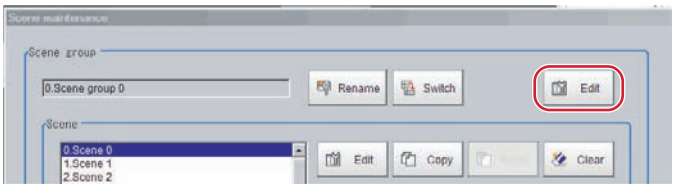
Copying and Deleting Scene Groups

1 **In the Main Window (layout 0), select [Scene maintenance] from the [Function] menu.**

The [Scene maintenance] dialog box is displayed.

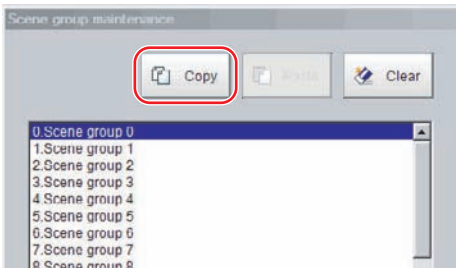
2 **Click the [Edit] button.**

The [Scene group maintenance] dialog box is displayed.



Copying

3 **Select the scene group to copy, and then click the [Copy] button.**



4 **Select the copy destination scene group and click [Paste].**

An overwrite confirmation dialog box is displayed.

5 **Click [Yes].**

The scene group selected as the destination is overwritten with the copied scene group data.

6 **Click the [Close] button.**

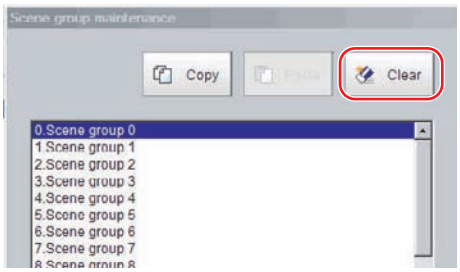
Deleting

The following data is deleted.

- **Name set for the scene group**
- **All scene data within the scene group**

7 **Select the scene group to delete, and then click the [Clear] button.**

A confirmation message is displayed.



8 **Click [Yes].**

The scene group data is deleted.

9 **Click the [Close] button.**

Changing the Scene Group Name

Scene groups can be named with any desired text. This is convenient to make scene groups easier to identify when there are many scene groups to manage.

1 **In the Main Window (layout 0), select [Scene maintenance] from the [Function] menu.**

The [Scene maintenance] dialog box is displayed.

2 **Click the [Rename] button.**

The [Scene group maintenance] dialog box is displayed.

3 **Click [...] next to [Scene group name].**

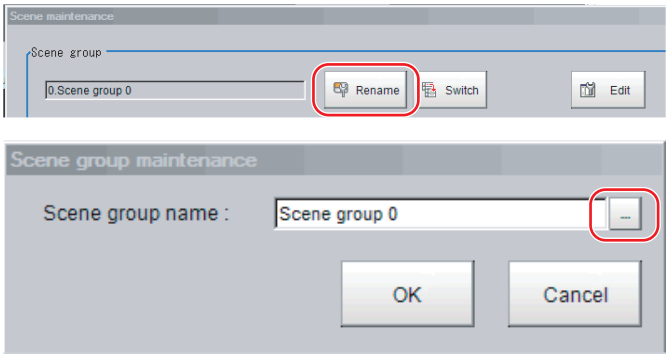
The on-screen keyboard is displayed.

4 **Set the name.**

Enter 15 characters or less.

5 **Click [OK].**

6 **Click the [Close] button.**



MEMO

Performing Test Measurement/Starting Operation

This section describes test methods for checking whether correct measurement can be performed at the set conditions and describes useful functions for operation.

Adjustment Windows and Run Windows

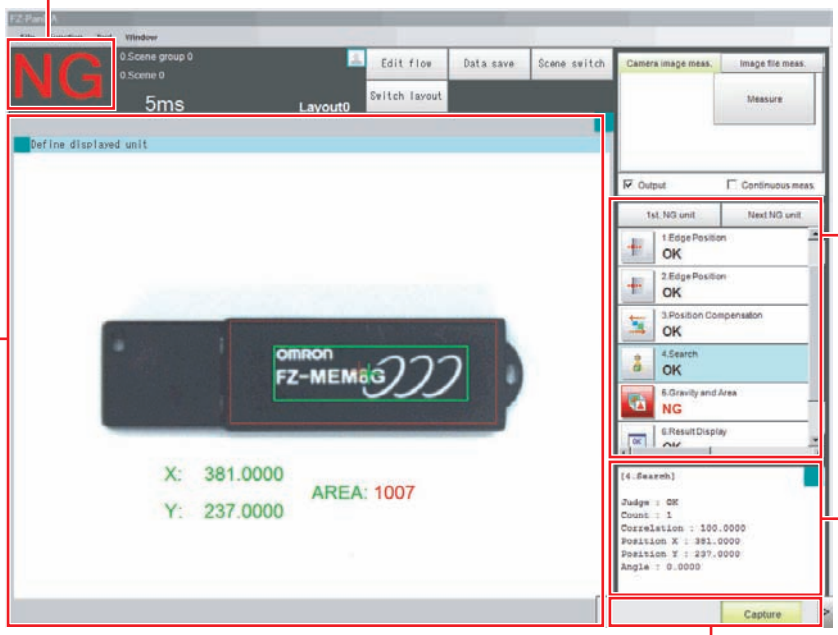
After test measurement and remeasurement are performed on the Main Window (layout 0), check the measurement results. If there are problems, adjust the processing item setting values of the processing units. When the measurement results are stable, change to the Main Window (layout 1) and perform measurements. This section describes the adjustment windows and run windows.

Main Window (Layout 0): Adjustment Window (Default)

Layout 0 is set as an adjustment window by default. (This can be changed in Layout Modification Mode.)

The judgement result and measurement time for the entire scene are displayed.

You can overlay the measurement results on the Camera image.



Units with NG results are displayed in red.

Measurement results are displayed in text format.

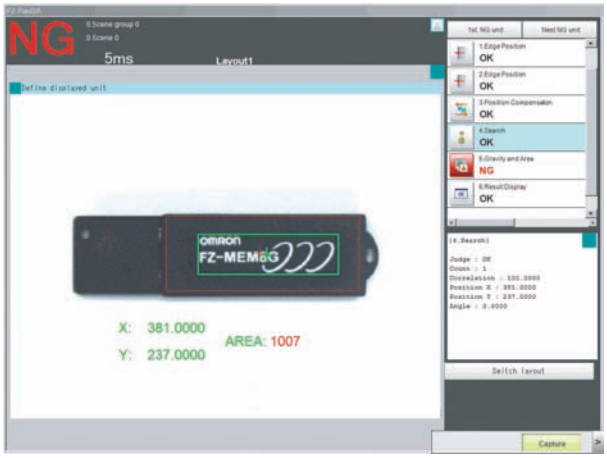
Measurement Manager Bar
This is displayed regardless of the layer that is displayed.

- [Capture]: Saves the contents that is displayed on the monitor as an image.
  Reference: ▶ Capturing Screen Images (p.123)
- [LCD Off] (Displayed only with LCD-integrated Controllers.)
  Turns OFF power to the LCD monitor.
  Reference: ▶ Turning OFF the LCD (p.44)
- [Line] (This is displayed only when the operation mode is set to Multi-line Random-trigger Mode.)
  This changes the line.
  Reference: ▶ Multi-line Random-trigger Mode (p.273)
- [Switch Screen] (This is displayed only when the operation mode is set to Non-stop Adjustment Mode.)
  This changes to the Non-stop Adjustment Mode Window.
  Reference: ▶ Non-stop Adjustment Mode (p.279)

Main Window (Layout 1): Run Window (Default)

Layout 1 is set as a run window by default. (This can be changed in Layout Modification Mode.)

The flow, detailed results, and tool box are displayed in the Control Area.



When processing takes a long time, it is necessary to check the processing items and setting values. The time required for measurement is also displayed with the measurement results, so use this for reference.

Main Window (Layout 1): Changing to the Run Window

1 **In the Main Window (layout 0), click the [Switch layout] button.**

2 **In the [Switch layout] dialog box, select the layout number (layout 1 by default).**

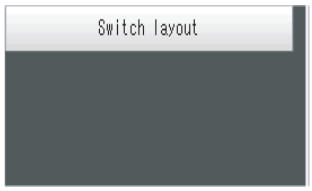
The display changes to the Main Window (layout 1).

Note

- You can make settings so that the run window is displayed whenever the power supply to the Controller is turned ON.
  Reference: ▶ Setting the Status at Startup: [Startup Settings] (p.293)
- The lighting gradually gets darker if it is used for a long time, so adjust judgement conditions periodically.
- Without stopping the measurement, you can change judgement conditions for a processing unit set for a scene.
  Reference: ▶ Changing Judgement Conditions without Stopping Measurement (p.111)

Main Window (Layout 1): Changing to the Adjustment Window

1 **In the Main Window (layout 1), click the [Switch layout] button in the Toolbox Pane.**



2 **In the [Switch layout] dialog box, select the layout number (layout 0 by default).**

The display changes to the Main Window (layout 0).

Camera Setup

FH/FZ5-series Sensor Controllers use the CameraLink standard for camera connection. At normal camera connection, a single camera cable is used for a single camera, and each camera is connected to the Sensor Controller in the CameraLink base configuration. Starting from camera connector 0 on the Sensor Controller, connect cameras to the camera connectors you want to use in ascending order of camera connector number.

IMPORTANT

- Immediately after starting up the FH/FZ5 Sensor Controller and immediately after changing scenes, there will be no input image. No input image is processed as the same color image as in the factory default state.
- If you open the Properties Dialog Box before inputting an image, click the Cancel button to close the dialog box. Pressing the OK button in the dialog box will change the setting to the same color camera setting as the factory default setting.

(Reference: ▶ The measurement NG (image mismatch) error will result when connecting a monochrome camera (p.832))

Using Two Cables for High-frame-rate Camera Use (FH Series Only)

FH-series Sensor Controllers support the CameraLink standard's base configuration and medium configuration. The medium configuration enables use at a higher frame rate than the base configuration, reducing image input time. The configuration that can be connected will depend on the connected cameras. See the instruction sheet of the cameras you are using.

To connect a camera in the medium configuration, you will need to use two camera cables of the same type and length to connect the camera and Sensor Controller (two-cable connection). Sensor Controller camera connectors should be connected as a pair with ascending consecutive even and odd numbered connectors. The table below shows the camera connector and camera number assignments when connecting the Sensor Controller and camera cable with a two-cable connection.

Camera numbers for operation modes other than multi-line random trigger mode

The table below shows the camera connector and camera number assignments when the operation mode is a mode other than the multi-line random trigger mode. A number corresponding to a Sensor Controller camera connector number is assigned for the camera number. When using two-cable connection, the even number in the pair of ascending consecutive camera connector numbers is assigned as the camera number.

Sensor Controller camera connector number	Camera number in application software		
	Example when all cameras use one-cable connection	Example when all cameras use two-cable connection	Example when combining one- and two-cable connection
0	0	0	0
1	1		1
2	2	2	2
3	3		
4	4	4	Not connected
5	5		5
6	6	6	6
7	7		

Camera numbers for multiple-line random trigger mode

The table below shows the camera connector and camera number assignments when the operation mode is the multi-line random trigger mode. This table is an example showing the camera number assignments when the number of lines is two.
Camera numbers are assigned for each line in ascending order starting from 0. For two-cable connection, camera numbers are assigned in ascending order regardless of camera connector number pairs.

<table>
<tr><th rowspan="3">Sensor Controller camera connector number</th><th colspan="6">Camera number in application software</th></tr>
<tr><th colspan="2">Example when all cameras use one-cable connection</th><th colspan="2">Example when all cameras use two-cable connection</th><th colspan="2">Example when combining one- and two-cable connection</th></tr>
<tr><th>Line 0</th><th>Line 1</th><th>Line 0</th><th>Line 1</th><th>Line 0</th><th>Line 1</th></tr>
<tr><td>0</td><td>0</td><td>---</td><td rowspan="2">0</td><td rowspan="2">---</td><td>0</td><td rowspan="2">---</td></tr>
<tr><td>1</td><td>1</td><td>---</td><td>1</td></tr>
<tr><td>2</td><td>2</td><td>---</td><td rowspan="2">2</td><td rowspan="2">---</td><td>2</td><td rowspan="2">---</td></tr>
<tr><td>3</td><td>3</td><td>---</td><td></td></tr>
<tr><td>4</td><td>---</td><td>0</td><td rowspan="2">---</td><td rowspan="2">0</td><td>---</td><td>Not connected</td></tr>
<tr><td>5</td><td>---</td><td>1</td><td>---</td><td>0</td></tr>
<tr><td>6</td><td>---</td><td>2</td><td rowspan="2">---</td><td rowspan="2">2</td><td rowspan="2">---</td><td rowspan="2">1</td></tr>
<tr><td>7</td><td>---</td><td>3</td></tr>
</table>

IMPORTANT

- When connecting cameras with two-cable connection, use a pair of ascending consecutive even and odd numbered Sensor Controller camera connector numbers for each camera. You will not be able to get the image acquisitions correctly if you use camera connectors with non-consecutive connector numbers, or if an odd numbered connector has a lower number than the even numbered connector it is paired with.
- CameraLink medium configuration is only supported by FH-Series Sensor Controllers. FZ5-series models, FZ4-series models and other non-FH-series Sensor Controllers only support the CameraLink base configuration.
- When [Operation mode setting] has been set to the multi-line random trigger mode, changing the camera connections or [Multi-line Random-trigger mode] in the system settings may change camera number assignments in line with the change made. After changing the camera connections or [Multi-line Random-trigger mode], before using the Sensor Controller, check the camera numbers in the property setting screen for the [Camera Image Input] processing items or [Camera Image Input FH] processing items.

Note

When connecting cameras with two-cable connection, the camera connection screen in the system settings displays the camera model for one member of the pair of camera connector numbers, and displays [Disconnect] for the other.

One-cable Camera Connection

FH/FZ5-series Sensor Controllers support the CameraLink standard's base configuration. In the base configuration, one camera cable is used to connect one camera. Starting from camera connector 0 on the Sensor Controller, connect cameras to the camera connectors you want to use in ascending order of camera connector number.

Preparing a Controller

No special preparations are required with this Controller because processing items are pre-installed.
Please check that the Controller is turned On and that the Main Window is displayed.
For details, refer to the *Instruction Manual*.
The first time the program is started, the Language Setting Dialog Box is displayed. Select the language.
Reference: ▶ Selecting the Language: [Language Setting] (p.292)

Adjusting the Camera

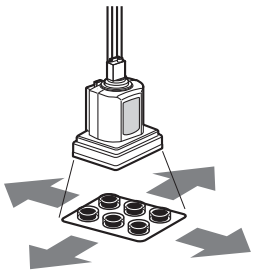
Check the images that are being taken.
Adjust the position of measurement objects and the focus of the lens.

1 **Click the upper left corner of the Image Pane, and then select [Image mode] – [▼] – [Through].**

The through image from the Camera is displayed in the Image Pane.

Reference: ▶ Arranging the Window Display (p.92)

2 **Adjust the position of measurement objects so that they display at the center of the monitor.**



Adjusting the Position of the Measurement Target

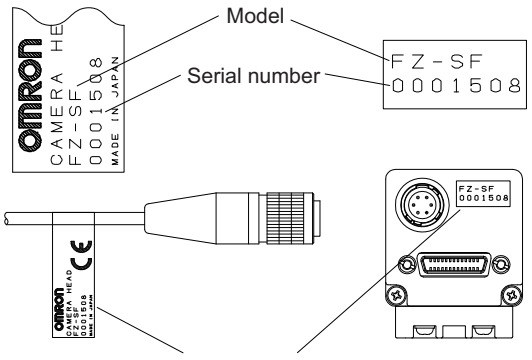
3 **Adjust the focal distance of the lens.**

When using an Auto-focus Camera or an Intelligent Camera, the focus and the iris can be automatically adjusted.

Note

- If a Camera is used together with a lens, turn the focus ring on the lens to adjust the focus.
  Refer to ▶ *Lens Adjustment Setting* in the *Vision System FH/FZ5 Series Processing Item Function Reference Manual* (Cat No. Z341).
- The light intensity of an Intelligent Camera can adjusted from the Controller.
  Refer to ▶ *Lighting Control* in the *Vision System FH/FZ5 Series Processing Item Function Reference Manual* (Cat No. Z341).

IMPORTANT



Check the model and serial number.

When using a Compact Digital Camera, check that the model and serial number of the Camera Head and Camera Amplifier match. When a Camera Head and Camera Amplifier with different models and serial numbers are connected, they may not operate correctly.

Intelligent Camera (with Lighting)

Proper lighting is of crucial importance to a Vision System.
If an Intelligent Camera is connected, lighting can be controlled from the Controller.
Features of Intelligent Cameras are as follows:

- **A single Camera enables testing of illumination from various angles, so it is possible to shorten the lighting setting time and test measurement time.**
- **The Controller controls the lighting, so lighting can be adjusted depending upon the product type.**
- **Reproducibility of lighting settings is improved.**
- **Settings can be modified without changing the lighting.**

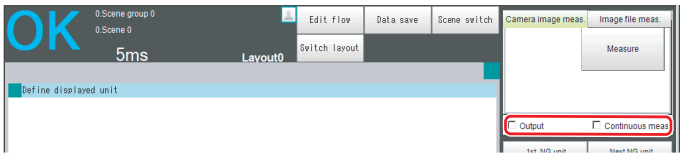
Refer to ▶ *Screen Adjustment Settings (Camera Image Input)* in the *Vision System FH/FZ5 Series Processing Item Function Reference Manual* (Cat No. Z341).

Executing Test Measurements

Test whether the intended measurement processing can be performed with the current setting contents. Look at test results and adjust the property settings of each processing unit.
Perform measurement according to the conditions set for the displayed scene.

1 **In the Main Window (layout 0), click the [Camera image meas.] button in the Measurement Pane.**

2 **Set the following items as test conditions.**



Parameter	Description
Output	Select this check box when the measurement results on the adjustment window are also to be output. Clear the checkmark when test measurement for the device is to be performed without results being output.
Continuous meas.	Select this check box when continuous measurement is to be performed. Click the [Measure] button to start continuous measurements.

3 **Click the [Measure] button in the Measurement Pane.**

The measurement is performed.

Note

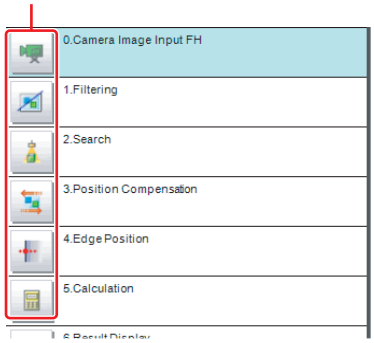
- If continuous measurements are in progress, the [Stop continuous meas.] button will be displayed. To stop continuous measurements, click the [Stop continuous meas.] button.

4 **Check the measurement results.**

5 **If necessary, adjust the setting values for each processing unit again.**

You can move to the property dialog box directly by clicking the button of any processing unit in the flow.

Property setting buttons



Note

Test images can be saved. This function is called logging. After setting conditions, the saved test images can be used in performing test measurements again.
Reference: ▶ Logging Measurement Values and Measurement Images (p.113)

IMPORTANT

The measurement interval and display update interval will vary for continuous measurements with test measurement settings and continuous measurements with serial commands or parallel commands.
Evaluate the measurement interval and display update interval by watching actual operation.

Key Points for Adjustment

This section describes key points for adjustment to improve measurement precision and shorten measurement time.

Stabilizing Measurements

This section describes key points for adjustment when measurements are not stable. There are two methods for improving measurement precision: Performing processing of images loaded from the Camera (filtering) or adjusting settings and parameters.

Adjusting Parameters of Each Processing Item

Adjustment to improve precision and stability depends on the processing item.
Refer to ▶ *Key Points for Adjustment* for individual processing items in the *Vision System FH/FZ5 Series Processing Item Function Reference Manual* (Cat No. Z341).

Processing Images (Filtering)

There are cases in which high-precision measurement is impossible, such as when using images loaded from the Camera that have noise, irregularities, or low contrast or when the background has patterns during defect measurement.
Refer to ▶ *Filtering* in the *Vision System FH/FZ5 Series Processing Item Function Reference Manual* (Cat No. Z341).

When measurement images have irregularities (searching and location positioning are not stable)	Measurement Images That Contain Noise
The Smoothing (Strong and Weak) and Median filter items are both effective. • **Smoothing (Strong and Weak)** This processing changes the shade of images so that irregularities are not as easily seen. **Weak Smoothing** Before Processing / After Processing  • **Median** In comparison with smoothing, the Median filter item allows for irregularities to be hidden without having to shade the edges of images.	The Dilate and Erosion filtering items are both effective. • **Dilate** When there is dark noise in an image, bright areas are enlarged to eliminate dark noise. • **Erosion** When there is bright noise in an image, bright areas are contracted to eliminate bright noise. **Erosion** Before Processing / After Processing 

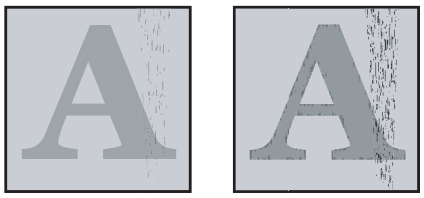
Measurement Images with Low Contrast (Defect Inspection Is Unstable)

The Extract Vertical Edges, Extract Horizontal Edges, and Extract Edges filtering items are effective.

- **Extract Vertical Edges**
  This extracts the vertical edges of an image.

Extract Vertical Edges

Before Processing / After Processing



- **Extract Horizontal Edges**
  This extracts the horizontal edges of an image.
- **Edge Extraction**
  This extracts all edges of an image.

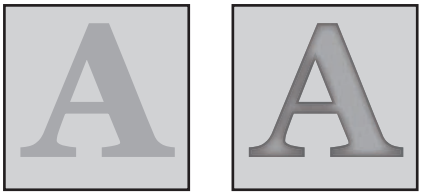
When Unidentifiable Shapes Are Present

The Edge Extraction filtering item is effective.

- **Edge Extraction**
  This is used to make the profile clearer and the shape more identifiable.

Edge Extraction

Before Processing / After Processing

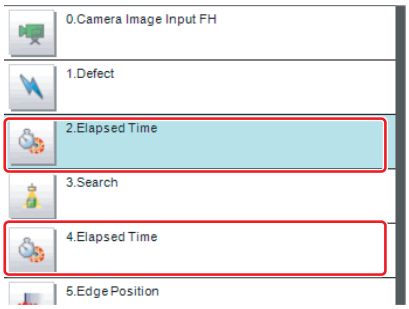


Shortening Processing Time

Checking Processing Times for Each Processing Unit

Find out which processing units are taking the most time and adjust the parameters of those processing items.

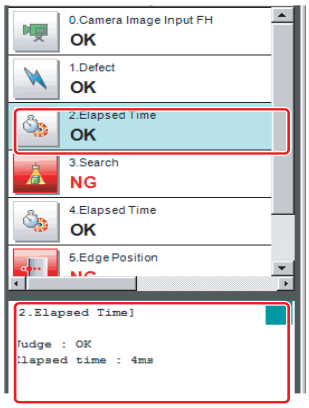
1 **Insert the Elapsed Time processing item after the processing unit for which time is to be measured.**



2 **Execute the measurement.**

3 **Click the Detail Result Area, and then click the Elapsed Time processing unit where time is to be checked.**

The elapsed time from the top of the flow to the relevant processing unit is displayed.



4 **Adjust the parameters of the processing units that are taking time.**

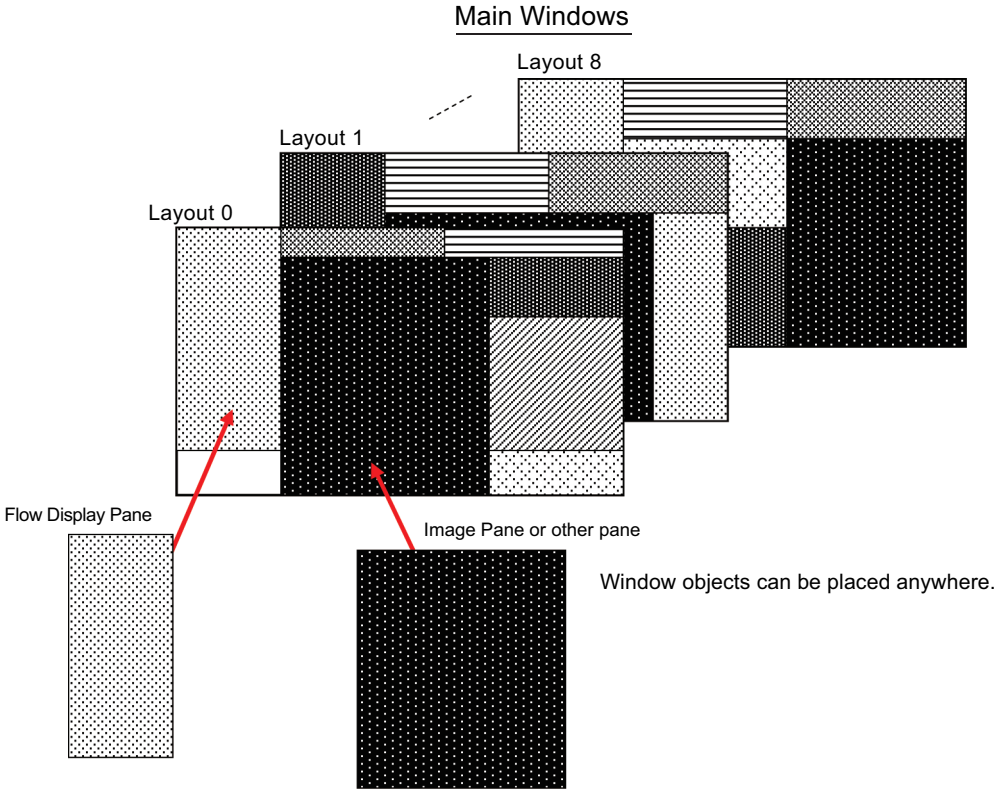
Refer to ▶ *Points for Adjustment* for individual processing items in the *Vision System FH/FZ5 Series Processing Item Function Reference Manual* (Cat No. Z341) for details on adjustment parameters.

Shortening the Processing Time for the Overall Measurement Flow

You can use parallel processing to shorten the processing time.
Parallel processing reduces the processing time of the overall measurement flow. It uses a multi-core CPU to process different parts of the measurement flow in parallel at the same time.
Reference: ▶ Parallel Processing (p.282)

Arranging Window Elements (Layout Modification)

Configuration elements for the Main Windows (called *window objects*) can be laid out and displayed anywhere. You can create a maximum of 9 Main Windows from layout 0 to layout 8.



Each Layout Window is set as follows by default:

Layout	Default setting	Behavior of output signals	
		RUN signal output	Signal output of results
Layout 0	Layout 0 is set as an adjustment window.	OFF	OFF
Layout 1	Layout 1 is set as a run window.	ON	ON
Layouts 2 to 7	Layouts 2 to 7 are for user-defined purposes and are created as they are needed. By default, these layouts have the same settings as layout 0.	OFF	OFF
Layout 8	Layout 8 is set as a remote operation window. When an error occurs in Layout 8, error dialog appears as a text string in the Error Pane instead of in an error dialog box.	OFF	OFF

- You can use these layouts to create Main Windows that show only the information that the operator needs, such as a window that shows only the image and the OK or NG status for operation in the field.
- You can change the position and size of the image display, or the processing unit to be displayed for each product type, inspection, or measurement.

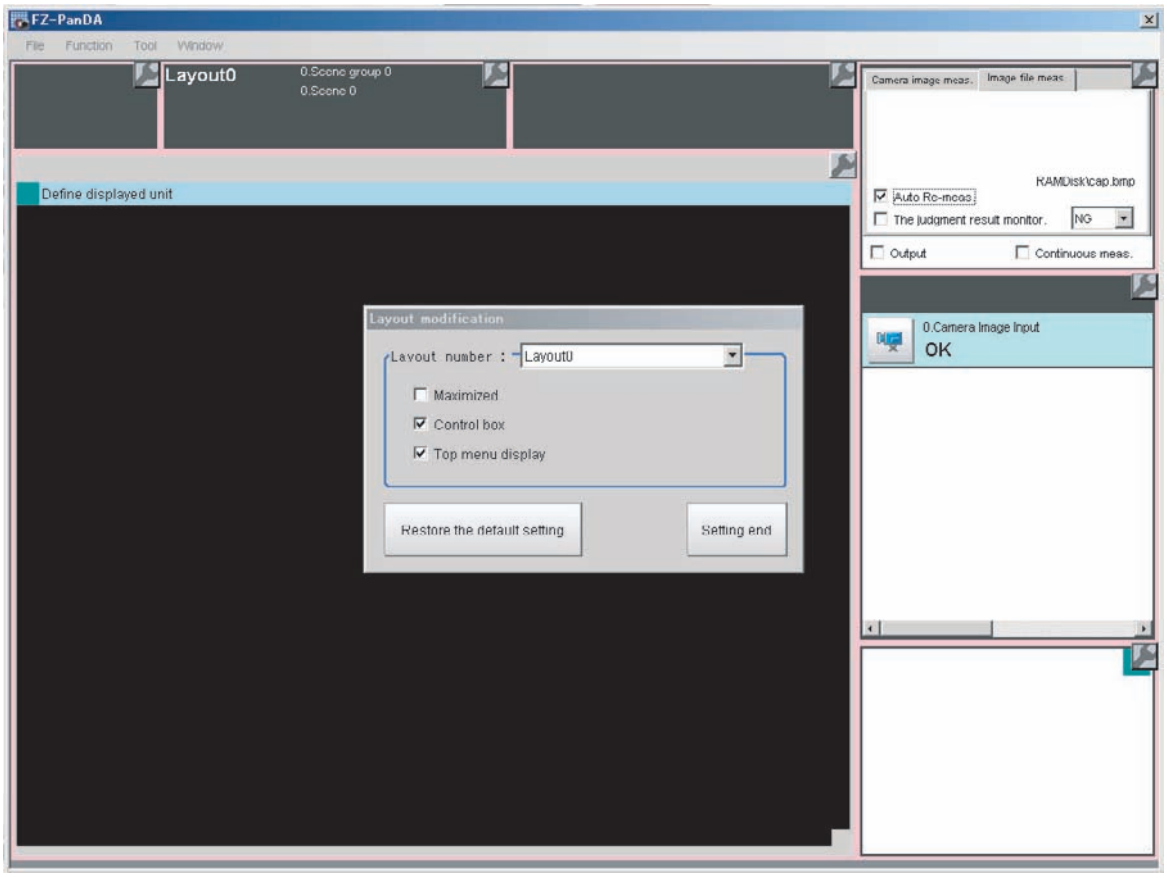
IMPORTANT

If the operation mode is set to Multi-line Random-trigger Mode, create a layout for each line.

Modifying Layouts

1 **Select [Layout modification] from the [Window] menu in the Main Window for layout 0.**

2 **The display changes to Layout Modification Mode, and the [Layout modification] dialog box is displayed as shown below.**



A icon will appear on the upper right corner of each object. This indicates that the Layout Modification Mode is active.

In this state, you can display and arrange the layout of objects in the Main Window.

*1: Normal menu and window operations are not available in Layout Modification Mode.

3 **Select a layout from [Layout 0] to [Layout 8] in the [Layout number] list.**

The Main Window that is behind the [Layout modification] dialog box will change.

You can perform the following operations.

- Click the icon in the upper right corner of each window object to display the settings dialog box for that object.
- Drag any window object to change its display position.
- Drag the border of any window object to change its size.
  With the FZ5, you can drag the icon in the lower right corner of a window object to size it.

The [Layout modification] dialog box displays the following elements.

Item		Description [Factory default]
Layout number		Select the number of the layout to edit. Layout 0 to layout 8
	Maximized	This function is exclusively for an FH Sensor Controller. Specify whether to always display the Main Window at its maximized state. • Selected: The Main Window will be displayed in its maximized state on the current display. This window cannot be moved. • [Not selected]: The Main Window will be set to the size and position that was in effect immediately before it was maximized.
	Control box	This function is exclusively for an FH Sensor Controller. Specify whether to display a Minimize Button and a Close Button in the Main Window. • [Selected]: A Close Button and Minimize Button will be displayed in the upper right corner of the Main Window. • Not selected: The Close Button and Minimize Button in the upper right corner of the Main Window will be hidden.
	Top menu display	Specify whether to display the main menu at the top of the Main Window. • [Selected]: A menu will be displayed at the top of the window. • Not selected: The menu at the top of the window will be hidden, and changes to the layout will no longer be allowed.
Restore the default setting		The layout for the selected layout number is restored to the default settings.
Setting end		This ends Layout Modification Mode.

Note

If you change to a layout number (default setting: layout 1) where the menu at the top of the Main Window is hidden and you exit the Settings Mode, the main menu at the top of the Main Window will disappear.
To display the menu at the top of the Main Window for that layout number, use the following procedure.

1 **Click the [Switch layout] button that appears in the lower right corner, and change to a layout number where the [Top menu display] option is selected.**

2 **Select [Layout modification] from the [Window] menu.**

3 **Change to the previous layout number (where the menu at the top of the Main Window is hidden).**

4 **Select the [Top menu display] option.**

IMPORTANT

- The setting of [Maximized] applies only to the FH.
  With the FZ5 Sensor Controller, the Main Window is always displayed at the maximum size regardless of the setting of [Maximized].
- The setting of [Control box] applies only to the FH.
  With the FZ5 Sensor Controller, the Minimize Button and Close Button are always hidden in the Main Window regardless of the setting of [Control box].

Common Operations for Window Objects

- Click inside a window object and drag it to change its position.
- Click the outer edge of a window object and drag it to change the size of the object.
- Click the  icon that appears on the upper right corner of any window object to open the settings dialog box for that object.

Adding Window Objects

1 **Right-click somewhere in the gray area outside the window objects and select the name of the object to add.**

Deleting Window Objects

FH/FZ5-11□□

1 **Right-click the window object and click [Delete the window].**

FZ5-L35□/FZ5-6□□

1 **Click [×] in the upper right corner of the window object.**

Restoring a Layout to its Default Settings

1 **Select the layout number for the layout to restore to its default settings, and click [Restore the default setting]. The layout for the selected number is restored to its default settings.**

To Exit Layout Modification Mode

1 **Click [Finish] on the layout modification Window to end layout modification.**

Changing the Layout Number from a Layout with a Hidden Top Menu

If you click [Finish] on a layout where the top menu display is hidden, the [Switch layout] button will automatically appear in the lower right corner. Click the [Switch layout] button and switch to a layout number where the [Top menu display] option is selected.

Saving the Layout in the Controller or a File

Saving the Layout in the Controller

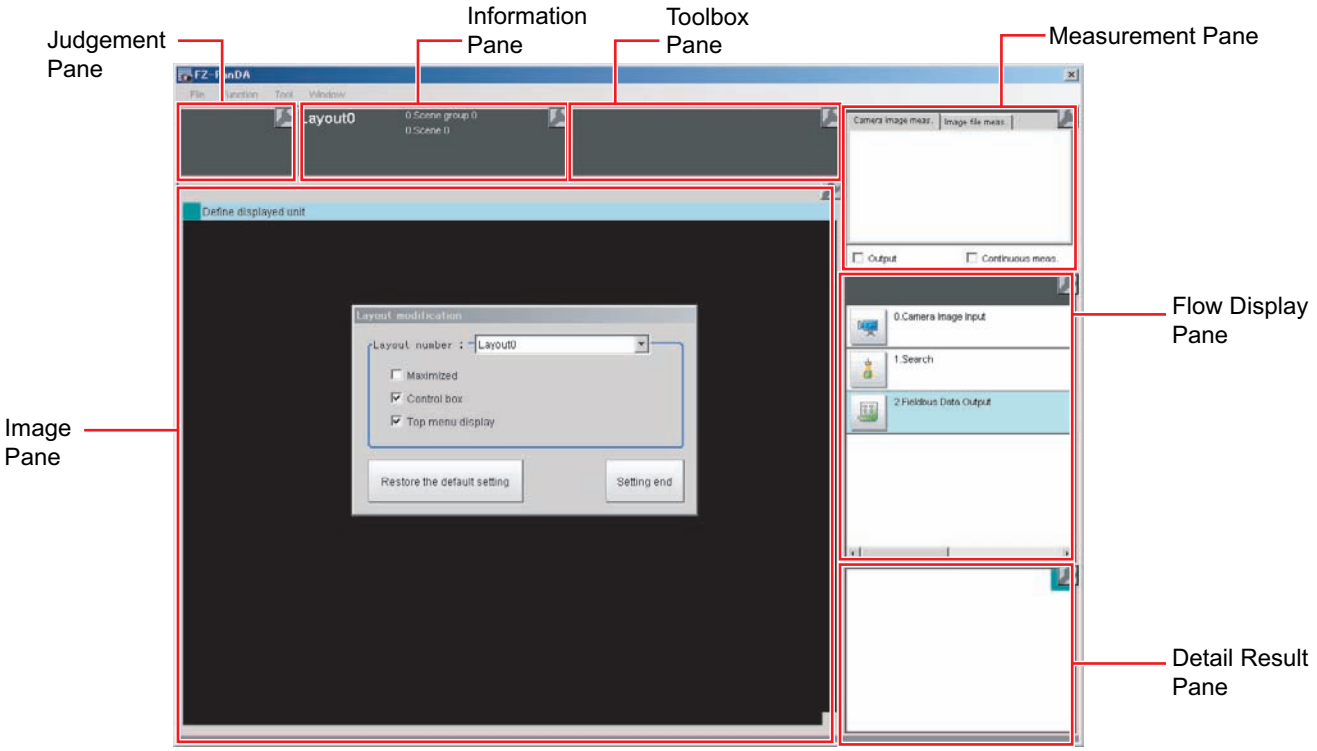
After you end Layout Modification Mode, click the [Data save] button.
Restart the Sensor Controller to reflect the layout modifications that were saved.

Saving a File

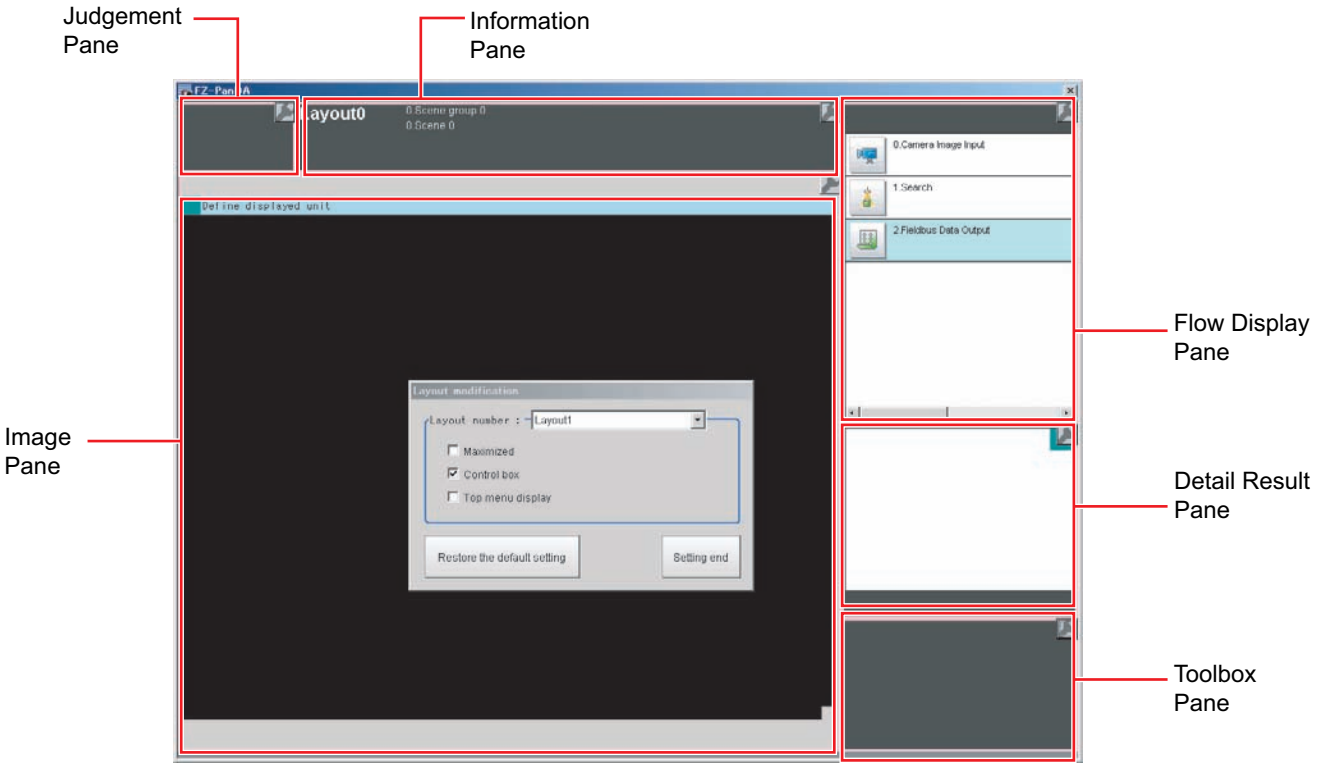
Select [Save to File] from the [Function] menu. Under [Settings data], set the [Data to be saved] to [System data] or [System + Scene group 0], specify the destination, and then click the [OK] button.

Window Objects That Can Be Laid Out or Changed

Layout 0 (in Default State) and Layouts 2 to 8



Layout 1 (in Default State)



You can place the following panes anywhere on a single layout.
You cannot have more than one pane of the same type.

Pane	Name displayed when a pane is added in Pane Customization Mode
Error Pane	FZ_PanDA.ErrorWindow
Flow Display Pane	FZ_PanDA.FlowWindow
Image Pane	FZ_PanDA.ImageContainerWindow
Information Pane	FZ_PanDA.InformationWindow
Judgement Pane	FZ_PanDA.JudgeWindow
Measurement Pane	FZ_PanDA.MeasureWindow
Detail Result Pane	FZ_PanDA.TextWindow
Toolbox Pane	FZ_PanDA.ToolWindow

Judgement Pane

This pane shows the overall judgement result. There are no parameters that can be set.

Information Pane

This pane shows various information about the measurement and scene.
The following elements are displayed.

- Processing Time
- Layout Number
- Scene Group Number: Scene Group Name
- Scene No.: Scene Name
- Logging Errors
- PLC Link Errors

Editable items	Description [Factory default]
View	Layout Number, Processing Time, Scene Group Number + Scene Group Name, Scene No. + Scene Name, Logging Error, PLC Link Error [All visible]
Layout	Drag to change positions.
Size	1 to 64 (the default size depends on the item)
Load default	Restores the default status.

Toolbox Pane

This pane shows the tool buttons.
The following elements are displayed.

- [Edit flow] button
- [Data save] button
- [Switch scene] button
- [Switch layout] button

Editable items	Description [Factory default]
Add Button	Each click adds one button.
Button size	Sets the size of each button in pixels. 16 × 999, [102 × 40]
Button row	Sets the number of buttons to display per row. 1 to 32, [3]

Measurement Pane

This pane is for measuring and re-measuring. The following elements are displayed. There are no editable elements.

- [Camera image meas.] tab page: [Measure] button
- [Image file meas.] tab page: [Selection image] button, [Re-meas.] button, [Auto Re-meas.] check box, and the [Judgement result monitor] check box
- [Output] check box
- [Continuous meas.] check box

Flow Display Pane

This pane shows the measurement flow.

The following elements are displayed.

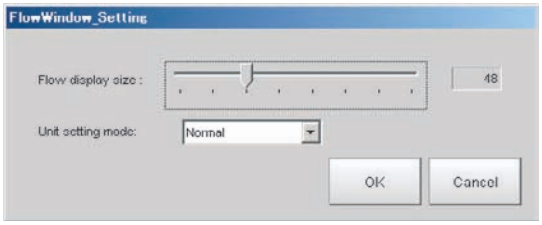
- [1st. NG unit] button
- [Next NG unit] button
- Measurement flow list

- **To Change the Size of the Flow Display Pane**

  Use the following procedure if the type of settings dialog box to display when you click a processing unit icon is set to the Simplified Non-stop Adjustment Mode:

  1 **In Layout Modification Mode, click the icon in the upper right corner of the Flow Display Pane.**

  2 **The following settings dialog box is displayed. Set the following settings.**



Editable items	Description [Factory default]
Flow display size	Sets the size of the Flow Display Pane. 16 to 128, [48]
Unit setting mode	Specifies the type of item settings dialog box to display when you click on a processing unit icon. • [Normal]: Displays a normal settings dialog box. • Simplified non-stop: Displays the Simplified Non-stop Adjustment Mode. • Nothing: No settings dialog box is displayed.

Detail Result Pane

This pane gives detailed measurement results in text format for the selected processing unit in the measurement flow.

The following elements are displayed.

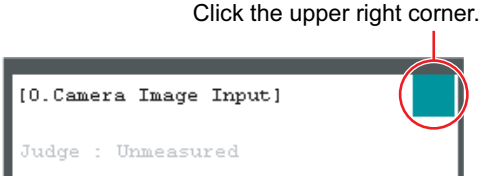
- Processing unit number, Item name
- Judgement

There are no parameters that can be set.

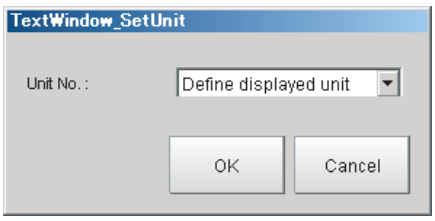
- **To Specify the Unit Number in the Detail Result Pane**

1 **Click the upper right corner of the Detail Result Pane.**

Click the upper right corner.



2 **The following dialog box is displayed. Use this to specify the display mode for the Detail Result Pane.**



Editable items	Description [Factory default]
Unit No.	[Define displayed unit] or Each unit number

Image Pane

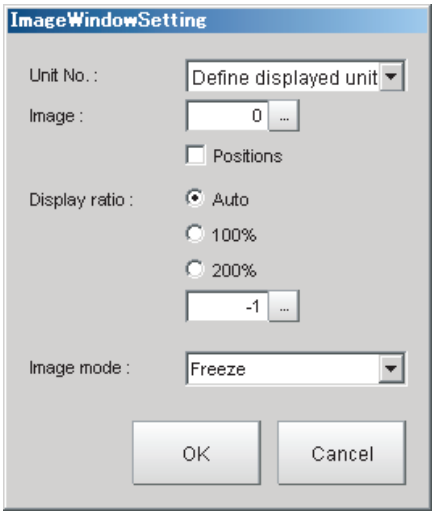
This pane shows the image.

- **To Change the Settings for the Image Display Pane**

1 **After you end Layout Modification Mode, click the upper left corner of the Image Display Pane.**

Click the upper left corner.

2 **A settings dialog box for the Image Display Pane is displayed.**



Editable items	Description [Factory default]
Unit No.	[Define displayed unit] or Each unit number If the [Define displayed unit] option is selected, the measurement image for the selected unit in the flow display is displayed. If you select another unit in the flow display, the measurement image that is displayed will also change according to the selected unit. If you specify a unit number, the measurement image will continue to display the image for that unit number, regardless of the selected unit in the flow display.
Image	Specify a sub image number for the measurement image to display. The sub image number that can be displayed depends on the processing item. *The menu is disabled if the [Display of position list] check box is selected. [0] to 100
Positions	Specify whether to display the position list. Select or [clear] the check box.
Display ratio	Specify the magnification at which to display the Image Display Pane. *The range for this parameter is from -1 to 1,600. [Auto], 100%, 200%, or numerical value
Image mode	Specify the image mode to use to display the image in the Image Display Pane. Through, [Freeze], or Last NG

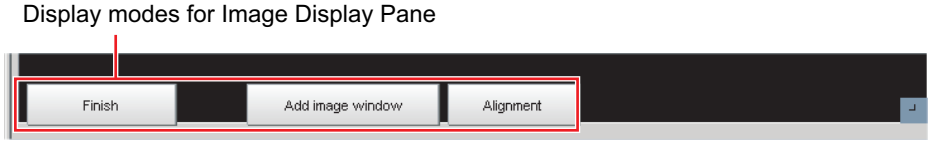
3 Click the [OK] button in the settings dialog box.

- **To Specify the Display Mode (e.g., Multiple Image Panes) of the Image Display Pane**

  You can display more than one Image Pane on any layout in the Image Display Pane (default: 0 to 23, you can use layout modification to limit the number of Image Panes).

1 In the state where you have exited layout modifications, click the upper right corner of the Image Display Pane.

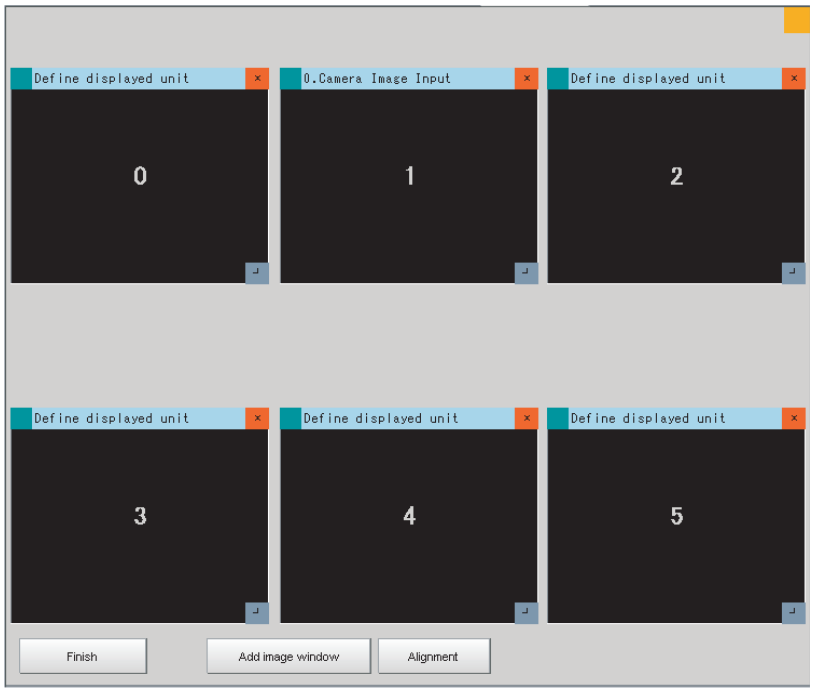
The button that sets the display mode for the Image Display Pane is displayed in the lower left of the Image Pane.



Button	Description
Add image window	Adds an Image Pane to display. Each click of this button adds an Image Pane with an unused image pane number for display. Click the [Alignment] button to tile the panes.
Alignment	Tiles the Image Panes. Click this button to align and evenly space the displayed Image Panes.
Finish	Ends the settings mode.

2 **Click the [Add Image Pane] button to add an Image Pane with the specified number to the display. Then click the [Align] button to align the Image Panes.**

Example: To display Image Pane numbers 0 to 5.



3 **Drag each pane to move or change its size.**

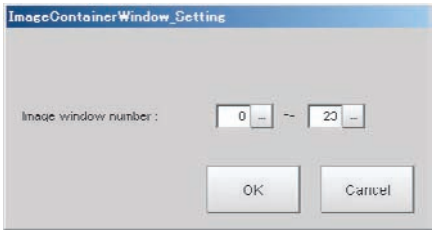
Click the ✕ button in the upper right corner of the Image Pane to delete it.

- **To Limit the Number of Image Panes Assignable to Each Layout**

  You can use layout modification to limit the range of Image Pane numbers that can be used on each layout.

  1 **When the Layout Modification Pane is active, click the icon in the upper right corner of the Image Display Pane.**

  2 **A settings dialog box for the Image Pane number range is displayed. Set following settings.**



Editable items	Description
Image window number	You can add up to 24 Image Panes within an Image Display Pane. 0 to 23

Error Pane

This pane shows errors, such as Camera connection errors.

If you add an Error Pane to a layout and an error occurs that normally causes an error dialog box to appear, an error string will appear in the Error Pane.
This is useful for remote operation or situations where an error dialog box would be difficult to close.
The latest error string (error message) is displayed on the top line of the error pane.
Also the error pane can display the error messages of up to 100.

The Error Pane is assigned to layout 8 by default.

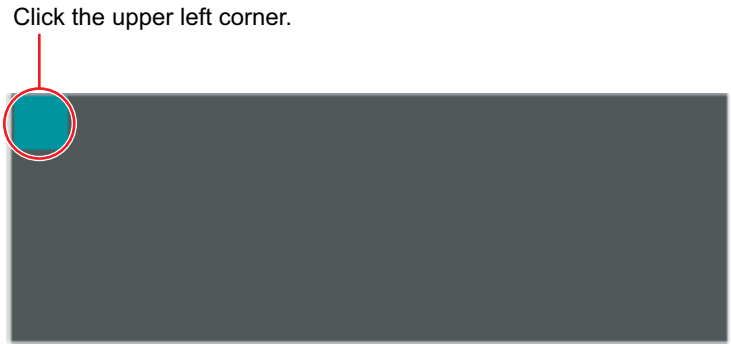
Note

The Error Pane displays error messages for the following errors.

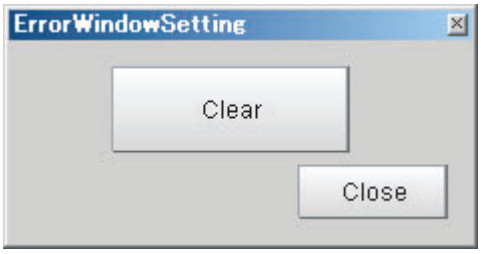
- System errors
- Fan or voltage errors
- Camera connection errors
- Changes made to the connected Camera
- Camera overcurrent detected
- Image logging disk write errors
- Output control timeout errors (parallel I/O, PLC Link, or EtherNet/IP)
- PLC Link communications errors
- Parallel I/O overcurrent detected
- Settings data load errors
- Settings data transfer errors
- Output timeouts (EtherCAT)
- Output buffer errors (EtherCAT)

- **Clearing the Error Display**

1 **In the state where you have exited layout modifications, click the upper left corner of the Error Pane.**



2 **The following dialog box is displayed. Click the [Clear] button.**



3 **Click the [Close] button on the settings dialog box to close the dialog.**

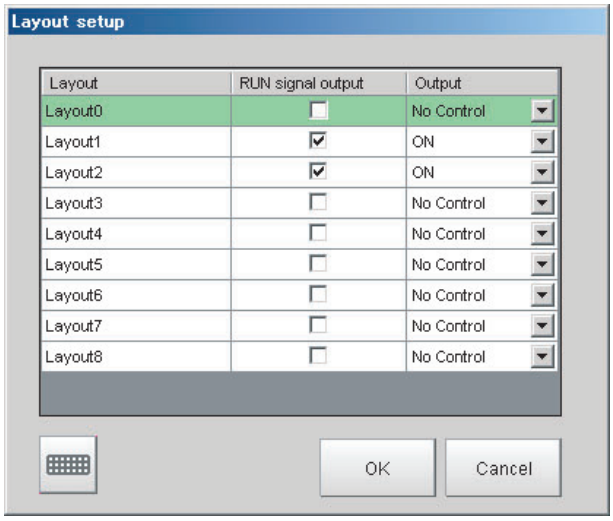
Setting the Behavior of Output Signals for Each Layout (Layout Settings)

For each layout, you can set the behavior of output signals after the layout is changed.

1 **Select [Layout setup] from the [Window] Menu for the main window for layout 0.**

2 **The following [Layout setup] dialog box is displayed.**

Set the name for layouts 0 to 8, and whether to use the RUN signal output and external output signals.



Item	Description
Layout	Layout name (up to 32 characters)
RUN signal output	The RUN signal for the layout is output if the check box is selected. Select or clear the check boxes.
Output	Set whether to output the execution results of Results Output processing units. • ON: Data is output when a Results Output processing unit is executed in the measurement flow during execution of a measurement. • OFF: Data is not output even when a results output processing unit is executed in the measurement flow for a measurement. • No Control: The state of the outputs that were in effect before the layout was changed are retained. If [Output] is selected immediately before the layout is changed, the outputs stay ON, and if [Output] is not selected, the outputs stay OFF.
Keyboard Button	Opens the onscreen keyboard so that you can change the layout name.

Changing Layout Numbers in the Main Window

You can change the layout in the Main Window to the layout number that was set in Layout Modification Mode. Use this when you need to work with more than one layout.

1 **Select [Switch layout] on the [Function] menu or from the tool buttons.**

2 **Select the layout number to change to and click the [OK] button to close the [Switch layout] dialog box.**

Item	Description
Layout number	Layout number (layout 0 to layout 8)

*1: You can also change the layout number with communications commands.

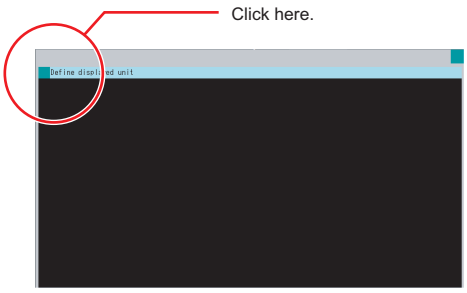
Troubleshooting

Symptom	Setting	Correction
The RUN signal is not output.	Layout setup	Select [Layout setup] from the [Window] menu and select the [RUN signal output] check box for the layout to be used.
The layout of the Image Display Area cannot be changed.	---	Click the button in the upper right corner of the Image Display Area if the Layout Modification Pane is not visible in the Main Window.
Cannot create a backup of the layout settings.	---	The layout data is included in the system data and saved in the Sensor Controller when you click [Data save]. To save or load the settings to or from a file, click [Save to file] or [Load from file] to save or load the system data and scene group 0 data or the system settings data.
An image of a window object remains in the Main Window when modifying the layout.	Layout modification	When you finish modifying the layout, the display will be updated and window objects will be displayed correctly.
Cannot output measurement results after the layout is changed	Layout setup	When "Output" is set to OFF in the layout setup of the layout that is used, measurement results are not output after startup or after the layout is changed. Even when the "Output" option is selected in the Measurement Pane of the Main Window, the "Output" setting in the layout setup is applied when the layout is changed. Before executing measurement, set "Output" to ON in the layout setup of the layout that is used.

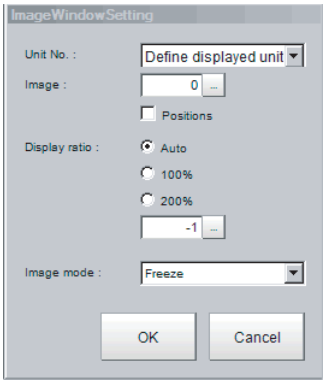
Changing the Image Mode and Other Display Contents

The display contents of the Image Display Pane can be changed in order to make the measurement status easier to understand.

1 **Click the upper left corner of the Image Display Pane that for which to change to display the settings dialog box for the Image Display Pane.**



2 **The settings dialog box for the Image Display Pane is displayed. Set the items.**



Editable item	Description [Factory default]
Unit No.	[Define displayed unit] or Unit No. If the [Define displayed unit] option is selected, the measurement image for the selected unit in the flow display is displayed. If you select another unit in the flow display, the measurement image that is displayed will also change according to the selected unit. If you specify a unit number, the measurement image will continue to display the image for that unit number, regardless of the selected unit in the flow display.
Image	Specify a sub image number for the measurement image to display. The sub image number that can be displayed depends on the processing item. The menu is disabled if the [Position] check box is selected. [0] to 100
Position	Specify whether to display the position list. Check box selected or [cleared].
Display ratio	Specify the magnification at which to display the Image Display Pane. The range for this parameter is from –1 to 1,600. [Auto], 100%, 200%, or numerical value
Image mode	Specify the image mode to use to display the image in the Image Display Pane. Through, [Freeze], or Last NG

Image Mode List

You can change the following items on the Main Window (layout 0).

Item	Description
Through	The latest image is always loaded from the Camera and displayed. When a through image is selected, saved images cannot be called up for measurement.
Freeze (default)	The image that was scanned in the immediately preceding measurement is displayed. Images are updated when measurements are performed.
Last NG	The latest NG error image resulting from an overall judgement is displayed. The latest measurement results are always shown in the overall judgement and measurement time. In this case, the overall judgement result and measurement time may conflict with the Camera image. Also, during continuous measurements, the last NG image cannot be displayed.

IMPORTANT

In the following cases, saved images cannot be called up for measurement.

- When a through image mode is selected in unused image display pane.
- When a through image mode is selected in hidden image display pane.

Note

- If a measurement trigger is input while using the multi-input function or immediately after BUSY is turned OFF (such as while the display is being updated), the last NG image cannot be displayed.
- [Display Last NG Image] serves a similar function as a processing item for which error images can be saved. If this processing item is used, the last NG image can be acquired without operation having any effect on operation.

Custom Dialogs

You can create a customized settings dialog box on the Sensor Controller and display it instead of the normal settings dialog boxes for each processing item.
This allows you to hide parameter settings that must not be changed during normal operation, and show only the parameters that are necessary. You can also create a settings dialog box that allows the operator to change the settings for more than one processing unit.

Flow of Use

From the [Custom dialog tool] menu, display the list of custom dialog boxes.

↓

Custom dialog tool

↓

Set the custom dialog box properties.

↓

Display and edit the normal dialog box edit display as necessary.

↓

Specify the target dialog box to use.

↓

Use the [Custom dialog] menu to call the target dialog box.

↓

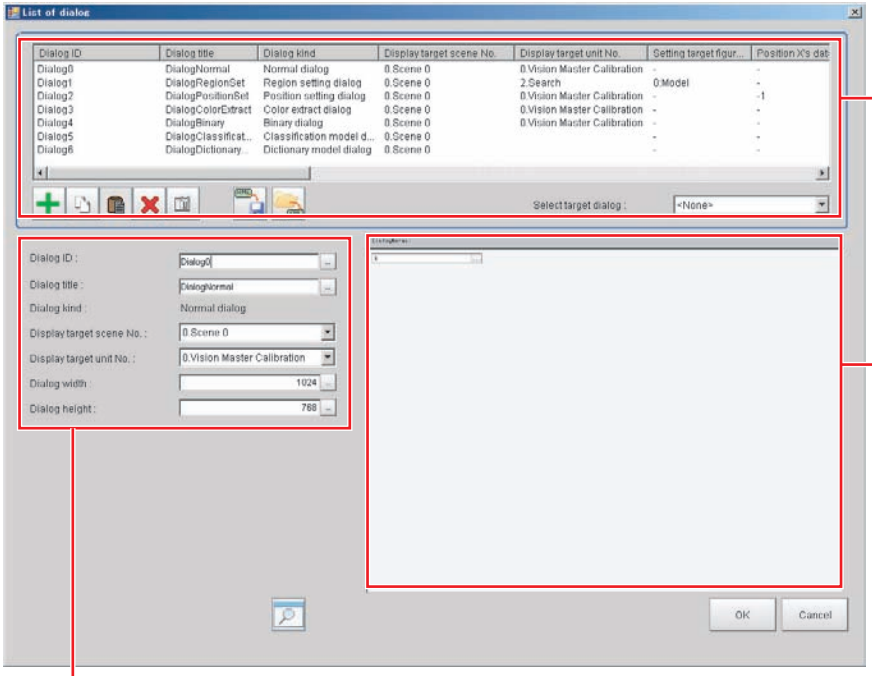
Save the settings of the custom dialog box ([Data save] or [Save to file])

Custom Dialog Tool

Use the following procedure to create a custom dialog box.

1 In the Main Window, select [Custom dialog tool] from the [Tool] menu.

The [List of dialog] display area is displayed. The Custom Dialog Tool has the following display areas: Dialog Box List Display Area, Dialog Box Properties Display Area, and Preview Area.



Dialog Box List Display Area

Preview Area

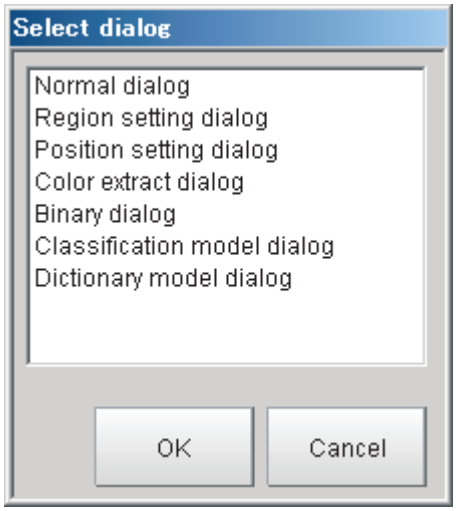
Dialog Box Properties Display Area for the List of Dialog Box Display Area.

Edit and Control Buttons for the List of Dialog Box Display Area.

Icon button	Function	Description
	Add dialog	Adds a dialog box. The [Select dialog] list is displayed.
	Copy dialog	Copies the selected dialog box.
	Paste dialog	Pastes the copied dialog box to the end of the list.
	Delete dialog	Deletes the selected dialog box.
	Edit normal dialog	If the selected dialog box is a normal dialog box, this button allows you to edit the contents that are displayed in the Edit Dialog Box Area. The [Edit Dialog] area is displayed.
	Save to dialog setting file	Saves the selected dialog box to a dialog box settings file (extension .xml).
	Load from dialog setting file	Loads the dialog box from the specified dialog box settings file (extension .xml) into the list of dialog boxes.
	Preview	Displays a working preview of the selected dialog box for testing.

2 In the list of dialog boxes, click the [Add dialog] button ().

The [Select dialog] list is displayed.



The types of dialog boxes that you can create are listed below.

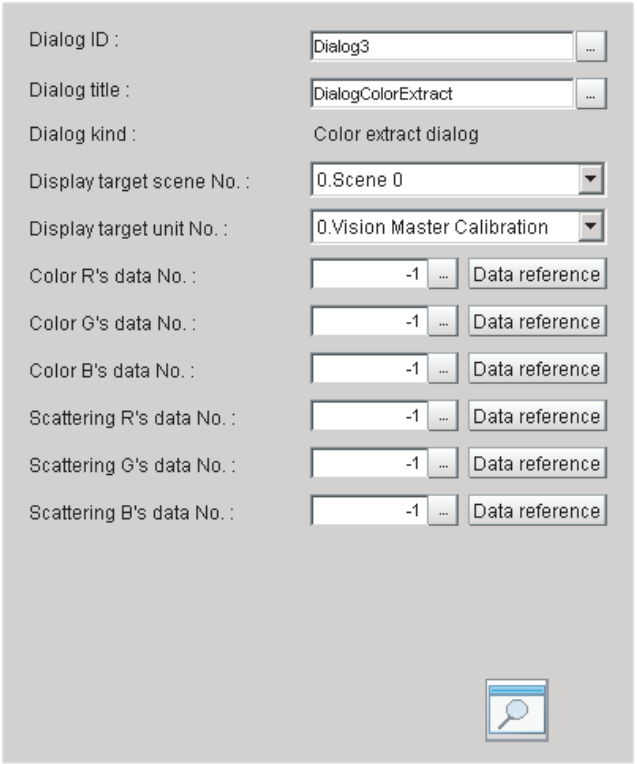
Dialog kind	Description
Normal dialog	This is a custom dialog box that you create with numeric input boxes, buttons, and other items.
Region setting dialog	This dialog box sets the measurement region for a processing item, such as the region setting for Search and Defect.
Position setting dialog	This dialog box sets the reference of a processing item, such as the reference position for Search and Edge Position.
Color extract dialog	This dialog box sets the color of a processing item, such as the color setting in Edge Position.
Binary dialog	This dialog box sets the binary setting of a processing item, such as the binary filter in Labeling.
Classification model dialog	You can register multiple models for this dialog box, such as for the model registration of the Classification processing item.
Dictionary model dialog	You can register multiple models for this dialog box, such as for the model registration of the Model Dictionary processing item.

3 Select the kind of dialog box to add, and then click the [OK] button.

The selected dialog box is added to the Dialog Box List Display Area.
Select the added dialog box from the dialog box list. The settings for the selected dialog box are displayed in the Edit Dialog Box Area while an image of the dialog box is displayed in the Preview Area.

4 Edit the settings for the dialog box. Some settings are common for all dialog boxes, and some are specific to individual dialog boxes.

Example: Dialog Box Properties Display Area for Color Extract Processing Item



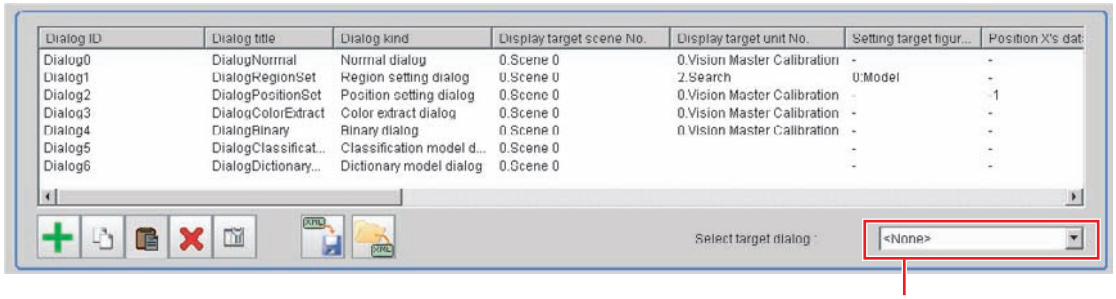
Refer to ▶ Settings in the Dialog Box Properties Display Area (p.98) for details on individual settings.

Note

If [Dialog kind] is set to [Normal dialog], click the [Edit normal dialog] button ( ) to edit the dialog box.
Reference: ▶ Editing a Normal Dialog Box (p.101)

5 **Edit the settings, and click the [Preview] button ( ) as necessary to confirm the edited dialog box.**

6 **Use the [Select target dialog] list to specify the ID of the dialog box to open.**



Select the target dialog box.

To open the dialog box that you specified in the [Select target dialog], select [Custom dialog] from the [Tool] menu.

7 **When you have finished editing, click the [OK] button to exit the List of Dialog Box Display Area.**

8 **Click [Data save] on the Main Window to save the settings.**

Note

To save the settings individually, click the [Save to dialog settings file] button ( ) to save the settings to a file. The saved file can be loaded with the [Load dialog settings file] button ( ).

Settings in the Dialog Box Properties Display Area

Common Settings Area

Parameter	Set value [Factory default]	Description
Dialog ID	The dialog box ID is set to a character string that is not assigned to any other dialog box.	The ID for the dialog box. To change the ID, click the [...] button and enter the desired characters.
Dialog title	A title that is based on the dialog box type is assigned.	The title for the dialog box. This appears in the title bar when the custom dialog box is displayed. To change the title, click the [...] button and enter the desired characters.
Dialog kind	---	The dialog box type is displayed.
Display target scene No.	[Scene 0]	This specifies the number of the scene for which to display the dialog box.

Parameter	Set value [Factory default]	Description
Display target unit No.	This is set to the unit that can be selected from the display target scene.	This specifies the number of the unit for which the dialog box is to be displayed instead of the default settings dialog box. The unit that can be specified depends on the type of the dialog box. Immediately after the dialog box is added, the unit with the lowest unit number out of the display target units in the display target scene is selected. If there is no display target unit, unit 0 is selected.

IMPORTANT

If you change the measurement flow after you create a custom dialog box, change the target unit number to match the edited measurement flow.

Individual Setting Region

• Normal Dialog Box Setting Region

Parameter	Set value [Factory default]	Description
Dialog width	100 to 1920 [1024]	Sets the width of the dialog box.
Dialog height	100 to 1080 [768]	Sets the height of the dialog box.

• Region Setting Dialog Box Setting Region

Parameter	Set value [Factory default]	Description
Setting target figure No.	[0]	Specifies the figure number for the unit to use for region setting. The number that you can select depends on the unit that is specified.

• Position Setting Dialog Box Setting Region

Parameter	Set value [Factory default]	Description
Position X's data No.	–1 to 999999999 [–1]	Specifies the data number on the X axis for the unit to use for position setting.
Position Y's data No.	–1 to 999999999 [–1]	Specifies the data number on the Y axis for the unit to use for position setting.
[Data reference] button	---	You can select the data number directly to enter it, or you can enter it with the [Data reference] button. Click the [Data reference] button to select the target data in the [Data reference] dialog box.

• Color Extract Dialog Box Setting Region

Parameter	Set value [Factory default]	Description
Color R's data No.	–1 to 999999999 [–1]	Specifies the R's data number for the color to specify for the unit to use for color extraction.
Color G's data No.	–1 to 999999999 [–1]	Specifies the G's data number for the color to specify for the unit to use for color extraction.
Color B's data No.	–1 to 999999999 [–1]	Specifies the B's data number for the color to specify for the unit to use for color extraction.

Parameter	Set value [Factory default]	Description
Scattering R's data No.	–1 to 999999999 [–1]	Specifies the R’s data number for the difference in the unit to use for color extraction.
Scattering G's data No.	–1 to 999999999 [–1]	Specifies the G’s data number for the difference in the unit to use for color extraction.
Scattering B's data No.	–1 to 999999999 [–1]	Specifies the B’s data number for the difference in the unit to use for color extraction.
[Data reference] button	---	You can select the data number directly to enter it, or you can enter it with the [Data reference] button. Click the [Data reference] button to select the target data in the [Data reference] dialog box.

- **Binary Dialog Box Setting Region**

Parameter	Set value [Factory default]	Description
Binary upper's data No.	–1 to 999999999 [–1]	Specifies the upper data number for the unit to use for binary filtering.
Binary lower's data No.	–1 to 999999999 [–1]	Specifies the lower data number for the unit to use for binary filtering.
Binary reverse's data No.	–1 to 999999999 [–1]	Specifies the reverse data number for the unit to use for binary filtering.
[Data reference] button	---	You can select the data number directly to enter it, or you can enter it with the [Data reference] button. Click the [Data reference] button to select the target data in the [Data reference] dialog box.

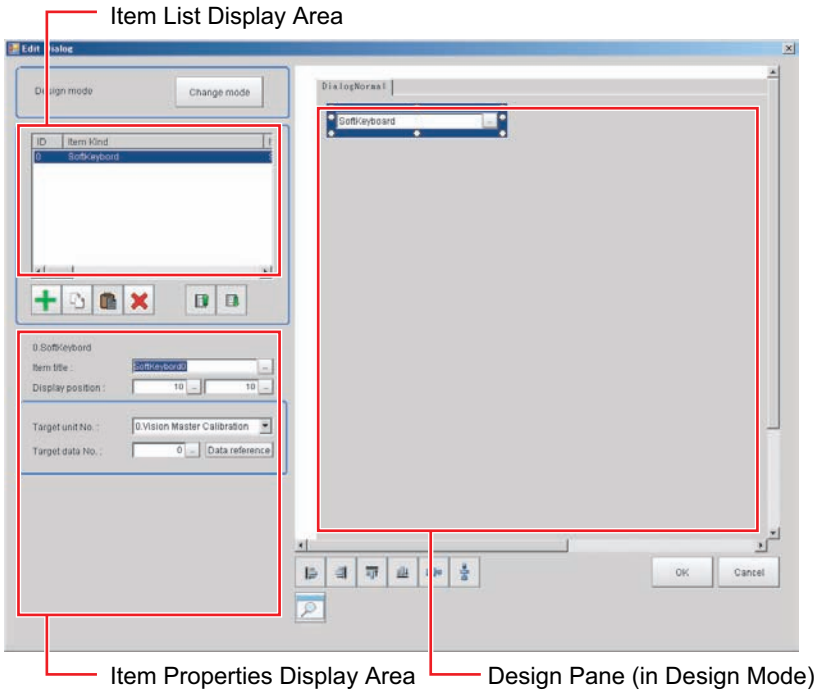
Editing a Normal Dialog Box

If [Dialog kind] is [Normal dialog], you can freely place objects, such as numeric entry boxes and buttons, anywhere in the Edit Dialog Box Area.

1 **In the Dialog Box List Display Area, select a dialog box that is of the normal dialog box type.**

2 **Click the [Edit normal dialog] button ( ).**

The [Edit Dialog] area is displayed.

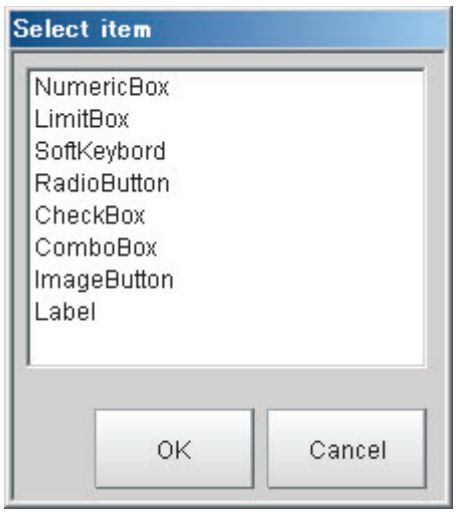


Edit and Control Buttons for the Edit Normal Dialog Box Window.

Icon button	Function	Description
Change mode	Change mode	Changes between the Design Pane and the List Pane.
	Add item	Adds an item. The [Select item] dialog box is displayed. Click the [OK] button to add the selected item to the end of the list.
	Copy item	Copies the item that is selected in the list.
	Paste item	Adds the copied item to the end of the list of items.
	Delete item	Deletes the item that is selected in the list.
	Move item up	Moves the selected item up one position in the list.
	Move item down	Moves the selected item down one position in the list.

3 In the list of dialog boxes, click the [Add item] button ().

The following [Select item] list is displayed.



The types of items that you can place on a normal dialog box are listed below.

Item kind	Description
Numeric Box	This setting object is used to enter numeric values. From the processing items, select the numeric data to use as the target for the entered numeric value.
Limit Box	This setting object is used to enter upper and lower limits. From the processing items, select the upper/lower limit to use as the target for the entered numeric value.
Soft Keyboard	This setting object is used to enter characters. From the processing items, select the character string data to use as the target for the entered a character string.
Radio Button	This setting object is used to enter button options. From the processing items, select the data to use as the target for the entered selection result.
Check Box	This setting object is used to enter check box options. From the processing items, select the data to use as the target for the check box selection result.
Combo Box	This setting object is used to enter from a combo box. From the processing items, select the data to use as the target for the entered selection result. You can also set the items to display in the combo box.
Image button	This setting object is used to accept inputs from a button. Buttons can be placed in any desired position and are used to execute predetermined events when the button is clicked.
Label	This setting object is used to display characters. You can display any character string in any desired position.

4 Select the kind of item to add, and then click the [OK] button.

The item is added to the Item List Display Area. At the same time, the added item is also displayed in the Design Pane.

5 To move the position where an added item is displayed in the Design Pane, drag the item to the desired position.

Drag the ❍ part of the border of the added item to change its display size.
Every item can be clicked, dragged, or selected as a group of items. Selected items are shown with a blue border.
The Design Pane of the Edit Normal Dialog Box Window has the following edit and control buttons.

Icon button	Function	Description
	Left	Aligns the left edges of the selected items to the left side of the first item that is selected.
	Right	Aligns the right edges of the selected items to the right side of the first item that is selected.
	Top	Aligns the top edges of the selected items to the top of the first item that is selected.
	Bottom	Aligns the bottom edges of the selected items to the bottom edge of the first item that is selected.
	Evenly space horizontally	Evenly adjusts the space on the left and right sides of the selected items.
	Evenly space vertically	Evenly adjusts the space on the tops and bottoms of the selected items.
	Preview	Displays a preview of the dialog box that is being created.

Note

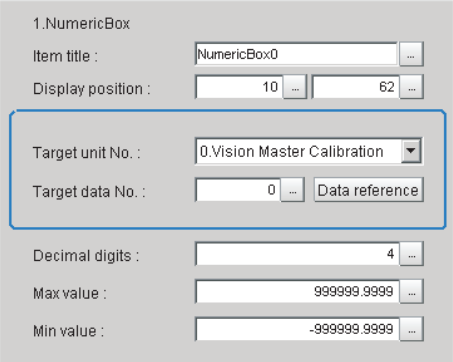
To view a list of the properties of an item in the Item List Display Area in Design Mode, click the [Change mode] button. The Item List Display Area changes to a list display.
Click the [Change mode] button in Edit Mode to return to the Design Mode.

6 Every time you click an item in the list, that item changes between selected (inverted display) and not selected.

If only one item is selected (inverted display), the settings for that item type are displayed in the Edit Item Area.

7 **Edit the settings for the item. Some settings are common for all items, and some are specific to individual items.**

Example: Item Properties Display Area for a Numeric Box



Refer to ▶ Settings in the Item Properties Display Area (p.105) that is described later in this section for details on all settings.

8 **When you have finished editing, click the [OK] button to exit the Edit Dialog Box Window.**

Settings in the Item Properties Display Area

- **Common Settings Area**

Parameter	Set value [Factory default]	Description
Item title	This is set to a character string that is not assigned to any other item in the list when the item is added.	To change the title of the dialog box item, click the [...] button and enter the desired characters.
Display position	X: 0 to 1920, [–] Y: 0 to 1080, [–]	These are the coordinates of the upper left corner of the dialog box item. The upper left corner of the dialog box is 0, 0. The display position is also updated when you drag an item to a new position in the Design Pane.

- **Individual Setting Region**

Numeric Box Setting Region

Parameter	Set value [Factory default]	Description
Target unit No.	0 to Number of units in the display target scene - 1 [0]	Specify the unit to be the target for the numeric setting. The units that are registered for the scene that is the target of the normal dialog box will appear in the combo box. Select the processing unit to set.
Target data No.	0 to 9999999 [0]	Specify the data to be the target for the numeric setting. Enter the data number for the unit that is specified by the target unit number.
[Data reference] button	---	You can select the data number directly to enter it, or you can enter it with the [Data reference] button. Click the [Data reference] button to select the target data in the [Data reference] dialog box.
Decimal digits	0 to 4 [4]	Enter the number of digits after the decimal place that can be entered by the numeric setting.
Max value	–999999.9999 to 999999.9999 [999999.9999]	Enter the maximum value that can be entered with the numeric setting.

Parameter	Set value [Factory default]	Description
Min value	–999999.9999 to 999999.9999 [–999999.9999]	Enter the minimum value that can be entered with the numeric setting.

Limit Box Setting Region

Parameter	Set value [Factory default]	Description
Target lower unit No.	0 to Number of units in the display target scene - 1 [0]	Specify the unit to be the target for the upper/lower limit setting. The units that are registered for the scene that is the target of the normal dialog box will appear in the combo box. Select the processing unit to set. Set the same unit number as the upper target unit number.
Target upper unit No.	Target upper unit No. [0]	Specify the unit to be the target for the upper/lower limit setting. The units that are registered for the scene that is the target of the normal dialog box will appear in the combo box. Select the processing unit that you wish to set. Set the same unit number as the lower target unit number.
Target data No.	0 to 9999999 [0]	Specify the data to be the target for the upper/lower limit setting. Enter the data number for the unit that is specified by the target unit number.
[Data reference] button	---	You can select the data number directly to enter it, or you can enter it with the [Data reference] button. Click the [Data reference] button to select the target data in the [Data reference] dialog box.
Decimal digits	0 to 4 [4]	Enter the number of digits after the decimal place that can be entered by the upper/lower limit setting.
Max value	–999999.9999 to 999999.9999 [999999.9999]	Enter the maximum value that can be entered with the upper/lower limit setting.
Min value	–999999.9999 to 999999.9999 [–999999.9999]	Enter the minimum value that can be entered with the upper/lower limit setting.

Soft Keyboard Setting Region

Parameter	Set value [Factory default]	Description
Target unit No.	0 to Number of units in the display target scene - 1 [0]	Specify the unit to be the target for the character string setting. The units that are registered for the scene that is the target of the normal dialog box will appear in the combo box. Select the processing unit to set.
Target data No.	0 to 9999999, [0]	Specify the data to be the target for the character string setting. Enter the data number for the unit that is specified by the target unit number.
[Data reference] button	---	You can select the data number directly to enter it, or you can enter it with the [Data reference] button. Click the [Data reference] button to select the target data in the [Data reference] dialog box.

Radio Button Setting Region

Parameter	Set value [Factory default]	Description
Target unit No.	0 to Number of units in the display target scene - 1 [0]	Specify the unit to be the target for the radio button setting. The units that are registered for the scene that is the target of the normal dialog box will appear in the combo box. Select the processing unit to set.
Target data No.	0 to 9999999 [0]	Specify the data to be the target for the radio button setting. Enter the data number for the unit that is specified by the target unit number.
[Data reference] button	---	You can select the data number directly to enter it, or you can enter it with the [Data reference] button. Click the [Data reference] button to select the target data in the [Data reference] dialog box.
Set value	0 to 9999999 [0]	Enter the numeric value to set in the target data when the radio button is selected. The numeric value that is set here is the set value for that selected radio button that is applied to the data for the target data number when the operator changes the radio button selection on the actual custom dialog box.

Check Box Setting Region

Parameter	Set value [Factory default]	Description
Target unit No.	0 to Number of units in the display target scene - 1 [0]	Specify the unit to be the target for the check box setting. The units that are registered for the scene that is the target of the normal dialog box will appear in the combo box. Select the processing unit to set.
Target data No.	0 to 9999999 [0]	Specify the data to be the target for the check box setting. Enter the data number for the unit that is specified by the target unit number.
[Data reference] button	---	You can select the data number directly to enter it, or you can enter it with the [Data reference] button. Click the [Data reference] button to select the target data in the [Data reference] dialog box.

Combo Box Setting Region

Parameter	Set value [Factory default]	Description
Target unit No.	0 to Number of units in the display target scene - 1 [0]	Specify the unit to be the target for the combo box setting. The units that are registered for the scene that is the target of the normal dialog box will appear in the combo box. Select the processing unit to set.
Target data No.	0 to 9999999 [0]	Specify the data to be the target for the combo box setting. Enter the data number for the unit that is specified by the target unit number.
[Data reference] button	---	You can select the data number directly to enter it, or you can enter it with the [Data reference] button. Click the [Data reference] button to select the target data in the [Data reference] dialog box.

Parameter	Set value [Factory default]	Description
[Item setting] button	---	Set the item name and the data that corresponds to the item name to display in the combo box. Make sure the number of items and the number of data match. • Items: Enter the name of the items to display in the combo box. • Data No.: Enter the target data numbers that correspond to each item in the combo box.

Image Button Setting Region

Parameter	Set value [Factory default]	Description
Event	• [OK] • Cancel • Launch dialog	Set the event to execute when the button is clicked. • OK: Overwrites the data setting and closes the dialog box. • Cancel: Cancels the data setting and closes the dialog box. • Launch dialog: Launches the specified custom dialog box.
Target dialog ID	The dialog box ID at the beginning of the list of dialog boxes (except the current dialog box)	This setting is displayed if you select [Launch dialog] for the event. When the button is clicked, select the ID for the dialog box to launch.

IMPORTANT

If you change the measurement flow after you create a custom dialog box, change the target unit number to match the edited measurement flow.

Launching Your Custom Dialog Box

1 **In the Main Window, select [Custom dialog] from the [Tool] menu.**

The custom dialog box that is set for [Select target dialog] is displayed.

2 **Change the setting for the processing unit in the custom dialog box that is displayed.**

Remeasuring Saved Images

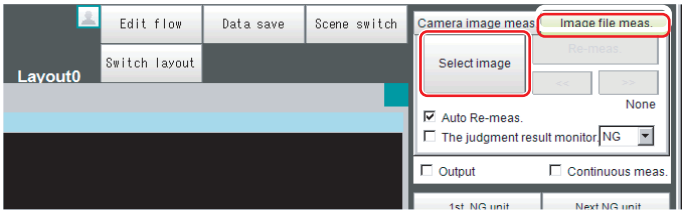
Images from when measurements are performed, including test measurements, can be saved. Remeasurement can be performed with saved images after conditions are adjusted in order to check whether the adjustment is appropriate.
The logging function is used to save images.
Reference: ▶ Logging Measurement Values and Measurement Images (p.113)
Images that can be remeasured include images saved in the Controller and images saved in external memory.

1 **Click the upper left corner of the Image Display Pane, set the [Image mode] to [Freeze], and then click the [OK] button.**

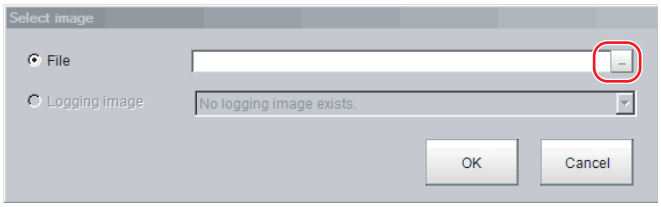
2 **In the Main Window (layout 0), click the [Image file meas.] button in the Measurement Pane.**

3 **Click the [Select image] button.**

The [Select image] dialog box is displayed.



4 **Click the [...] button and specify the file to display.**



Parameter	Description
File	Specify images saved in the external memory or in the RAM disk.
Logging image	Specify images that are logged in the Controller memory.

5 **The selected image is displayed at the lower left of the File Explorer Pane.**

When there are multiple camera images in a file, as for a logging image when multiple cameras are connected, use the [<<] and [>>] buttons to switch images.

6 **Click [OK].**

The path and file name of the image are displayed under [Select image].

7 **In the Main Window (layout 0), click the [Re-meas.] button under [Image file meas.] .**

Measurement of the selected image is performed.

Note

About Automatic Remeasurement

Displayed images can be automatically remeasured by selecting [Auto Re-meas.].

IMPORTANT

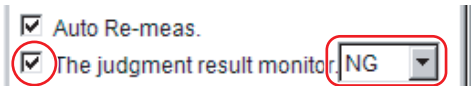
When remeasuring an image with the Controller, it is necessary to have a Camera connected that is appropriate to the image size. For example, if the image file for remeasurement contains 2-megapixel images and a 0.3-megapixel Camera is connected to the Controller or if a Camera is not connected, measurement will not be performed correctly due to a memory deficiency. Perform remeasurement after connecting a Camera appropriate to the image size.

Improving Adjustment Efficiency

You can make adjustments more efficiently when you need to measure a large amount of image samples and classify or perform adjustments with each judgement.
Files in which there are both NG error files and OK files can be continuously remeasured automatically, with the system stopping at images with a specified condition (OK/NG) and moving those files.

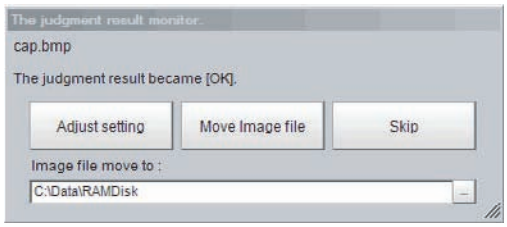
1 **In the Main Window (layout 0), click the [Image file meas.] button in the Measurement Pane.**

2 **Select [The judgment result monitor] and set the judgement conditions for identification.**

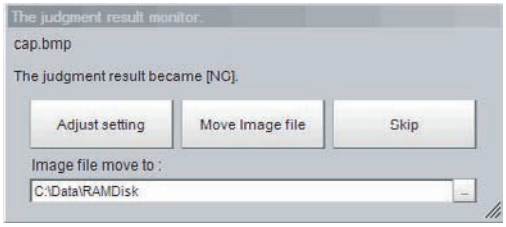


If the specified judgement condition is achieved when continuous measurement is performed, measurement stops and the following message is displayed.

If OK Is Selected



If NG Is Selected



3 **Select the processing for the measured image.**

Adjusting Flow Settings

Click the [Adjust setting] button.

Moving Images to Specified Folder

Specify the folder to move the image file to.
Click the [Move Image file] button.
Click the [Skip] button to skip processing and remeasure the next image.

Changing Judgement Conditions without Stopping Measurement

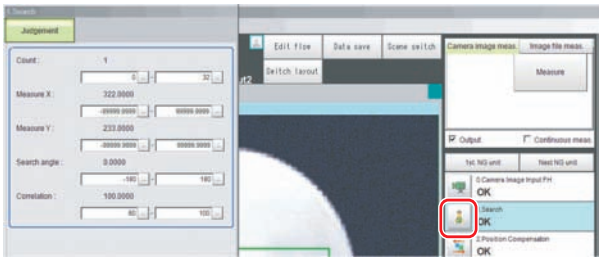
Using the simplified non-stop adjustment function makes it possible to change the judgement conditions of processing units for the currently displayed scene without stopping the measurement processing.

Using the Simplified Non-stop Adjustment Mode

1. **Select [Layout modification] from the [Window] menu.**
2. **In the Layout Modification Mode, select the layout number to use the Simplified Non-stop Adjustment Mode.**
3. **Click the upper right corner of the Flow Display Pane.**
4. **Change the [Unit setting mode] under [Flow Windows_Setting] to [Simplified non-stop], and then click the [OK] button.**
5. **Click [Setting end] on the Layout Modification Pane.**

   Simplified Non-stop Adjustment Mode is enabled for the modified layout window.
6. **Click the icon of the processing unit with the judgement condition to be adjusted.**

   The [Judgement] dialog box for the selected processing unit is displayed.
   If you click the icon of the processing unit that does not have a [Judgement] setting item, the Judgement Pane is not displayed.



7. **Modify the judgement conditions for the processing units.**
8. **Click [OK].**

   The Judgement Pane closes, and the display returns to the Main Window.
   The changed contents are shown in the displayed scene.
   If you need to modify the judgement conditions for more than one processing unit, repeat steps 6 to 8.

Disabling the Simplified Non-stop Adjustment Mode

If the Simplified Non-stop Adjustment Mode is enabled, you cannot make detailed parameter settings for processing units.
To make detailed parameters for processing units, stop the measurement process and disable the Simplified Non-stop Adjustment Mode.

1. **Select [Layout modification] from the [Window] menu.**
2. **In the [Layout number] on the Layout Modification Pane, select the layout number for which to disable the Simplified Non-stop Adjustment Mode.**
3. **Click the upper right corner of the Flow Display Pane.**
4. **Change the [Unit setting mode] under [Flow Windows_Setting] to [Normal], and then click the [OK] button.**
5. **Click [Setting end] on the Layout Modification Pane.**

   Simplified Non-stop Adjustment Mode is disabled for the modified Layout Window.

IMPORTANT

While simplified non-stop adjustment is being performed, do not change the scene group or scene.

Changing Regions as a Batch [Shift area]

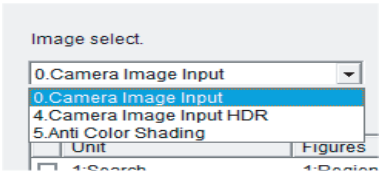
Figure data for multiple processing items can be changed as a batch.

1 **In the edit flow window, click the [Shift area] button.**

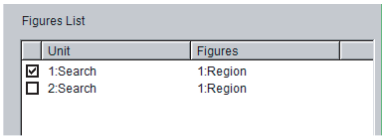
The [Shift area] dialog box is displayed.

2 **Select the processing item in which to change the region.**

Only image setting processing items included in [Input image] and [Compensate image] are displayed.

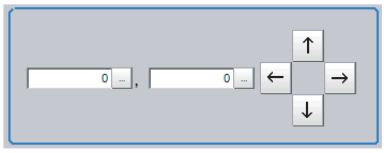


3 **Select the registration region to change.**



4 **Click [Move] and input the value or click the arrows to move the image.**

Images can also be directly dragged to move them.



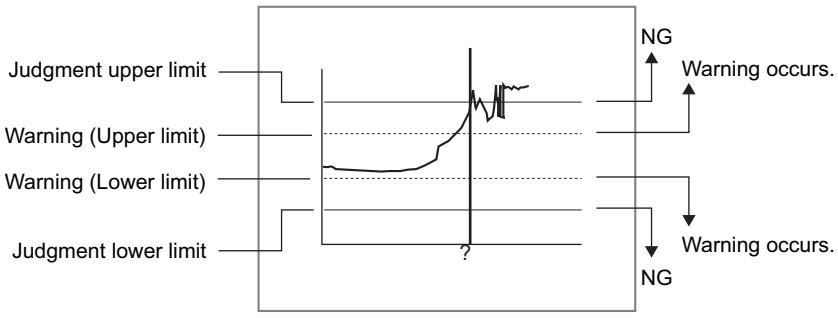
5 **Click [OK].**

The change is registered.

Monitoring Measurement Value Trends

By monitoring the trends in measurement values, the occurrence of defects can be prevented in advance and this information can also be helpful in NG error occurrence cause analysis. Use the [Trend Monitor] processing item to monitor the measurement values.

Refer to ▶ *Trend Monitor* in the *Vision System FH/FZ5 Series Processing Item Function Reference Manual* (Cat No. Z341).



Note

- If the measurement value is within the alarm range, the [Warning] message is displayed.
- If a result output-related processing item is used, this allows for output to external devices when a warning occurs.
- Through trend monitor judgement, trends can be managed and NG error images can be saved.
  To save only NG error images identified by trend monitor judgement, create settings so that overall judgements from processing units other than [Trend Monitor] are not included in the judgement.

Logging Measurement Values and Measurement Images

What Is Logging?

Logging is used to save Camera input images or measurement results when executing measurements.
Depending on what is being saved, there is image logging (camera input images) and data logging (measurement results)
There are two logging functions, system logging and logging that is performed by the processing item.

Image Logging

- **System Logging**
  To enable system logging, set the logging setting in the system settings.
  The image logging process is executed after processing of the measurement flow is completed.
- **Processing Item Logging**
  To enable processing item logging, register the processing items in the measurement flow.
  This logging function allows you to more flexible set the logging conditions than system logging. If there is more than one Image Logging Unit in the measurement flow, image logging is performed based on the logging conditions for the most recently executed Image Logging Unit.

Data Logging

- **System Logging**
  Data logging does not have a system logging function. Use a Data Logging processing item.
- **Processing Item Logging**
  To enable processing item logging, register the processing items in the measurement flow.
  If there is more than one data logging unit in the measurement flow, data logging is performed based on the logging conditions for the most recently executed data logging unit.

Reference: ▶ Specifying Logging Conditions for Images (p.120)

IMPORTANT

- If you use logging, the measurement processing time will take longer for the time required for the logging process. If you need to continuously execute measurements that include image logging, allow enough time between measurement triggers after the first trigger to allow time for image processing.
  Reference: ▶ Calculating the Measuring Interval ([Logging]) (p.119) for information on measuring the time for image logging processing.
- Processing item image logging and system image logging cannot be used at the same time.
  If you prefer to use the Image Logging processing item, click [System Settings] – [Other] – [Logging setting] – [Image logging], and then select [None].

This section describes system logging in detail.
Refer to the ▶ *Vision System FH/FZ5 Series Processing Item Function Reference Manual* (Cat No. Z341) for information on processing item logging
The processing items that can be logged are described below.

Image Conversion Logging

This logging saves Camera images.
Image conversion logging allows you to draw a rectangle around a portion of the image to save. You can also

specify the format of the saved image (BMP or JPG).

Data Logging

Measurement data is saved.

System Logging

There are 2 different logging methods.

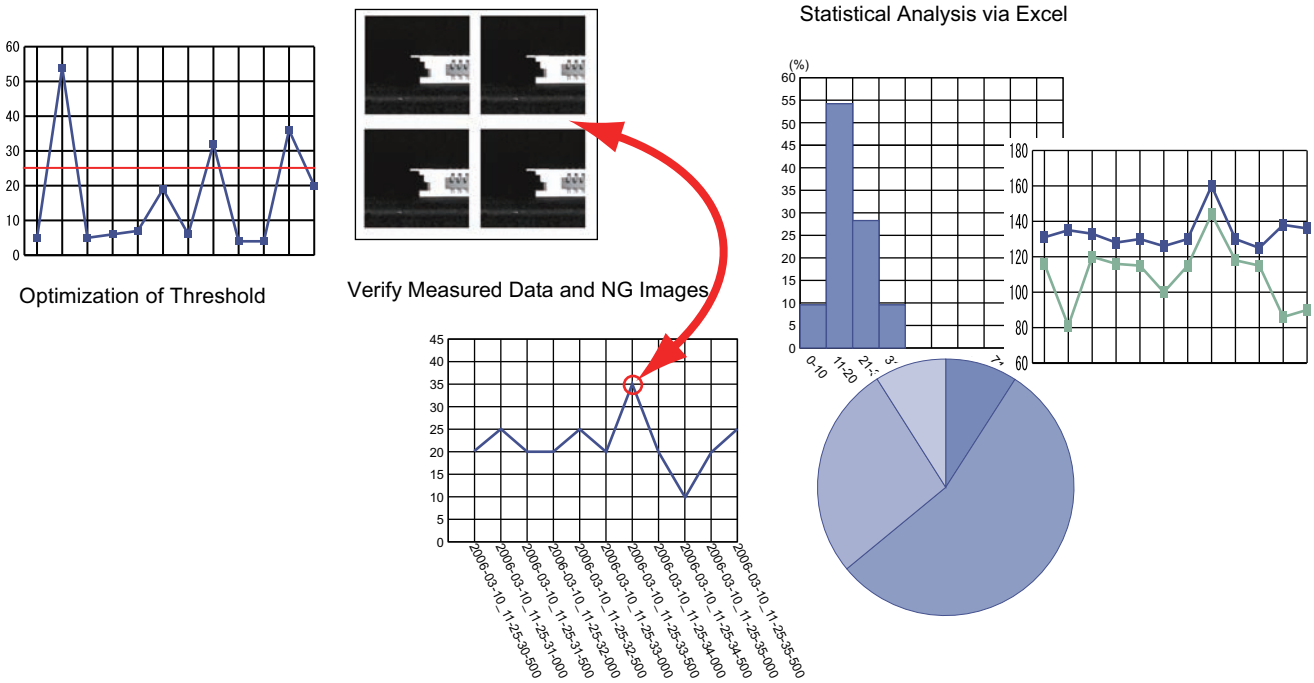
Logging Images That Are Currently Displayed

Reference: ▶ Logging Current Image [Save Last Logging Image] (p.114)

Automatically Logging Images During Measurements

Reference: ▶ Setting Logging Conditions [Logging Setting] (p.115)

Images and measurement data can be saved in external memory, which makes them useful for the following kinds of adjustment.



Logging Current Image [Save Last Logging Image]

This section explains the method for logging the latest input image that is currently being displayed.

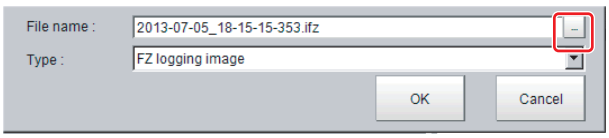
1 **In the Main Window, select [Save last logging image] from the [Function] menu.**

The File Explorer is displayed.

2 **Set the logged image save destination.**

Specify the image file save destination (RAM disk or external memory).

3 **Edit the file name as required.**



4 **Click [OK].**

After the logging operation is complete, the Save Last Logging Image Dialog Box closes.

Setting Logging Conditions [Logging Setting]

Set the logging timing and the save destination.

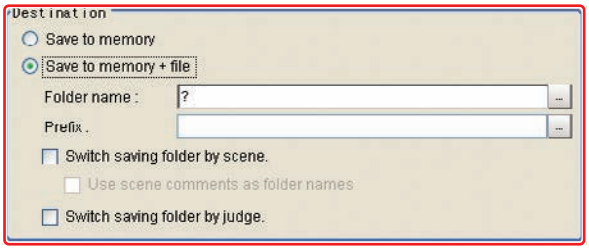
1 **In the Main Window, select [System settings] – [Other] – [Logging setting] from the [Tool] menu.**

The [Logging settings] dialog box is displayed.

2 **Set the image logging conditions.**

Parameter	Set value [Factory default]	Description
Image logging	[None]	No images are saved. When logging images with the processing item [Image Logging], select [None].
	Only NG	Only images with an overall judgement of NG are saved.
	All	All measured images are saved. Some images may not be saved if [Measurement] is set in [Logging priority] in step 4.

3 **Set the logged image save destination.**



Note

In order to perform logging quickly, image files are first saved in the Controller memory (RAM). The Controller memory (RAM) for saving images is a ring memory. If the maximum number of save images is reached, images will be overwritten starting with the oldest saved image if further images are saved.
Reference: ▶ About Number of Logging Images (p.856)
Also, the Controller memory is cleared if the power supply is turned OFF.
To keep images, select [Save to memory + file] and save images to external memory.

Parameter	Set value [Factory default]	Description
Destination	[Save to memory]	Saves data to the Controller memory (RAM).
	Save to memory + file	Images that are saved to the Controller memory (RAM) are saved as files to a external memory or to the RAM disk in the Controller.

When [Save to memory + file] in [Destination] is selected, set the destination and file names.

Parameter	Set value [Factory default]	Description
Folder name	• [RAM Disk] • Drive name of external memory device	Specify the image file save destination (RAM disk or external memory). Logged images are saved in the specified save destination folder. (Maximum number of characters: 128 single-byte characters)
Prefix	---	Set the prefix for the saved file name. (Maximum number of characters: 32 single-byte characters) The set character string is added at the beginning of the name of the save file.
Switch saving folder by scene	• Selected • [Not selected]	If this check box is selected, folders that correspond to scene numbers are automatically created and image files are divided by scene and saved.
Use scene comments as folder names	• [Selected] • Not selected	If this check box is selected, folders to which image files are allocated are named using the scene names and scene group names at the time of measurement execution. If a scene name or scene group name contains a character not supported for file names or folder names, a logging error is generated when logging is executed.
Switch saving folder by judge	• Selected • [Not selected]	If this check box is selected, OK/NG folders are automatically created and image files are divided by the overall judgement and saved.

[Save to Memory + File] Setting Example and Save Destination

Setting example	Destination
• Folder name: USB disk • Prefix: image_ • [Switch saving folder by scene]: selected • [Switch saving folder by judge.]: Selected	The settings in the example on the left are applied as shown below. • OK image save destination: \USBDisk\S000-000\OK\image_(Measurement ID).IFZ • NG image save destination:\USBDisk\S000-002\NG\image_(Measurement ID).IFZ

4 Set the image logging priority conditions.

When the measurement takt time is short, a time lag may occur in writing from the Controller memory (RAM) to the RAM disk in the Controller and a temporary lack of available space in the Controller memory may occur. Select whether logging or measurement has priority at these times.

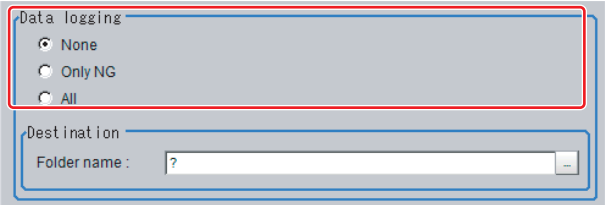
Parameter	Set value [Factory default]	Description
Logging priority	[Logging]	When there is no available space in the Controller memory (RAM), subsequent measurement cannot be received until memory becomes available. All measurement target images are logged, but the measurement takt time becomes longer.
	Measurement	Measurements will continue even if there is no available space in the Controller memory. New logging is not performed until space becomes available in the Controller memory (RAM). The measurement takt time is maintained, but some measurements may not be logged.

Note

To save all images that are logged, set the operation mode of the Sensor Controller to [High-speed logging mode] and the Logging setting (logging condition) to [Logging]. At the same time, select [Tool] – [System Settings] – [Startup], and turn OFF [Parallel execute] under the operation mode In the startup settings.

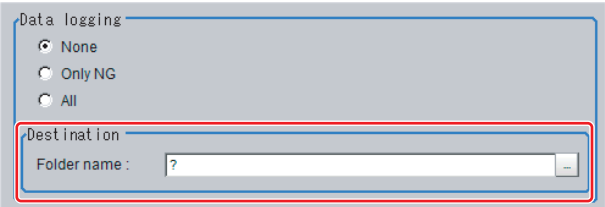
5 Set the data logging conditions.

The data format is set with the Data Logging processing item.



Parameter	Set value [Factory default]	Description
Data logging	[None]	Measurement data is not saved.
	Only NG	Measurement data is saved when an NG error occurs in a unit before a [Data Logging] unit. If an NG error occurs after the [Data Logging] processing unit, data logging is not performed.
	All	All measurement data is saved.

6 Set the logged data save destination.



Parameter	Set value [Factory default]	Description
Folder name	• [RAM Disk] • USBDisk	The data is saved in the specified destination folder (RAM disk or USB disk). Set the file name with the [Data Logging] processing unit. (Maximum number of characters: 128 single-byte characters)

7 Click [Apply].

IMPORTANT

- Logging images saved in the Controller memory are overwritten starting with the oldest image if the upper limit for the number of save images is exceeded.
  Reference: ▶ About Number of Logging Images (p.856)
- The data saved in the Controller memory or RAM disk is deleted when the Controller is restarted.
- If [Camera Image Input] is used several times in a flow, the image from the last [Camera Image Input] is saved.
- If external memory or a network drive is specified as the save destination, the processing time may be longer or may fluctuate. Be sure to check functionality thoroughly before starting operation.
- If images and data are logged to a network drive, the use of the multi-input function and other heavy measurement loads on the Controller may slow down communications and cause logging errors. In this case, set the measurement takt time so that there is some leeway.

Note

Image Logging

- When the number of files in the save destination folder increases, the time needed to save images increases.
- If image transfer is disabled using the Camera selection setting for the [Camera Image Input] unit, black images are saved instead of images from the disabled Camera.

Number of Images That Can Be Saved

- This will depend on the size and resolution of the images and the number of Cameras connected.
- The number of images that can be saved on the RAM disk in the Controller or a USB disk depends on available space.
  - If the RAM disk is selected, the memory available in the RAM disk
  - If USB disk is selected, it depends on the USB disk capacity.
- The following restrictions apply to a USB disk. (There is no limit if an NTFS-formatted USB disk is used with the FH/FZ5-11□□.)
  - When saving image files directly under the root directory, the number of images that can be saved is about 126.
  - When saving in sub-folders (e.g., \USBDisk\SUB), a maximum of 999 images can be saved in each folder. Change to a different folder to save another 999 images up to the maximum memory capacity.

Loading Data to a Computer

- Default settings are set so that logging data is saved in the RAM disk in the Controller.
  When logging data is loaded to a computer, set [USBDisk] as the save destination. Logging data is first saved to the RAM disk in the Controller. It can then be copied from the RAM disk and saved to the USB disk using [Copy files] in [Save to file].

Calculating the Measuring Interval ([Logging])

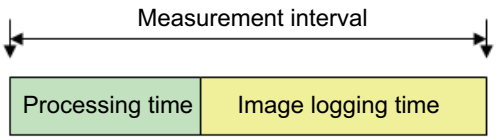
If the [Logging priority] is set to [Logging] (default), all target images that are measured are logged. This may extend the processing time by the time that is required to generate the image files.
The method that is used to calculate the measurement interval that allows logging without affecting the processing time is described below.

Measurement Interval Conditions

The measurement interval must satisfy the following condition.

Processing time + Image logging time < Measurement interval

- Processing time: The time that is displayed in the upper left corner.
- Image logging time: The time that is required to generate and save a single image file.



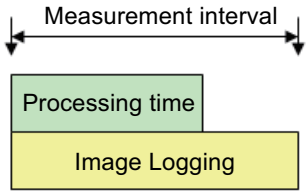
Example:

If the processing time is 100 ms and the image logging time is 150 ms, set the measurement interval to at least 250 ms.

Note: If the Operation Mode Is Set to [High-speed Logging Mode]

In High-speed Logging Mode, images can be logged while measurement is in progress.
The measurement interval must satisfy the following condition.

Image logging time < Measurement interval



- **Example:**

  If the processing time is 100 ms and the image logging time is 150 ms, set the measurement interval to at least 150 ms.

Specifying Logging Conditions for Images

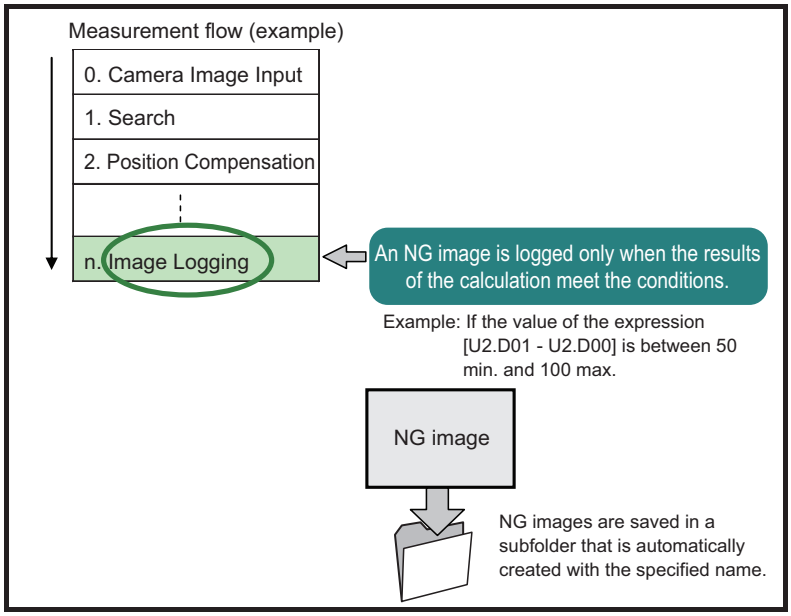
If you need to specify conditions for logging images, you should use the processing item logging function.
For example, the system logging function logs all images if the overall judgement is NG or OK.
Processing item logging (image logging), logs images only when a condition is met.
The following examples illustrate how processing item logging can be used.

Possible Saving Conditions

Images can be logged when measurement data or result data of a calculation falls within a specified range.
Example 1: Image logging is performed if the individual judgements of three measurement units U1, U3, and U5 are NG.
Example 2: Image logging is performed if the edge with for a measurement unit is below a set value.

Destination of Images

This diagram shows the destination for saving if [Save to memory + file] is selected in the system image logging settings.



IMPORTANT

If [Logging setting] is set to [Image logging], select [None] (default).

Note

If you set more than one Image Logging processing unit in the measurement flow, the most recently executed image logging is enabled.

Analyzing Logged Data

Acquired data is referred to and processed, and settings are analyzed.

Checking Logged Data with a Computer

This section uses the example of saving logging data in USB memory.

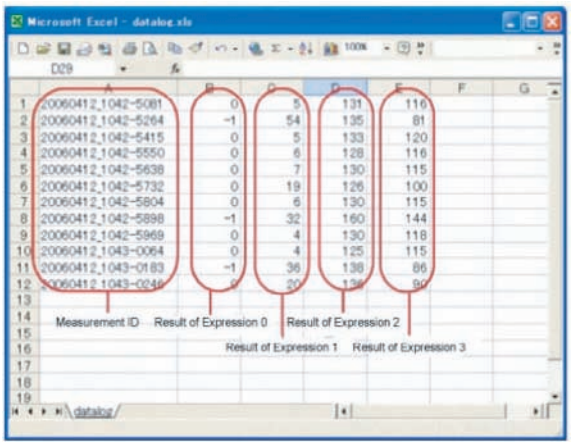
1 Copy the logged data in the external memory to the computer.

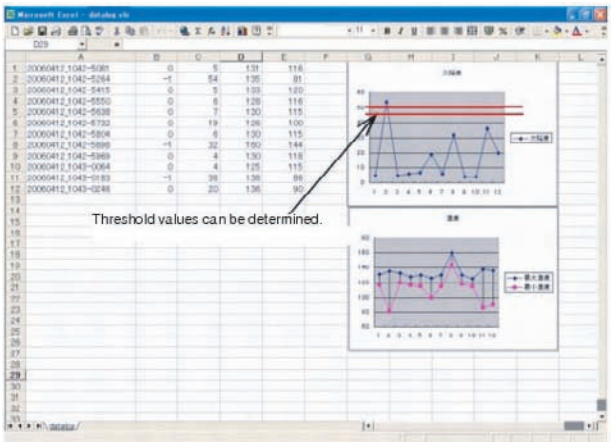
2 Open the copied data in the folder using an application associated with the extension (.csv).

This procedure describes an example for data displayed in Excel.

3 Use Excel graphing and functions to process and analyze data.

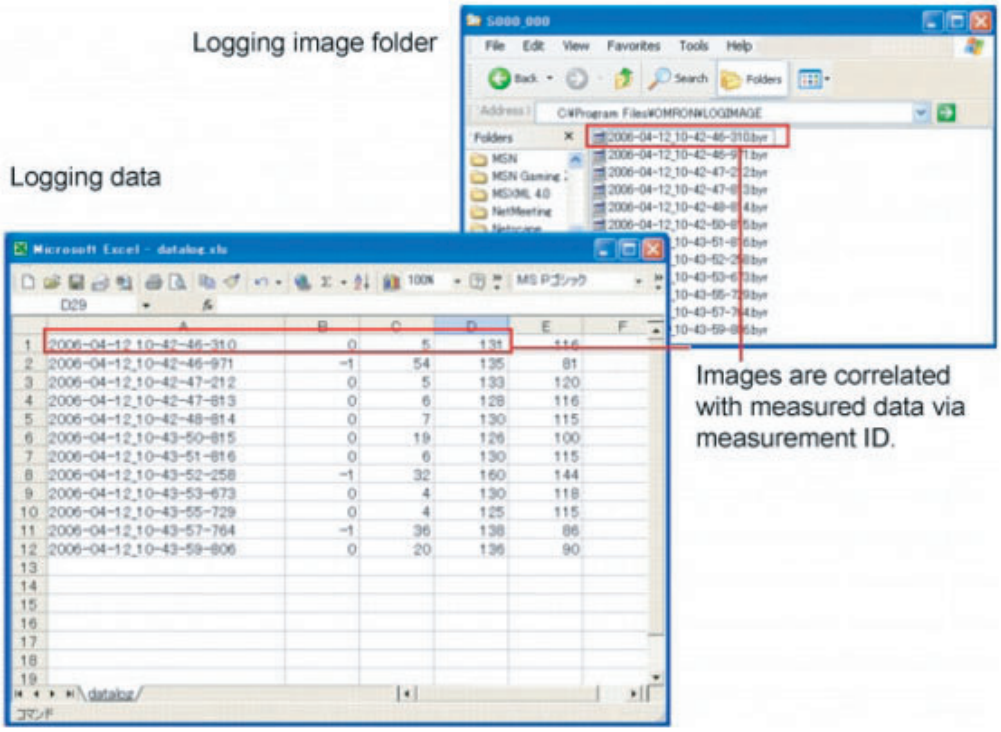
For example, the optimum threshold value can be calculated.





Comparing Logged Data and Logged Images

Compare image and measurement data to confirm correctness and to make analyzing trends for when NG occurs easier.
The measurement data and image data stored with the logging functions are associated through measurement IDs based on the measurement date and time. One image data file contains the Camera image data of all the connected units.
In this way, measurement data can be made to always correspond with image data.
Verify data with the measurement ID.

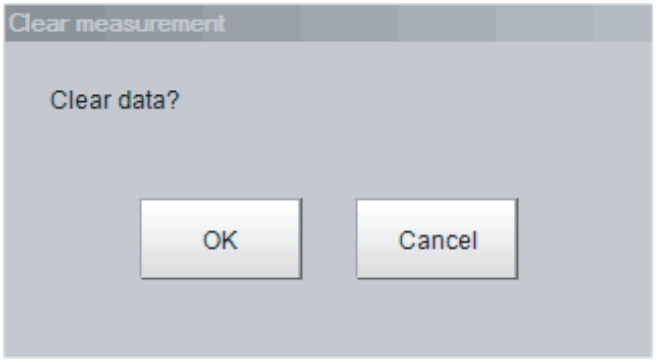


Clearing Measurement Results

Use the following procedure to clear all of the currently displayed scenes.
This function sets the expression which calculates the measurement count, and is convenient when that count is to be reset.

1 **In the adjustment window (layout 0), select [Clear measurement] from the [Function] menu.**
A confirmation dialog box is displayed.

2 **Click [OK].**
The measurement results are deleted.



Clearing Saved Images

Use the following procedure to clear all of the images that are currently logged in the Controller.

1 **In the Main Window (layout 0), select [Clear logging image] from the [Function] menu.**

A confirmation dialog box is displayed.

2 **Click [OK].**

Note

If you want to keep the logged images as files, save the logged images to the external memory by clicking [Function] – [Save to file] – [Logging image] before clearing them.
Reference: ▶ Saving Logged Images in the Controller Memory (RAM) to a RAM Disk or an External Memory Device (p.259)

Capturing Screen Images

The contents displayed in the run window can be captured. Saved images can be loaded into the computer and pasted into documents.

Set the destination for captured images.

Use the following procedure to set the save destination for the image captured with the screen capture function.

1 **In the Main Window, select [System settings] – [Other] – [Screen capture setting] from the [Tool] menu.**

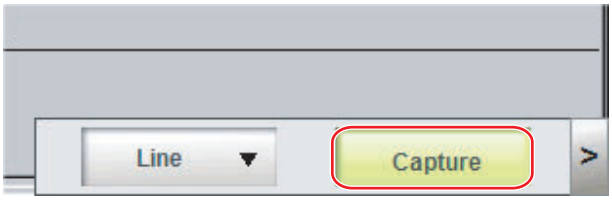
The [Screen capture] settings dialog box is displayed.

2 **Specify the save destination for captured images.**

3 **Click [Apply].**

Saving Captured Images

1 **Click the [Capture] button that is displayed on the Measurement Manager Bar that is located in the bottom right corner.**



IMPORTANT

The capture takes a few seconds and measurement cannot be performed at this time.

Note

- An alternative operation is to select [Screen capture] from the [Function] menu.
- I you use the [Capture] button that is located in the bottom right corner to capture the screen image in Multi-line Random-trigger Mode, the image is always saved to the destination that is set for line 0.

Captured Image Files

This section explains the format and file names for captured images.

Item	Description
File format	The file format is BMP.
File name	The file name is the date and time at which the capture was performed. YYYY-MM-DD_HH-MM-SS-MS.BMP Year (4 digits) -Month- Date_ Hour- Minute- Second- Millisecond Example: The file name for a capture date and time of 3/10/2007, 11:25:30.500: 2007-03-10_11-25-30-500.BMP

Note

The following cannot be captured.

- The dialog box to select a file or a folder
- Confirmation message when LCD is turned OFF

Using the Operation Log

The operation log automatically saves the contents of Controller operations and setting changes in a unique file delimited with semicolons (;). Setting content changes are recorded as a time series, which makes change management possible.

Setting the Operation Log

Set the destination to save the operation log. Also set whether to start recording the operation log during startup.

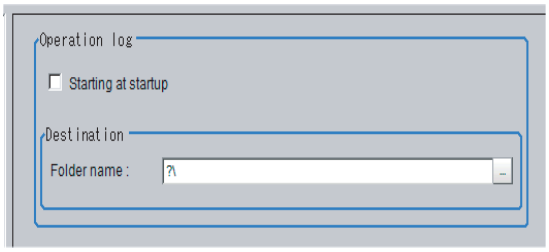
Sample output of an operation log: This log records the date, time, user name, and the operation that was performed.

```
2012/09/10 13:08:41;Administrator;SetSystemData,"OperationLog","enabled","1";0
2012/09/10 13:08:41;Administrator;MeasureStart;0
2012/09/10 13:08:41;Administrator;GetSceneGroupNo;0
. . .
```

1 **In the Main Window, select [System settings] – [Other] – [Operation log setting] from the [Tool] menu.**

The [Operation log settings] dialog box is displayed.

2 **Set the parameters for the operation log.**



Parameter	Description
Starting at startup	Select this check box to record the operation log from startup The setting will be reflected from the next time that the system starts up.
Folder name	Specify the destination folder name. The operation log file name is the date and time at which the operation log was started plus the .log extension. Example: 2012-09-29_12-39-04.log • In Multi-line Random Trigger Mode, set a save destination for each line. The file name is the date and time at which the operation log was started + the line number + the "log" extension. Example: 2012-09-29_12-39-04_Line0.log This setting only takes effect after you save the data and restart the system. • In non-stop adjustment mode, the non-stop adjustment log is saved with the file name *(date and time at which the operation log was started)*_Nonstop.log.

3 **Click [Apply].**

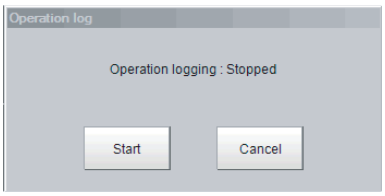
Checking and Changing the Operation Log Status

Use the following procedure to check the current operation log function status

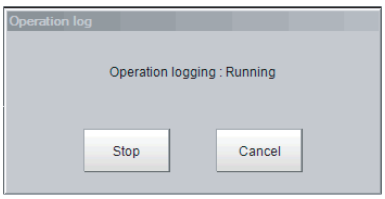
1 In the Main Window, select [Operation log] from the [Function] menu.

The current operation log status is displayed.
To change the current operation log status, click either [Start] or [Stop].
To close the dialog box without changing the current operation log status, click the [Cancel] button.

Dialog Box When Operation Log Is Stopped



Dialog Box While Operation Log Is Being Recorded



Loading an Operation Log File

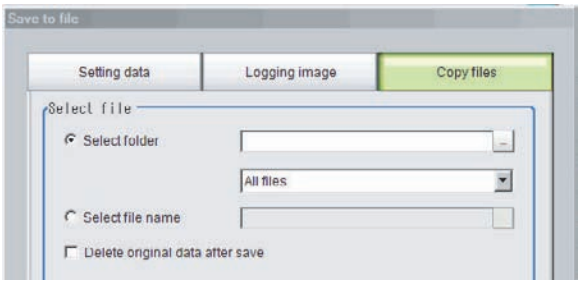
When the log file is set to be saved to the Controller, use the following procedure to load the file.

1 Insert an external memory device into the Controller.

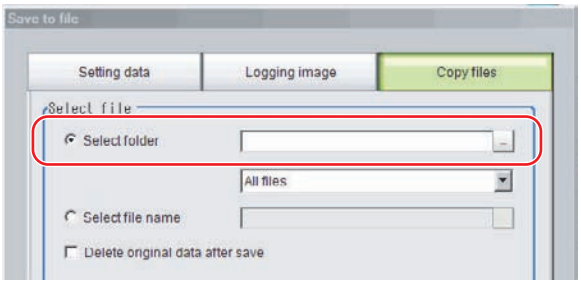
2 In the Main Window, select [Save to file] from the [Function] menu.

The [Save to file] dialog box is displayed.

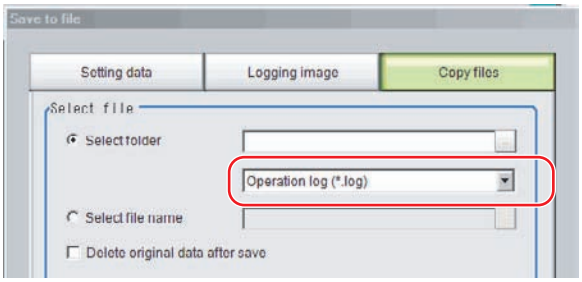
3 Click the [Copy files] tab.



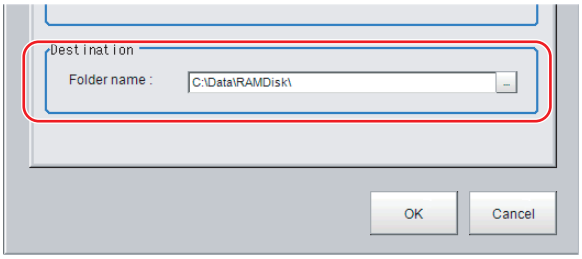
4 Select [Select folder], click [...], and specify the folder.B



5 Select the operation log (*.log).



6 **Specify the destination.**



7 **Click [OK].**

For details on operation logs, refer to ▶ Operation Log Format (p.127)

IMPORTANT

- The contents of operation log files cannot be checked on the Controller. On a computer, open the file with a text editor or change the delimiter characters to semicolons (;) with spreadsheet software.

Operation Log Format

- The operation log is stored in a file indicating the date and time as follows in the directory specified as the save destination.

  YYYY-MM-DD_HH-MM-SS.log (example: 2012-08-20_12-00-00.log)

  - The date and time shown in the file name is the date and time at which the log file was created.
  - The maximum size for a log file is about 100 KB. If a log file is larger than 100 KB, the log is written into a new file from the first record over 100 KB.
  - When the log is stopped, then started again, a new log file is created.
  - Also during operation log execution, if the save destination directory setting is changed, at that point in time, a new log file is created in the new save destination.
  - Use Excel or other spreadsheet software to open the log file. If the characters are corrupted, set the character code to UTF-8 with the BOM.

- Basically, information is saved when the API (application program interface) is executed. The information for one piece of API information is written into the log file as one record.

  The format for each record that is written to the log file is as follows: The four data items, the execution date and time, user name, input information, and output information, are delimited with semicolons. At the end of the record, a line feed (CR+LF) is entered.

  (Execution date and time); *(user name)*; *(input information)*; *(output information)* [LF+CR]

  Examples:
  2012/08/20 12:00:00;user0;SetSystemData,"XX","YY","ZZ";0
  2012/08/20 12:00:01;user0;GetSystemData,"XX","YY";0,"ZZ"

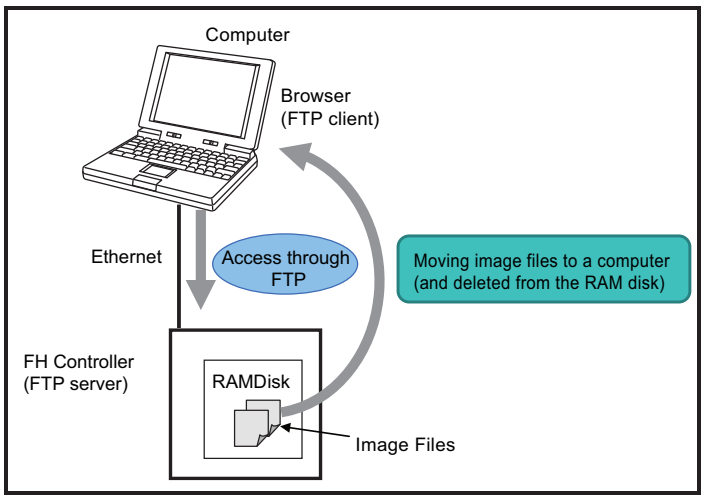
Execution date and time	The date and time at which this API was executed.
Username	The name of user that is currently logged in. If no one is logged in, the character string "no login" is entered.
Input information	The API name is followed by supplemental information (mostly API arguments), delimited with commas.
Output information	The API return value is followed by supplemental information (mostly information gotten by the API), delimited with commas.

For the operation log input information written in the operation log file, refer to ▶ Operation log input information list (p.864).

Saving Data to an External Device

To Save Logged Image Files to a Computer (FTP)

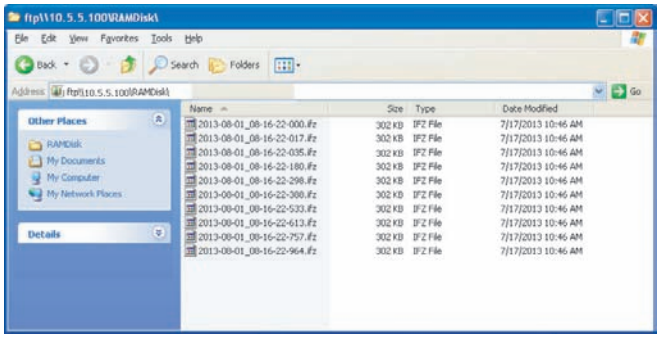
The FTP is useful when you need to save logged image files to a computer.
For example, if you need to move logged images to a computer before the RAM disk or external memory device becomes full, you can use the FTP to access the FH Controller through a browser and directly specify the files in the RAM disk or an external memory device to move.



Operating Procedure

Use the following procedure to access the RAM disk in the FH Controller.

1. **Set up the computer and Controller for communications through Ethernet.**
2. **Start the Explorer on the computer.**
3. **In the address bar, type the IP address and destination and press the Enter Key.**
   Example: [ftp://10.5.5.100/RAMDisk/]
4. **The contents of the RAM disk is displayed as shown below.**



5. **Drag files as you would with normal file operations.**

IMPORTANT

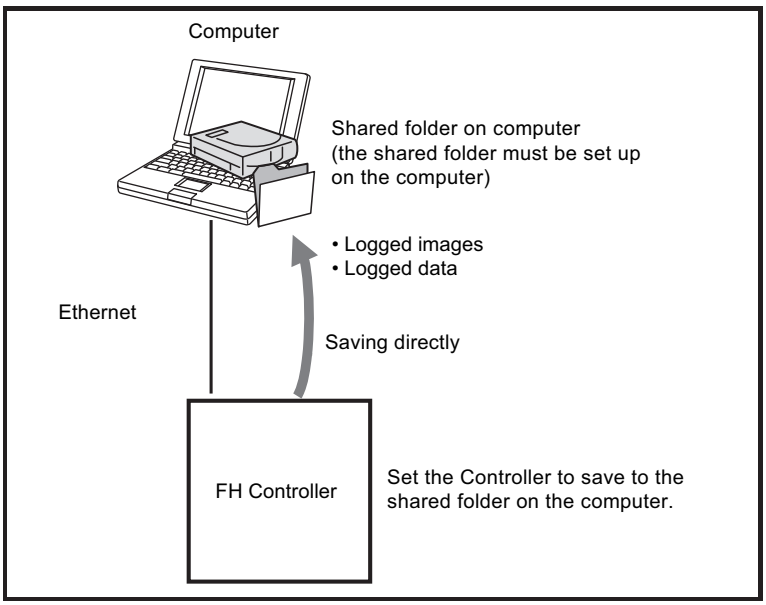
Use only single-byte alphanumeric characters for the file names of files to send and receive with the FTP.

Note

- There are no limitations on when you can perform FTP file transfers.
- Processing of FTP file transfers does not affect the transaction time. However, It does affect the load on the network.

Saving Logged Data Directly to a Shared Folder on a Computer ([Network Drive Function])

The network drive function is useful when you need to log image data or measurement data directly to a computer that is connected by Ethernet.
You can specify the shared folder on a computer that is connected by Ethernet as the destination for logged images or logged data.
Direct reading and writing with the computer eliminates the need for reading and writing through external memory devices. At the same time, it also eliminates the need to stop measurements to insert or remove the external memory device.



Setting Up the Network Drive

Register the shared folder on a network drive with a shared name.

1. **Select [Tool] menu – [System settings] – [Other] – [Network drive setting].**
2. **Select the shared name to be registered, and click [Edit].**
3. **Enter the information to be allocated to the network drive (shared name), such as the shared folder, user name, and password.**
4. **The shared folder on the connected computer will appear in the [Select file/folder] dialog box.**

Specifying a Network Drive as a Logging Destination

To import logging data to a computer, you specify a shared folder on the computer as the logged data destination.

Note

The data saving time depends on the conditions of the network and the destination drive.
Be sure to check the operation thoroughly with the actual devices that will be used.

MEMO

Using Tools

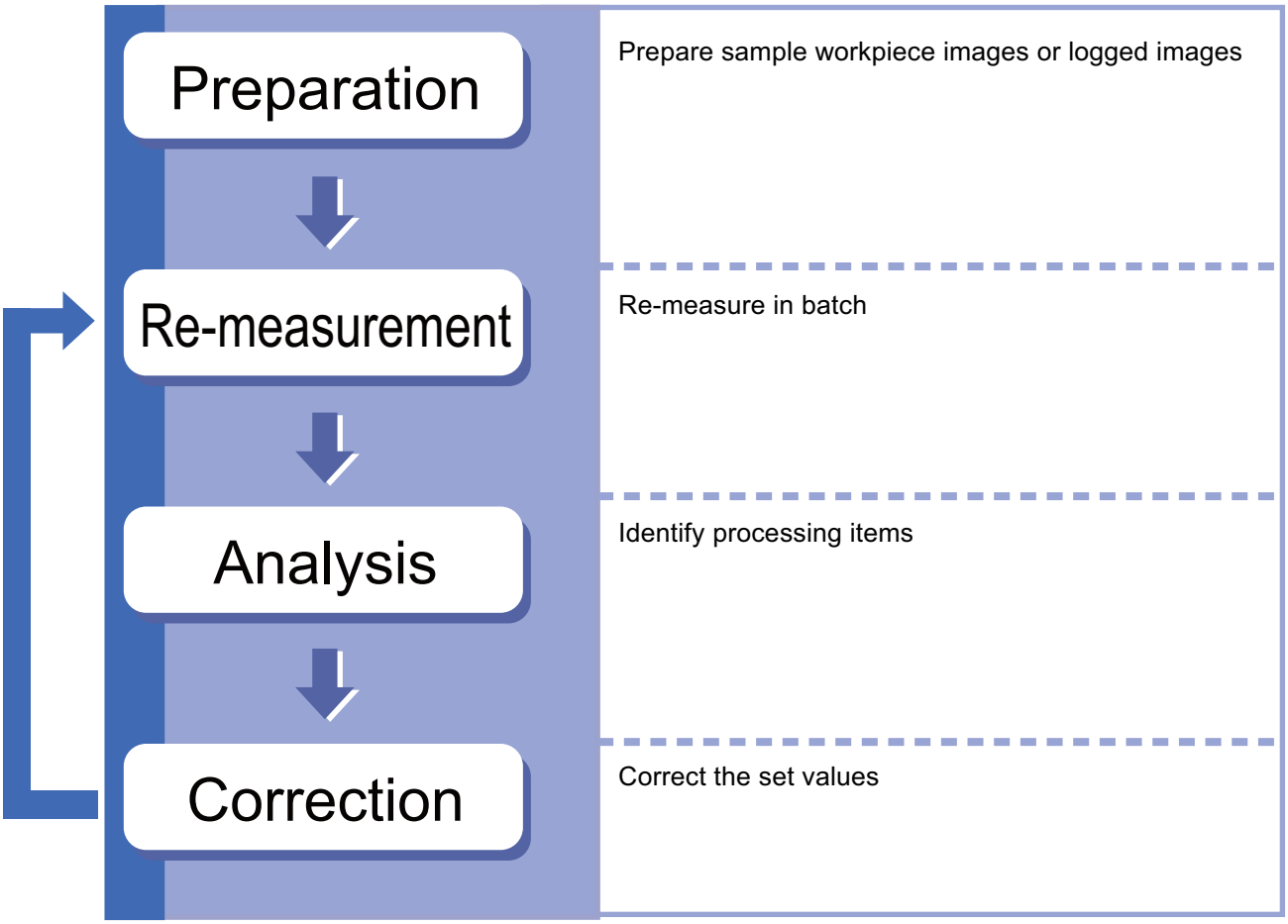
This section describes adjustments during startup and convenient tools for operation.

Using NG Analyzer

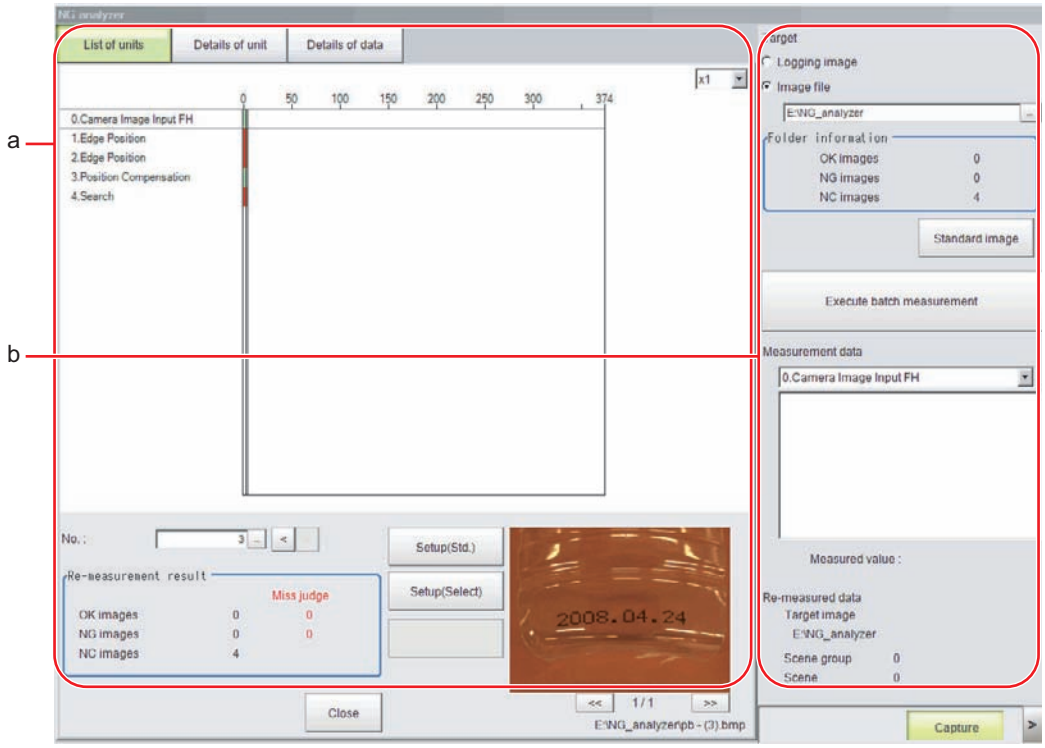
Start the NG analyzer by selecting [Tool] - [NG analyzer] from the Controller menu.
This tool, which analyzes setting flows, is used mainly in 2 ways.

- **Adjustment of measurement set values during start-up**
  Use sample workpiece images to analyze optimal set values for the processing flow.
- **Analysis of NG causes during operation**
  Use logged images to analyze NG causes.

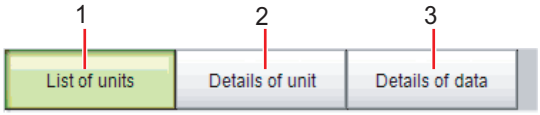
The operation flow is as follows.



Layout of NG Analyzer Window



a. Analysis result display area



1. List of units

A list of units currently set is shown together with analysis results.

2. Details of unit

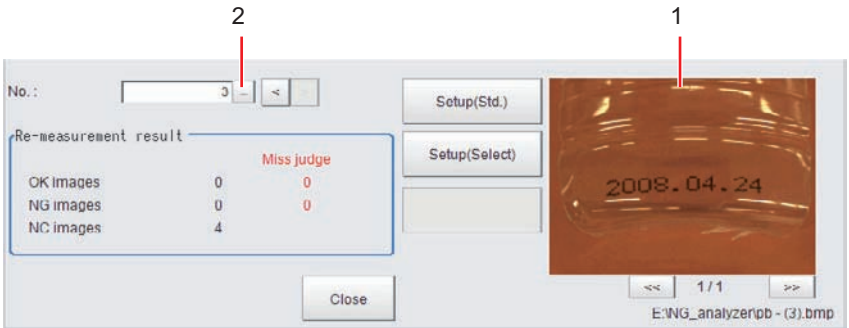
Detailed analysis results of each unit are shown.

3. Details of data

Detailed results of analysis data are shown.

1. Display ratio

Sets magnification to display.



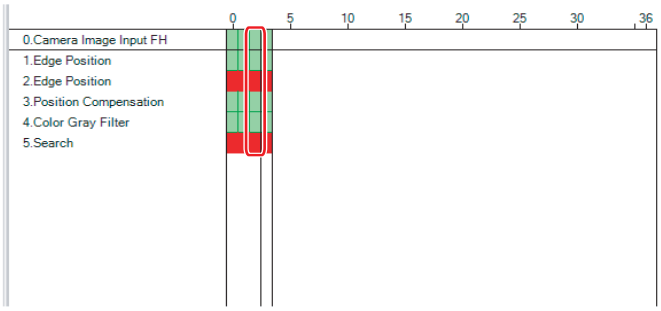
1. Image display area

Displays selected images.

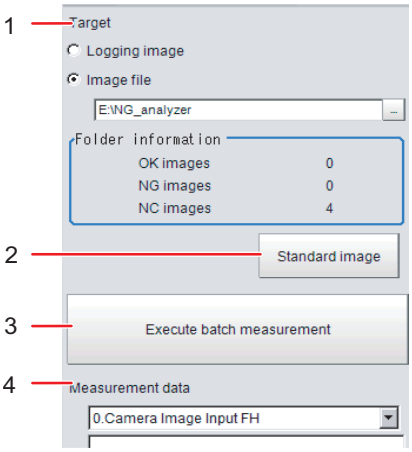
2. Image selection

Selects the image number to be displayed in the image display area.

Images can be displayed by directly clicking the analysis result window.



b. Control area



1. **Target**
   Sets images to be measured.
2. **Standard image**
   Standard image
3. **Execute batch measurement**
   All of the specified target images are measured continuously.
4. **Measurement data**
   Display the desired unit in the list of units and select the unit based on details of unit and details of data.

Using Method of NG Analyzer

IMPORTANT

- Classify sample images beforehand into the OK folder containing images you want to judge OK or NG folder containing images you want to judge NG. (The applicable file types are "*.IFZ", "*.BYR" and "*.BMP".)
- Do not input external commands or STEP signals while the NG analyzer is running (excluding during non-stop adjustment).

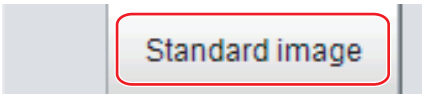
1 On the Main Window, select [Tool] menu - [NG analyzer].

The analyzer window is displayed.

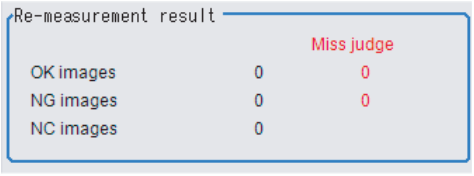
2 Specify the image file.

Specify the upper rank folder containing the OK and NG folders.
Although logged images of the Controller can be set as the target, all logged images of the Controller, if selected, are treated as "Not yet judged."

You can also set a reference image to perform adjustment.



3 Click files in the folder are displayed.

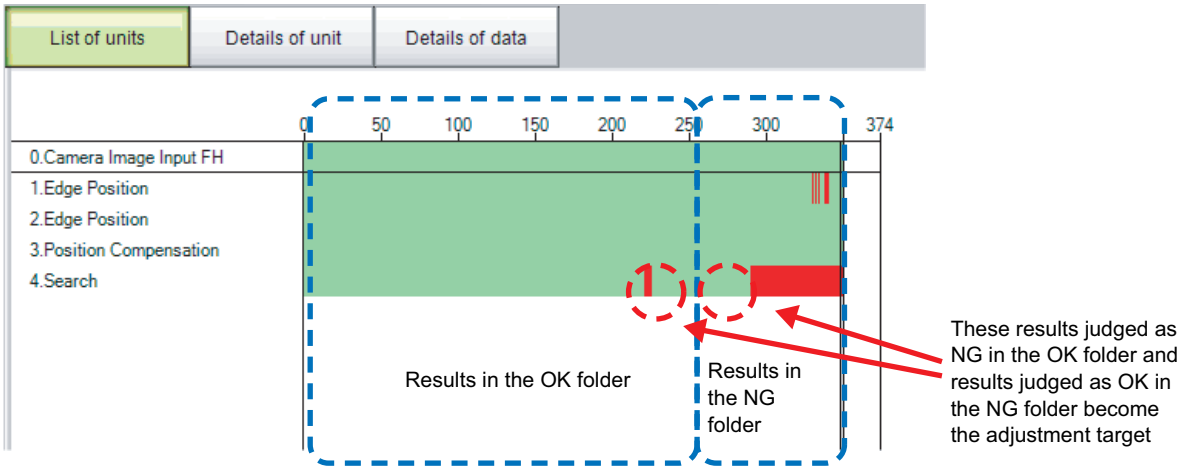


4 Click [Execute batch measurement].

All images in the folder are measured in batch.

5 Measurement results are displayed.

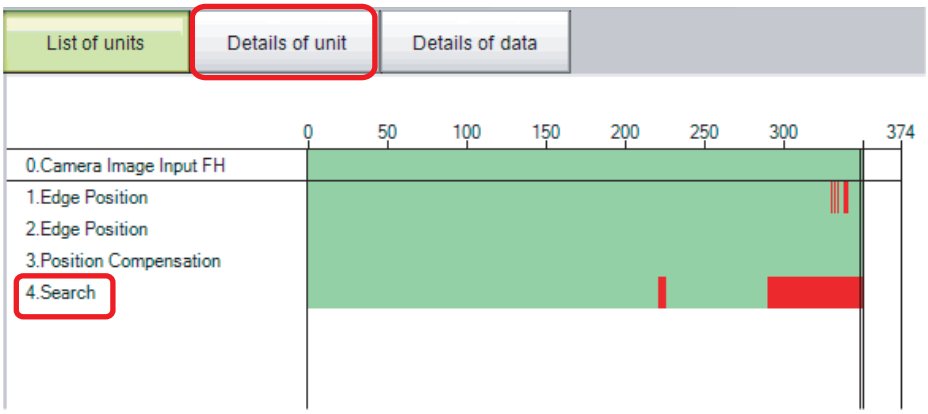
The results in the OK folder are shown first, followed by the results in the NG folder.
Green indicates OK, while red indicates NG.



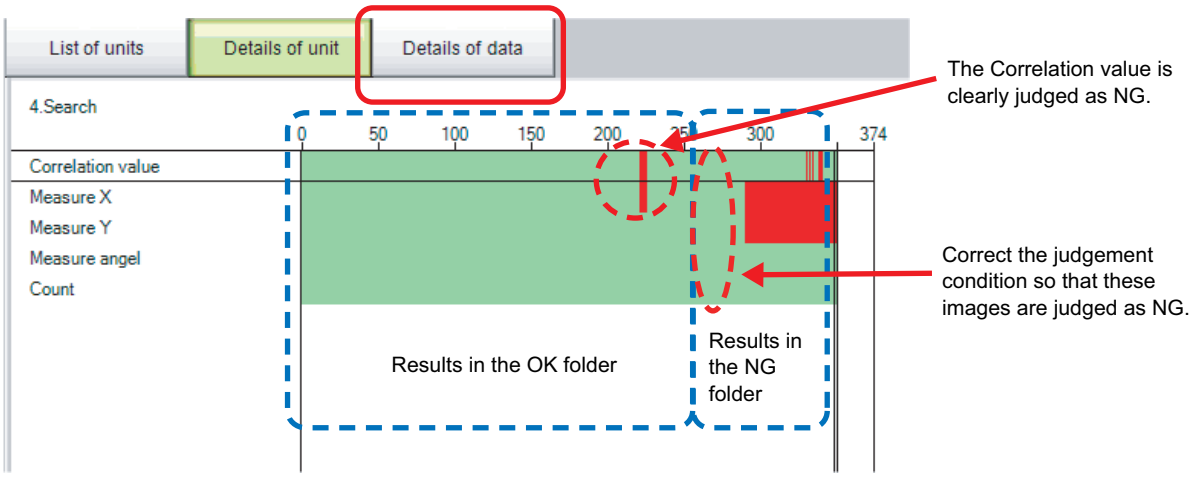
Adjust the set values of each unit until no images are incorrectly judged.

6 Select the processing item to be adjusted, and click [Details of unit].

In the above example, [Search] becomes the adjustment target.

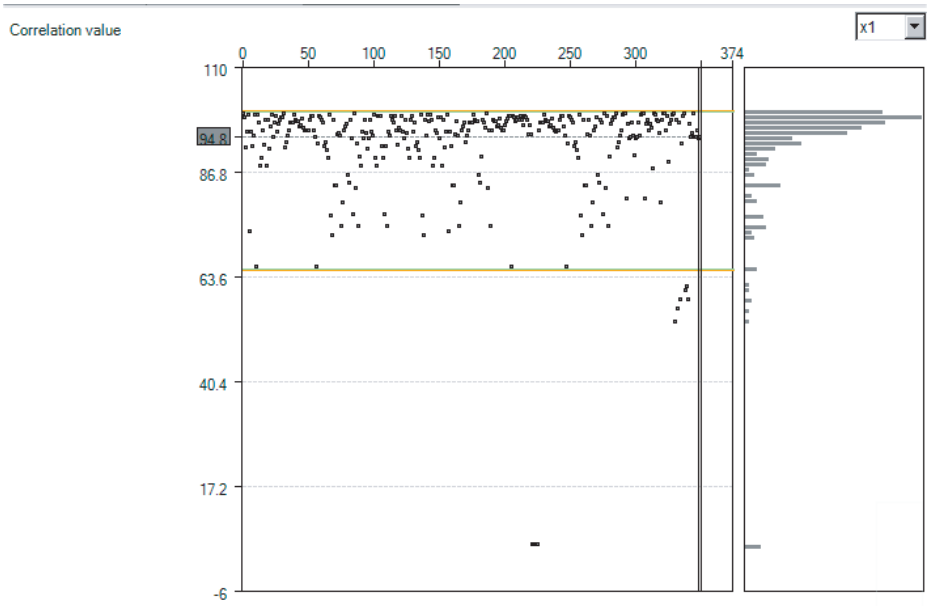


7 The cause of NG is displayed. To check the details of values further, click [Details of data].



8 **Adjust the processing item by referring to the displayed content.**

In the following example, Correlation values are clearly lower on some images.
Based on the revealed cause of incorrect judgement, use the [Set up(Std.)] and [Set up(Select)] buttons to change the set values of the processing unit.



9 **Repeat steps 5 to 8 to correct the set values corresponding to all causes of incorrect judgement.**

10 **Select [Execute batch measurement] to confirm that no images are incorrectly judged.**

If there are still incorrectly judged images, repeat the same procedure until a re-measurement finds no incorrectly judged images.

Note

- OK/NG judgements can be changed by using [Send to OK folder] and [Send to NG folder] buttons. In this case, the changes will not be reflected until re-measurement is performed.

Updating the Reference Position

This section describes the use of a table to set or change the reference position for more than one processing unit that you specify in the measurement flow.
To open the Update Standard Position Tool, select [Update standard position tool] from the [Tool] menu.

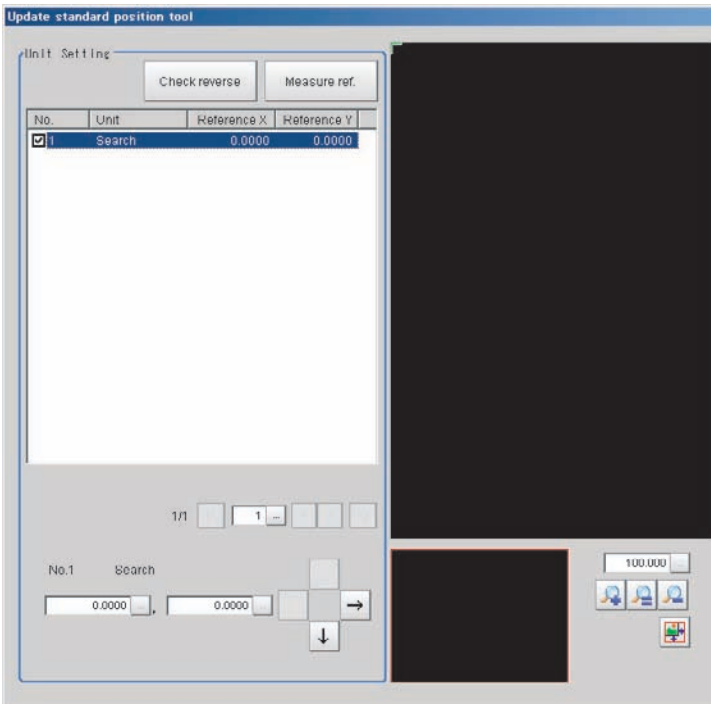
Items That Can Be Updated with the Update Standard Position Tool

You can specify the following processing items.

- Search
- Sensitive Search
- Shape Search II
- Shape Search III
- EC Circle Search
- ECM Search
- Ec Corner
- Ec Cross
- Edge Position
- Scan Edge Position
- Circular Scan Edge Position
- Intersection
- Gravity and Area
- Labeling

Operating Procedure for the Update Standard Position Tool

You can update the reference position for measurement items in the flow without the need to open the settings dialog box for each item.
For example, you can easily update the reference position that is updated frequently in an alignment application.
To update the reference position, display the image to use as the reference in advance.
When you click the [Measure ref.] button, the Controller measures the reference for all units that are selected in the [Unit Setting] list and updates the reference position.
If you click the reference position in the window, or enter the coordinates as a numeric value, the reference positions for the selected units in the [Unit Setting] list are updated with this value.



To Cancel Updating

Click the [Cancel] button. You can return the reference positions to the values that were in effect before you opened this tool.

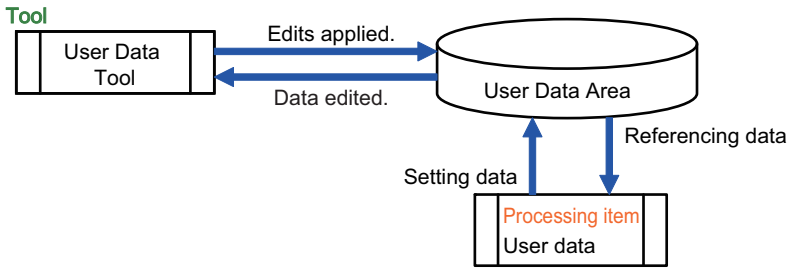
Note

Most of the measurement items automatically update their reference positions when the model or measurement area is changed. If you need to change the reference position after it is updated, place the workpiece in the desired position, and then update the reference positions for all measurement units that you wish to change in the measurement flow.

Using the User Data Tool

This tool is used to share data within the Controller.
The data set as user data is shared between scenes or scene groups. Data sharing across lines is not allowed in Multi-line Random Trigger Mode. The data variables that are created are stored in the user data region on the Controller.

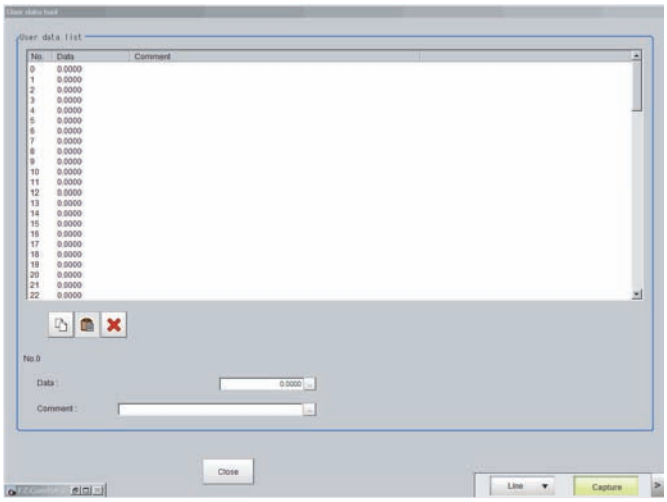
Use the User Data Tool to edit data, specify initial values, and enter data comments.
Use the User Data processing item to set or reference the values in the data that you set with the User Data Tool.
Refer to ▶ *User Data* in the *Vision System FH/FZ5 Series Processing Item Function Reference Manual* (Cat No. Z341).



Setting Procedure for User Data

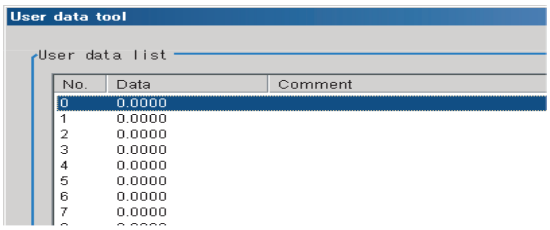
1 **To set user data, select [User data tool] from the [Tool] menu in the Main Window.**

The [User data tool] dialog box is displayed.



2 **Click the data to set.**

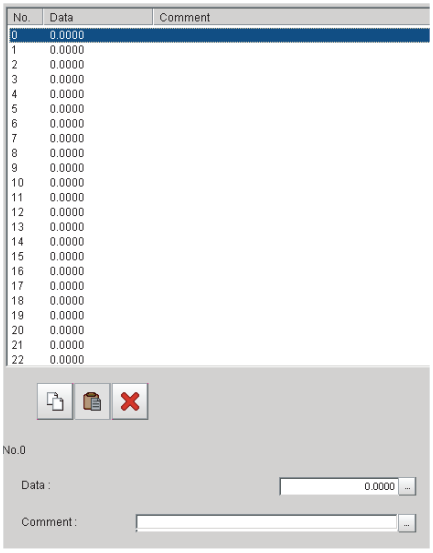
3 **Set the default value for the data.**



Item	Set value [Factory default]	Description
Data	–999999999.9999 to 999999999.9999 [0.0000]	Set the specified user data value.

4 Enter the comment for the data.

The entered comment will be displayed in the [Comment] column of the User Data processing item.



Item	Set value [Factory default]	Description
Comment	64 characters max. [Space]	Set the comment for the specified user data.

5 Repeat steps 2 to 4 to set the user data.

6 To stop the User Data Tool, click [Close].

7 Click [Data save] to save the specified user data to the Controller.

Creating Scene Group Data for More Than 128 Scenes

You can increase the number of scenes in a scene group to more than 128 scenes.
To start the Conversion Scene Group Data Tool, select [Conversion scene group data tool] from the [Tool] menu.

Overview

Normally, there are 128 scenes in a scene group.
For example, if you need to create scenes for each model and there are more than 128 models, you would normally create more than one scene group to manage the scenes.
However, you can use the Conversion Scene Group Data Tool to increase the number of scenes in a scene group to up to 1,024 scenes. A scene group data file (extension *.sgp) is used as the target data.
The ability to manage all of the scenes in one scene group even if there are more than 128 scenes simplifies the PLC control programming and data management.

IMPORTANT

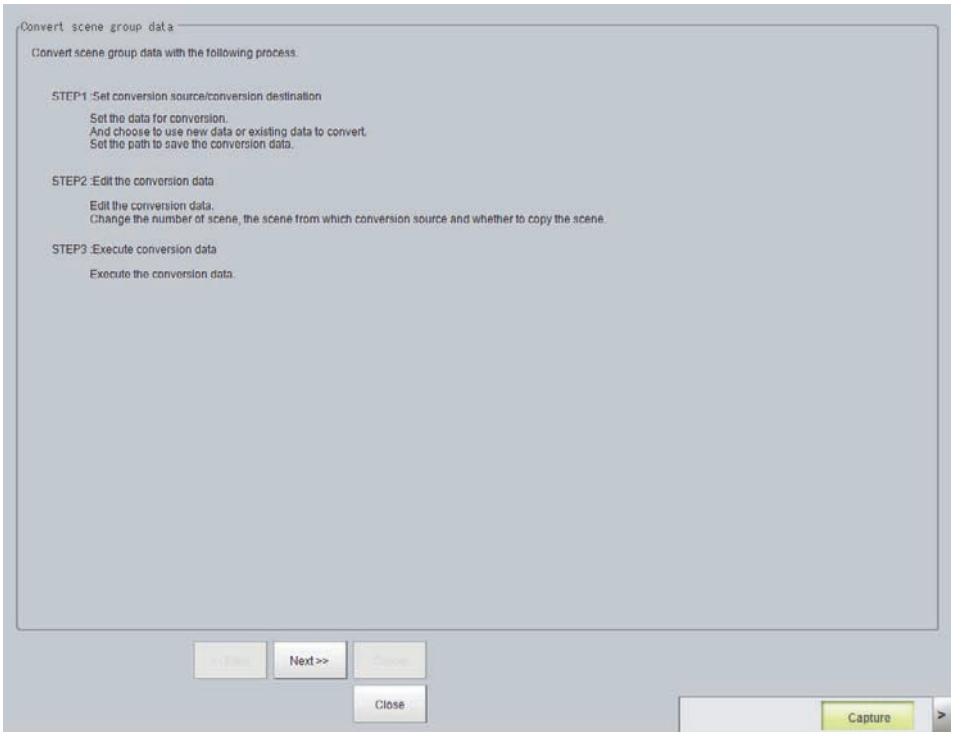
If you increase the data size of a scene group, you may not be able to save the data in the Controller or load the data from a file, depending on the number of scenes and the settings.
If that occurs, change the measurement flow, separate the scene group, or otherwise reduce the data size of the scene group.

Note

For best results, organize the scenes in the scene groups before you use the Conversion Scene Group Data Tool. For example, arrange the scenes in the proper order, delete unnecessary scenes, and assign scene names.
Reference: ▶ Editing Scenes (p.62)

Using the Conversion Scene Group Data Tool

When you start the Conversion Scene Group Data Tool, the following dialog boxes are displayed in order.
You complete the following three dialog boxes as a progressive process that sets, edits, and converts the scene group data.
Step 1: Set the target data to convert.
Step 2: Edit the data.
Step 3: Execute the conversion process.

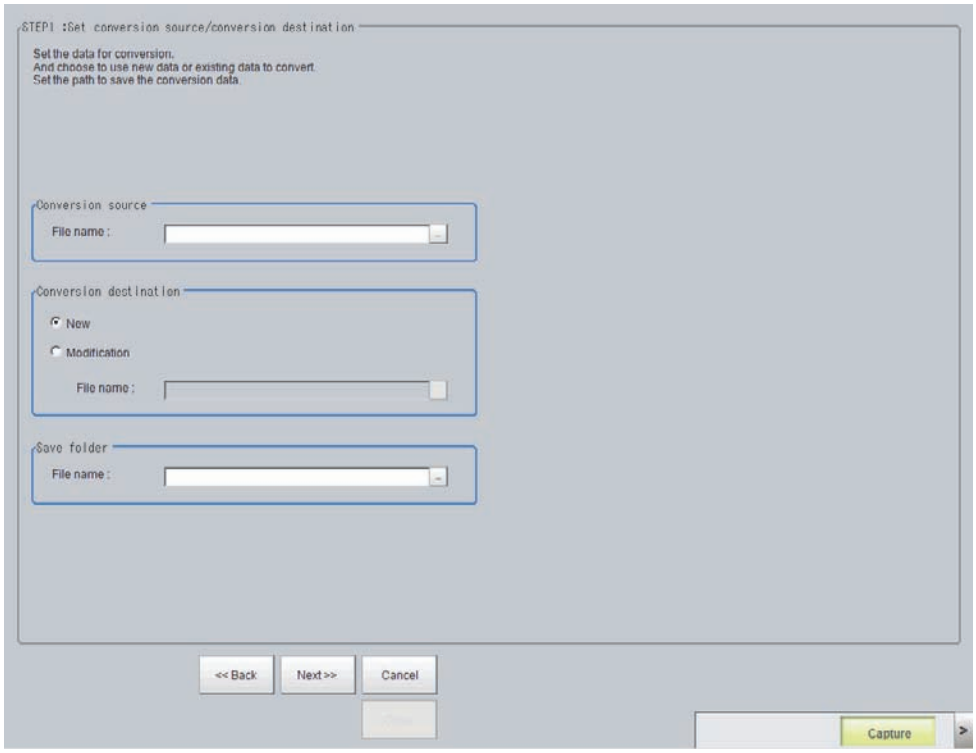


Use the following procedure and the guidance in the dialog boxes to perform the conversion process.

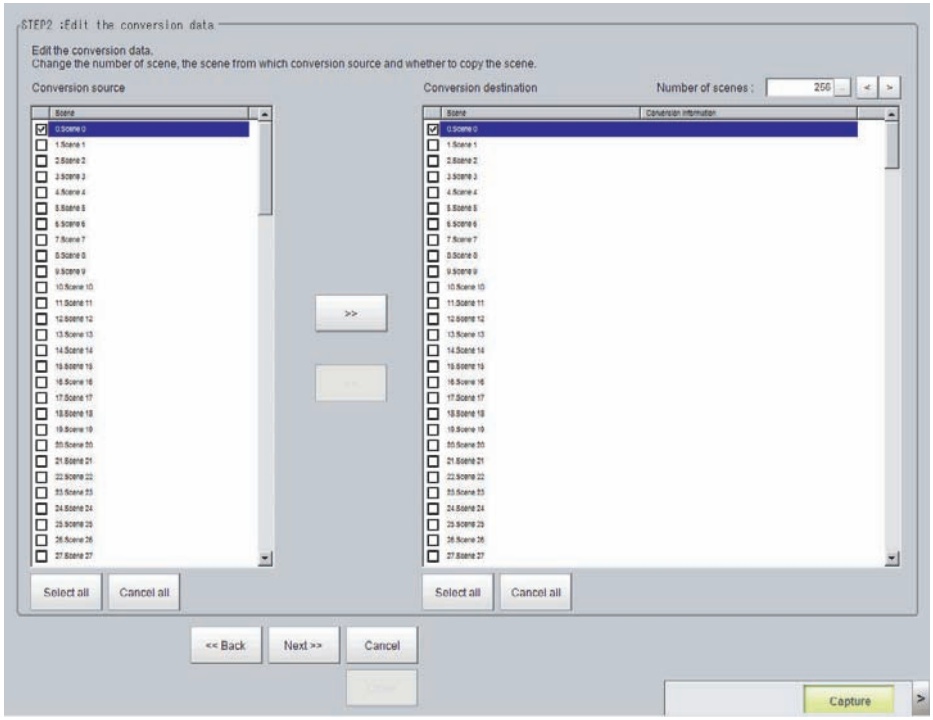
1 **Select [Conversion scene group data tool] from the [Tool] menu.**

The Conversion Scene Group Data Tool is started.

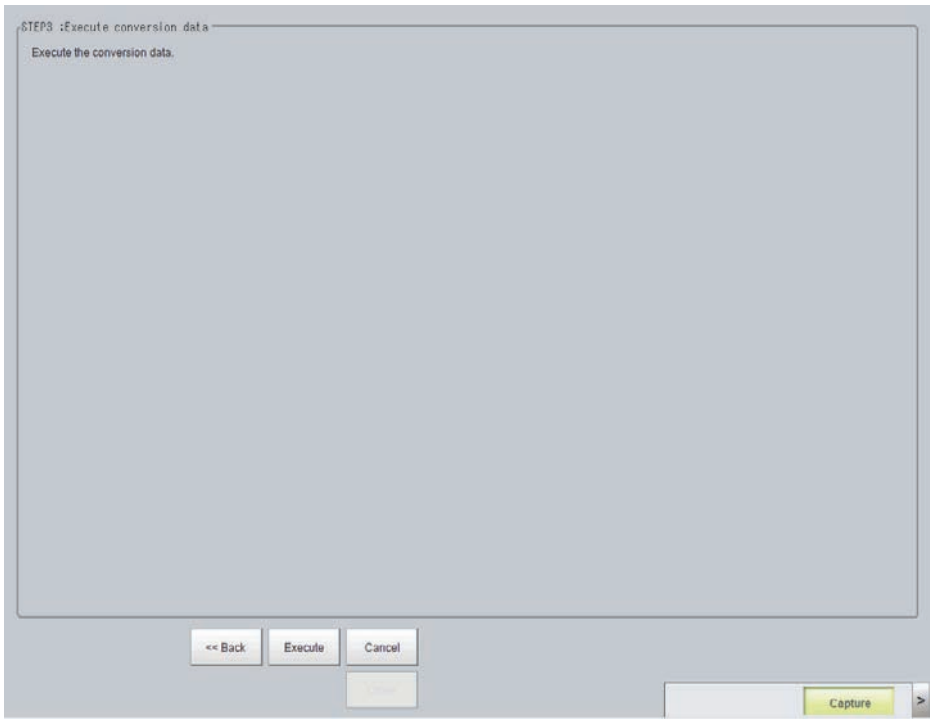
2 **Click the [Load] button. The [STEP 1] dialog box is displayed.**



3 **Specify the target data file to convert, the destination data file, and the destination. Then click the [Next >>] button. The [STEP 2] dialog box is displayed.**



4 **Edit the scene in the converted scene group.**
Click the [>>] button to copy the conversion source to the conversion destination. Click the [<<] button to copy the conversion destination to the conversion source.
When you are done editing, click the [>>] button. The [STEP 3] dialog box is displayed.



5 **Specify the destination for the converted file, and then click the [Execute] button.**
A progress bar is displayed and the conversion is executed.

Outputting a List of Scene Data Set Values

To create a list of the set values for scene data, use the Settings Download and Upload Tools.
With the set value download function, the specified scene data set values can be output to a CSV file. Different set values in multiple scene data can be easily checked by opening the CSV files in an editor. (Only data that can be read and written can be downloaded.) The Setting Values Upload Tool allows you to upload a CSV file with confirmed set values.

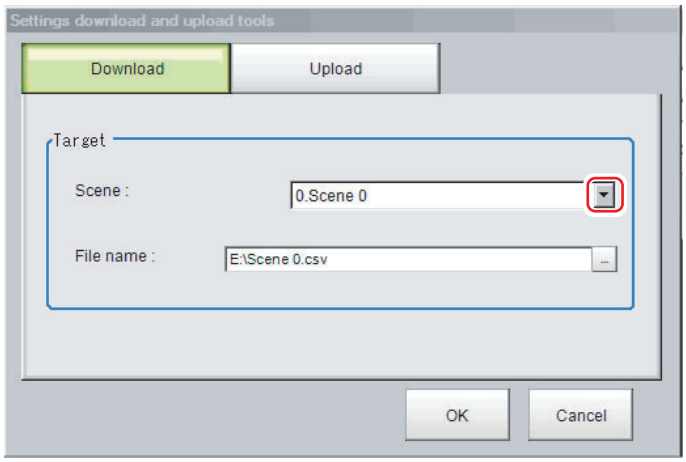
Downloading Set Values

1 **Change to the scene group that contains the scene with the set values to download.**

2 **When saving data to external memory, insert an external memory device into the Sensor Controller.**

3 **In the adjustment window (layout 0), select [Settings download and upload tools] from the [Tool] menu.**

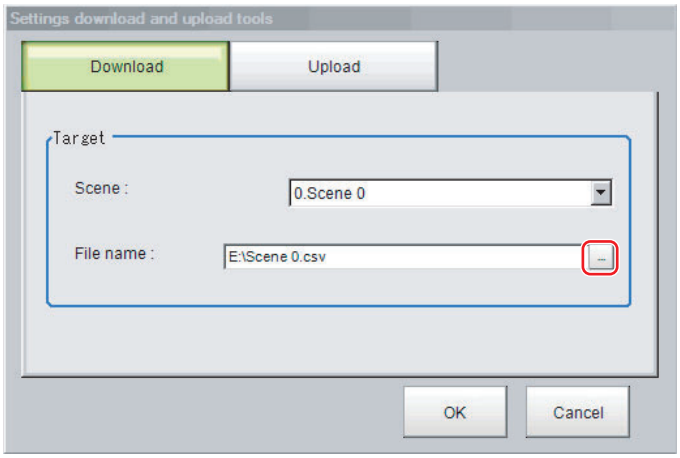
4 **Click [Download] and select the scene with the set values to download.**



5 **Specify the save destination folder and file name, and click [OK].**

6 **Click [OK].**

The data will be saved to the save destination.



About Downloaded CSV Files

The character code of the downloaded CSV file is Unicode (UTF-8).
You can open this file with the Notepad in Windows. When using another editor to open the file, set the character code to Unicode (UTF-8).

The CSV file format is as follows:

SceneTitle,*Scene title name,Author,Note*
#Processing item number in the flow,Processing item identifier,Processing unit title identifier,data title,data

Double-byte characters are enclosed in the double quotation marks (").
Refer to the individual processing items and the external reference tables in the *Vision System FH/FZ5 Series Processing Item Function Reference Manual* (Cat No. Z341) for the parameters for all data. Only data that can be read and written can be downloaded.

Output Example of CSV File:

SceneTitle	Scene 0	
#0	Camera image	Camera Image Input
#1	Search	Search
outputCoordinate	Output coordinates	0
calibration	Calibration	0
overallJudge	Reflect to overall judgement	0
searchMode	Search mode	0
rotation	With/without rotation	0
endAngle	Upper limit of the rotation angle	180
startAngle	Lower limit of the rotation angle	–180
angleSkip	Skipping angle	5
smartMode	Smart mode	1
stability	Stab.(CR)	12
accuracy	Prec.	2
searchSpeed	Stab.(PT)	3
referencePosX	Reference X	0
referencePosY	Reference Y	0
detectionPosX	Detection point X	0
detectionPosY	Detection point Y	0
subPixel	Sub-pixel	0
candidateLevel	Candidate Point Level	70
upperX	Upper limit of measure X	99999.9999
lowerX	Lower limit of measure X	–99999.9999
upperY	Upper limit of measure Y	99999.9999
lowerY	Lower limit of measure Y	–99999.9999
upperAngle	Upper limit of the angle	180
lowerAngle	Lower limit of the angle	–180
upperCorrelation	Upper limit of corr.	100
lowerCorrelation	Lower limit of corr.	60
savemdlimg	Save registered model	0
thersDetail	Candidate Point Level	75
sort	Sort condition	1
searchNo	Search No.	0
upperCount	Upper limit of count judgement	32
lowerCount	Lower limit of count judgement	0
isMulti	Multiple output	0

Uploading Set Values

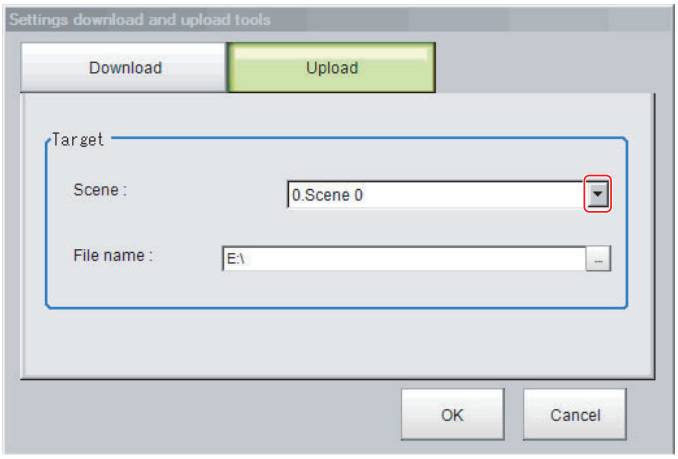
Use the following procedure to upload the CSV file to which you downloaded the set values.

1 **Change to the scene group that contains the scene with the set values to be uploaded.**

2 **When reading from external memory, insert an external memory device into the Sensor Controller.**

3 **In the Main Window, select [Settings download and upload tools] from the [Tool] menu.**

4 **Click [Upload] and select the scene with the set values to be uploaded.**



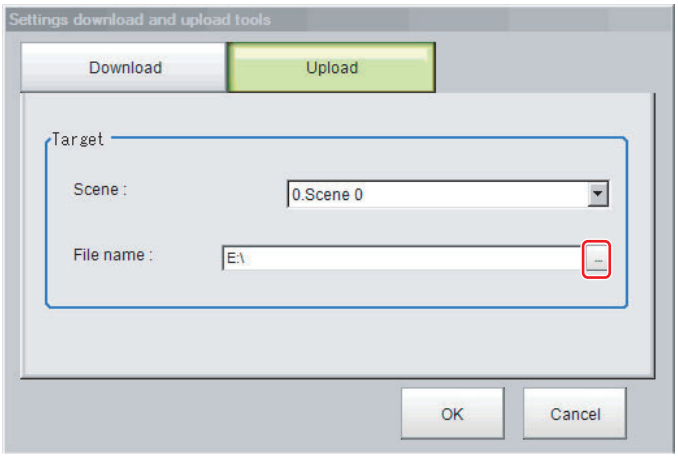
IMPORTANT

To upload the file, set the character code to Unicode (UTF-8) with the Windows Notepad or another application, and save the file in CSV format.

5 **Specify the folder and the name of the file to be loaded, and click [OK].**

6 **Click [OK].**

The data is uploaded.



IMPORTANT

- If data with fewer units than the scene data to be loaded is uploaded, the parts that do not exist in the CSV file will not change.
  Example:

CSV File
1. Camera Image Input
2. Search
3. Defect

Upload

Flow before Overwrite
1. Camera Image Input
2. Search
3. Camera Image Input
4. Search
5. Calculation

Modified Flow
1. Camera Image Input
2. Search
3. Defect
4. Search
5. Calculation

In this example, the processing item for unit 3 will be changed from Camera Image Input to Defects. The parts that do not exist in the CSV file (4. Search processing item and 5. Calculation processing item in the flow) will not change.

Saving Image Files to a RAM Disk or External Memory Device

This section describes how to save logged images and image files that are saved in the Controller to a RAM disk or external memory device in the Controller. The storage format (bitmap or Jpeg) can be specified when the images are saved.

IMPORTANT

During saving, do not restart, turn OFF the power supply, or remove the external memory device. The data may be corrupted.

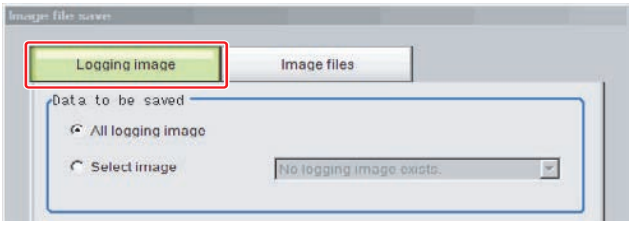
Saving Logging Images

1 **When saving data to external memory, insert an external memory device into the Sensor Controller.**

2 **In the Main Window, select [Image file save] from the [Tool] menu.**

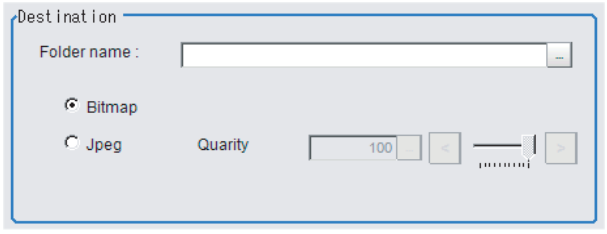
The [Image file save] dialog box is displayed.

3 **Click [Logging image] and select the logged image to save.**



Parameter	Set value [Factory default]	Description
Data to be saved	[All logging image]	All logged images are saved.
	Select image	The selected logged images are saved. Click [▼] to select the image to save. When [Latest measurement] - [logging image] is selected, the save file name will be LoggingImage000.ifz.

4 **Specify the destination folder name and file type.**



Parameter	Set value [Factory default]	Description
Folder name	---	Specify the destination folder name.
Format	• [Bitmap] • Jpeg	Select the image format in which to save the image.
Quality	0 to 100 [100]	Specify the quality of the Jpeg image to save

5 **Click [OK].**

The image is logged in the selected destination. If a file with the same name already exists in the destination folder, the newly saved file overwrites the old one.

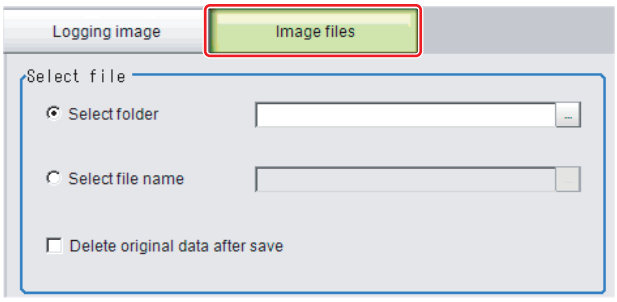
Saving Image Files

1 **When saving data to external memory, insert an external memory device into the Sensor Controller.**

2 **In the Main Window, select [Image file save] from the [Tool] menu.**

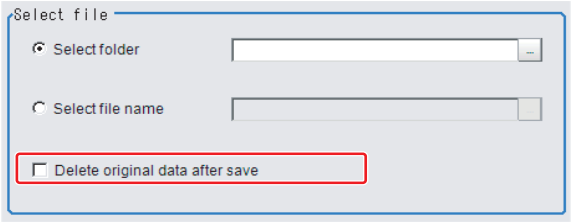
The [Image file save] dialog box is displayed.

3 **Click the [Image files] tab to select the image files to save.**

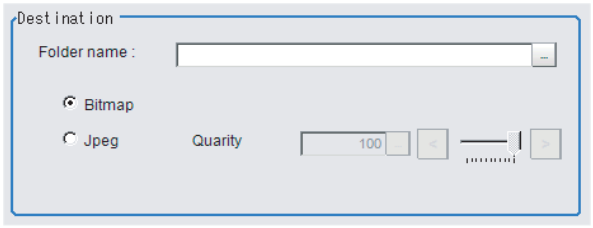


Parameter	Set value [Factory default]	Description
Select file	[Select folder]	Saves multiple files in a folder. Click the [...] button and specify the source folder to copy or move.
	Select file name	Saves the selected image file Click the [...] button and specify the file name.

4 **To delete the source file after saving a copy to external memory, select [Delete original data after save].**



5 **Specify the destination folder name and file type.**



Parameter	Set value [Factory default]	Description
Folder name	---	Specify the destination folder name.
Format	• [Bitmap] • Jpeg	Select the image format in which to save the image.
Quality	0 to 100 [100]	Specify the quality of the Jpeg image to save

6 **Click [OK].**

The image is logged in the selected destination. If a file with the same name already exists in the destination folder, the newly saved file overwrites the old one.

Note

The save file name is the original file name and the extension.

Using the Registered Image Manager

You can save images used for model registration and reference registration as registration images and can reference them later and use them for re-registration and adjustment of reference positions.
You can register the [Latest logging image], [Logging image], and [Image file].

IMPORTANT

- USB memory is required to use this function.
- The registered image is saved in the USB memory.
- Do not change the configuration of files in the USB memory. Changing it would make it impossible to read the files.

Registering Images

This section describes how to register, load, or delete an image.

Registering Images

Use the following procedure to register images. You can register up to 1,000 images.

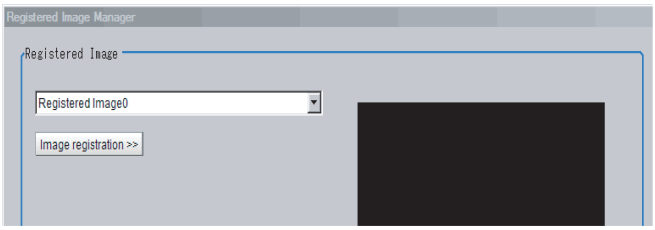
1 **Insert a USB memory device into the Sensor Controller.**

2 **In the Main Window, select [Registered Image Manager] from the [Tool] menu.**

The [Registered Image Manager] dialog box is displayed.

3 **Select an index number for the image to register.**

The number can be selected from 0 to 999.



Note

- If an image already exists, it will be displayed.
- If a number that has been assigned to an existing image is selected for the image to register, the existing image will be replaced by the new image.

4 **Click the [Image registration>>] button.**

5 **Select the image to register.**

Parameter	Set value	Description
Image registration	Last logging image	Registers the most recent image used for measurements.
	Logging image Registration	Registers a logged image that was saved in the main memory. Click [▼] to select the image to register.
	Image file	Registers an existing image file.

6 **Click the [Registration] button.**

In step 3, if you select a registered image number that is already registered, an overwrite confirmation dialog box is displayed. If you want to replace the existing image with a new image, click [Yes].

7 **Click the [Image registration<<] button.**

Deleting Images

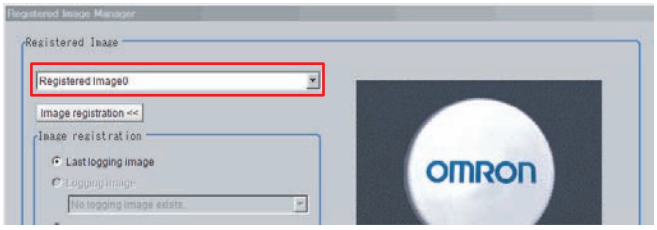
Use the following procedure to delete registered images.

1 **In the Main Window, select [Registered Image Manager] from the [Tool] menu.**

The [Registered Image Manager] dialog box is displayed.

2 **Click the [Image registration>>] button.**

3 **Select the registered image number of the image to delete.**



Note

When more than one image is included in the registration images, click the [<<] or [>>] buttons under the Preview Area as necessary to move forward or back to the desired image.

4 **Click the [Delete] button.**

A deletion confirmation dialog box is displayed.

5 **Click [Yes].**

The selected image is deleted.

6 **Click the [Image registration<<] button.**

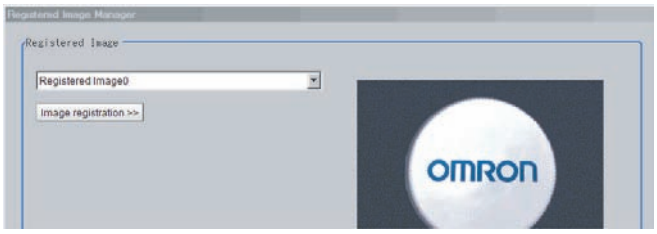
Loading an Image

A registered image can be loaded as a measurement image.

1 **In the Main Window, select [Registered Image Manager] from the [Tool] menu.**

The [Registered Image Manager] dialog box is displayed.

2 **Select the registered image number of the image to load.**



Note

When more than one image is included in the registration images, click the [<<] or [>>] buttons under the Preview Area as necessary to move forward or back to the desired image.

3 **Click the [Read] button.**

The selected image is loaded as a measurement image.

The [Registered Image Manager] window is closed and the display returns to the Main Window.

Using Accounts

You can restrict access to the Sensor Controller for specific users and user groups. Because you can set a password for each user and can enable or disable operations for each user group, you can flexibly manage users to match the way they use the system, for example system administrators, onsite managers, and system operators.
It is also possible to link to the operation log and get operation records for the currently logged-in user.

Setting Accounts (Account List)

This section explains how to add, edit, and delete user accounts.

IMPORTANT

In order to add, edit, or delete a user account, you must be logged in with a user account belonging to group UG0. The default user name and password are both “Administrator”.

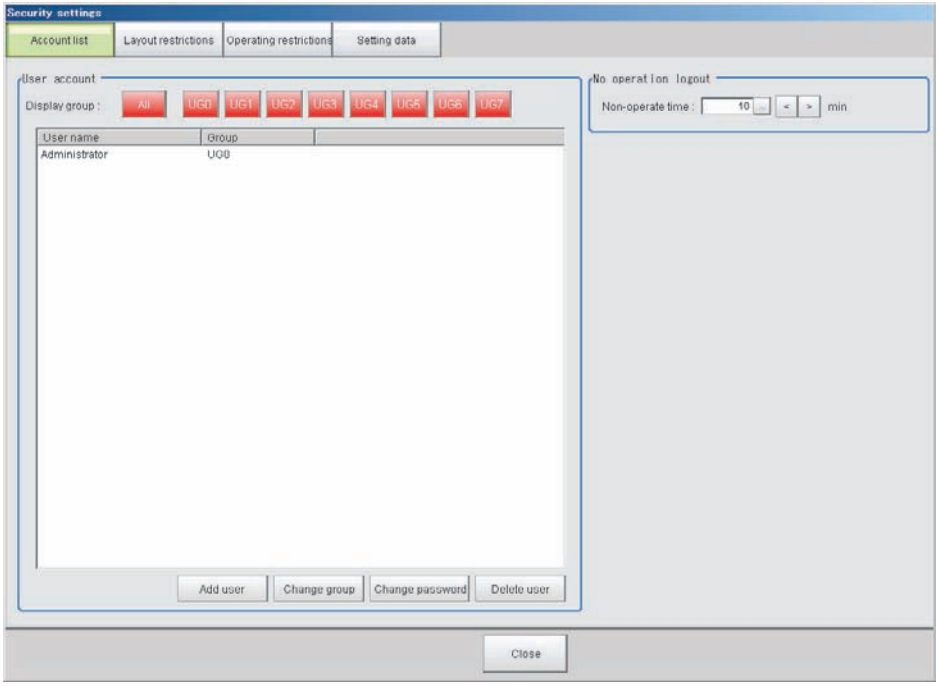
Adding Accounts

Use the following procedure to add a user account.

1 **In the Main Window, select [Security settings] from the [Tool] menu.**

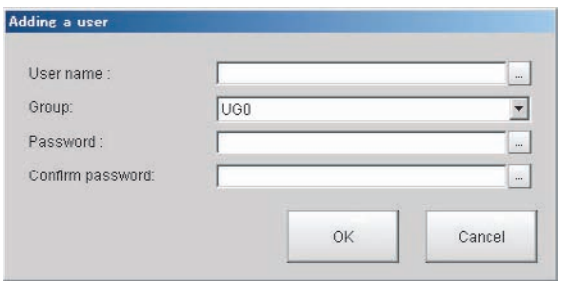
The [Security settings] dialog box is displayed.

2 **Click [Account list] in the Item Tab Area.**



3 **Click the [Add user] button.**

4 **Set the user information.**



Parameter	Description
User name	Enter a user name of 1 to 20 single-byte alphanumeric characters. The names are case sensitive.
User group ID	Select the user group (UG0 to UG7) to which the user being added will belong. If you select [UG0], the user can use all functions. Reference: ▶ Setting User Group Operation Restrictions (p.153)
Password	Enter the login password for the user being added. The password for the new user must have one or more characters. The password for a transferred user may also be 0 characters.
Confirm password	Enter the login password again.

5 **Click [OK].**

The user account is added and displayed in the user account list.

Editing an Account

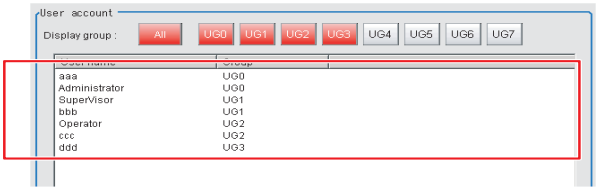
Use the following procedure to change the user group or password for the user account.

1 **In the Main Window, select [Security settings] from the [Tool] menu.**

The [Security settings] dialog box is displayed.

2 **Click [Account list] in the Item Tab Area.**

3 **Click the user to change the user group or password for.**



4 **Click the [Change group] button or the [Change password] button.**

5 **Change the user group or password.**

6 **Click [OK].**

The user information is changed.

Deleting an Account

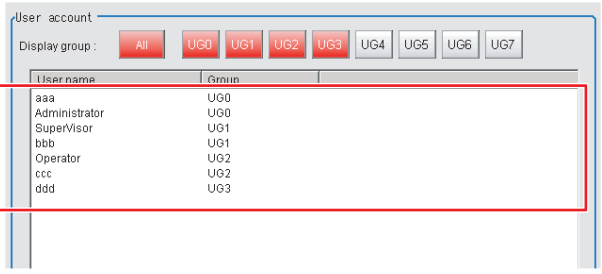
Use the following procedure to add a user account.

1 **In the Main Window, select [Security settings] from the [Tool] menu.**

The [Security settings] dialog box is displayed.

2 **Click [Account list] in the Item Tab Area.**

3 **Click the user to delete.**



4 **Click the [Delete user] button.**

A confirmation message is displayed.

5 **Click [Yes].**

The user account is deleted.

Setting Automatic Logout

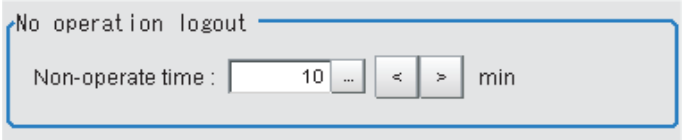
Use the following procedure to set the length of time before an account is automatically logged out when no operation is performed.

1 **In the Main Window, select [Security settings] from the [Tool] menu.**

The [Security settings] dialog box is displayed.

2 **Click [Account list] in the Item Tab Area.**

3 **In the [No operation logout] area, set the non-operation time.**



Parameter	Set value [Factory default]	Description
Non-operate time	1 to 999 [10]	Set the length of time before an account is automatically logged out when no operation is performed. Set the time in minutes.

Setting Layout Restrictions

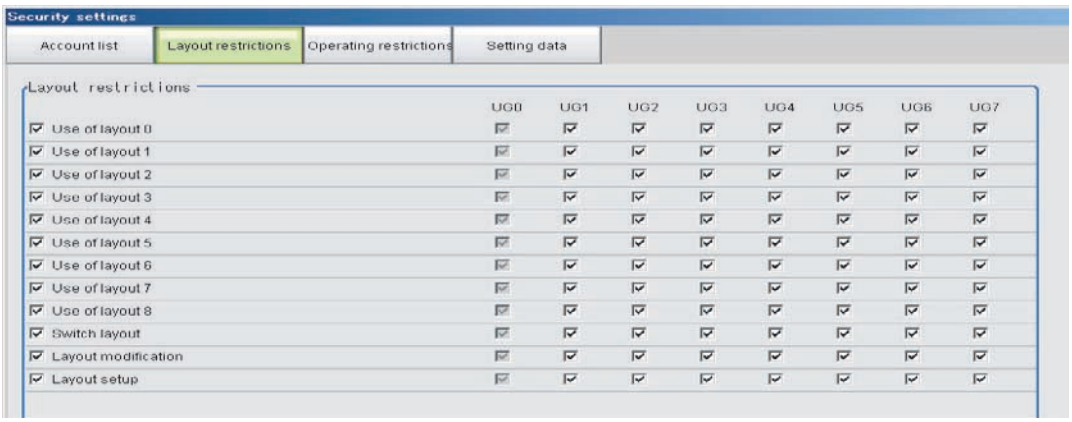
Set the right to change the layout for each user group.
You can restrict the functions that can be used by each user group.
The layouts that can be restricted are as follows:

Security setting items
Use of layout 0
Use of layout 1
Use of layout 2
Use of layout 3
Use of layout 4
Use of layout 5
Use of layout 6
Use of layout 7
Use of layout 8
Switch layout
Layout modification
Layout setup

1 **In the Main Window, select [Security settings] from the [Tool] menu.**

The [Security settings] dialog box is displayed.

2 **Click the [Layout restrictions] tab in the Item Tab Area.**



3 **The layout restriction settings are changed.**

4 **Click the [Close] button.**

Setting User Group Operation Restrictions

You can restrict the functions that can be used by each user group.
You can set restrictions for the following items.

Security setting item	Security setting item
End	Tool button setting
Measure	Image container window setting
Scene switch	Flow window setting
Scene maintenance	Image window setting
Edit flow	Text window setting
Clear measurement	Error window setting
Clear logging image	Output
Screen capture	Continuous meas.
Save last logging image	Select image
Data save	Auto Re-meas.
Save to file	The Judgement result monitor
Load from file	Re-meas.
System initialization	LCD off
System restart	User data tool
Transfer data	Setting download and upload tools
Nonstop data transfer	Image file save
Start/stop of operation log	Flow Viewer
System information	Calibration support tool
NG analyzer	Update Standard position tool
System settings	Conversion scene group data tool
Registered image management	Custom dialog tool
Customize I/O command	Custom dialog
Unit setting mode starting	Scene Group Saving Destination Settings
Nonstop adjustment mode starting	Scene Control Macro Tool
Image window layout setting	Master scene update
Information window setting	Master scene edit

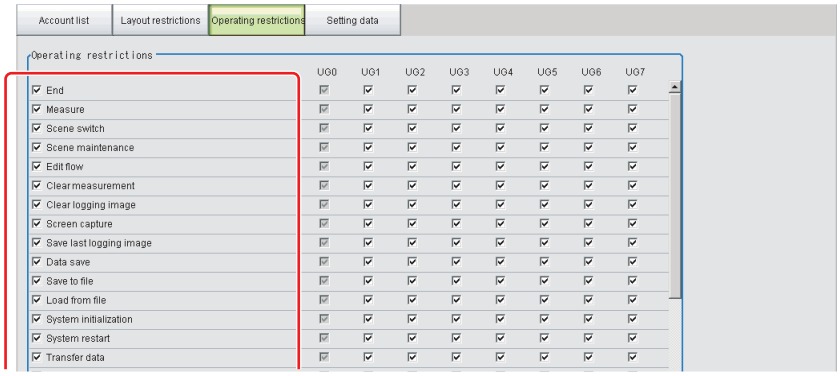
1 **In the Main Window, select [Security settings] from the [Tool] menu.**

The [Security settings] dialog box is displayed.

2 **Click the [Operating restrictions] tab in the Item Tab Area.**

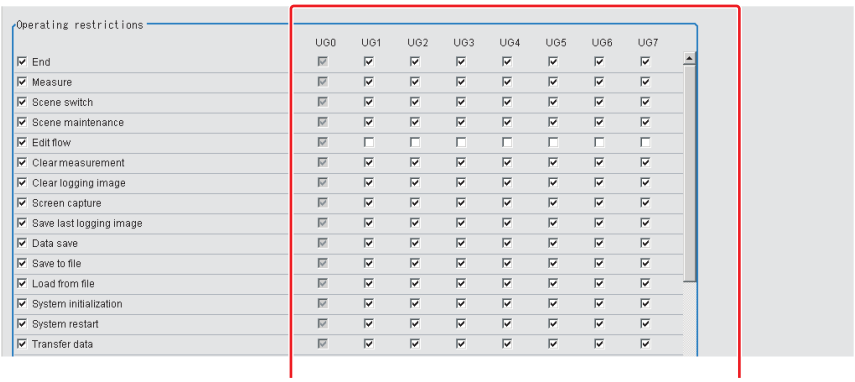
3 **Select the operations to restrict.**

Operations that are not selected are not restricted.



4 **Select the operations to be permitted by user groups (UG1 to UG7) on the right side**

Operations that are not selected cannot be used by users belonging to that group.



5 **Click the [Close] button.**

Operation restrictions are set for the user groups.

Saving/Loading/Deleting the Security Settings

You can save user account and user group settings (security settings). You can load or delete saved settings. Account setting data is not included in BKD files.

Saving Security Settings

You can save user account and user group settings to a RAM disk or an external memory device.

IMPORTANT

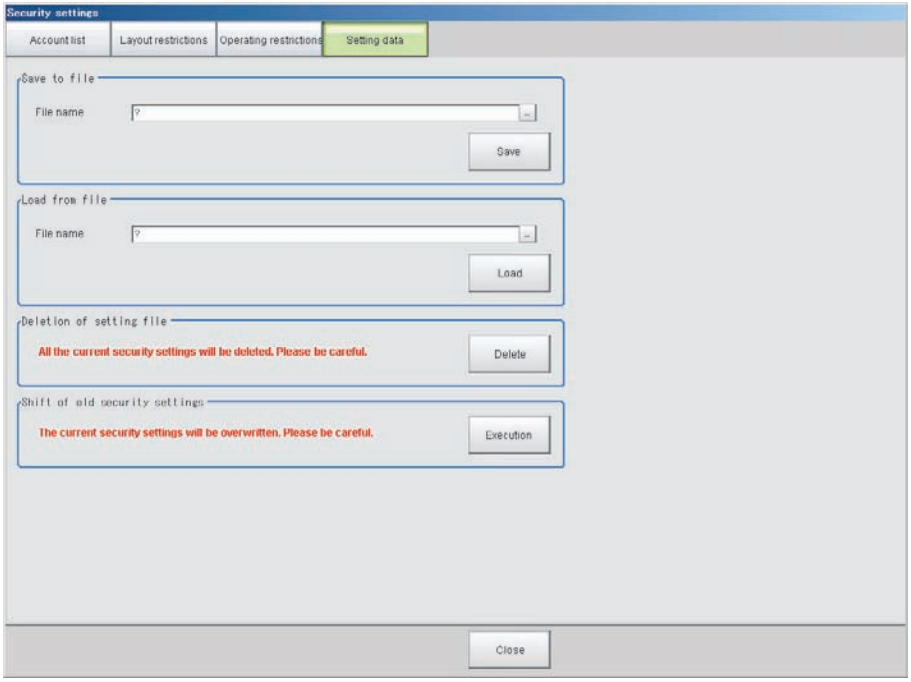
- During data transfer to external memory, do not remove the external memory device until the transfer is completed. Data and the external memory may be corrupted.
- Timing for Saving Data in the Controller
  Data is saved to the Controller only when the [Security setting] dialog box is closed with the [Close] button.
  When the login window is ended by selecting [Cancel], data is not saved to the Controller. The set contents or loaded contents are cleared by when the Controller is restarted.
- Remote Operation Security
  The security settings are synchronized, but for users who are logged in, the security settings are managed separately on the local side and remote side. Therefore, even if users are logged in on the local side, they must log in again on the remote side.

1 **When saving data to external memory, insert an external memory device into the Sensor Controller.**

2 **In the Main Window, select [Security settings] from the [Tool] menu.**

The [Security settings] dialog box is displayed.

3 **Click the [Setting data] tab in the Item Tab Area.**



4 **Specify the destination folder and file name in the [Save to file] area.**

5 **Click the [Save] button.**

Loading Security Settings

Use the following procedure to load a saved user account and user group settings into the Sensor Controller.

1 **If the settings were stored in external memory, insert the external memory device containing the security settings to load into the Sensor Controller.**

2 **In the Main Window, select [Security settings] from the [Tool] menu.**

The [Security settings] dialog box is displayed.

3 **Click the [Setting data] tab in the Item Tab Area.**

4 **Select the file to load from the [Load from file] area.**

5 **Click the [Load] button.**

Deleting Security Settings

Use the following procedure to delete saved user accounts and user groups settings.

IMPORTANT

The current security settings are all deleted.

1 **In the Main Window, select [Security settings] from the [Tool] menu.**

The [Security settings] dialog box is displayed.

2 **Click the [Setting] tab in the Item Tab Area.**

3 **Click the [Delete] button in the [Deletion of setting file] area.**

The security settings are deleted and the following message is displayed.
“User info logging in was changed now. Please log in again.”

4 **Click [OK].**

The [Login] dialog box is displayed.
Enter the default user and password “Administrator”.
If the login dialog box is canceled, the [Security setting] dialog box is closed.

Transferring Old Security Settings

Password settings that were made in FZ4 version 4.12 or earlier can be inherited.

IMPORTANT

This procedure will overwrite the current security settings.

1 **In the Main Window, select [Security settings] from the [Tool] menu.**

The [Security settings] dialog box is displayed.

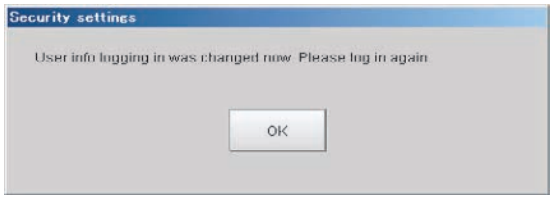
2 **Click the [Setting] tab in the Item Tab Area.**

3 **Click the [Execution] button in the [Shift of old security settings] area.**

The confirmation message, “The settings will be transferred and saved in the Controller. OK?” is displayed.

4 **Click [Yes].**

The security settings are changed.



5 **Click [OK].**

Enter the user name and password to log in again.

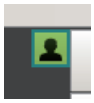
Changing between User Accounts

After a user account is set up, you can log in and log out with the registered user account. If you try to execute an operation you do not have the right to use, the login dialog box is displayed even though you are already logged in. In that case, log in with a user account that has the right to execute that operation.

Logging in

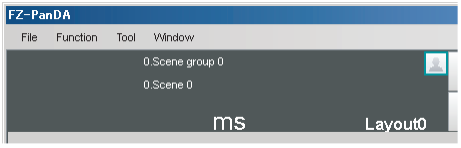
When you log in, you can execute those operations that the user account you logged in with has the right to execute.


Logged out


Logged in

1 **Click the [Login] icon in the Measurement Information Display Area.**

The [Login] dialog box is displayed.

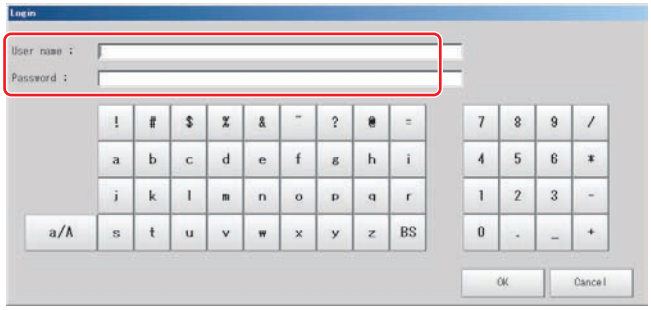


2 **Enter the user account and password.**

If the login fails, “The user name or the password is wrong” is displayed.
Enter the correct user name and password.

3 **Click [OK].**

The login icon switches to the logged-in state.



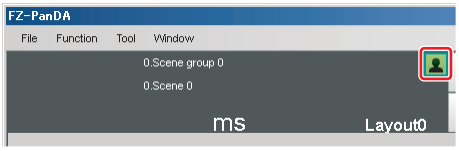
Logging Out

Note

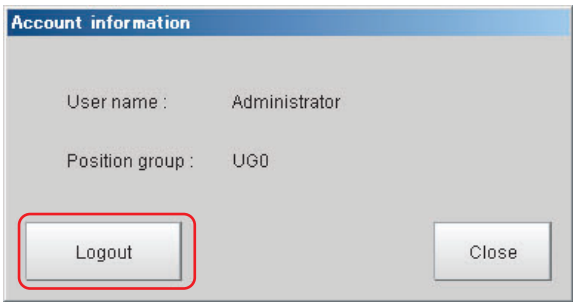
- If an automatic logout time is set, you will be automatically logged out if you do not perform any operations at all for the set time period.
- If you do not perform any operation, such as changing the display, after an automatic logout, the [Login] icon will still appear in the logged in state.
  The icon display will be updated if you change from the Main Window to a settings dialog box or an operating dialog box, or if you change to another layout.

1 **Click the [Login] in the Measurement Information Display Area.**

The [Login] dialog box is displayed.



2 **Click the [Logout] button.**



3 **Click [OK].**

The login icon changes to the logged-out state.

Overview

You can remotely edit the processing items that are performed by the Controller or remotely perform actual measurements on the Controller from a network computer. This function is enabled only with Ethernet connection.
Contact your OMRON sales representative to obtain a copy of the FZ_FH Remote Operation Tool CD-ROM.

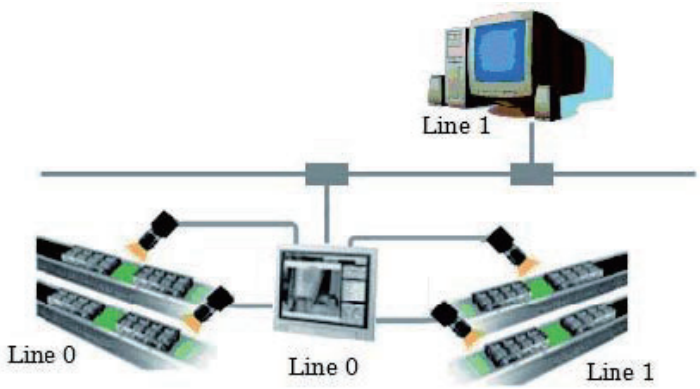
IMPORTANT

A Controller window cannot be operated or displayed simultaneously on multiple PCs on the network.

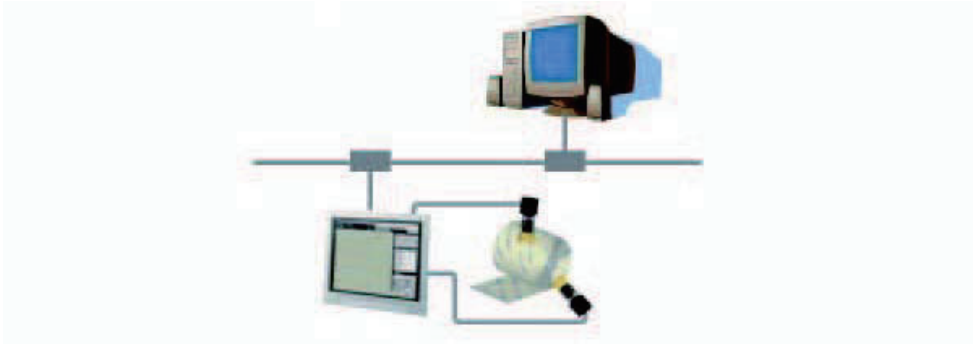
For example, the function can be used as follows.

(1) GUI operations, such as editing multiple inspection and measurement line processing items and changing the settings, can be performed by a single dedicated PC.

(2) Two measurement windows in Multi-line Random Trigger Mode can each be operated from a different window.



(3) In the non-stop adjustment mode, adjustments can be made remotely without having to stop the line measurement.



Note that the physical storage location of the data, such as the set data and the data save location, is on the Controller side.

Operation Environment Condition

The following is required on both the Controller and on the remote operation PC to start the remote operation.

- **Controller side = Set up a server to achieve the remote operation.**
- **Remote operation PC side = Prepare a communications environment and a GUI environment for remote operation.**

Install FZ_FH Remote Operation Tool on the remote operation PC.

IMPORTANT

Make sure that the software installed on the remote operation PC is of the same version as that on the Controller.

Environment Settings on the Remote Operation PC

The recommended operating conditions for the remote operation PC are as follows:

CPU	Intel Pentium Processor (SSE2 or higher)
OS	Windows 7 Professional (32/64bit) or Enterprise (32/64bit) or Ultimate (32/64bit), Windows 8 Pro (32/64bit) or Enterprise (32/64bit), Windows 8.1 Pro (32/64bit) or Enterprise (32/64bit)
Memory	2 GB (3 GB or higher recommended)
Hard disk free capacity	2 GB or higher
Display window	Resolution 1,024 x 768 dots or higher Display color True Color (32 bits)
Network	10Base-T compatible network (100Base-TX recommended)
Optical drive	CD/DVD drive

To use the remote operation software, Microsoft .NET Frameworks 3.5 must be installed.
This CD-ROM contains the Microsoft .NET Frameworks 3.5 installer. Please use as required.

IMPORTANT

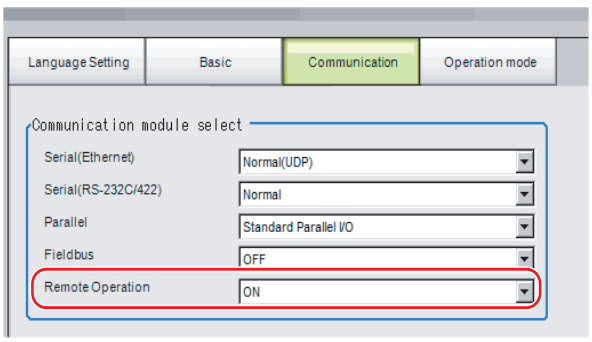
For remote operation, use the following ports on the PC to communicate with the Sensor Controller.
9900 + Line number (e.g., port 9900 for line 0)
9910 + Line number (e.g., port 9910 for line 0)
9920 + Line number (e.g., port 9920 for line 0)
Line Numbers: FH: 0 to 7, FZ5: 0 or 1
If you cannot use these ports due to security settings of your PC or the domain security policies, you will not be able to perform remote operation.
Either change the settings of the PC firewall or security software or contact your domain security administrator.

Set the network

Specify the network settings on both the Controller and the remote operation PC.
In the Communications Module, specify the module by serial (Ethernet) and enter the IP address.

1 Set the Communications Module.

In the Main Window, select [System settings] – [Startup] – [Startup settings] from the [Tool] menu, and then select [ON] for Remote Operation in the Communications Module Area.

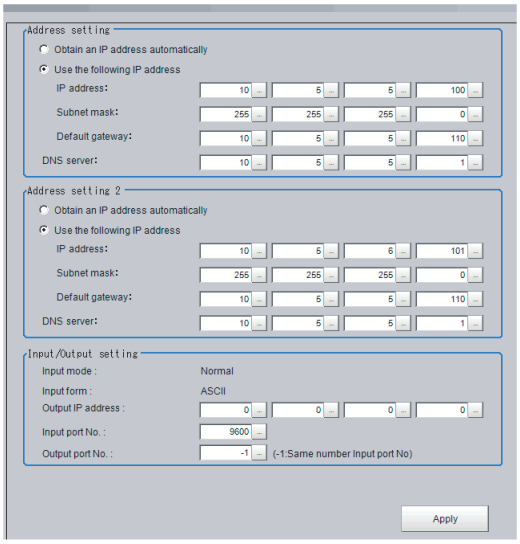


2 **Click the [Data save] button to save the settings, and then restart the Controller.**

3 **Set the IP address.**

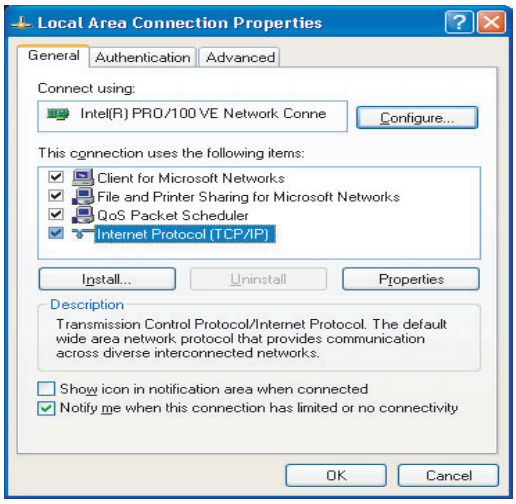
In the Main Window, select [System settings] – [Communications] – [Ethernet (*****)] from the [Tool] menu, and then set the IP address.

*1: The “(*****)” shows the [Serial (Ethernet)] setting that was set in [Communications] in step 1 above.

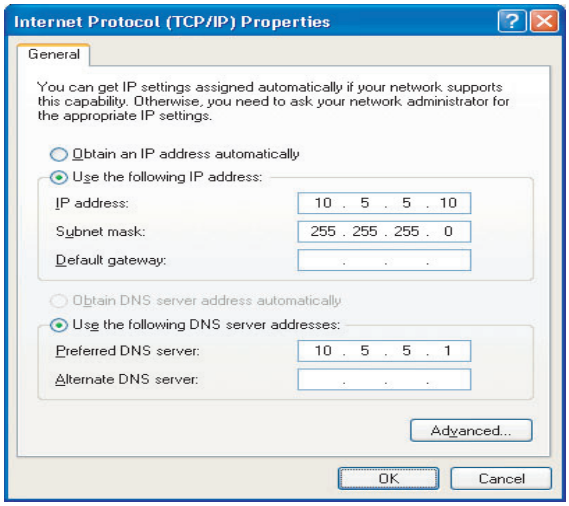


Next, specify the IP address of the remote operation PC.

Open the local area connection properties on the remote operation PC.



Enter the IP address.



How to Start

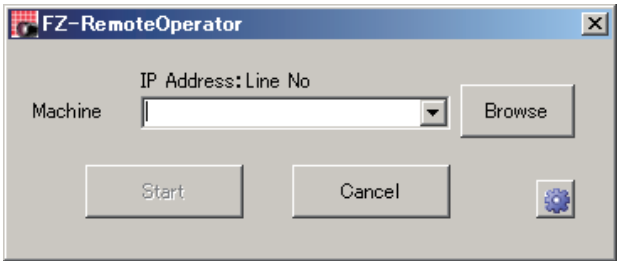
1 **From the Start button located in the lower left area of the window of the remote operation PC, select [Start] – [All Programs] – [OMRON] – [FZ_FH Simulator] and then select and launch [FZ_FH Remote Operation Tool].**

Note that a window that can be measured must be displayed on the Sensor Controller.

IMPORTANT

Do not connect or disconnect [FZ_FH Remote Operation Tool] when a measurement is being performed or the system is running.

2 **In the dialog box displayed by [FH-RemoteOperator], select or directly enter the IP address and the "Line No" of the Controller to be connected. Press [Browse] to find the IP address and the line number of the Controller that can be connected.**



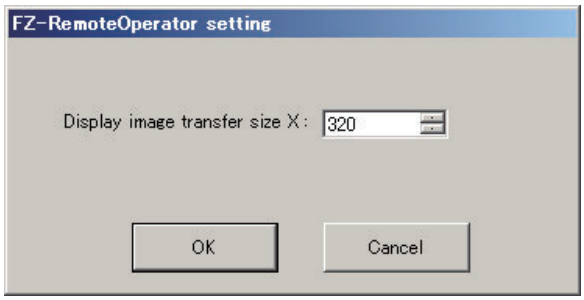
* The "Line No" selected here is one of the following based on the system's operation mode.

Operation mode		Setting
High-speed logging mode		Line No = 0
Parallel-operation high-speed mode		Line No = 0
Non-stop adjustment	Measurement window	Line No = 0
	Non-stop adjustment window	Line No = 1
Multi-line random trigger*1	Line 0 side	Line No = 0
	Line 1 side	Line No = 1
	Line 2 side	Line No = 2
	Line 3 side	Line No = 3
	Line 4 side	Line No = 4
	Line 5 side	Line No = 5
	Line 6 side	Line No = 6
	Line 7 side	Line No = 7

The Sensor Controller cannot be connected if a line number other than above is specified.

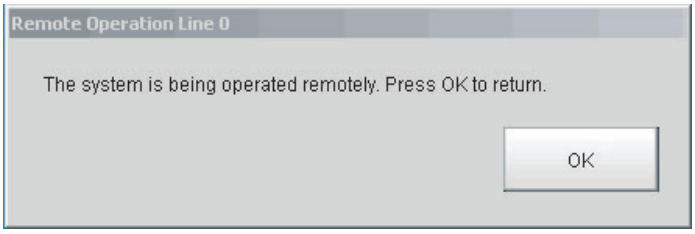
*1: The FZ5 Sensor Controller always has two lines (lines 0 and 1), and the FH Sensor Controller allows you to select as many as eight lines. Set the line numbers based on the number of lines that are set for the Sensor Controller.

3 **As necessary, change the size of the image to transfer for the remote operation. Select ( ) to set.**

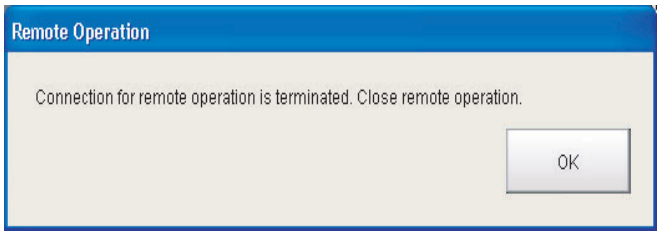


Description	Set value [Factory default]	Description
Display image transfer size (Size of the image to transfer)	4 to 640 [320]	This sets the size of the image to display on the remote operation window.

4 **Press the [Start] button. Once a remote operation is established, the remotely operated dialog box is displayed on the Sensor Controller.**



Click the OK button on this dialog box to terminate the remote operation from the Sensor Controller. The dialog box on which the remote operation is disconnected is displayed on the remote operation PC.



Note

When either the remote operation PC or the sensor controller is changed to Layout 8 by layout switching, operation on the sensor controller will be possible during remote operation. Note the following when using this feature:

- Switch either the remote operation PC or the sensor controller to Layout 8 with layout switching, not both. If the layouts of both are changed to Layout 8, operation on the sensor controller will not be possible during remote operation and only normal remote operation can be executed.
- The layout can be changed to Layout 8 during remote operation. After changing to Layout 8 on the remote operation PC, close the remotely operated dialog box on the sensor controller. Operation on the sensor controller will be possible while remote operation continues. To exit remote operation from this state, exit the FZ_FH Remote Operation Tool on the remote operation PC.
- In Layout 8, an error dialog is not shown when an error occurs and instead an error string appears in the Error Pane. If an error occurs, check the error in the Error Pane.
- In Layout 8, only the panes below can be used in Layout Modification Mode. To use panes other than these, use a layout other than Layout 8 and execute normal remote operation.
  - Flow Display Pane
  - Image Pane
  - Judgment Pane
  - Error Pane

Differences from Local Operation and Limitations

There is no fundamental difference between local operation on the Sensor Controller and a remote operation. The remote operation PC windows, however, are not the windows of the Sensor Controller themselves, and thus, the following types of operational differences can be caused.
Major differences are listed as follows:

Functions to be noted	Description	
	Sensor Controller side	Remote operation PC side
Controller restart	Automatically restart	The Controller automatically restarts, while the remote operation PC waits to be shut down manually.
Operation mode	All modes can be executed.	All modes, except for the single-line high-speed mode, can be executed.
Screen capture	The controller windows can be captured.	The controller windows are captured (not the remote PC windows).
Date-time setting	The system date and time on the controller are changed.	The system date and time on the remote operation PC are changed.
Image display	Displays the camera images and figures, etc.	Displays the camera images and figures, etc. with low pixels.
Unit Macro, Unit Calculation Macro, Customize I/O Command	Program editing is allowed.	Program editing is not allowed.
Number of Image Panes	24 max. (Image Pane numbers 0 to 23)	16 max.
Layout change	Changes in layout settings will be reflected to the actual layout after the layout switches.	Changes in layouts settings will be reflected to the actual layout immediately after finalizing the layouts edit. The displays on the Sensor Controller side and the remote operation PC side will not be consistent before the layout switch is performed on the sensor controller.

IMPORTANT

- The remote recovery dialog box may be hidden behind a window in the non-stop adjustment mode and the multi-line random trigger mode, both of which contain two windows. Press [Alt+Tab] to switch between the windows.
- The remote operation communications will be disconnected if a network setting (such as the IP address, subnet mask, and default gateway) on the Sensor Controller is modified using the remote operation function. Restart FZ_FH Remote Operation Tool to establish the connection again. The connection may fail immediately after a network setting has been changed. Please wait for a while before reconnecting.

Using Communication Command Macro

You can define your own communications commands for the Sensor Controller.
You use this to create functionality that is not available in the standard communications commands, or to perform more than one process with a single communications command.
Example: To tell the Controller to change scenes and then execute a measurement from an external device.

Without using communication command macro	Using communication command macro
You must send two standard communications commands: the Switch Scene command (scene) and the Measurement command (measure).	Create a measurement command (mymeasure) that is set to execute scene switching followed by measurement processing. One execution of the mymeasure command actually executes two commands, the scene command and measure command, which simplifies the control logic from the PLC or other external device.

User-defined communications commands are registered as a character string of the command name for the non-procedure communications protocol, or as a command number from 0 to 255. These commands are stored in the main memory of the Sensor Controller.
If you send a registered communications command from the external device to the Sensor Controller, the Sensor Controller recognizes it and performs the specified processing.

Flow Viewer

The Flow Viewer shows the flow of processing for the measurement flow that is currently being edited.
When you edit a flow, the Flow Viewer is useful when you need to check the relationship of branches and folders.
Refer to the following reference for information on the Flow Viewer.
Reference: ▶ Displaying and Checking Processing Branches in a Scene (p.56)

Custom Dialog Boxes

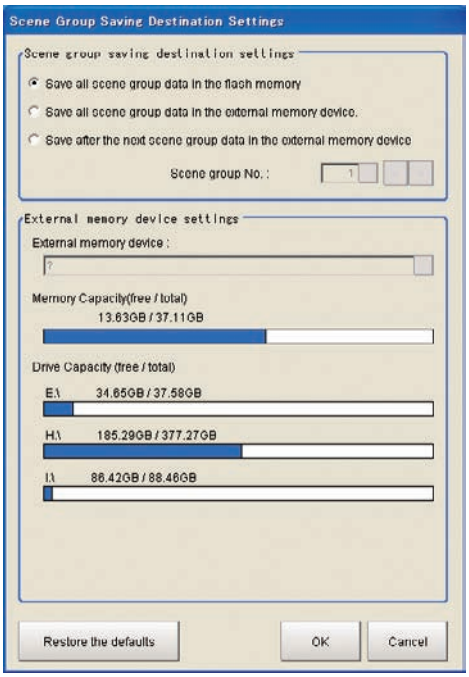
You can create a customized settings dialog box and display it instead of the normal settings dialog box that is provided for each processing item.
This allows you to create settings dialog boxes that match your application.
Refer to the following reference for information on custom dialog boxes.
Reference: ▶ Custom Dialog Tool (p.95)

Calibration Support Tool

The Calibration Support Tool provides a graphic display of the calibration setting results (relationship of the positions of the Camera coordinates and the actual coordinates) when you set the alignment function.
This allows you to check for major errors in the calibration results.
Refer to the following reference for information on the Calibration Support Tool.
Reference: ▶ Other Alignment Functions (p.315)

Saving Scene Groups in External Memory

The scene group storage location specification function lets you specify a location other than the Sensor Controller's flash memory for saving scene group data. After specifying the desired storage location with this function, menus and buttons such as "Data save"save set data in the specified external memory instead of in the Sensor Controller's flash memory. This function is handy as it enables use of external memory when the volume of scene group data is too large to save all of it in the Sensor Controller's flash memory.



1 **Select [Scene Group Saving Destination Setting] from the [Tool] menu in the main screen.**

A screen for specifying the scene group storage location appears.

2 **Set each item in the [Scene group saving destination on settings] area.**

Variable name	Set value [Factory default]	Content
Scene Group Saving Destination Setting	• [Save all scene group data in the flash memory] • Save all scene group data in the external memory device • Save after the next scene group data in the external memory device	Sets the storage location for scene group data. • Store all in Sensor Controller memory • Store all in external storage device • Store scene groups after scene group below in external storage device
Scene group No.	0 to 9999[1]	This setting is valid when [Save after the next scene group data in the external memory device] is selected for [Scene Group Saving Destination Setting]. Used to set the number of the first scene group to store in external memory. For the FH/FZ5 series, if 32 or above is specified for the scene group number, all scene group data will be saved to the Sensor memory.

3 Set each item in the [External memory device settings] area.

Variable name	Set value [Factory default]	Content
External memory device settings	---	This setting is valid when [Save all scene group data in the external memory device] or [Save after the next scene group data in the external memory device] is selected for [Scene Group Saving Destination Setting]. Used to set the storage location folder.

4 Click [OK].

The scene group storage location is changed, and the current scene group data is transferred from the current storage location to the new storage location. An error message is displayed if the transfer fails.

IMPORTANT

- An error message is displayed if scene group data storage or reading fails. See ”Error Messages and Troubleshooting” in the Appendices for descriptions of error messages.
  Reference: ▶ Error Messages and Troubleshooting (p.827)
- When the scene group data storage location setting is changed, a file with the [*.scg] file extension is created at the new storage location. Do not change the file name or file extension of this scg file. If you change, the setting will not be saved or read correctly.

Note

- You can check the available capacity of external memory in the setting screen. We recommend that you check whether your external memory has enough available capacity before changing the storage location.
- Clicking [Restore the defaults] restores the settings to their factory default values. There are different factory default settings for the FZ5-L35□/6□□ and the FH/FZ5-11□□.
- FZ5-L35□/6□□: Scene group 0 is stored in the Sensor Controller's flash memory; scene groups 1 and after are stored in external memory.
- FH/FZ5-11□□: Scene group 0, and scene groups 1 and after are stored in the Sensor Controller's flash memory.

MEMO

Optimizing Scenes (Measurement Flows) (Macro Customize Functions)

Basic Knowledge For Macro Customize Functions

In the FH/FZ5 series, the macro customize functions can be used to realize finely adjusted and expandable image processing.

What are the Macro Customize Functions?

The macro customize functions enable you to perform various types of calculations that are more advanced than normal "Calculation" processing items, as well as functions such as measurement flow/scene control, creation of communication commands, various types of display control, and result output control.
On the FH/FZ5 series, the following types of macro functions can be used.
The four types below can be used.

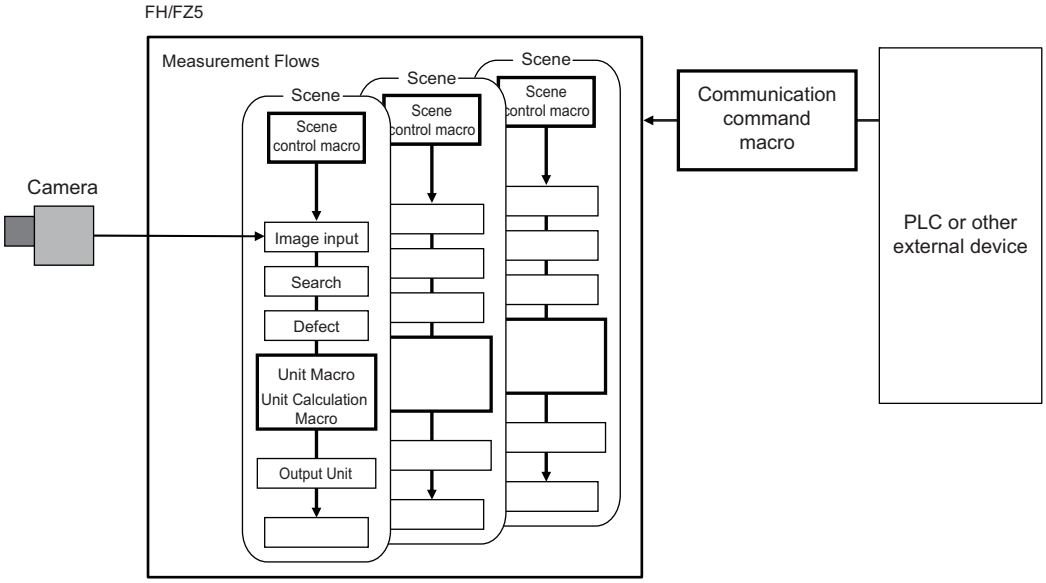
- "Unit Calculation Macro" processing item
- "Scene Control Macro" tool
- "Communication Command Macro" tool
- "Unit Macro" processing item

List of Macro Customize Functions

A table of the macro customize functions and their approximate level of difficulty is shown below.
The level of difficulty varies by function. Check the approximate level of difficulty in the table when considering the expansion you want to create.

Approximate level of difficulty	Function	Description	Target function			
			Measurement processing		Communication commands	Scene control
			Calculation	Other than calculation (*1)		
High ↓ Low	(1) "Unit Macro" processing item	Use this function to supplement and expand measurement processing performed by processing units. In addition to measurement processing, you can create your own custom processing such as result display processes and measurement initialization processes for scene control.	❍	❍	---	---
	(2) "Communication Command Macro" tool	Use this function to supplement and expand communication commands. You can create custom communication commands that implement functions that do not exist in the standard communication commands, and combine multiple communication commands into a signal command.	---	---	❍	---
	(3) "Scene Control Macro" tool	Use this function to supplement and expand measurement flow and scene control. You can create branches and repeated processes in the measurement flow, and add processing units to the measurement flow and configure settings for the units.	---	---	---	❍
	(4) "Unit Calculation Macro" processing item	Use this function to supplement and expand calculation processes during measurement. A "Calculation" processing item allows you to implement complex operations with difficult settings, and calculations that include a logical expression or repeated process.	❍	---	---	---

*1: Result display (graphic display, detailed text display, etc.), result output, and creation of initial processing



In the following cases, a macro customize function is used.

In this case		Macro customize function that is used
Calculation	Use to perform a calculation process that is difficult or cannot be expressed using a "Calculation" processing item, such as those below. • Use to execute a calculation process that extends over multiple lines or contains a logical expression, conditional branch, loop process, or data setting process • Use to execute a customized judgement process for calculation results	"Unit Calculation Macro" processing item
Calculation, Display result, Output result	Use to customize or include the following processes that that are difficult or cannot be expressed with a "Calculation" processing item or a "Unit Calculation Macro" processing item. • Display result (graphic display, detailed text display, etc.) • Output result • Measurement initialization processing (scene switching, etc.)	"Unit Macro" processing item
Scene control	Use to perform the following types of scene control in a measurement flow. • Changing the settings of multiple processing units at once • Managing the common data of multiple processing units • Adding/deleting a processing unit	"Scene Control Macro" tool
Communication commands	Use to create expansions that are difficult or cannot be expressed with the standard communication commands, such as those below. • Not to exist in the standard communication commands. • To combine multiple communication command functions into a single communication command	"Communication Command Macro" tool

Structure of This Chapter

The relation between the contents of this chapter and the macro customize functions is shown below. Refer to function items that you need to use.

Item name	Macro customize function			
	Unit Calculation Macro	Scene Control Macro	Communication Command Macro	Unit Macro
What are the Macro Customize Functions? Reference: ▶Basic Method for Writing Programs (p.205)	Required	Required	Required	Required
How to use Macro Customize Functions Reference: ▶How to use Macro Customize Functions (p.176)	Required	Required	Required	Required
Basic Method for Writing Programs Reference: ▶Basic Method for Writing Programs (p.205)	Required	Required	Required	Required
How to Write Advanced Programs	---	---	---	---
Data Types Related to Processing Units Reference: ▶Data Types Related to Processing Units (p.217)	As needed	As needed	As needed	As needed
Data Types Related to the System Reference: ▶Data Types Related to the System (p.222)	As needed	As needed	As needed	As needed
Scope of Data and Save Area Reference: ▶Scope of Data and Save Area (p.225)	As needed	As needed	As needed	As needed
Status Transitions and Execution Timing Reference: ▶State Transitions and Execution Timing (p.227)	Not required	As needed	Not required	As needed
Exclusive Control in a Process Reference: ▶Exclusive Control in a Process (p.234)	As needed	As needed	As needed	As needed
Preparations for use of macro customize functions	---	---	---	---
Preparations for use of the "Unit Calculation Macro" processing item Reference: ▶Procedure for Using the unit calculation macro processing item (p.178)	Required	Not required	Not required	Not required
Preparations for use of the scene control macro Reference: ▶Procedure for Using the Scene Control Macro Tool (p.179)	Not required	Required	Not required	Not required
Preparations for use of the communication command macro tool Reference: ▶Procedure for Using the Communication Command Macro Tool (p.179)	Not required	Not required	Required	Not required
Preparations for use of the "Unit Macro" processing item Reference: ▶Procedure for Using the unit macro processing item (p.180)	Not required	Not required	Not required	Required
How to Use the Debug Function Reference: ▶How to Use the Debug Function (p.235)	Required	Required	Required	Required
Troubleshooting Reference: ▶Troubleshooting (p.245)	As needed	As needed	As needed	As needed

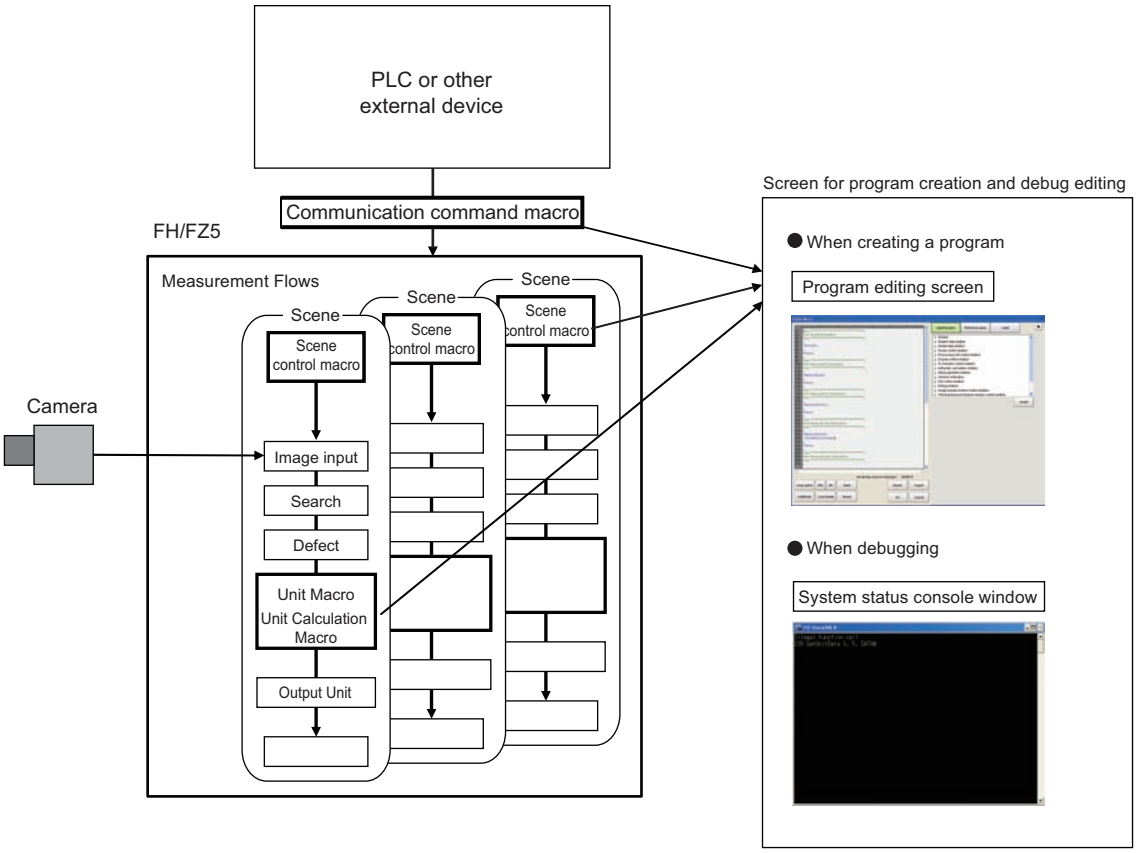
How to use Macro Customize Functions

Necessity of referring to this manual	Unit Macro	Communication Command Macro	Scene Control Macro	Unit Calculation Macro
	Required			

You can use the macro customize functions to program processes that you want to add or expand. On a sensor controller, you can execute the macro customize functions and debug the programs of the macro customize functions.

Components of the Macro Customize Functions

The macro customize functions consist of program editing screens that enable the creation and editing of programs in the processing items, tool setting screens, and setting screens of each function, and a system status console window that allow errors to be checked when a program operates abnormally.



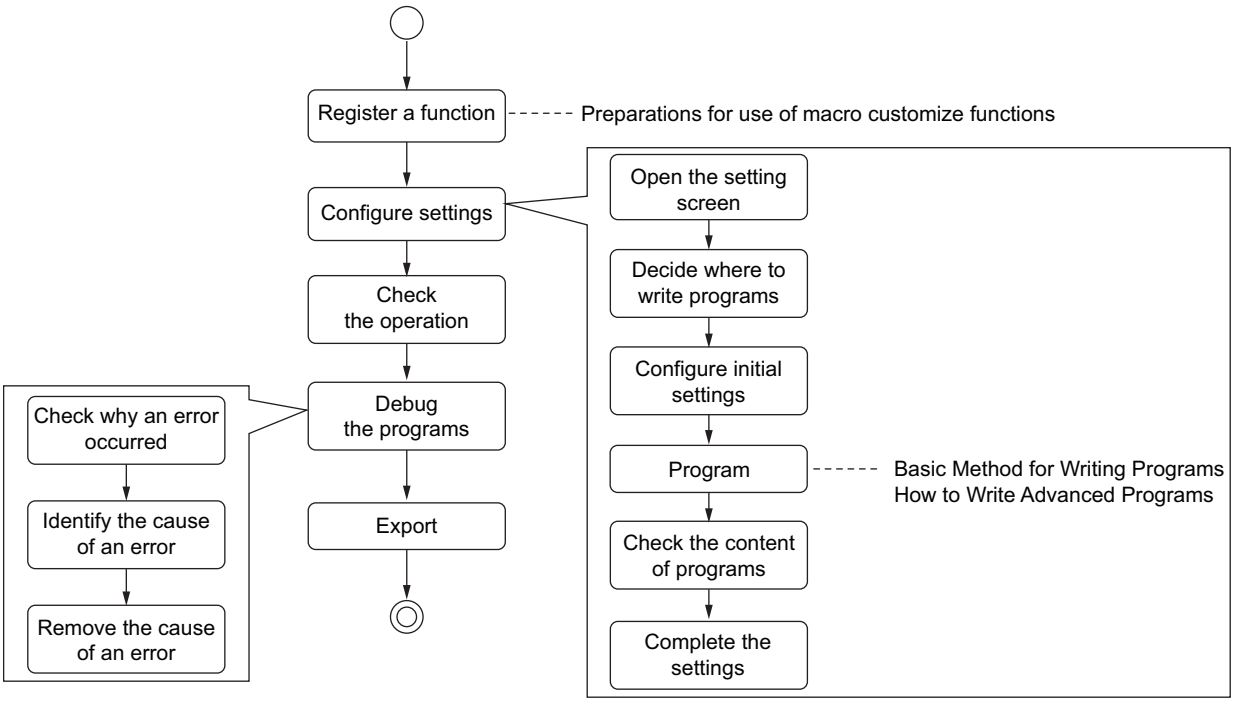
Function	Description
Program editing screen	Setting screen of each function in the macro customize functions. The contents of the program editing screen vary by function. Use the program editing screen to create programs. You can create and edit programs, and use input auxiliary functions. Reference: ▶Components of the Program Editing Screen (p.181)
System status console window	Console window that shows the system status as text. When a program created with a macro customize function operates abnormally, a description of the error appears in text in the system status console window. Use this to debug the program. Reference: ▶Description of the System Status Console Window (p.185)

IMPORTANT

The setting screens of the macro customize functions cannot be displayed by remote operation. To change settings, directly open the setting screen of the function on the sensor controller.

Procedures for Using the Macro Customize Functions

The procedures for using macro customize functions are two types, one is a common procedure used for all functions and second is the specific procedures for each function.
The flow of the common procedure for using the macro customize functions is shown below.



Item name	Step	Description
Function registration and preparation	Register a function and display the setting screen	Register processing items that enable use of macro customize functions in the measurement flow, and open tool screens. The specific procedure depends on the each functions.
---	↓	---
Function settings	Default function settings	Prepare initial settings and variables. The specific procedure depends on the each functions.
	↓	---
	Creation of processing content	Write the program.
	↓	---
Program debug	Check operation	Check if your settings operate as expected.
	↓	---
	Debug	Debug the program. Debug helps you identify the cause of the unexpected operation and correct the program to run as expected.
---	↓	---
Save	Save settings	Save your changes.

Procedure for Using the unit calculation macro processing item

The usage flow for the unit calculation macro processing item and the basic usage procedures are described below.

Item name	Step	Description
---	Registration of a unit calculation macro processing unit	Add a macro calculation processing unit to the measurement flow.
---	↓	---
Settings of the unit calculation macro processing unit (Reference: ▶Description of the Setting Screen for the "Unit Calculation Macro" Processing Item and How to Configure Settings (p.186))	Selection of operators	Select the checkboxes of operators to be used in the unit calculation macro.
	↓	---
	Reference Variable Settings	Set reference values if reference values will be used. Set reference values in order to use data other than that of the macro calculation processing unit, such as external reference data of other processing units and system data.
	↓	---
	Program input	Write the program.
	↓	---
	Setting of judgement conditions	Set the conditions used to judge calculation results.
	↓	---
Program debug (Reference: ▶How to Use the Debug Function (p.235))	Check operation	Check if your settings operate as expected.
	↓	---
	Debug	Debug the program. Debug helps you identify the cause of the unexpected operation and correct the program to make the unit calculation macro processing unit operate as expected.
---	↓	---
---	Saving settings	Save your changes.

Procedure for Using the Scene Control Macro Tool

The usage flow for the scene control macro tool and the basic usage procedures are described below.

Item name	Step	Description
---	Starting the Scene Control Macro Tool	Start the scene control macro tool from the external tools.
---	↓	---
Scene control macro tool settings (Reference: ▶Description of the Setting Screen of the Scene Control Macro Tool and How to Configure Settings (p.190))	Reference Variable Settings	Set reference values if reference values will be used. Set reference variables in order to use data such as external reference data of processing units and system data.
	↓	---
	Unit label settings	Set unit labels if unit labels will be used. Make preparations to use unit labels, rather than processing unit numbers, to reference processing units in the measurement flow.
	↓	---
	Program input	Write in the program.
	↓	---
Program debug (Reference: ▶How to Use the Debug Function (p.235))	Check operation	Check if your settings operate as expected.
	↓	---
	Debug	Debug the program. Debug helps you identify the cause of the unexpected operation and correct the program to make the scene control macro operate correctly.
---	↓	---
---	Saving settings	Save your changes.

Procedure for Using the Communication Command Macro Tool

The usage flow for the communication command macro tool and the basic usage procedure are described below.

Item name	Step	Description
---	Starting the communication command macro tool	Start the communication command macro tool from the external tools.
---	↓	---
Communication command macro settings (Reference: ▶Components of the Setting Screen of the Communication Command Macro Tool and How to Configure Settings (p.192))	Selection of the communication command macro to be used	Select the checkbox of the communication command macro to be used.
	↓	---
	Communication command macro name setting	Assign a name to the communication command macro.
	↓	---
	Flow signal output setting	Select whether flow signals such as the BUSY signal are turned ON or left OFF during processing. To turn on, select the checkbox.
	↓	---
	Program input	Write the program.
	↓	---

Item name	Step	Description
Program debug (Reference: ▶How to Use the Debug Function (p.235))	Check operation	Check if your settings operate as expected.
	↓	---
	Debug	Debug the program. Debug helps you identify the cause of the unexpected operation and correct the program to make the communication command macro operate as expected.
---	↓	---
---	Save settings	Save your changes.

Procedure for Using the unit macro processing item

The usage flow for the unit macro processing item and the basic usage procedure are described below.

Item name	Step	Description
---	Registration of unit macro processing unit	Add a unit macro processing unit to the measurement flow.
---	↓	---
Settings of unit macro processing unit (Reference: ▶Description of the Setting Screen of the "Unit Macro" Processing Item and How to Configure Settings (p.199))	Reference Variable Settings	Set reference values if reference values will be used. Set reference values to use data other than that of the unit macro processing unit, such as external reference data of other processing units and system data.
	↓	---
	Program input	Write the program.
	↓	---
Program debug (Reference: ▶How to Use the Debug Function (p.235))	Check operation	Check if your settings operate as expected.
	↓	---
	Debug	Debug the program. Debug helps you identify the cause of the unexpected operation and correct the program to make the unit macro processing unit operate as expected.
---	↓	---
---	Save settings	Save your changes.

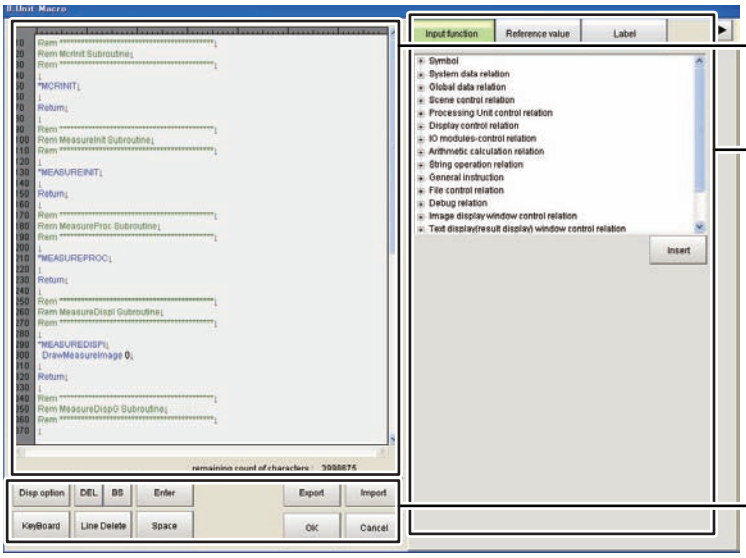
Components of the Screens and How to Configure Settings

Necessity of referring to this manual	Unit Macro	Communication Command Macro	Scene Control Macro	Unit Calculation Macro
	Required			

The setting screens of the macro customize function consist of a program editing screen that is shared by all macro customize functions, and individual setting screens for each function. The setting methods vary by setting screen.

Components of the Program Editing Screen

Use the program editing screen to edit a program in the macro customize function.
The program editing screen consists of the areas below.

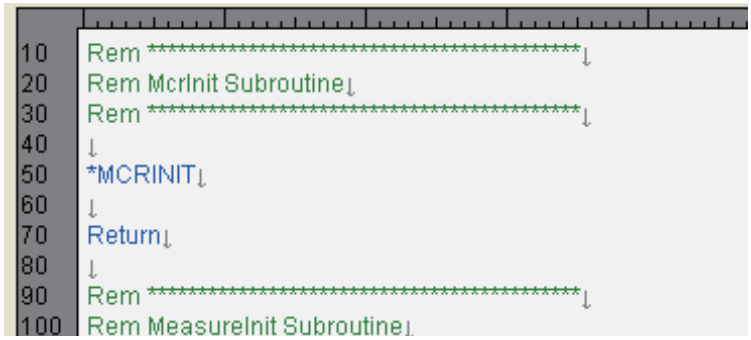


1. Program area

The program appears in this area. You can select whether some of the display items are displayed.
Reference: ▶Display option (p.184)

- **Program input area**

  This area is used to enter programs. Create and edit programs in this area.
  When creating and editing programs in the program input area, use the keyboard to write the program in the program input area.
  The screen keyboard can be displayed from the operation button area.
  Reference: ▶3. Operation button area (p.183)



  The control characters are displayed in a visible form in the program area. Control characters are included in the remaining count of characters, and thus are convenient for checking the number of inputtable characters.
  Reference: ▶Remaining count of characters (p.182)

Display (color)	Description
↓ (gray)	Indicates a line break.
□ (gray)	Indicates a double-byte space.
→ (gray)	Indicates the tab character.

Note

If a function entered in a program cannot be used, the function name is shown in red.
Whether or not a function can be used depends on the macro customize function that is used. For details, refer to Macro Function Reference.
Reference: ▶Macro Reference (p.320)

- **Line number display area**

  The program line number appears in this area. The line number is used in debugging. For lines and line numbers, refer to the Basic Syntax section.
  Reference: ▶Basic Syntax (p.205)

Note

Line numbers are assigned as unique numbers in the program. When multiple calculation expressions are set for one unit calculation macro processing unit, unique line numbers that are not redundantly used in the multiple calculation expressions in the processing unit are assigned.

- **Remaining count of characters**

  This area shows how many characters can still be input in the program. Control characters such as line breaks and tabs are also included in the count. Create/edit the program so that the remaining inputtable character count is 0 or more. If a program has more than the inputtable character count, it may not operate correctly.

2. Supplemental Program Input Area

This area can be used to input supplemental settings and perform supplemental input operations for programs.
To hide the supplemental program input area and enlarge the program area, click the [▶] button.

- **Input function tab**

  This area shows a list of the macro functions.
  After selecting a macro function in the list, click the [Insert] button to insert the selected macro function immediately behind the cursor position in the program input area.

- **Reference variable tab**

  This area allows to register and configure settings for reference variables. This area only appears in the program editing screen for unit calculation macros, scene control macros, and unit macros.
  The reference variables set in the reference variable area can only be used by the processing unit of that setting screen and in scene control macro of that setting screen. The reference variable area is not shown in the communication command macro.
  For details on reference variables, refer to the Variables section.
  Reference: ▶Variable (p.209)

Button	Description
Add	Displays the reference variable setting screen. You can add a reference variable in the setting screen.

Button	Description
Delete	Deletes a reference variable that has been selected in the reference variable list.
Edit	Displays the setting screen of a reference variable selected in the reference variable list. You can change the settings of the reference variable in the setting screen.
Export	Displays the save file screen. Specify the save location and file name to save the settings of the current reference variable list as a file in XML format.
Import	Displays the import file screen. Imports reference variables saved in a file in XML format.

- **Label tab**

  This area can be used to set and reference unit labels. Unit labels can only be set in the scene control macro. In a macro customize function other than the scene control macro, unit labels settings can only be referenced.

  For details on unit labels, refer to "Description of the Setting Screen for the Scene Control Macro Tool and How to Configure Settings".

  Reference: ▶Description of the Setting Screen of the Scene Control Macro Tool and How to Configure Settings (p.190)

3. Operation button area

This area contains buttons for editing programs by button operation, and buttons for exporting and importing programs.

The buttons that can be used depend on the macro customize function that is used.

Buttons that can be used in the operation button area are shown below.

Button	Description
Display Settings	Shows the display settings screen.
Keyboard	Displays the screen keyboard. Use the keyboard to create and edit programs.
DEL	Deletes one character immediately after the cursor position in the program input area.
BS	Deletes one character immediately in front of the cursor position in the program input area.
Enter	Breaks the line at the cursor position in the program input area.
Line Delete	Deletes the line where the cursor is located in the program input area.
Space	Inserts a space at the cursor position in the program input area.
Clear	Initializes the program shown in the program area. Only appears for the scene control macro.
Export	Exports the program file to a file. Only appears for the unit macro.
Import	Imports a program file from a file. Only appears for the unit macro.
OK	Finalizes the edited contents of the program editing screen and closes the screen. If there is a error in the program, the error dialog box appears. If the error dialog box appears, remove the error and click OK. A description of the error appears in the system status console window.
Cancel	Discards editing changes in the program editing screen and closes the screen.

Note

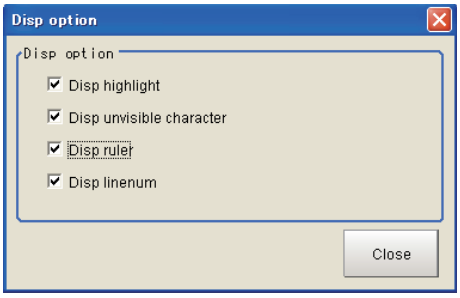
- In the FH/FZ5-11□□ series, the layout of the keyboard on the sensor controller is the same as an English keyboard.To enter Japanese, you can change the input mode with [ALT] + [~].
- If the program has an error, the error dialog appears when [OK] is clicked. If the error dialog box appears, remove the error and click OK to complete the settings. You can view a description of the error in the system status console window.

  Reference: ▶Structure of the System Status Console Window (p.185)

Display option

To add the contents of the program area and make the program easier to view, change the program area display settings.

The settings in the display option screen are described below.



Setting item	Setting value [Factory default]	Description
Disp highlight	• [Checked] • Unchecked	Highlights the program in the program input area with distinguishing colors. Each keyword type is highlighted in a different color, making the program easier to view.
Disp unvisible character	• [Checked] • Unchecked	Shows line breaks, tabs, and other control characters in the program input area in a visible form. For details on hidden characters, refer to Program Area. Reference: ▶1. Program area (p.181)
Disp ruler	• [Checked] • Unchecked	Shows the ruler in the program input area. Showing the ruler makes it easier to check the number of characters on one line.
Disp linenum	• [Checked] • Unchecked	Shows line numbers on the left side of the program input area. Showing line numbers makes it easier to identify lines where errors occur during debug.

Note

- You can increase the size of the program area and make the program easier to view by removing the checkmarks from the display settings. This will also improve the program editing response.
- The display settings are not saved. When the program editing screen is closed, the display settings revert to the factory default settings.
- When "Disp highlight" is selected and a function entered in a program cannot be used, the function name is shown in red.
  Whether or not a function can be used depends on the macro customize function that is used. For details, refer to Macro Function Reference.
  Reference: ▶Macro Reference (p.320)

Reference Variable Settings

Set the reference variables used in the program. For details on reference variables, refer to the Variables section.

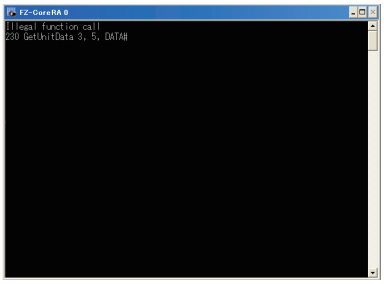
Reference: ▶Variable (p.209)

Description of the System Status Console Window

Use the system status console window to debug macro customize programs and check error descriptions.

Structure of the System Status Console Window

The information shown in the system status console window is described below.



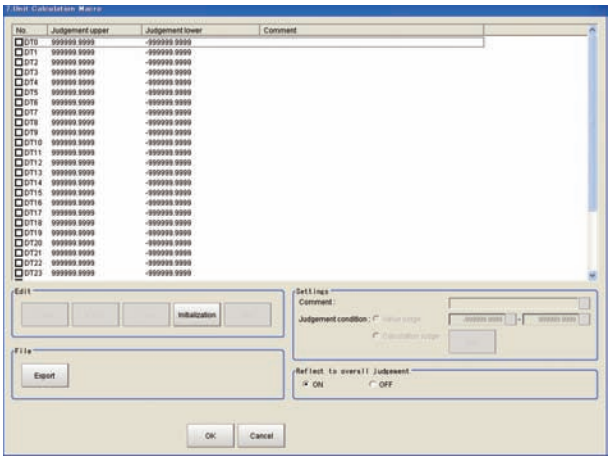
Display area	Description
System status display area	Shows the system status as text. When a program created with a macro customize function operates abnormally, a description of the error appears in the system status display area. For the information displayed, refer to "Checking the System Status Console Window". Reference: ▶Structure of the System Status Console Window (p.185)

IMPORTANT

- In the FZ5-L3□□/FZ5-6□□ series, the system status console window appears in the full screen. To display the main screen, program editing screen, and other sensor controller screens, connect a USB keyboard and change the screen with [ALT] + [TAB].
- Do not close the system status console window by a method such as clicking the "x" button in the upper right corner of the system status console window. The system may not operate correctly. If the system status console window is accidentally closed, save your settings and restart the sensor controller.

Description of the Setting Screen for the "Unit Calculation Macro" Processing Item and How to Configure Settings

The components of the properties screen of the "Unit Calculation Macro" processing item are described below.



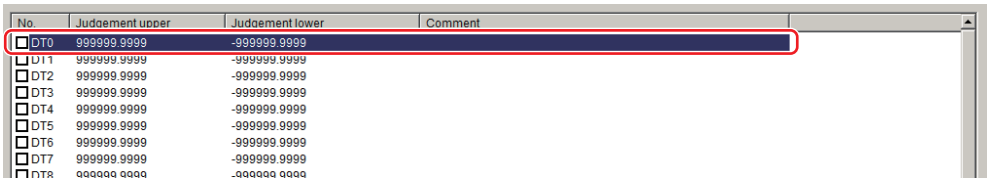
Selection of Operators (Unit Calculation Macro)

32 calculation processes from "DT0" to "DT31" can be set per unit.

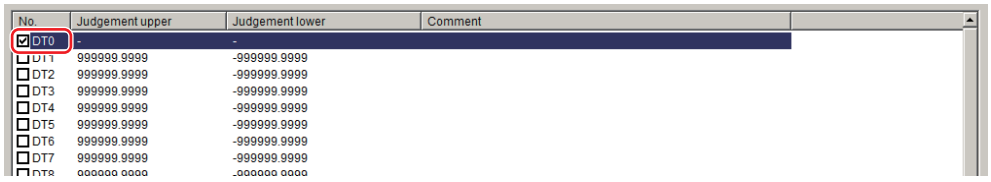
Note

- Calculation results cannot be output to external devices when you only set up macro calculations. When calculation results are output to external devices, set processing items related to results output in units after "Unit Calculation Macro" with flow editing.
  Reference: ▶"Output Result" in the "Vision System FH/FZ5 Series Processing Item Function Reference Manual (Cat. No. Z341)"

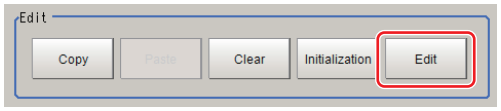
1 **From the list, click the operator of the calculation processing to be set.**



2 **Place a check at the operator to use to perform the calculation processing.**



3 **In the Edit area, click [Edit].**

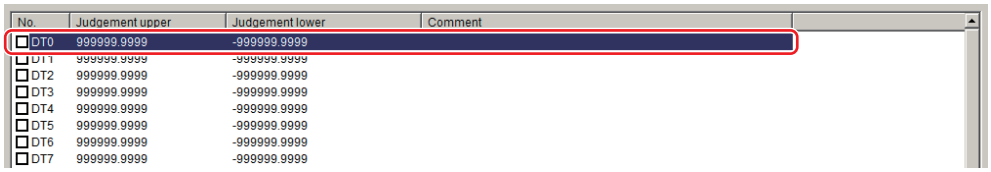


The unit calculation macro setting window is displayed.

Editing Operator (Unit Calculation Macro)

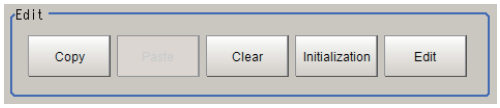
The calculation processing you have set can be copied or cleared.

1 **From the list, click the operator of the calculation processing whose setting is to be edited.**



2 **Select each operation in the "Edit" area.**

The Unit Calculation Macro settings screen is displayed.



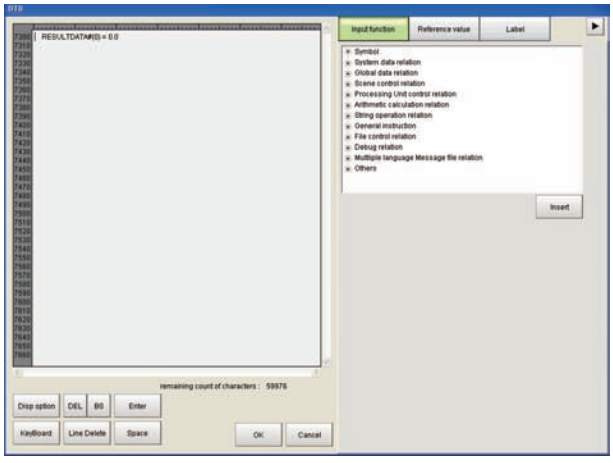
Item	Description
Copy	Copy the setting of the operator you have selected from the list. The copied setting can be pasted to other operator via [Paste].
Paste	Paste the copied set value to the operator selected from the list. Data that can be pasted includes valid/invalid flags, calculation macro codes, comments, judgement flags, upper/lower limits of figure judgement and judgement macro codes.
Clear	Initialize the setting of the operator you have selected from the list. Data to be initialized includes valid/invalid flags, calculation macro codes, comments, judgement flags, upper/lower limits of figure judgement and judgement macro codes.
Initialization	Initialize the settings of all operators.
Edit	Edit the setting of the operator you have selected from the list. Reference: ▶Selection of Operators (Unit Calculation Macro) (p.186)

Reference Variable Settings (Unit Calculation Macro)

Set the reference variables used in the program. The reference variables setting method is the same method as for the unit macro processing item.

Reference: ▶Description of the Setting Screen of the "Unit Macro" Processing Item and How to Configure Settings (p.199)

Program Input (Unit Calculation Macro)

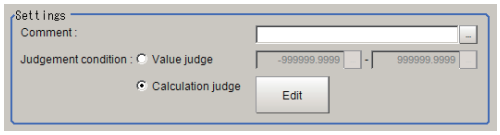


The program input method is the same method as for the unit macro processing item.

Reference: ▶Program Input (Unit Macro) (p.200)

Judgement Condition Settings (Unit Calculation Macro)

1 Set judgement conditions in the "Settings" area.



Setting item	Setting value [Factory default]	Description
Comment	-	Enter a comment on the calculation processing you have selected from the list.
Judgement condition	• [Value judge] • Calculation judge	Select whether to use a figure or macro judgement for the calculation result. If figure judgement selected, set the upper/lower limits of OK judgement. If macro judgement is selected, click [Edit] and define the calculation processing to be performed on the calculated value.

2 Select whether or not to reflect the judgement result in the scene overall judgement in "Reflect to overall judgement" area.

Setting item	Setting value [Factory default]	Description
Reflect to overall judgement	• [ON] • OFF	Specify whether or not the judgement results of this processing unit is reflected in the scene overall judgement.

Key Points for Test Measurement and Adjustment (Unit Calculation Macro)

To increase the speed and precision of measurement, you can adjust the parameters by performing and checking the results of test measurements.
The following content is displayed in the "Detail result" area as text.

Displayed items	Description
Judge	Judgement result
Calculation 0 comment	Calculation 0 value
Calculation 1 comment	Calculation 1 value
:	:
Calculation 31 comment	Calculation 31 value

The image specified in the sub image in image display setting is displayed in the image display area.

Sub image.	Explanation of image to be displayed
0	Measurement image

Key Points for Adjustment

Select the adjustment method referring to the following points.

An error message appears on the console

Parameter to be adjusted	Troubleshooting
-	Refer to the error messages list. Reference: ▶"Appendixes About Macro Functions List of Macro Error Messages" in the "Vision System FH/FZ5 Series User's Manual (Z340)"

Nothing happens when [DEL], [BS], [Enter], etc., is clicked

Parameter to be adjusted	Troubleshooting
-	Nothing happens while the focus is not on the code window (key entry cursor is not displayed). Click the position you want to operate, and then click the button.

Want to include a line feed code in a string

Parameter to be adjusted	Troubleshooting
Macro code	Add (+) CR → Chr$(13) LF → Chr$(10) to the string.

Calculation result is indicated as "Unmeasured"

Parameter to be adjusted	Troubleshooting
Enabled/disabled	Place a check to enable the operator.
Calculation judgement	The judgement result may not be set correctly in calculation judgement.

Measurement Results That Can be Output (Unit Calculation Macro)

The measurement results provided by the unit calculation macro are shown below. The measurement results appear in the detailed result area, and you can also use a result output processing item to output the measurement results to an external device.

Refer to the "External Reference Tables" for the parameters that can be referenced, including measurement results. (Reference: ▶External Reference Tables (Unit Calculation Macro) (p.189))

Measurement items	Character string	Description
Judge	JG	Judgement result
Calculation result 0	DT00	Calculation result 0
Judgement result 0	JG00	Judgement result 0
:		
Calculation result 31	DT31	Calculation result 31
Judgement result 31	JG31	Judgement result 31

External Reference Tables (Unit Calculation Macro)

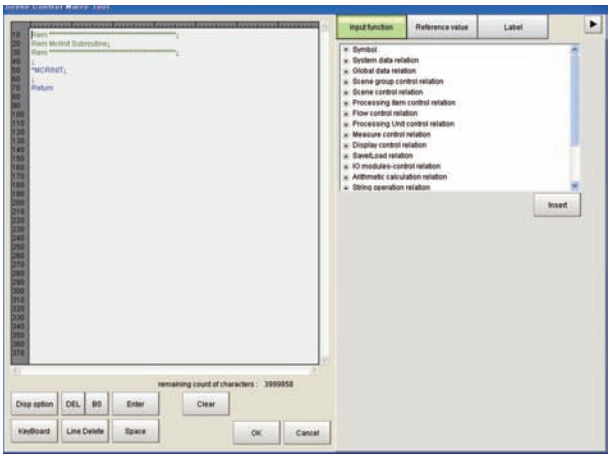
By specifying a number, you can access the following data from processing items that support processing unit data setting/acquisition, and from control commands.

No.	Data name	Set/Get	Data range
0	Judge	Get only	0: No judgement (unmeasured) 1: Judgement result OK -1: Judgement result NG
5 + N × 1 (N = 0 to 31)	Calculation result N (N = 0 to 31)	Set/Get	-99999.9999 to 99999.9999
37 + N × 1 (N = 0 to 31)	Judgement result N (N = 0 to 31)	Get only	-99999.9999 to 99999.9999

Description of the Setting Screen of the Scene Control Macro Tool and How to Configure Settings

The setting screen for the scene control macro tool is the same as program editing screen. For details on the setting screen, refer to Program Editing Screen.

Reference: Reference: ▶Components of the Screens and How to Configure Settings (p.181)



Reference Variable Settings (Scene Control Macro)

Set the reference variables used in the program. The reference variables setting method is the same method as for the unit macro processing item.

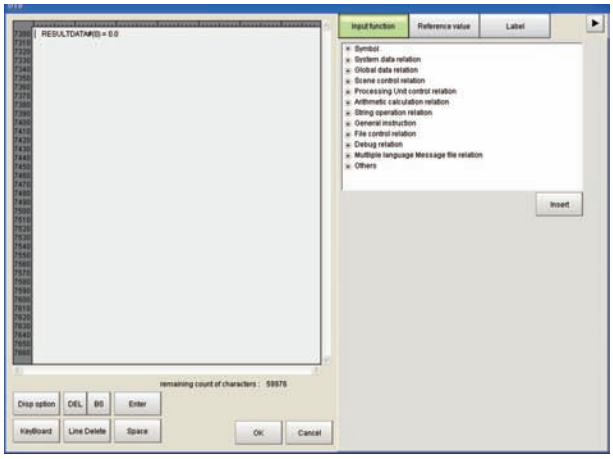
Reference: ▶Reference Variable Settings (Unit Macro) (p.199)

Unit Label Settings (Scene Control Macro)

The scene control macro can be used to set unit labels. Setting a unit label allows you to specify a processing unit in a program by the label rather than the processing unit number. The unit label and Ut function is used to specify the processing unit. Using unit labels eliminates the need to change the program when processing unit numbers change due to a change of measurement flow.

Macro customize functions other than the scene control macro cannot be used to set unit labels. These can only be used to reference unit labels already set with the scene control macro.

Reference: ▶ Ut function in Macro Command Reference (p.793)



1 **Click the "Label" tab of the setting screen for the scene control macro tool.**

A list of the unit labels set in the current scene appears.

2 **Click [Edit].**

The Unit Label screen appears.

3 **Select the processing unit for which you want to set a unit label.**

4 Click "..." and set the unit label name.

You will return to the setting screen for the scene control macro tool.

Setting item	Setting value	Description
Unit Label	1 to 32 characters	Set the unit label name. The unit label set here is used by the Ut function. You can use letters, numbers, ".", and "_" in the unit label name.

5 Click [OK].

You will return to the setting screen for the scene control macro tool.

IMPORTANT

- The unit label setting is saved in the scene data of the scene. If you want to use the unit label setting in another scene, repeat the setting in that scene, or use the scene maintenance function to copy the scene.
  Reference: ▶Editing Scenes (p.62)
- The same unit label cannot be set twice in the same scene.

Note

"Edit" only appears on the Label tab of the setting screen of the scene control macro.

Program Input (Scene Control Macro)

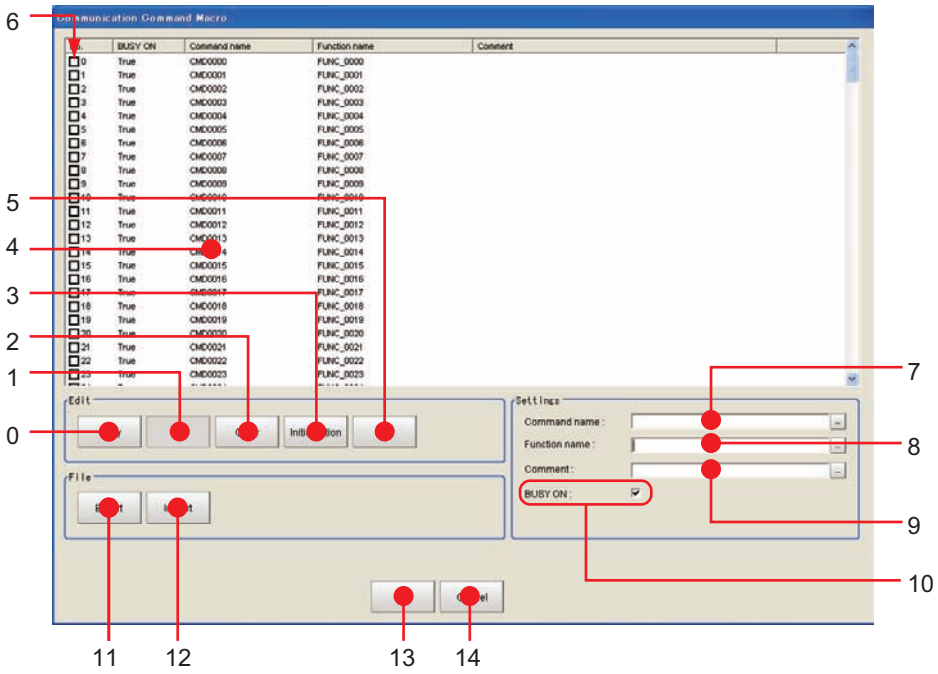
The program input method is the same method as for the unit macro processing item.
Reference: ▶Program Input (Unit Macro) (p.200)

Components of the Setting Screen of the Communication Command Macro Tool and How to Configure Settings

The components of the setting screen of the communication command macro tool are indicated below.

Setting Procedure for Communication Command Macro

You can create and edit communication command macros.
You can create up to 256 (from 0 to 255) communication command macros.



Descriptions of Dialog Box Objects

No.	Name	Description
0	Copy	Copy the selected command in the command list.
1	Paste	The selected command is overwritten by the copied information. Paste targets are "comments" and "processing details." If nothing is copied, this is disabled.
2	Clear	The information of selected command, such as "Command name" "Function Name" "Comment" and "Program" is initialized.
3	Initialization	The information of all of the commands, such as "Command name" "Function Name" "Comment" and "Program" is initialized.
4	Command list	Display the list of the commands.
5	Edit	Launch the Macro program editor for selected command.
6	Enable/Disable	Set/display whether custom command is enabled. If defined but not checked, it is not executed.
7	Command name	Display and edit command name for selected command.
8	Function name	Display and edit function name for selected command.
9	Comment	Display and edit comment for selected command.
10	Busy On	Set/display whether to change to measurement stop state (MeasureStop) before executing command. If checked, BUSY is turned ON while command is executing, and then after execution of command has finished, measurement stop state is released (MeasureStart). Afterwards, a MeasureInit event is raised.
11	Export	Export the macro program to file.
12	Import	Import the macro program from file. The existing data will be overwritten.
13	OK	Save change and return to Main window.
14	Cancel	Return to Main window without saving.

Selection of the Communication Command Macro to be Used (Communication Command Macro)

Select the command to define in the [No.] column.
Only the command numbers that are selected are enabled.

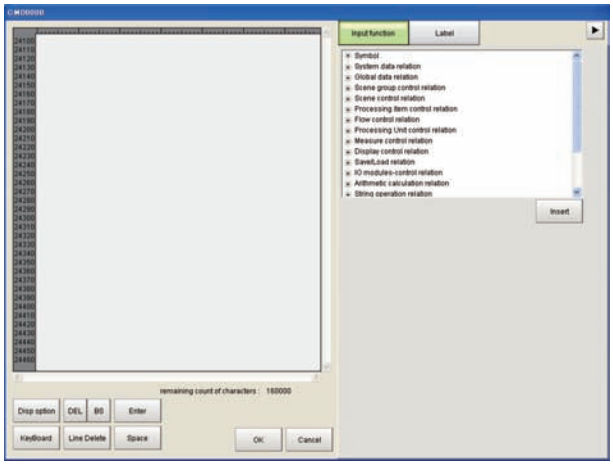
Communication Command Macro Name Setting (Communication Command Macro)

In the [Command name] field in the Settings Area, enter the command name to use with the non-procedure communications protocol.

Flow Signal Output Setting (Communication Command Macro)

Use the [BUSY ON] check box in Settings Area in the lower right corner to specify whether the BUSY flag should be turned ON (TRUE) or OFF (FALSE) when the communication command macro that is being set is in execution.

Program Input (Communication Command Macro)



The program input method is the same method as for the unit macro processing item.
Reference: ▶Program Input (Unit Macro) (p.200)

Specifying Communication Command Macro

The specified custom communications command is sent from an external device to the Sensor Controller.
The method that you use to specify a command depends on the communications protocol.

Non-procedure Protocol

Specify the name of the command as an ASCII character string.
Send the ASCII character string as *Command_name* + (space) + *Numeric_value*.
If you intend to use standard commands, do not use an existing command name.

Example: Specifying the Command Name “mycommand” with a Numeric Value of 18 as the Argument.

Send “mycommand 18”.

Using the Parallel Interface

Specify the command number from 0 to 255 as a binary number.
Specify the command number with the seven bits from DI0 to DI6.
Then, turn ON the Command Execution Bit, DI7. Because the command must be specified with seven bits, the method differs as shown below for command numbers 0 to 127 and command numbers 128 to 255.

- **For 0 to 127: Specify the number as is with the seven bits from DI0 to DI6.**

  Example: For command number 120, the binary notation for decimal 120 is 011 1000. Set each bit as follows: DI6: 0, DI5:1, DI4: 1, DI3: 1, DI2: 0, DI1:0, DI0: 0. Then, after 1 ms, change DI7 from OFF to ON to execute the command.

- **For 128 to 255: Use the terminal offset command DIOFFSET to add the difference from 0 to 127 with the seven bits from DI0 to DI6.**

  Example: For command number 150, the binary notation for 150 is 1001 0110, which requires eight bits. In this case, you use the terminal offset command DIOFFSET to add, for instance, half of 150, or 75. Then you can use DI0 to DI6 to specify the remaining 75. This procedure is given below.

  1. **Send "DIOFFSET 75."**
  2. **The binary notation for decimal 75 is 100 1011. Set each bit as follows: DI6: 1, DI5:0, DI4: 0, DI3: 1, DI2: 0, DI1:1, DI0: 1. Then, after 1 ms, change DI7 from OFF to ON to send the command.**
  3. **Send "DIOFFSET 0."**

     (0: OFF 1: ON)

- **Using PLC Link, EtherCAT, or EtherNet/IP (Except for the Non-procedure Protocol and Parallel Interface)**

  Specify the command number from 0 to 255 as a hexadecimal number from 00 to FF hex to represent the command code (CMD-CODE).

  Specify the command code in order from the upper digits to the lower digits of the hexadecimal number. This corresponds to the upper word address and the lower (smaller) word address in the I/O memory in the PLC. In this case, the highest number is FF hex, so we specify a number from 00 to FF hex to the first word in the Command Area + 2, and 0000 hex to the first word in the Command Area + 3.

  Example: For command number 120, the hexadecimal notation is 0078 hex. Specify 0078 hex in the first word in the Command Area + 2, and 0000 hex in the first word in the Command Area + 3.

Common Behavior of Custom Commands

Basic sequence

Normally, each of IO commands are processed in the sequence as below :

1. **Check input command and parameters are valid (range or type)**
2. **Body of the procedure**
3. **Output the result or response**

The way of input/output command, parameters and response depends on the type of IO modules.
For detail, please refer the pages shown below.

Reference: ▶Creating serial command (p.197)
Reference: ▶Creating Parallel Command (p.198)
Reference: ▶Creating PLC Link Command (p.198)
Reference: ▶Creating Fieldbus command (p.198)

Control BUSY signal

Basically the BUSY flag on Customize I/O command list window should be ON.

IMPORTANT

Executing measurement (Measure command) with BUSY flag set to ON causes error.
If you need to combine measure command with the command which is necessary to set BUSY ON (For example, switching scene and executing measurement), please set BUSY flag OFF and write the program like this way:

```
MeasureStop                    ' Set BUSY ON (Forbid measurement)
ChangeSceneArgumentValue#(0)   ' Execute the command which can be used with BUSY ON condition
MeasureStart                   ' Set BUSY OFF (Permit measurement) before measurement
Measure                        ' Execute measurement
```

Standard IO commands

When the same command name / command id of standard IO commands are used for customize IO commands, customize IO command has the priority and the standard IO commands are not executed.
If you execute the standard IO commands after execution of customize IO commands, please add the line as below.

```
CommandExecute&=False
```

In this case, standard IO commands is executed just after executing customize IO command.

Calling the procedure defined on the other commands

It is possible to call the procedure defined on the other commands during the command execution.
Each command has "Function name", and it is used to call the procedure.

Example: When command is defined as the table below and we intend to call procedure of CMD0 from CMD1,

Command No.	Command name	Busy	Function name
0	CMD0	False	FUNC0
1	CMD1	True	FUNC1

the codes for CMD1 should be like this:

```
Gosub *FUNC0
```

This case, the behavior of the command like BUSY depends on the caller (CMD1), and BUSY stays ON until the end of the procedure.

IMPORTANT

Please be careful not to make the commands calling each other (In the case above, CMD0 also calls FUNC1), because it makes infinite loop.

Define the different procedure according to I/O module

The variable IoIdent$ stores the IO module identification name, which received the current I/O command.
When you define the different procedure for each I/O modules, please make branch by the value of IoIdent$.

Example: The command which receives "Serial" for serial command, and "Ethernet" for UDP normal

```
If IoIdent$ = "SerialNormal" Then
    Response$ = "Serial"
Elseif IoIdent$ = "UdpNormal" Then
    Response$ = "Ethernet"
Endif
```

Creating serial command

Command parameters

Received text string is split by space character(" ") into command and parameters, and stored in the predefined variables shown below:

Variable name	Type	Content
ArgumentsLength&	Integer	Number of parameters (0 to 32)
ArgumentString$()	Array of text string	Array of parameters (string) Allocate a number of array elements equal to the number of parameters.
ArgumentValue#()	Array of real numbers	Array of parameters converted to number *1: If conversion fails, set to 0. Allocate a number of array elements equal to the number of parameters.

```
"AAA param0 param1 param2"
```

When the system received the string as above, parameters are set like this :

```
ArgumentsLength&:    3 (number of parameters)

ArgumentString$(0):  param0 (String type)
ArgumentString$(1):  param1 (String type)
ArgumentString$(2):  param2 (String type)

ArgumentValue#(0):   numeric value converted from param0 (0 when conversion failed)
ArgumentValue#(1):   numeric value converted from param1 (0 when conversion failed)
              (2):   numeric value converted from param2 (0 when conversion failed)
```

Example: The command "SC 1" that switches scene 1

```
ScaneChange ArgumentValue#(0)
```

Response output

Result of the command procedure can be returned to the system by setting the value on these variables.

Variable name	Type	Content
ResponseString$	Text string	Output data
ResponseCode&	Integer	Result of command • 0 : success (returns "OK") • non 0 : fail (returns "NG")

Example: The command "TEST"

```
ResponseString$ = "TestString"
```

Command and response will be like this :

```
-> TEST
<- TestString
<- OK
```

Creating Parallel Command

Response output

Result of the command procedure can be returned to the system by setting the value on these variables.

Variable name	Type	Content
ResponseCode&	Integer	Command execution result • 0 : Command processing successful • Other than 0 : Command processing failed (The ERROR signal turns on.)

Creating PLC Link Command

Command parameters

The command parameters are stored on the predefined variables as below.

Variable name	Type	Content
ArgumentsLength&	Integer	Number of parameters (0 to 6)
ArgumentValue#()	Array of real numbers	Array of parameters *1: integer type of data for 2 channels Allocate a number of array elements equal to the number of parameters.

Response output

Result of the command procedure can be returned to the system by setting the value on these variables.

Variable name	Type	Content
ResponseValue&()	Array of Integers	Output data
ResponseCode&	Array of real numbers	Command execution result • 0: Command processing successful • –1 : Command processing failed

Creating Fieldbus command

Command parameters

The command parameters are stored on the predefined variables as below.

Variable name	Type	Content
ArgumentsLength&	Integer	Number of parameters (0 to 3)
ArgumentValue#()	Array of real numbers	Array of parameters *1: integer type of data for 2 channels Allocate a number of array elements equal to the number of parameters.

Response output

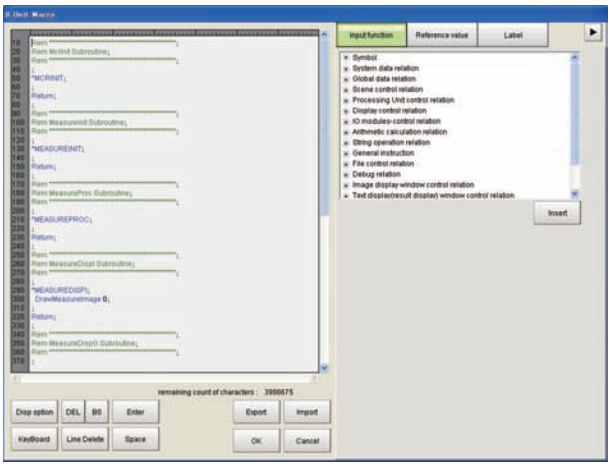
Result of the command procedure can be returned to the system by setting the value on these variables.

Variable name	Type	Content
ResponseValue&()	Array of Integers	Output data
ResponseCode&	Array of real numbers	Command execution result • 0 : Command processing successful • –1 : Command processing failed

Description of the Setting Screen of the "Unit Macro" Processing Item and How to Configure Settings

The properties screen of the "Unit Macro" processing item is the same as the program editing screen. For details on the setting screen, refer to Program Editing Screen.

Reference: ▶Components of the Program Editing Screen (p.181)

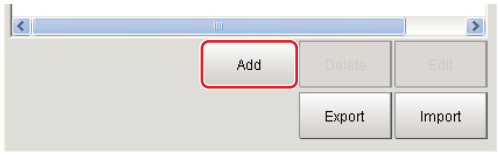


Reference Variable Settings (Unit Macro)

Set up the reference variables used for function.

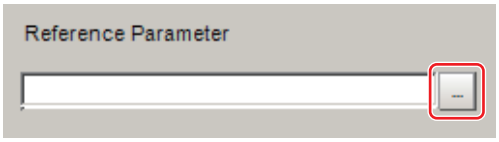
1 **Click [Add] in the reference variable list on the macro setting screen.**

A reference variable window is displayed.

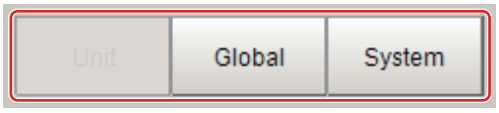


2 **Click [...] to set the variable name.**

The variable name must consist of alphanumeric characters beginning with a capital letter.

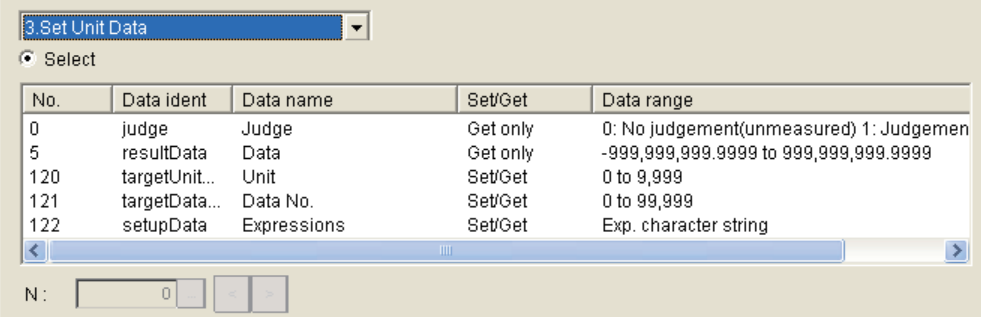


3 **Set the variable to be referenced.**



If Unit is selected

Select the processing item to be referenced, and then select the data to be referenced from the list.

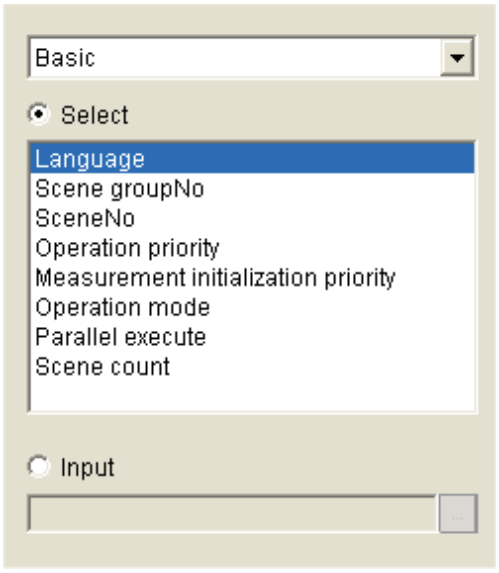


If Global is selected

Click [...] to set the variable value.

If System is selected

Place a check at applicable [Select] to select the type of system variable, and then select the variable to be referenced from the list.
To set a variable value, place a check at [Enter] and then click [...] to set the variable value.



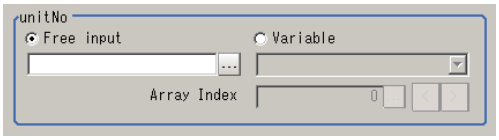
4 **Click [OK].**

Program Input (Unit Macro)

1 **From the function list, select the function to be inserted.**

When the function is selected, an operand list appears below the function list.

2 Set the operand.



Setting item	Setting value [Factory default]	Description
Operand input method	[Free input]	Select this option if you want to enter an operand freely. Click [...] to set the operand.
	Variable	Select this option if you want to select an operand from variables. Click [▼] to select the variable. You can select any variable or reference variable currently defined in the macro code.
Array Index	0 to number of arrangements [0]	If the selected variable is an arrangement variable, set the arrangement number to be used as an operand.

3 Click [Insert].

The set function is registered and appears in the code edit window in the top left-hand corner.

4 Repeat steps 3 to 6, and set the calculation processing.

5 When the setting of calculation processing is complete, click [OK].

Key Points for Test Measurement and Adjustment (Unit Macro)

To increase the speed and precision of measurement, you can adjust the parameters by performing and checking the results of test measurements.

The following content is displayed in the "Detail result" area as text.

Displayed items	Description
Judge	Judgement result

The image specified in the sub image in image display setting is displayed in the image display area.

Sub image.	Explanation of image to be displayed
0	Measurement image

Key Points for Adjustment

Select the adjustment method referring to the following points.

An error message appears on the console

Parameter to be adjusted	Troubleshooting
-	Refer to the error messages list. Reference: ▶"Appendixes About Macro Functions List of Macro Error Messages" in the "Vision System FH/FZ5 Series User's Manual (Z340)"

Nothing happens when [DEL], [BS], [Enter], etc., is clicked

Parameter to be adjusted	Troubleshooting
-	Nothing happens while the focus is not on the code window (key entry cursor is not displayed). Click the position you want to operate, and then click the button.

Want to include a line feed code in a string

Parameter to be adjusted	Troubleshooting
Macro code	Add (+) CR → Chr$(13) LF → Chr$(10) to the string.

The "Positions" display is not as desired

Parameter to be adjusted	Troubleshooting
Macro code	When "Positions" is OFF, the display processing written in "*MEASUREDISPT" and "*MEASUREDISPG" is executed. When "Positions" is ON, only what is written in "*MEASUREDISPG" is displayed.

Data acquisition sometimes fails during measurement

Parameter to be adjusted	Troubleshooting
Macro code	Data may be stolen by the applicable communication processing unless the communication processing is stopped using the SetPollingState function. (Example) Receive data without TCP procedure Write the processing in the following sequence. • SetPollingState "TcpNormal", false • Data receive processing • SetPollingState "TcpNormal", true

Check the measurement results that can be displayed and output in "Measurement Results For Which Output is Possible".

Reference: ▶External Reference Table (Unit Macro) (p.203)

Measurement Results For Which Output is Possible (Unit Macro)

To output the measurement result of the unit macro, assign the value of the unit macro calculation result to data output or the calculation processing item and output externally.

External Reference Table (Unit Macro)

The external reference table of the unit macro does not contain any data that can be referenced.

Saving and Loading Programs

Programs created using macro customize functions can be saved and loaded as scene data in the "Unit Macro" processing item and the "Unit Calculation Macro" processing item, and the settings can be saved in the sensor controller with "Data save". With the exception of some functions, standalone programs can be saved and loaded. A standalone program can be handled as a file, and thus programs can be managed and analyzed on a computer using a text editor.

Saving and Loading Programs in the Unit Calculation Macro

In the "Unit Calculation Macro" processing item, programs can be saved and loaded to the scene data as a processing unit, and in the properties screen of the processing item, programs can be saved as standalone programs. A program saved in the "Unit Calculation Macro" processing item can be loaded using the properties screen of the "Unit Macro" processing item.

Follow the procedure below to output the set calculation process to a file.

1 **Click [Export] in the file area.**

2 **Specify the file save location and the file name, and click [OK].**

IMPORTANT

- A standalone program cannot be loaded in the "Unit Calculation Macro" processing item. If loading is necessary, load scene data.
- The programs saved in the "Unit Calculation Macro" processing item include created programs and programs automatically generated by the processing item. Do not change an automatically generated program. If changed, the program may not load normally.

Saving and Loading Programs in the Scene Control Macro

In the scene control macro tool, a program can be saved when "Data save" is executed.

Saving and Loading Programs in the Communication Command Macro

In the communication command macro tool, a program can be saved when "Data save" is executed, and a standalone program can be saved and loaded in the tool setting screen.

- **Saving a program**

  1 **Click [Export] in the file area.**

  2 **Specify the file save location and the file name, and click [OK].**

- **Loading a program**

  1 **Click [Import] in the file area.**

  2 **Select the saved file (.mcr) that you want to load and click [OK].**

IMPORTANT

The programs saved in the communication command macro tool include created programs and programs automatically generated by the tool. Do not change an automatically generated program. If changed, the program may not load normally.

Saving and Loading Programs in the Unit Macro

In the "Unit Macro" processing item, programs can be saved and loaded to the scene data as a processing unit, and in the properties screen of the processing item, standalone programs can be saved and loaded.

- **Saving a program**

  1 **Click [Export] in the file area.**

  2 **Specify the file save location and the file name, and click [OK].**

- **Loading a program**

  1 **Click [Import] in the file area.**

  2 **Select the saved file (.mcr) that you want to load and click [OK].**

Basic Method for Writing Programs

Necessity of referring to this manual	Unit Macro	Communication Command Macro	Scene Control Macro	Unit Calculation Macro
	Required			

The macro customize functions of the FH/FZ5 series use an interpreter-type programming language. Unlike a compiler-type language, programs can be created in an interpreter programming language without the need for a special development environment to compile the program. In contrast to programming languages such as C, you can create and run programs on the FH/FZ5 without a development environment.
The programming language that is used for the macro customize functions is based on BASIC, which is also an interpreter-type programming language. The programming syntax of the macro customize functions is based on the syntax of BASIC, with the addition of some specialized syntax.
The syntax required to create programs for the macro customize functions is explained in the following, starting from basic syntax.

Basic Syntax

There are some rules which must be known as a minimum in order to create programs.
The examples below include the basic program syntax that is used in the macro customize functions.

Basic syntax examples

```
*MEASUREDISPT

  Rem Displays text in the Detail result pane.
  DrawText "Judgment OK", 1, 0

Return
```

Types of syntax	Description
Character	Alphabetical characters, numbers, symbols, and special marks can be used as characters. Reference: ▶Character (p.206)
Line	A unit that is composed of a line number and a statement is called a line. Reference: ▶Line (p.206)
Line number	The number that is automatically assigned to each line when a program is loaded in the FH/FZ5 system is called a line number. Reference: ▶Line number (p.206)
Statement	The program processing text written on each line is called a statement. Reference: ▶Statement (p.207)
Label	A name assigned to a line in the program to enable identification is called a label. Reference: ▶Label (p.207)
Subroutine	A part of a program that is enclosed by <label> - Return is called a subroutine. Reference: ▶Subroutine (p.207)
Comment	Text that follows a Rem function or a single quotation is called a comment. Reference: ▶Comment (p.207)

Character

Alphabetical characters, numbers, symbols, and the special marks below can be used as characters. Lower case alphabetical characters are only recognized as lower case when enclosed by double quotation marks ("). Lower case and upper case are not distinguished anywhere else. With the exception of text enclosed by double quotations (") and comments described below, use only half-width alphanumeric characters and half-width symbols.

Special marks that can be used	Description
Colon (:)	Use as a separator when writing two or more lines as a single line.
Comma (,)	Use as a separator when listing parameters.
Semicolon (;)	Use as a separator when listing parameters in output text.
Apostrophe (')	Add in front of a comment. A Rem statement can also be used to indicate a comment.
Asterisk (*)	Add in front of a label name.
Space ( )	Always insert a space between a macro function and its arguments. Spaces can also be inserted wherever desired. However, a space must not be inserted inside a macro function name, variable name, or value.
Double Quotation Marks (")	Use to enclose a character string value.
Ampersand (&)	Use as a type identifier for integer type variables. Always add after an integer type variable name or array name. For details on type identifiers, refer to the "Naming Rules for Variables" section. Reference: ▶Naming Rules for Variables (p.210)
Pound Sign (#)	Use as a type identifier for double precision real number variables. Always add after a double precision real number variable name or array name. For details on type identifiers, refer to the "Naming Rules for Variables" section. Reference: ▶Naming Rules for Variables (p.210)
Dollar Mark ($)	Use as a type identifier for character string type variables. Always add after a character string type variable name or array name. For details on type identifiers, refer to the "Naming Rules for Variables" section. Reference: ▶Naming Rules for Variables (p.210)
"At" mark (@)	Use as a type identifier for reference variables. Always add at the end of a reference variable name. For details on type identifiers, refer to the "Naming Rules for Variables" section. Reference: ▶Naming Rules for Variables (p.210)

Line

A unit that is composed of a line number and a statement is called a line. Blank lines and lines that are only comments are acceptable in a program. Multiple statements (multi-statement) can also be written on one line. When writing multiple statements, separate each statement with a colon (:).

Line number

The number that is automatically assigned to each line when a program is loaded in the FH/FZ5 system is called a line number. Line numbers are mainly used during program debugging to identify the locations of errors. Do not write line numbers when creating a program.

Statement

The program processing text written on each line is called a statement. A statement is mainly composed of expressions of minimum processing units. A statement can be up to 245 characters in length. If a statement is longer than 245 characters, an error will occur and program execution will stop.
Three types of statements exist, as shown below.

Statement type	Description
Execution statement	A statement that describes and executes processing of a macro function.
Non-execution statement	A statement that indicates comments but does not execute processing.
Label	A statement that defines a program branch destination.

Label

A name assigned to a line in the program to enable identification is called a label. Among the basic syntax examples, the following is a label.

```
*MEASUREDISPT
```

By specifying a label, the "Goto <label>" macro function can be used to jump to the specified label position.

Subroutine

A part of a program that is enclosed by <label> - Return is called a subroutine. Write the process that you want to execute in the subroutine. Among the basic syntax examples, the following is a subroutine.

```
*MEASUREDISPT

Return
```

Subroutines can be executed by calling the "GoSub <label>" function. By writing each standalone process as a subroutine, the visibility of the program is improved.
The processing component of a macro customize function consists of subroutines. By writing "GoSub*(function name)", the processing component of the corresponding macro customize function can be called and executed from another subroutine.
Some of the subroutines of macro customize functions are predefined in the system. The timing at which system defined subroutines are called is fixed. Write the process in a subroutine appropriately for its purpose. For details on calling these subroutines, refer to "Status Transitions and Execution Timing".
Reference: ▶State Transitions and Execution Timing (p.227)

Comment

Text that follows a Rem function or a single quotation is called a comment. You can insert any comment line in a program. When a Rem function or single quotation (') is written in a program, the line is treated as a comment.
Among the basic syntax examples, the following is a comment.

```
Rem Displays text in the text result field.
```

IMPORTANT

Do not mix a non-comment statement together with a comment on the same line. The comment may not be correctly recognized and the program may not operate correctly.

Constant

Among the values and character strings used in programming, a constant is a value that never changes and has a unique assigned name.
Use constants for fixed numeric values and character strings that you want to use repeatedly in a program.

How to Use Constants

Constants are used as shown below.
(Example)

```
A& = 255
AA& = &h7f

B# = 3.14

C$ = "TEST STRING"
```

Constant Data Types

Constants that can be used in macro customize functions are shown below.

Constant type	Description	Data range	Number of bytes per data item
Integer	Used for signed integer values.	-2147483648 to 2147483647	4 bytes
Double precision real number	Used for double precision type real numbers.	-1.0E30 to 1.0E30	8 bytes
Character string	Used for character strings.	Up to 255 characters	Max. 256 bytes

Integer constants can be written in several bases, including decimal. Base notations that can be used in macro customize functions are shown below.

Base	Notation method	Example	Mathematical notation
Decimal	Not required	100, 3456	100, 3456
Hexadecimal	&h	&hff, &h7fff	$(ff)_{16}$, $(7fff)_{16}$
Octal	&o	&o77, &o3447	$(77)_{8}$, $(3447)_{8}$
Binary	&b	&b1111, &b01100111	$(1111)_{2}$, $(01100111)_{2}$

Note

When a program that uses hexadecimal constants is displayed using the List function, the hexadecimal notation is converted to decimal notation.
Example: When a program with "A& = &hff" is displayed using the List function, this is shown as "A& = 255".

Variable

Among the values and character strings used in programming, a variable has a unique assigned name and is used for data that changes. Use a variable for a numeric value or character string that you want to use repeatedly in a program, but whose specific value or character string changes.

How to Use Variables

Variables are used as shown below.

(Example)

```
Rem 1. Declare the variable
Dim POSITION#(1)

Rem 2. Assign a value to the variable
A# = 320.0
B# = 310.0

Rem 3. Reference the variable
POSITION#(0) = A#
POSITION#(1) = B#
C# = POSITION#(0) + POSITION#(1)
```

1. **Declare the variable**

   By declaring a variable, you allocate the data area required for the variable and enable use of the variable in the remainder of the program. Declaration is only required for array variables, and the Dim function is used for this purpose. For variables other than array variables, the data area is automatically allocated when the program is executed, and thus declaration is not necessary.

2. **Assign a value to the variable**

   Assign a value to a variable. Only values of the same data type as the variable can be assigned. In the case of a variable other than an array variable, the variable can be used without being declared. Add the type identifier to the end of the name, and use a variable name that is not used by any other variables. For details on type identifiers, refer to the "Naming Rules for Variables" section.

   Reference: ▶Naming Rules for Variables (p.210)

3. **Reference the variable**

   Reference the value that is assigned to a variable. You can also pass the variable to a macro function as an argument.

Variable Data Types

Before a variable can be used, the data area that will hold the data must be allocated. A data type defines the size of the data area to be allocated and how the data is handled.

Data types and data ranges of variables that can be used in macro customize functions are described below. Select the data type based on the use and objective of the variable.

Data type	Description	Data range	Number of bytes per data item	Variable area size
Integer	Used for signed integer values.	-2147483648 to 2147483647	4 bytes	Variable length
Double precision real number	Used for double precision type real numbers.	-1.0E30 to 1.0E30	8 bytes	
Character string	Used for character strings.	Up to 255 characters	Max. 256 bytes	

Naming Rules for Variables

Rules exist for variable names. Decide variable names based on the rules below.

Variable name		
Start	**Middle**	**End**
Alphabet ('A' to 'Z', 'a' to 'z')	Alphabet, numbers, Symbols ('A' to 'Z', 'a' to 'z', '0' to '9', underline '_', period '.'))	Type identifier ('&', '#', '$', '@')(*1)

*1: "@" is a type identifier for reference variables, and cannot be used for variables that the user directly declares in the program.

An identifier that identifies the type of data used as a variable is referred to as a "type identifier". Always add a type identifier to the end of the variable name of each variable. For the type identifiers that can be used, refer to "Variable Types and Type Identifiers".

IMPORTANT

- A variable name that starts with a number cannot be used.
- A special mark cannot be used in a variable name.
- A variable name that is reserved cannot be used. However, a variable name that includes a reserved variable name can be used.
  Example: (Cannot be used) WAIT& → (Can be used) WAITTIME&
- Letters are not case sensitive.
  Example: AA& and aa& are treated as the same variable.

Variable Types and Type Identifiers

Variable types that can be used in macro customize functions are shown below. Use a variable type that matches your use and purpose.

- **Temporary variable**

  This can handle one datum and is the most basic variable.
  Type identifiers that can be used with temporary variables are shown below.

Temporary variable data type	Type identifier	Example
Integer	&	A& = 1
Double precision real number	#	A# = 12.34
Character string	$	AA$ = ”OMRON”

IMPORTANT

The type identifier cannot be omitted. Always add a type identifier to the end of the variable name.

- **Array Variable**

  When you want to handle multiple data items of the same data type as a group, you can assign numbers to temporary variables. Such temporary variables are called array variables. Macro customize functions enable the use of up to four dimensions of array variables. The element number of an array starts from 0, and the number of elements is "element number + 1".

  Example

```
Rem One-dimensional array with four elements A&(0), A&(1), A&(2), A&(3)
Dim A&(3)

Rem Two-dimensional array with 11 x 11 = 121 elements
Dim B&(10, 10)

Rem Three-dimensional array with 101 x 101 x 101 = 1030301 elements
Dim C&(100, 100, 100)
```

  Type identifiers that can be used with array variables are shown below.

Array variable data type	Type identifier	Example
Integer	&()	A&(1) = 1
Double precision real number	#()	A#(2) = 12.34
Character string	$()	AA$(3) = ”OMRON”

IMPORTANT

- Even if the name is the same, an array variable name and a regular temporary variable name are distinguished.

Example: The variables below are all distinguished.
A&, A&(10), A#(10), A$, A$(10)

- Even if the number of dimensions is different, two arrays with the same variable name are treated as the same array. In this case, the array defined last is effective.

Example: The variables below are all distinguished.
A&(10), A&(10,10), A&(10,10,10), A&(10,10,10,10)

- **Reference variables**

  Reference variables are defined by the user and can be used to reference processing unit data, global data, and system data.
  (Note: When using the macro customize functions, a macro function can be used to set and acquire data such as processing unit data and system data. Reference variables allow you to handle processing unit data and system data without using the macro function.)
  Reference variables must be set in advance in the program editing screen of the "Unit Macro" processing item, "Unit Calculation Macro" processing item, or the scene control macro tool. For the setting method, refer to the setting screen of each processing item.
  Type identifiers that can be used with reference variables are shown below.

Reference variable data type	Type identifier	Example
Integer	@	A@ = 1
Double precision real number	@	A@ = 12.34
Character string	@	AA@ = ”OMRON”

IMPORTANT

- The data type (integer, real number, etc.) can be identified from the type identifier of a regular variable, however, identification of the data type from the type identifier is not possible with a reference variable. When using a reference variable, check the data type of the allocated data before using the variable.
- A reference variable setting is saved in scene data, however, the value itself that the reference variable references is not saved. If you want to save the value, use user data.

Note

- If you set the name of a reference variable in the properties screen the type identifier of the reference variable is automatically added to the reference variable name.
- There is no limit on the number of reference variable settings that can be added. However, there is a limit on the total number of variable name characters per processing unit of the unit macro and unit calculation macro, and the number of variable name characters per scene of the scene control macro. When the reference variable name is 32 characters, about 1000 reference variable settings can be added.

Operator

A symbol that indicates an operation in a program is referred to as an operator. Operators are used to add a process to a variable, as well as to calculate and compare variables and constants.

How to Use Operators

Operators are used as shown below.

(Example)

```
A& = 0                          1
B& = 2
C& = 4
D& = 35
E& = 0

A& = 1 + 2 + 4 / 2              2

F& = (A& + B&) * C&             5
                                3
If F& < D& Then
   E& = D& AND 31               4
Else
   E& = F& AND 31
EndIf
```

1. Assignment Operator

An assignment operator is used to assign a value to a variable.
Assignment operators that can be used in macro customize functions are shown below.

Operator	Description of operation	Example
=	Assigns the value on the right side to the left side	A& = B& + C&

Note

- When "=" is used in a conditional comparison such as an "If - Then" or "Select" statement, "=" is treated as a relational operator that determines whether the left side and right side are equal.
- When a double precision real number variable value is assigned to an integer variable, the digits to the right of the decimal point are rounded off.
- If the value assigned to an integer variable is other than -2147483648 to 2147483647, an overflow error will occur when the assignment takes place.
- If the value assigned to a double precision real number variable is other than -1.0E30 to 1.0E30, an overflow error will occur when the assignment takes place.

2. Arithmetic Operator

An arithmetic operator performs an arithmetic operation, exponent operation, or remainder operation on numerical value data, Division by 0 results in an error. If the interim result of an arithmetic operation such as addition, subtraction, or multiplication is outside the range -1.0e30 to 1.0e30, an error will result.
Arithmetic operators that can be used in macro customize functions are shown below.

Operator	Description of operation	Example	Mathematical notation
+	Addition	A& + B&	A+B
-	Subtraction	A& - B&	A-B
*	Multiplication	A& * B&	AxB or AB
/	Division	A& / B&	A/B
^	Exponent operation	A& ^ B&	A^B
mod	Remainder	A& MOD B&	A-[A/B]Å~B [] is the Gauss symbol

3. Relational Operator

A relational operator compares two numerical data items or two character data items. If the result of the comparison is true, (-1) is returned. If false, (0) is returned. Normally this is used in an "If - Then" statement for such purposes as controlling the flow of the program.
Relational operators that can be used in macro customize functions are shown below.

Operator	Description of operation	Example
=	Equal	A& = B&
<>, ><	Not equal	A& <> B&, A& >< B&
<	Less than	A& < B&
>	Greater than	A& > B&
<=, =<	Less than or equal to	A& <= B&, A& =< B&
>=, =>	Greater than or equal to	A& >= B&, A& => B&

Note

When "=" is used in other than a conditional comparison such as an "If - Then" or "Select" statement, "=" is treated as a assignment operator that assigns the value on the right side to the left side.

4. Logic Operator

A logic operator is used to investigate multiple conditions, and perform bit operations and binary operations on exponential values. Logic operators that can be used in macro customize functions are shown below.

Operator	Description of operation	Example
NOT	Not	NOT A&
AND	Logical AND	A& AND B&
OR	Logical OR	A& OR B&
XOR	Exclusive OR	A& XOR B&

Operation Order of Operators

When multiple operators are included in one expression, operations are executed in the order of priority of the operators. If you want to control the order of the operations, enclose the operations you want to perform first in parentheses.

The order of priority of the operators is shown below.

Order of priority	Operator
1	Operation enclosed in parentheses
2	Macro function
3	Exponent operation (^)
4	Minus sign (-)
5	Multiplication and division (*, /)
6	Remainder (mod)
7	Addition and subtraction (+, -)
8	Relational operators (< >, =, etc.)
9	NOT
10	AND
11	OR
12	XOR

Expression

Constants, variables, and the operators that join them are referred to as an expression. Not only joined constants and variables, but the constants and variables themselves are expressions, and expressions and combinations of expressions form a statement.

Numerical expression

An expression that returns a numerical value is referred to as a numerical expression. This joins numerical constants, numerical variables, and macro functions that return numerical values with arithmetical operators and logic operators. Multiple numerical expressions can be joined by enclosing the expressions in parentheses.

(Example of a numerical expression)

```
A& = 1 + 2 + 4 / 2
```

Character expression

An expression that returns a character string is referred to as a character expression. This joins character string constants, character string variables, and macro functions that return character strings with plus signs. Multiple character expressions can be joined by enclosing the expressions in parentheses.

(Example of a character expression)

```
B$ = "OMRON" + "FH"
```

Relational expression

An expression that joins two numerical expressions by a relational operator is referred to as a relational expression.

(Example of a relational expression)

```
If A& < 10 Then

EndIf
```

Logical Expression

An expression that joins multiple relational expressions by a logical operator is referred to as a logical expression. This is used to execute bit operations and binary operations, and to evaluate complex conditions. When the operation result of a logical expression is other than 0, the result is treated as true, and when 0, the result is treated as false. Because a logical expression returns a numerical value, it can also be treated as a numerical expression.

(Example of a logical expression)

```
A& = D& AND &b110000
```

Function

An expression that executes a predefined command or operation, or an expression that executes a predefined operation on a specific specified value (argument) and returns the result of the operation, is generally referred to as a function. In particular, the functions that can be used in the macro customize functions are referred to as macro functions. Macro functions include functions that do not return a value, and functions that return the numerical value or character string that is the result of the operation.
Macro functions that do not return a value are written with the macro function name and argument separated by a space. Macro functions that return a value add an argument enclosed by parentheses ( ) to the end of the macro function name.

(Example of a macro function that does not return a value)

```
ChangeScene 1
```

(Example of a macro function that returns a value)

```
C# = Abs(-10)
```

How to Write Advanced Programs

Macro customize functions can be used for a variety of purposes depending on the content being programmed. This section explains programming techniques that can be used together with the basic program writing methods to widen the range of application of the macro customize functions.

Data Types Related to Processing Units

Necessity of referring to this manual	Unit Macro	Communication Command Macro	Scene Control Macro	Unit Calculation Macro
	As needed			

A macro customize function program can be used to change measurement flow settings by setting or acquiring processing unit data such as external reference data, figure data, and model data. The data access method varies depending on the data type.

Type	Description
External reference data	Data used to set or acquire settings and measurement values of a processing unit. Reference: ▶External reference data (p.217)
Figure data	Data used to set and acquire region figures and model figures of a processing unit. Reference: ▶Figure data (p.218)
Model data	Data used to set the model registration of a processing unit. Reference: ▶Model data (p.220)
Image data	Image data held by the processing unit itself. Reference: ▶Image data (p.221)

External reference data

External reference data is used to set and acquire settings and measurement values of a processing unit. Unique numbers are assigned to the external reference data within the processing item, and thus data can be set and acquired by specifying the processing unit number and external reference data number.
In addition to a macro customize program, external reference data can be set and acquired from a processing unit data setting item, processing unit data acquisition item, and a communication commands.
A processing unit data setting macro function or processing unit data acquisition macro function is used to set or acquire external reference data from a macro customize program. Numerical data and character string data can be set and acquired in a processing unit data setting macro function or processing unit data acquisition macro function. However, figure data such as measurement region data, model data such as search model and model edge data, and image data such as camera images and measurement filter images cannot be set or acquired by this method. Refer to the figure data, model data, and image data sections.

- **Acquisition of Data**

  External reference data is acquired by the methods below.

  Example: Acquiring External Reference Data No. 5 (correlation value) of the Unit No. 1 Search processing unit.

```
Rem The correlation value is a real number, and thus the variable type identifier is #
GetUnitData 1, 5, DATA#
```

  Example: Acquiring External Reference Data No. 7 (decode character string) of the Unit No. 2D code code processing unit.

```
Rem The decode character string is a character string, and thus the variable type identifier is $
GetUnitData 2, 7, DATA$
```

- **Data Settings**

  The method of setting external reference data is as follows.

  Example: Setting 1 in External Reference Data No. 147 (search number) of the Unit No. 1 search processing unit

```
SetUnitData 1, 147, 1
```

  Example: Setting "comparison" in External Reference Data No. 300 (index 0 comparison character string) of the Unit No. 2 two-dimensional code processing unit.

```
SetUnitData 2, 300, "comparison"
```

  In addition to the external reference data number, external reference data can also be set and acquired using the external reference data identification name.

  Example: Setting "1" in the external reference data ID name "searchNo" (search number) of the Unit No. 1 search processing unit.
  The result is the same as when "1" is set in External Reference Data No. 147.

```
SetUnitData, "searchNo", 1
```

Figure data

Figure data is used to set and acquire region figures and model figures of a processing unit. When there are multiple figure data items in the processing item, a unique number is assigned to each figure data item. This makes it possible to set and acquire figure data by specifying the processing unit number and figure number. In addition to a macro customize program, figure data can be set and acquired from a processing unit figure setting item and a processing unit figure acquisition item.
A processing unit figure setting macro function or processing unit figure acquisition macro function is used to set or acquire figure data from a macro customize program.
Reference: ▶List of Figure Numbers (p.366)

- **Format of Figure Data**

Figure data is specified using an array variable. The elements of the array are described below.

Array element	Description	Description
figure(0)	Figure data header information	This is figure data header information. Includes the number of figures and figure data size information. Upper 16 bits: Number of figures Lower 16 bits: Number of bytes of figure data size (figure array length x 4) Figure data header information = Number of bytes of figure data size + Number of figure data x 65536 Number of figures:Sets the number of figures included in the figure data. Normally 1 should be set. If you are combining multiple figures, set the number of figures that are combined (2 or more). Number of bytes of figure data size: Set the figure data size converted into bytes. Set the value that is 4 times the number of array elements. Example: One rectangle Array length = 5, Number of figure data items = 1 Number of bytes of figure data size = 5 x 4 = 20 Figure data header information = 20 + 1 x 65536 = 65556
figure(1)	Figure 0 type information	Type information of figure 0 data. Includes drawing mode and figure type information. Upper 16 bits: Drawing mode Lower 16 bits: Figure type Figure type information = Figure type + Drawing mode x 65536 Drawing mode:Set whether the figure drawing mode is OR mode or NOTE mode. Normally 0 (OR mode) should be set. When multiple figures are used and you want to exclude some of the figures, set 1 (NOT mode) for the 2nd or later figures. Figure type:Set the figure type (line, rectangle, etc.). Example: One rectangle (drawing mode is OR) Figure type = 8, Drawing mode = 0 Figure type information = 8 + 0 x 65536 = 8
figure(2)	Figure 0 data	Figure data of figure 0. The size and content depends on the figure type.
figure(M)	Figure 1 type information	Type information of figure 1 data.
figure(M+1)	Figure 1 data	Figure data of figure 1. The size and content depends on the figure type.
⋮	⋮	⋮
figure(N*M)	Figure N type information	Type information of figure N data.
figure(N*M+1)	Figure N data	Figure data of figure N. The size and content depends on the figure type.

- **Acquisition of Data**

Figure data is acquired by the method below.

Example: Acquiring Figure Data No. 1 (region figure) of the Unit No. 1 Search processing unit

```
Dim FIGURE&(5)
GetUnitFigure 1, 1, FIGURE&()
```

Note

If the number of array elements is insufficient for the figure data to be set or acquired, an error will occur when setting or acquiring is attempted.

- **Data Settings**

  The method of setting figure data is as follows.

  Example: Changing the upper left coordinates of Figure Data No. 1 (region figure) of the Unit No. 1 Search processing unit to (100, 50).

```
Dim FIGURE&10
GetUnitFigure 1, 1, FIGURE&()

FIGURE&(2) = 100
FIGURE&(3) = 50

SetUnitFigure 1, 1, FIGURE&()
```

  Example: Changing the ellipse x direction radius of Figure 1 of Figure Data No. 0 (region figure (Figure 0: rectangle, Figure 1: Ellipse)) of the Unit No. 2 defect processing unit to 100

```
Dim FIGURE&(10)
GetUnitFigure 2, 0, FIGURE&()

Rem FIGURE&(0): Figure data header information
Rem FIGURE&(1): Figure 0 type information
Rem FIGURE&(2) to FIGURE&(5): Upper left XY coordinates and lower right XY coordinates of rectangle
Rem FIGURE&(6): Figure 1 type information
Rem FIGURE&(7) to FIGURE&(8): XY coordinates of center of ellipse
Rem FIGURE&(9) to FIGURE&(10): X direction radius and Y direction radius of ellipse
FIGURE&(9) = 100

SetUnitFigure 2, 0, FIGURE&()
```

Model data

Model data is used to set the model registration of a processing unit. When there are multiple model data items in the processing item, a unique number is assigned to each model data item. This makes it possible to set model data by specifying the processing unit number and model number.
In addition to a macro customize program, model data can also be set by communication commands.
A processing unit figure setting macro function or model copy macro function is used to set model data from a macro customize program.
Reference: ▶Model Number List (p.369)

IMPORTANT

Model data has a close association with the model figure and model parameters. Before re-registering a model, set the model figure and model parameters.

- **Acquisition of Data**

  Execution of only model data acquisition is not possible.

- **Data Settings**

  The data setting method is as follows.

  Example: Re-registering the model of the Unit No. 1 Search processing unit (model figure: rectangle) in the current measurement image, and changing the reference position (0, 0) and detection point coordinates (320, 240).

```
Dim FIGURE&(5)
GetUnitFigure 1, 0, FIGURE&()

Rem Re-register the model by setting the model figure
SetUnitFigure 1, 0, FIGURE&()

Rem Update reference coordinates XY
SetUnitData 1, 129, 0
SetUnitData 1, 130, 0

Rem Update detection point coordinates XY
SetUnitData 1, 132, 320
SetUnitData 1, 133, 240
```

Image data

Image data is data that is held by the processing unit itself.
When there are multiple image data items in the processing item, a unique number is assigned to each image data item. This makes it possible to set image data by specifying the processing unit number and image number.
Image data can be set from a macro customize program.
An image data setting macro function is used to set image data from a macro customize program.
Reference: ▶Image Number List (p.371)

- **Image Data Types**

  There are two types of image data.

Type	Image number	Description
Measurement image data	0 to 3	Image data that the processing unit registered in the measurement flow processes at the time of measurement. Normally the image is set in measurement image 0 at the time of measurement for "image input related" processing items and "image conversion related" processing items such as camera image input and filtering, and image processing is performed using images set in other processing items. Normally 0 should be used.
Processing unit image data	0 to max. 31	Image data that is held by each processing item separately from the measurement image data. The number of processing unit image data items and content varies by processing item. Use when you want to retain image processing results such as binarized images and filtered images.

- **Acquisition of Data**

  Execution of only image data acquisition is not possible.

- **Data Settings**
  The image data setting method is as follows.

  Example: Setting Image Data No. 0 (filtered image) of the Unit No. 2 filtering processing unit in Filtered Image No. 0, and enabling use as a measurement image by following processing units

```
SetMeasureImage 0, 2, 0
```

Data Types Related to the System

Necessity of referring to this manual	Unit Macro	Communication Command Macro	Scene Control Macro	Unit Calculation Macro
	As needed			

With a macro customize program, data that does not depend on the measurement flow can be set and acquired by setting or acquiring data held by the system such as user data, global data, and system data.
The data access method varies depending on the data type.

Type	Description
Global Data	Data that can be set or acquired from a different scene using a macro function. Use this when you want to set or acquire a temporary numerical value or character string that does not need to be saved, such as an interim result of an operation. Reference: ▶Global Data (p.222)
User Data	Data that can be set or acquired from a different scene using a user data processing item. Use this when you want to set or acquire numerical values that you want to be retained even if a power interruption occurs or restart is performed. Reference: ▶User Data (p.223)
System data	Data that can be set or acquired from a different scene using a macro function. Use this when you want to set or acquire numerical values or character strings that you want to be retained even if a power interruption occurs or restart is performed. Reference: ▶System data (p.224)

Global Data

Global data is data that can be set or acquired from a different scene using a macro function. Use this when you want to set or acquire a temporary numerical value or character string that does not need to be saved, such as an interim result of an operation, as global data.(Reference: ▶Scope of Data and Save Area (p.225))
Use system data when you need to save data that you want to set or acquire from a different scene.
Reference: ▶System data (p.224)

To use new global data, first specify a data identification name and register the global data. Half-width alphabetical characters and the half-width marks "." and "_" can be used for the global data identification name.
Reference: ▶Macro Reference (p.320)

- **Acquisition of Data**

  Global data is acquired by the method below.

  Example: Acquiring the data identification name "GData" value

```
Rem Register the default value 10 for the global data whose data identification name is "GData"
AddGlobalData "GData", 10

Rem Acquire global data with the data identification name "GData"
GetGlobalData "GData", GDATA&
```

- **Data Settings**

  The global data setting method is as follows.

  Example: Setting the data identification name "GData" value

```
Rem Register the default value 10 for the global data whose data identification name is "GData"
AddGlobalData "GData", 10

Rem Set 15 in the global data with the data identification name "GData"
SetGlobalData "GData", 15
```

User Data

User data is data that can be set or acquired from a different scene using a user data processing item. Use user data to set or acquire numerical values that you want to be retained even if a power interruption occurs or restart is performed. (Reference: ▶Scope of Data and Save Area (p.225))
When there is no need to save data that you want to set or acquire from a different scene, use global data.
When you want to set or acquire not only numerical values but character strings as well, use system data.
Reference: ▶Global Data (p.222)
Reference: ▶System data (p.224)

Before using user data, you can set the default values of the user data with the user data tool.
Reference: ▶ "User Data Tool" in the “Vision System FH/FZ5 Series User's Manual (Cat. No. Z340)”.

- **Acquisition of Data**

  The user data processing item is used to acquire user data.
  Reference: ▶"User Data" in the “Vision System FH/FZ5 Series Processing Item Function Reference Manual (Cat. No. Z341)"

- **Data Settings**

  A user data processing item is used to set user data.
  Reference:▶"User Data" in the “Vision System FH/FZ5 Series Processing Item Function Reference Manual (Cat. No. Z341)"

System data

System data is data that can be set or acquired from a different scene using a macro function. Use system data to set or acquire numerical values or character strings that you want to be retained even if a power interruption occurs or restart is performed. (Reference: ▶Scope of Data and Save Area (p.225))
Use global data when there is no need to save data that you want to set or acquire from a different scene.
When you only want to set or acquire numerical values, use user data.
Reference: ▶User Data (p.223)
Reference: ▶Global Data (p.222)

To use new system data, first specify a data identification name and register the system data that you want to use. Data identification names for system data are Data ID Name 0 and Data ID Name 1. The data identification names that correspond to ID Information 0 and ID Information 1 must be specified as arguments of the macro function.
For details on identification information and data identification names, refer to the system data list.
Reference: ▶Macro Reference (p.320)
Reference: ▶System Data List (p.327)

- **Acquisition of Data**

  System data is acquired by the method below.

  Example: Acquiring the value of "Initial Scene No." in the system data

```
Rem Acquire the system data whose Data ID Name 0 is "Configuration" and whose Data ID Name 1 is "initialSceneNo"
GetSystemData "Configuration", "initialSceneNo", SDATA&
```

- **Data Settings**

  The system data setting method is as follows.

  Example: Setting the values of Data ID Name 0 "PanDA" and Data ID Name 1 "SData"

```
Rem Register 0 as the default value of the system data whose Data ID Name 0 is "PanDA" and whose Data ID Name 1 is "SData"
AddSystemData "PanDA", "SData", 0

Rem Set 5 in the system data whose Data ID Name 0 is "PanDA" and whose Data ID Name 1 is "SData"
SetSystemData "PanDA", "SData", 5
```

IMPORTANT

When adding new data to the system data, specify "PanDA" in Data ID Name 0.

Scope of Data and Save Area

Necessity of referring to this manual	Unit Macro	Communication Command Macro	Scene Control Macro	Unit Calculation Macro
	As needed			

In macro customize functions, many types of data can be used as needed for the objective, and variables and macro functions can be used to set and acquire data.
The areas where data settings and acquisition can be executed and the areas where data are saved vary depending on the type of data.
By using data types appropriately for the objective of the program, macro customize functions enable the creation of programs that are easy to change and maintain.

Data Scope

The locations from which data can be set and acquired are limited; data cannot be set or acquired from any location. This limited location is called a "scope", and data can only be set and acquired from within the scope defined for that data type.
Data scopes that can be used with the macro customize functions are as follows.

Scope	Description
Within a processing unit Within a communication command macro	Only enabled within one processing unit. If multiple processing units of the "Unit Macro" processing item or the "Unit Calculation Macro" processing item are included in the measurement flow, variables with the same name that exist in the processing units will be treated as separate variables. If multiple communication command macros are defined in the communication command macro tool, variables with the same name that exist in the communication command macros will be treated as separate variables.
Within a scene	Enabled within one scene. If multiple processing units of the "Unit Macro" processing item or the "Unit Calculation Macro" processing item are included in the measurement flow, the same data can be set and acquired from each processing unit. The data cannot be set or acquired from the processing unit of a different scene.
Within the system	Enabled with the system. If multiple processing units of the "Unit Macro" processing item or the "Unit Calculation Macro" processing item are included in the measurement flow, the same data can be set and acquired from each processing unit. If multiple communication command macros are defined in the communication command macro tool, the same data can be set and acquired from each communication command macro. The data can be set and acquired from all scene groups and scenes.

Data Save Area

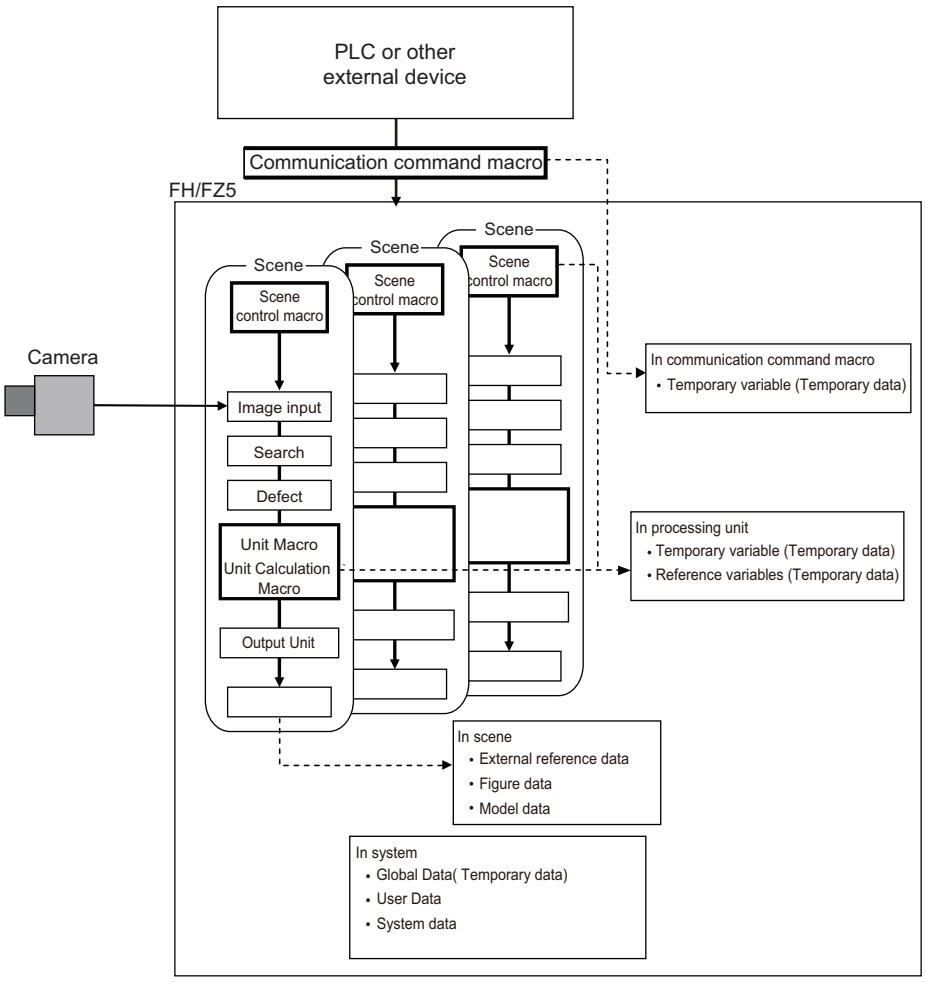
Some data that can be used in the macro customize functions is only saved temporarily in memory, and some data is saved as scene or system data. The area where data is saved is called the "save area", and each data item has a specific save area.

Data save areas that can be used with the macro customize functions are as follows.

Save area	Can be saved? ❍: Yes ×: No	Description
Temporary data	×	Data temporarily stored in memory. Cleared when the power is interrupted or the system is restarted. Use to handle data that does not need to be stored, such as temporary data used during an operation.
Scene data	❍	Data stored as scene data. In addition to "Data save", scene data can be saved to a file as a scene data file. For example, use this when handling data that must be saved for each scene, such as measurement parameters and other settings that vary by product type,
System data	❍	Data stored as system data. In addition to "Data save", system data can be saved to a file as a system data file. For example, use this to handle common scene data (not for only a single scene) and common system system data, such as reference settings and other inspection settings that do not depend on the product type.

Scope and Save Area by Variable and Data Item

The relation between the scope and save area of each variable and data item that can be used in a macro customize function is shown below.

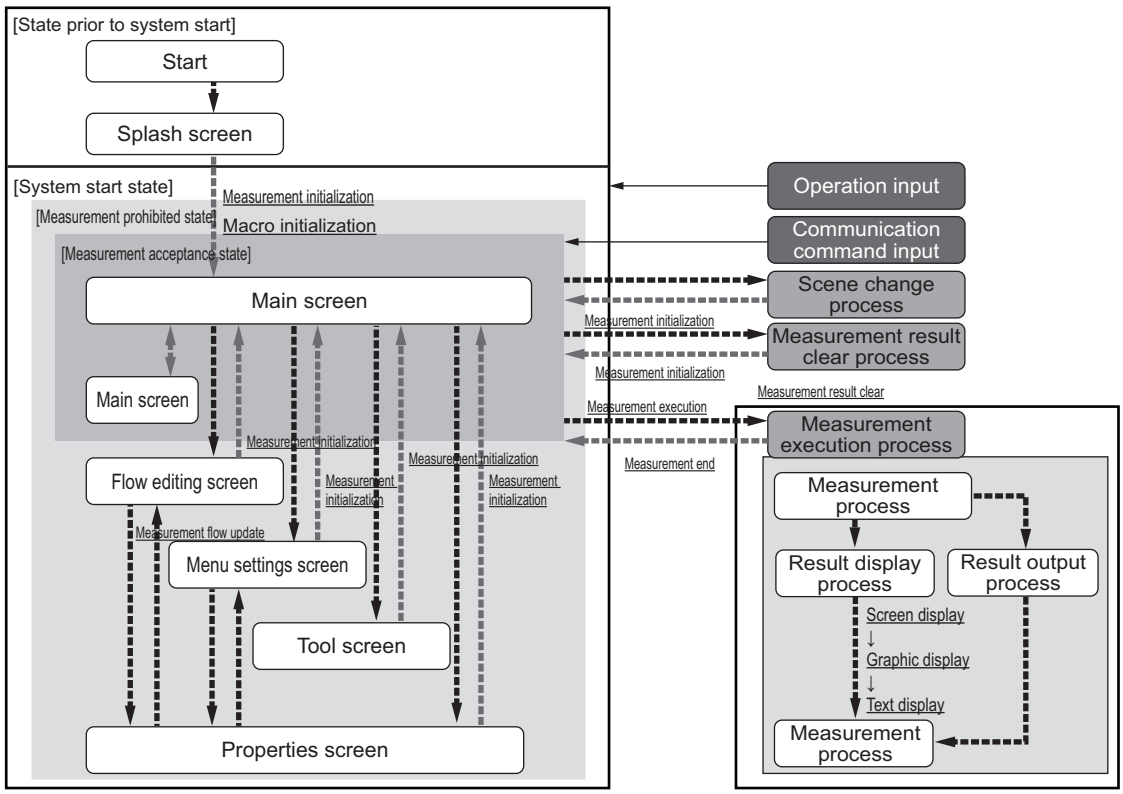


Variable and Data Types	Scope	Can be saved? ❍: Yes ×: No	Data save area
Temporary variable (array variable) Reference: ▶Temporary variable (p.211)	Within a processing unit Within a communication command macro	×	Temporary data
Reference variables Reference: ▶Reference variables (p.212)			
External reference data Reference: ▶External reference data (p.217)	Within a scene	❍	Scene data
Figure data Reference: ▶Figure data (p.218)			
Model data Reference: ▶Model data (p.220)			
Image data Reference: ▶Image data (p.221)		×	Temporary data
Global Data Reference: ▶Global Data (p.222)	Within the system		
User Data Reference: ▶User Data (p.223)		❍	System data
System data Reference: ▶System data (p.224)			

State Transitions and Execution Timing

Necessity of referring to this manual	Unit Macro	Communication Command Macro	Scene Control Macro	Unit Calculation Macro
	As needed			Not required

Macro customize programs consist of several subroutines. Subroutines are generally executed when there is a change of screen, setting, or data status. The subroutines that can be used vary by function of the macro customize functions. Decide which subroutine will be used based on the execution timing of the processes being programmed.



Execution timing and corresponding subroutines are shown below. For a description of each execution timing, refer to the execution timing details.
Reference: ▶Details of subroutine execution timing (p.231)

Execution timing of subroutine		Pre-defined subroutine name	Unit Calculation Macro	Scene Control Macro	Communication Command Macro	Unit Macro	Description
Program initialization		*MCRINIT	---	❍	---	❍	Executed immediately after the program is loaded.
Measurement initialization		*MEASUREINIT	---	---	---	❍	Executed before starting measurements.
Measurement execution		*MEASUREPROC	❍	---	❍	❍	Executed when measurement processing is executed.
Display	Image display	*MEASUREDISPI	---	---	---	❍	Called when an image is displayed in the image window.
	Text display	*MEASUREDISPT	---	---	---	❍	Called when text is displayed in the text window.
	Graphic display	*MEASUREDISPG	---	---	---	❍	Called when a graphic display is displayed in the image window.
Measurement flow update		*RENUMPROC	---	---	---	❍	Called when the processing unit reference number is updated.
Measurement result clear		*CLEARMEASUREDATA	---	---	---	❍	Called when the processing unit measurement results are initialized.

System Status Transitions and Possibility of Execution

There are processes that cannot be executed in certain system statuses in a macro customize program.
The following types of system status exist.

Status type	Description
Measurement prohibited state	Measurement instructions and communication commands cannot be accepted in this state. The measurement prohibited state generally occurs during startup, as well as changes of measurement flow, processing unit and system settings, and when executing a process or operation. In the measurement prohibited state, the BUSY signal or similar status signal turns ON.
Measurement acceptance state	Measurement instructions and communication commands can be accepted in this state. In the measurement accepted state, the BUSY signal or similar status signal turns OFF.

The relation between sensor controller system statuses and execution timing is shown below.

Execution timing of subroutine		Pre-defined subroutine name	System status ❍: Can be executed ×: Cannot be executed	
			Measurement prohibited state	Measurement acceptance state
Program initialization		*MCRINIT	❍	×
Measurement initialization		*MEASUREINIT	❍	×
Measurement execution		*MEASUREPROC	×	❍
View	Screen display	*MEASUREDISPI	❍	❍
	Text display	*MEASUREDISPT	❍	❍
	Graphic display	*MEASUREDISPG	❍	❍
Measurement flow update		*RENUMPROC	❍	×
Measurement result clear		*CLEARMEASUREDATA	❍	×

In some cases it is possible to change the measurement prohibited state and measurement acceptance state while in a subroutine. For details, refer to Exclusive Control in a Process.
Reference: ▶Exclusive Control in a Process (p.234)

Execution During Screen Transitions

Some types of macro customize program execution timing occur in relation to screen transitions.
The following types of screen transitions exist.

Original screen	New screen	Description
Splash screen	Main screen	Transition from splash screen at startup to main screen.
Main screen	Flow edit screen Menu settings screen Tool screen Properties screen	Transition from main screen to other screens.
Flow edit screen	Main screen	Transition from flow edit screen to main screen.
	Properties screen	Transition from flow edit screen to properties screen of processing unit.
Menu settings screen	Main screen	Transition from menu settings screen to main screen.
Tool screen	Main screen	Transition from tool screen to main screen.
	Properties screen	Transition from tool screen to properties screen of processing unit.
Properties screen	Main screen	Transition from properties screen of processing unit to main screen.
	Flow edit screen	Transition from properties screen of processing unit to flow edit screen.

The relation between sensor controller screen transitions and program execution timing is shown below.

Execution timing of subroutine		Pre-defined subroutine name	Screen transition	
			Original screen	New screen
Program initialization		*MCRINIT	Splash screen	Main screen
			Properties screen (macro customize functions)	Flow edit screen Main screen
			Tool screen (Macro customize functions)	Main screen
Measurement initialization		*MEASUREINIT	Splash screen Flow edit screen Menu screen Tool screen Properties screen	Main screen
Measurement execution		*MEASUREPROC	---	---
View	Screen display	*MEASUREDISPI	---	---
	Text display	*MEASUREDISPT	---	---
	Graphic display	*MEASUREDISPG	---	---
Measurement flow update		*RENUMPROC	---	---
Measurement result clear		*CLEARMEASUREDATA	---	---

Process Transitions and Execution Timing

The timing of macro customize program execution is related both to screen transitions and to the type of process being executed.

The relation between process type and program execution timing is shown below.

Process	Timing of execution during process	Timing of execution after process	Description
Scene change	---	Measurement initialization	Scene change process. The program is executed when measurement initialization takes place after the process.
Measurement result clear	Measurement result clear	Measurement initialization	Measurement result clear process. The program is executed when the measurement result is cleared during the process, and when measurement initialization takes place after the process.
Measurement execution	Measurement execution	---	Measurement execution process. The program is executed when measurement is executed during the process.
	Image display	---	Image display process. The program is executed when the image is displayed during the process.
	Graphic display	---	Graphic display process. The program is executed when graphic display takes place during the process.
	Text display	---	Text display process. The program is executed when text display takes place during the process.
Flow edit	Measurement flow update	---	Measurement flow update process. The program is executed when the flow is updated during the process. Measurement flow update consists of the operation and process by which the unit number of the measurement flow is updated, including adding and deleting processing units to the measurement flow and changing the order.

Menu Operations and Execution Timing

In addition to screen transitions and process transitions, the timing of macro customize program execution is also related to the type of menu operation being executed.
The relation between process type and program execution timing is shown below.

Process	Timing of execution during process	Timing of execution after process	Description
Processing unit selection	Image display	---	Selection of a processing unit in the measurement flow when "Define displayed unit" is selected. The program is executed when the image is displayed.
	Graphic display	---	Selection of a processing unit in the measurement flow when "Define displayed unit" is selected. The program is executed when graphic display takes place.
	Text display	---	Selection of a processing unit in the measurement flow when "Define displayed unit" is selected. The program is executed when text display takes place.

Image Display Status Transitions and Execution Timing

The timing of execution of macro customize programs related to display is also related to image window display changes. The relation between the image window and program execution timing is shown below.

Process	Execution timing of processing	Execution timing of processing	Description
Image mode change	Image display Graphic display	Measurement initialization	Image mode change process for the image window. The program is executed when image display and graphic display take place during the process, and when measurement initialization takes place after the process.
Image mode "Through camera image"	Image display Graphic display	Image display Graphic display	Process when the image mode of the image window is "Through camera image". The program is executed repeatedly when image display or graphic display take place while the image mode is "Through camera image".

IMPORTANT

If the image mode of one or more of the image windows of the 24 image windows that can be used in the layout function is "Through camera image", the "Image mode 'Through camera image'" process is executed repeatedly.

Details of subroutine execution timing

Details of the execution timing of subroutines pre-defined in the system are described below.

- **Measurement Initialization (*MCRINIT)**

  The program initialization subroutine is executed immediately after the program is loaded. The process is executed at the timing below.

Details of subroutine execution timing	Main cases of execution
Immediately after macro editing	• When the program editing screen is opened • When the program editing screen is closed
When scene data is loaded	• At startup • When the scene group is changed • When any of the following data is loaded •Scene data •Scene group data •System + Scene group 0 data •Backup data

- **Measurement Initialization (*MEASUREINIT)**

The measurement initialization subroutine is executed before starting measurements. The process is executed at the timing below.

Details of subroutine execution timing	Main cases of execution
Screen transition to main screen	• When the flow edit screen is closed • When the system settings screen is closed • When any other screen that can be called from the main screen such as a scene change screen or saved screen is closed
Immediately after a layout change	• When the layout change is closed • When an image window setting or text window setting is changed
When a setting communication macro function or acquisition communication macro function is executed	When measurement is executed or a communication command other than continuous execution is executed
Immediately after the measurement start macro function is executed	When the "MeasureStart" macro function is executed

IMPORTANT

When the image mode is "Freeze camera image", the measurement image is not updated when measurement initialization takes place. However, if at least one of the 24 image windows that can be used in the layout function is set to "Through camera image", the measurement image of the "Freeze image camera" image windows will be updated. Use the main window layout change function to change the image of all image windows to "Freeze camera image", or use the SetImageWindow macro function to set the image mode of all image windows to "Freeze camera image" before the measurement initialization process is executed.
Reference: ▶Changing the Image Mode and Other Display Contents (p.92)

- **Measurement Execution (*MEASUREPROC)**

The measurement subroutine is executed when measurement is executed. The process is executed at the timing below.

Details of subroutine execution timing	Main cases of execution
When measurement is executed	• When measurement is executed or a communication command for continuous measurement execution is executed • When a measurement button such as the measurement button in the main screen or in the properties screen is clicked • When the "Measure" macro function is executed

- **Image display (*MEASUREDISPI)**

The image display subroutine is executed when an image is displayed. The process is executed at the timing below.

Details of subroutine execution timing	Main cases of execution
When the image display is updated	• When measurement is executed • When the image window settings are changed • When the image window is set to "Define displayed unit" and the processing unit selected in the measurement flow is changed

IMPORTANT

When the position list display is ON, the image display subroutine and text display subroutine are not executed. If you want to execute the image display subroutine or text display subroutine, set the position list display to OFF.
Reference: ▶Changing the Image Mode and Other Display Contents (p.92)

- **Text display (*MEASUREDISPT)**

  The text display subroutine is executed when text display is executed. The process is executed at the timing below.

Details of subroutine execution timing	Main cases of execution
When the text display is updated	• When measurement is executed • When the text window settings are changed • When the text window is set to "Define displayed unit" and the processing unit selected in the measurement flow is changed

- **Graphic display (*MEASUREDISPG)**

  The graphic display subroutine is executed when graphic display is executed. The process is executed at the timing below.

Details of subroutine execution timing	Main cases of execution
When the image display is updated	• When measurement is executed • When the image window settings are changed • When the image window is set to "Define displayed unit" and the processing unit selected in the measurement flow is changed

- **Measurement Flow Update (*RENUMPROC)**

  The measurement flow update subroutine is executed when the measurement flow is changed. The process is executed at the timing below.

Details of subroutine execution timing	Main cases of execution
Immediately after flow editing	• When the order of the processing units in the measurement flow is changed, such as adding, deleting, or moving a processing unit. • When a macro function is executed that changes a processing unit number in the measurement flow or changes a processing unit, such as "AssignUnit", "CopyUnit", or "DeleteUnit".

- **Measurement result clear (*CLEARMEASUREDATA)**

  The measurement result clear subroutine is executed when measurement results are cleared. The process is executed at the timing below.

Details of subroutine execution timing	Main cases of execution
When measurement results are cleared	• When "OK" is clicked in the measurement result clear screen. • When the macro function that clears measurement results is executed

Exclusive Control in a Process

Necessity of referring to this manual	Unit Macro	Communication Command Macro	Scene Control Macro	Unit Calculation Macro
	Not required	As needed		Not required

This primarily uses the communication command macro and scene control macro.
Macro customize functions can be used to perform exclusive control in a process when a program is executed. Exclusive control must be performed during measurement and communication in order to prevent incorrect processing, such as a communication command or screen operation accidentally changing the scene during measurement, or subsequently received data overwriting the currently received data.

Exclusive Control in a Measurement Process

When executing the setting or acquisition of processing unit data, system data, or other data in a program, and when executing a process that changes scene data such as a scene change, you must first stop acceptance of measurement triggers during processing.. If data setting, data acquisition, or a scene change is executed when measurement triggers can be accepted and a measurement trigger is then input during the setting process, acquisition process, or scene change, there is a risk that data inconsistencies will occur or an incorrect measurement result will be output. .
To stop acceptance of measurement trigger input, use the MeasureStop function. To enable acceptance of measurement trigger input, use the MeasureStart function.

Example

```
Rem Stop measurement
MeasureStop

Rem Change the scene or set data

Rem Resume measurement
MeasureStart
```

IMPORTANT

- When executing a scene change or setting or acquiring processing unit data or other data in a program, always execute MeasureStop beforehand. If data setting or data acquisition is executed without executing MeasureStop, there is a risk that measurement may be executed during the setting process or acquisition process and cause inconsistencies in the data being set or acquired.
- If MeasureStart is not executed after MeasureStop, it will not be possible to accept measurement triggers. The BUSY signal will remain ON.
- If measurement is to be executed after MeasureStop in a program, execute MeasureStart before executing measurement. If MeasureStart is not executed after MeasureStop, measurement will not take place when executed.
- With the unit macro or unit calculation macro, create a program that does not require exclusive control. Switching between the measurement prohibited state and measurement acceptance state during measurement execution may cause unexpected operation.

Exclusive Control in a Communication Process

In communication with external devices, the FH/FZ5 series uses a polling process to monitor statuses and data transmission/reception. When sending or receiving data, stop the communication polling process before sending/receiving the data. If data transmission/reception is executed without stopping the polling process, there is a risk that inconsistencies may occur in the received data, such as data other than the intended data being received from the external device.
To change the state of the communication polling process, use the SetPollingState function.

Example: Receiving data in serial normal communication

```
Rem Stop serial normal communication
SetPollingState "SerialNormal", FALSE

Rem Receive the data

Rem Start serial normal communication
SetPollingState "SerialNormal", TRUE
```

How to Use the Debug Function

Necessity of referring to this manual	Unit Macro	Communication Command Macro	Scene Control Macro	Unit Calculation Macro
	Required			

Invalid macro function calls and programming bugs in macro customize programs may cause errors in operation. Identifying the cause and correcting the program when an error occurs is called "debugging". Macro customize functions have a support function for program debugging, allowing errors to be efficiently removed.

IMPORTANT

- In the FZ5-L3□□/FZ5-6□□ series, the system status console window appears in the full screen. To display the main screen, program editing screen, and other sensor controller screens, connect a USB keyboard and change the screen with [ALT] + [TAB].
- Do not close the system status console window by a method such as clicking the "x" button in the upper right corner of the system status console window. The system may not operate correctly. If the system status console window is accidentally closed, save your settings and restart the sensor controller.

Debug Preparations

Debugging is performed in macro customize functions by changing the execution form of the program and using macro functions that are effective for debugging. Debugging can be performed by writing a macro function for debugging in the program and executing the program, and by directly entering a macro function from the system status console window and executing the program.

Program Execution Mode

Macro customize functions allow the execution mode of the program to be specified for each function. By varying the execution form of the program as appropriate for the execution conditions, program management and debugging is made easier.

Execution form	Description
Release mode	Execution form used for program execution and regular use. In release mode, the macro functions that are entered in the program for debugging are disabled.
Debugging mode	This is a convenient execution form used for program correction and debugging. Some debugging functions are only enabled in debug mode.

IMPORTANT

Specify the program execution mode individually for each function and unit.
- Unit calculation macro, unit macro: specify by processing unit
- Scene control macro: specify by scene
- Communication command macro: specify by communication macro

Macro Functions Used for Debugging

Some of the macro functions are effective for debugging. Macro functions that are effective for debugging are described below.

Macro function	Description
Debug (Reference: Details (p.456))	Select the program usage mode and the information output method used when the program is executed. 0 : Set the usage mode to release mode. When an error occurs, an error description is output. 1 : Set the usage mode to release mode. When an error occurs, an error description is output to the system status console window. This setting is recommended for times other than when debugging is performed. 2 : Set the usage mode to release mode. When the program is executed, the contents of each line are output to the system status console window. 3 : Set the usage mode to release mode. When an error occurs, an error description is output to the message box. 16 : Set the usage mode to debug mode. When an error occurs, an error description is output. 17 : Set the usage mode to debug mode. When an error occurs, an error description is output to the system status console window. 18 : Set the usage mode to debug mode. When the program is executed, the contents of each line are output to the system status console window. This setting is recommended when normal debugging is performed. 19 : Set the usage mode to debug mode. When an error occurs, an error description is output to the message box.
Stop (Reference: Details (p.747))	This macro function is only enabled in debug mode. The function is used to stop execution of the program at a break point. You can also set conditions with the SetStop function. When the specified conditions are met, execution of the program stops.
SetStop (Reference: Details (p.723))	This macro function is only enabled in debug mode. Use this function to set conditions for stopping the program with the Stop function.
DebugPrint (Reference: Details (p.458))	This macro function is only enabled in debug mode. The function displays text in the system status console window.

When performing debugging, you can enter macro functions for debugging in the system console window and execute the program. This allows you to conveniently execute and stop the program in the system status console window without the need to edit the program.
Macro functions for debugging that are convenient to use in the system status console window are described below.

Macro function	Description
Cont (Reference: Details (p.436))	Resumes execution of the program after it has been stopped using the Stop function. Parameters can be specified to execute the program in steps. No parameters: Resumes execution of the program. The next program is executed until it ends or an error occurs. 0 : Executes the program by step-in execution. If the current program line calls a subroutine, the subroutine is entered and is executed in steps. Otherwise, the current statement is executed and the program is stopped at the next line. 1 : Executes the program by step-over execution. If the current program line calls a subroutine, the entire subroutine is executed and the program stops at the next line after the subroutine call. Otherwise, the current statement is executed and the program is stopped at the next line. 2 : Executes the program by step-out execution. If the current program line is a subroutine that was called from a subroutine, the entire subroutine after the current program line is executed, and the program stops at the next line of the subroutine that called the subroutine. Otherwise, the program is executed until it ends or an error occurs.
Varlist (Reference: Details (p.796))	Displays information on the variables with the specified variable names in a list in the system status console window.

Debug Procedure

When an error occurs in the program, follow the steps below to correct the program.

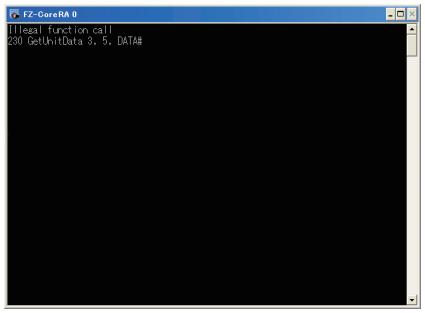
Item name	Step	Description
---	Error occurs	An error occurs during program execution and the program execution process is forcibly stopped.
---	↓	---
Checking Why an Error Occurred (Reference: ▶Checking Why an Error Occurred (p.238))	Check the console window	A brief description of the error appears in the system status console window. Check the error description.
	↓	---
	Identify the nature of the error	Refer to the error list to identify the nature of the error.
---	↓	---
Starting Debug	Change to debug mode	Specify 18 for the debug function parameter at the start of the *MCRINIT subroutine or process of the program.
---	↓	---
Identifying the Cause of an Error (Reference: ▶Identifying the Cause of an Error (p.241))	Identifying the location of the error	Based on the identified nature of the error, determine which line and where on the line the error occurred.
	↓	---
	Identifying the Cause of the Error	Identify why the error occurred.
---	↓	---
Removing the Error (Reference: ▶Removing the Error (p.244))	Correct and check operation	Correct the program to remove the cause of the error, and check operation to verify that the error has been eliminated.
---	↓	---
Exiting Debug	Change to release mode	Specify 1 for the debug function parameter at the start of the *MCRINIT subroutine or process of the program.
---	↓	---
---	Save settings	Save your changes.

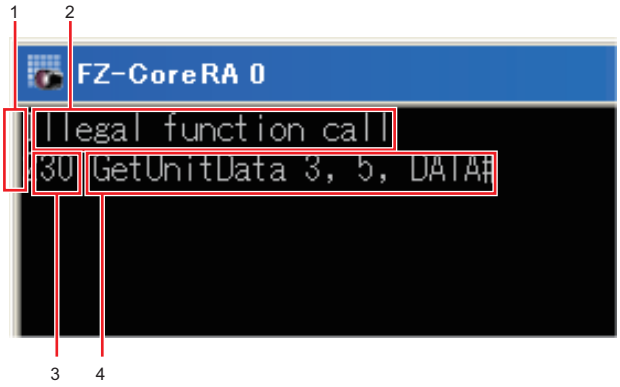
Checking Why an Error Occurred

When an error occurs during program execution, the subroutine process in which the error occurred is forcibly stopped. Processes other than that in which the sensor controller error occurred continue to run.
If an error occurs in the "Unit Macro" processing item or "Unit Calculation Macro" processing item, the measurement result of the processing unit in which the error occurred is NG. If an error occurs in a communication command macro, the macro function returns an error. Regardless of where the error occurred, error information appears in the system status console window.

Checking the System Status Console Window

When an error occurs in the program, error information appears in the system status console window. You can check this information to determine the type of error and the location of the error in the program.





1. Error module

Module that outputs the error. This indicates the function of the macro customize function that output the error.

Content displayed for error module	Description
Macro(U**)>	The unit macro or unit calculation macro outputted error information. "**" after "U" shows the processing unit number.
Macro(SC)>	The scene control macro outputted error information.
Macro(IO)>	The communication command macro outputted error information.

2. Error Message

This message indicates error factors. Based on the error message, you can check what type of error occurred.

3. Error line number

The number of the line where the error occurred. Use this to determine the location of the error in the program.

4. Error statement

Written program content on the line where the error occurred. The error statement allows you to check the written content of the program without opening the program editing screen.

Identifying the Error

Based on the error message that appears in the system status console window, check the error in the error list.

If this gives you sufficient information to identify the cause of the error, remove the error.

Reference: ▶Error List (p.320)

IMPORTANT

If you specified 0 or 16 for the debug function parameter, the error information will not appear in the system status console window. Specify a value other than 0 or 16 for the debug function parameter.

Reference: ▶Macro Functions Used for Debugging (p.236)

Starting Debug

After checking why the error occurred, start Debug. The *MEASUREPROC subroutine of the "Unit Macro" processing item is used for debugging below as an example.

Open the program editing screen and write the debug function at the beginning of the *MCRINIT subroutine or the program.

Example: To start Unit Macro Debug, specify 18 in the parameter for the debug function in the *MCRINIT subroutine.

```
*MCRINIT

  Rem Specify 18 for the debug function parameter and execute
  Debug 18

Return
```

Example: To start Communication Command Macro Debug, specify 18 in the parameter for the debug function at the beginning of the communication command program to be debugged.

```
Rem Specify 18 for the debug function parameter and execute
Debug 18

Rem The actual communication command process is written from here
ChangeScene 1
SetUnitData 2, 101, 0
```

Note

- In addition to 1 and 18, other values are available for the parameter for the debug function. Use these when appropriate.
  Reference: ▶Macro Functions Used for Debugging (p.236)
- If you delete the debug function from the program, the usage mode will remain the usage mode that was set the last time the debug function was executed. Restart to return the usage mode to its default setting.

Identifying the Cause of an Error

After changing the usage mode to debug mode, identify the cause of the error.

Identifying the location of the error

To identify the cause of the error, determine where the error occurred.

1 **Enter the Stop function in the program editing screen.**

Example: Enter the Stop function immediately before the measurement process in the *MEASUREPROC subroutine of the Unit Macro.

```
*MEASUERPROC

  Rem Execute the Stop function and stop the program at this line
  Stop

  Rem The actual measurement process is written from here
  POS.X#=(POS0.X@ + POS1.X@) / 2
  POS.Y#=(POS0.Y@ + POS1.Y@) / 2
  Print POS.Y# / POS.X#

Return
```

2 **Close the program editing screen with [OK] and return to the main screen.**

The main screen appears.

3 **Execute measurement.**

When the Stop function line is executed, the measurement process stops. If the process does not stop, check if debug mode is specified with the debug function parameter.

4 **Check the system status console window.**

The system status console window shows the following:

```
Macro(U3)210 *MEASUREPROC
Macro(U3)220 Stop
Macro(U3)>
```

5 **Execute the program in steps of one line.**

Enter "Cont 1" in the system status console window from your keyboard.

```
Macro(U3)210 *MEASUREPROC
Macro(U3)220 Stop
Macro(U3)>cont 1
```

6 Hit [Enter] on your keyboard.

One line of the program is executed. The program stops at the next line.

```
Macro(U3)210 *MEASUREPROC
Macro(U3)220 Stop
Macro(U3)>cont 1
Macro(U3)230 POS.X#=(POS0.X@ + POS1.X@) / 2
Macro(U3)>
```

7 Continue to step through the program unit until the line that contains the error is executed.

When the line with the error is executed, error information appears in the system status console window. In this case, it can be seen that the error is caused by dividing by 0.

```
Macro(U3)210 *MEASUREPROC
Macro(U3)220 Stop
Macro(U3)>cont 1
Macro(U3)230 POS.X#=(POS0.X@ + POS1.X@) / 2
Macro(U3)>cont 1
Macro(U3)240 POS.Y#=(POS0.Y@ + POS1.Y@) / 2
Macro(U3)>cont 1
Macro(U3)250 Print POS.Y# / POS.X#
Macro(U3)Division by zero in 250
Macro(U3)>
```

Identifying the Cause of the Error

Once you have determined the location of the error, identify the cause.

8 Check the value of the variable.

Enter "VarList" in the system status console window from your keyboard.

9 Hit [Enter] on your keyboard.

The variables are listed in the system status console window.

```
Macro(U3)210 *MEASUREPROC
Macro(U3)220 Stop
Macro(U3)>cont 1
Macro(U3)230 POS.X#=(POS0.X@ + POS1.X@) / 2
Macro(U3)>cont 1
Macro(U3)240 POS.Y#=(POS0.Y@ + POS1.Y@) / 2
Macro(U3)>cont 1
Macro(U3)250 Print POS.Y# / POS.X#
Macro(U3)Division by zero in 250
Macro(U3)>varlist
POS.X#=0.000
POS0.X@=0.000
POS1.X@=0.000
POS.Y#=211.000
POS0.Y@=209.000
POS1.Y@=213.000
```

From the program contents and the variable list, it can be seen that the value of the POS.X# denominator variable is 0.

Stopping the Program When a Specific Condition is Met

The program can also be stopped when a specific condition is met. Setting an appropriate condition allows the location of an error to be identified more efficiently than with the Stop function.

```
*MCRINIT

  Debug 18

  Rem Use the SetStop function to set the program stop condition so that the program stops when the Stop function
  parameter is 0
  SetStop Str$(0)

Return

*MEASUERPROC

  Rem The actual measurement process is written from here
  POS.X#=(POS0.X@ + POS1.X@) / 2
  POS.Y#=(POS0.Y@ + POS1.Y@) / 2

  Rem Execute the Stop function and stop the program on the line where the POS.X# variable is 0
  Stop Str$(POS.X#)
  Rem Execute the Stop function and stop the program on the line where the POS.Y# variable is 0
  Stop Str$(POS.Y#)

  Print POS.Y# / POS.X#

Return
```

Note

A true/false relational expression (true: -1, false: 0) can also be set as the condition for the Stop function.
Example: Stopping the program when the CORRELATION& variable is less than 60
Set the SetStop function so that the program stops when the relational expression is true (-1).
SetStop Str$(-1)
The Stop function stops the program when the relational expression CORRELATION& < 60 is true.
Stop Str$(CORRELATION& < 60)

Removing the Error

Once you have identified the cause of the error, correct the program to remove the error.
After correcting the program, check operation and verify that the error has been removed.

Example: Use "If" to prevent the denominator from becoming 0 in the *MEASUREPROC subroutine of the Unit Macro.

```
*MEASUERPROC

  Rem Execute the Stop function and stop the program at this line
  Stop

  Rem The actual measurement process is written from here
  POS.X#=(POS0.X@ + POS1.X@) / 2
  POS.Y#=(POS0.Y@ + POS1.Y@) / 2

  Rem Add an "If" statement so that division is only executed when POS.X# is not 0
  If POS.X# <> 0 Then
  Print POS.Y# / POS.X#
  EndIf

Return
```

Exiting Debug

When the error has been removed, exit Debug.
Open the program editing screen and change the parameter of the debug function at the beginning of the *MCRINIT subroutine or the program.

Example: To exit Unit Macro Debug, change the parameter for the debug function in the *MCRINIT subroutine to 1.

```
*MCRINIT

Rem Specify 1 for the debug function parameter and execute
Debug 1

Return
```

Note

- In addition to 1 and 18, other values are available for the parameter for the debug function. Use these when appropriate. Reference: ▶Macro Functions Used for Debugging (p.236)
- If you delete the debug function from the program, the usage mode will remain the usage mode that was set the last time the debug function was executed. Restart to return the usage mode to its default setting.

Troubleshooting

Necessity of referring to this manual	Unit Macro	Communication Command Macro	Scene Control Macro	Unit Calculation Macro
	As needed			

When a unit macro customize function does not operate correctly, refer to the following to correct settings or operation.

Troubleshooting for Programming

Problems that are commonly encountered when creating programs with the macro customize functions are described below, along with the actions to take.

Problem	Cause	Action
No response when [DEL], [BS], or [Enter] is clicked.	The focus may not be on the program window.	Click the place you want to work in and then click the button.
A symbol cannot be entered when using a USB keyboard.	The USB key layout may be different from the keys that appear on the keyboard.	The keyboard of the FH/FZ5 series uses the same layout as an English keyboard. Either read the keys on your keyboard as English layout keys, or click [Keyboard] in the program editing screen and use the screen keyboard that appears.
It takes time for the screen keyboard to appear.	The sensor controller and external devices may be connected by Ethernet.	If the sensor controller is connected to external devices by Ethernet, it occasionally takes time for the screen keyboard to appear. In a network environment that does not use a router or DNS server, set the same value for the default gateway and DNS server address as the IP address.
A value is not set in a variable.	The variable in which you are attempting to set a value may have a type mistake.	If the variable name including the identifier is not correct, the variable will be treated as a different variable in the program. Make sure the variable name is correct.
A comment or character string output with the Print function changes to nonsense characters.	Characters other than Japanese or English characters may be used in the Print function or comment.	Characters other than Japanese or English characters cannot be used in the Print function or a comment. Use either Japanese characters or English characters.
When attempting to close the program editing screen, an error occurs and the screen does not close.	Full-width space characters or tab characters may be used in the program.	An error will occur if you attempt to close the program screen editing screen when full-width spaces or tabs are included in parts of the program other than character string type constants and variables. Select "Disp unvisible characters" in the "Disp option" of the program editing screen, and then locate the full-width spaces and tabs in the program and remove.
	An incorrect macro function name may be used in the program.	If an incorrect macro function name is used in the program, an error will occur when the program editing screen is closed. Select "Disp highlight" in the "Disp option" of the program editing screen, and then locate macro function names that are not color-highlighted and correct the macro function names.

Problem	Cause	Action
When loading a program created with the text editor on a computer, only part of the program loads.	The program may exceed the maximum number of characters that can be entered.	A program that exceeds the maximum number of characters that can be entered will not load correctly. Check the remaining number of characters that can be entered in the program editing screen, and correct the program so that is does not exceed the character limit.

Troubleshooting When Checking Operation

Problems that are commonly encountered when checking the operation of programs created with the macro customize functions are described below, along with the actions to take.

Troubleshooting when checking the operation of the Unit Calculation Macro

Problem	Cause	Action
An error message appears in the system status console window	An error occurred when the program was executed.	Check the error message and correct the error. Reference: ▶Error List (p.320) If it is difficult to identify the error, use Debug to determine the cause and correct the error. Reference: ▶How to Use the Debug Function (p.235)
The calculation result of the Unit Calculation Macro processing unit is unmeasured.	The operator setting may be disabled.	Set the check box of the operater on to enable.
	The calculation judgement conditions may not be correctly set.	Set judgement conditions that are appropriate for the calculation and expected result.

Troubleshooting when checking the operation of the Scene Control Macro

Problem	Cause	Action
An error message appears in the system status console window	An error occurred when the program was executed.	Check the error message and correct the error. Reference: ▶Error List (p.320) If it is difficult to identify the error, use Debug to determine the cause and correct the error. Reference: ▶How to Use the Debug Function (p.235)
Data reception during processing sometimes fails.	The communication process may not have been stopped with the SetPollingState function.	Before acquiring data with the ReceiveData or other function, use the SetPollingState function to stop the communication process. Reference: ▶SetPollingState in Macro Function Reference (p.711)
Processing unit figure data cannot be set with the SetUnitFigure function, and registered figures are also cleared.	There may not be a measurement image when the figure is set.	When there is no measurement image because the processing unit is in the unmeasured state or otherwise, the figure setting will fail and previous settings will be cleared. Execute measurement before setting a figure, or use the ImageUpdate function to ready an image. Reference: ▶ImageUpdate Function (p.568)
Measurement or re-measurement does not take place in the program	"Measure" or "Remeasure" is executed in the program in the measurement prohibited state.	When executing measurement with "Measure" or "Remeasure" in the program, use the MeasureStop function and MeasureStart function to appropriately control the measurement trigger input prohibited/ allowed state. Reference: ▶Exclusive Control in a Measurement Process (p.234)

Troubleshooting when checking the operation of the communication command macro

Problem	Cause	Action
An error message appears in the system status console window	An error occurred during program execution.	Check the error message and correct the error. Reference: ▶Error List (p.320) If it is difficult to identify the error, use Debug to determine the cause and correct the error. Reference: ▶How to Use the Debug Function (p.235)
Data reception during processing sometimes fails.	The communication process may not have been stopped with the SetPollingState function.	Before acquiring data with the ReceiveData or other function, use the SetPollingState function to stop the communication process. Reference: ▶SetPollingState in Macro Function Reference (p.711)
Processing unit figure data cannot be set with the SetUnitFigure function, and registered figures are also cleared.	There may not be a measurement image when the figure is set.	When there is no measurement image because the processing unit is in the unmeasured state or otherwise, the figure setting will fail and previous settings will be cleared. Execute measurement before setting a figure, or use the ImageUpdate function to ready an image. Reference: ▶ImageUpdate Function (p.568)
The communication command macro does not execute	The command may not be enabled.	In the setting screen of the communication command macro tool, select the checkbox of the command "No." that you want to execute.
	You are using simulation software.	The communication function does not operate on a computer. To execute commands created with the communication command macro, execute on the sensor controller.
	"Measure" or "Remeasure" is executed in the program when there is a checkmark in BUSY ON.	Before using "Measure" or "Remeasure" in command processing to execute measurement, remove the checkmark from BUSY ON of that command in the setting screen of the communication command macro tool. In addition, use the MeasureStop function and MeasureStart function to appropriately control the measurement trigger input prohibited/allowed state. Reference: ▶Exclusive Control in a Measurement Process (p.234)
The normal communication command macro is not executed.	The character string sent as the command may not match the character string set in the command name.	Send the same character string for the command as the command name set in the setting screen of the communication command macro tool.
A communication command macro other than the normal communication command macro is not executed.	The numerical value sent as the command parameter when the command is executed does not match the command number.	As the command, send the same numerical value in hexadecimal (binary in the case of parallel) as the command number shown in the setting screen of the communication command macro tool.
A command parameter cannot be specified in the communication command macro	You may be attempting to specify a command parameter in the parallel communication command macro.	A command parameter cannot be specified in the communication command macro in parallel communication. Consider one of the following methods: • Set the necessary data in the processing unit with the processing unit data setting, and then execute the command. • Use a communication method other than parallel.
BUSY ON does not take place when the communication command macro is executed	The BUSY ON checkbox may not be selected.	Select the BUSY ON checkbox of the command you want to execute in the setting screen of the communication command macro tool.

Problem	Cause	Action
The communication command macro settings cannot be saved.	You may be attempting to save the settings with [Save to file].	The communication command macro settings are not saved in the backup data (.bkd). Consider one of the following methods: • Execute [Data save]. • Execute [Export] in the setting screen of the communication command macro tool.
A standard communication command does not execute	The name of a command created with the communication command macro may be the same as the name of the standard communication command.	If the name of a command created with the communication command macro function is the same as the name of a standard communication command, the command created with the communication command macro is given priority. Change the command name to a name that is different from the standard communication command.

Troubleshooting when checking the operation of the unit macro

Problem	Cause	Action
An error message appears in the system status console window	An error occurred when the program was executed.	Check the error message and correct the error. Reference: ▶Error List (p.320) If it is difficult to identify the error, use Debug to determine the cause and correct the error. Reference: ▶How to Use the Debug Function (p.235)
Data reception during processing sometimes fails	The communication process may not have been stopped with the SetPollingState function.	Before acquiring data with the ReceiveData or other function, use the SetPollingState function to stop the communication process. Reference: ▶SetPollingState in Macro Function Reference (p.711)
Unit figure data cannot be set, and registered figures are also cleared.	There may not be a measurement image when the figure is set.	When there is no measurement image because the processing unit is in the unmeasured state or otherwise, the figure setting will fail and previous settings will be cleared. Execute measurement before setting a figure, or use the ImageUpdate function to ready an image. Reference: ▶ImageUpdate Function (p.568)
A figure drawn with a macro function of screen display window control such as DrawLine is drawn at a position different from the actual coordinates	The processing unit that corrects position such as Position Correction may be registered in the measurement flow.	If the processing unit that corrects position is registered in the measurement flow, the coordinate values that can be acquired with the UnitData function are the values before correction. In the parameter of the DrawLine function, specify the processing unit number of the unit macro processing unit for which the program is written.
A measurement result image such as the labeling binary image using the display process subroutine cannot be displayed.	You may be attempting to display the measurement result image of a processing unit that does not have image data.	The main measurement processing items such as labeling do not have image data. The measurement result image of a processing item that does not have image data cannot be acquired in the program. To acquire and display a measurement result image, use a processing item that has image data such as Advanced Filter. For image numbers and image content that can be referenced in each processing item, refer to the image number list. Reference: ▶Image Number List (p.371)

Problem	Cause	Action
An error such as "Illegal function call" or "zero divide" occurs when the sensor controller is started	It is possible that a value is used that has not been initialized in the display process subroutine.	If executed with the display process subroutine in the unmeasured state, a data acquire process such as GetGlobalData may fail, or a variable may be used without a set value. Change the program for the display process subroutine so that the drawing process is only executed after measurement.
The processing result of the Unit Macro processing unit text display process and screen display process do not appear	The position list display may be ON.	When the position list display is ON, the text display subroutine and image display subroutine are not executed. Set the position list display to OFF. Reference: ▶Changing the Image Mode and Other Display Contents (p.92)
An "illegal function call" error sometimes occurs when search process unit measurement is executed with the MeasureProc function	It is possible that no objects were found in the search process.	An "illegal function call" error occurs if there are 0 detections in the measurement process of the search process unit executed with the MeasureProc function. Use the Try - Catch - End Try function to create a process that handles the occurrence of 0 detections in the program. Reference: ▶Try - Catch - End Try function (p.773)

Troubleshooting during debugging

Problems that are commonly encountered when debugging programs with the macro customize functions are described below, along with the actions to take.

Problem	Cause	Action
An error message appears in the system status console window	An error occurred when the program was executed.	Check the error message and correct the error. Reference: ▶Error List (p.320) If it is difficult to identify the error, use Debug to determine the cause and correct the error. Reference: ▶How to Use the Debug Function (p.235)
The system status console window appears in the full screen	In the FZ5-L35□/FZ5-6□□ series, the system status console window appears in the full screen.	Connect a USB keyboard to the sensor controller and change the screen with [Alt] + [Tab].
A comment or character string output with the Print function changes to nonsense characters.	Characters other than Japanese or English characters may be used in the Print function or comment.	Characters other than Japanese or English characters cannot be used in the Print function or a comment. Use either Japanese characters or English characters.
The Stop function does not stop the program	It is possible that release mode is specified in the Debug function.	When release mode is specified in the Debug function, functions such as the Stop function and DebugPrint function are not enabled. Execute the Debug function in the *Mcrinit subroutine to change from release mode to debug mode. Reference: ▶How to Use the Debug Function (p.235)
A debug character string cannot be output to the system status console window with the DebugPrint function	It is possible that release mode is specified in the Debug function.	When release mode is specified in the Debug function, functions such as the Stop function and DebugPrint function are not enabled. Execute the Debug function to change from release mode to debug mode. Reference: ▶How to Use the Debug Function (p.235)

Problem	Cause	Action
A specific program line does not execute	It is possible that a statement and a comment are written on the same line.	A statement is sometimes not executed correctly when written on the same line as a comment. Write the comment on a separate line.
	It is possible that release mode is specified in the Debug function.	When release mode is specified in the Debug function, functions such as the Stop function and DebugPrint function are not enabled. Execute the Debug function to change from release mode to debug mode. Reference: ▶How to Use the Debug Function (p.235)

Troubleshooting during regular operation

Problems that are commonly encountered when using the macro customize functions in regular operation are described below, along with the actions to take.

Problem	Cause	Action
An error message appears in the system status console window	An error occurred when the program was executed.	Check the error message and correct the error. Reference: ▶Error List (p.320) If it is difficult to identify the error, use Debug to determine the cause and correct the error. Reference: ▶How to Use the Debug Function (p.235)
A standard communication command does not execute	The name of a command created with the communication command macro may be the same as the name of the standard communication command.	If the name of a command created with the communication command macro function is the same as the name of a standard communication command, the command created with the communication command macro is given priority. Change the command name to a name that is different from the standard communication command.
Communication command macro settings cannot be applied to another sensor controller	It is possible that only backup data (.bkd) is loaded in the other sensor controller.	Communication command macro settings cannot be saved in backup data (.bkd). Load the settings file (.mcr) that was saved by executing [Save] in the setting screen of the communication command macro tool into the other sensor controller separately from the backup data (.bkd). To load the settings file (.mcr) in the other sensor controller, execute [Load] in the setting screen of the communication command macro tool of that sensor controller.
The communication command macro settings cannot be saved.	You may be attempting to save the settings with [Save to file].	The communication command macro settings are not saved in the backup data (.bkd). Consider one of the following methods: • Execute [Data save]. • Execute [Export] in the setting screen of the communication command macro tool.
Measurement flow settings that were saved are cleared after restart	It is possible that the *SaveProc subroutine is written in the scene control macro program.	When the *SaveProc subroutine is written, only the process written in the *SaveProc subroutine is executed when [Data save] is executed or scene data is saved. For this reason, overall scene data changes are not saved in the way they are when regular [Data save] or [Save to file] is executed. Delete the *SaveProc subroutine.

Problem	Cause	Action
An "Illegal function call" error sometimes occurs when search process unit measurement is executed with the MeasureProc function	It is possible that no objects were found in the search process.	An "Illegal function call" error occurs if there are 0 detections in the measurement process of the search process unit executed with the MeasureProc function. Use the Try - Catch - End Try function to create a process that handles the occurrence of 0 detections in the program. Reference: ▶Try - Catch - End Try function (p.773)

MEMO

Saving and Loading Data

This section explains the methods for saving and loading settings and image data.

Saving Data to the FH/FZ5

This section describes how to save and load setting data and logged image data.

About Saving Areas

The following saving areas can be used with the Sensor Controller.

Saving area		Description	Save operation
Controller	Flash memory in the Controller	The settings data (system data, scene data, and scene group data) is saved to this area when you click the [Data save] button. Data is held even after the power supply is turned OFF. When you restart the Controller, the Controller loads and enables the settings data in the flash memory.	[Function] – [Data save] or the [Data save] button
	Controller memory (RAM)	This is the area where images are temporarily stored when logging images using the logging function. This memory is a ring memory. Images will be overwritten starting with the oldest image if the maximum number of save images is exceeded.	[Function] – [Save to file] or [Function] – [Screen capture]
	RAM disk in the Controller	You can use this memory as a temporary file save destination. The data is cleared when you turn OFF the power supply to the Controller. As this is memory inside the Controller, files can be saved and read faster than when using external memory. The FH or FZ5-11□□ has 256 MB, and the FZ5-L3□□ has 40 MB. The capacity cannot be changed. The RAM disk data can be sent to or received from external devices using FTP.	
External memory	USB memory	This memory is used to back up setting data as a precaution, to copy setting data to another Controller, and to load data to a computer.	
	SD memory card (FH only)		
	Shared folder on a computer connected to the network		

The following table shows the data that is retained or cleared and the data that is saved when the power supply is turned OFF.

Saving area		At power OFF	Target data				
			Settings data (system data, scene data, and scene group data)	Logged images	Last input image and last NG image	Logged data (measure-ment data)	Capture image
Sensor Controller	Flash memory of the Controller	Retained	OK	---	---	---	---
	Controller memory (RAM)	Cleared	---	OK*1			
	RAM disk	Cleared	OK	OK*2	OK	OK	OK
External memory	USB memory	Retained					
	SD Memory Card						
	Shared folder on a computer connected to the network						

*1: If images are being logged by the system, select [Other] – [Logging setting] under the System Settings and set the destination for logged images to [Save to memory].

*2: If images are being logged by the system, select [Other] – [Logging setting] under the System Settings and set the destination for logged images to [Save to memory + file].

IMPORTANT

During data transfer, do not turn OFF the power supply.

- When a message indicating that processing such as saving or loading is in progress is displayed, do not restart the Controller or turn OFF the power supply. Data will be corrupted and the system will not work properly at the next startup.
- Do not remove external memory devices while data is being saved or loaded. The operation of the Controller may corrupt the data or external memory.
- Do not change the file name extension of saved files. If you change the extension, the file cannot be loaded as setting data. In addition, if setting data in which the extension was changed is loaded, the system may not work properly later.
- Depending on the settings, saving may fail due to insufficient external memory space. If saving fails and the error message “Please check” appears, check to see if there is unnecessary data in the external memory and save the data after the unnecessary data has been deleted.

External Drive Names

Drive Names for USB Devices

The Controller has four USB connectors. If more than one USB memory device is inserted, specify the USB memory drive that is to be the destination.

- FZ5-L3□□ or FZ5-6□□
  The drive names of USB memory devices are USBDisk, USBDisk2, USBDisk3 and USBDisk4 according to the sequence in which devices are inserted into the Controller.
  If the Controller, however, is started with more than one USB memory device inserted, drive names will be assigned based on the ports in which the USB memory devices are inserted. Depending on the Controller type, USB memory devices are recognized and drive name are assigned using the following sequence.
  - **Controller with Integrated Panel**
    1: Left side on the front - 2: Right side on the front - 3: Front on the side - 4: Back on the side
  - **Box-type Controller**
    1: Lower left on the front - 2: Lower right on the front - 3: Upper left on the front - 4: Upper right on the front
- FH or FZ5-11□□
  USB devices are assigned as drives E:\, F:\, G:\ and H:\ in the order in which they are inserted.

Drive Name of SD Memory Card

The FH has a slot for an SD memory card. This is a fixed drive and is assigned as drive M:\.

IMPORTANT

Box-type Controllers

If USB memory devices are separately connected to adjacent USB connectors, the contact between USB memory devices may possibly lead to failure or damage.

Using External Memory Devices (FH and FZ5-11☐☐ Only)

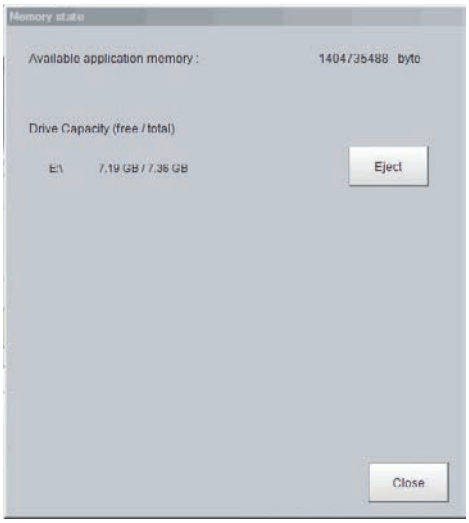
Use the following procedure to remove the external memory device.

1 **In the Main Window, select [System information] from the [Function] menu.**

The system information is displayed.

2 **Information on each drive is displayed in the [Memory state] dialog box.**

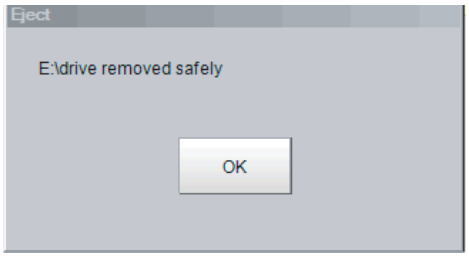
If an external memory device is inserted, the [Eject] button is displayed.



3 **To remove the USB memory device, click the [Eject] button.**

When the USB Memory Can Be Removed

If the removal failed, wait until the external memory device is no longer being accessed and then try removing the device again.



IMPORTANT

Never try to remove the external memory device while the external memory is being accessed. A critical failure may occur.

Shared folder on a computer connected to the network

You can setup a network drive in the system settings. This allows the Sensor Controller to handle the hard drive on a computer that is connected to the local area network as an external memory device.

Reference: ▶ Setting a Network Drive: [Network Drive Setting] (p.298)

Saving Settings Data to the Flash Memory and Restarting

This section describes the procedure to save settings data (system data, scene data, and scene group data) in the flash memory of the Sensor Controller. Make sure to save setting data when settings have been changed. Every time you restart the Sensor Controller, the data in the memory is loaded and enabled.

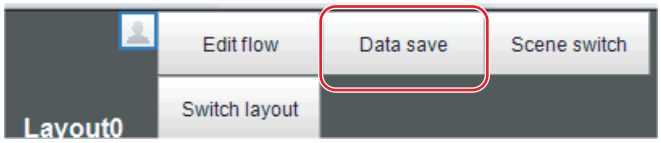
IMPORTANT

- If [Save to file] is performed for [System + scene group 0 data], the data being saved will also be saved to the Controller's flash memory at the same time. Do not turn OFF the power supply during this process. The Controller may not start up properly the next time it is turned ON.
- During data transfer to external memory, do not remove the external memory device until the transfer is completed. Data and the external memory may be corrupted.

1 **For the FZ5-L3□□ or FZ5-6□□, insert an external memory device into the Controller before you save the data for scene groups 1 to 31.**

2 **Click [Save to file] on the toolbar in the Main Window.**

A confirmation message is displayed.



3 **Click [OK].**

For the FZ5-L3□□ or FZ5-6□□, system data and data from scene group 0 are saved to the Controller's flash memory and the data from scene groups 1 to 31 is saved to the external memory.
For the FH or FZ5-11, all data is saved in the flash memory in the Controller by default.
If more than one USB memory device is connected to the Controller, check in the File Explorer to confirm that the external memory device where the scene group data is to be saved is recognized as the USB disk.

Reference: ▶ External Drive Names (p.255)

Note

If an external memory device is not inserted, a confirmation message is displayed.
If [OK] is clicked, only system data and data from scene group 0 are saved in the Controller's flash memory.

Saving Settings Data to the Controller RAM Disk or an External Memory Device

Setting data files are saved to the RAM disk or to an external memory device. The following table lists the data that can be saved.

Data	Description
System data (*.ini)	This is the setting data, such as the [System] menu setting contents, that is shared within the Controller.
Scene data (*.scn)	This is the data for one scene. It contains the sequence of units set in each scene and the set values of units within the scenes.
Scene group data (*.sgp)	The data of scene group with 128 scenes.
Scene + Scene group 0 data (*.bkd)	This is the system data and the data from scene group 0 combined.

IMPORTANT

During saving, do not restart, turn OFF the power supply, or remove the external memory device. Data will be corrupted and the system will not work properly at the next start-up. Be particularly careful if you use [Save to file] for [System + Scene group 0 data] because the data to be saved will also be saved to the flash memory in the Controller at the same time.

Note

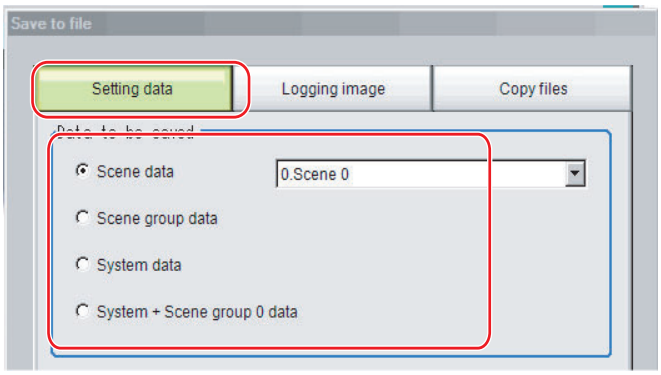
When [Save to file] is executed, the data to save is also saved in the Controller (except scene group data 1 to 31).

1 **To save to a USB memory device (or an SD memory card), insert the USB memory device (or SD memory card) in the Controller.**

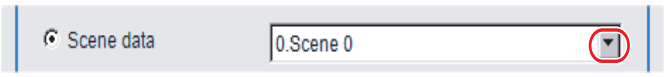
2 **In the Main Window, select [Save to file] from the [Function] menu.**

The [Save to file] dialog box is displayed.

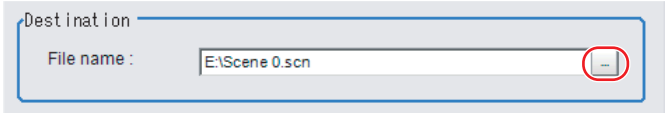
3 **Click the [Setting data] button to select the data to save.**



4 **If you selected [Scene data], click the [▼] button to select the number of the scene to save.**



5 **Specify the destination folder and file name.**



6 **Click [OK].**

The dialog box showing transfer status is displayed, and the data is sent to the save destination.

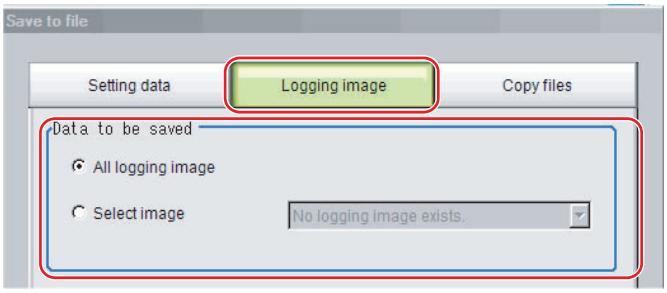
Use the following procedure to save image data that was logged in the Controller memory (RAM) to the Controller RAM disk or an external memory device.
Reference: ▶ About Saving Areas (p.254)

1 **When saving data to external memory, insert an external memory device into the Sensor Controller.**

2 **In the Main Window, select [Save to file] from the [Function] menu.**

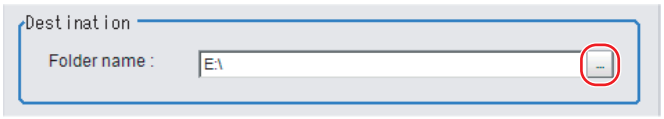
The [Save to file] dialog box is displayed.

3 **Click [Logging image] and select the file to save.**



Parameter	Set value [Factory default]	Description
Data to be saved	[All logging image]	All logged images are saved.
	Select image	The selected logged images are saved. Click [▼] to select the image to save. When [Latest measurement] - [Logging image] is selected, the save file name will be LoggingImage000.ifz.

4 **Specify the destination folder.**



5 **Click [OK].**

The dialog box showing transfer status is displayed, and the data is sent to the save destination.

Note

When you select [All logging image], the save file name is automatically specified as “Loggingimage000.ifz”.
The file names for the images through the image captured one trigger before the latest image are:
- Loggingimage000.ifz (latest image)
- Loggingimage001.ifz (image from one trigger ago)

The file names for the images through the image captured two triggers ago are:
- Loggingimage000.ifz (latest image)
- Loggingimage001.ifz (image from one trigger ago)
- Loggingimage002.ifz (image from two triggers ago)

The file names for the images through the image captured n triggers before the latest image are:
- Loggingimage000.ifz (latest image)
- Loggingimage001.ifz (image from one trigger ago)
- Loggingimage002.ifz (image from two triggers ago)

:
- Loggingimage n.ifz (image from n triggers ago)

Copying or Moving Files between the Controller RAM Disk and an External Memory Device

You can copy or move files between the Controller RAM disk and an external memory device.
Images and data saved on the RAM disk are deleted if the power supply is turned OFF. To keep these images and data, copy or move them to an external memory device. The types of files that can be copied/moved are as follows:

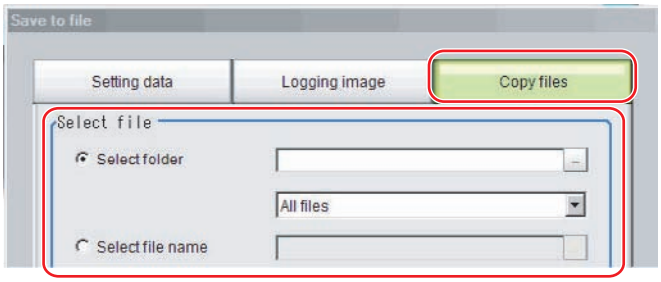
- **Settings data (system data, scene data, and scene group data)**
- **Logged images**
- **Logged data (measurement data)**

1 **Insert an external memory device in the Controller.**

2 **In the Main Window, select [Save to file] from the [Function] menu.**

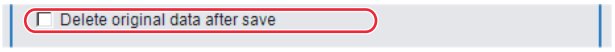
The [Save to file] dialog box is displayed.

3 **Click [Copy files] and select the file or folder to copy or move.**



Parameter	Set value [Factory default]	Description
Select file	[Select folder]	Copies or moves more than one file in the folder. • Click the [...] button and specify the source folder to copy or move a file from. • Click the [▼] button and specify the file type to select. Select [All files] to copy or move all files in the folder. Specify a file type to copy or move only files that are of the specified file type in the folder.
	Select file name	Copies or moves the selected file. Click the [...] button and specify the file name.

4 **To delete the copy source file after saving a copy to external memory, select [Delete original data after save].**



5 **Click [OK].**

The dialog box showing transfer status is displayed, and the data is sent to the save destination.

Loading Settings Data from the Controller RAM Disk or an External Memory Device to the Sensor Controller

Use the following procedure to load settings data that is saved in the Controller RAM disk or an external memory device to the Sensor Controller The scene name and scene group name that have been loaded are displayed in the Measurement Information Display Area.

IMPORTANT

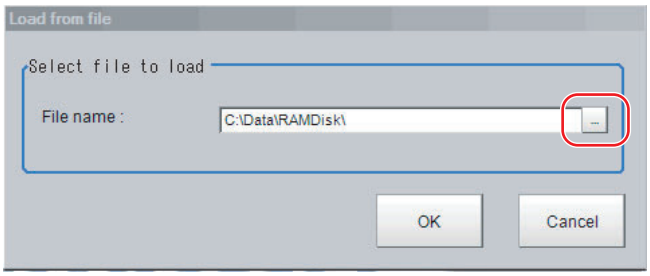
- Always restart the Sensor Controller immediately after you load the settings data.
- If [Load from file] is performed for [System + scene group 0 data], the data being loaded will also be saved to the Controller's flash memory at the same time. While data is being loaded, do not restart, turn OFF the power supply, or remove the external memory device. Data will be corrupted and the system will not work properly at the next startup.

1 **Perform either of the following in advance.**
Send the setting data to the Controller's RAM disk using the FTP.
Insert the external memory device that has the load data saved in it into the Controller.

2 **In the Main Window, select [Load from file] from the [Function] menu.**

The [Load from file] dialog box is displayed.

3 **Select the file to load.**



4 **Click [OK].**

A dialog box that shows the progress of the transfer is displayed, and the data is transferred.
The loaded settings are enabled after the Sensor Controller is restarted.

5 **Select [System restart] from the [Function] menu.**

MEMO

Changing the System Environment

This section describes the settings that are related to the system environment of the Controller.

Checking the Camera Connections: [Camera Connection]

You can check whether a Camera is connected. There are no settings.

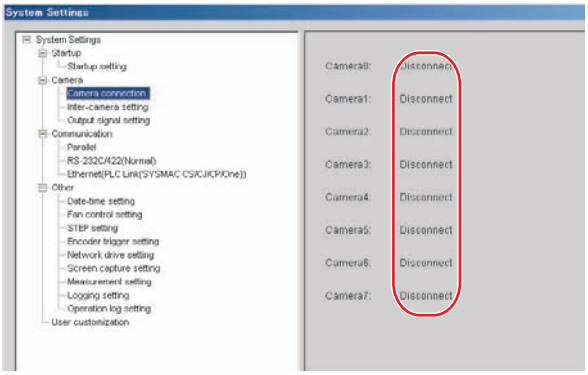
1 **In the Main Window, select [System settings] – [Camera] – [Camera Connection] from the [Tool] menu.**

The Camera Connection Settings View is displayed.

2 **Check the connection status.**

If a Camera is connected, the model name of the connected Camera will appear next to the Camera number.

If the [Disconnect] is displayed even though a Camera is connected, there could be a failure or a disconnected line. Check the physical connection.



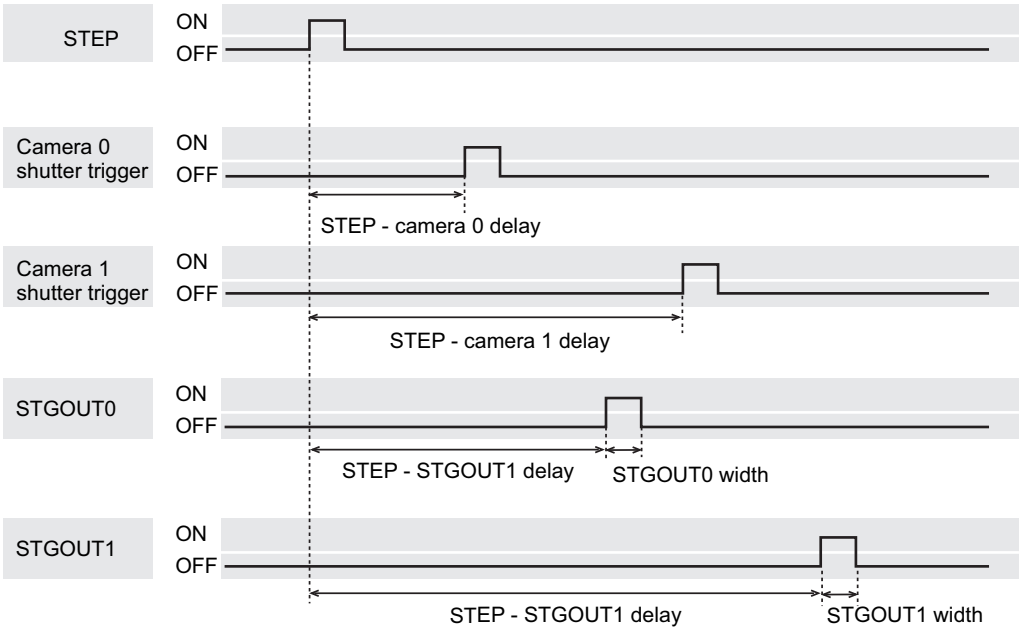
3 **Click the [Close] Button.**

Setting the Trigger Delay: [Inter-camera setting]

This setting is used to set the delay time from when the STEP signal for the input trigger is received to when the shutter is triggered. You can use this to prevent mutual interference caused by the lighting when more than one Camera is used, or as a simple trigger delay when only one Camera is used.

Note

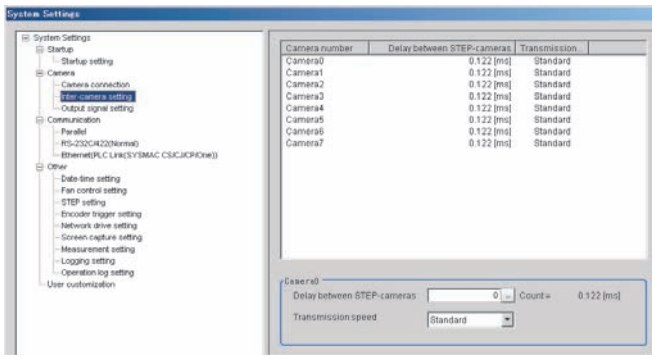
Set the STGOUT pulse width in the [Electronic flash setting] under the [Camera Image Input] processing item.
Reference: ▶ *Electronic Flash Setting* of the *Vision System FH/FZ5 Series Processing Item Function Reference Manual* (Cat No. Z341).



1 **In the Main Window, select [System settings] – [Camera] – [Inter-camera setting] from the [Tool] menu**

The Inter-camera Settings View is displayed.

2 **Click a camera number to set the delay between STEP-cameras and then specify the delay count value.**



Item	Set value [Factory default]	Description
Delay count between STEP-cameras	[0] to 511 (1 count is 30 μs)	Sets the delay time from when the STEP signal is received to when exposure begins for the selected Camera. The delay count - delay time conversion equation for the FH sensor controller and that for the FZ5 sensor controller differ. The delay time for the FZ5 equivalents to the displayed time value in the Delay between STEP-cameras field (= Delay count between STEP-cameras x 30 μs + 122 μs). The delay time for the FH equivalents to the displayed time value in the Delay between STEP-cameras field (= Delay count between STEP-cameras x 30 μs + 122 μs) + 120 μs.
Transmission speed	[Standard] or High speed	If you use a Camera Cable that is shorter than 5 m to connect to an FH-SC□□/SM□□ Camera, you can use the [High speed mode] option to reduce the image input time. Refer to the instruction manual for the Camera for actual frame rates.

Note

With an FH Sensor Controller, you can make Camera settings for up to eight Cameras.
With an FZ5 Sensor Controller, you can make Camera settings for up to four Cameras, Camera 0 to Camera 3. If you make changes to the setting for Camera 4 to Camera 7, they will not be applied.

Setting the SHTOUT Signal: [Output Signal Settings]

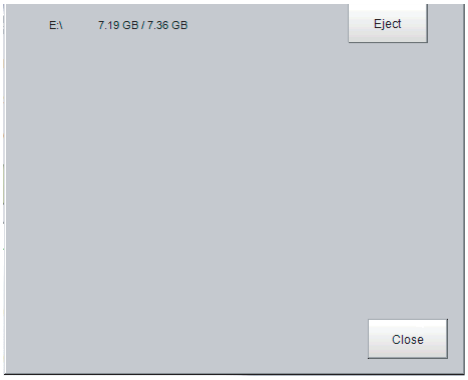
This function is exclusively for an FH Sensor Controller.
This setting affects the SHTOUT signal that is output when the exposure of the Camera ends.You can detect when the exposure ends with the SHTOUT signal to minimize the time to hold the workpieces still for taking images. This allows you to move the workpiece or the Camera immediately after the exposure ends.

IMPORTANT

- You cannot use the STGOUT signal and SHTOUT signal at the same time. Select the signal to output depending on the application.
- The settings of the [Output Signal Settings] apply only to the FH.
  With an FZ5 Sensor Controller, you cannot use the SHTOUT signal even if you change the settings of the [Output Signal Settings].

1 In the Main Window, select [Camera] – [Output Signal Setting] from the [Tool] menu.

The settings dialog box for the output signals is displayed.



Note

- If the [Output Signal Selection] is set to [SHTOUT], the SHTOUT signal is output according to the settings in the [Individual Line Settings]. If the [Output Signal Selection] is set to *STGOUT*, the STGOUT signal is output according to the settings in the [Electronic flash setting] for a camera input processing item.
- When using the Multi-line Random-trigger Mode, the output signal selection can be set only to the output signal setting for line 0. The settings for line 0 are applied to the other lines.
- If you are using the Camera Image Input HDR, or Camera Image Input HDR Lite processing item, the SHTOUT signal is output for every image that is taken.
- If you are using the Through Image Mode in the Main Window, the SHTOUT signal is output for every through image that is taken.

2 Select the output signal in the [Common setting] Area.

Parameter	Set value [Factory default]	Description
Output signal	[STGOUT]	The STGOUT signal line is used for the STGOUT signal. If you selected STGOUT, you cannot use the SHTOUT output signal.
	SHTOUT	The STGOUT signal line is used for the SHTOUT signal. If you selected SHTOUT, you cannot use the STGOUT output signal.

3 Set SHTOUT for each line in the [Line setting] area.

Parameter	Set value [Factory default]	Description
SHTOUT signal delay [μs]	[0] to 1,000	Sets the delay time from when exposure is completed to when the SHTOUT signal turns ON in increments of 10 μs.
SHTOUT signal pulse width [μs]	40 to 10,000 [5,000]	Sets the time to output the SHTOUT signal in increments of 10 μs.
SHTOUT signal pulse polarity	• [Positive] • Negative	Set the polarity of the SHTOUT signal. Positive: The SHTOUT signal changes from OFF to ON when the exposure ends. Negative: The SHTOUT signal changes from ON to OFF when the exposure ends.

4 Click the [Close] Button.

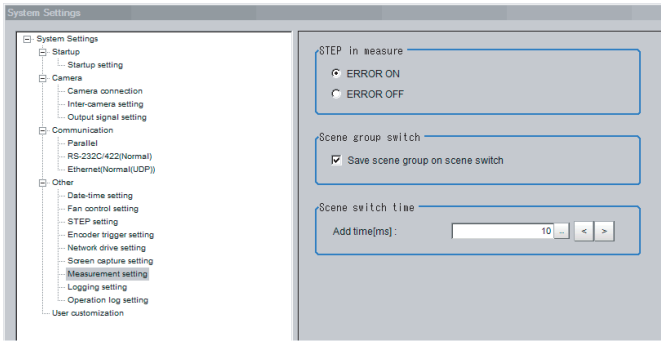
Setting the Conditions That Are Related to Operation during Measurement

You can change the following items of operation during measurement.

- **Operation when the next STEP signal is input during measurement**
- **Whether the scene group is saved when you change to another scene group**
- **Setting for when the scene switch time is short and switching cannot be detected by the external device**

1 **In the Main Window, select [System settings] – [Other] – [Measurement settings] from the [Tool] menu.**

The [Measurement settings] dialog box is displayed.



2 **Set each item as necessary.**

Item	Set value [Factory default]	Description
STEP in measure	[ERROR ON] or ERROR OFF	Set whether the ERROR signal output turns ON when the next STEP signal is input during measurement.
Save scene group on switch scene	• [Selected] • Not selected	Set the operation to perform when you change to another scene group. Set whether the scene group should be saved when changing to another scene group. The scene group switching time can be reduced if the selection of this check box is cleared, but if the power supply is turned OFF without saving changes to settings, the changes will be lost.
Scene switch time, Add time [ms]	0 to 1000 [10]	The BUSY signal is turned ON during scene switching. When this time is short and the change from ON to OFF cannot be detected by the external device, you can increase the time that the BUSY signal is ON. Set in units of 1 ms. Click [<] or [>] to increment the time in 5 ms from the displayed value.

Note

The [Save scene group on switch scene] is linked to the setting for the [Scene group switch] dialog box. Settings specified later override the previous settings.
Reference: ▶ Changing the Scene or Scene Group (p.60)

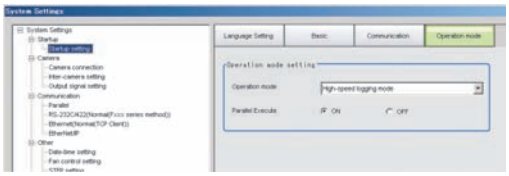
3 **Click the [Apply] button.**

Setting the Operation Mode

This section describes the operation mode (FH and FZ5-11□□ only). You can use a multi-core CPU to set the operation mode according to the application conditions. This helps to reduce the takt time and downtime. The operation mode is set in the startup settings.

Reference: ▶ Setting the Status at Startup: [Startup Settings] (p.293)

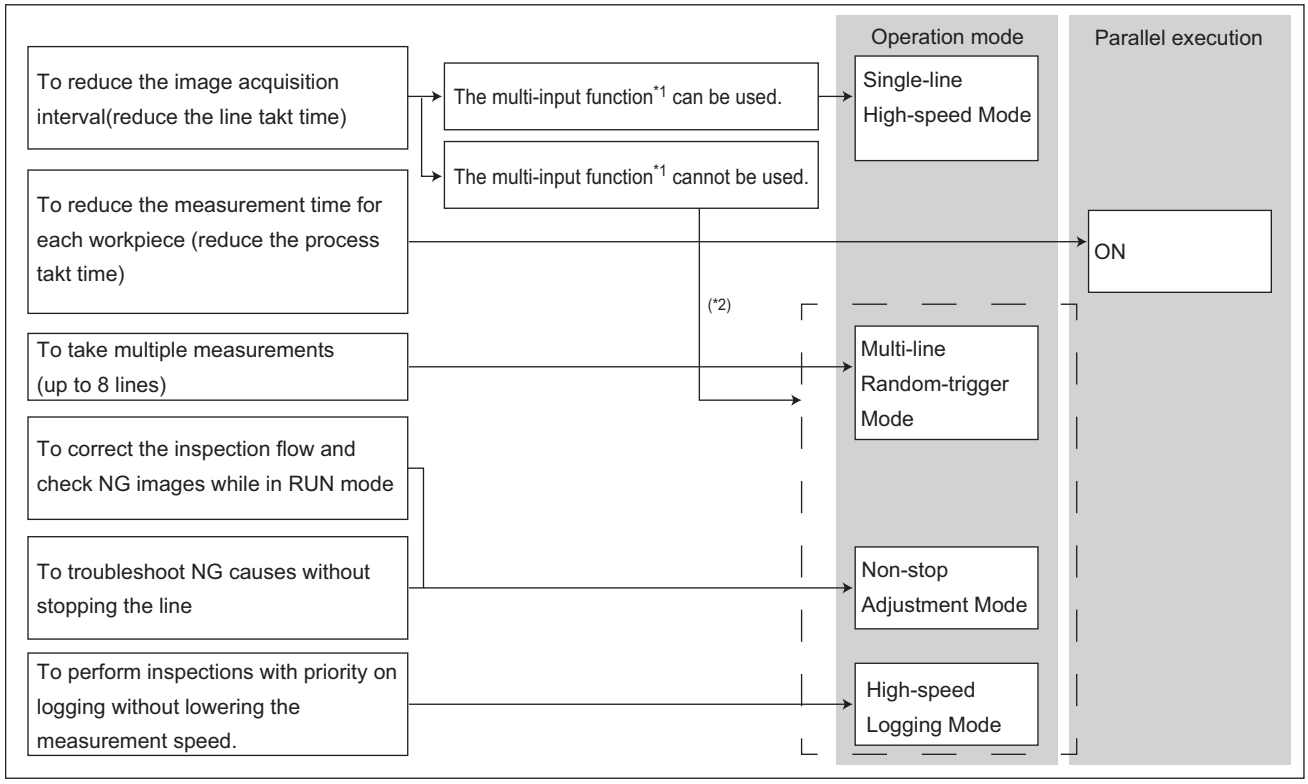
1. **In the Main Window, select [System settings] – [Startup settings] from the [Tool] menu.**
2. **Click the [Operation mode] button and select the operation mode.**
3. **Click the [Apply] Button.**



4. **In the Adjustment Window (layout 0), click [Data save] in the Toolbox Pane to save the settings data.**
5. **In the Main Window, select [System restart] from the [Function] menu.**
   The System restart dialog box is displayed.
6. **Click [OK].**

Guidelines for Selecting the Operation Mode

This section describes how to set the operation mode that best suits the purpose.



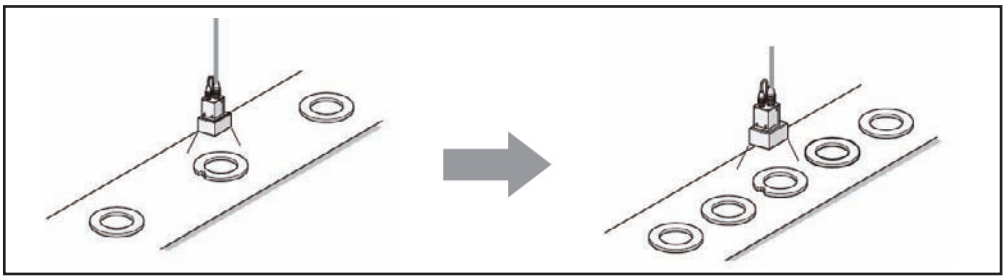
*1: The next STEP input is acknowledged as soon as image input processing is completed. There is no need to wait for measurement processing to finish. Reference: ▶ Multi-input Function (p.270)

*2: When the multi-input function cannot be used, use other operation modes than the Single-line High-speed Mode.

Single-line High-speed Mode

You can execute the measurement process with a multi-core CPU to inspect more measurement targets in the same amount of time than conventional processing would allow. This Single-line High-speed Mode uses up to four CPU cores (or eight threads) on a CPU that take turns in the execution of a single inspection flow every time the STEP signal is input. This improves the performance of the conventional multi-input function. Executing the parallel processing function on the CPU will reduce the shortest takt time by up to 1/4.

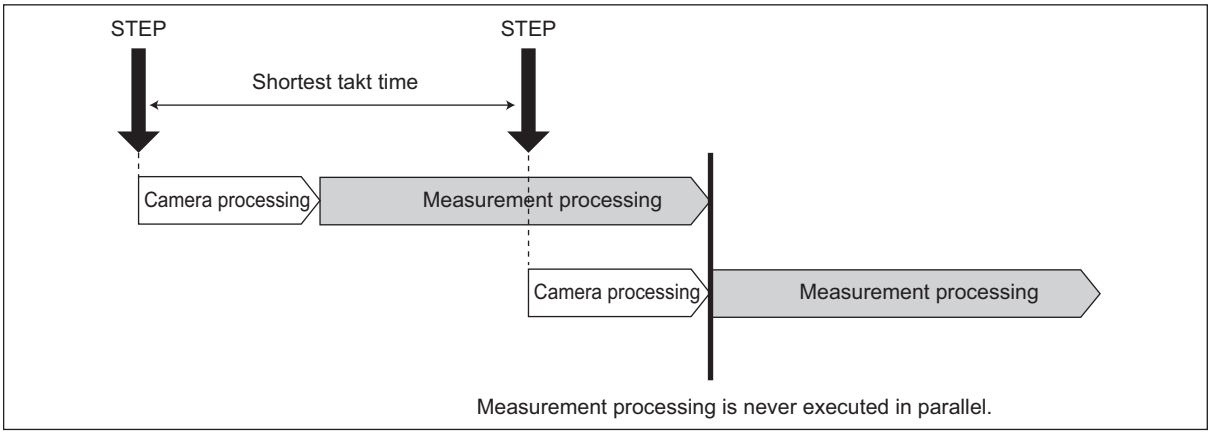
Reference: ▶ Multi-input Function (p.270)



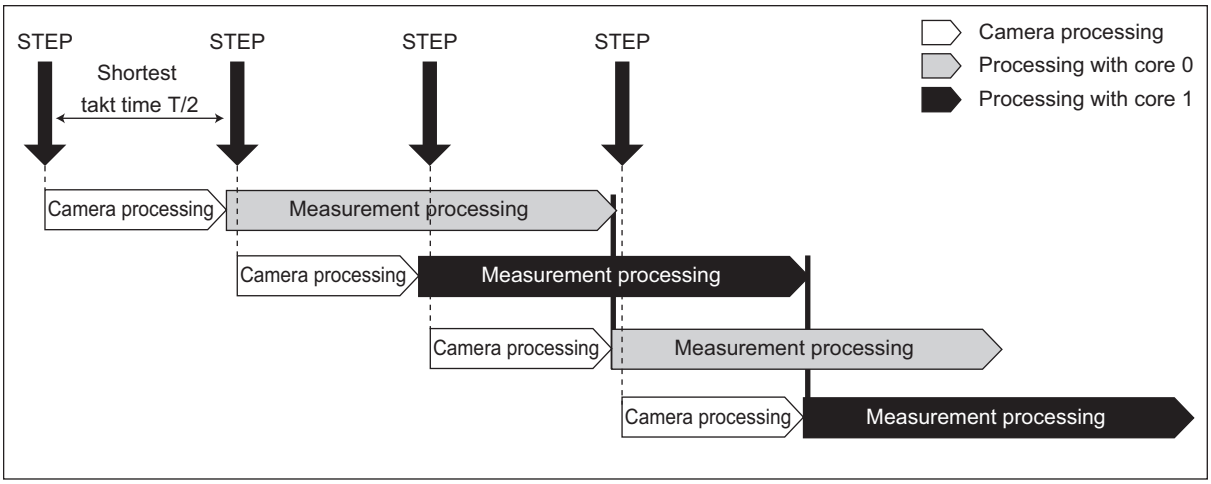
IMPORTANT

- The transaction time for a single workpiece is shorter if parallel processing is enabled to allow parallel execution of the measurement flow.
  Reference: ▶ Parallel Processing (p.282)
- It is assumed that the [Single-line High-speed mode] will be used together with the multi-input function. Refer to *Multi-input Function* before you use the Single-line High-speed Mode. When the multi-input function cannot be used, the Single-line High-speed Mode is not available. Choose another mode than the Single-line High-speed Mode.
  Reference: ▶ Multi-input Function (p.270)
- Certain processing items, such as Serial Output, Parallel Output, and Parallel Judgement Output, may slow the performance of the Single-line High-speed Mode if they are programmed in the first half of the flow. If the performance drops significantly, consider the use of parallel processing.
- Some processing items, such as Trend Monitor and Last NG Display, may not function correctly if Single-line High-speed Mode is used. Do not use them. Calculation items that use the numeric values from a previous STEP may not calculate correctly.
- You cannot use Simplified Non-stop Adjustment Mode during Single-line High-speed Mode.
- External commands to obtain and set unit data will not operate correctly. Do not use them.
- If you use external commands, the BUSY signal will remain ON during command execution longer than for other operation modes.

Example: Normal processing with a single core CPU



Example: Processing using the Single-line High-speed Mode with dual-core CPU



Multi-input Function

With the multi-input function, images are input consecutively and at high speed. This allows the next STEP to be received as soon as the image input processing is completed. There is no need to wait for measurement processing to be completed. You can check whether image input processing has been completed with the status of the READY signal.

There are no setting items for the multi-input function. You can use it as long as you can check the status of the READY signal, i.e., as long as you use parallel or EtherCAT communications. With EtherCAT communications, check the status of the Trigger Ready signal instead of the READY signal.

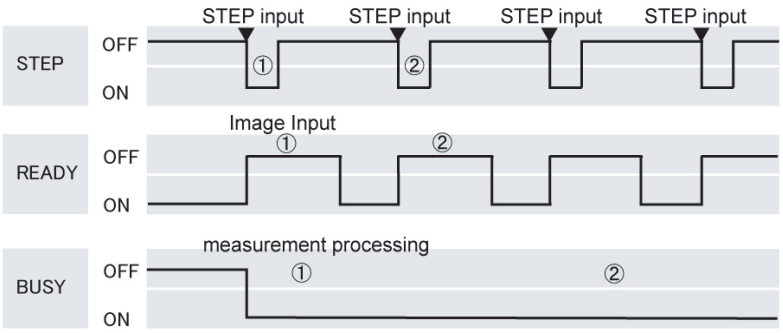
The FH and FZ5-11□□ both have a multi-core CPU. Using the Single-line High-speed Mode, where each CPU core takes turns in the execution of the measurement processing, realizes the reduction of the minimum takt time (*1) by up to 1/2 for the FZ5 and 1/4 for the FH.

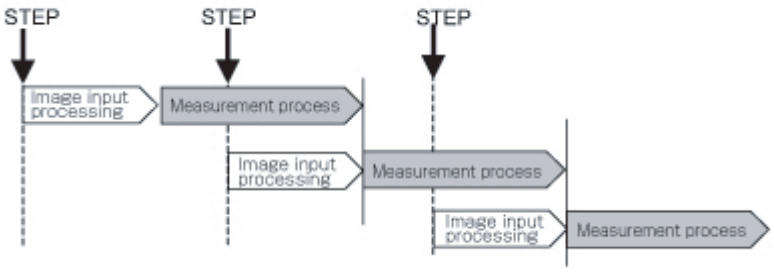
*1: The fastest takt time is the shortest time that is allowed to accept a STEP signal without causing a backlog of images.

Reference: ▶ Single-line High-speed Mode (p.269)

IMPORTANT

- Click the upper left of the Image Display Pane and set the [Image mode] to [Freeze Image].
  Reference: ▶ Changing the Image Mode and Other Display Contents (p.92)
- If images are taken consecutively at high speed, the number of images that you will be able to take will be limited. If the maximum number of images have been taken, the READY signal will not turn ON until the current processing is completed. Therefore, the next STEP signal cannot be input.
- The ERROR terminal turns ON if a STEP signal is input when the READY signal is OFF.
- If the trigger signal input rate is faster than the communications output, some STEP signals may be skipped and measurements may be delayed. Make sure to input trigger signals at a rate that does not cause a delay in communications.
- The multi-input function cannot be used in the following cases:
  - The flow has more than one camera input processing unit.
  - The flow has a Camera Image Input HDR processing unit.
  - Branching is used to process a single camera input processing unit more than one time.
  - Measurement triggers are input through means other than parallel or EtherCAT communications (non-procedure commands or measurement commands through a PLC Link).
- If a measurement trigger is input while using the multi-input function, the last NG image cannot be displayed.
- If images and data are logged to a network drive, the use of the multi-input function and other heavy measurement loads on the Controller may slow down communications and cause logging errors. In this case, set the measurement takt time so that there is some leeway.
- When the multi-input function is in use, most of the resources of the CPU are allocated to measurement processing. This may cause performance drops (delayed responses and packet losses) and communications errors. Do not use Ethernet/IP communications while the multi-input function is in use.





The following processing items are supported in the single-line high-speed mode:

OK:Supported processing item, RST: Processing item with restricted support, ---:Unsupported processing item

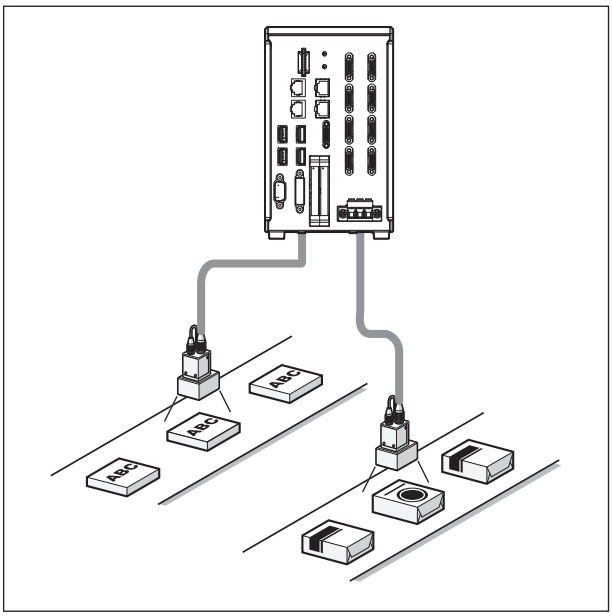
Processing item	Support	Processing item	Support	Processing item	Support
Camera Image Input	OK	Glue Bead Inspection	OK	Statistics	---
Camera Image Input FH	OK	Position Compensation	OK	Reference Calib Data	OK
Camera Image Input HDR	OK	Filtering	OK	Position Data Calculation	OK
Camera Image Input HDR Lite	OK	Back Ground Suppression	OK	Stage Data	OK
Camera Switching	OK	Brightness Correct Filter	OK	Robot Data	OK
Measurement Image Switching	OK	Color Gray Filter	OK	Vision Master Calibration	---
Search	OK	Extract Color Filter	OK	PLC Master Calibration	---
Flexible Search	OK	Anti-Color Shading	OK	Convert Position Data	OK
Sensitive Search	OK	Stripes Removal Filter II	OK	Movement Single Position	OK
ECM Search	OK	Polar Transformation	OK	Movement Multi Points	OK
EC Circle Search	OK	Trapezoidal Correction	OK	Detection Point	OK
Ec Corner	OK	Machine Simulator	OK	Camera Calibration	OK
Ec Cross	OK	Image Subtraction	RST(*1)	Data Save	OK
Shape Search II	OK	Advanced Filter	OK	Conditional Branch	OK
Classification	OK	Panorama	OK	End	OK
Edge Position	OK	Calculation	OK	DI Branch	OK
Edge Pitch	OK	Line Regression	OK	Control Flow Normal	---
Scan Edge Position	OK	Circle Regression	OK	Control Flow PLC Link	---
Scan Edge Width	OK	Precise Calibration	OK	Control Flow Parallel	---
Circular Scan Edge Position	OK	User Data	OK	Control Flow Fieldbus	---
Circular Scan Edge Width	OK	Set Unit Data	OK	Selective Branch	OK
Color Data	OK	Get Unit Data	OK	Data Output	OK
Gravity and Area	OK	Set Unit Figure	OK	Parallel Data Output	OK
Labeling	OK	Get Unit Figure	OK	Parallel Judgement Output	OK
Label Data	OK	Trend Monitor	---	Fieldbus Data Output	OK
Defect	OK	Image Logging	RST(*2)	Result Display	OK
Precise Defect	OK	Image Conversion Logging	RST(*2)	Display Image File	OK
Fine Matching	OK	Data Logging	OK	Display Last NG Image	---
Character Inspection	OK	Elapsed Time	OK	OCR	OK
Date Verification	OK	Wait	OK	OCR User Dictionary	OK
Model Dictionary	OK	Focus	OK		
2D Codes	OK	Iris	OK		
Circle Angle	OK	Unit Macro	OK		
Shape Search III	OK	Unit Calculation Macro	OK		
Intersection	OK	Parallelize	OK		
Barcode	OK	Parallelize Task	OK		

*1: The [Subtract only] measurement mode is not supported.

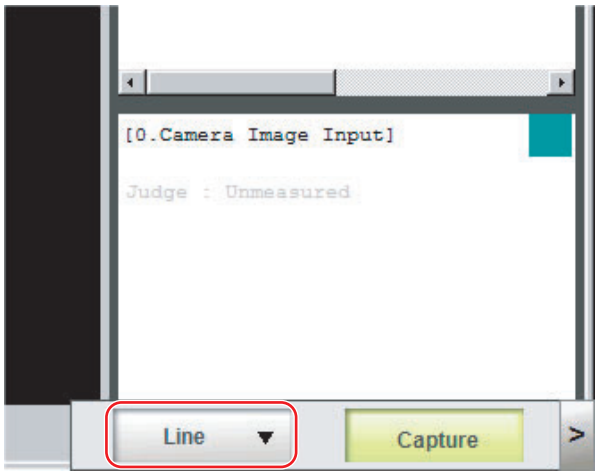
*2: Images from odd numbered measurements and from even numbered measurements are logged in separate folders.

Multi-line Random-trigger Mode

You use the Multi-line Random-trigger Mode when you want to measure more than one line with a single Controller. You can measure the inputs from different Cameras on up to eight independent lines. You can set a scene group data and scene data for each line.

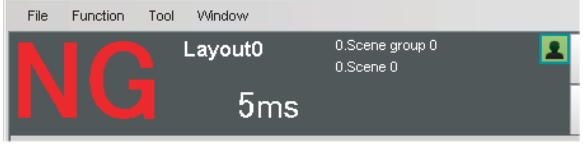


You can switch between lines with the [Line] button that is located on the Measurement Manager Bar that is displayed at the bottom right of the window.

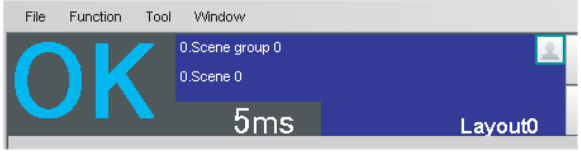


Each line has its own background color.

Line 0



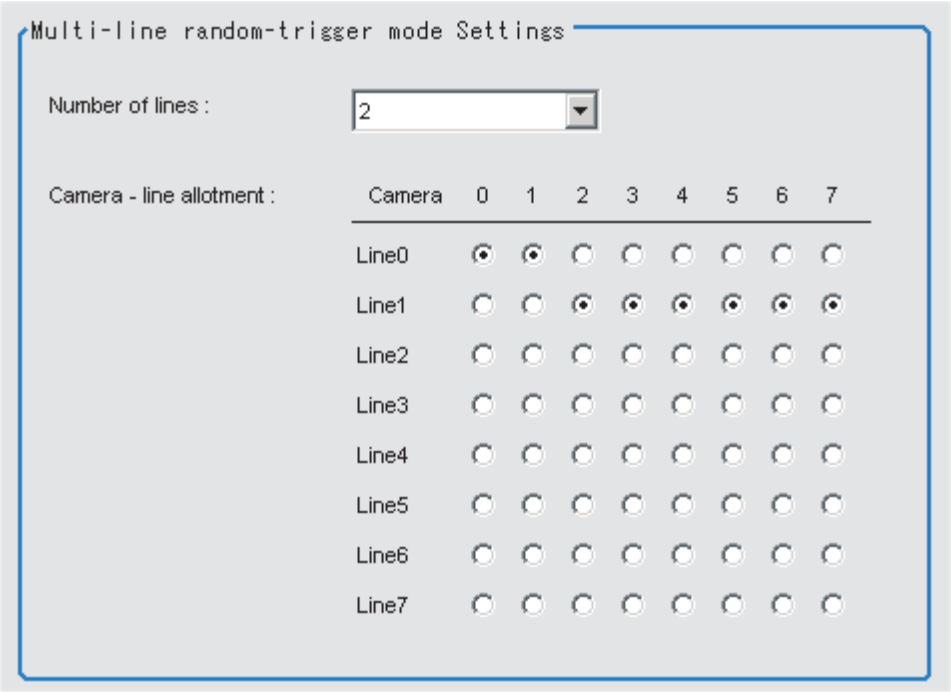
Line 1



FH Series

You can assign any of up to eight Cameras to each line.
Use the following procedure to assign Cameras to each line.

1. **In the Main Window, select [System settings] – [Startup] – [Startup settings] from the [Tool] menu.**
2. **Select [Multi-line random-trigger mode] for the operation mode on the [Operation Mode] tab page.**
3. **Set the [Number of lines] and [Camera - line allotment] on the [Multi-line random-trigger mode settings] tab page.**



For example, if you use Cameras 0 and 1 on line 0, you can use up to 6 Cameras from Camera 2 to 7 on line 2.

IMPORTANT

The settings of the [Multi-line Random-trigger Mode Settings] apply only to the FH.
With an FZ5 Sensor Controller, there will always be two lines and fixed lines assignments to the Cameras regardless of the settings of the [Multi-line Random-trigger Mode Settings].

FZ5 Series

The Camera assignments for each line are listed below.

Camera number	Software handling
Camera 0	Camera 0 on line 0
Camera 1	Camera 0 on line 1
Camera 2	Camera 1 on line 0
Camera 3	Camera 1 on line 1

Functional Limitations of Multi-line Random-trigger Mode

Keep the following points in mind when using Multi-line Random-trigger Mode.

Item		FH Series	FZ5 Series
Processing time		• If the STEP signal is input at the exact same time across lines 0 to 7, the measurement on the other lines may be delayed. • If data logging occurs at the same time for lines 0 to 7, the transaction time may take longer than usual.	• If the STEP signal is input at the exact same time for lines 0 and 1, the measurement on the other lines may be delayed (roughly by the same amount of the Camera Image Input processing unit). • If data logging occurs at the same time for lines 0 and 1, the transaction time may take longer than usual.
Number of Cameras		The maximum number of Cameras is 8 for all lines.	The maximum number of Cameras is 2 for all lines.
Error messages		Error messages are the same for lines 0 to 7. If an error occurs on any of the lines, an error message is displayed and the ERROR signal is turned ON.	Error messages are the same for lines 0 and 1. If an error occurs on any of the lines, an error message is displayed and the ERROR signal is turned ON.
Saving data in the Controller		To save data in the Controller, the data must be saved separately for each line.	
Save folder for captured images		When you click [Capture] on the Measurement Manager Bar, captured images are always saved to the folder that is specified for line 0.	
Date-time setting, language setting, and operation mode setting		These settings can be set only for line 0.	These settings can be set only for line 0.
View		Images can be displayed simultaneously for all lines. Use layout modification to set the display position and size for the Main Window for each line.	One of the lines must be selected to be displayed. Both lines cannot be displayed at the same time.
Communications	Parallel	• Parallel communications can be set only for line 0. The settings for line 0 are applied to lines 1 to 7. • Refer to *Table 1* for details on the differences between parallel I/O and parallel terminals.	• Parallel communications can be set only for line 0. The settings for line 0 are applied to line 1. • Refer to *Table 1* for details on the differences between parallel I/O and parallel terminals.
	Normal (RS-232C/422)	RS-232C can be set only for line 0. Lines 1 to 7 cannot be used.	RS-232C/422 can be set only for line 0. Line 1 cannot be used.
	Normal (Ethernet)	• To use Ethernet, use a different port number for each line. • The IP address for the Controller cannot be set for lines 1 to 7 (the IP address for line 0 is used).	• To use Ethernet, use a different port number for each line. • The IP address for the Controller cannot be set for line 1 (the IP address for line 0 is used).
	PLC Link (RS-232C/422)	RS-232C can be set only for line 0. Lines 1 to 7 cannot be used.	RS-232C/422 can be set only for line 0. Line 1 cannot be used.
	PLC Link (Ethernet)	• To use Ethernet, use a different port number for each line. • The IP address for the Controller cannot be set for lines 1 to 7 (the IP address for line 0 is used).	• To use Ethernet, use a different port number for each line. • The IP address for the Controller cannot be set for line 1 (the IP address for line 0 is used).
	EtherNet/IP	• To use EtherNet/IP, use an EDS file that matches the line to be used. • Specify different I/O memory addresses for the sending area and receiving area for line 0 and for lines 1 to 7.	• To use EtherNet/IP, use an EDS file that matches the line to be used. • Specify different I/O memory addresses for the sending area and receiving area for line 0 and for line 1.

Item		FH Series	FZ5 Series
STEP setting		The value of the STEP signal filter for the most recently set line is applied to all lines.	The value of the STEP signal filter for each line is applied to each line.
Communications	EtherCAT	• The Communications Module is set for each line. • The communications settings can be set only for line 0. The settings are applied as shown below. [Output control]: The same setting is used for all lines. [Output period]: The same setting is used for all lines. [Output setting]: This is set for each line. • The I/O ports (areas) for communications between the FH and master are assigned as shown below. I/O ports for the Command Area: I/O ports are assigned for each line. I/O ports for the Response Area: I/O ports are assigned for each line. I/O ports for the Output Area: I/O ports are assigned for each line. The I/O ports in the Sysmac Error Status Area are shared by all lines. • The maximum data size that can be output depends on the number of lines, as shown in the following table. 1 to 5 lines: 256 bytes max. 6 to 8 lines: 128 bytes max.	---

Table 1: Parallel I/O Functions and Parallel Terminals for Multi-line Mode

- **FH Series**

I/O	Number of lines			
	1 line	2 lines	3 to 4 lines	5 to 8 lines
STEP	STEP0	STEP0 or STEP1	STEP0 to STEP3	STEP0 to STEP7
DSA	DSA0	DSA0 or DSA1	No output	
DI	DI0 to DI7	DILINE0, DI0 to DI7 (shared by all lines)	DILINE0 to DILINE1, DI0 to DI7 (shared by all lines)	DILINE0 to DILINE2, DI0 to DI7 (shared by all lines)
ENC (phases A, B, and Z)	ENC0 phase A, ENC0 phase B, and ENC0 phase Z	ENC0 phase A, ENC1 phase A, ENC0 phase B, ENC1 phase B, ENC0 phase Z, and ENC1 phase Z	No output	
ACK	ACK (shared by all lines)			
STGOUT/ SHTOUT	STGOUT0/ SHTOUT0	STGOUT0/SHTOUT0 or STGOUT1/SHTOUT1	STGOUT0 to STGOUT3 or SHTOUT0 to SHTOUT3	STGOUT0 to STGOUT7 or SHTOUT0 to SHTOUT7
RUN	RUN0	RUN0 or RUN1	RUN0 to RUN3	No output
GATE	GATE0	GATE0 and GATE1	No output	
BUSY	BUSY0	BUSY0 or BUSY1	BUSY0 to BUSY3	BUSY0 to BUSY7
OR	OR0	OR0 or OR1	OR0 to OR3	OR0 to OR7

I/O	Number of lines			
	1 line	2 lines	3 to 4 lines	5 to 8 lines
ERROR	ERROR0	ERROR0 or ERROR1	ERROR0 to ERROR3	ERROR (shared by all lines)
READY	READY0	READY0 or READY1	READY0 to READY3	READY0 to READY7
DO	DO0 to DO15	Line 0: D00 to D07 Line 1: D8 to D15	No output	

- **FZ5 Series**

I/O	Number of lines	
	1 line	2 lines
STEP	STEP0	STEP0 or STEP1
DSA	DSA0	DSA0 or DSA1
DI	DI0 to DI7	DI0 to DI7
RUN	RUN	None (This input is assigned as the BUSY signal for line 1.)
GATE	GATE0	GATE0 and GATE1
BUSY	BUSY	BUSY or RUN
OR	OR0	OR0 or OR1
ERROR	ERROR	ERROR (shared by all lines)
READY	READY0	READY0 or READY1
DO	DO0 to DO15	Line 0: D00 to D07 Line 1: D8 to D15

The following processing items are supported in the multi-line random trigger mode:

OK:Supported processing item, RST: Processing item with restricted support, ---:Unsupported processing item

Processing item	Support	Processing item	Support	Processing item	Support
Camera Image Input	OK	Glue Bead Inspection	OK	Statistics	OK
Camera Image Input FH	OK	Position Compensation	OK	Reference Calib Data	OK
Camera Image Input HDR	OK	Filtering	OK	Position Data Calculation	OK
Camera Image Input HDR Lite	OK	Back Ground Suppression	OK	Stage Data	OK
Camera Switching	OK	Brightness Correct Filter	OK	Robot Data	OK
Measurement Image Switching	OK	Color Gray Filter	OK	Vision Master Calibration	OK
Search	OK	Extract Color Filter	OK	PLC Master Calibration	OK
Flexible Search	OK	Anti-Color Shading	OK	Convert Position Data	OK
Sensitive Search	OK	Stripes Removal Filter II	OK	Movement Single Position	OK
ECM Search	OK	Polar Transformation	OK	Movement Multi Points	OK
EC Circle Search	OK	Trapezoidal Correction	OK	Detection Point	OK
Ec Corner	OK	Machine Simulator	OK	Camera Calibration	OK
Ec Cross	OK	Image Subtraction	RST(*1)	Data Save	OK
Shape Search II	OK	Advanced Filter	OK	Conditional Branch	OK
Classification	OK	Panorama	OK	End	OK
Edge Position	OK	Calculation	OK	DI Branch	RST(*2)
Edge Pitch	OK	Line Regression	OK	Control Flow Normal	OK
Scan Edge Position	OK	Circle Regression	OK	Control Flow PLC Link	OK
Scan Edge Width	OK	Precise Calibration	OK	Control Flow Parallel	RST(*2)
Circular Scan Edge Position	OK	User Data	OK	Control Flow Fieldbus	OK
Circular Scan Edge Width	OK	Set Unit Data	OK	Selective Branch	OK
Color Data	OK	Get Unit Data	OK	Data Output	OK
Gravity and Area	OK	Set Unit Figure	OK	Parallel Data Output	RST(*2)
Labeling	OK	Get Unit Figure	OK	Parallel Judgement Output	RST(*2)
Label Data	OK	Trend Monitor	OK	Fieldbus Data Output	OK
Defect	OK	Image Logging	OK	Result Display	OK
Precise Defect	OK	Image Conversion Logging	OK	Display Image File	OK
Fine Matching	OK	Data Logging	OK	Display Last NG Image	OK
Character Inspection	OK	Elapsed Time	OK	OCR	OK
Date Verification	OK	Wait	OK	OCR User Dictionary	OK
Model Dictionary	OK	Focus	OK		
2D Codes	OK	Iris	OK		
Circle Angle	OK	Unit Macro	OK		
Shape Search III	OK	Unit Calculation Macro	OK		
Intersection	OK	Parallelize	OK		
Barcode	OK	Parallelize Task	OK		

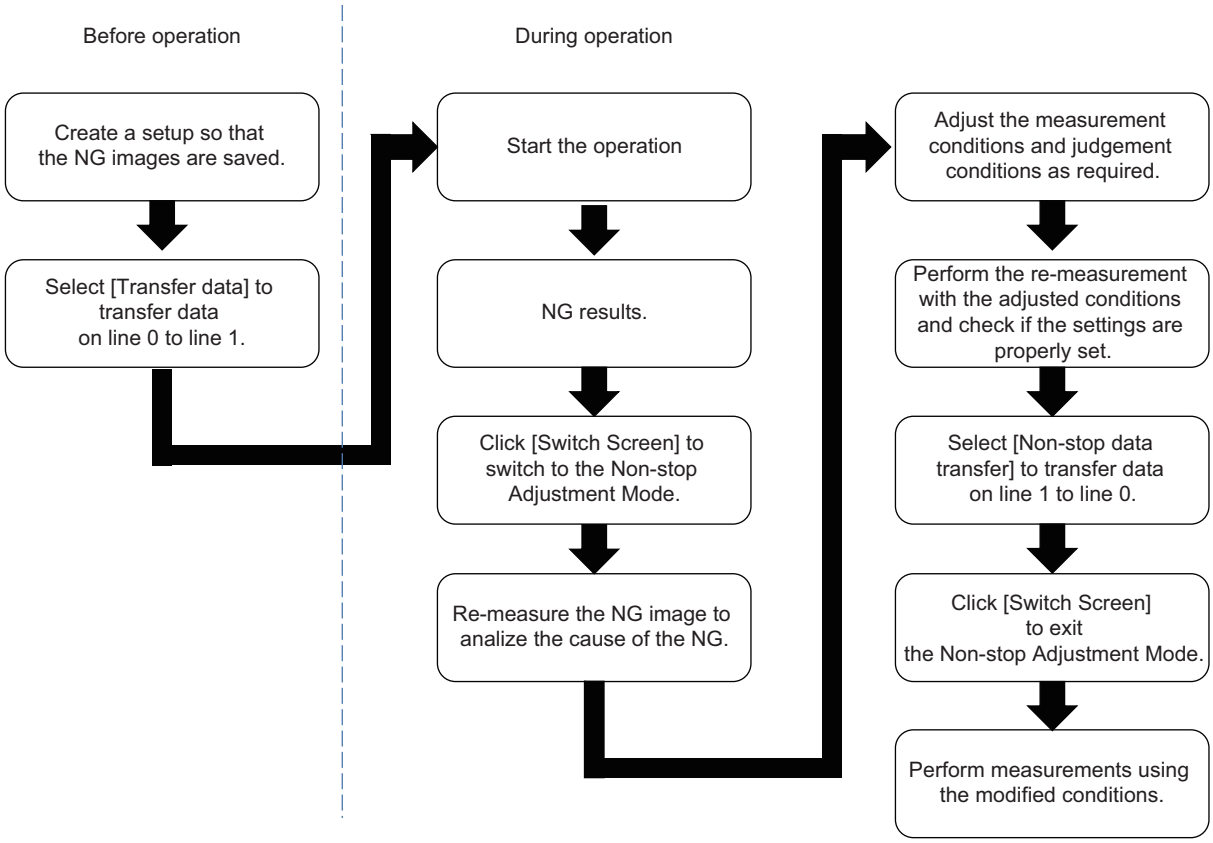
*1: [DI Register] in the measurement mode is not supported.

*2: Not supported for 3 or more lines.

Non-stop Adjustment Mode

Non-stop Adjustment Mode allows you to change and adjust the measurement flow without having to stop the measurement processing during operation.
You use saved image files to adjust measurement flows. You can apply a modified measurement flow while the Controller is in RUN mode.
Lines 0 and 1 are used simultaneously for the execution of the Non-stop Adjustment Mode.
While the measurement process is executed for line 0, you modify and adjust the measurement flow for line 1.

Typical Application for Non-stop Adjustment Mode



1 **In the Main Window for line 0, select [Transfer data] from the [Function] menu.**

The settings for line 0 are transferred to line 1.
Measurement does not stop and continues to be executed on line 0.

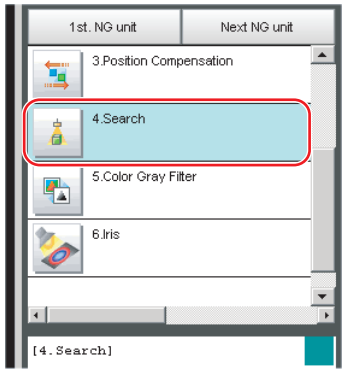
2 **Click the [Switch Screen] button to display line 1 (Non-stop Adjustment Mode Window).**

The display for line 0 changes to the display for line 1. (The main display for line 1 is called the Non-stop Adjustment Mode Window.) This display change will not stop the measurements on images input to line 0.

3 **Click the icon portion of the processing unit to adjust.**

To change the flow, use the [Edit flow] tool in the Toolbox Pane.

The settings dialog box for the selected unit is displayed.



4 **Modify the processing units.**

5 **Click [OK].**

The settings dialog box closes and the display returns to the Main Window (layout 1).
If you need to modify the judgement conditions for more than one processing unit, repeat steps 3 to 4.
At this stage, none of the changes have been applied.

6 **In the Main Window for line 1, select [Non-stop data transfer] from the [Function] menu.**

The changes to the settings for line 1 are saved as the settings for line 0 without affecting the measurement.

7 **Click the [Switch Screen] button to display the Main Window for line 0.**

IMPORTANT

- When you execute [Non-stop data transfer], the results in the Trend Monitor and Expression processing units are cleared.
- If you use an external command to switch scenes or scene groups, or change the settings of processing units while the Controller is in RUN mode, the results of those changes will not be applied when you change to line 1 (Adjustment Mode) in the Main Window.
- If you use the Non-stop Adjustment Mode after you change the scene group while the Controller is in RUN mode, you may overwrite the scene group data unintentionally.
- The only communications commands that are accepted during a non-stop data transfer are the measurement (parallel, non-procedure, or PLC Link) and continuous measurement (parallel only) commands.
- The larger the size of the scene group file, the longer the non-stop data transfer will take.
- The communications settings cannot be changed in Non-stop Adjustment Mode.
- Do not register a new Camera Image Input processing unit in the Non-stop Adjustment Mode Window.
- If there is not enough external memory when you are about to do a non-stop data transfer, the transfer may not start. Set the destination for image logging to a destination that does not use external memory.
- Do not change the image mode for Non-stop Adjustment Mode.
- If you use image logging in Non-stop Adjustment Mode, the non-stop data transfer may become unavailable. To avoid this, set the trigger frequency so that it is longer than the logging time.
- Do not change the settings for image input processing items and do not change Camera parameters in the system settings in Non-stop Adjustment Mode.

The following processing items are supported in Non-Stop Adjustment Mode:

OK:Supported processing item, RST: Processing item with restricted support, ---:Unsupported processing item

Processing item	Support	Processing item	Support	Processing item	
Camera Image Input	---	Glue Bead Inspection	OK	Statistics	---
Camera Image Input FH	---	Position Compensation	OK	Reference Calib Data	OK
Camera Image Input HDR	---	Filtering	OK	Position Data Calculation	OK
Camera Image Input HDR Lite	---	Back Ground Suppression	OK	Stage Data	OK
Camera Switching	OK	Brightness Correct Filter	OK	Robot Data	OK
Measurement Image Switching	OK	Color Gray Filter	OK	Vision Master Calibration	---
Search	OK	Extract Color Filter	OK	PLC Master Calibration	---
Flexible Search	OK	Anti-Color Shading	OK	Convert Position Data	OK
Sensitive Search	OK	Stripes Removal Filter II	OK	Movement Single Position	OK
ECM Search	OK	Polar Transformation	OK	Movement Multi Points	OK
EC Circle Search	OK	Trapezoidal Correction	OK	Detection Point	OK
Ec Corner	OK	Machine Simulator	OK	Camera Calibration	OK
Ec Cross	OK	Image Subtraction	RST(*1)	Data Save	OK
Shape Search II	OK	Advanced Filter	OK	Conditional Branch	OK
Classification	OK	Panorama	OK	End	OK
Edge Position	OK	Calculation	OK	DI Branch	---
Edge Pitch	OK	Line Regression	OK	Control Flow Normal	RST(*3)
Scan Edge Position	OK	Circle Regression	OK	Control Flow PLC Link	RST(*3)

OK:Supported processing item, RST: Processing item with restricted support, ---:Unsupported processing item

Processing item	Support	Processing item	Support	Processing item	
Scan Edge Width	OK	Precise Calibration	OK	Control Flow Parallel	RST(*3)
Circular Scan Edge Position	OK	User Data	OK	Control Flow Fieldbus	RST(*3)
Circular Scan Edge Width	OK	Set Unit Data	OK	Selective Branch	OK
Color Data	OK	Get Unit Data	OK	Data Output	OK
Gravity and Area	OK	Set Unit Figure	OK	Parallel Data Output	OK
Labeling	OK	Get Unit Figure	OK	Parallel Judgement Output	OK
Label Data	OK	Trend Monitor	---	Fieldbus Data Output	OK
Defect	OK	Image Logging	OK	Result Display	OK
Precise Defect	OK	Image Conversion Logging	RST(*2)	Display Image File	OK
Fine Matching	OK	Data Logging	RST(*2)	Display Last NG Image	OK
Character Inspection	OK	Elapsed Time	OK	OCR	OK
Date Verification	OK	Wait	OK	OCR User Dictionary	OK
Model Dictionary	OK	Focus	OK		
2D Codes	OK	Iris	OK		
Circle Angle	OK	Unit Macro	OK		
Shape Search III	OK	Unit Calculation Macro	OK		
Intersection	OK	Parallelize	OK		
Barcode	OK	Parallelize Task	OK		

*1: The [Subt./Reg.] and [DI Register] measurement modes are not supported.

*2: Re-measured images in Non-Stop Adjustment Mode are logged.

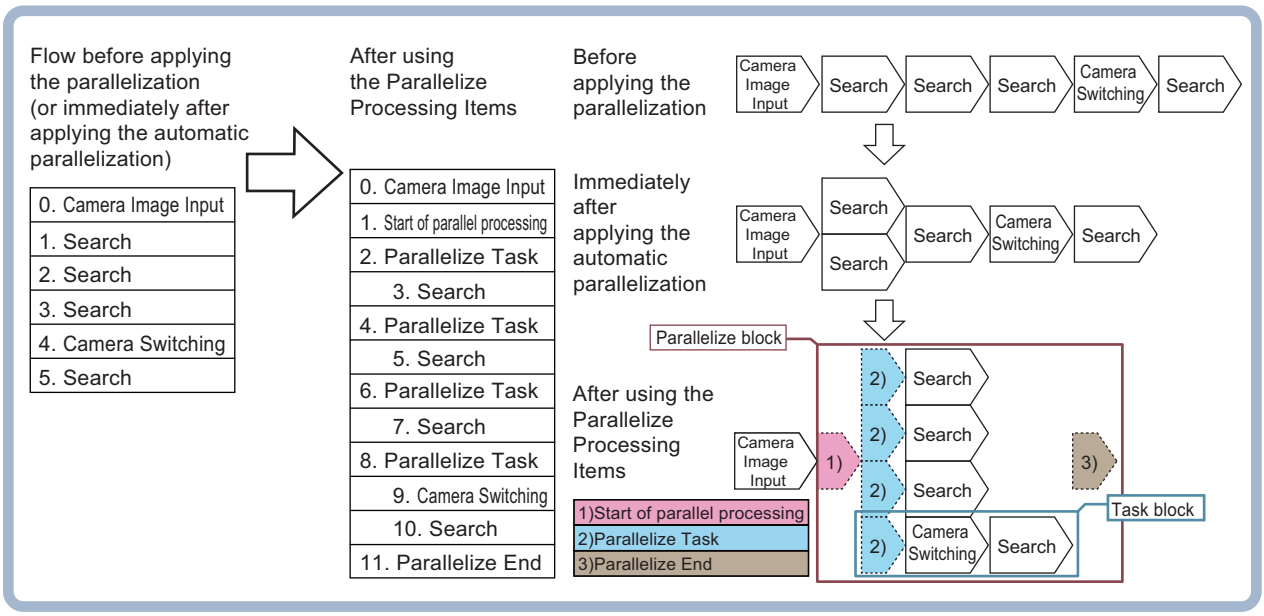
*3: Sensor Controller in Non-Stop Adjustment Mode does not stand by.

High-speed Logging Mode

Normally, the sensor controller executes measurements, image logging, and image display in a designated order. The FH and FZ5-11□□-series controllers execute those processes with one of the CPU cores, and uses the rest of the cores for measurement processing. This ensures that measurements are always processed at maximum performance.

Parallel Processing

There are two types of processings in the Parallel processing: The automatic parallelization and the manual parallelization
The automatic parallelization automatically parallelize a measurement flow if [Parallel Execute] under [Operation mode setting] is turned ON.
The manual parallelization allows you to use the processing items for the parallel processings to freely edit a measurement flow after automatic parallelization. If you use the manual parallelization to separate a part of the measurement flow into multiple tasks and process each of these tasks in parallel reduces the processing time of the measurements.



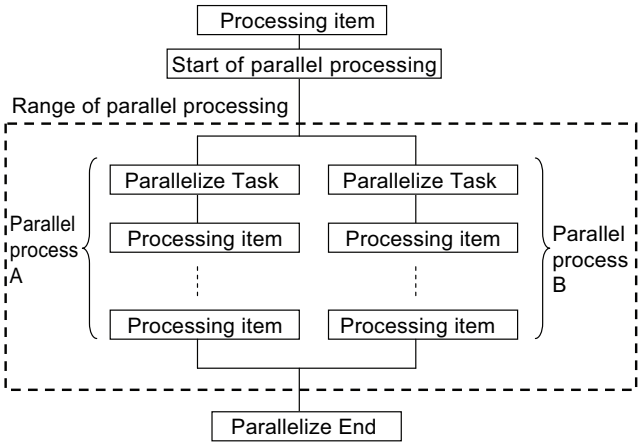
Parallel Processing Settings (Automatic Parallelization)

Parallel processing is available when [Parallel Execute] is set to ON in the [Operation mode setting].
If you turn ON [Parallel Execute] under [Operation mode setting], the automatic parallelization automatically classify the type of processing items assigned to the measurement flow and parallelize the items of the same classification.
Refer to the following for details on the processing items that are automatically parallelized.
Reference: ▶ Processing Items That Supports Automatic Parallelization (p.289)

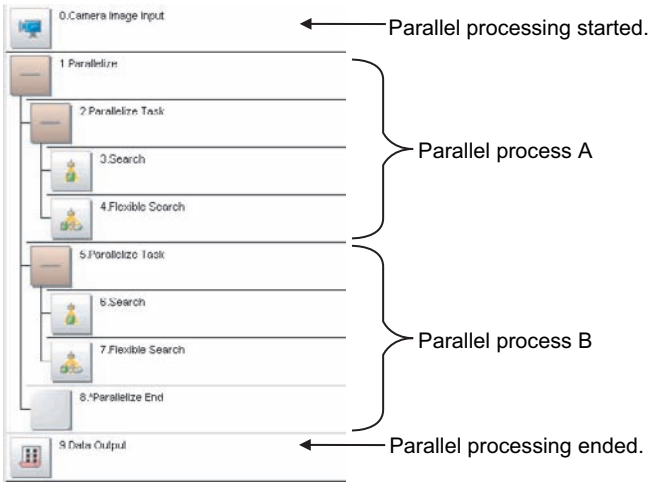
Parallel Processing Settings (Manual Parallelization)

If you use the FH/FZ5-□□-series sensor controller, you will need to convert the measurement flow to the parallelized flow in advance to specify the parallel processings. The conversion can be done by applying the automatic parallelization to the flow and use the parallel processing items.
To specify this yourself, add parallel items as shown in the following measurement flow.

Concept of a Parallelized Measurement Flow



Actual Measurement Flow



- The range from the Parallelize processing item to the Parallelize End processing item is called a *parallel block*.
- The processing range of the Parallelized Task processing item is called a *task block*. A task block is a group of measurement processes that are distributed among multi-core CPU.
  In a measurement flow, the task block is the range from the Parallelize Task processing item to the next Parallelize Task processing item, or the point that comes just before the Parallelize End processing item.

IMPORTANT

With the FZ5-L35□/6□□, you cannot use the operation modes or parallel processing.

Note

If [Parallel process] is turned OFF in the [Operation mode setting], the measurement flow is processed in series in order of the unit numbers.

Processing Items for Parallel Processing

You use the following three items.

Processing item category	Processing item name	Description
Measurement support processing items (Inspection and measurement support items)	Parallelize	Marks the beginning of a parallel processing range.
	Parallelize Task	Gives the range to be parallelized.
	Parallelize End	Marks the end of a parallel processing range.

Reference: Refer to the *Vision System FH/FZ5 Series Processing Item Function Reference Manual* (Cat No. Z341) for information on the parameters for the above items.

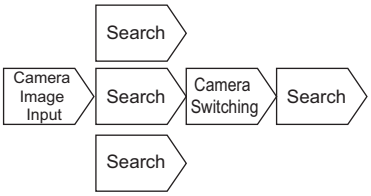
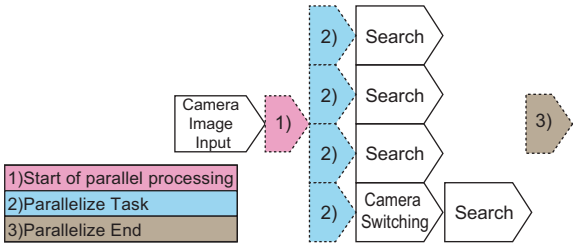
Programming Parallel Processing Items

- Always use the Parallelize and Parallelize End processing items in pairs.
- Do not place processing items between the Parallelize processing item and Parallelize Tack processing item. Any processing units that are placed in this position will not be executed.
- Always insert the Parallelize Task processing item between the Parallelize processing item and Parallelize End processing item.
- If you program a branch under a Parallelize Task processing item, make sure the branch stays within the range of the Parallelize Task processing item.

Specific Example of Parallel Processing

You can use the automatic parallelization in conbination with the manual parallelization to parallelize the following example flow. First, use the automatic parallelization to parallelize three inspection items, 3. Search, 5. Search, and 7. Search. Then, use the manual parallelization to parallelize two inspection items, 9. Camera Switching and 10. Search.

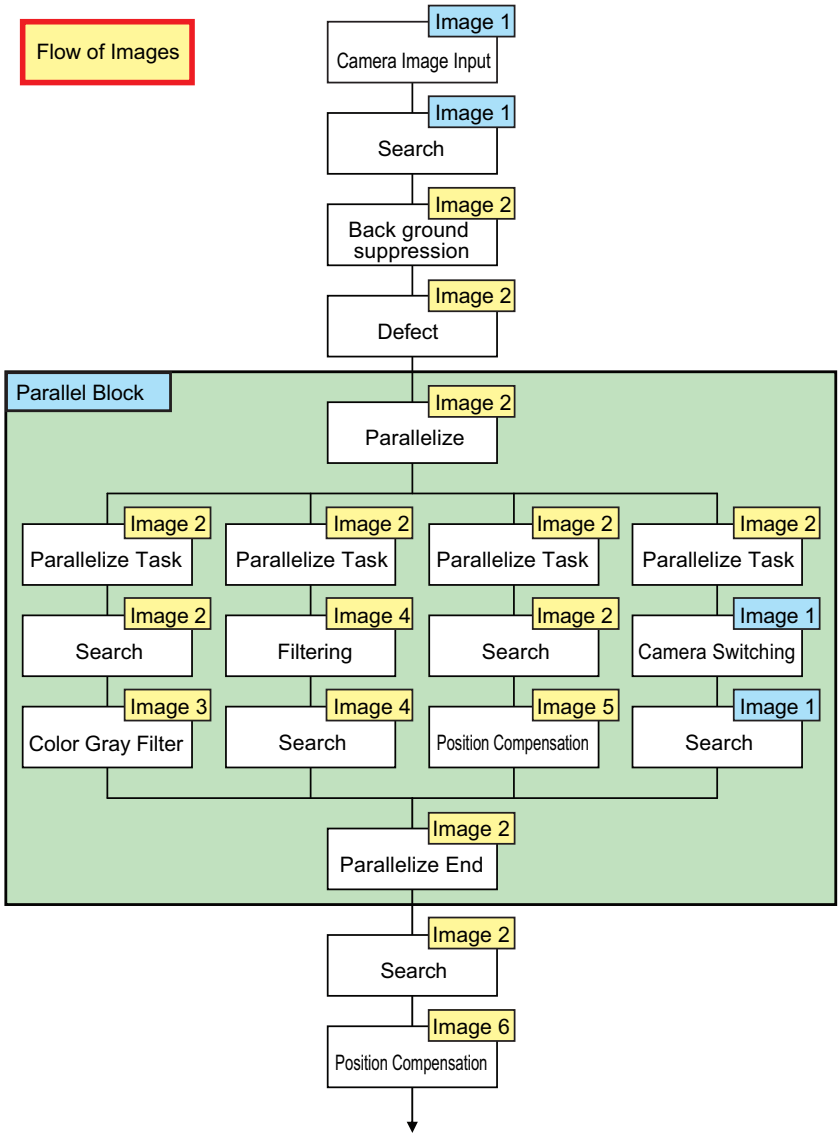
Example:

Parallelization processing application status	Measurement flow	Flow of processing	Remarks
Before applying the parallelization (automatic parallelization: OFF, manual parallelization: not applied)	0. Camera Image Input 1. Search 2. Search 3. Search 4. Camera Switching 5. Search		
Applying the automatic parallelization only	Same as above.		When using four CPU cores on the FH sensor controller
Applying the automatic parallelization in conbination with the manual parallelization	0. Camera Image Input 1. Start of parallel processing 2. Parallelize Task 3. Search 4. Parallelize Task 5. Search 6. Parallelize Task 7. Search 8. Parallelize Task 9. Camera Switching 10. Search 11. Parallelize End		

Restrictions

Flow of Images

- The image at the start of parallel processing is passed on to the next item for use in each task block.
- The image for the start of parallel operation is also the image that is used after the Parallelize End processing item.

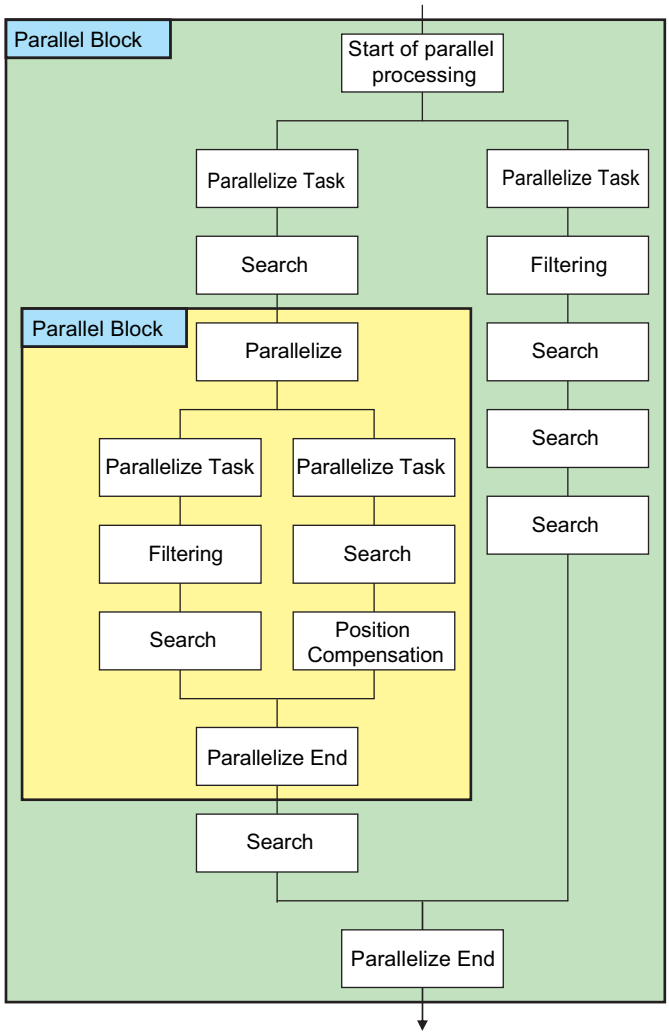


- Note that you cannot assign any Camera Image Input processing item within task blocks.
  To change the Camera image, you must use the Camera Switching processing item and set a different Camera number.

Nesting

- Parallel blocks can be nested.

Example:



Processing Order

The processing order for Parallelize Task processing items is not defined. Do not program a measurement flow that relies on a specific processing order of Parallelize Task processing items.

Transaction Times

Parallel processing may not be able to reduce transaction times as effectively as expected due to the processing time of each task, and the number of CPU cores. Before commissioning the system, always check the transaction time with the actual measurement flow and image.

Inserting Camera Image Input Processing Items

- Do not insert a Camera Image Input processing item inside the range of a Parallelize Task processing item. To use a different Camera image for a measurement under a Parallelize Task processing item, use the Camera Switching processing item inside the range of the Parallelize Task processing item.
- When inside the range of a Parallelize Task processing item, do not reference an image that is processed under a different Parallelize Task processing item.

Getting and Setting Data

- Do not get or set user data and system data inside task blocks.
- When inside the range of a Parallelize Task processing item, do not get or set unit data from a processing unit that is under a different Parallelize Task processing item.
- Do not input or output to the same file from a processing unit that is under a different Parallelize Task processing item.

Accessing the Same File

Do not input or output to the same file that is used for data logging in a task block.

Setting Procedures

This section describes the settings that are required to enable parallel processing.

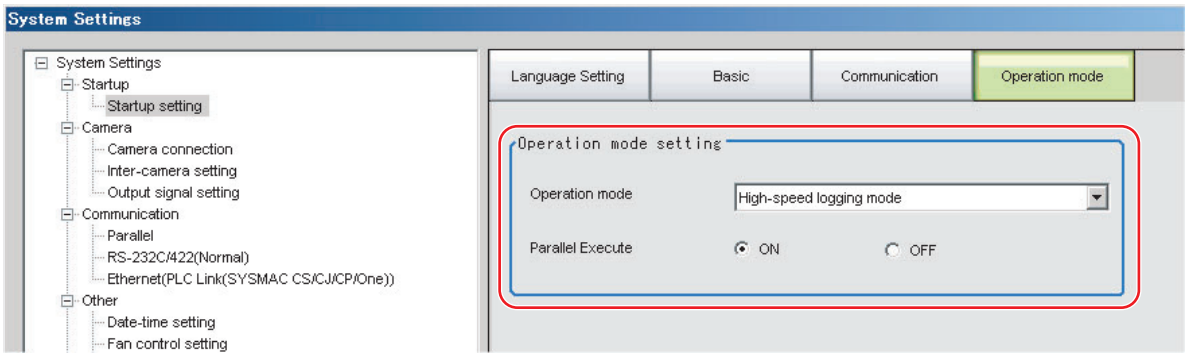
Setting the Operation Mode

Use the following procedure to set [Parallel processing] or ON in the [Operation mode setting].

1 **In the Main Window (layout 0), select [Tool] – [System Settings].**

2 **In the [Startup setting], click the [Operation mode] tab.**

3 **Select [ON] for [Parallel Execute].**



Note

With an FZ5 Sensor Controller, the functions of the operation modes may be affected if you turn ON parallel processing. If you log everything with High-speed Logging Mode, turn OFF parallel processing to take advantage of the features of the operation mode. If you use parallel processing with an FZ5 Sensor Controller, we recommend that you turn ON parallel processing in High-speed Logging Mode and then do not use the logging function, which is a feature of the operation mode.

Editing the Measurement Flow

1 **In the Main Window (layout 0), select [Function] – [Edit Flow].**

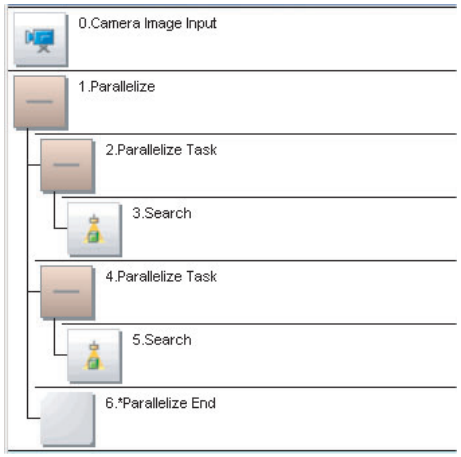
2 **Drag the following processing items from under [Inspection and measurement support items], or click the [Insert] button.**

- Parallelize
- Parallelize Task

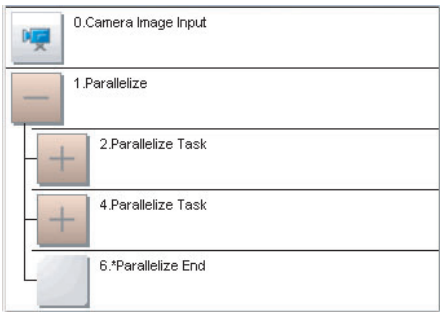
3 **Program the processing items to execute in parallel between two Parallelize Task processing items.**

Example:

Fully Expanded



Parallel Blocks Collapsed



The Parallelize processing item is the parent, and everything up to the Parallelize End processing item is a child. Additionally, the Parallelize Task processing item is the parent, and everything up to the step just before the next Parallelize Task processing item or the Parallelize End processing item is a child.

There is no settings dialog box for Parallelize processing items.
The Parallelize End processing item is not displayed in the list of items. (It is registered as a set with the Parallelize processing item).

Precautions on Flow Editing

- Editing operations on parallel blocks and task blocks, such as register, move, delete, copy, and paste, and unit save and load operations, are performed on the parallel block as a group.
- You cannot delete a Parallelize End processing item by itself. It is deleted as a set with the Parallelize processing item.
- You cannot paste or move items within the same parallel block.
- If a processing item in a parallel block is set to reference data in a processing unit that is inside the block, and if that block is copied, the reference will be replaced with a reference to the unit in the block at the copy destination.
  References to a unit that is outside the parallel block will be retained.
- You cannot move, paste, or load a task block outside of its parallel block.

Processing Items That Supports Automatic Parallelization

The following table lists the processing items that support the automatic parallelization.

Processing item	Parallelize processing	Processing item	Parallelize processing	Processing item	Parallelize processing
Camera Image Input	---	Glue Bead Inspection	OK	Statistics	---
Camera Image Input FH	---	Position Compensation	---	Reference Calib Data	---
Camera Image Input HDR	---	Filtering	---	Position Data Calculation	---
Camera Image Input HDR Lite	---	Back Ground Suppression	---	Stage Data	---
Camera Switching	---	Brightness Correct Filter	---	Robot Data	---
Measurement Image Switching	---	Color Gray Filter	---	Vision Master Calibration	---
Search	OK	Extract Color Filter	---	PLC Master Calibration	---
Flexible Search	OK	Anti-Color Shading	---	Convert Position Data	---
Sensitive Search	OK	Stripes Removal Filter II	---	Movement Single Position	---
ECM Search	OK	Polar Transformation	---	Movement Multi Points	---
EC Circle Search	OK	Trapezoidal Correction	---	Detection Point	---
Ec Corner	OK	Machine Simulator	---	Camera Calibration	---
Ec Cross	OK	Image Subtraction	---	Data Save	---
Shape Search II	OK	Advanced Filter	---	Conditional Branch	---
Classification	OK	Panorama	---	End	---
Edge Position	OK	Calculation	---	DI Branch	---
Edge Pitch	OK	Line Regression	---	Control Flow Normal	---
Scan Edge Position	OK	Circle Regression	---	Control Flow PLC Link	---
Scan Edge Width	OK	Precise Calibration	---	Control Flow Parallel	---
Circular Scan Edge Position	OK	User Data	---	Control Flow Fieldbus	---
Circular Scan Edge Width	OK	Set Unit Data	---	Selective Branch	---
Color Data	OK	Get Unit Data	---	Data Output	---
Gravity and Area	OK	Set Unit Figure	---	Parallel Data Output	---
Labeling	OK	Get Unit Figure	---	Parallel Judgement Output	---
Label Data	---	Trend Monitor	---	Fieldbus Data Output	---
Defect	OK	Image Logging	---	Result Display	---
Precise Defect	OK	Image Conversion Logging	---	Display Image File	---
Fine Matching	OK	Data Logging	---	Display Last NG Image	---
Character Inspection	OK	Elapsed Time	---	OCR	OK
Date Verification	---	Wait	---	OCR User Dictionary	---
Model Dictionary	---	Focus	OK		
2D Codes	---	Iris	OK		
Circle Angle	OK	Unit Macro	---		
Shape Search III	---	Unit Calculation Macro	---		
Intersection	OK	Parallelize	---		
Barcode	OK	Parallelize Task	---		

Note

Some processing items are processed in parallel as individual units.

Troubleshooting

Symptom	Correction
An error message is displayed on the console.	Refer to the error message list.
Insertion position for a unit to parallelize	Do not place items between the Parallelize item and Parallelize Task item. Any process units that are placed in that position will not be executed.
Insertion position of a parallelize processing items	Always insert the Parallelize Task processing item between Parallelize and Parallelize End processing items.
Setting the Parallelize processing item	Always use Parallelize and Parallelize End processing items in pairs.
Camera Image Input processing items are not being processed in parallel	Do not insert Camera Image Input processing items under Parallelize Tasks processing items.
	To use a different Camera image for a measurement under a Parallelize Task processing item, use the Camera Switching processing item under the Parallelize Task processing item.
Getting and setting the image from more than one Parallelize Task processing item	Do not reference an image that is processed under a different Parallelize Task processing item.
Getting and setting data from more than one Parallelize Task processing item	Do not get or set figure data from a processing unit that is inside the range of a different Parallelize Task processing item.
	Do not get or set unit data from a processing unit that is inside the range of a different Parallelize Task processing item.
Accessing files from under more than one Parallelize Task processing item	Do not input or output to the same file from a processing unit that is under a different Parallelize Task processing item.
Conditional branching under a Parallelize Task processing item	If you need to branch the processing items inside the range of a Parallelize Task processing item, make sure the branch stays within the range of the Parallelize Task processing item.
Processing order of Parallelize Tasks	The processing order for the Parallelize Task processing units is not defined.
	Do not program a measurement flow that relies on a specific processing order of Parallelize Task processing items.
Improving processing speed with Parallelize Task processing item	Parallel processing may not reduce transaction times as much as expected due to the processing time of each task, and the number of CPU cores.
	Before commissioning the system, always check the transaction time with the actual measurement flow and image.
Getting and setting system data under Parallelize Task processing items	Do not get or set user data and system data within the range of a Parallelize Task processing item.

Setting the Operating Environment of the System

This section describes how to set the operating environment of the Controller. The following settings are available.

- **Reference: ▶ Setting the Date and Time: [Date-time Setting] (p.291)**
- **Reference: ▶ Selecting the Language: [Language Setting] (p.292)**
- **Reference: ▶ Setting the Fan Rotation Speed: [Fan Control Setting] (p.292)**
- **Reference: ▶ Setting the Status at Startup: [Startup Settings] (p.293)**
- **Reference: ▶ Setting the Encoder Trigger: [Encoder Trigger Setting] (p.296)**
- **Reference: ▶ Setting the STEP Input Detection Pulse Width: [STEP Setting] (p.297)**
- **Reference: ▶ Setting a Network Drive: [Network Drive Setting] (p.298)**
- **Reference: ▶ Setting Up a Connection to the External Device**

  This section describes communications with external devices, such as computers and programmable controllers.
  Refer to the *Vision System FH/FZ5 Series User's Manual for Communications Settings* (Cat No. Z342) for details.

The Controller model and measurement application version can also be checked.

- **Reference: ▶ Checking System Information: [System Information] (p.300)**

Setting the Date and Time: [Date-time Setting]

Confirm that the date and time on the built-in calendar are correct, and make corrections if they are not.
The log data dates and times are set based on contents set here.

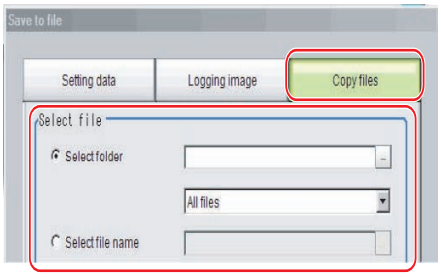
1 **In the Main Window, select [System settings] – [Other] – [Date-time setting] from the [Tool] menu.**

The [Date-time setting] dialog box is displayed.

2 **Set the current date and time.**

3 **Click the [Apply] button.**

Close the [Date-time setting] dialog box.



Selecting the Language: [Language Setting]

You can set the language to use for the characters displayed on the displays. Information about the application software will be displayed in the selected language.

Note

- When a Controller with default settings is started, the [Language setting] dialog box is automatically displayed.
- The Controller default setting is for Japanese-language displays. If the language setting is changed to a non-Japanese language, the system automatically restarts.

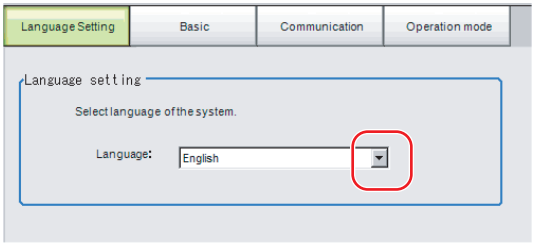
1 **In the Main Window, select [System settings] – [Startup] – [Startup setting] – [Language Setting] from the [Tool] menu.**

The [Language setting] dialog box is displayed.

2 **Click the [▼] button to select the language.**

3 **Click the [Apply] button.**

4 **Return to the Main Window and click the [Data save] button. Then, restart the Controller. The Controller will start up with the selected language.**



Note

The [Language setting] dialog box is also displayed when restarting after executing [System initialization].

Setting the Fan Rotation Speed: [Fan Control Setting]

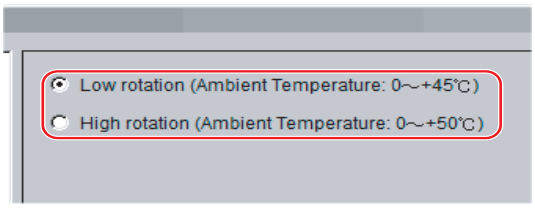
This procedure describes how to set the rotation speed of the Controller fan.
FZ5-6☐☐ or FZ5-11☐☐ only

Note

The default setting is for low rotation. Use fast rotation when using the system in a high-temperature environment between 45 and 50°C.

1 **In the Main Window, select [System settings] – [Other] – [Fan control setting] from the [Tool] menu.**

The [Fan control setting] dialog box is displayed.



2 **Select a fan setting.**

Set value [Factory default]	Description
• [Low rotation (Ambient temperature: 0 to +45°C)] • High rotation (Ambient temperature: 0 to +50°C)	Select the rotation speed of the fan.

3 **Click the [Apply] button.**

IMPORTANT

The fan control setting is disabled for the FZ5-L35☐.

Setting the Status at Startup: [Startup Settings]

You can set the status when the power supply is turned ON.

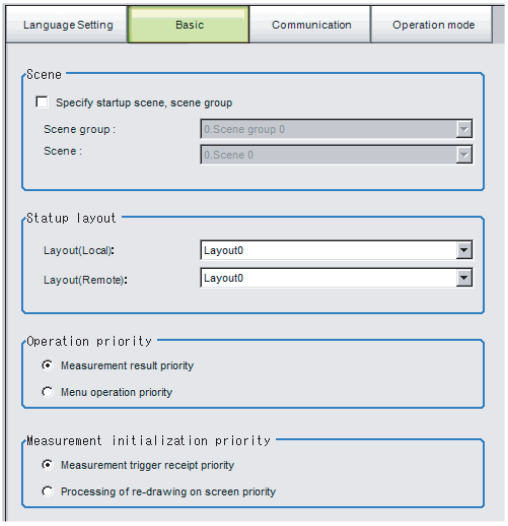
Inspection can be set to start immediately after the power supply is turned ON by setting the number of the scene in which the measurement contents are set.

In addition, settings can be done so that the [Camera setting] dialog box is not displayed during startup.

1 In the Main Window, select [System settings] – [Startup] – [Startup setting] from the [Tool] menu.

The [Startup setting] dialog box is displayed.

2 Change the settings.



Basic

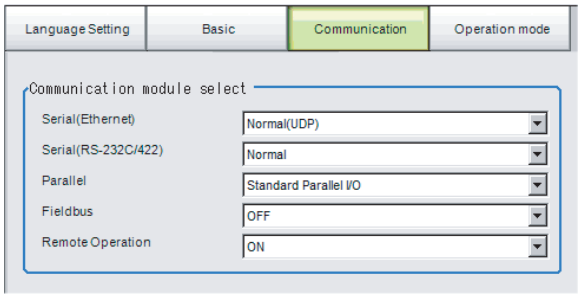
Item		Set value [Factory default]	Description
Scene			
	Specify startup scene, scene group	[Selected]	The selected scene and scene group will be the scene and scene group at startup.
		Not selected	The scene or scene group that is set in the Controller when you execute [Data save] will be used at startup.
	Scene group	Scene groups 0 to 31 [Scene group 0]	Select the scene group to use at startup.
	Scene	Scene 0 to 127 [Scene 0]	Select the scene to use at startup.
Startup layout		Layout (local): Layout 0 Layout (remote): Layout 0	Select the layout number to display at startup. Layout (local): Layouts 0 to 8 Layout (remote): Layouts 0 to 8
Operation priority			Specify whether measurement results display or menu operation is to have priority.
		[Measurement result priority]	Processing of measurement results display is given priority. Menu operation will be harder to receive because of its lowered priority status.
		Menu operation priority	Menu operation is given priority. Measurement results displays may be incomplete.

Item	Set value [Factory default]	Description
Measurement initialization priority		Measurement initialization is performed immediately after changing the scene or editing the flow. You can specify whether measurement trigger reception or display refresh processing has priority during measurement initialization.
	[Measurement trigger receipt priority]	Measurement trigger reception is given priority even during measurement initialization processing. Displays for display refresh processing may be incomplete because of its lowered priority status.
	Processing of re-drawing on screen priority	Processing of refreshing the display is given priority. Measurement triggers will not be received until display refresh processing is completed.

Communication

Select the type of Communications Module to determine the communications method to use for communications between the Sensor Controller and external device.

Refer to the ▶ *Vision System FH/FZ5 Series User's Manual for Communications Settings* (Cat No. Z342) for details on how to select the Communications Module.



Item	Set value [Factory default]	Description
Serial (Ethernet)	• [Normal (UDP)] • Normal (TCP) • Normal (TCP Client) • Normal (UDP) (Fxxx series method) • PLC Link (SYSMAC CS/CJ/CP/One) • PLC Link (MELSEC QnU/Q/QnAS) • PLC Link (JEPMC MP)	Specify the Communications Module. Serial/Ethernet Normal/Normal (Fxxx series method): Communications are performed with external devices through [non-procedure] communications. For differences between [Normal] and [Normal (Fxxx series method)], refer to the following note. PLC Link: Communications are performed through link areas with the PLC. Parallel Standard Parallel I/O: Communications are performed with a standard parallel interface. Fieldbus: Communications are performed through EtherCAT communications and the EtherNet/IP interface. EtherCAT is supported only by the FH-series Controllers. Remote operation: The Controller is operated from an external device.
Serial (RS-232C/RS-422)	• [Normal] • Normal (Fxxx series method) • PLC Link (SYSMAC CS/CJ/CP/One) • PLC Link (MELSEC QnU/Q/QnAS)	
Parallel	[Standard Parallel I/O]	
Fieldbus	• [OFF] • EtherCAT • EtherNet/IP	
Remote Operation	• [ON] • OFF	

IMPORTANT

- Do not set [EtherNet/IP] and [PLC Link] at the same time. They cannot be used at the same time.
  Example:
  Set [Serial (Ethernet)] to [PLC Link] and [Fieldbus] to [EtherNet/IP] at the same time.
  Set [Serial (RS-232C/422)] to [PLC Link] and [Fieldbus] to [EtherNet/IP] at the same time.
  Set both [Serial (Ethernet)] and [Serial (RS-232C/422)] to [PLC Link], and set [Fieldbus] to [EtherNet/IP] at the same time.
- Do not set both [Serial (Ethernet)] and [Serial (RS-232C/422)] to [PLC Link]. They cannot be used at the same time.

Note

- Normal (Fxxx Series Method)
  With the [Normal (Fxxx series method)] communications method, the OK response timing in relation to MEASURE commands is different from that of the [Normal] communications method.

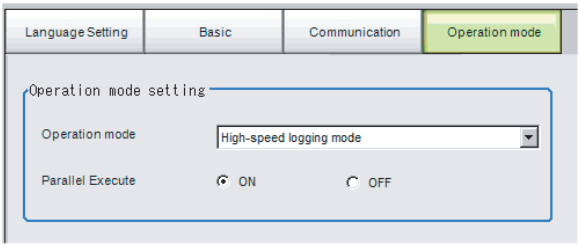
Normal (Fxxx series method) communications method	Normal communications method
MEASURE Measurement result OK	MEASURE OK Measurement result

- OK Responses
  With [Normal communications method], an OK response is returned when the Controller receives a command.
  With [Normal (Fxxx series method)], an OK response is returned when command execution is completed.

Operation Mode (FH and FZ5-11□□ Only)

Select the operation mode according to the application conditions, such as the operation mode that is best suited to reduce the takt time and downtime.

Refer to ▶ Setting the Operation Mode (p.268) for details on the operation modes.



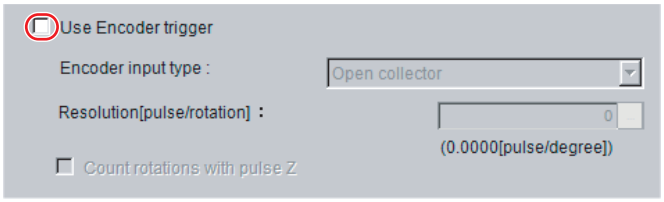
Item	Set value [Factory default]	Description
Operation mode	• Single-line High-speed mode • [High-speed logging mode] • Multi-line random-trigger mode • Non-stop adjustment mode	Set the operation mode. Reference: ▶ Setting the Operation Mode (p.268)
Parallel execution	ON/[OFF]	Set whether to uses parallel processing. Reference: ▶ Parallel Processing Settings (Automatic Parallelization) (p.282)

3 Click the [Apply] button.

Setting the Encoder Trigger: [Encoder Trigger Setting]

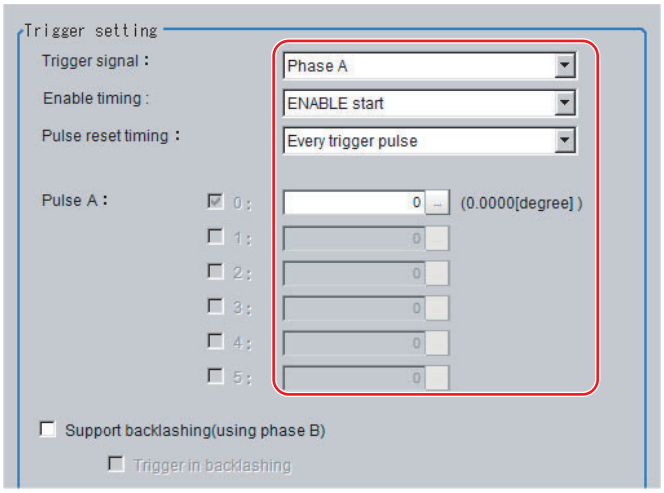
1 **In the Main Window, select [System setting] – [Other] – [Encoder trigger setting] from the [Tool] menu.**

2 **Set the target encoder.**



Parameter	Set value [Factory default]	Description
Use encoder trigger	• Selected • [Not selected]	Set whether to use an encoder trigger.
Encoder input type	• [Open collector] • Line driver	Select the output format of the encoder to be connected.
Resolution [pulse/ rotation]	[1] to 65535	Set how many pulses equal one rotation. Set this parameter according to the resolution of the encoder.
Count rotations with pulse Z	• Selected • [Not selected]	When this check box is selected, pulse Z is used to determine whether a full rotation is made. When this check box is cleared, a full rotation is determined by whether the number of pulses on phase A has reached the resolution of phase A.

3 **Set the advanced settings for the trigger as necessary.**



Parameter	Set value [Factory default]	Description
Trigger signal	[Phase A] Phase Z	Set the phase to use as the trigger signal.

Parameter	Set value [Factory default]	Description
Phase A		
Enable timing	• [ENABLE start] • STEP start	Set the timing for starting the pulse count. ENABLE start: Counts the pulses input during the measurement trigger receipt period. STEP start: The pulses are not counted even during the measurement trigger receipt period until the STEP signal is input.
Pulse reset timing	• [Every trigger pulse] • Every rotation (Pulse Z) • Every STEP	Set the timing for resetting. When [ENABLE start] is specified for [Enable timing], you can select [Every trigger pulse] or [Every rotation (Pulse Z)]. When [STEP start] is specified for [Enable timing], you can select [Every trigger pulse] or [Every STEP]. When [Every rotation] is set, multiple pulses can be set for phase A.
Phase A trigger	[0] to 65535	Set how many pulses it takes to produce a trigger.
Support backlashing (using phase B)	• Selected • [Not selected]	Set whether the rotation direction is detected.
Trigger in backlashing	• Selected • [Not selected]	Set to produce a trigger during reverse rotation.
Phase Z		
Enable timing	• [ENABLE start] • STEP start	Set the timing for starting the pulse count. ENABLE start: Counts the pulses input during the measurement trigger receipt period. STEP start: The pulses are not counted even during the measurement trigger receipt period until the STEP signal is input.
Pulse reset timing	• [Every trigger pulse]	Set the timing for resetting the pulse count of phase Z. Every trigger pulse: Reset the pulse count of phase Z each time the Sensor Controller outputs a trigger.
Phase Z trigger	[1] to 1023	Set how many pulses it takes to produce a trigger.

IMPORTANT

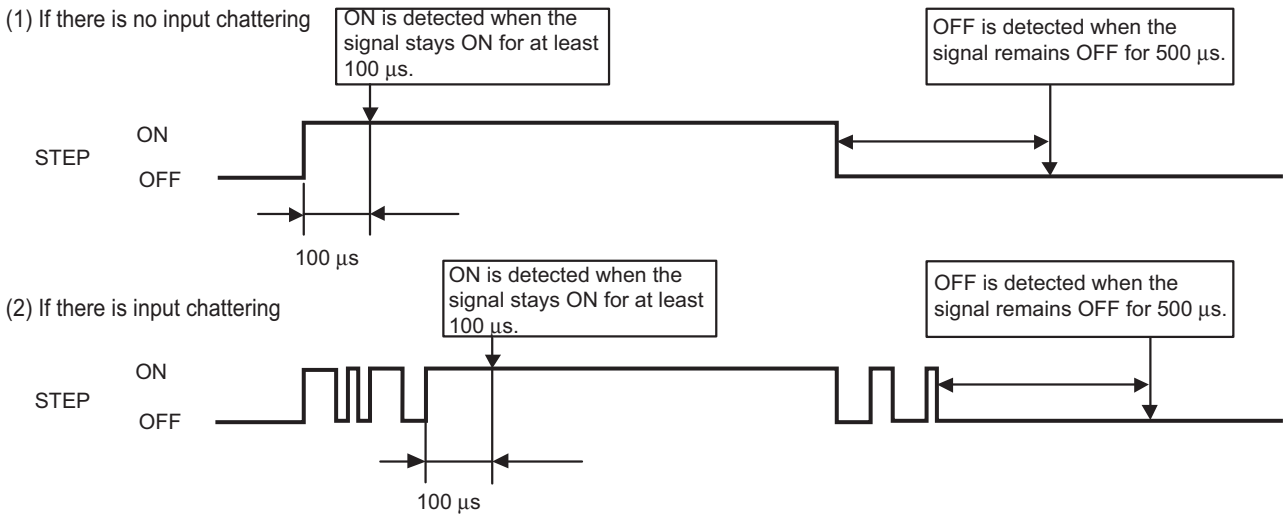
The encoder trigger setting is disabled for the FZ5-L35□.

Setting the STEP Input Detection Pulse Width: [STEP Setting]

You can set a filter as a countermeasure against STEP input chattering and to prevent operation malfunctions due to noise.

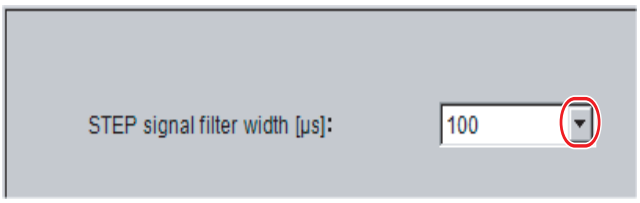
Filter Set Value of 100 μs (Default Value)

The STEP signal is detected as being ON at the point it is ON continuously for at least 100 μs, and measurement begins at that point. Accordingly, STEP signal detection is delayed by an amount of time equivalent to the set filter value. Also, when turning from ON to OFF, the OFF filter fixed at 500 μs is activated and the STEP signal is detected as OFF when it is OFF for at least 500 μs.



1 **In the Main Window, select [System settings] – [Other] – [Controller] – [STEP setting] from the [Tool] menu.**

2 **Set the filter width in the [STEP setting] area.**



Parameter	Set value [Factory default]	Description
STEP signal filter width [µs]	• [100] • 200 • 300 • 400 • 500	Set the filtering width.

IMPORTANT

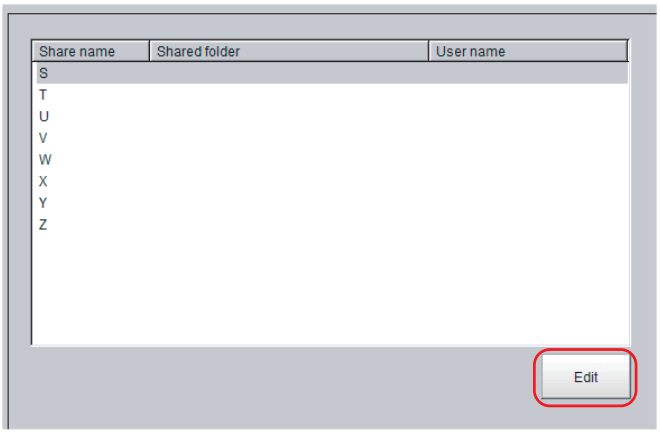
If you use Multi-line Random-trigger Mode for an FH-series Controller, the value for the STEP signal filter on line 0 is applied to all lines.

Setting a Network Drive: [Network Drive Setting]

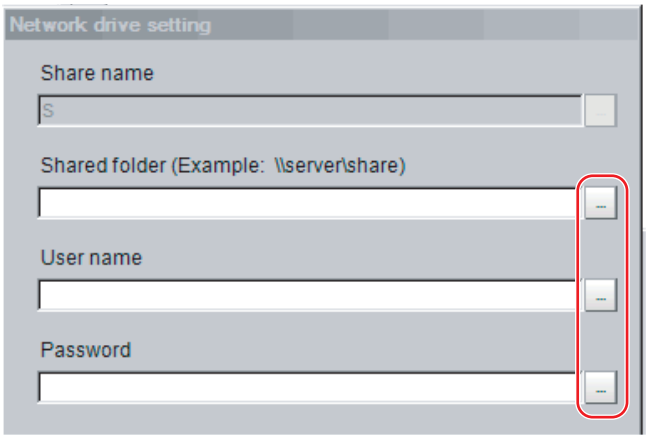
You can save logging images to an external device, such as a network-connected computer with a shared folder, using a network drive.
You can also load setting data saved in a network drive into the Controller.
If you register a shared folder on the network drive, the network drive connected to the [Select file] or [Select folder] dialog box of the FH/FZ5 software will be displayed.

1 **In the Main Window, select [System settings] – [Other] – [Network drive settings] from the [Tool] menu.**

2 **Select the share name to register, and click the [Edit] button.**



3 Enter the information to be allocated to the network drive (shared name), such as the shared folder, user name, and password.



Parameter	Set value [Factory default]	Description
Share name	S,T,U,V,W,X,Y,Z	This is the name to be recognized by the Controller as a network drive. Only 1 unit can be connected. For FZ5-L3□□ or FZ5-6□□, the shared name is displayed in the \Network folder. For the FH or FZ5-11□□, a drive letter such as E:, F:, G:, or H: is displayed.
Shared folder	---	Specify the shared folder name. Create a folder in advance on a computer that has a DNS resolved name on the network, and specify this as the shared folder name. For example, if the host name (computer name) is VISION and the shared name is COMMON, specify a name such as \\VISION\COMMON.
Username	---	Enter the user name and the password to access the network drive. If you do not know the user name or the password, contact the device network administrator.
Password	---	

4 Click the [Apply] button.

The settings are enabled after the Controller is restarted.

Note

With the FH/FZ5-11□□, you can still connect to the network drive even if you specify the IP address instead of the host name when you set the shared folder name. With the FZ5-L35□/FZ5-6□□, you will not be able to connect to the network drive if you specify the IP address.

IMPORTANT

- If images and data are logged to a network drive, the use of the multi-input function and other heavy measurement loads on the Controller may slow down communications and cause logging errors. In this case, set the measurement takt time so that there is some leeway.
- If many drives are set as network drives, it may require a greater amount of time to start the Controller.
- Do not pull out the LAN cable while the Controller is accessing the network drives.
- Start the Controller when the network drives are ready. Connections cannot be established if the network drives are not ready when the Controller is started.
- If there is no access to a network drive for a certain amount of time, the connection will be automatically cut off depending on the network drive setting at the connection target. Make sure that the setting at the connection target is not set to automatic disconnection.
- Dates and Times of the Updates of the Files Created at Network Logging
  If the time zone of the external device is different from the time zone setting of the Controller [GMT-08:00 Pacific Time (US & Canada): Do not automatically adjust clock for Daylight Saving Time], the date and time actually written may be different from the date and time of the file update. Adjust the time zone of the external device to match that of the Controller.

Checking System Information: [System Information]

The Controller model and measurement application version can be checked.
You can check the external memory status only on the FH and the FZ5-11□□.
Reference: ▶ Using External Memory Devices (FH and FZ5-11□□ Only) (p.256)

1 **In the Main Window, select [System information] from the [Function] menu.**

The [System information] dialog box is displayed.

2 **Check the information.**

The Controller model and measurement application version can be checked.

3 **Click the [Memory state] button.**

You can check the following information.

Available Application Memory

The application memory is the memory that is used by all applications.
Knowing the available memory provides a rough standard for confirming status while operating.

Available data memory (FZ5-L3□□ or FZ5-6□□ only)

The data memory is the amount of memory that can be used for scene group data. Check the available memory that can be used for unit data and settings data for each unit.

4 **Click the [Close] button.**

The [System information] dialog box closes.

Note

Application Memory

This memory is used for all applications.
This memory area is used temporarily by the application. By confirming available memory, this provides a rough standard for confirming status while operating. The user cannot access it.
With the FH or FZ5-11□□, a warning message is displayed when the amount of available application memory falls below 1 GB. Adjust the inspection flow so that the available memory is more than 1 GB.

Data Memory (FZ5-L3□□ or FZ5-6□□)

This memory is used to hold the current setting details of scene group data.
If the capacity of this memory is exceeded, adding units and copying scenes cannot be performed to edit the flow.
The available data memory can be checked from the system menu.
Reference: ▶ Checking System Information: [System Information] (p.300)

Appendices

Alignment

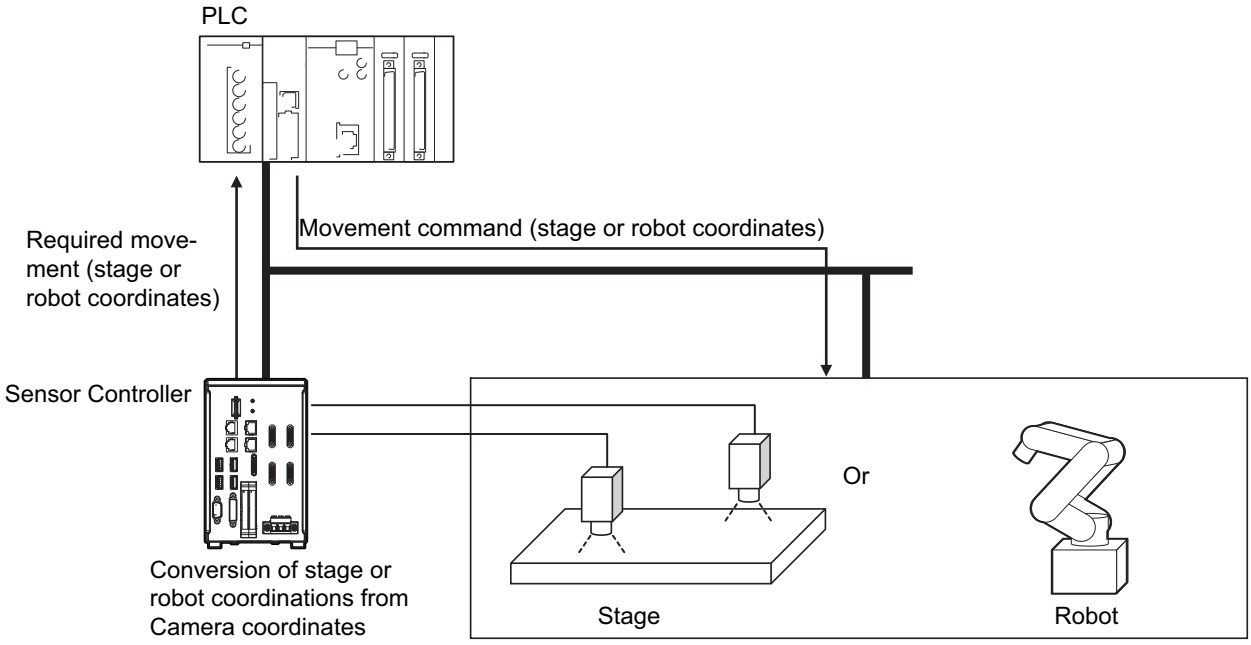
An FH/FZ5-series Controller has an alignment function that you can use to position workpieces for stage and robot applications (hereafter called *external devices*).
Alignment executes a combination of more than one processing item.
This section gives an overview of alignment, and provides details on its use in general.
Refer to the ▶ *Vision System FH/FZ5 Series Processing Item Function Reference Manual* (Cat No. Z341) for details on the processing items that are used for alignment.

Overview

An FH/FZ5-series Controller measures the position of alignment marks on the workpiece, or the position of specific features of the workpiece (such as corners), and outputs the axis movement to a PLC or stage controller, which aligns the workpiece into a specified position (hereafter called the *reference position*).
The PLC or stage controller simply needs to send a movement command to the external device for the axis movement amount that was output from the FH/FZ5-series Controller to align the position of the workpiece.

The following two measurement flows are used to perform alignment. Set both flows for each scene.

- Calibration flow
  This flow associates the coordinates of the external device with the Camera coordinates.
- Alignment flow
  This flow uses the calibration parameters that were created in the calibration flow to align the position of the workpiece.



Supported External Devices

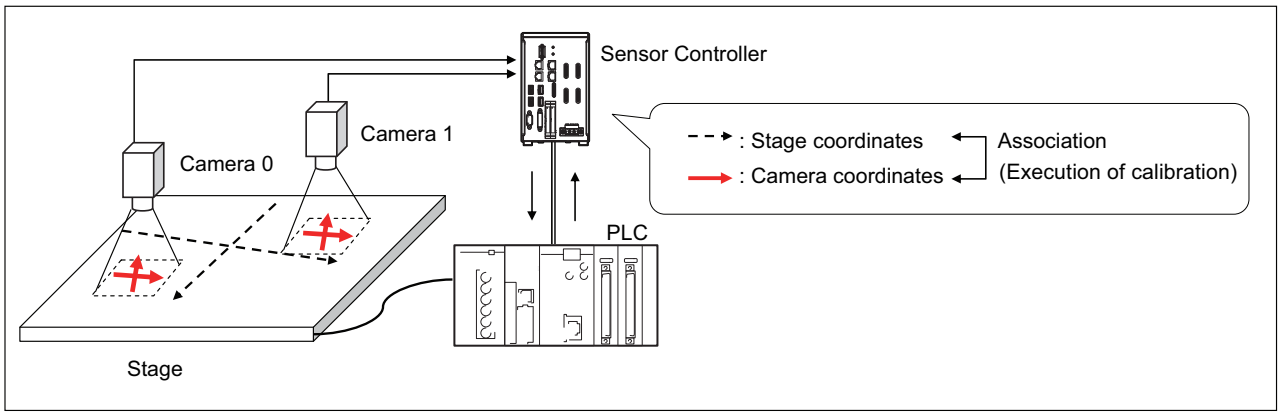
Alignment is supported for the following external devices.

Type			
Stages	XY		
	XYθ	θ axis: Direct drive	Camera axis movement: None
			Camera axis movement: X axis
			Camera axis movement: Y axis
			Camera axis movement: XY axes
		θ axis: Linear drive	Camera axis movement: None
			Camera axis movement: X axis
			Camera axis movement: Y axis
			Camera axis movement: XY axes
	θXY	θ axis: Direct drive	Camera axis movement: None
			Camera axis movement: X axis
			Camera axis movement: Y axis
			Camera axis movement: XY axes
	XYθ	θ axis: Linear drive	Camera axis movement: None
			Camera axis movement: X axis
			Camera axis movement: Y axis
			Camera axis movement: XY axes
	UVW	Direct fulcrum motion	
		Rotary fulcrum motion	
	UVWR	Direct fulcrum motion	
		Rotary fulcrum motion	
Robots	3 axes		
	4 axes	Control method: Fixed positions	
		Control method: Measured positions	

Calibration

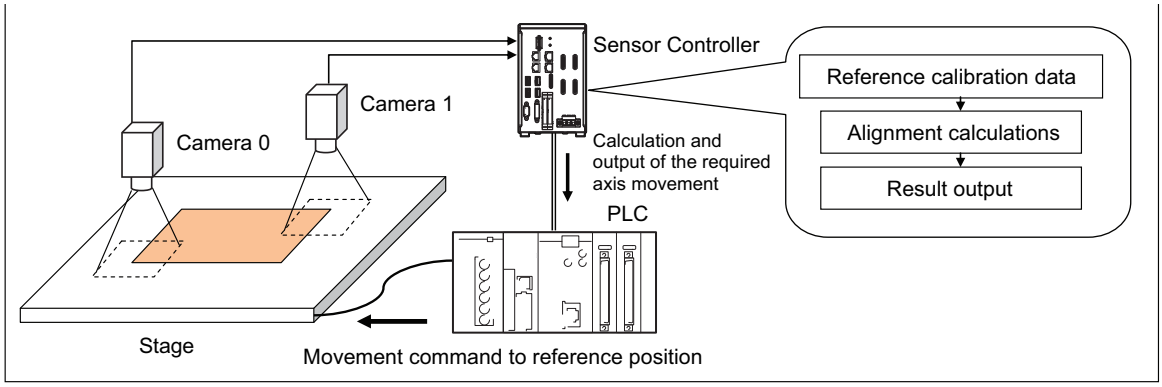
The Camera and external device both have their own coordinates, so it is necessary to calculate the relationship of the coordinates of the Camera and the external device in advance. This process is called *calibration*.

The FH/FZ5-series Controller uses the external device to move the workpiece and measure it. This is repeated to calculate the calibration parameter.



Alignment: Aligning the Position of the Workpiece

The image taken by the Camera is used to measure the position of alignment marks on the workpiece, or the position of specific features of the workpiece (such as corners) in Camera coordinates (pixels). The calibration parameter that is calculated in the calibration flow is used to convert the measured position into the coordinates of the external device. Finally, the axis movement that is required to match the measured position to the reference position angle is calculated and output to the external device.



Execution of Calibration

This section uses two Cameras and a stage as the external device to describe an example for setting the calibration flow and executing calibration.

Example: Calibration Flow Example for 2 Cameras

Camera Image Input processing item	· · · Select Camera 0.
↓	
Machine setting processing item	· · · Register [stage data] or [robot data] depending on the external device to be used, and then set the information for the external device. The machine setting processing item can be placed anywhere in the measurement flow other than at the beginning of the measurement flow.
↓	
Position measurement processing item	· · · Measure the position of the alignment marks to use as the sampling target, or the position of a special feature (such as corners) of the workpiece from the image in Camera 0. To output the position on the image, set [Calibration] on the [Output parameter] tab page to OFF.
↓	
Camera Switching processing item	· · · Select Camera 1.
↓	
Position measurement processing item	· · · Measure the position of the alignment marks to use as the sampling target, or the position of a special feature (such as corners) of the workpiece from the image in Camera 1. To output the position on the image, set [Calibration] on the [Output parameter] tab page to OFF.
↓	
Calibration processing items	· · · Register [Vision Master Calibration] or [PLC Master Calibration]. Set the machine setting processing item to reference, and then set the sampling.
↓	
Data output processing items	· · · Depending on the communications method, register either [Data output] or [Fieldbus data output]. The Vision Master Calibration processing item requires that you send three parameters to the PLC: [Axis movement required for sampling], [Origin return], and [Completion Timing of Calibration]. Each parameter is output after a reference of the calculation for the Vision Master Calibration processing item is made. With the PLC Master Calibration processing item, it is not necessary to send data to the PLC, so Data Output processing items are not necessary.

Settings for the External Device to Use

Set the parameters that describe the type and specifications of the external device to use.

Add the following machine setting processing item to the flow.

● Stage Data

If you are using a stage, use this processing item.
Refer to ▶ *Stage Data* of the *Vision System FH/FZ5 Series Processing Item Function Reference Manual* (Cat No. Z341).

● Robot Data

If you are using a robot, use this processing item.
Refer to ▶ *Robot Data* of the *Vision System FH/FZ5 Series Processing Item Function Reference Manual* (Cat No. Z341).

The settings for the machine setting processing item can be referenced by the following processing items.

- Vision Master Calibration
- PLC Master Calibration
- Convert Position Data
- Movement Single Position
- Movement Multi Points

Note

The settings for the machine setting processing item can be referenced from other scenes.
The machine setting processing item can be registered only in one scene group, but it can be referenced from other the scenes. This enables central management of the external device settings.

Settings for the Calibration To Execute

Add the following Calibration processing items to the flow.

● Vision Master Calibration

This processing item calculates the axis movement that is necessary to perform the most accurate calibration within the range of movement of the external device.
This routine is performed repeatedly to create very precise calibration parameters: Output the calculated axis movement to the PLC, the PLC moves the external device, the PLC sends a measurement command to the FH/FZ5-series Controller.
Reference the Reference Calib Data processing item to access the calibration parameters that were created.
Refer to ▶ *Vision Master Calibration* of the *Vision System FH/FZ5 Series Processing Item Function Reference Manual* (Cat No. Z341).

IMPORTANT

If you are using the measurement results of the Sensor Controller to operate an external device, make sure the movement is within the range of the movement of the external device before you implement the move.

Note

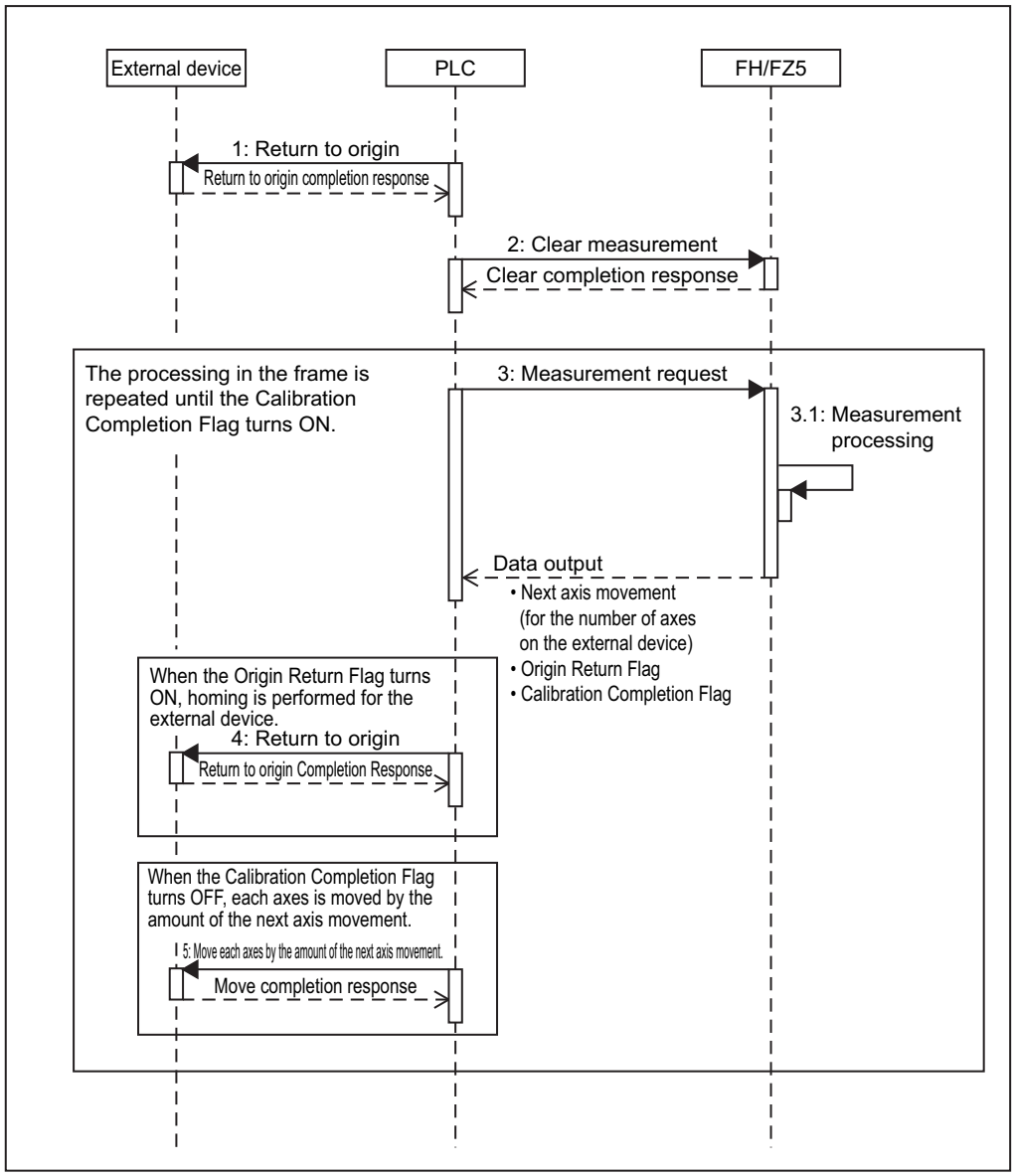
The Vision Master Calibration processing item has the following advantages over PLC Master Calibration processing item.

- There is no need for trial and error with the order of movements or axis movement, even with complicated external devices, such as UVW stages.
  The order of movement of the workpiece during calibration and the axis movement of the external device that corresponds to each movement are automatically calculated.
- Even if you change the calibration conditions, you do not have to change the PLC ladder programming.
  If you add more sampling points for better precision, or if you reduce the number of sampling points to shorten the calibration time, you can still use the ladder programming as is.
- In conjunction with calibration, you can also calculate the parameters for distortion correction at the same time.
  This allows you to align positions with even greater accuracy without doing any extra work.

● PLC Master Calibration

The PLC Master Calibration processing item creates calibration data as follows: The axis movement that is necessary for sampling is calculated in advance. Then, the following routine is repeated: the PLC moves the external device, it sets the axis movement in the FH/FZ5-series Controller, and then it sends a measurement command to the FH/FZ5-series Controller.
Reference the *Reference Calib Data* item to access the calibration data that was created.
Refer to ▶ *PLC Master Calibration* of the *Vision System FH/FZ5 Series Processing Item Function Reference Manual* (Cat No. Z341).

● Vision Master Calibration

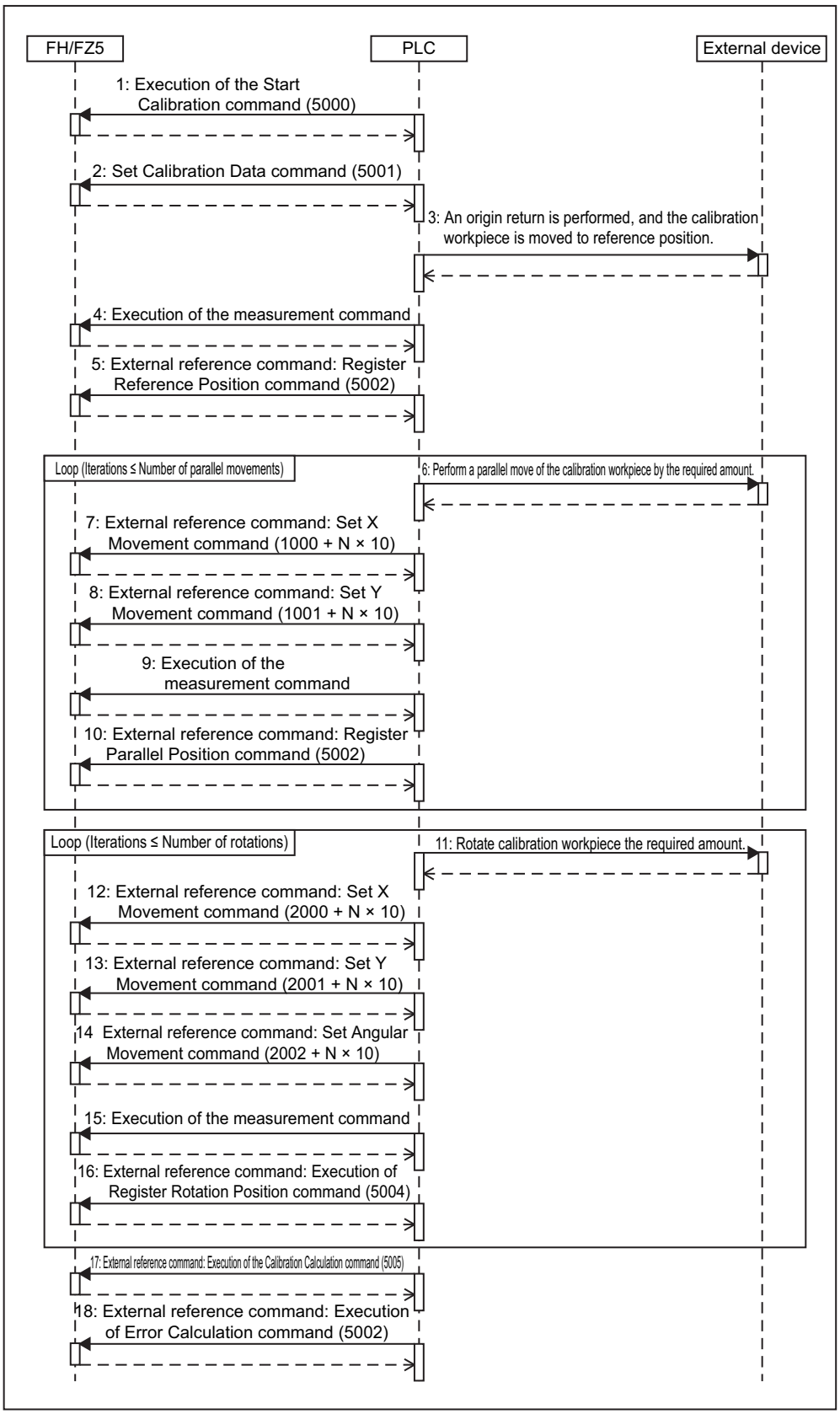


● PLC Master Calibration

• Standard Mode

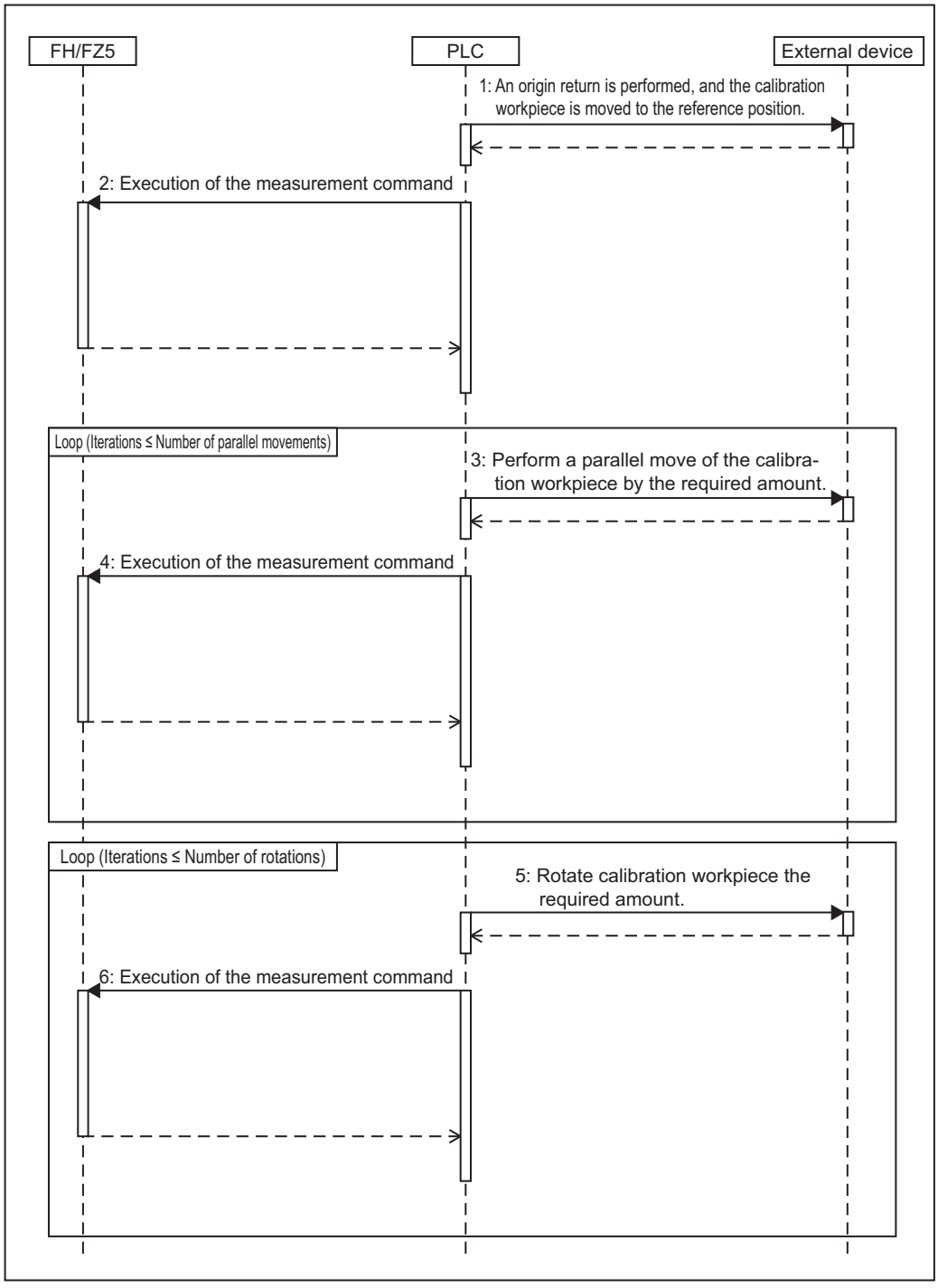
The numbers in parenthesis are the external reference numbers that are used when commands are executed for the PLC Master Calibration processing unit.

Refer to ▶ *External Reference Table* under *PLC Master Calibration* in the *Vision System FH/FZ5 Series Processing Item Function Reference Manual* (Cat No. Z341).



- **Easy Mode**

  The Calibration Execution Flag and movement in actual coordinates must be set in advance in the PLC Master Calibration settings dialog box.



Troubleshooting

PLC Master Calibration

Refer to ▶ *Key Points for Adjustment* under *PLC Master Calibration* in the *Vision System FH/FZ5 Series Processing Item Function Reference Manual* (Cat No. Z341).

Vision Master Calibration

Refer to ▶ *Key Points for Adjustment* under *Vision Master Calibration* in the *Vision System FH/FZ5 Series Processing Item Function Reference Manual* (Cat No. Z341).

Calibration Support Tool

Symptom	Type	Parameters to adjust	Correction
The unit number is set to [None] and cannot be selected.		---	Check whether there are any processing items that retain calibration data that are registered in the current scene.
The setting data is initialized every time the Toolbox is opened.		• Tool Settings Save Button • Tool Settings Load Button	This is the specified behavior. If you need to save the settings data, use the [Save] button in the Toolbox. The saved settings can be loaded with the [Load] button.
How do you rotate the graphic display for better visibility?		• Standard axis • Axis direction • Direction of rotation	You can rotate the graphic display along the standard axis, axis direction, and rotation direction on the [Display setting] tab page.

Alignment

This section will use two Cameras to describe the procedures for setting the alignment flow and executing alignment.

Example 1: Alignment Flow Example (Position Data Calculation and Axis Movement Conversion) for Two Cameras

Camera Image Input processing item	Select Camera 0.
↓	
Reference Calib Data processing item	Select the Vision Master Calibration or PLC Master Calibration processing item on the calibration scene with the reference calibration data.
↓	
Position measurement processing item	Measure the position of the alignment marks or the position of a special feature of the workpiece from the image in Camera 0. To output the actual coordinates of the position, set [Calibration] on the [Output parameter] tab page to OFF. Out of the general measurement processing items (Inspections and Measurements), you can use processing items with measurement results that have an X and Y in the calculation strings. Set the workpiece in reference position and then set the reference position.
↓	
Camera Switching item	Select Camera 1.
↓	
Reference Calib Data processing item	Select Image Master Calibration or PLC Master Calibration on the calibration scene with the reference calibration data.
↓	
Position measurement processing item	Measure the position of the alignment marks or the position of a special feature of the workpiece from the image in Camera 1. To output the actual coordinates of the position, set [Calibration] on the [Output parameter] tab page to OFF. Out of the general measurement items (Inspections and Measurements), you can use items with measurement results that an X and Y in the calculation strings. Set the workpiece in reference position and then set the reference position.
↓	
Convert Position Data processing item	Use the two measurement positions to calculate the reference position and reference angle of the workpiece. Set [Reference XY] in the expression of the two-position measurement processing item as position XY for point 0 and point 1.
↓	
Position Data Calculation processing item	Calculate the measurement position and measurement angle of the workpiece with the two-point measurement position. Set [Coordinates XY] in the expression of the two-position measurement processing item as position XY for point 0 and point 1.
↓	

Convert Axis Movement processing item	· · ·	Set the machine setting processing item to reference. Then set the expressions for the two Position Data Calculation processing items, [Calculate position XY] and [Calculate angle TH], in the [Reference position XY] and [Reference angle], and the [Measurement position XY] and [Measurement angle] in the Movement Single Position processing item. Before you execute the measurement, set the [Current setting] with the position of the axis of the external device when the measurement was performed.
↓		
Data output processing item	· · ·	Depending on the communications method, register either [Data output] or [Fieldbus data output]. Send each [Axis movement] expression in the Movement Single Position processing item to the PLC.

Example 2: Alignment Flow Example (Movement Multi Points) for Two Cameras

Camera Image Input item	· · ·	Select Camera 0.
↓		
Reference Calib Data processing item	· · ·	Select the Vision Master Calibration or PLC Master Calibration processing item on the calibration scene with the reference calibration data.
↓		
Position measurement processing item	· · ·	Measure the position of the alignment marks or the position of a special feature of the workpiece from the image in Camera 0. To output the actual coordinates of the position, set [Calibration] on the [Output parameter] tab page to OFF. Set the workpiece in reference position and then set the reference position.
↓		
Camera Switching processing item	· · ·	Select Camera 1.
↓		
Reference Calib Data processing item	· · ·	Select the Vision Master Calibration or PLC Master Calibration processing item on the calibration scene with the reference calibration data.
↓		
Position measurement processing item	· · ·	Measure the position of the alignment marks or the position of a special feature of the workpiece from the image in Camera 1. To output the actual coordinates of the position, set [Calibration] on the [Output parameter] tab page to OFF. Set the workpiece in reference position and then set the reference position.
↓		
Movement Multi Points processing item	· · ·	Set the machine setting processing item to reference. Then set the expressions for the position measurement processing items, [Reference XY] and [Measure XY] in the [Reference position XY] and [Measurement position XY] in the Movement Multi Points processing item. Before you execute the measurement, set the [Current setting] with the position of the axis of the external device when the measurement was performed.
↓		
Data output processing item	· · ·	Depending on the communications method, register either [Serial data output] or [Fieldbus data output]. Send each [Axis movement] expression in the Movement Multi Points processing item to the PLC.

● Convert Position Data + Movement Single Position

Use these settings to align the measured position and measured angle with the reference position and reference angle.
The movement that is required to align each axis of the external device is calculated and output.
Refer to ▶ *Convert Position Data* and *Movement Single Position* in the *Vision System FH/FZ5 Series Processing Item Function Reference Manual* (Cat No. Z341).

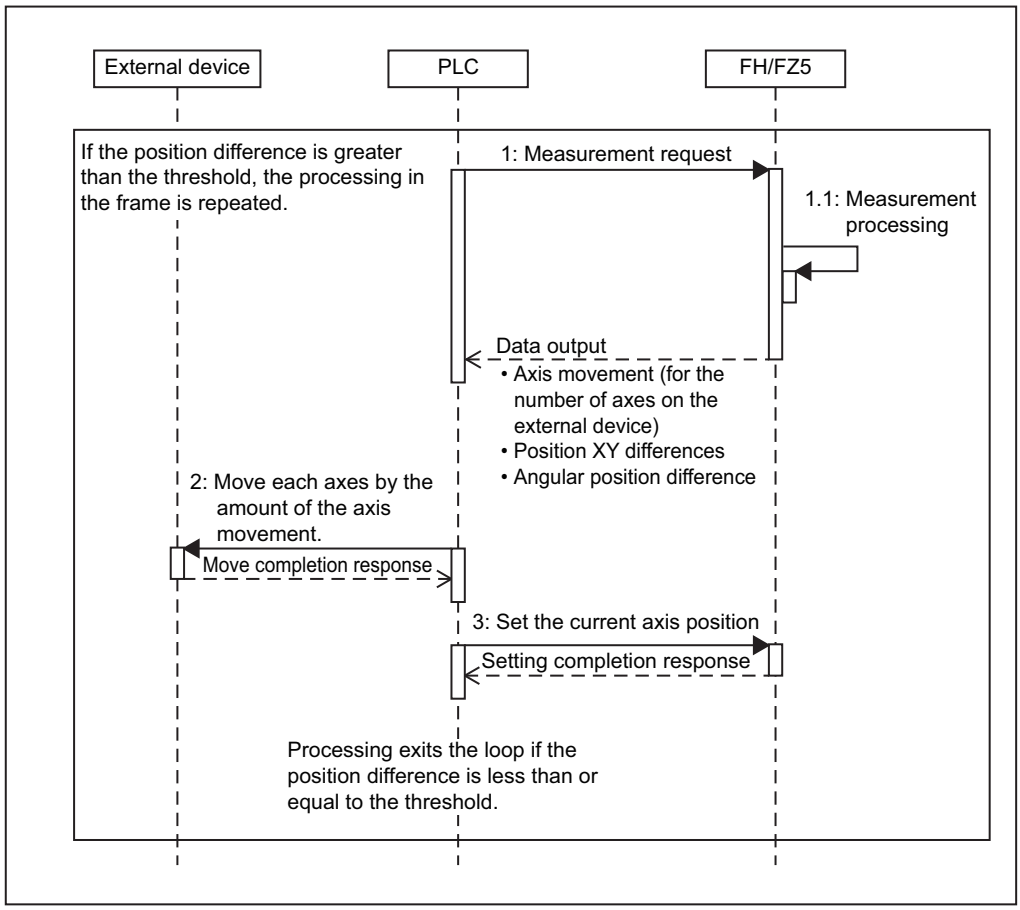
● Movement Multi Points

Use this setting to align more than one measured position with corresponding reference positions.
The movement that is required to align each axis of the external device is calculated and output.
Refer to ▶ *Movement Multi Points* in the *Vision System FH/FZ5 Series Processing Item Function Reference Manual* (Cat No. Z341).

IMPORTANT

If you are using the measurement results of the Sensor Controller to operate an external device, make sure the movement is within the range of the movement of the external device before you implement the move.

Executing Alignment



● Checking the Calibration Results

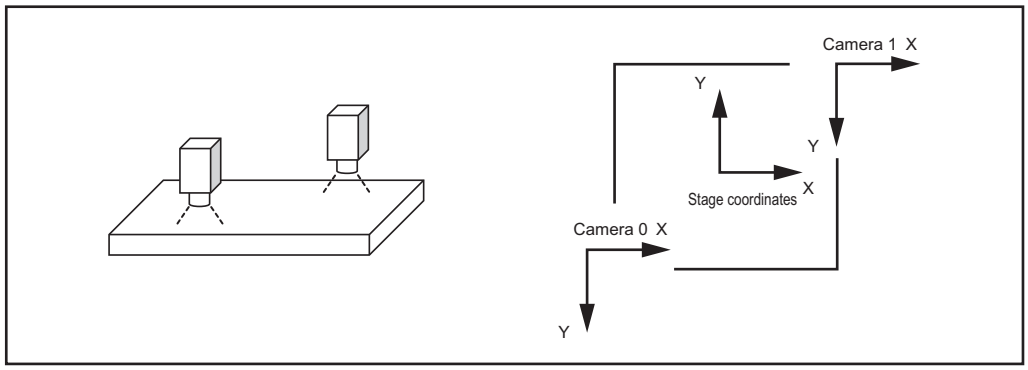
The Calibration Support Tool uses the calibration parameters to graphically display the relationship of the positions that are represented by the Camera coordinates and the actual coordinates. You can compare the relationship of the displayed position with the actual position of the equipment. This allows you to easily check the validity of the calibration parameters. To start the Calibration Support Tool, select [Calibration support tool] from the [Tool] menu.

• Overview

The calibration data for the specified data number that is stored in the processing unit for the specified unit number is used to graphically display the positioning relationship of the Camera coordinates and the actual coordinates.

You can specify the following processing items for the processing units

- Camera Image Input
- Camera Image Input HDR
- Camera Image Input HDR Lite
- Camera Image Input FH
- Vision Master Calibration
- PLC Master Calibration
- Camera Calibration
- Reference Calib Data
- Precise Calibration

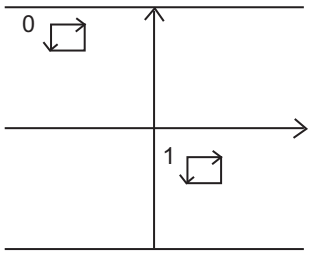


Generate calibration parameters

A comparison of the device drawings and the graphic display will easily show calibration errors.

Camera 0	Camera 1
A = 0.2500	A = 0.2500
B = 0.0000	B = 0.0000
C = −480.0000	C = 160.0000
D = 0.0000	D = 0.0000
E = −0.2500	E = −0.2500
F = 480.0000	F= −160.0000

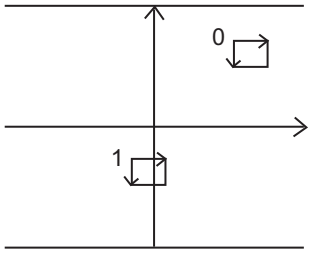
Graphic display



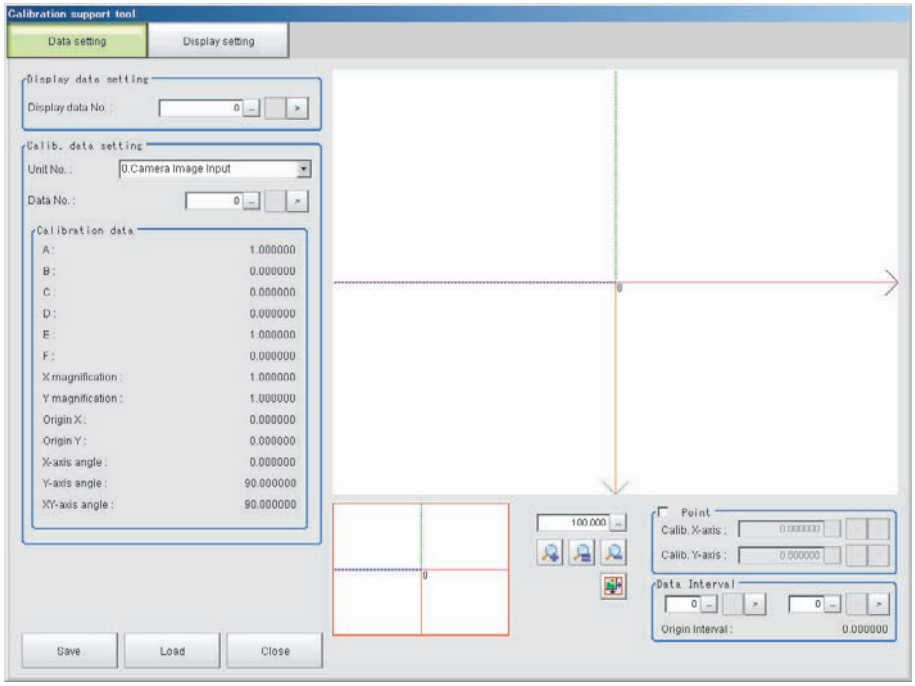
Calibration succeeded.

Camera 0	Camera 1
A = 0.2500	A = 0.2500
B = 0.0000	B = 0.0000
C = 480.0000	C = −120.0000
D = 0.0000	D = 0.0000
E = −0.2500	E = −0.2500
F = 480.0000	F = −120.0000

Graphic display



Calibration failed.



Refer to the *Vision System FH/FZ5 Series Processing Function Reference Manual* (Cat No. Z341) for further information on the alignment processing items.

● Machine Simulator

The Machine Simulator processing item allows you to see how the alignment marks move on the image while each axes of the stage or robot is being controlled. You can use it to verify the operation of the alignment processing items even if the actual stage or robot is not available. The following example uses two Cameras to describe the use of the Machine Simulator processing item to verify the operation of a Vision Master Calibration processing item.

Example: Usage Example for Two Cameras

Camera Image Input processing item	· · · Select Camera 0.
↓	
Machine setting processing item	· · · Register [stage data] or [robot data] depending on the external device to be used, and then set the information for the external device. The machine settings processing item can be programmed anywhere other than at the beginning of the flow.
↓	
Set Unit Data processing item	· · · Register processing unit data setting processing items for the number of axes in the selected external device. For example, if you selected an XYθ stage, there are three axes. You would register three units in the flow. • Target unit: The next registered Machine Simulator processing item • Data No.: Each axis movement (number 128 to 135) • Settings data: The next set of axis movements (NMX to NMR) for the Vision Master Calibration processing item
↓	
Machine Simulator processing item	· · · Set the machine setting processing item to reference, and then set the conditions of the actual coordinates to confirm
↓	
Position measurement processing item	· · · Measure the position of the alignment marks, or the position of a special feature of the workpiece from the image in Camera 0. Out of the general measurement processing items (Inspections and Measurements), you can use processing items with measurement results that have an X and Y in the calculation strings. To output the position on the image, set [Calibration] on the [Output parameter] tab page to OFF.
↓	
Camera Switching processing item	· · · Select Camera 1.
↓	
Set Unit Data processing item	· · · Register processing unit data setting processing items for the number of axes in the selected external device. For example, if you selected an XYθ stage, there are three axes. You would register three units in the flow. • Target unit: The next registered Machine Simulator processing item • Data No.: Each axis movement (number 128 to 135) • Settings data: The next set of axis movements (NMX to NMR) for the Vision Master Calibration processing item
↓	
Machine Simulator processing item	· · · Set the machine setting processing item to reference, and then set the conditions of the actual coordinates to confirm

↓

Position measurement processing item	· · ·	Measure the position of the alignment marks or the position of a special feature of the workpiece from the image in Camera 1. To output the position on the image, set [Calibration] on the [Output parameter] tab page to OFF. Out of the general measurement processing items (Inspections and Measurements), you can use processing items with measurement results that have an X and Y in the calculation strings.
↓		
Vision Master Calibration processing item	· · ·	Set the machine setting processing item to reference, and then set the sampling.

- **Using the Flow**

  The motionless workpiece (or a saved image) is continuously measured until the calibration is completed. This allows you to easily verify the operation of the calibration flow with the FH/FZ5-series Controller. Conventionally, you could not verity the calibration flow unless the Controller was integrated into the system.

Refer to ▶ *Machine Simulator* of the *Vision System FH/FZ5 Series Processing Item Function Reference Manual* (Cat No. Z341).

Alignment Processing Items

The following processing items can be used as alignment processing items.
Refer to the following manual for details on processing items.
Reference: ▶ *Vision System FH/FZ5 Series Processing Item Function Reference Manual* (Cat No. Z341).

Processing item category	Sub category	Processing item name	Description
Measurement support processing items (Inspection and measurement support items)	Machine setting processing items	Stage Data	This processing item sets and retains data that is related to a stage. Enter the data based on the specifications of the stage to use.
		Robot Data	This processing item sets and retains data that is related to the robot. Enter the data based on the specifications of the robot to use.
	Calibration processing items	Vision Master Calibration	This processing item is specifically for the calibration of the Camera coordinates and external device coordinates. This processing item automatically calculates all axis movements of the external device that are required for calibration. This allows the calibration process to be carried out with greater ease and precision.
		PLC Master Calibration	This processing item is specifically for the calibration of the Camera coordinates and external device coordinates. This processing item allows you to freely set the axis movement of the external device that is required for calibration.
	Reference Calib Data	Reference Calib Data	This processing item is used to reference calibration data and distortion data that is stored in other processing items. The referenced data is used to perform coordinate conversion processing and distortion correction on the measurement results that follow this processing item.
	Alignment calculations	Movement Single Position	This processing item calculates all axis movements of the external device that are required to match the measured position and angle to the reference position and angle.
		Movement Multi Points	This processing item calculates all axis movements of the external device that are required to match the measured position to the reference position. You can set up to eight points each for the measured position and reference position.
		Convert Position Data	This processing item calculates the position data and angle data after each axis of the external device is moved by a required amount. Use this processing item when you want to move the stage by a required amount, and then perform a measurement to obtain the position data and angle data before the stage was moved.
		Position Data Calculation	This processing item uses the measurement results from more than one processing item to calculate the position data and angle data to be used in a Movement Single Position processing item.
Image Correction processing items ([Compensate image])	Scrolling	Machine Simulator	This processing item allows you to see how the alignment marks move on the image while each axes of the external device is being controlled. You can use it to verify the operation of the alignment processing items even if the actual stage or robot is not available.

Error List

If an error occurs during execution of the program of a macro customize function, you can identify the cause of the error from the error number and error message that are displayed. The error messages that may appear are described below.

No.	Error Message	Explanation	Action
1	NEXT without FOR	A Next statement occurs without a corresponding For statement.	Write For properly in a For - Next statement.
2	Syntax error	Incorrect function format or spelling, or a function is not used in accordance with the rules.	Check the macro function reference and correct the function format or spelling.
3	RETURN without GOSUB	A Return statement occurs without a corresponding Gosub statement.	A Gosub function does not exist or is not written correctly. Write the Gosub function correctly.
5	Illegal function call	The argument specified for a function is not within the allowed range,or an error occurred when the function was executed.	Occurs when a macro function argument is not within the allowed range. Check the macro function reference and set the macro function argument appropriately. If the macro function is executed from an inappropriate subroutine, execute from an appropriate subroutine.
6	Overflow	The calculation result or entered value is not within the allowed range for double precision real numbers, or the character string length is over the allowed range.	Occurs when a double precision real number value is not within the range -1.0E30 to 1.0E30. Correct the process so that the value is within the double precision real number range. Occurs when the length of a character string variable or constant is over 255 characters. Shorten the character string length.
7	Out of memory	Insufficient sensor controller working memory,or a loop process is nested too deep.	Occurs when too many character strings or array variables are used in a program. Decrease the number used until the error no longer occurs. Occurs when the number of nested levels in a loop process is too large. Decrease the number of nested levels until the error no longer occurs.
8	Undefined line number	The program branches to an undefined line number.	Occurs when a Goto or other function branches to a non-existent line number. Specify an existing line number, or use a <label> instead of a line number to specify the branch destination.
9	Subscript out of range	An attempt was made to use an array element with an index over the declared maximum.	Occurs when an attempt is made to access an array variable over the number of array variable elements declared in the Dim instruction. Change the element number to a number equal to or less than the number of elements, or change the number of declared elements.
11	Division by Zero	Division by zero was attempted.	Division by 0 is not possible. Divide by a value other than 0.
13	Type mismatch	Variable types do not match, such as on the left or right side of an expression, or in the argument of a function.	Occur when variable types do not match on the left or right side of an expression, or in the argument of a function. Specify the correct variable type or array variable. Occurs when a character string is assigned to a numerical value, or a numerical value is assigned to a character string. Assign a numerical value to a numerical value, and a character string to a character string.
15	String too long	The number of characters in a character string assigned to a character string variable is over 255.	If the character string variable is over 255 characters, break the string into two variables.

No.	Error Message	Explanation	Action
18	Undefined array	An undefined array is used.	Declare the array variable with a Dim instruction before using the array.
23	Line buffer overflow	An attempt was made to input more than the allowed character limit (255 bytes) on one line.	Generally occurs when one line of data is received by serial communication or from a memory card. Use the Input$ function to input the required number of bytes.
26	FOR without NEXT	A For statement occurs without a corresponding Next statement.	Occurs when a Next instruction does not exist or is not written correctly in a For - Next statement. Write the Next instruction correctly.
32	Undefined label	An attempt was made to access an undefined label.	Occurs when the referencing and referenced label names do not match. Use a defined label name.
121	CASE without SELECT	A Case statement occurs without a corresponding Select statement.	Occurs when a Select instruction does not exist or is not written correctly in a Select - Case statement. Write the Select instruction correctly.
122	END SELECT without SELECT	An End Select statement occurs without a corresponding Select statement.	Occurs when a Select instruction does not exist or is not written correctly in a Select - Case statement. Write the Select instruction correctly.
123	SELECT without END SELECT	A Select statement occurs without a corresponding End Select statement.	Occurs when an End Select instruction does not exist or is not written correctly in a Select - Case statement. Write the End - Select instruction correctly.
124	CASE without END SELECT	A Case statement occurs without a corresponding End Select statement.	Occurs when an End Select instruction does not exist or is not written correctly in a Select - Case statement. Write the End - Select instruction correctly.
125	ELSEIF without IF	An ElseIf statement occurs without a corresponding If statement.	Occurs when an If instruction does not exist or is not written correctly in an If Else statement. Write the If instruction correctly.
126	ELSE without IF	ELSE statement occurs without a corresponding If statement.	Occurs when an If instruction does not exist or is not written correctly in an If Else statement. Write the If instruction correctly.
127	ENDIF without IF	Endif statement occurs without a corresponding If statement.	Occurs when an If instruction does not exist or is not written correctly in an If - EndIf statement. Write the If instruction correctly.
128	IF without ENDIF	If statement occurs without a corresponding EndIf statement.	Occurs when an EndIf instruction does not exist or is not written correctly in an If - EndIf statement. Write the EndIf instruction correctly.
129	ELSEIF without ENDIF	EndIf statement occurs without a corresponding ElseIf statement.	Occurs when an EndIf instruction does not exist or is not written correctly in an If - EndIf statement. Write the EndIf instruction correctly.
130	ELSE without ENDIF	Else statement occurs without a corresponding Endif statement.	Occurs when an EndIf instruction does not exist or is not written correctly in an If - EndIf statement. Write the EndIf instruction correctly.
135	DO without LOOP	Do statement occurs without a corresponding Loop statement.	Occurs when an Loop instruction does not exist or is not written correctly in a Do Loop While statement. Write the Loop instruction correctly.
136	LOOP without DO	Loop statement occurs without a corresponding Do statement.	Occurs when an Do instruction does not exist or is not written correctly in a Do Loop While statement. Write the Do instruction correctly.
140	EXIT without FOR	An Exit Do statement occurs without a corresponding For statement.	Occurs when a For instruction does not exist or is not written correctly in a For Next statement. Write the For instruction correctly.
141	EXIT without DO	Exit Do statement occurs without a corresponding Do statement.	Occurs when an Do instruction does not exist or is not written correctly in a Do Loop While statement. Write the Do instruction correctly.

IMPORTANT

- When an error occurs in a process other than a communication command macro or Try - Catch - End Try, the error number cannot be used to check the error. Determine the nature of the error from the error message that appears in the system status console window.

Note

- Error information appears in the system status console window.

List of Reserved Words

Macro functions, operators, and other character strings predefined in the system are referred to as "reserved words". Other than in comments, do not use reserved words in macro customize programs. If a reserved word is inappropriately used, an error message appears.

List of Reserved Words

Reserved words that are defined in the FH/FZ5 series are shown below.

A

Abs	ACTIVE&	ACTIVETABLE&
AddGlobalData	AddSystemData	And
Append	ApproximationCircle	ARGMENTCOUNTMAX&
ARGMENTSLENGTH&	ARGMENTSTRING$	ARGMENTVALUE#
ARGUMENTSLENGTH&	ARGUMENTSTRING$	ARGUMENTVALUE#
As	Asc	AssignUnit
Atn		

B

BusyOut	BITPATTERN&	

C

Call	Case	Catch
ChangeScene	ChangeSceneGroup	CheckUnit
Chr$	Clear	ClearMeasureData
ClearScene	ClearSceneGroup	Close
CloseTextData	CMD$	COMMANDAREA&
COMMANDCODE&	COMMANDCOUNTMAX&	COMMANDDATA&
COMMANDDELIMITER$	COMMANDEXECUTE&	COMMANDLINE$
COMMANDMEMORYADDRESS&	COMMANDRESPONSE&	COMMANDRESPONSE&
COMMANDSTRING$	COMMENTSTRING$	Cont
CopyMeasureImage	CopyScene	CopySceneGroup
CopyUnit	CopyUnitFigure	

D

DATACOUNT&	Date$	Debug
DebugPrint	Delete	DeleteUnit
Dim	DisplaySubNo	DISPLAYTEXT$
DisplayUnitNo	Do	Dposline
DrawArc	DrawArcW	DrawBox
DrawCircle	DrawCircleW	DrawCursor
DrawEllipse	DrawFigure	DrawFillImage
DrawJudgeText	DrawLine	DrawLineW
DrawMeasureImage	DrawPoint	DrawPolygon
DrawSearchFigure	DrawText	DrawTextG
DrawUnitImage	Dskf	

E

ElapsedTime	Else	Elseif
End	EndIf	Eof
Erase	Erl	Err
Errcmnd$	Errno	Error
ExecuteImageLogging	Exit	ExitFzProcess
Exp		

F

FALSE	Fcopy	Files
Fix	FlowProcListSize	FlowUnitListSize
FONTSIZE_NORMAL	FONTSTYLE_BOLD	FONTSTYLE_ITALIC
FONTSTYLE_NORMAL	FONTSTYLE_STRIKEOUT	FONTSTYLE_UNDERLINE
For		

G

GetAll	GetGlobalData	GetImageSize
GetImageWindow	GetMeasureOut	GetPlcData
GetPollingState	GetPort	GetSceneData
GetSystemData	GetText$	GetTextWindow
GetUnitData	GetUnitFigure	Global
Gosub	Goto	

H

Hex$		

I

If	ImageFormat	ImageUpdate
initialLayoutNo	initialRemoteLayoutNo	Input
Input$	InsertUnit	Int
IOIDENT$	IsFile	ItemCount
ItemIdent$	ItemInfo	ItemTitle$

J

JGINDEX&	JUDGE&	JUDGE_ERROR
JUDGE_IMAGEERROR	JUDGE_MEMORYERROR	JUDGE_MODELERROR
JUDGE_NC	JUDGE_NG	JUDGE_OK
JUDGELOWER#	JUDGEMACROFLAG&	JudgeOut
JUDGEUPPER#		

K

Keyword	Kill	

L

Layout?_Title (? represents a number)	Layout?_WindowSetting (? represents a number)	Layout?_output (? represents a number)
Layout?_runout (? represents a number)	LayoutSetting?_? (? represents a number)	LCase$
Left$	Len	LF
Line	List	Load
LoadBackupData	LoadScene	LoadSceneGroup
LoadSystemData	LoadUnitData	Local
Log	Loop	

M

Measure	MeasureDispG	MeasureId$
MeasureProc	MeasureStart	MeasureStop
MEASURESTOPTABLE&	Mid$	Mkdir
Mod	MoveUnit	

N

New	Next	Not

O

On	Open	OpenTextData
Option	Or	Output

P

ParallelExecute	PARAOFFSET&	Piece$
PLCRCVDATA&	PLCSNDDATA&	Print
PS_DASH	PS_DASHDOT	PS_DASHDOTDOT
PS_DOT	PS_INSIDEFRAME	PS_NULL
PS_SOLID	PutAll	PutPort

R

RaiseOptionEvent	ReadPlcMemory	ReceiveData
RefreshImageWindow	RefreshJudgeWindow	RefreshTextWindow
RefreshTimeWindow	Rem	Remeasure
RenumUnitNo	RESPONSEAREA&	RESPONSECODE&
RESPONSEMEMORYADDRESS&	RESPONSESTRING$	RESPONSEVALUE&
RESULTDATA#	RESULTJUDGE&	Return
RGB	Right$	Rmdir
Run		

S

Save	SaveBackupData	SaveData
SaveImage	SaveMeasureImage	SaveScene
SaveSceneGroup	SaveSystemData	SaveUnitData
SceneCount	SceneDescription$	SceneGroupCount
sceneGroupDataPath	SceneGroupNo	SceneGroupTitle$
SceneMaker$	SceneNo	SceneTitle$
ScreenCapture	Select	SendData
SendString	SetDisplayUnitNo	SetDrawStyle
SetGlobalData	SetImageWindow	SetJudgeWindow
SetMeasureImage	SetMeasureOut	SetPlcData
SetPollingState	SetSceneData	SetSceneDescription
SetSceneGroupTitle	SetSceneMaker	SetSceneTitle
SetStop	SetSystemData	SetTextStyle
SetTextWindow	SetTimeWindow	SetUnitData
SetUnitFigure	SetUnitJudge	SetUnitTitle
SetUserSubroutine	SetVar	Sin
Sqr	StartTimer	StartupLanguageSet
StartupSceneCheck	Step	Stop
Str$	Str2$	Setupproc
Sub	SystemReset	

T

TA_BASELINE	TA_BOTTOM	TA_CENTER
TA_LEFT	TA_NOUPDATECP	TA_RIGHT
TA_RTLREADING	TA_TOP	TA_UPDATECP
TAB	Tan	Task
Then	Time$	Timer
TJGFLAG&	To	TotalJudge
TransformAngle	TransformArea	TransformDist
TransformLine	TransformXY	TRUE
Try		

U

UCase$	UnitCount	UnitData
UnitData$	UnitData2	UnitInfo
UnitItemIdent$	UnitJudge	UnitNo
UnitTitle$	Ut	

V

Val	VarList	VarPop
VarPtr	VarPush	VarSave

W

Wait	While	WritePlcMemory

X

Xor		

List of Reserved Global Data Words

Global data that is reserved in the FH/FZ5 series is shown below.

ControlFlowParallelCommand_status	editScnGroupNo	editScnNo
editState	ImageWindowOrigin_IoCommand	ImageWindowSubNoMax_IoCommand
LayoutNoLocal_IoCommand	LayoutNoRemote_IoCommand	LineNoLocal_IoCommand
LineNoRemote_IoCommand	ParallelDIOffset	RemoteOperationStatus_IoCommand

System Data List

The ID information and ID names required to set or acquire system data are indicated below.

Identifier information 0	Identifier information 1	Data identifier name
Basic	---	Configuration
	Language Sets the display language. jpn: Japanese eng: English chs: Simplified Chinese cht: Traditional Chinese kor: Korean deu: German fra: French esp: Spanish ita: Italian	Language
	Initial scene group number	initialSceneGroupNo
	Initial scene number	initialSceneNo
	Operation priority 0: Measurement result priority 1: Menu operation priority	operationPriority
	Measurement initialization priority 0: Measurement trigger receipt priority 1: Processing of re-drawing on screen priority	measureInitPriority
	Operation mode 0: High-speed Logging Mode 1: Single-line High-speed Mode 2: Multi-line Random-trigger Mode 3: Non-stop Adjustment Mode	operatingMode
	Parallel execution 0: OFF 1: ON	parallelExecute
	Number of scenes	SceneCount

Identifier information 0	Identifier information 1	Data identifier name
Communication	---	IoModule
	Serial (Ethernet) UdpNormal: Normal (UDP) TcpNormal: Normal (TCP) TcpClient: Normal (TCP Client) UdpNormal2: Normal (UDP) (Fxxx series method) UdpPlcLink: PLC Link (SYSMAC CS/CJ/CP/One) UdpPlcLinkM: PLC Link (MELSEC QnU/Q/QnAS series) UdpPlcLinkY: PLC Link (JEPMC MP)	ioIdent2
	Serial (RS-232C/422) SerialNormal: Normal SerialNormal2: Normal (Fxxx series method) SerialPlcLink: PLC Link (SYSMAC CS/CJ/CP/One) SerialPlcLinkM: PLC Link (MELSEC QnU/Q/QnAS series)	ioIdent1
	Parallel ParallelIo: Standard Parallel I/O	ioIdent0
	Fieldbus *1: Blank character: None EtherCAT: EtherCAT EtherNet/IP: EtherNet/IP	ioIdent3
	Remote Operation RemoteServer: Yes *1: Blank character: None	ioIdent4
Multi-line Random-trigger Mode Settings	---	MultiLineRandom
	Number of lines (2/3/4/5/6/7/8)	lineCount
	Camera - line assignment 0	physicalCameraMask0
	Camera - line assignment 1	physicalCameraMask1
	Camera - line assignment 2	physicalCameraMask2
	Camera - line assignment 3	physicalCameraMask3
	Camera - line assignment 4	physicalCameraMask4
	Camera - line assignment 5	physicalCameraMask5
	Camera - line assignment 6	physicalCameraMask6
	Camera - line assignment 7	physicalCameraMask7

Identifier information 0	Identifier information 1	Data identifier name
Camera settings	---	CameraControl
	STEP - camera 0 delay	cameraDelay0
	STEP - camera 1 delay	cameraDelay1
	STEP - camera 2 delay	cameraDelay2
	STEP - camera 3 delay	cameraDelay3
	STEP - camera 4 delay	cameraDelay4
	STEP - camera 5 delay	cameraDelay5
	STEP - camera 6 delay	cameraDelay6
	STEP - camera 7 delay	cameraDelay7
	Baud rate 0 0: Normal 1: Fast	transferRate0
	Baud rate 1 0: Normal 1: Fast	transferRate1
	Baud rate 2 0: Normal 1: Fast	transferRate2
	Baud rate 3 0: Normal 1: Fast	transferRate3
	Baud rate 4 0: Normal 1: Fast	transferRate4
	Baud rate 5 0: Normal 1: Fast	transferRate5
	Baud rate 6 0: Normal 1: Fast	transferRate6
	Baud rate 7 0: Normal 1: Fast	transferRate7
	Output signal setting 0: STGOUT 1: SHTOUT	timingSignal
	SHTOUT signal delay	shtoutDelay
	SHTOUT signal pulse width	shtoutWidth
	SHTOUT signal pulse polarity 0: Negative polarity 1: Positive polarity	shtoutPolarity

Identifier information 0	Identifier information 1	Data identifier name
Parallel	---	ParallelIo
	Output polarity 0: ON at NG 1: ON at OK	polarity
	Output control 0: OFF 1: Handshaking 2: Synchronization output	handshake
	Output cycle	cycleTime
	Gate ON delay	riseTime
	Output time	outputTime
	Timeout	timeout
	Number of delay	delayCount
	One-shot OR signal 0: OFF 1: ON	orOutMode
	OR signal output time	orOutputTime
Normal	---	SerialNormal
	Timeout	timeout
	Interface 0: RS-232C 1: RS-422	rsMode
	Baud rate (2,400, 4,800, 9,600, 19,200, 38,400, 57,600, or 115,200)	baudRate
	Data length (7 or 8)	byteSize
	Parity 0: OFF 1: Odd 2: Even	parity
	Stop bits (1 or 2)	stopBits
	Flow control 0: OFF 1: Xon/Xoff	softFlow
	Delimiter 0: CR 1: LF 2: CR + LF	delimiter

Identifier information 0	Identifier information 1	Data identifier name
Normal (Fxxx series method)	---	SerialNormal2
	Timeout	timeout
	Interface 0: RS-232C 1: RS-422	rsMode
	Baud rate (2,400, 4,800, 9,600, 19,200, 38,400, 57,600, 115,200)	baudRate
	Data length (7 or 8)	byteSize
	Parity 0: OFF 1: Odd 2: Even	parity
	Stop bits (1 or 2)	stopBits
	Flow control 0: OFF 1: Xon/Xoff	softFlow
	Delimiter 0: CR 1: LF 2: CR + LF	delimiter
RS-232C/422 PLC Link (SYSMAC CS/CJ/CP/One)	---	SerialPlcLink
	Output control 0: OFF 1: Handshaking	handshake
	Timeout	timeout
	Interface 0: RS-232C 1: RS-422	rsMode
	Baud rate (2,400, 4,800, 9,600, 19,200, 38,400, 57,600, 115,200)	baudRate
	Data length (7 or 8)	byteSize
	Parity 0: OFF 1: Odd 2: Even	parity
	Stop bits (1 or 2)	stopBits
	Flow control 0: OFF 1: Xon/Xoff	softFlow
	Delimiter 0: CR 1: LF 2: CR + LF	delimiter
	Command Area type	commandArea
	Command Area address	commandMemoryAddress
	Response Area type	responseArea
	Response Area address	responseMemoryAddress
	Data Output Area type	outputArea
	Data Output Area address	outputMemoryAddress

Identifier information 0	Identifier information 1	Data identifier name
RS-232C/422 PLC Link (SYSMAC CS/CJ/CP/One)	Asynchronous output 0: OFF 1: ON	outputBuffering
	Retry interval	responseTimeout
	Polling cycle	pollingMinCycle
RS-232C/422 PLC Link (MELSEC QnU/Q/QnAS)	---	SerialPlcLinkM
	Output control 0: OFF 1: Handshaking	handshake
	Timeout	timeout
	Interface 0: RS-232C 1: RS-422	rsMode
	Baud rate (2,400, 4,800, 9,600, 19,200, 38,400, 57,600, 115,200)	baudRate
	Data length (7 or 8)	byteSize
	Parity 0: OFF 1: Odd 2: Even	parity
	Stop bits (1 or 2)	stopBits
	Flow control 0: OFF 1: Xon/Xoff	softFlow
	Delimiter 0: CR 1: LF 2: CR + LF	delimiter
	Command Area type	commandArea
	Command Area address	commandMemoryAddress
	Response Area type	responseArea
	Response Area address	responseMemoryAddress
	Data Output Area type	outputArea
	Data Output Area address	outputMemoryAddress
	Asynchronous output 0: OFF 1: ON	outputBuffering
	Retry interval	responseTimeout
	Polling cycle	pollingMinCycle
Ethernet PLC Link (SYSMAC CS/CJ/CP/One)	---	UdpPlcLink
	Output control 0: OFF 1: Handshaking	handshake
	Command Area type	commandArea
	Command Area address	commandMemoryAddress
	Response Area type	responseArea
	Response Area address	responseMemoryAddress

Identifier information 0	Identifier information 1	Data identifier name
Ethernet PLC Link (SYSMAC CS/CJ/CP/One)	Data Output Area type	outputArea
	Data Output Area address	outputMemoryAddress
	Asynchronous output 0: OFF 1: ON	outputBuffering
	Retry interval	responseTimeout
	Retry interval 2	responseTimeout2
	Polling cycle	pollingMinCycle
	Automatic 0: OFF 1: ON	enableDhcp
	IP address	ipAddress
	Subnet mask	subnetMask
	Default gateway	defaultGateway
	DNS server	dns
	Output IP address	destIpAddress
	Input port number	portNo
	Output port number	portNo2
Ethernet PLC Link (MELSEC QnU/Q/QnAS)	---	UdpPlcLinkM
	Output control 0: OFF 1: Handshaking	handshake
	Command Area type	commandArea
	Command Area address	commandMemoryAddress
	Response Area type	responseArea
	Response Area address	responseMemoryAddress
	Data Output Area type	outputArea
	Data Output Area address	outputMemoryAddress
	Asynchronous output 0: OFF 1: ON	outputBuffering
	Retry interval	responseTimeout
	Retry interval 2	responseTimeout2
	Polling cycle	pollingMinCycle
	Automatic 0: OFF 1: ON	enableDhcp
	IP address	ipAddress
	Subnet mask	subnetMask
	Default gateway	defaultGateway
	DNS server	dns
	Output IP address	destIpAddress
	Input port number	portNo
	Output port number	portNo2

Identifier information 0	Identifier information 1	Data identifier name
Normal (UDP)	---	UdpNormal
	Automatic 0: OFF 1: ON	enableDhcp
	IP address	ipAddress
	Subnet mask	subnetMask
	Default gateway	defaultGateway
	DNS server	dns
	Output IP address	destIpAddress
	Input port number	portNo
	Output port number	portNo2
Normal (TCP)	---	TcpNormal
	Automatic 0: OFF 1: ON	enableDhcp
	IP address	ipAddress
	Subnet mask	subnetMask
	Default gateway	defaultGateway
	DNS server	dns
	Output IP address	destIpAddress
	Input port number	portNo
	Output port number	portNo2
Normal (TCP Client)	---	TcpClient
	Automatic 0: OFF 1: ON	enableDhcp
	IP address	ipAddress
	Subnet mask	subnetMask
	Default gateway	defaultGateway
	DNS server	dns
	Output IP address	destIpAddress
	Input port number	portNo
	Output port number	portNo2

Identifier information 0	Identifier information 1	Data identifier name
Normal (UDP Fxxx series method)	---	UdpNormal2
	Automatic 0: OFF 1: ON	enableDhcp
	IP address	ipAddress
	Subnet mask	subnetMask
	Default gateway	defaultGateway
	DNS server	dns
	Output IP address	destIpAddress
	Input port number	portNo
	Output port number	portNo2
	Connect to	serverIpAddress
Ethernet PLC Link (JEPMC MP)	---	UdpPlcLinkY
	Output control 0: OFF 1: Handshaking	handshake
	Command Area type	commandArea
	Command Area address	commandMemoryAddress
	Response Area type	responseArea
	Response Area address	responseMemoryAddress
	Data Output Area type	outputArea
	Data Output Area address	outputMemoryAddress
	Asynchronous output 0: OFF 1: ON	outputBuffering
	Retry interval	responseTimeout
	Retry interval 2	responseTimeout2
	Polling cycle	pollingMinCycle
	Automatic 0: OFF 1: ON	enableDhcp
	IP address	ipAddress
	Subnet mask	subnetMask
	Default gateway	defaultGateway
	DNS server	dns
	Output IP address	destIpAddress
	Input port number	portNo
EtherCAT	---	EtherCAT
	Output control 0: OFF 1: Handshaking	handshake
	Output cycle	cycleTime
	Output time	outputTime
	Timeout	timeout

Identifier information 0	Identifier information 1	Data identifier name
EtherCAT	Data output number for line 0 (number of data output items for line 0) 32: Result Data Format 0 (8 DINT) 64: Result Data Format 1 (16 DINT) 128: Result Data Format 2 (32 DINT) 256: Result Data Format 3 (64 DINT) 2097152: Result Data Format 4 (4 LREAL) 4194304: Result Data Format 5 (8 LREAL) 8388608: Result Data Format 6 (16 LREAL) 16777216: Result Data Format 7 (32 LREAL) 1572872: Result Data Format 8 (2 DINT + 3 LREAL) 3145744: Result Data Format 9 (4 DINT + 6 LREAL) 6291488: Result Data Format 10 (8 DINT + 12 LREAL) 12582976: Result Data Format 11 (16 DINT + 24 LREAL)	outputDataSize0
	Data output number for line 1 (number of data output items for line 1) 32: Result Data Format 0 (8 DINT) 64: Result Data Format 1 (16 DINT) 128: Result Data Format 2 (32 DINT) 256: Result Data Format 3 (64 DINT) 2097152: Result Data Format 4 (4 LREAL) 4194304: Result Data Format 5 (8 LREAL) 8388608: Result Data Format 6 (16 LREAL) 16777216: Result Data Format 7 (32 LREAL) 1572872: Result Data Format 8 (2 DINT + 3 LREAL) 3145744: Result Data Format 9 (4 DINT + 6 LREAL) 6291488: Result Data Format 10 (8 DINT + 12 LREAL) 12582976: Result Data Format 11 (16 DINT + 24 LREAL)	outputDataSize1
	Data output number for line 2 (number of data output items for line 2) 32: Result Data Format 0 (8 DINT) 64: Result Data Format 1 (16 DINT) 128: Result Data Format 2 (32 DINT) 256: Result Data Format 3 (64 DINT) 2097152: Result Data Format 4 (4 LREAL) 4194304: Result Data Format 5 (8 LREAL) 8388608: Result Data Format 6 (16 LREAL) 16777216: Result Data Format 7 (32 LREAL) 1572872: Result Data Format 8 (2 DINT + 3 LREAL) 3145744: Result Data Format 9 (4 DINT + 6 LREAL) 6291488: Result Data Format 10 (8 DINT + 12 LREAL) 12582976: Result Data Format 11 (16 DINT + 24 LREAL)	outputDataSize2

Identifier information 0	Identifier information 1	Data identifier name
EtherCAT	Data output number for line 3 (number of data output items for line 3) 32: Result Data Format 0 (8 DINT) 64: Result Data Format 1 (16 DINT) 128: Result Data Format 2 (32 DINT) 256: Result Data Format 3 (64 DINT) 2097152: Result Data Format 4 (4 LREAL) 4194304: Result Data Format 5 (8 LREAL) 8388608: Result Data Format 6 (16 LREAL) 16777216: Result Data Format 7 (32 LREAL) 1572872: Result Data Format 8 (2 DINT + 3 LREAL) 3145744: Result Data Format 9 (4 DINT + 6 LREAL) 6291488: Result Data Format 10 (8 DINT + 12 LREAL) 12582976: Result Data Format 11 (16 DINT + 24 LREAL)	outputDataSize3
	Data output number for line 4 (number of data output items for line 4) 32: Result Data Format 0 (8 DINT) 64: Result Data Format 1 (16 DINT) 128: Result Data Format 2 (32 DINT) 256: Result Data Format 3 (64 DINT) 2097152: Result Data Format 4 (4 LREAL) 4194304: Result Data Format 5 (8 LREAL) 8388608: Result Data Format 6 (16 LREAL) 16777216: Result Data Format 7 (32 LREAL) 1572872: Result Data Format 8 (2 DINT + 3 LREAL) 3145744: Result Data Format 9 (4 DINT + 6 LREAL) 6291488: Result Data Format 10 (8 DINT + 12 LREAL) 12582976: Result Data Format 11 (16 DINT + 24 LREAL)	outputDataSize4
	Data output number for line 5 (number of data output items for line 5) 32: Result Data Format 0 (8 DINT) 64: Result Data Format 1 (16 DINT) 128: Result Data Format 2 (32 DINT) 256: Result Data Format 3 (64 DINT) 2097152: Result Data Format 4 (4 LREAL) 4194304: Result Data Format 5 (8 LREAL) 8388608: Result Data Format 6 (16 LREAL) 16777216: Result Data Format 7 (32 LREAL) 1572872: Result Data Format 8 (2 DINT + 3 LREAL) 3145744: Result Data Format 9 (4 DINT + 6 LREAL) 6291488: Result Data Format 10 (8 DINT + 12 LREAL) 12582976: Result Data Format 11 (16 DINT + 24 LREAL)	outputDataSize5

Identifier information 0	Identifier information 1	Data identifier name
EtherCAT	Data output number for line 6 (number of data output items for line 6) 32: Result Data Format 0 (8 DINT) 64: Result Data Format 1 (16 DINT) 128: Result Data Format 2 (32 DINT) 256: Result Data Format 3 (64 DINT) 2097152: Result Data Format 4 (4 LREAL) 4194304: Result Data Format 5 (8 LREAL) 8388608: Result Data Format 6 (16 LREAL) 16777216: Result Data Format 7 (32 LREAL) 1572872: Result Data Format 8 (2 DINT + 3 LREAL) 3145744: Result Data Format 9 (4 DINT + 6 LREAL) 6291488: Result Data Format 10 (8 DINT + 12 LREAL) 12582976: Result Data Format 11 (16 DINT + 24 LREAL)	outputDataSize6
	Data output number for line 7 (number of data output items for line 7) 32: Result Data Format 0 (8 DINT) 64: Result Data Format 1 (16 DINT) 128: Result Data Format 2 (32 DINT) 256: Result Data Format 3 (64 DINT) 2097152: Result Data Format 4 (4 LREAL) 4194304: Result Data Format 5 (8 LREAL) 8388608: Result Data Format 6 (16 LREAL) 16777216: Result Data Format 7 (32 LREAL) 1572872: Result Data Format 8 (2 DINT + 3 LREAL) 3145744: Result Data Format 9 (4 DINT + 6 LREAL) 6291488: Result Data Format 10 (8 DINT + 12 LREAL) 12582976: Result Data Format 11 (16 DINT + 24 LREAL)	outputDataSize7
EtherNet/IP	---	EtherNetIP
	Output control 0: OFF 1: Handshaking	handshake
	Output cycle	cycleTime
	Output time	outputTime
	Timeout	timeout
measurement	---	Measure
	Fan control setting 0: Slow rotation 1: Fast rotation	fanControl
	Destination folder	captureDirectory
	STEP in measure 0: ERROR signal output OFF for STEP input during measurement 1: ERROR signal output ON for STEP input during measurement	stepError
	Scene group switch 0: Do not save when changing scene group 1: Save when changing scene group	sceneGroupSave
	Scene switch time	sceneWaitTime

Identifier information 0	Identifier information 1	Data identifier name
	---	Trigger
	STEP signal filter width 5: 100 11: 200 18: 300 24: 400 30: 500	digitalFilter
	Use Encoder trigger 0: Not used 1 : Used	encoderEnabled
	Encoder input selection 0: Open collector 1: Line driver	encoderInputPulse
	Phase A resolution	encoderResolutionA
	Count rotations with phase Z 0: Do not count 1: Count	encoderZ
Other	Encoder start	encoderStart
	Encoder count	encoderCount
	Phase A trigger timing usage flag 0	encoderTriggerAngleA0
	Phase A trigger timing usage flag 1	encoderTriggerAngleA1
	Phase A trigger timing usage flag 2	encoderTriggerAngleA2
	Phase A trigger timing usage flag 3	encoderTriggerAngleA3
	Phase A trigger timing usage flag 4	encoderTriggerAngleA4
	Phase A trigger timing usage flag 5	encoderTriggerAngleA5
	Support backlashing (phase B) 0: Do not support 1: Support	encoderBacklash
	Trigger in backlashing 0: Do not generate 1: Generate	encoderBacklashTrigger
	Phase A	encoderTriggerCountA
	Phase Z	encoderTriggerCountZ
	---	NetworkDrive
	S	drive0
	T	drive1
	U	drive2
Network drive setting	V	drive3
	W	drive4
	X	drive5
	Y	drive6
	Z	drive7
	---	Logging
Logging setting	Image Logging 0: None 1: Save for only NG 2: All	imageLogging

Identifier information 0	Identifier information 1	Data identifier name
Logging setting	Destination Specify the image file save destination (RAM disk or USB memory).	imageLoggingDirectory
	Prefix	imageLoggingHeader
	Switch saving folder by scene 0: Disabled 1: Enabled	imageLoggingScene
	Switch saving folder by judge 0: Disabled 1: Enabled	imageLoggingJudge
	Logging priority 0: Give priority to logging 1: Give priority to measurement takt	imageLoggingPriority
	Data Logging 0: None 1: Save for only NG 2: All	dataLogging
	Destination folder name	dataLoggingDirectory
User definition	---	PanDA

List of I/O Modules

I/O module settings required for communication with external devices are indicated below.

Identification name	IO module name	References
EtherNetIP	EtherNet/IP Interface communication	Reference: Details (p.341)
ParallelIo	Parallel Interface communication	Reference: Details (p.343)
SerialNormal SerialNormal2 (Fxxx series method)	Serial Interface Normal communication	Reference: Details (p.344)
SerialPlcLinkM	Serial Interface PLC Link (MELSEC QnU/Q/QnAS) communication	Reference: Details (p.346)
SerialPlcLink	Serial Interface PLC Link (SYSMAC CS/CJ/CP/One) communication	Reference: Details (p.348)
TcpClient	TCP Client Normal communication	Reference: Details (p.351)
TcpNormal	TCP Normal communication	Reference: Details (p.352)
UdpNormal UdpNormal2 (Fxxx series method)	UDP Normal communication	Reference: Details (p.354)
UdpPlcLinkM	PLC Link (MELSEC QnU/Q/QnAS) communication	Reference: Details (p.356)
UdpPlcLinkY	PLC Link (JEPMC MP) communication	Reference: Details (p.358)
UdpPlcLink	PLC Link (SYSMAC CS/CJ/CP/One) communication	Reference: Details (p.361)

EtherNetIP

EtherNet/IP Interface communication

IoModule identification name

EtherNetIP

Overview

This is a module is for sending and receiving commands and data by Ethernet/IP protocol.

System data

Identification	Meaning	Initial value
handshake	Output control 0: OFF 1: Handshaking	0
cycleTime	Output period [ms]	100
outputTime	Output time [ms]	50
timeout	Timeout [s]	100

Supported functions

IoInitialize	OK	---
GetPort	OK	Reference: ▶ GetPort (p.542)
PutPort	OK	Reference: ▶ PutPort (p.645)

BusyOut	---	---
JudgeOut	---	---
RunOut	---	---
GetAll	OK	Reference: ▶ GetAll (p.528)
PutAll	OK	Reference: ▶ PutAll (p.643)
ReceiveData	OK	Reference: ▶ ReceiveData (p.651)
SendData	OK	Reference: ▶ SendData (p.696)
SendString	OK	Reference: ▶ SendString (p.698)
ReadPlcMemory	---	---
WritePlcMemory	---	---
SetPlcData	---	---
GetPlcData	---	---

Example

Receive data

```
Dim data&(256)
Dim ipaddr&(4)
' Gets the five data.
ReceiveData "UdpPlcLink", data5&(), 4*5, size&
```

Transmit data.

Set an IP address and parameter size (*4 integer type domain) in a parameter to use an Ethernet.

```
' Transmit the five data.
SendData "UdpPlcLink", data&(), 4*5
```

ParallelIo

Parallel Interface communication

IoModule identification name

ParallelIo

Overview

This is a module is for sending and receiving commands and data via the Parallel interface.

System data

Identification	Meaning	Initial value
polarity	Output polarity 0: ON when NG 1: ON when OK	0
handshake	Output control 0: None 1: Handshake 2: Synchronous output	0
cycleTime	Cycle time [0.1ms]	100
riseTime	Rise time [0.1ms]	10
outputTime	Output time [0.1ms]	50
timeout	Timeout [0.1ms]	100
delayCount	Number of delay	1
orOutMode	One-shot OR signal 0: OFF 1: ON	0
orOutputTime	Time of one-shot output when OK signal [0.1ms]	50

Supported functions

IoInitialize	OK	---
GetPort	OK	Reference: ▶ GetPort (p.542)
PutPort	OK	Reference: ▶ PutPort (p.645)
BusyOut	OK	Reference: ▶ BusyOut (p.419)
JudgeOut	OK	Reference: ▶ JudgeOut (p.585)
RunOut	OK	Reference: ▶ RunOut (p.667)
GetAll	OK	Reference: ▶ GetAll (p.528)
PutAll	OK	Reference: ▶ PutAll (p.643)
ReceiveData	OK	Reference: ▶ ReceiveData (p.651)
SendData	OK	Reference: ▶ SendData (p.696)
SendString	OK	Reference: ▶ SendString (p.698)
ReadPlcMemory	---	---
WritePlcMemory	---	---
SetPlcData	---	---
GetPlcData	---	---

Example

Receive data

A parameter of ReceiveData, the parameter size are unnecessary.

```
Dim data&(256)

'Gets the five data.
ReceiveData "ParallelIo", data&(), 4*5, size&
```

Send data

A parameter of SendData, the parameter size are unnecessary.

```
Dim data&(256)

'Transmit the five data.
SendData "ParallelIo", data&(), 4*5
```

SerialNormal

Serial Interface Normal communication

IoModule identification name

SerialNormal

SerialNormal2 (Fxxx series method)

Overview

This is a module is for sending and receiving commands and data via the serial interface.

System data

Identification	Meaning	Initial value
rsMode	Interface 0: RS-232C 1: RS-422	0
baudRate	Baud rate [bps]	38400
byteSize	Data length [bit] 7 or 8	8
parity	Parity 0: None 1: Odd number 2: Even number	0
stopBits	Stop bits [bit] 0: 1 1: 2	0
softFlow	Flow control 0: None 1: Xon/Xoff	0
delimiter	Delimiter 0: CR 1: LF 2: CR + LF	0
timeout	Timeout [s]	5

Supported functions

IoInitialize	OK	---
GetPort	OK	Reference: ▶ GetPort (p.542)
PutPort	OK	Reference: ▶ PutPort (p.645)
BusyOut	---	---
JudgeOut	---	---
RunOut	---	---
GetAll	OK	Reference: ▶ GetAll (p.528)
PutAll	OK	Reference: ▶ PutAll (p.643)
ReceiveData	OK	Reference: ▶ ReceiveData (p.651)
SendData	OK	Reference: ▶ SendData (p.696)
SendString	OK	Reference: ▶ SendString (p.698)
ReadPlcMemory	---	---
WritePlcMemory	---	---
SetPlcData	---	---
GetPlcData	---	---

Example

Receive data

A parameter of ReceiveData, the parameter size are unnecessary.

```
Dim data&(256)

'Gets the five data.
ReceiveData "SerialPlcLink", data&(), 4*5, size&
```

Send data

A parameter of ReceiveData, the parameter size are unnecessary.

```
Dim data&(256)

'Transmit the five data.
SendData "SerialPlcLink", data&(), 4*5
```

SerialPlcLinkM

Serial Interface PLC Link (MELSEC QnU/Q/QnAS) communication

IoModule identification name

SerialPlcLinkM

Overview

This is a module is for sending and receiving commands and data via the serial PLC Link interface.

System data

Identification	Meaning	Initial value
rsMode	Interface 0: RS-232C 1: RS-422	0
baudRate	Baud rate [bps]	9600
byteSize	Data length [bit] 7 or 8	7
parity	Parity 0: None 1: Odd number 2: Even number	2
stopBits	Stop bits [bit] 0: 1 1: 2	1
softFlow	Flow control 0: None 1: Xon/Xoff	0
timeout	Timeout[s]	5

PLC Link data

Identification	Meaning	Initial value
commandArea	Command area Area	CIO Area (CIO)
commandMemoryAddress	Command area Address	0
responseArea	Response area Area	CIO Area (CIO)
responseMemoryAddress	Response area Address	100
outputArea	Data Output area Area	CIO Area (CIO)
outputMemoryAddress	Data Output area Address	200
handshake	Handshaking	1
responseTimeout	Retry interval [ms]	10000

Area classification

Area classification name	Area classification number
Data register	168
File register	175
Link register	180

Supported functions

IoInitialize	OK	---
GetPort	OK	Reference: ▶ GetPort (p.542)
PutPort	OK	Reference: ▶ PutPort (p.645)
BusyOut	---	---
JudgeOut	---	---
RunOut	---	---
GetAll	OK	Reference: ▶ GetAll (p.528)
PutAll	OK	Reference: ▶ PutAll (p.643)
ReceiveData	OK	Reference: ▶ ReceiveData (p.651)
SendData	OK	Reference: ▶ SendData (p.696)
SendString	OK	Reference: ▶ SendString (p.698)
ReadPlcMemory	OK	Reference: ▶ ReadPlcMemory (p.649)
WritePlcMemory	OK	Reference: ▶ WritePlcMemory (p.807)
SetPlcData	OK	Reference: ▶ SetPlcData (p.709)
GetPlcData	OK	Reference: ▶ GetPlcData (p.538)

Example

Receive data

A parameter of ReceiveData, the parameter size are unnecessary.

```
Dim data&(256)

'Gets the five data.
ReceiveData "SerialPlcLinkM", data&(), 4*5, size&
```

Send data

A parameter of ReceiveData, the parameter size are unnecessary.

```
Dim data&(256)

'Transmit the five data.
SendData "SerialPlcLinkM", data&(), 4*5
```

SerialPlcLink

Serial Interface PLC Link (SYSMAC CS/CJ/CP/One) communication

IoModule identification name

SerialPlcLink

Overview

This is a module is for sending and receiving commands and data via the serial PLC Link interface.

System data

Identification	Meaning	Initial value
rsMode	Interface 0: RS-232C 1: RS-422	0
baudRate	Baud rate [bps]	9600
byteSize	Data length [bit] 7 or 8	7
parity	Parity 0: None 1: Odd number 2: Even number	2
stopBits	Stop bits [bit] 0: 1 1: 2	1
softFlow	Flow control 0: None 1: Xon/Xoff	0
timeout	Timeout [s]	5

PLC Link data

Identification	Meaning	Initial value
commandArea	Command area Area	CIO Area (CIO)
commandMemoryAddress	Command area Address	0
responseArea	Response area Area	CIO Area (CIO)
responseMemoryAddress	Response area Address	100
outputArea	Data Output area Area	CIO Area (CIO)
outputMemoryAddress	Data Output area Address	200
handshake	Handshaking	1
responseTimeout	Retry interval [ms]	10000

Area classification

Area classification name	Area classification number
CIO Area (CIO)	176
Work Area (WR)	177
Holding Bit Area(HR)	178
Auxiliary Bit Area (AR)	179
DM Area (DM)	130
EM Area (EM0)	160
EM Area (EM1)	161
EM Area (EM2)	162
EM Area (EM3)	163
EM Area (EM4)	164
EM Area (EM5)	165
EM Area (EM6)	166
EM Area (EM7)	167
EM Area (EM8)	168
EM Area (EM9)	169
EM Area (EMA)	170
EM Area (EMB)	171
EM Area (EMC)	172

Supported functions

IoInitialize	OK	---
GetPort	OK	Reference: ▶ GetPort (p.542)
PutPort	OK	Reference: ▶ PutPort (p.645)
BusyOut	---	---
JudgeOut	---	---
RunOut	---	---
GetAll	OK	Reference: ▶ GetAll (p.528)
PutAll	OK	Reference: ▶ PutAll (p.643)
ReceiveData	OK	Reference: ▶ ReceiveData (p.651)
SendData	OK	Reference: ▶ SendData (p.696)
SendString	OK	Reference: ▶ SendString (p.698)
ReadPlcMemory	OK	Reference: ▶ ReadPlcMemory (p.649)
WritePlcMemory	OK	Reference: ▶ WritePlcMemory (p.807)
SetPlcData	OK	Reference: ▶ SetPlcData (p.709)
GetPlcData	OK	Reference: ▶ GetPlcData (p.538)

Example

Receive data

A parameter of ReceiveData, the parameter size are unnecessary.

```
Dim data&(256)

'Gets the five data.
ReceiveData "SerialPlcLink", data&(), 4*5, size&
```

Send data

A parameter of ReceiveData, the parameter size are unnecessary.

```
Dim data&(256)

' Transmit the five data.
SendData "SerialPlcLink", data&(), 4*5
```

Gets 7ch data from 10ch of the DM area.
Gets the data from readData().

```
Dim readData&(256)
Dim data3$(21)

'Gets the data from PLC
ReadPlcMemory "SerialPlcLink", 130, 10, 7, readData&()

'Gets the data of the real number type
GetPlcData "SerialPlcLink", readData&(), 0, 8, data1#
'Gets the data of the integer type
GetPlcData "SerialPlcLink", readData&(), 8, 4, data2&
'Gets the data of the character type
GetPlcData "SerialPlcLink", readData&(), 12, 5, data2&
```

Set data to writeData().
Write in data for 7ch from 10ch of the DM area.

```
Dim writeData&(256)

'Set the data (123.45) of the real number type.
SetPlcData "SerialPlcLink", writeData&(), 0, 8, 123.45
'Set the data (20) of the integer type.
SetPlcData "SerialPlcLink", writeData&(), 32, 4, 20
'Set the data (OMRON) of the character type.
SetPlcData "SerialPlcLink", writeData&(), 36, 5, "OMRON"

'Write in data for 7ch from 10ch of the DM area.
WritePlcMemory "SerialPlcLink", 130, 10, 7, writeData&()
```

TcpClient

TCP Client Normal communication

IoModule identification name

TcpClient

Overview

This is a module is for sending and receiving commands and data by Ethernet TCP Client protocol.

System data

Identification	Meaning	Initial value
enableDhcp	Enable DHCP 0: Disabled 1: Enabled	0
ipAddress	IP address of the system	10.5.5.100
subnetMask	Subnet mask	255.255.255.0
defaultGateway	Default gateway	10.5.5.110
dns	DNS server	10.5.5.1
serverIpAddress	Server IP address	10.5.5.101
portNo	Port number to receive commands	9600

Supported functions

IoInitialize	OK	---
GetPort	OK	Reference: ▶ GetPort (p.542)
PutPort	OK	Reference: ▶ PutPort (p.645)
BusyOut	---	---
JudgeOut	---	---
RunOut	---	---
GetAll	OK	Reference: ▶ GetAll (p.528)
PutAll	OK	Reference: ▶ PutAll (p.643)
ReceiveData	OK	Reference: ▶ ReceiveData (p.651)
SendData	OK	Reference: ▶ SendData (p.696)
SendString	OK	Reference: ▶ SendString (p.698)
ReadPlcMemory	---	Reference: ▶ ReadPlcMemory (p.649)
WritePlcMemory	---	Reference: ▶ WritePlcMemory (p.807)
SetPlcData	---	Reference: ▶ SetPlcData (p.709)
GetPlcData	---	Reference: ▶ GetPlcData (p.538)

Example

Receive data.

Set an IP address and parameter size (*4 integer type domain) in a parameter to use an Ethernet.

```
Dim data&(256)
Dim ipaddr&(4)
'Set the IP address of the destination.
ipaddr&(0) = 10
ipaddr&(1) = 5
ipaddr&(2) = 5
ipaddr&(3) = 101
'Gets the five data.
ReceiveData "TcpClient", data&(), 4*5, size&, ipaddr&(), 4*4
```

Send data

Set an IP address and parameter size (*4 integer type domain) in a parameter to use an Ethernet.

```
Dim data&(256)
Dim ipaddr&(4)
'Set the IP address of the destination.
ipaddr&(0) = 10
ipaddr&(1) = 5
ipaddr&(2) = 5
ipaddr&(3) = 101
'Transmit the five data.
SendData "TcpClient", data&(), 4*5, ipaddr&(), 4*4
```

TcpNormal

TCP Normal communication

IoModule identification name

TcpNormal

Overview

This is a module is for sending and receiving commands and data by Ethernet TCP protocol.

System data

Identification	Meaning	Initial value
enableDhcp	Enable DHCP 0: Disabled 1: Enabled	0
ipAddress	IP address of the system	10.5.5.100
subnetMask	Subnet mask	255.255.255.0
defaultGateway	Default gateway	10.5.5.110
dns	DNS server address	10.5.5.1
portNo	Port number to receive commands	9600

Supported functions

IoInitialize	OK	---
GetPort	OK	Reference: ▶ GetPort (p.542)
PutPort	OK	Reference: ▶ PutPort (p.645)
BusyOut	---	---
JudgeOut	---	---
RunOut	---	---
GetAll	OK	Reference: ▶ GetAll (p.528)
PutAll	OK	Reference: ▶ PutAll (p.643)
ReceiveData	OK	Reference: ▶ ReceiveData (p.651)
SendData	OK	Reference: ▶ SendData (p.696)
SendString	OK	Reference: ▶ SendString (p.698)
ReadPlcMemory	---	---
WritePlcMemory	---	---
SetPlcData	---	---
GetPlcData	---	---

Example

Receive data

Set an IP address and parameter size (*4 integer type domain) in a parameter to use an Ethernet.

```
Dim data&(256)
Dim ipaddr&(4)
'Set the IP address of the destination.
ipaddr&(0) = 10
ipaddr&(1) = 5
ipaddr&(2) = 5
ipaddr&(3) = 101
'Gets the five data.
ReceiveData "TcpNormal", data&(), 4*5, size&, ipaddr&(), 4*4
```

Send data

Set an IP address and parameter size (*4 integer type domain) in a parameter to use an Ethernet.

```
Dim data&(256)
Dim ipaddr&(4)
'Set the IP address of the destination.
ipaddr&(0) = 10
ipaddr&(1) = 5
ipaddr&(2) = 5
ipaddr&(3) = 101
'Transmit the five data.
SendData "TcpNormal", data&(), 4*5, ipaddr&(), 4*4
```

UdpNormal

TCP Normal communication

IoModule identification name

UdpNormal
UdpNormal2 (Fxxx series method)

Overview

This is a module is for sending and receiving commands and data by Ethernet UDP protocol.

System data

Identification	Meaning	Initial value
enableDhcp	Enable DHCP 0: Disabled 1: Enabled	0
ipAddress	IP address of the system	10.5.5.100
subnetMask	Subnet mask	255.255.255.0
defaultGateway	Gateway address	10.5.5.110
dns	DNS server address	10.5.5.1
destIpAddress	Destination IP address to send data	0.0.0.0
portNo	Port number to receive commands	9600
portNo2	Port number to send data	-1

(*) If the input port number and the output port number are the same setting, set the output port number to -1.

Supported functions

IoInitialize	OK	---
GetPort	OK	Reference: ▶ GetPort (p.542)
PutPort	OK	Reference: ▶ PutPort (p.645)
BusyOut	---	---
JudgeOut	---	---
RunOut	---	---
GetAll	OK	Reference: ▶ GetAll (p.528)
PutAll	OK	Reference: ▶ PutAll (p.643)
ReceiveData	OK	Reference: ▶ ReceiveData (p.651)
SendData	OK	Reference: ▶ SendData (p.696)
SendString	OK	Reference: ▶ SendString (p.698)
ReadPlcMemory	---	---
WritePlcMemory	---	---
SetPlcData	---	---
GetPlcData	---	---

Example

Receive data

Set an IP address and parameter size (*4 integer type domain) in a parameter to use an Ethernet.

```
Dim data&(256)
Dim ipaddr&(4)
'Set the IP address of the destination.
ipaddr&(0) = 10
ipaddr&(1) = 5
ipaddr&(2) = 5
ipaddr&(3) = 101
'Gets the five data.
ReceiveData "UdpNormal", data&(), 4*5, size&, ipaddr&(), 4*4
```

Send data

Set an IP address and parameter size (*4 integer type domain) in a parameter to use an Ethernet.

```
Dim data&(256)
Dim ipaddr&(4)
'Set the IP address of the destination.
ipaddr&(0) = 10
ipaddr&(1) = 5
ipaddr&(2) = 5
ipaddr&(3) = 101
'Transmit the five data.
SendData "UdpNormal", data&(), 4*5, ipaddr&(), 4*4
```

UdpPlcLinkM

PLC Link (MELSEC QnU/Q/QnAS) communication

IoModule identification name

UdpPlcLinkM

Overview

This is a module is for sending and receiving commands and data by Ethernet PLC Link protocol.

System data

Identification	Meaning	Initial value
enableDhcp	Enable DHCP 0: Disabled 1: Enabled	0
ipAddress	IP address of the system	10.5.5.100
subnetMask	Subnet mask	255.255.255.0
defaultGateway	Gateway address	10.5.5.110
dns	DNS server	10.5.5.1
delayCount	Output IP address	0.0.0.0
portNo	Input port No.	9600

PLC Link data

Identification	Meaning	Initial value
commandArea	Command area Area	Data register
commandMemoryAddress	Command area Address	0
responseArea	Response area Area	Data register
responseMemoryAddress	Response area Address	100
outputArea	Data Output area Area	Data register
outputMemoryAddress	Data Output area Address	200
handshake	Handshaking	1
responseTimeout	Retry interval [ms]	10000
responseTimeout2	Retry interval 2 [ms]	1000

Area classification

Area classification name	Area classification number
Data register	168
File register	175
Link register	180

Supported functions

IoInitialize	OK	---
GetPort	OK	Reference: ▶ GetPort (p.542)
PutPort	OK	Reference: ▶ PutPort (p.645)
BusyOut	---	---
JudgeOut	---	---
RunOut	---	---
GetAll	OK	Reference: ▶ GetAll (p.528)
PutAll	OK	Reference: ▶ PutAll (p.643)
ReceiveData	OK	Reference: ▶ ReceiveData (p.651)
SendData	OK	Reference: ▶ SendData (p.696)
SendString	OK	Reference: ▶ SendString (p.698)
ReadPlcMemory	OK	Reference: ▶ ReadPlcMemory (p.649)
WritePlcMemory	OK	Reference: ▶ WritePlcMemory (p.807)
SetPlcData	OK	Reference: ▶ SetPlcData (p.709)
GetPlcData	OK	Reference: ▶ GetPlcData (p.538)

Example

Receive data

Set an IP address and parameter size (*4 integer type domain) in a parameter to use an Ethernet.

```
Dim data&(256)
Dim ipaddr&(4)
'Set the IP address of the destination.
ipaddr&(0) = 10
ipaddr&(1) = 5
ipaddr&(2) = 5
ipaddr&(3) = 101
'Gets the five data.
ReceiveData "UdpPlcLinkM", data&(), 4*5, size&, ipaddr&(), 4*4
```

Send data

Set an IP address and parameter size (*4 integer type domain) in a parameter to use an Ethernet.

```
Dim data&(256)
Dim ipaddr&(4)
'Set the IP address of the destination.
ipaddr&(0) = 10
ipaddr&(1) = 5
ipaddr&(2) = 5
ipaddr&(3) = 101
'Transmit the five data.
SendData "UdpPlcLinkM", data&(), 4*5, ipaddr&(), 4*4
```

Gets 7ch data from 10ch of the Data register area.

Gets the data from readData().

```
Dim readData&(256)
Dim data3$(21)

'Gets the data from PLC
ReadPlcMemory "UdpPlcLinkM", 168, 10, 7, readData&()

'Gets the data of the real number type
GetPlcData "UdpPlcLinkM", readData&(), 0, 8, data1#
'Gets the data of the integer type
GetPlcData "UdpPlcLinkM", readData&(), 8, 4, data2&
'Gets the data of the character type
GetPlcData "UdpPlcLinkM", readData&(), 12, 5, data2&
```

Set data to writeData().
Write in data for 7ch from 10ch of the DM area.

```
Dim writeData&(256)

'Set the data (123.45) of the real number type.
SetPlcData "UdpPlcLinkM", writeData&(), 0, 8, 123.45
'Set the data (20) of the integer type.
SetPlcData "UdpPlcLinkM", writeData&(), 32, 4, 20
'Set the data (OMRON) of the character type.
SetPlcData "UdpPlcLinkM", writeData&(), 36, 5, "OMRON"

'Write in data for 7ch from 10ch of the Data register area.
WritePlcMemory "UdpPlcLinkM", 168, 10, 7, writeData&()
```

UdpPlcLinkY

PLC Link (JEPMC MP) communication

IoModule identification name

UdpPlcLinkY

Overview

This is a module is for sending and receiving commands and data by Ethernet PLC Link protocol.

PLC Link data

Identification	Meaning	Initial value
enableDhcp	Enable DHCP 0: Disabled 1: Enabled	0
ipAddress	IP address	10.5.5.100
subnetMask	Subnet mask	255.255.255.0
defaultGateway	Gateway address	10.5.5.110
dns	DNS server address	10.5.5.1
delayCount	Output IP address	0.0.0.0
portNo	Input port number	9600

PLC Link data

Identification	Meaning	Initial value
commandArea	Command area Area	Data register
commandMemoryAddress	Command area Address	0
responseArea	Response area Area	Data register
responseMemoryAddress	Response area Address	100
outputArea	Data Output area Area	Data register
outputMemoryAddress	Data Output area Address	200
handshake	Handshaking	1
responseTimeout	Retry interval [ms]	10000
responseTimeout2	Retry interval 2 [ms]	1000

Area classification

Area classification name	Area classification number
Data register	176

Supported functions

IoInitialize	OK	---
GetPort	OK	Reference: ▶ GetPort (p.542)
PutPort	OK	Reference: ▶ PutPort (p.645)
BusyOut	---	---
JudgeOut	---	---
RunOut	---	---
GetAll	OK	Reference: ▶ GetAll (p.528)
PutAll	OK	Reference: ▶ PutAll (p.643)
ReceiveData	OK	Reference: ▶ ReceiveData (p.651)
SendData	OK	Reference: ▶ SendData (p.696)
SendString	OK	Reference: ▶ SendString (p.698)
ReadPlcMemory	OK	Reference: ▶ ReadPlcMemory (p.649)
WritePlcMemory	OK	Reference: ▶ WritePlcMemory (p.807)
SetPlcData	OK	Reference: ▶ SetPlcData (p.709)
GetPlcData	OK	Reference: ▶ GetPlcData (p.538)

Example

Receive data

Set an IP address and parameter size (*4 integer type domain) in a parameter to use an Ethernet.

```
Dim data&(256)
Dim ipaddr&(4)
'Set the IP address of the destination.
ipaddr&(0) = 10
ipaddr&(1) = 5
ipaddr&(2) = 5
ipaddr&(3) = 101
'Gets the five data.
ReceiveData "UdpPlcLinkY", data&(), 4*5, size&, ipaddr&(), 4*4
```

Send data

Set an IP address and parameter size (*4 integer type domain) in a parameter to use an Ethernet.

```
Dim data&(256)
Dim ipaddr&(4)
'Set the IP address of the destination.
ipaddr&(0) = 10
ipaddr&(1) = 5
ipaddr&(2) = 5
ipaddr&(3) = 101
'Transmit the five data.
SendData "UdpPlcLinkY", data&(), 4*5, ipaddr&(), 4*4
```

Gets 7ch data from 10ch of the Data register area.

Gets the data from readData().

```
Dim readData&(256)
Dim data3$(21)

'Gets the data from PLC
ReadPlcMemory "UdpPlcLinkY", 176, 10, 7, readData&()

'Gets the data of the real number type
GetPlcData "UdpPlcLinkY", readData&(), 0, 8, data1#
'Gets the data of the integer type
GetPlcData "UdpPlcLinkY", readData&(), 8, 4, data2&
'Gets the data of the character type
GetPlcData "UdpPlcLinkY", readData&(), 12, 5, data2&
```

Set data to writeData().

Write in data for 7ch from 10ch of the DM area.

```
Dim writeData&(256)

'Set the data (123.45) of the real number type.
SetPlcData "UdpPlcLinkY", writeData&(), 0, 8, 123.45
'Set the data (20) of the integer type.
SetPlcData "UdpPlcLinkY", writeData&(), 32, 4, 20
'Set the data (OMRON) of the character type.
SetPlcData "UdpPlcLinkY", writeData&(), 36, 5, "OMRON"

'Write in data for 7ch from 10ch of the Data register area.
WritePlcMemory "UdpPlcLinkY", 176, 10, 7, writeData&()
```

UdpPlcLink

PLC Link (SYSMAC CS/CJ/CP/One) communication

IoModule identification name

UdpPlcLink

Overview

This is a module is for sending and receiving commands and data by Ethernet PLC Link protocol.

System data

Identification	Meaning	Initial value
enableDhcp	Enable DHCP 0: Disabled 1: Enabled	0
ipAddress	IP address of the system	10.5.5.100
subnetMask	Subnet mask	255.255.255.0
defaultGateway	Gateway address	10.5.5.110
dns	DNS server DNS server address	10.5.5.1
delayCount	Output IP address	0.0.0.0
portNo	Input port No.	9600

PLC Link data

Identification	Meaning	Initial value
commandArea	Command area Area	CIO Area (CIO)
commandMemoryAddress	Command area Address	0
responseArea	Response area Area	CIO Area (CIO)
responseMemoryAddress	Response area Address	100
outputArea	Data Output area Area	CIO Area (CIO)
outputMemoryAddress	Data Output area Address	200
handshake	Handshaking	1
responseTimeout	Retry interval [ms]	10000
responseTimeout2	Retry interval 2 [ms]	1000

Area classification

Area classification name	Area classification number
CIO Area (CIO)	176
Work Area (WR)	177
Holding Bit Area (HR)	178
Auxiliary Bit Area (AR)	179
DM Area (DM)	130
EM Area (EM0)	160
EM Area (EM1)	161
EM Area (EM2)	162
EM Area (EM3)	163
EM Area (EM4)	164
EM Area (EM5)	165
EM Area (EM6)	166
EM Area (EM7)	167
EM Area (EM8)	168
EM Area (EM9)	169
EM Area (EMA)	170
EM Area (EMB)	171
EM Area (EMC)	172

Supported functions

IoInitialize	OK	---
GetPort	OK	Reference: ▶ GetPort (p.542)
PutPort	OK	Reference: ▶ PutPort (p.645)
BusyOut	---	---
JudgeOut	---	---
RunOut	---	---
GetAll	OK	Reference: ▶ GetAll (p.528)
PutAll	OK	Reference: ▶ PutAll (p.643)
ReceiveData	OK	Reference: ▶ ReceiveData (p.651)
SendData	OK	Reference: ▶ SendData (p.696)
SendString	OK	Reference: ▶ SendString (p.698)
ReadPlcMemory	OK	Reference: ▶ ReadPlcMemory (p.649)
WritePlcMemory	OK	Reference: ▶ WritePlcMemory (p.807)
SetPlcData	OK	Reference: ▶ SetPlcData (p.709)
GetPlcData	OK	Reference: ▶ GetPlcData (p.538)

Example

Receive data

Set an IP address and parameter size (*4 integer type domain) in a parameter to use an Ethernet.

```
Dim data&(256)
Dim ipaddr&(4)
'Set the IP address of the destination.
ipaddr&(0) = 10
ipaddr&(1) = 5
ipaddr&(2) = 5
ipaddr&(3) = 101
'Gets the five data.
ReceiveData "UdpPlcLink", data&(), 4*5, size&, ipaddr&(), 4*4
```

Send data

Set an IP address and parameter size (*4 integer type domain) in a parameter to use an Ethernet.

```
Dim data&(256)
Dim ipaddr&(4)
'Set the IP address of the destination.
ipaddr&(0) = 10
ipaddr&(1) = 5
ipaddr&(2) = 5
ipaddr&(3) = 101
'Transmit the five data.
SendData "UdpPlcLink", data&(), 4*5, ipaddr&(), 4*4
```

Gets 7ch data from 10ch of the DM area.

Gets the data from readData().

```
Dim readData&(256)
Dim data3$(21)

'Gets the data from PLC
ReadPlcMemory "UdpPlcLink", 130, 10, 7, readData&()

'Gets the data of the real number type
GetPlcData "UdpPlcLink", readData&(), 0, 8, data1#
'Gets the data of the integer type
GetPlcData "UdpPlcLink", readData&(), 8, 4, data2&
'Gets the data of the character type
GetPlcData "UdpPlcLink", readData&(), 12, 5, data2&
```

Set data to writeData().

Write in data for 7ch from 10ch of the DM area.

```
Dim writeData&(256)

'Set the data (123.45) of the real number type.
SetPlcData "UdpPlcLink", writeData&(), 0, 8, 123.45
'Set the data (20) of the integer type.
SetPlcData "UdpPlcLink", writeData&(), 32, 4, 20
'Set the data (OMRON) of the character type.
SetPlcData "UdpPlcLink", writeData&(), 36, 5, "OMRON"

'Write in data for 7ch from 10ch of the DM area.
WritePlcMemory "UdpPlcLink", 130, 10, 7, writeData&()
```

Figure Data List

To set or acquire a model figure or region figure held in a processing unit, use an array to specify the figure data to be set or acquired

Figure Data Structure List

The structure of figure data is indicated below.

Array element	Description	Description
figure(0)	Figure data header information	This is figure data header information. Includes the number of figures and figure data size information. Upper 16 bits: Number of figures Lower 16 bits: Number of bytes of figure data size (figure array length x 4) Figure data header information = Number of bytes of figure data size + Number of figure data x 65536
figure(1)	Figure 0 type information	Type information of figure 0 data. Includes drawing mode and figure type information. Upper 16 bits: Drawing mode Lower 16 bits: Figure type Figure type information = Figure type + Drawing mode x 65536
figure(2)	Figure 0 data	Figure data of figure 0. The size and content depends on the figure type.
figure(M)	Figure 1 type information	Type information of figure 1 data.
figure(M+1)	Figure 1 data	Figure data of figure 1. The size and content depends on the figure type.
...	...	...
figure(N*M)	Figure N type information	Type information of figure N data.
figure(N*M+1)	Figure N data	Figure data of figure N. The size and content depends on the figure type.

List of Figure Data Array Elements

The array elements of each figure and corresponding settings are shown below. The information shown in the table is for a figure count of 1.

Figure	Figure kind	Array element	Description
Point	1	figure(0)	Figure data header information
		figure(1)	Figure type information
		figure(2)	X-coordinate
		figure(3)	Y-coordinate
Line	2	figure(0)	Figure data header information
		figure(1)	Figure type information
		figure(2)	First point X
		figure(3)	First point Y
		figure(4)	Second point X
		figure(5)	Second point Y

Figure	Figure kind	Array element	Description
Wide line	4	figure(0)	Figure data header information
		figure(1)	Figure type information
		figure(2)	First point X
		figure(3)	First point Y
		figure(4)	Second point X
		figure(5)	Second point Y
		figure(6)	Width
Rectangle	8	figure(0)	Figure data header information
		figure(1)	Figure type information
		figure(2)	Upper left X
		figure(3)	Upper left Y
		figure(4)	Lower right X
		figure(5)	Lower right Y
Ellipse	16	figure(0)	Figure data header information
		figure(1)	Figure type information
		figure(2)	X-coordinate of center
		figure(3)	Y-coordinate of center
		figure(4)	X-direction radius
		figure(5)	Y-direction radius
Circle	32	figure(0)	Figure data header information
		figure(1)	Figure type information
		figure(2)	X-coordinate of center
		figure(3)	Y-coordinate of center
		figure(4)	Radius
Wide circle	64	figure(0)	Figure data header information
		figure(1)	Figure type information
		figure(2)	X-coordinate of center
		figure(3)	Y-coordinate of center
		figure(4)	Radius
		figure(5)	Width
Arc	128	figure(0)	Figure data header information
		figure(1)	Figure type information
		figure(2)	X-coordinate of center
		figure(3)	Y-coordinate of center
		figure(4)	Radius
		figure(5)	Start angle
		figure(5)	End angle

Figure	Figure kind	Array element	Description
Wide arc	256	figure(0)	Figure data header information
		figure(1)	Figure type information
		figure(2)	X-coordinate of center
		figure(3)	Y-coordinate of center
		figure(4)	Radius
		figure(5)	Start angle
		figure(6)	End angle
		figure(7)	Width
Polygon	512	figure(0)	Figure data header information
		figure(1)	Figure type information
		figure(2)	Number of vertexes
		figure(3)	First point X
		figure(4)	First point Y
		figure(5)	Second point X
		figure(6)	Second point Y
		. . .	. . .
		figure(19)	9th point X
		figure(20)	9th point Y
		figure(21)	10th point X
		figure(22)	10th point Y

List of Figure Numbers

To set or acquire a model figure or region figure held in a processing unit, specify the number of the figure to be set or acquired. The figure numbers and figures of each processing item are shown below.

Parameter	Figure number	Description
Search	0	Model registration figure (model registration)
	1	Measurement region figure (region setting)
Flexible Search	0	Model registration figure for model 0 (model registration)
	1	Model registration figure for model 1 (model registration)
	2	Model registration figure for model 2 (model registration)
	3	Model registration figure for model 3 (model registration)
	4	Model registration figure for model 4 (model registration)
	5	Measurement region figure (region setting)
Sensitive Search	0	Model registration figure(model registration)
	1	Measurement region figure (region setting)

Parameter	Figure number	Description
ECM Search	0	Model registration figure(model registration)
	1	Measurement region figure (region setting)
	2	Mask registration figure (model registration)
	3	Model registration figure (error model registration)
	4	Mask registration figure(error model registration)
EC Circle Search	0	Model registration figure(model registration)
	1	Measurement region figure (region setting)
Shape Search II	0	Model registration figure(model registration)
	1	Measurement region figure (region setting)
Shape Search III	0	Model registration figure(model registration)
	1	Measurement region figure (region setting)
Ec Corner	0	Measurement region figure (region setting)
Ec Cross	0	Measurement region figure (region setting)
Classification	0	Index 0, model 0 registration model figure (model registration)
	1	Index 0, model 1 registration model figure (model registration)
	2	Index 0, model 2 registration model figure (model registration)
	3	Index 0, model 3 registration model figure (model registration)
Classification	4	Index 0, model 4 registration model figure (model registration)
	5	Index 1, model 0 registration model figure (model registration)
	6	Index 1, model 1 registration model figure (model registration)
	. . .	. . .
	178	Index 35, model 3 registration model figure (model registration)
	179	Index 35, model 4 registration model figure (model registration)
	180	Measurement region figure (region setting)
	181	Index 36, model 0 registration model figure (model registration)
	182	Index 36, model 1 registration model figure (model registration)
	. . .	. . .
	999	Index 199, model 3 registration model figure (model registration)
	1000	Index 199, model 4 registration model figure (model registration)
Edge Position	0	Measurement region figure (region setting)
Edge Pitch	0	Measurement region figure (region setting)
Scan Edge Position	0	Measurement region figure (region setting)
Scan Edge Width	0	Measurement region figure (region setting)
Circular Scan Edge Position	0	Measurement region figure (region setting)
Circular Scan Edge Width	0	Measurement region figure (region setting)
Intersection	0	Measurement region figure for straight line 0 (region setting)
	2	Measurement region figure for straight line 1 (region setting)
Color Data	0	Measurement region figure (region setting)

Parameter	Figure number	Description
Gravity and Area	0	Measurement region figure (region setting)
Labeling	0	Measurement region figure (region setting)
Label Data	---	No figure
Defect	0	Measurement region figure (region setting)
Precise Defect	0	Measurement region figure (region setting)
Fine Matching	0	Model registration figure(model registration)
Character inspection	0	Measurement region figure (region setting)
Date Verification	---	No figure
Model Dictionary	0	Index 0, model 0 registration model figure (model registration)
	1	Index 0, model 1 registration model figure (model registration)
	2	Index 0, model 2 registration model figure (model registration)
	3	Index 0, model 3 registration model figure (model registration)
	4	Index 0, model 4 registration model figure (model registration)
	5	Index 1, model 0 registration model figure (model registration)
	6	Index 1, model 1 registration model figure (model registration)
	. . .	. . .
	178	Index 35, model 3 registration model figure (model registration)
	179	Index 35, model 4 registration model figure (model registration)
	180	Measurement region figure (region setting)
2DCode	0	Measurement region figure (region setting)
Barcode	0	Measurement region figure (region setting)
Circle Angle	0	Measurement region figure (region setting)
Glue Bead Inspection	---	No figure
Shape Search +	0	Model registration figure(model registration)
	1	Measurement region figure (region setting)
Advanced labeling +	0	Measurement region figure (region setting)
Barcode +	0	Measurement region figure (region setting)
2DCode +	0	Measurement region figure (region setting)
Position Compensation	0	Measurement region figure (region setting)
Filtering	0	Measurement region figure (region setting)
Background Suppression	0	Measurement region figure (region setting)
Brightness Correct Filter	0	Measurement region figure (region setting)
Color Gray Filter	0	Measurement region figure (region setting)
Extract Color Filter	0	Measurement region figure (region setting)
Anti Color Shading	0	Measurement region figure (region setting)
Stripes Removal Filter II	0	Measurement region figure (region setting)
Polar Transformation	0	Measurement region figure (region setting)
Trapezoidal Correction	0	Measurement region figure (region setting)
Machine Simulator	---	No figure
Image Subtraction	0	Model registration figure(model registration)

Parameter	Figure number	Description
Advanced filter	0	Measurement region figure (region setting)
Panorama	---	No figure
Trapezoidal Correction +	0	Measurement region figure (region setting)
Stripes Removal Filter +	0	Measurement region figure (region setting)
Halation suppression +	0	Measurement region figure (region setting)

Model Number List

When you need to re-register the processing unit model, specify the number of the model to be re-registered. The model numbers and models of each processing item are shown below.

Parameter	Modle number	Description
Search	0	Search model
Flexible Search	0	Index 0, model 0 search model
	1	Index 0, model 1 search model
	2	Index 0, model 2 search model
	3	Index 0, model 3 search model
	4	Index 0, model 4 search model
Sensitive Search	0	Search model
ECM Search	0	Search model
EC Circle Search	0	Search model
Shape Search II	0	Search model
Shape Search III	0	Search model
Ec Corner	0	Reference position
Ec Cross	0	Reference position
Classification	0	Index 0, model 0 search model
	1	Index 0, model 1 search model
	2	Index 0, model 2 search model
	3	Index 0, model 3 search model
	4	Index 0, model 4 search model
	5	Index 1, model 0 search model
	6	Index 1, model 1 search model
	. . .	. . .
	178	Index 35, model 3 search model
	179	Index 35, model 4 search model
	180	Index 36, model 0 search model
	181	Index 36, model 1 search model
	. . .	. . .
	998	Index 199, model 3 search model
	999	Index 199, model 4 search model

Parameter	Modle number	Description
Edge Position	0	Edge model
Edge Pitch	0	Edge model
Scan Edge Position	0	Edge model
Scan Edge Width	0	Edge model
Circular Scan Edge Position	0	Edge model
Circular Scan Edge Width	0	Edge model
Intersection	0	Edge model
Color Data	0	Color Data model
Gravity and Area	0	Gravity and Area model
Labeling	0	Labeling model
Label Data	---	No model
Defect	0	Defect model
Precise Defect	0	Precise Defect model
Fine Matching	0	Fine Matching model
Character inspection	---	No model
Date Verification	---	No model
Model Dictionary	0	Index 0, model 0 search model
	1	Index 0, model 1 search model
	2	Index 0, model 2 search model
	3	Index 0, model 3 search model
	4	Index 0, model 4 search model
	5	Index 1, model 0 search model
	6	Index 1, model 1 search model
	. . .	. . .
	178	Index 35, model 3 search model
	179	Index 35, model 4 search model
2DCode	---	No model
Barcode	---	No model
Circle Angle	---	No model
Glue Bead Inspection	---	No model
Shape Search +	0	Search model
Advanced labeling +	0	Reference position
Barcode +	---	No model
2DCode +	---	No model

Image Number List

When you need to access image data in a processing unit, specify the number of the image that you want to access. The image numbers and images of each processing item are shown below.

Item	Image number	Description
Search	-	No image
Flexible Search	-	No image
Sensitive Search	-	No image
ECM Search	-	No image
EC Circle Search	-	No image
Shape Search II	-	No image
Shape Search III	-	No image
Ec Corner	-	No image
Ec Cross	-	No image
Classification	-	No image
Edge Position	-	No image
Edge Pitch	-	No image
Scan Edge Position	-	No image
Scan Edge Width	-	No image
Circular Scan Edge Position	-	No image
Circular Scan Edge Width	-	No image
Intersection	-	No image
Color Data	-	No image
Gravity and Area	-	No image
Labeling	-	No image
Label Data	-	No image
Defect	-	No image
Precise Defect	-	No image
Fine Matching	-	No image
Character Inspection	-	No image
Date Verification	-	No image
Model Dictionary	-	No image
2DCode	-	No image
Barcode	-	No image
Character Recognition	-	No image
Character Dictionary	-	No image
Circle Angle	-	No image
Glue Bead Inspection	-	No image
Shape Search +	-	No image
Advanced labeling +	-	No image
Barcode +	-	No image
2DCode +	-	No image
Camera Image Input	0	Camera image

Item	Image number	Description
Camera Image Input FH	0	Camera image
Camera Image Input HDR	0	Camera image
Camera Image Input HDR Lite	0	Camera image
Camera Switching	0	Camera image
Measurement Image Switching	-	No image
Position Compensation	0	Position compensated image
Filtering	0	Filtered image
Background Suppression	0	Background suppressed image
Brightness Correct Filter	0	Brightness corrected image
Color Gray Filter	0	Color gray image
Extract Color Filter	0	Color extracted image
Anti Color Shading	0	Anti color shading image
Stripes Removal Filter II	0	Stripes removed image
Polar Transformation	0	Polar transformed image
Trapezoidal Correction	0	Trapezoidal corrected image
Machine Simulator	0	Axis shifted image
Image Subtraction	0	Subtraction image
Advanced filter	0	Output image 0
	1	Output image 1
	2	Output image 2
	3	Output image 3
Panorama	0	Panorama image
Trapezoidal Correction +	0	Trapezoidal corrected image
Stripes Removal Filter +	0	Stripes removed image
Halation Suppression +	0	Halation suppressed image
Unit Macro	0	Processing unit image 0
	1	Processing unit image 1
	...	...
	30	Processing unit image 30
	31	Processing unit image 31
Unit Calculation Macro	-	No image
Calculation	-	No image
Line Regression	-	No image
Circle Regression	-	No image
Precise Calibration	0	Corrected image
User Data	-	No image
Set Unit Data	-	No image
Get Unit Data	-	No image
Set Unit Figure	-	No image
Get Unit Figure	-	No image
Trend Monitor	-	No image
Image Logging	-	No image

Item	Image number	Description
Image Conversion Logging	-	No image
Data Logging	-	No image
Elapsed Time	-	No image
Wait	-	No image
Focus	-	No image
Iris	-	No image
Parallelize	-	No image
Parallelize Task	-	No image
Statistics		No image
Reference Calib Data	0	Corrected image
Position Data Calculation	-	No image
Stage Data	-	No image
Robot Data	-	No image
Vision Master Calibration	-	No image
PLC Master Calibration	-	No image
Convert Position Data	-	No image
Movement Single Position	-	No image
Movement Multi Points	-	No image
Detection Point	-	No image
Camera Calibration	-	No image
Data Save	-	No image
Conditional Branch	-	No image
End	-	No image
DI Branch	-	No image
Control Flow Normal	-	No image
Control Flow PLC Link	-	No image
Control Flow Parallel	-	No image
Control Flow Fieldbus	-	No image
Selective Branch	-	No image
Data Output	-	No image
Parallel Data Output	-	No image
Parallel Judgement Output	-	No image
Fieldbus Data Output	-	No image
Result Display	-	No image
Display Image File	0	File display image 0
	1	File display image 1
	2	File display image 2
	3	File display image 3
Display Last NG Image	0	NG image 0
	1	NG image 1
	2	NG image 2
	3	NG image 3

List of Sub-Image Numbers

In addition to measurement images, the processing unit also contains images being processed and images that have been processed, and these can be displayed in the image window as sub-images. To display a sub-image, specify the number of the sub-image. The sub-image numbers and sub-images of each processing item are shown below.

Item	Sub-image	Description
Search	0	Measurement image
Flexible Search	0	Measurement image
Sensitive Search	0	Measurement image
ECM Search	0	Measurement image
	1	Image with the edge that matches the measurement image overlaid
EC Circle Search	0	Measurement image
Shape Search II	0	Measurement image
Shape Search III	0	Measurement image
	1	Display of measurement object and corresponding model.
	2	Edge image
	3	Display of edge image and corresponding model
Ec Corner	0	Measurement image
Ec Cross	0	Measurement image
Classification	0	Measurement image
Edge Position	0	Measurement image
	1	Profile image
Edge Pitch	0	Measurement image
	1	Profile image
Scan Edge Position	0	Measurement image
	1	Scan image
Scan Edge Width	0	Measurement image
	1	Scan image
Circular Scan Edge Position	0	Measurement image
	1	Edge position display image of each divided region
Circular Scan Edge Width	0	Measurement image
	1	Edge position image of each region division
Intersection	0	Measurement image
	1	Scan region image
Color Data	0	Measurement image
	1	Masked image
Gravity and Area	0	Measurement image
	1	Color extraction image (when measuring a color image) Binary image (when measuring a monochrome image)
Labeling	0	Measurement image
	1	Color extraction image (when measuring a color image) Binary image (when measuring a monochrome image)
Label Data	0	Measurement image

Item	Sub-image	Description
Defect	0	Measurement image
	1	Masked defect profile image (with area measurement)
Precise Defect	0	Measurement image
	1	Masked defect profile image (with area measurement)
Fine Matching	0	Measurement image
	1	Difference image
Character Inspection	0	Measurement image
Date Verification	0	Measurement image
Model Dictionary	0	Measurement image
2DCode	0	Measurement image
Barcode	0	Measurement image
OCR	0	Measurement image
OCR User Dictionary	0	Measurement image
Circle Angle	0	Measurement image
Glue Bead Inspection	0	Measurement image
	1	All-color extraction image / Selection color image / Binary image
	3	Binary image (route displayed when model is registered)
	4	Binary image
Shape Search +	0	Measurement image
Advanced labeling +	0	Measurement image
	1	Color extraction image (when measuring a color image) Binary image (when measuring a monochrome image)
Barcode +	0	Measurement image
2DCode +	0	Measurement image
Camera Image Input	0	Camera image
Camera Image Input FH	0	Camera image
Camera Image Input HDR	0	Camera image
Camera Image Input HDR Lite	0	Camera image
Camera Switching	0	Camera image
Measurement Image Switching	0	Image after switching
	1	Measurement image
Position Compensation	0	After scroll
	1	Before scroll
Filtering	0	Filtered image
Background Suppression	0	Background suppressed image
Brightness Correct Filter	0	Brightness corrected image
Color Gray Filter	0	Color gray image
Extract Color Filter	0	Color extracted image
	1	Measurement image
Anti Color Shading	0	Anti color shading image
Stripes Removal Filter II	0	Color extracted image

Item	Sub-image	Description
Polar Transformation	0	Circle image
	1	Measurement image
Trapezoidal Correction	0	Trapezoidal corrected image
Machine Simulator	0	Axis shifted image
Image Subtraction	0	Subtraction image
	1	Measurement image
Advanced filter	0	Advanced filter image
	1	Output image 0
	2	Output image 1
	3	Output image 2
	4	Output image 3
	5	Output image
Panorama	0	Panorama image
Trapezoidal Correction +	0	Trapezoidal correction image
	1	Measurement image
Stripes Removal Filter +	0	Stripes removed image
Halation Suppression +	0	Halation suppressed image
Unit Macro	0 to 100	Can defined as desired
Unit Calculation Macro	0	Measurement image
Calculation	0	Measurement image
Line Regression	0	Measurement image
Circle Regression	0	Measurement image
Precise Calibration	0	High-precision calibration image
User Data	0	Measurement image
Set Unit Data	0	Measurement image
Get Unit Data	0	Measurement image
Set Unit Figure	0	Measurement image
Get Unit Figure	0	Measurement image
Trend Monitor	0	Trend graph display
	1	Histogram display
Image Logging	0	Measurement image
Image Conversion Logging	0	Measurement image
Data Logging	0	Measurement image
Elapsed Time	0	Measurement image
Wait	0	Measurement image
Focus	0	Measurement image
Iris	0	Measurement image
Parallelize	0	Measurement image
Parallelize Task	0	Measurement image

Item	Sub-image	Description
Statistics	0	Graph display corresponding to "Data number" in setting data
	1	Data graph display of data 0
	2	Data graph display of data 1
	3	Data graph display of data 2
	4	Data graph display of data 3
	5	Data graph display of data 4
	6	Data graph display of data 5
	7	Data graph display of data 6
	8	Data graph display of data 7
Reference Calib Data	0	Calibration reference image
Position Data Calculation	0	Measurement image
Stage Data	0	Measurement image
Robot Data	0	Measurement image
Vision Master Calibration	0	Measurement image + Calibration progress display
	1	Measurement image
PLC Master Calibration	0	Measurement image + Calibration progress display
	1	Measurement image
Convert Position Data	0	Measurement image
Movement Single Position	0	Measurement image
Movement Multi Points	0	Measurement image
Detection Point	0	Measurement image
Camera Calibration	0	Measurement image
Data Save	0	Measurement image
Conditional Branch	0	Measurement image
End	0	Measurement image
DI Branch	0	Measurement image
Control Flow Normal	0	Measurement image
Control Flow PLC Link	0	Measurement image
Control Flow Parallel	0	Measurement image
Control Flow Fieldbus	0	Measurement image
Selective Branch	0	Measurement image
Data Output	0	Measurement image
Parallel Data Output	0	Measurement image
Parallel Judgement Output	0	Measurement image
Fieldbus Data Output	0	Measurement image
Result image	0	Result image
Display Image File	0	Image 0
	1	Image 1
	2	Image 2
	3	Image 3

Item	Sub-image	Description
Display Last NG Image	0	Last NG
	1	1st preceding NG image (when two or more images are saved) Most recent NG image (when less than two images are saved)
	2	2nd preceding NG image (when three or more images are saved) Most recent NG image (when less than three images are saved)
	3	3rd preceding NG image (when four or more images are saved) Most recent NG image (when less than four images are saved)

Memory Usage Guidance For Processing Items

Free space is required in the application memory in order to add a processing unit to the measurement flow. The amount of free space required in application memory consists of both the amount necessary to add a processing unit to the measurement flow, and the amount necessary to configure settings for the added unit and execute measurement.
When adding a new processing unit to the measurement flow, or if the application memory becomes insufficient when settings are configured, refer to the table below for the approximate amount of application memory used and delete a processing unit from the measurement flow or adjust the processing unit settings.
You can check the amount of free application memory in "Available application memory:". (Reference: ▶Checking System Information: [System Information] (p.300))
To adjust each processing unit, refer to the section for each processing item in the Processing Item Function Reference Manual. (Reference▶: Processing Item Function Reference Manual (Cat. No. Z341))

Item	Approximate amount of application memory used		Main settings that affect change of memory used
	Adding to measurement flow	Change due to settings	
Search	Approx. 5 KB	Approx. 544 MB (for monochrome image) Approx. 1609 MB (for color image)	Model registration figure, model parameters, saved settings of model registration image, image size, whether image is color or monochrome.
Flexible Search	Approx. 2KB	Approx. 2,655MB (for monochrome image) Approx. 7,856MB (for color image)	Model registration count, model registration figure, model parameters, image size, whether image is color or monochrome.
Sensitive Search	Approx. 8KB	Approx. 1,088MB (for monochrome image) Approx. 3,218MB (for color image)	Model registration figure, model parameters, image size, whether image is color or monochrome.
ECM Search	Approx. 2KB	Approx. 2MB	Model registration figure, error model registration figure, extraction amount of edge extraction, image size.
EC Circle Search	Approx. 24KB	Approx. 1MB	-
Shape Search II	Approx. 6KB	Approx. 57MB	Model registration figure, model parameters, saved settings of model registration image, image size, shape of edge in image.
Shape Search III	Approx. 6KB	Approx. 335MB	Model registration figure, model parameters, saved settings of model registration image, image size, shape of edge in image.
Ec Corner	Approx. 62KB	No change	-
Ec Cross	Approx. 34KB	No change	-
Classification	Approx. 41KB	Approx. 543,477MB (for monochrome image) Approx. 1,608,757MB (for color image)	Model registration count, model registration figure, model parameters, image size, whether image is color or monochrome.
Edge Position	Approx. 1KB	Approx. 1MB	Measurement region figure, image size.
Edge Pitch	Approx. 5 KB	Approx. 1MB	Measurement region figure, image size.
Scan Edge Position	Approx. 2KB	Approx. 12MB	Measurement region figure, number of measurement points, image size.
Scan Edge Width	Approx. 2KB	Approx. 23MB	Measurement region figure, number of measurement points, image size.
Circular Scan Edge Position	Approx. 2KB	Approx. 42MB	Measurement region figure, number of measurement points, image size.
Circular Scan Edge Width	Approx. 2KB	Approx. 83MB	Measurement region figure, number of measurement points, image size.
Intersection	Approx. 4KB	Approx. 23MB	Measurement region figure, image size.

Item	Approximate amount of application memory used		Main settings that affect change of memory used
	Adding to measurement flow	Change due to settings	
Color Data	Approx. 1KB	Approx. 25MB	Measurement region figure, selected region figure of static mask, image size.
Gravity and Area	Approx. 1KB	Approx. 25MB	Measurement region figure, selected region figure of static mask, image size.
Labeling	Approx. 2,000KB	Approx. 25MB	Measurement region figure, selected region figure of static mask, image size.
Label Data	Approx. 1KB	No change	-
Defect	Approx. 1KB	Approx. 51MB	Measurement region figure, selected region figure of static mask, image size.
Precise Defect	Approx. 20KB	Approx. 29MB	Measurement region figure, detection parameter settings, selected region figure of static mask, image size.
Fine Matching	Approx. 1KB	Approx. 38MB (for monochrome image) Approx. 63MB (for color image)	Model registration figure, image size, whether image is color or monochrome.
Character inspection	Approx. 4KB	No change	-
Date Verification	Approx. 1KB	Approx. 1MB	Length of character string, code parameter settings.
Model Dictionary	Approx. 4KB	Approx. 95,567MB (for monochrome image) Approx. 282,801MB (for color image)	Model registration count, model registration figure, model parameters, image size, whether image is color or monochrome.
2DCode	Approx. 12KB	Approx. 1MB	Length of comparison character string, number and length of sorting comparison character strings.
Barcode	Approx. 3KB	Approx. 1MB	Length of comparison character string, number and length of sorting comparison character strings.
OCR	Approx. 704KB	Approx. 1MB	Number and length of character string formats, number and length of comparison strings, number and length of teaching correct response values, code parameter settings.
OCR User Dictionary	Approx. 1KB	Approx. 1MB	-
Circle Angle	Approx. 1KB	Approx. 1MB	Graduated angle, whether image is color or monochrome.
Glue Bead Inspection	Approx. 1KB	Approx. 25MB	Measurement region figure, mask region figure, image size.
Shape Search +	Approx. 10KB	No change	-
Advanced labeling +	Approx. 472KB	No change	-
Barcode +	Approx. 6KB	No change	-
2DCode +	Approx. 3KB	No change	-
Camera Image Input	Approx. 1KB	Approx. 9MB (for monochrome image) Approx. 17MB (for color image)	Model registration figure of calibration settings, image size, whether image is color or monochrome.
Camera Image Input FH	Approx. 1KB	Approx. 9MB (for monochrome image) Approx. 17MB (for color image)	Model registration figure of calibration settings, image size, whether image is color or monochrome.

Item	Approximate amount of application memory used		Main settings that affect change of memory used
	Adding to measurement flow	Change due to settings	
Camera Image Input HDR	Approx. 3KB	Approx. 9MB (for monochrome image) Approx. 17MB (for color image)	Model registration figure of calibration settings, image size, whether image is color or monochrome.
Camera Image Input HDR Lite	Approx. 1KB	Approx. 9MB (for monochrome image) Approx. 17MB (for color image)	Model registration figure of calibration settings, image size, whether image is color or monochrome.
Camera Switching	Approx. 1KB	No change	-
Measurement Image Switching	Approx. 1KB	No change	-
Position Compensation	Approx. 1KB	Approx. 1MB	Number and length of calculation expressions
Filtering	Approx. 1KB	No change	-
Background Suppression	Approx. 1KB	No change	-
Brightness Correct Filter	Approx. 1KB	No change	-
Color Gray Filter	Approx. 1KB	No change	-
Extract Color Filter	Approx. 1KB	No change	-
Anti Color Shading	Approx. 1KB	No change	-
Stripes Removal Filter II	Approx. 1KB	No change	-
Polar Transformation	Approx. 1KB	Approx. 1MB	Circle radius, circle width, end point overlay, image size.
Trapezoidal Correction	Approx. 1KB	Approx. 1MB	Number and length of calculation expressions
Machine Simulator	Approx. 1KB	No change	-
Image Subtraction	Approx. 1KB	Approx. 63MB	Model registration figure, image size.
Advanced filter	Approx. 9KB	No change	-
Panorama	Approx. 240KB	Approx. 13MB	Whether brightness correction reference image is used, image size.
Trapezoidal Correction +	Approx. 1KB	Approx. 1MB	Number and length of calculation expressions
Stripes Removal Filter +	Approx. 1KB	No change	-
Halation Suppression +	Approx. 1KB	No change	-
Unit Macro	Approx. 1KB	Approx. 4MB	Length of program code
Unit Calculation Macro	Approx. 1KB	Approx. 2MB	Length of program code
Calculation	Approx. 1KB	Approx. 1MB	Number and length of calculation expressions, number and length of comments.
Line Regression	Approx. 3KB	Approx. 1MB	Number of points, number and length of calculation expressions
Circle Regression	Approx. 2KB	Approx. 1MB	Number and length of calculation expressions

Item	Approximate amount of application memory used		Main settings that affect change of memory used
	Adding to measurement flow	Change due to settings	
Precise Calibration	Approx. 660KB	Approx. 51MB	Measurement region figure, image size, specified calibration method.
User Data	Approx. 6KB	Approx. 1MB	Number and length of calculation expressions
Set Unit Data	Approx. 1KB	Approx. 1MB	Length of calculation expression.
Get Unit Data	Approx. 1KB	No change	-
Set Unit Figure	Approx. 1KB	Approx. 1MB	Number and length of calculation expressions, number of figures, figure type.
Get Unit Figure	Approx. 1KB	No change	-
Trend Monitor	Approx. 20KB	Approx. 3MB	Length of calculation expression, number of items saved.
Image Logging	Approx. 1KB	Approx. 1MB	Length of calculation expression, length of sub-folder name, length of prefix
Image Conversion Logging	Approx. 200KB	Approx. 1MB	Length of calculation expression, length of folder name, length of prefix
Data Logging	Approx. 1KB	Approx. 1MB	Number and length of calculation expressions, number and length of comments, length of file name, length of sub-folder name, length of prefix.
Elapsed Time	Approx. 1KB	No change	-
Wait	Approx. 1KB	No change	-
Focus	Approx. 1KB	Approx. 13MB	Figure in region settings, image size.
Iris	Approx. 1KB	Approx. 13MB	Figure in region settings, image size.
Parallelize	Approx. 1KB	No change	-
Parallelize Task	Approx. 1KB	No change	-
Statistics	Approx. 8KB	Approx. 1MB	Number and length of calculation expressions, number and length of comments.
Reference Calib Data	Approx. 1KB	No change	-
Position Data Calculation	Approx. 2KB	Approx. 1MB	Number and length of calculation expressions
Stage Data	Approx. 1KB	No change	-
Robot Data	Approx. 1KB	No change	-
Vision Master Calibration	Approx. 82KB	Approx. 402MB	Image size.
PLC Master Calibration	Approx. 81KB	Approx. 402MB	Image size.
Convert Position Data	Approx. 2KB	Approx. 1MB	Number and length of calculation expressions
Movement Single Position	Approx. 4KB	Approx. 1MB	Number and length of calculation expressions
Movement Multi Points	Approx. 5 KB	Approx. 1MB	Number and length of calculation expressions
Detection Point	Approx. 1KB	Approx. 1MB	Number and length of calculation expressions

Item	Approximate amount of application memory used		Main settings that affect change of memory used
	Adding to measurement flow	Change due to settings	
Camera Calibration	Approx. 7KB	No change	-
Data Save	Approx. 1KB	Approx. 1MB	Number and length of calculation expressions, number and length of comments.
Conditional Branch	Approx. 1KB	Approx. 1MB	Number and length of calculation expressions
End	Approx. 1KB	No change	-
DI Branch	Approx. 1KB	No change	-
Control Flow Normal	Approx. 1KB	No change	-
Control Flow PLC Link	Approx. 1KB	No change	-
Control Flow Parallel	Approx. 1KB	No change	-
Control Flow Fieldbus	Approx. 1KB	No change	-
Selective Branch	Approx. 3KB	Approx. 1MB	Length of conditional expression, number and length of branch setting comments.
Data Output	Approx. 14KB	Approx. 1MB	Communication module settings, number and length of calculation expressions, number and length of comments.
Parallel Data Output	Approx. 1KB	Approx. 1MB	Communication module settings, number and length of calculation expressions, number and length of comments.
Parallel Judgement Output	Approx. 1KB	Approx. 1MB	Number and length of calculation expressions, number and length of comments.
Fieldbus Data Output	Approx. 6KB	Approx. 1MB	Communication module settings, number and length of calculation expressions, number and length of comments.
Result Display	Approx. 1KB	Approx. 1MB	Result display type, number of results displayed, number and length of calculation expressions, number and length of characters.
Display Image File	Approx. 1KB	Approx. 1MB	Number and length of file names.
Display Last NG Image	Approx. 3KB	Approx. 1MB	Number and length of calculation expressions, number and length of comments.

Macro Function List

Macro functions that can be used in macro customize functions are shown below.

Alphabetical Order

Command Name	Function	Classification	Unit Calculation Macro	Scene Control Macro	Communication Command Macro	Unit Macro	References
Abs	Gets the absolute value of the specified expression	Arithmetic Calculation	OK	OK	OK	OK	Reference: ▶ Details (p.403)
AddGlobalData	Adds the global data	Control Global Data	OK	OK	OK	OK	Reference: ▶ Details (p.405)
AddSystemData	Adds the system data	System Data	OK	OK	OK	OK	Reference: ▶ Details (p.407)
AND	Gets the logical product of two expressions	Arithmetic Calculation	OK	OK	OK	OK	Reference: ▶ Details (p.409)
ApproximationCircle	Gets the approximate circle	Arithmetic Calculation	OK	OK	OK	OK	Reference: ▶ Details (p.411)
Asc	Gets the character code of the specified character	String Operation	OK	OK	OK	OK	Reference: ▶ Details (p.413)
AssignUnit	Registers the processing unit	Flow control	---	OK	OK	---	Reference: ▶ Details (p.415)
Atn	Gets the arctangent of the specified expression	Arithmetic Calculation	OK	OK	OK	OK	Reference: ▶ Details (p.417)
BusyOut	Sets the output state of the processing busy signal	IO Module Control	---	OK	OK	OK	Reference: ▶ Details (p.419)
Call	Executes the registered user-defined function	Debug Command	OK	OK	OK	OK	Reference: ▶ Details (p.421)
ChangeScene	Change the scene	Scene control	---	OK	OK	---	Reference: ▶ Details (p.423)
ChangeSceneGroup	Changes the scene group	Scene group control	---	OK	OK	---	Reference: ▶ Details (p.424)
CheckUnit	Checks the registration status of a processing unit	Flow control	---	OK	OK	---	Reference: ▶ Details (p.425)
Chr$	Determining the character of the specified character code	String Operation	OK	OK	OK	OK	Reference: ▶ Details (p.427)
ClearMeasureData	Clears the measurement results of the processing unit	Measurement control	---	OK	OK	OK	Reference: ▶ Details (p.429)
ClearScene	Clears the scene data	Scene control	---	OK	OK	---	Reference: ▶ Details (p.430)
ClearSceneGroup	Clears scene group data	Scene group control	---	OK	OK	---	Reference: ▶ Details (p.431)
Close	Closes up the file	File Control	OK	OK	OK	OK	Reference: ▶ Details (p.432)
CloseTextData	Close up a messages file	Multilingual Support Message Functions	OK	---	---	OK	Reference: ▶ Details (p.434)

Command Name	Function	Classification	Unit Calculation Macro	Scene Control Macro	Communication Command Macro	Unit Macro	References
Cont	Resumes execution of the program after it has been stopped	Debug Command	OK	OK	OK	OK	Reference: ▶ Details (p.436)
CopyMeasureImage	Copies the measurement image as an image of the Unit Macro processing unit	Processing unit control	---	---	---	OK	Reference: ▶ Details (p.438)
CopyScene	Copies scene data	Scene control	---	OK	OK	---	Reference: ▶ Details (p.440)
CopySceneGroup	Copies scene group data	Scene group control	---	OK	OK	---	Reference: ▶ Details (p.441)
CopyUnit	Copies a processing unit	Flow control	---	OK	OK	---	Reference: ▶ Details (p.442)
CopyUnitFigure	Copies figure data to the processing unit	Processing unit control	---	OK	OK	OK	Reference: ▶ Details (p.444)
CopyUnitImage	Copies a processing unit image as a unit macro processing unit image	Processing unit control	---	---	---	OK	Reference: ▶ Details (p.446)
CopyUnitModel	Copies the model data of a processing unit	Processing unit control	---	OK	OK	OK	Reference: ▶ Details (p.448)
Cos	Gets the cosine of the specified expression	Arithmetic Calculation	OK	OK	OK	OK	Reference: ▶ Details (p.450)
Crspoint	Gets the intersection between 2 straight lines	Arithmetic Calculation	OK	OK	OK	OK	Reference: ▶ Details (p.452)
Date$	Reads out the date from the internal clock	Arithmetic Calculation	OK	OK	OK	OK	Reference: ▶ Details (p.454)
Debug	Set the program execution form and information output method	Debug Command	OK	OK	OK	OK	Reference: ▶ Details (p.456)
DebugPrint	Outputs debug information to the system status console window	Debug Command	OK	OK	OK	OK	Reference: ▶ Details (p.458)
DeleteUnit	Deletes a processing unit	Flow control	---	OK	OK	---	Reference: ▶ Details (p.460)
Dim	Defines the array variable	General instruction	OK	OK	OK	OK	Reference: ▶ Details (p.461)
DisplaySubNo	Get the sub-image number of the displayed sub-image	Image Window Control	---	---	---	OK	Reference: ▶ Details (p.462)
DisplayUnitNo	Gets the selection state of the processing unit number of the flow window	Display control	---	OK	OK	---	Reference: ▶ Details (p.464)
Do Loop While	Repeatedly executes the statements between Do and Loop while the specified condition meets	General instruction	OK	OK	OK	OK	Reference: ▶ Details (p.465)
Dposline	Gets the shortest distance between the line and point	Arithmetic Calculation	OK	OK	OK	OK	Reference: ▶ Details (p.467)
DrawArc	Draw the arc on the image window	Image Window Control	---	---	---	OK	Reference: ▶ Details (p.469)

Command Name	Function	Classification	Unit Calculation Macro	Scene Control Macro	Communication Command Macro	Unit Macro	References
DrawArcW	Draw the wide arc on the image window	Image Window Control	---	---	---	OK	Reference: ▶ Details (p.471)
DrawBox	Draws a rectangle on the image window	Image Window Control	---	---	---	OK	Reference: ▶ Details (p.473)
DrawCircle	Draw a circle on the image window	Image Window Control	---	---	---	OK	Reference: ▶ Details (p.475)
DrawCircleW	Draw the wide circle on the image window	Image Window Control	---	---	---	OK	Reference: ▶ Details (p.477)
DrawCursor	Draw the cross-hair cursor on the image window	Image Window Control	---	---	---	OK	Reference: ▶ Details (p.479)
DrawEllipse	Draw the ellipse on the image window	Image Window Control	---	---	---	OK	Reference: ▶ Details (p.481)
DrawFigure	Draw a figure on the image window	Image Window Control	---	---	---	OK	Reference: ▶ Details (p.483)
DrawFillImage	Draw the fill image on the image window	Image Window Control	---	---	---	OK	Reference: ▶ Details (p.485)
DrawJudgeText	Draws the judgement result of the character string on the text display screen	Text Window Control	---	---	---	OK	Reference: ▶ Details (p.486)
DrawLine	Draw a straight line on the image window	Image Window Control	---	---	---	OK	Reference: ▶ Details (p.488)
DrawLineW	Draw the wide straight line on the image window	Image Window Control	---	---	---	OK	Reference: ▶ Details (p.490)
DrawMeasureImage	Draw the measurement image on the image window	Image Window Control	---	---	---	OK	Reference: ▶ Details (p.492)
DrawPoint	Draw a point on the image window	Image Window Control	---	---	---	OK	Reference: ▶ Details (p.493)
DrawPolygon	Draw a polygon on the image window	Image Window Control	---	---	---	OK	Reference: ▶ Details (p.495)
DrawSearchFigure	Draw the search figure on the image window	Image Window Control	---	---	---	OK	Reference: ▶ Details (p.497)
DrawText	Draw a character string on the text window	Text Window Control	---	---	---	OK	Reference: ▶ Details (p.500)
DrawTextG	Draw a character string on the image window	Image Window Control	---	---	---	OK	Reference: ▶ Details (p.502)
DrawUnitImage	Display the "other unit image" on the image window	Image Window Control	---	---	---	OK	Reference: ▶ Details (p.504)
Dskf	Gets the free space on disk drives	File Control	OK	OK	OK	OK	Reference: ▶ Details (p.506)
ElapsedTime	Gets the elapsed time since starting the measurement	Others	OK	---	---	OK	Reference: ▶ Details (p.508)
Eof	Examines the end of the file	File Control	OK	OK	OK	OK	Reference: ▶ Details (p.510)
Erase	Releases array variable	General instruction	OK	OK	OK	OK	Reference: ▶ Details (p.512)

Command Name	Function	Classification	Unit Calculation Macro	Scene Control Macro	Communication Command Macro	Unit Macro	References
Errcmnd$	Get the function name of the macro where an error occurred	General instruction	OK	OK	OK	OK	Reference: ▶ Details (p.513)
Errno	Gets the error number	General instruction	OK	OK	OK	OK	Reference: ▶ Details (p.515)
ExecuteImageLogging	Executes image logging	Others	---	---	---	OK	Reference: ▶ Details (p.517)
ExitFzProcess	Terminate the Sensor Controller	Others	---	OK	OK	---	Reference: ▶ Details (p.519)
Exp	Gets the value of the exponential function of the base e natural logarithm	Arithmetic Calculation	OK	OK	OK	OK	Reference: ▶ Details (p.520)
Fcopy	Copies the file	File Control	OK	OK	OK	OK	Reference: ▶ Details (p.522)
Fix	Gets the integer of a value by rounding off digits to the right of the decimal point	Arithmetic Calculation	OK	OK	OK	OK	Reference: ▶ Details (p.524)
For To Step Next	Repeats the statements between the For and Next statements	General instruction	OK	OK	OK	OK	Reference: ▶ Details (p.526)
GetAll	Gets the input states of all input terminals	IO Module Control	---	OK	OK	OK	Reference: ▶ Details (p.528)
GetGlobalData	Gets the global data	Control Global Data	OK	OK	OK	OK	Reference: ▶ Details (p.530)
GetImageSize	Gets the image size of the processing unit image	Processing unit control	OK	OK	OK	OK	Reference: ▶ Details (p.532)
GetImageWindow	Get the state of the image window	Display control	---	OK	OK	OK	Reference: ▶ Details (p.533)
GetMeasureOut	Gets the external output setting for measurement results	Measurement control	---	OK	OK	---	Reference: ▶ Details (p.536)
GetPlcData	Gets data read with the ReadPlcMemory function	IO Module Control	---	OK	OK	OK	Reference: ▶ Details (p.538)
GetPollingState	Gets the polling state of the communication module	IO Module Control	---	OK	OK	OK	Reference: ▶ Details (p.540)
GetPort	Gets the input state of the specified input terminal	IO Module Control	---	OK	OK	OK	Reference: ▶ Details (p.542)
GetSceneData	Gets data related to the scene control macro	Scene control	OK	OK	OK	OK	Reference: ▶ Details (p.544)
GetSystemData	Gets the system data	System Data	OK	OK	OK	OK	Reference: ▶ Details (p.546)
GetText$	Get a text data from a messages file	Multilingual Support Message Functions	OK	---	---	OK	Reference: ▶ Details (p.548)
GetTextWindow	Gets the state of the text window	Display control	---	OK	OK	OK	Reference: ▶ Details (p.550)
GetUnitData	Gets the data of a processing unit	Processing unit control	OK	OK	OK	OK	Reference: ▶ Details (p.552)

Command Name	Function	Classification	Unit Calculation Macro	Scene Control Macro	Communication Command Macro	Unit Macro	References
GetUnitFigure	Gets figure data to the processing unit	Processing unit control	OK	OK	OK	OK	Reference: ▶ Details (p.554)
Gosub	Operate the specified subroutine	General instruction	OK	OK	OK	OK	Reference: ▶ Details (p.556)
Goto	Moves the process to the statement line with a specified label	General instruction	OK	OK	OK	OK	Reference: ▶ Details (p.558)
Hex$	Converts the value in the expression to the hexadecimal value in character string format	String Operation	OK	OK	OK	OK	Reference: ▶ Details (p.560)
If Then Else	Controls the process flow according to the specified condition	General instruction	OK	OK	OK	OK	Reference: ▶ Details (p.562)
If Then Elseif Else EndIf	Controls the process flow according to the specified condition	General instruction	OK	OK	OK	OK	Reference: ▶ Details (p.564)
ImageFormat	Gets the image format of the image in the processing unit	Processing unit control	OK	OK	OK	OK	Reference: ▶ Details (p.566)
ImageUpdate	Updates the image input from the camera	Measurement control	---	OK	OK	---	Reference: ▶ Details (p.568)
Input#	Reads data from the file	File Control	OK	OK	OK	OK	Reference: ▶ Details (p.569)
Input$	Reads binary data from the file	File Control	OK	OK	OK	OK	Reference: ▶ Details (p.571)
InsertUnit	Inserts a processing unit	Flow control	---	OK	OK	---	Reference: ▶ Details (p.573)
Int	Converts numeric value to integer value	Arithmetic Calculation	OK	OK	OK	OK	Reference: ▶ Details (p.575)
Isfile	Checks the attribute and the existence of the file	File Control	OK	OK	OK	OK	Reference: ▶ Details (p.577)
ItemCount	Gets the number of useable processing item types	Control processing item	---	OK	OK	---	Reference: ▶ Details (p.579)
ItemIdent$	Gets the identification name of the processing item	Control processing item	---	OK	OK	---	Reference: ▶ Details (p.580)
ItemInfo	Gets the processing item information	Control processing item	---	OK	OK	---	Reference: ▶ Details (p.582)
ItemTitle$	Gets the processing item title	Control processing item	---	OK	OK	---	Reference: ▶ Details (p.584)
JudgeOut	Sets the output state of the overall judgement signal	IO Module Control	---	OK	OK	OK	Reference: ▶ Details (p.585)
Kill	Deletes a file	File Control	OK	OK	OK	OK	Reference: ▶ Details (p.587)
LCase$	Converts an upper case letter to a lower case letter	String Operation	OK	OK	OK	OK	Reference: ▶ Details (p.589)
Left$	Extracts the specified length of characters from the left side of character string	String Operation	OK	OK	OK	OK	Reference: ▶ Details (p.591)

Command Name	Function	Classification	Unit Calculation Macro	Scene Control Macro	Communication Command Macro	Unit Macro	References
Len	Gets the length of the specified character string	String Operation	OK	OK	OK	OK	Reference: ▶ Details (p.593)
Line Input#	Reads the data of one line from the file	File Control	OK	OK	OK	OK	Reference: ▶ Details (p.595)
List	Outputs all or a part of program list in the system status console window	Debug Command	OK	OK	OK	OK	Reference: ▶ Details (p.597)
LoadBackupData	Loads the system + scene group 0 data	Data Save/ Load	---	OK	OK	---	Reference: ▶ Details (p.599)
LoadScene	Loads the scene data	Data Save/ Load	---	OK	OK	---	Reference: ▶ Details (p.601)
LoadSceneGroup	Loads the scene group data	Data Save/ Load	---	OK	OK	---	Reference: ▶ Details (p.603)
LoadSystemData	Loads the system data	Data Save/ Load	---	OK	OK	---	Reference: ▶ Details (p.605)
LoadUnitData	Loads the processing unit data	Data Save/ Load	---	OK	OK	---	Reference: ▶ Details (p.607)
Log	Gets the natural logarithm	Arithmetic Calculation	OK	OK	OK	OK	Reference: ▶ Details (p.609)
Lsqumeth	Gets the approximate straight line from the coordinates of multiple points using the least squares method	Arithmetic Calculation	OK	OK	OK	OK	Reference: ▶ Details (p.610)
Measure	Executes measurement processing	Measurement control	---	OK	OK	---	Reference: ▶ Details (p.612)
MeasureDispG	Executes display of the measurement result of the processing unit	Processing unit control	---	---	---	OK	Reference: ▶ Details (p.614)
MeasureId$	Gets the measurement identification	Processing unit control	OK	---	---	OK	Reference: ▶ Details (p.615)
MeasureProc	Executes measurement processing in a processing unit	Processing unit control	---	---	---	OK	Reference: ▶ Details (p.616)
MeasureStart	Allows input of the measurement trigger	Measurement control	---	OK	OK	---	Reference: ▶ Details (p.617)
MeasureStop	Prohibits input of the measurement trigger	Measurement control	---	OK	OK	---	Reference: ▶ Details (p.619)
Mid$	Extracts a part from the character string	String Operation	OK	OK	OK	OK	Reference: ▶ Details (p.621)
Mkdir	Builds a directory	File Control	OK	OK	OK	OK	Reference: ▶ Details (p.623)
MOD	Gets the remainder	Arithmetic Calculation	OK	OK	OK	OK	Reference: ▶ Details (p.624)
MoveUnit	Moves a processing unit	Flow control	---	OK	OK	---	Reference: ▶ Details (p.625)
NOT	Gets the "not" result (negation) of the expression	Arithmetic Calculation	OK	OK	OK	OK	Reference: ▶ Details (p.626)

Command Name	Function	Classification	Unit Calculation Macro	Scene Control Macro	Communication Command Macro	Unit Macro	References
Open For Append As#	Open the file in append mode	File Control	OK	OK	OK	OK	Reference: ▶ Details (p.627)
Open For Input As#	Open the file in reading mode	File Control	OK	OK	OK	OK	Reference: ▶ Details (p.629)
Open For Output As#	Opens the file in writing mode	File Control	OK	OK	OK	OK	Reference: ▶ Details (p.631)
OpenTextData	Opens a messages file	Multilingual Support Message Functions	OK	---	---	OK	Reference: ▶ Details (p.633)
OR	Gets the logical sum of two expressions	Arithmetic Calculation	OK	OK	OK	OK	Reference: ▶ Details (p.635)
Piece$	Extract the part of the character string which was separated by delimiter from the string	String Operation	OK	OK	OK	OK	Reference: ▶ Details (p.637)
Print	Outputs data in the system status console window	Debug Command	OK	OK	OK	OK	Reference: ▶ Details (p.639)
Print#	Outputs data in a file	File Control	OK	OK	OK	OK	Reference: ▶ Details (p.641)
PutAll	Sets the output state of all output terminals	IO Module Control	---	OK	OK	OK	Reference: ▶ Details (p.643)
PutPort	Sets the output state of the specified output terminal	IO Module Control	---	OK	OK	OK	Reference: ▶ Details (p.645)
RaiseOptionEvent	Notifies option events to the UI screen	Others	---	OK	OK	---	Reference: ▶ Details (p.647)
ReadPlcMemory	Reads a value from the PLC memory area	IO Module Control	---	OK	OK	OK	Reference: ▶ Details (p.649)
ReceiveData	Receives data	IO Module Control	---	OK	OK	OK	Reference: ▶ Details (p.651)
RefreshImageWindow	Updates the image window	Display control	---	OK	OK	---	Reference: ▶ Details (p.653)
RefreshJudgeWindow	Updates the judgement window	Display control	---	OK	OK	---	Reference: ▶ Details (p.654)
RefreshTextWindow	Updates the text display window	Display control	---	OK	OK	---	Reference: ▶ Details (p.655)
RefreshTimeWindow	Updates the display of the information window	Display control	---	OK	OK	---	Reference: ▶ Details (p.656)
Rem	Puts a comment in the program	Others	OK	OK	OK	OK	Reference: ▶ Details (p.657)
Remeasure	Executes remeasurement	Measurement control	---	OK	OK	---	Reference: ▶ Details (p.658)
RenumUnitNo	Gets the processing unit number after flow edit	Others	OK	---	---	OK	Reference: ▶ Details (p.660)
RGB	Gets the color value	Arithmetic Calculation	OK	---	---	OK	Reference: ▶ Details (p.662)

Command Name	Function	Classification	Unit Calculation Macro	Scene Control Macro	Communication Command Macro	Unit Macro	References
Right$	Extracts the specified length of characters from the right side of character string	String Operation	OK	OK	OK	OK	Reference: ▶ Details (p.664)
Rmdir	Deletes a directory	File Control	OK	OK	OK	OK	Reference: ▶ Details (p.666)
RunOut	Sets the output state of the RUN signal	IO Module Control	---	OK	OK	OK	Reference: ▶ Details (p.667)
SaveBackupData	Saves the system + scene group 0 data	Data Save/ Load	---	OK	OK	---	Reference: ▶ Details (p.669)
SaveData	Saves the data to the controller	Data Save/ Load	---	OK	OK	---	Reference: ▶ Details (p.670)
SaveImage	Saves image data	Measurement control	---	OK	OK	---	Reference: ▶ Details (p.671)
SaveMeasurImage	Saves the measurement image of the processing unit	Processing unit control	---	---	---	OK	Reference: ▶ Details (p.673)
SaveScene	Saves the scene data	Data Save/ Load	---	OK	OK	---	Reference: ▶ Details (p.675)
SaveSceneGroup	Saves the scene group data	Data Save/ Load	---	OK	OK	---	Reference: ▶ Details (p.677)
SaveSystemData	Saves the system data	Data Save/ Load	---	OK	OK	---	Reference: ▶ Details (p.679)
SaveUnitData	Saves a processing unit	Data Save/ Load	---	OK	OK	---	Reference: ▶ Details (p.681)
SceneCount	Gets the number of scenes that can be used	Scene control	---	OK	OK	---	Reference: ▶ Details (p.683)
SceneDescription$	Gets the scene description	Scene control	OK	---	OK	---	Reference: ▶ Details (p.684)
SceneGroupCount	Gets the number of useable scene groups	Scene group control	---	OK	OK	---	Reference: ▶ Details (p.686)
SceneGroupNo	Gets the scene group number of the current scene group	Scene group control	---	OK	OK	---	Reference: ▶ Details (p.687)
SceneGroupTitle$	Gets the title of the scene group	Scene group control	---	OK	OK	---	Reference: ▶ Details (p.688)
SceneMaker$	Gets the scene creator	Scene control	---	OK	OK	---	Reference: ▶ Details (p.689)
SceneNo	Gets the scene number of the current scene	Scene control	---	OK	OK	---	Reference: ▶ Details (p.691)
SceneTitle$	Gets the scene title	Scene control	---	OK	OK	---	Reference: ▶ Details (p.692)
ScreenCapture	Saves the capture of the screen	Others	---	OK	OK	---	Reference: ▶ Details (p.693)
Select Case Case Else End Select	Controls the process flow according to the specified condition	General instruction	OK	OK	OK	OK	Reference: ▶ Details (p.694)
SendData	Sends data	IO Module Control	---	OK	OK	OK	Reference: ▶ Details (p.696)

Command Name	Function	Classification	Unit Calculation Macro	Scene Control Macro	Communication Command Macro	Unit Macro	References
SendString	Sends the character string data	IO Module Control	---	OK	OK	OK	Reference: ▶ Details (p.698)
SetDisplayUnitNo	Sets the processing unit number in the flow window to the selected state	Display control	---	OK	OK	---	Reference: ▶ Details (p.699)
SetDrawStyle	Set the drawing attributes of the graphic figure	Image Window Control	---	---	---	OK	Reference: ▶ Details (p.700)
SetGlobalData	Sets the global data	Control Global Data	OK	OK	OK	OK	Reference: ▶ Details (p.702)
SetImageWindow	Sets the state of the image window	Display control	---	OK	OK	OK	Reference: ▶ Details (p.704)
SetMeasureImage	Sets the measurement image of the processing unit	Processing unit control	---	---	---	OK	Reference: ▶ Details (p.707)
SetMeasureOut	Sets the external output setting for the measurement result	Measurement control	---	OK	OK	---	Reference: ▶ Details (p.708)
SetPlcData	Creates the data that is written with the WritePlcMemory function	IO Module Control	---	OK	OK	OK	Reference: ▶ Details (p.709)
SetPollingState	Sets the execution status of the communication module	IO Module Control	---	OK	OK	OK	Reference: ▶ Details (p.711)
SetSceneData	Sets data for the scene control macro	Scene control	OK	OK	OK	OK	Reference: ▶ Details (p.713)
SetSceneDescription	Sets the scene description	Scene control	---	OK	OK	---	Reference: ▶ Details (p.715)
SetSceneGroupTitle	Sets the title of the scene group	Scene group control	---	OK	OK	---	Reference: ▶ Details (p.717)
SetSceneMaker	Sets the creator of the scene	Scene control	---	OK	OK	---	Reference: ▶ Details (p.719)
SetSceneTitle	Sets the title of a scene	Scene control	---	OK	OK	---	Reference: ▶ Details (p.721)
SetStop	Sets the conditions for stopping program execution	Debug Command	OK	OK	OK	OK	Reference: ▶ Details (p.723)
SetSystemData	Sets the system data	System Data	OK	OK	OK	OK	Reference: ▶ Details (p.725)
SetTextStyle	Set the draw attributes of the character string	Image Window Control	---	---	---	OK	Reference: ▶ Details (p.727)
SetTextWindow	Sets the state of the text window	Display control	---	OK	OK	OK	Reference: ▶ Details (p.729)
SetUnitData	Sets the data of a processing unit	Processing unit control	OK	OK	OK	OK	Reference: ▶ Details (p.731)
SetUnitFigure	Sets the figure data of the processing unit	Processing unit control	---	OK	OK	OK	Reference: ▶ Details (p.733)
SetUnitJudge	Sets the judgement result of a processing unit	Processing unit control	OK	---	---	OK	Reference: ▶ Details (p.735)
SetUnitTitle	Sets the title of a processing unit	Processing unit control	---	OK	OK	OK	Reference: ▶ Details (p.737)

Command Name	Function	Classification	Unit Calculation Macro	Scene Control Macro	Communication Command Macro	Unit Macro	References
SetUserSubroutine	Register a user-defined function that has been defined in the external DDL file	Debug Command	OK	OK	OK	OK	Reference: ▶ Details (p.739)
SetVar	Sets all variables with the specified variable names	Others	OK	OK	OK	OK	Reference: ▶ Details (p.741)
Sin	Gets the sine of the specified expression	Arithmetic Calculation	OK	OK	OK	OK	Reference: ▶ Details (p.743)
Sqr	Determines the square root	Arithmetic Calculation	OK	OK	OK	OK	Reference: ▶ Details (p.745)
StartTimer	Starts the elapsed time measurement	Others	OK	OK	OK	OK	Reference: ▶ Details (p.746)
Stop	Stops program execution	Debug Command	OK	OK	OK	OK	Reference: ▶ Details (p.747)
Str$	Converts a numeric value in the numeric character string	String Operation	OK	OK	OK	OK	Reference: ▶ Details (p.749)
Str2$	Converts a value to a numeric character string in the specified formats	String Operation	OK	OK	OK	OK	Reference: ▶ Details (p.751)
SystemReset	Reboots the Sensor Controller	Others	---	---	OK	---	Reference: ▶ Details (p.754)
Tan	Gets the tangent of the specified expression	Arithmetic Calculation	OK	OK	OK	OK	Reference: ▶ Details (p.755)
Time$	Reads out the clock time from the internal clock	Arithmetic Calculation	OK	OK	OK	OK	Reference: ▶ Details (p.757)
Timer	Gets the elapsed time	Others	OK	OK	OK	OK	Reference: ▶ Details (p.759)
TotalJudge	Gets the total judgement result	Processing unit control	OK	---	---	OK	Reference: ▶ Details (p.761)
TransformAngle	Applies the calibration result and position correction amount in the angle value	Processing unit control	OK	---	---	OK	Reference: ▶ Details (p.763)
TransformArea	Applies the calibration result and position correction amount in the area value	Processing unit control	OK	---	---	OK	Reference: ▶ Details (p.765)
TransformDist	Applies a calibration result and position correction amount to a distance value	Processing unit control	OK	---	---	OK	Reference: ▶ Details (p.767)
TransformLine	Applies the calibration result and position correction amount to a line component value	Processing unit control	OK	---	---	OK	Reference: ▶ Details (p.769)
TransformXY	Applies the calibration result and position correction amount to coordinate values	Processing unit control	OK	---	---	OK	Reference: ▶ Details (p.771)
Try Catch End Try	Detects an error occurrence and executes an exception process	General instruction	OK	OK	OK	OK	Reference: ▶ Details (p.773)
UCase$	Converts an lower case letter to a upper case letter	String Operation	OK	OK	OK	OK	Reference: ▶ Details (p.775)

Command Name	Function	Classification	Unit Calculation Macro	Scene Control Macro	Communication Command Macro	Unit Macro	References
UnitCount	Gets the number of registered processing units	Flow control	---	OK	OK	---	Reference: ▶ Details (p.777)
UnitData	Gets the numerical data of a processing unit	Processing unit control	OK	OK	OK	OK	Reference: ▶ Details (p.778)
UnitData$	Gets the character string data of the specified processing unit	Processing unit control	OK	OK	OK	OK	Reference: ▶ Details (p.780)
UnitData2	Gets the drawing coordinate data of a processing unit	Processing unit control	OK	---	---	OK	Reference: ▶ Details (p.782)
UnitInfo	Gets the processing unit information	Processing unit control	OK	OK	OK	OK	Reference: ▶ Details (p.784)
UnitItemIdent$	Gets the processing item identification name of the specified processing unit	Processing unit control	OK	OK	OK	OK	Reference: ▶ Details (p.786)
UnitJudge	Gets the judgement result of a processing unit	Processing unit control	OK	---	---	OK	Reference: ▶ Details (p.788)
UnitNo	Gets the processing unit number	Processing unit control	OK	---	---	OK	Reference: ▶ Details (p.790)
UnitTitle$	Gets the title of a processing unit	Processing unit control	OK	OK	OK	OK	Reference: ▶ Details (p.791)
Ut	Gets a processing unit number based on the specified unit label	Scene control	OK	OK	OK	OK	Reference: ▶ Details (p.793)
Val	Converts a numeric character string to numeric value	String Operation	OK	OK	OK	OK	Reference: ▶ Details (p.794)
VarList	Outputs a list of the values of the specified variables in the system status console window	Debug Command	OK	OK	OK	OK	Reference: ▶ Details (p.796)
VarPop	Restores the value of the variables that are saved temporarily	Others	OK	OK	OK	OK	Reference: ▶ Details (p.798)
VarPush	Saves the value of the variables that are saved temporarily	Others	OK	OK	OK	OK	Reference: ▶ Details (p.801)
VarSave	Saves the values of the variables in the scene data	Others	---	OK	---	---	Reference: ▶ Details (p.804)
Wait	Pauses the program process for the specified amount of time elapses	Others	---	OK	OK	OK	Reference: ▶ Details (p.806)
WritePlcMemory	Writes values in the PLC memory area	IO Module Control	OK	OK	OK	OK	Reference: ▶ Details (p.807)
XOR	Gets the exclusive disjunction (XOR) of two expressions	Arithmetic Calculation	OK	OK	OK	OK	Reference: ▶ Details (p.809)

Function-based Index

General Instructions

Command	Function	References
Dim	Defines the array variable.	Reference: ▶ Details (p.461)
Do Loop While	Repeatedly executes the statements between Do and Loop while the specified condition meets.	Reference: ▶ Details (p.465)
Erase	Releases array variable.	Reference: ▶ Details (p.512)
Errcmnd$	Get the function name of the macro where an error occurred.	Reference: ▶ Details (p.513)
Errno	Gets the error number.	Reference: ▶ Details (p.515)
For To Step Next	Repeats the statements between the For and Next statements.	Reference: ▶ Details (p.526)
Gosub	Operate the specified subroutine.	Reference: ▶ Details (p.556)
Goto	Move the process to the statement line with a specified label.	Reference: ▶ Details (p.558)
If Then Else	Controls the process flow according to the specified condition.	Reference: ▶ Details (p.562)
If Then Elseif Else EndIf	Controls the process flow according to the specified condition.	Reference: ▶ Details (p.564)
Select Case Case Else End Select	Controls the process flow according to the specified condition.	Reference: ▶ Details (p.694)
Try Catch End Try	Detects an error occurrence and executes an exception process.	Reference: ▶ Details (p.773)

Arithmetic Calculations

Command	Function	References
Abs	Gets the absolute value of the specified expression.	Reference: ▶ Details (p.403)
AND	Gets the logical product of two expressions.	Reference: ▶ Details (p.409)
ApproximationCircle	Gets the approximate circle.	Reference: ▶ Details (p.411)
Atn	Getting the arctangent of the specified expression.	Reference: ▶ Details (p.417)
Cos	Gets the cosine of the specified expression.	Reference: ▶ Details (p.450)
Crspoint	Gets the intersection between 2 straight lines.	Reference: ▶ Details (p.452)
Date$	Reads out the date from the internal clock.	Reference: ▶ Details (p.454)
Dposline	Gets the shortest distance between the line and point.	Reference: ▶ Details (p.467)
Exp	Gets the value of the exponential function of the base e natural logarithm.	Reference: ▶ Details (p.520)
Fix	Gets the integer of a value by rounding off digits to the right of the decimal point.	Reference: ▶ Details (p.524)
Int	Converts numeric value to integer value.	Reference: ▶ Details (p.575)
Log	Gets the natural logarithm.	Reference: ▶ Details (p.609)
Lsqumeth	Gets the approximate straight line from the coordinates of multiple points using the least squares method.	Reference: ▶ Details (p.610)
MOD	Gets the remainder.	Reference: ▶ Details (p.624)
NOT	Gets the "not" result (negation) of the expression.	Reference: ▶ Details (p.626)
OR	Gets the logical sum of two expressions.	Reference: ▶ Details (p.635)

Command	Function	References
RGB	Gets the color value.	Reference: ▶ Details (p.662)
Sin	Gets the sine of the specified expression.	Reference: ▶ Details (p.743)
Sqr	Determining the square root.	Reference: ▶ Details (p.745)
Tan	Gets the tangent of the specified expression.	Reference: ▶ Details (p.755)
Time$	Reads out the clock time from the internal clock.	Reference: ▶ Details (p.757)
XOR	Gets the exclusive disjunction (XOR) of two expressions.	Reference: ▶ Details (p.809)

String Operations

Command	Function	References
Asc	Gets the character code of the specified character.	Reference: ▶ Details (p.413)
Chr$	Determining the character of the specified character code.	Reference: ▶ Details (p.427)
Hex$	Converts the value in the expression to the hexadecimal value in character string format.	Reference: ▶ Details (p.560)
LCase$	Converts an upper case letter to a lower case letter.	Reference: ▶ Details (p.589)
Left$	Extracts the specified length of characters from the left side of character string.	Reference: ▶ Details (p.591)
Len	Gets the length of the specified character string.	Reference: ▶ Details (p.593)
Mid$	Extract a part from the character string.	Reference: ▶ Details (p.621)
Piece$	Extract the part of the character string which was separated by delimiter from the string.	Reference: ▶ Details (p.637)
Right$	Extracts the specified length of characters from the right side of character string.	Reference: ▶ Details (p.664)
Str$	Converts a numeric value in the numeric character string.	Reference: ▶ Details (p.749)
Str2$	Converts a value to a numeric character string in the specified formats.	Reference: ▶ Details (p.751)
UCase$	Converts an lower case letter to a upper case letter.	Reference: ▶ Details (p.775)
Val	Converts a numeric character string to numeric value.	Reference: ▶ Details (p.794)

Scene Controls

Command	Function	References
ChangeScene	Change the scene.	Reference: ▶ Details (p.423)
ClearScene	Clears the scene data.	Reference: ▶ Details (p.430)
CopyScene	Copies scene data.	Reference: ▶ Details (p.440)
GetSceneData	Gets data related to the scene control macro.	Reference: ▶ Details (p.544)
SceneCount	Gets the number of scenes that can be used.	Reference: ▶ Details (p.683)
SceneDescription$	Gets the scene description.	Reference: ▶ Details (p.684)
SceneMaker$	Gets the scene creator.	Reference: ▶ Details (p.689)
SceneNo	Gets the scene number of the current scene.	Reference: ▶ Details (p.691)
SceneTitle$	Gets the scene title.	Reference: ▶ Details (p.692)
SetSceneData	Sets data for the scene control macro.	Reference: ▶ Details (p.713)
SetSceneDescription	Sets the scene description.	Reference: ▶ Details (p.715)
SetSceneMaker	Sets the creator of the scene.	Reference: ▶ Details (p.719)
SetSceneTitle	Sets the title of a scene.	Reference: ▶ Details (p.721)

Command	Function	References
Ut	Gets a processing unit number based on the specified unit label.	Reference: ▶ Details (p.793)

Scene Group Controls

Command	Function	References
ChangeSceneGroup	Changes the scene group.	Reference: ▶ Details (p.424)
ClearSceneGroup	Clears scene group data.	Reference: ▶ Details (p.431)
CopySceneGroup	Copies scene group data.	Reference: ▶ Details (p.441)
SceneGroupCount	Gets the number of useable scene groups.	Reference: ▶ Details (p.686)
SceneGroupNo	Gets the scene group number of the current scene group.	Reference: ▶ Details (p.687)
SceneGroupTitle$	Gets the title of the scene group.	Reference: ▶ Details (p.688)
SetSceneGroupTitle	Sets the title of the scene group.	Reference: ▶ Details (p.717)

Processing Item Controls

Command	Function	References
ItemCount	Gets the number of useable processing item types.	Reference: ▶ Details (p.579)
ItemIdent$	Gets the identification name of the processing item.	Reference: ▶ Details (p.580)
ItemInfo	Gets the processing item information.	Reference: ▶ Details (p.582)
ItemTitle$	Gets the processing item title.	Reference: ▶ Details (p.584)

Processing Flow Controls

Command	Function	References
AssignUnit	Registers the processing unit.	Reference: ▶ Details (p.415)
CheckUnit	Checks the registration status of a processing unit.	Reference: ▶ Details (p.425)
CopyUnit	Copies a processing unit.	Reference: ▶ Details (p.442)
DeleteUnit	Deletes a processing unit.	Reference: ▶ Details (p.460)
InsertUnit	Inserts a processing unit.	Reference: ▶ Details (p.573)
MoveUnit	Moves a processing unit.	Reference: ▶ Details (p.625)
UnitCount	Gets the number of registered processing units.	Reference: ▶ Details (p.777)

Processing Unit Controls

Command	Function	References
CopyMeasureImage	Copies the measurement image as an image of the Unit Macro processing unit.	Reference: ▶ Details (p.438)
CopyUnitFigure	Copies figure data to the processing unit.	Reference: ▶ Details (p.444)
CopyUnitImage	Copies a processing unit image as a unit macro processing unit image.	Reference: ▶ Details (p.446)
CopyUnitModel	Copies the model data of a processing unit.	Reference: ▶ Details (p.448)
GetImageSize	Gets the image size of the processing unit image.	Reference: ▶ Details (p.532)
GetUnitData	Gets the data of a processing unit.	Reference: ▶ Details (p.552)

Command	Function	References
GetUnitFigure	Gets figure data to the processing unit.	Reference: ▶ Details (p.554)
ImageFormat	Gets the image format of the image in the processing unit.	Reference: ▶ Details (p.566)
MeasureDispG	Executes display of the measurement result of the processing unit.	Reference: ▶ Details (p.614)
MeasureId$	Gets the measurement identification.	Reference: ▶ Details (p.615)
MeasureProc	Executes measurement processing in a processing unit.	Reference: ▶ Details (p.616)
SaveMeasureImage	Saves the measurement image of the processing unit.	Reference: ▶ Details (p.673)
SetMeasureImage	Sets the measurement image of the processing unit.	Reference: ▶ Details (p.707)
SetUnitData	Sets the data of a processing unit.	Reference: ▶ Details (p.731)
SetUnitFigure	Sets the figure data of the processing unit.	Reference: ▶ Details (p.733)
SetUnitJudge	Sets the judgement result of a processing unit.	Reference: ▶ Details (p.735)
SetUnitTitle	Sets the title of a processing unit.	Reference: ▶ Details (p.737)
TotalJudge	Gets the total judgement result.	Reference: ▶ Details (p.761)
TransformAngle	Applies the calibration result and position correction amount in the angle value.	Reference: ▶ Details (p.763)
TransformArea	Applies the calibration result and position correction amount in the area value.	Reference: ▶ Details (p.765)
TransformDist	Applies a calibration result and position correction amount to a distance value.	Reference: ▶ Details (p.767)
TransformLine	Applies the calibration result and position correction amount to a line component value.	Reference: ▶ Details (p.769)
TransformXY	Applies the calibration result and position correction amount to coordinate values.	Reference: ▶ Details (p.771)
UnitData	Gets the numerical data of a processing unit.	Reference: ▶ Details (p.778)
UnitData$	Gets the character string data of the specified processing unit.	Reference: ▶ Details (p.780)
UnitData2	Gets the drawing coordinate data of a processing unit.	Reference: ▶ Details (p.782)
UnitInfo	Gets the processing unit information.	Reference: ▶ Details (p.784)
UnitItemIdent$	Gets the processing item identification name of the specified processing unit.	Reference: ▶ Details (p.786)
UnitJudge	Gets the judgement result of a processing unit.	Reference: ▶ Details (p.788)
UnitNo	Gets the processing unit number.	Reference: ▶ Details (p.790)
UnitTitle$	Gets the title of a processing unit.	Reference: ▶ Details (p.791)

Measurement Controls

Command	Function	References
ClearMeasureData	Clears the measurement results of the processing unit.	Reference: ▶ Details (p.429)
GetMeasureOut	Gets the external output setting for measurement results.	Reference: ▶ Details (p.536)
ImageUpdate	Updates the image input from the camera.	Reference: ▶ Details (p.568)
Measure	Executes measurement processing.	Reference: ▶ Details (p.612)
MeasureStart	Allows input of the measurement trigger.	Reference: ▶ Details (p.617)
MeasureStop	Prohibits input of the measurement trigger.	Reference: ▶ Details (p.619)
Remeasure	Executes remeasurement.	Reference: ▶ Details (p.658)
SaveImage	Saves image data.	Reference: ▶ Details (p.671)

Command	Function	References
SetMeasureOut	Sets the external output setting for the measurement result.	Reference: ▶ Details (p.708)

IO Module Controls

Command	Function	References
BusyOut	Sets the output state of the processing busy signal.	Reference: ▶ Details (p.419)
GetAll	Gets the input states of all input terminals.	Reference: ▶ Details (p.528)
GetPlcData	Gets data read with the ReadPlcMemory function.	Reference: ▶ Details (p.538)
GetPollingState	Gets the polling state of the communication module.	Reference: ▶ Details (p.540)
GetPort	Gets the input state of the specified input terminal.	Reference: ▶ Details (p.542)
JudgeOut	Sets the output state of the overall judgement signal.	Reference: ▶ Details (p.585)
PutAll	Sets the output state of all output terminals.	Reference: ▶ Details (p.643)
PutPort	Sets the output state of the specified output terminal.	Reference: ▶ Details (p.645)
ReadPlcMemory	Reads a value from the PLC memory area.	Reference: ▶ Details (p.649)
ReceiveData	Receives data.	Reference: ▶ Details (p.651)
RunOut	Sets the output state of the RUN signal.	Reference: ▶ Details (p.667)
SendData	Sends data.	Reference: ▶ Details (p.696)
SendString	Sends the character string data.	Reference: ▶ Details (p.698)
SetPlcData	Creates the data that is written with the WritePlcMemory function.	Reference: ▶ Details (p.709)
SetPollingState	Sets the execution status of the communication module.	Reference: ▶ Details (p.711)
WritePlcMemory	Writes values in the PLC memory area.	Reference: ▶ Details (p.807)

Display Controls

Command	Function	References
DisplayUnitNo	Gets the selection state of the processing unit number of the flow window.	Reference: ▶ Details (p.464)
GetImageWindow	Get the state of the image window.	Reference: ▶ Details (p.533)
GetTextWindow	Gets the state of the text window.	Reference: ▶ Details (p.550)
RefreshImageWindow	Updates the image window.	Reference: ▶ Details (p.653)
RefreshJudgeWindow	Updates the judgement window.	Reference: ▶ Details (p.654)
RefreshTextWindow	Updating the text display window.	Reference: ▶ Details (p.655)
RefreshTimeWindow	Updates the display of the information window.	Reference: ▶ Details (p.656)
SetDisplayUnitNo	Sets the processing unit number in the flow window to the selected state.	Reference: ▶ Details (p.699)
SetImageWindow	Sets the state of the image window.	Reference: ▶ Details (p.704)
SetTextWindow	Sets the state of the text window.	Reference: ▶ Details (p.729)

Image Window Controls

Command	Function	References
DisplaySubNo	Get the sub-image number of the displayed sub-image.	Reference: ▶ Details (p.462)

Command	Function	References
DrawArc	Draw the arc on the image window.	Reference: ▶ Details (p.469)
DrawArcW	Draw the wide arc on the image window.	Reference: ▶ Details (p.471)
DrawBox	Draws a rectangle on the image window.	Reference: ▶ Details (p.473)
DrawCircle	Draw a circle on the image window.	Reference: ▶ Details (p.475)
DrawCircleW	Draw the wide circle on the image window.	Reference: ▶ Details (p.477)
DrawCursor	Draw the cross-hair cursor on the image window.	Reference: ▶ Details (p.479)
DrawEllipse	Draw the ellipse on the image window.	Reference: ▶ Details (p.481)
DrawFigure	Draw a figure on the image window.	Reference: ▶ Details (p.483)
DrawFillImage	Draw the fill image on the image window.	Reference: ▶ Details (p.485)
DrawLine	Draw a straight line on the image window.	Reference: ▶ Details (p.488)
DrawLineW	Draw the wide straight line on the image window.	Reference: ▶ Details (p.490)
DrawMeasureImage	Draw the measurement image on the image window.	Reference: ▶ Details (p.492)
DrawPoint	Draw a point on the image window.	Reference: ▶ Details (p.493)
DrawPolygon	Draw a polygon on the image window.	Reference: ▶ Details (p.495)
DrawSearchFigure	Draw the search figure on the image window.	Reference: ▶ Details (p.497)
DrawTextG	Draw a character string on the image window.	Reference: ▶ Details (p.502)
DrawUnitImage	Display the "other unit image" on the image window.	Reference: ▶ Details (p.504)
SetDrawStyle	Set the drawing attributes of the graphic figure.	Reference: ▶ Details (p.700)
SetTextStyle	Set the draw attributes of the character string.	Reference: ▶ Details (p.727)

Text Window Controls

Command	Function	References
DrawJudgeText	Draws the judgement result of the character string on the text display screen.	Reference: ▶ Details (p.486)
DrawText	Draw a character string on the text window.	Reference: ▶ Details (p.500)

System Data

Command	Function	References
AddSystemData	Adds the system data.	Reference: ▶ Details (p.407)
GetSystemData	Gets the system data.	Reference: ▶ Details (p.546)
SetSystemData	Sets the system data.	Reference: ▶ Details (p.725)

Global Data

Command	Function	References
AddGlobalData	Adds the global data.	Reference: ▶ Details (p.405)
GetGlobalData	Gets the global data.	Reference: ▶ Details (p.530)
SetGlobalData	Sets the global data.	Reference: ▶ Details (p.702)

Data Save/Load

Command	Function	References
LoadBackupData	Loads the system + scene group 0 data.	Reference: ▶ Details (p.599)
LoadScene	Loads the scene data.	Reference: ▶ Details (p.601)
LoadSceneGroup	Loads the scene group data.	Reference: ▶ Details (p.603)
LoadSystemData	Loads the system data.	Reference: ▶ Details (p.605)
LoadUnitData	Loads the processing unit data.	Reference: ▶ Details (p.607)
SaveBackupData	Saves the system + scene group 0 data.	Reference: ▶ Details (p.669)
SaveData	Saves the data to the controller.	Reference: ▶ Details (p.670)
SaveScene	Saves the scene data.	Reference: ▶ Details (p.675)
SaveSceneGroup	Saves the scene group data.	Reference: ▶ Details (p.677)
SaveSystemData	Saves the system data.	Reference: ▶ Details (p.679)
SaveUnitData	Saves a processing unit.	Reference: ▶ Details (p.681)

File Controls

Command	Function	References
Close	Closes up the file.	Reference: ▶ Details (p.432)
Dskf	Gets the free space on disk drives.	Reference: ▶ Details (p.506)
Eof	Examines the end of the file.	Reference: ▶ Details (p.510)
Fcopy	Copies the file.	Reference: ▶ Details (p.522)
Input#	Reads data from the file.	Reference: ▶ Details (p.569)
Input$	Reads binary data from the file.	Reference: ▶ Details (p.571)
Isfile	Checks the attribute and the existence of the file.	Reference: ▶ Details (p.577)
Kill	Deletes a file.	Reference: ▶ Details (p.587)
Line Input#	Reads the data of one line from the file.	Reference: ▶ Details (p.595)
Mkdir	Build a directory	Reference: ▶ Details (p.623)
Open For Append As#	Open the file in append mode.	Reference: ▶ Details (p.627)
Open For Input As#	Open the file in reading mode.	Reference: ▶ Details (p.629)
Open For Output As#	Opens the file in writing mode.	Reference: ▶ Details (p.631)
Print#	Outputs data in a file.	Reference: ▶ Details (p.641)
Rmdir	Deletes a directory.	Reference: ▶ Details (p.666)

Multilingual Support Message Functions

Command	Function	References
CloseTextData	Close up a messages file.	Reference: ▶ Details (p.434)
GetText$	Get a text data from a messages file.	Reference: ▶ Details (p.548)
OpenTextData	Opens a messages file.	Reference: ▶ Details (p.633)

Debug Commands

Command	Function	References
Call	Executes the registered user-defined function.	Reference: ▶ Details (p.421)
Cont	Resumes execution of the program after it has been stopped.	Reference: ▶ Details (p.436)
Debug	Set the program execution form and information output method.	Reference: ▶ Details (p.456)
DebugPrint	Outputs debug information to the system status console window.	Reference: ▶ Details (p.458)
List	Outputs all or a part of program list in the system status console window.	Reference: ▶ Details (p.597)
Print	Outputs data in the system status console window.	Reference: ▶ Details (p.639)
SetStop	Sets the conditions for stopping program execution.	Reference: ▶ Details (p.723)
SetUserSubroutine	Register a user-defined function that has been defined in the external DDL file.	Reference: ▶ Details (p.739)
Stop	Stops program execution.	Reference: ▶ Details (p.747)
VarList	Outputs a list of the values of the specified variables in the system status console window.	Reference: ▶ Details (p.796)

Others

Command	Function	References
ElapsedTime	Gets the elapsed time since starting the measurement.	Reference: ▶ Details (p.508)
ExecuteImageLogging	Executes image logging.	Reference: ▶ Details (p.517)
ExitFzProcess	Terminate the Sensor Controller.	Reference: ▶ Details (p.519)
RaiseOptionEvent	Notifies option events to the UI screen.	Reference: ▶ Details (p.647)
Rem	Put a comment in the program.	Reference: ▶ Details (p.657)
RenumUnitNo	Gets the processing unit number after flow edit.	Reference: ▶ Details (p.660)
ScreenCapture	Saves the capture of the screen.	Reference: ▶ Details (p.693)
SetVar	Sets all variables with the specified variable names.	Reference: ▶ Details (p.741)
StartTimer	Starts the elapsed time measurement.	Reference: ▶ Details (p.746)
SystemReset	Reboots the Sensor Controller.	Reference: ▶ Details (p.754)
Timer	Gets the elapsed time.	Reference: ▶ Details (p.759)
VarPop	Restores the value of the variables that are saved temporarily.	Reference: ▶ Details (p.798)
VarPush	Saves the value of the variables that are saved temporarily.	Reference: ▶ Details (p.801)
VarSave	Saves the values of the variables in the scene data.	Reference: ▶ Details (p.804)
Wait	Pauses the program process for the specified amount of time elapses.	Reference: ▶ Details (p.806)

Abs

Gets the absolute value of the specified expression.

Format

Abs(<expression>)

Parameter

Parameter name	Data type	Description
<expression>	Integer type Double precision real number data type	Expression to get the absolute value

Return value

Returns a double precision real absolute value.

Description

Gets the absolute value of the expression specified in the <expression> parameter.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a value is assigned to the return value variable or the variable is not used in an expression, a "Syntax error" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

Gets the difference between the two points (X1, Y1) and (X2, Y2).

```
X1# = 100
Y1# = 200
X2# = 200
Y2# = 100

DX# = Abs(X1# - X2#)
DY# = Abs(Y1# - Y2#)
```

The result is shown below.

```
DX# = 100
DY# = 100
```

Useable Modules

Unit Calculation Macro / Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 3.50 or later

Related Items

GetUnitData (Reference: ▶ Details (p.552))

UnitData (Reference: ▶ Details (p.778))

AddGlobalData

Adds the global data.

Format

AddGlobalData <dataIdent>, <data>

Parameter

Parameter name	Data type	Description
<dataIdent>	Character string type	Identification name of the global data to add
<data>	Integer type Double precision real number data type Character string type	Value added in the global data

Return value

None.

Description

Adds the data of the identification name specified in the <dataIdent> parameter to the global data, and sets the value specified in the <data> parameter in the added data.
If the global data of the specified identification name already exists, the process ends without taking any action.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a character string longer than 255 characters is specified in the <dataIdent> parameter, a "String too long" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

Adds the global data that has the identification name "ABC", and sets 1 as the value.

```
Rem Add global data "ABC", and set 1 as the value
AddGlobalData "ABC", 1

Rem Get the value (integer value) set in the global data "ABC", and store in the variable DATA&.
GetGlobalData "ABC", DATA&
```

Useable Modules

Unit Calculation Macro / Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 3.50 or later

Related Items

GetGlobalData (Reference: ▶ Details (p.530))

SetGlobalData (Reference: ▶ Details (p.702))

AddSystemData

Adds the system data.

Format

AddSystemData <dataIdent0>, <dataIdent1>, <data>

Parameter

Parameter name	Data type	Description
<dataIdent0>	Character string type	Data identification name of identification information 0 of system data to be added (specify "PanDA")
<dataIdent1>	Character string type	Data identification name of identification information 1 of system data to be added
<data>	Integer type Double precision real number data type Character string type	Value of the system data to add

Return value

None.

Description

Adds the data of identification information 1 specified in the <dataIdent1> parameter, which belongs to identification information 0 specified in the <datdaIdent0> parameter, to the system data, and sets the value specified in the <data> parameter in the added data.
If an identification name other than "PanDA" is specified in the <datadent0> parameter, an "Illegal function call" error will occur.
If the system data of identification information 1 that belongs to the specified identification information 0 is already registered, no action is taken and the process ends.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If an identification name that does not exist is specified in the <dataIdent0> parameter, an "Illegal function call" error will occur.
If a character string longer than 255 characters is specified in the <dataIdent1> parameter, a "String too long" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

Adds the data of identification information 1, "LoggingCount", to the system data of identification information 0, "PanDA". Sets 20 for the setting data.

```
AddSystemData "PanDA", "LoggingCount", 20
```

Useable Modules

Unit Calculation Macro / Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 3.50 or later

Related Items

AddGlobalData (Reference: ▶Details (p.405))
GetSystemData (Reference: ▶ Details (p.546))
SetSystemData (Reference: ▶ Details (p.725))

AND

Gets the logical product of two expressions.

Format

<expression1> AND <expression2>

Parameter

Parameter name	Data type	Description
<expression1>	Integer type	Expression to calculate the logical product
<expression2>	Integer type	Expression to calculate the logical product

Return value

Returns the logical product as an integer value.

Description

Gets the logical product by bit of the expression specified in the <expression1> parameter and the expression specified in the <expression2> parameter.
When the values of the <expression1> parameter and <expression2> parameter are double precision real values, the decimal part of the returned logical product is rounded off.
This can also be used as an And condition in an If statement. For details on the logical expression, refer to "Operators".
Reference: ▶Operator (p.213)
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a value outside the range -2147483648 to 2147483647 is specified as an integer parameter, an "Overflow" error will occur.
If a value is assigned to the return value variable or the variable is not used in an expression, a "Syntax error" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

Gets the logical product of variable X and variable Y.

```
X& = 15
Y& = 8

DATA& = X& AND Y&
```

The result is shown below.

```
DATA& = 8
```

Useable Modules

Unit Calculation Macro / Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 3.50 or later

Related Items

GetUnitData (Reference: ▶ Details (p.552))
OR (Reference: ▶ Details (p.635))
XOR (Reference: ▶ Details (p.809))
NOT (Reference: ▶ Details (p.626))
UnitData (Reference: ▶ Details (p.778))

ApproximationCircle

Gets the approximate circle.

Format

ApproximationCircle <count>, <x()>, <y()>, <centerX>, <centerY>, <radius>

Parameter

Parameter name	Data type	Description
<count>	Integer type	Number of the specified coordinate
<x()>	Integer array Double precision real number array	1D array stored each point of the X-coordinate
<y()>	Integer array Double precision real number array	1D array stored each point of the Y-coordinate
<centerX>	Double precision real number data type	Central X coordinate of the approximate circle
<centerY>	Double precision real number data type	Central Y coordinate of the approximate circle
<radius>	Double precision real number data type	Radius of the approximate circle

Return value

None.

Description

Gets the approximate circle from the specified number of points, whose coordinates are specified in the <x()> and <y()> parameters, in the <count> parameter. In the <centerX> and <centerY> parameters, respectively specify the variables that will hold the center X coordinate and center Y coordinate of the approximate circle gotten. In the <radius> parameter, specify the variable that will hold the radius of the circle gotten.
In the <count> parameter, specify an integer value of at least 3.
In the <x()> parameter and <y()> parameter, respectively specify a 1D integer array variable or double precision real number array variable, where a number of coordinate values greater than or equal to the number specified in the <count> parameter are stored, without adding element numbers but adding () to the variables.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a non-existent number, numerical value, or combination of data types or values is specified for a parameter, an "Illegal function call" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

Calculates the approximate circle and getting the center coordinates and radius from each of the three points (50, 50), (100, 100), and (150, 50).

```
Dim X&(3), Y&(3)

Rem Assign the three values of the coordinate value to the array.
X&(0) = 50
Y&(0) = 50
X&(1) = 100
Y&(1) = 100
X&(2) = 150
Y&(2) = 50

Rem Calculate the approximate circle and get the center coordinates and the radius.
ApproximationCircle 3, X&(), Y&(), CENTERX#, CENTERY#, RADIUS#
```

Useable Modules

Unit Calculation Macro / Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 3.50 or later

Related Items

GetUnitData (Reference: ▶ Details (p.552))
UnitData (Reference: ▶ Details (p.778))
Lsqumeth (Reference: ▶ Details (p.610))

Asc

Gets the character code of the specified character.

Format

Asc <string>

Parameter

Parameter name	Data type	Description
<string>	Character string type	Character string that requests the character code.

Return value

Returns the integer type character code value in decimal.

Description

Gets the character code of the first character in a character string specified in the <string> parameter in ASCII code.
Asc is the inverse function of Chr$. Chr$ returns the character corresponds to the specified character code.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
Even if a character string longer than 255 characters is specified for a character string parameter, an error will not occur.
If a value is assigned to the return value variable or the variable is not used in an expression, a "Syntax error" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

Gets the character code of letter "A".

```
CHARA$ = "A"

CODE& = Asc(CHARA$)
```

The result is shown below.

```
CODE& = 65
```

Useable Modules

Unit Calculation Macro / Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 3.50 or later

Related Items

Chr$ (Reference: ▶ Details (p.427))
LCase$ (Reference: ▶ Details (p.589))
Len (Reference: ▶ Details (p.593))
Piece$ (Reference: ▶ Details (p.637))
Str$ (Reference: ▶ Details (p.749))
UCase$ (Reference: ▶ Details (p.775))
Hex$ (Reference: ▶ Details (p.560))
Left$ (Reference: ▶ Details (p.591))
Mid$ (Reference: ▶ Details (p.621))
Right$ (Reference: ▶ Details (p.664))
Str2$ (Reference: ▶ Details (p.751))
Val (Reference: ▶ Details (p.794))

AssignUnit

Registers the processing unit.

Format

AssignUnit <unitNo>, <itemIdent>

Parameter

Parameter name	Data type	Description
<unitNo>	Integer type	Processing unit number on the measurement flow to register the processing unit (0 to the enrollment number of the processing units at the current scene)
<itemIdent>	Character string type	Identification name of the processing item to register as a processing unit

Return value

None.

Description

Registers the processing item with the identification name specified in the <itemIdent> parameter as a processing unit in the position in the measurement flow specified in the <unitNo> parameter.
If a processing unit is already registered in the position specified in the <unitNo> parameter, that processing unit is overwritten.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a non-existent number, numerical value, or combination of data types or values is specified for a parameter, an "Illegal function call" error will occur.
If an identification name that does not exist is specified as the <itemIdent> parameter, an "Illegal function call" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- Execute this macro function when the BUSY signal or other measurement in progress signal is ON and measurement is prohibited. (Reference: ▶State Transitions and Execution Timing (p.227))

Example

Registers the search processing unit at the end of the measurement flow.

```
Rem Get the number of processing units registered in the current measurement flow.
UNUM& = UnitCount

Rem Add the search processing unit.
AssignUnit UNUM&, "Search"
```

Useable Modules

Scene Control Macro / Communication Command Macro

Supported Versions

Version 3.50 or later

Related Items

CheckUnit (Reference: ▶ Details (p.425))
DeleteUnit (Reference: ▶ Details (p.460))
MeasureStart (Reference: ▶ Details (p.617))
MoveUnit (Reference: ▶ Details (p.625))
CopyUnit (Reference: ▶ Details (p.442))
InsertUnit (Reference: ▶ Details (p.573))
MeasureStop (Reference: ▶ Details (p.619))
UnitCount (Reference: ▶ Details (p.777))

Atn

Getting the arctangent of the specified expression.

Format

Atn(<expression>)

Parameter

Parameter name	Data type	Description
<expression>	Integer type Double precision real number data type	Expression that gets the arc tangent

Return value

Returns the arctangent as a double precision real value in the range -π/2 to π/2 radians.

Description

Gets the arctangent of the expression specified in the <expression> parameter.
To convert the gotten value to an angle, multiply by π/180.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a value is assigned to the return value variable or the variable is not used in an expression, a "Syntax error" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

Gets the arctangent of the variable X#.

```
X# = 1

XX# = Atn(X#)*180/3.141592
```

The result is shown below. (The returned value is rounded off to the nearest thousandth.)

```
XX# = 45.000
```

Useable Modules

Unit Calculation Macro / Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 3.50 or later

Related Items

Cos (Reference: ▶ Details (p.450))
Sin (Reference: ▶ Details (p.743))
UnitData (Reference: ▶ Details (p.778))
GetUnitData (Reference: ▶ Details (p.552))
Tan (Reference: ▶ Details (p.755))

BusyOut

Sets the output state of the processing busy signal.

Format

BusyOut <ioIdent>, <state>

Parameter

Parameter name	Data type	Description
<ioIdent>	Character string type	Identification name of the communication module to be used ("ParallelIo" or "EtherCAT") (Reference: ▶List of I/O Modules (p.341))
<state>	Integer type	Output state of terminal 0: Output OFF 1: Output ON

Return value

None.

Description

Set the output state specified in the <state> parameter in the processing busy signal, such as the BUSY signal, of the communication module specified in the <ioIdent> parameter.
Normally "ParallelIo" or "EtherCAT" should be specified in the <ioIdent> parameter.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If an identification name that does not exist is specified in the <ioIdent> parameter, an "Illegal function call" error will occur.
Even if an output status parameter value that does not exist (i.e., other than 0 and 1) is specified in the <state> parameter, an error will not occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

In the communication command macro, sets the BUSY signal of parallel I/O to ON.

```
IOMODULE$ = "ParallelIo"

Rem Set the output state.
BusyOut IOMODULE$, 1
```

Useable Modules

Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 3.50 or later

Related Items

GetAll (Reference: ▶ Details (p.528))
GetPort (Reference: ▶ Details (p.542))
JudgeOut (Reference: ▶ Details (p.585))
PutAll (Reference: ▶ Details (p.643))
PutPort (Reference: ▶ Details (p.645))
RunOut (Reference: ▶ Details (p.667))

Call

Executes the registered user-defined function.

Format

Call <subroutineIdent>[; | , <argument>...]

Parameter

Parameter name	Data type	Description
<subroutineIdent>	Character string type	Identification name of the user-defined function that has been registered
<argument>	Integer type Double precision real number data type Character string type	Argument of the user-defined function that has been registered

Return value

None.

Description

Calls a user-defined function with the specified identification name by the <subroutineIdent> parameter and executes the function. If the user-defined function has arguments, the specified data by the <argument> parameter is passed to the user-defined function. In the <argument> parameter, specify an argument according to the user-defined function definition.
Register a user-defined function with the SetUserSubroutine function before executing this function.
How user-defined functions work depends on the defined processes in external DLL files.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a user-defined function that is not registered is specified for a parameter, an "Illegal function call" error will occur.
If a character string longer than 255 characters is specified for a character string parameter, a "String too long" error will occur.
In case an error is returned as a result of the user-defined function execution, this Call function will return one of the following errors.

- Syntax error
- Illegal function call
- Out of memory
- Type missmatch

If this macro function calls a user-defined function that has not been programmed with a supported interface, an error occurs during the user-defined function processing.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- Only the user-defined functions that have been defined in programmed DLL files by the supported interfaces are accepted to this macro function. If other DLL files or user-defined functions than above is used, unexpected operation may occur such as a measurement error or the Sensor Controller abnormal termination. For user-defined functions creation, refer to the FH-AP1.
- Depending on the called user-defined function processings by this macro function, unexpected operation may occur such as a measurement error or the Sensor Controller abnormal termination. Be sure to fully check the operations and debug with external devices disconnected from the sensor controller in advance.
- To operate this macro function in an actual environment with external devices connected to, always apply external fail safe measures to the system.

Example

With identification name "USR", registers a user-defined function "UserProc0" that has been defined in MacroUserProc.dll. Then, specifies the identification name to call the user-defined function and executes it.

```
Rem Register the user-defined function so that the function can be used in this program
SetUserSubroutine "USR", "MacroUserProc", "UserProc0"

Rem Call the registered user-defined function and execute it
Call "USR", 0
```

Useable Modules

Unit Calculation Macro / Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 5.20 or later

Related Items

SetUserSubroutine (Reference: ▶ Details (p.739))

ChangeScene

Change the scene.

Format

ChangeScene <sceneNo>

Parameter

Parameter name	Data type	Description
<sceneNo>	Integer type	Scene number to change (0 to 127)

Return value

None.

Description

Changes the current scene to the scene with the scene number specified in the <sceneNo> parameter.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a non-existent number, numerical value, or combination of data types or values is specified for a parameter, an "Illegal function call" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- Execute this macro function when the BUSY signal or other measurement in progress signal is ON and measurement is prohibited. (Reference: ▶State Transitions and Execution Timing (p.227))

Example

Changes the current scene to scene number 2.

```
ChangeScene 2
```

Useable Modules

Scene Control Macro / Communication Command Macro

Supported Versions

Version 3.50 or later

Related Items

ChangeSceneGroup (Reference: ▶ Details (p.424))
MeasureStart (Reference: ▶ Details (p.617))
MeasureStop (Reference: ▶ Details (p.619))
SceneCount (Reference: ▶ Details (p.683))
SceneNo (Reference: ▶ Details (p.691))
SceneGroupNo (Reference: ▶ Details (p.687))

ChangeSceneGroup

Changes the scene group.

Format

ChangeSceneGroup <sceneGroupNo>, <sceneNo>

Parameter

Parameter name	Data type	Description
<sceneGroupNo>	Integer type	New scene group (0 to 31)
<sceneNo>	Integer type	Scene number to change (0 to 127)

Return value

None.

Description

Changes the current scene group to the scene specified in the <sceneNo> parameter, which belongs to the scene group specified in the <sceneGroupNo> parameter.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a non-existent number, numerical value, or combination of data types or values is specified for a parameter, an "Illegal function call" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- Execute this macro function when the BUSY signal or other measurement in progress signal is ON and measurement is prohibited. (Reference: ▶State Transitions and Execution Timing (p.227))

Example

Changes the current scene group to scene 2 of scene group 10.

```
ChangeSceneGroup 10, 2
```

Useable Modules

Scene Control Macro / Communication Command Macro

Supported Versions

Version 3.50 or later

Related Items

ChangeScene (Reference: ▶ Details (p.423))
MeasureStop (Reference: ▶ Details (p.619))
SceneGroupNo (Reference: ▶ Details (p.687))
MeasureStart (Reference: ▶ Details (p.617))
SceneGroupCount (Reference: ▶ Details (p.686))

CheckUnit

Checks the registration status of a processing unit.

Format

CheckUnit(<unitNo>)

Parameter

Parameter name	Data type	Description
<unitNo>	Integer type	Processing unit number (0 to (number of processing units of current scene minus one)) of processing unit whose registration status is to be checked.

Return value

Returns the registration status as an integer.

- 0: Processing unit not registered
- 1: Processing unit already registered

Description

Checks if the processing unit in the position in the measurement flow specified in the <unitNo> parameter has been registered.

If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.

Even if a non-existent number, numerical value, or combination of data types or values is specified for the parameter, an error will not occur.

If a value is assigned to the return value variable or the variable is not used in an expression, a "Syntax error" error will occur.

If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

Gets the processing unit title if a processing unit has been registered in processing unit number 3 in the measurement flow.

```
Rem Check the registration status of unit number 3.
If CheckUnit(3) = 1 Then
  Rem Get the title of the processing unit.
  TITLE$ = UnitTitle$(3)
Endif
```

Useable Modules

Scene Control Macro / Communication Command Macro

Supported Versions

Version 3.50 or later

Related Items

AssignUnit (Reference: ▶ Details (p.415))
DeleteUnit (Reference: ▶ Details (p.460))
MeasureStart (Reference: ▶ Details (p.617))
MoveUnit (Reference: ▶ Details (p.625))
CopyUnit (Reference: ▶ Details (p.442))
InsertUnit (Reference: ▶ Details (p.573))
MeasureStop (Reference: ▶ Details (p.619))
UnitCount (Reference: ▶ Details (p.777))

Chr$

Determining the character of the specified character code.

Format

Chr$(<expression>)

Parameter

Parameter name	Data type	Description
<expression>	Integer type	Expression to get the character (0 to 255)

Return value

Returns the character string type character.

Description

Gets the character of the ASCII character code specified in the <expression> parameter.
ASCII control codes can also be specified in the <expression> parameter.
Chr$ is the inverse function of Asc. Asc returns the character code in decimal corresponds to the specified character.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a non-existent number, numerical value, or combination of data types or values is specified for a parameter, an "Illegal function call" error will occur.
If a value outside the range -2147483648 to 2147483647 is specified as an integer parameter, an "Overflow" error will occur.
If a value is assigned to the return value variable or the variable is not used in an expression, a "Syntax error" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

Gets the characters for ASCII codes "48" and "13".

```
CHARA1$ = Chr$(48)
CHARA2$ = Chr$(13)
```

The result is shown below.

```
CHARA1$ = "0"
CHARA2$ = "CR"
```

Useable Modules

Unit Calculation Macro / Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 3.50 or later

Related Items

Asc (Reference: ▶ Details (p.413))
LCase$ (Reference: ▶ Details (p.589))
Len (Reference: ▶ Details (p.593))
Piece$ (Reference: ▶ Details (p.637))
Str$ (Reference: ▶ Details (p.749))
UCase$ (Reference: ▶ Details (p.775))
Hex$ (Reference: ▶ Details (p.560))
Left$ (Reference: ▶ Details (p.591))
Mid$ (Reference: ▶ Details (p.621))
Right$ (Reference: ▶ Details (p.664))
Str2$ (Reference: ▶ Details (p.751))
Val (Reference: ▶ Details (p.794))

ClearMeasureData

Clears the measurement results of the processing unit.

Format

ClearMeasureData [<unitNo>]

Parameter

Parameter name	Data type	Description
<unitNo>	Integer type	Processing unit number (0 to (the number of registered processing units in the current scene minus one))

Return value

None.

Description

Clears the measurement results of the processing unit specified in the <unitNo> parameter.
If the <unitNo> parameter is omitted or -1 is specified, all the measurement results of the processing units will be cleared.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a non-existent number, numerical value, or combination of data types or values is specified for a parameter, an "Illegal function call" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- Execute this macro function when the BUSY signal or other measurement in progress signal is ON and measurement is prohibited. (Reference: ▶State Transitions and Execution Timing (p.227))

Example

In the communication command macro, clears the measurement results of the search processing unit (Processing Unit number 2).

```
Rem Clear the measurement results.
ClearMeasureData 2
```

Useable Modules

Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 3.50 or later

Related Items

UnitNo (Reference: ▶ Details (p.790))

ClearScene

Clears the scene data.

Format

ClearScene <sceneNo>

Parameter

Parameter name	Data type	Description
<sceneNo>	Integer type	Scene number of scene to be cleared (0 to 127)

Return value

None.

Description

Clears the setting information in the scene with the scene number specified in the <sceneNo> parameter, and restores the scene to the factory default state.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
Even if a non-existent number, numerical value, or combination of data types or values is specified for a parameter, an "Illegal function call" error will not occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- Execute this macro function when the BUSY signal or other measurement in progress signal is ON and measurement is prohibited. (Reference: ▶State Transitions and Execution Timing (p.227))

Example

Clears the scene data of scene number 2.

```
ClearScene 2
```

Useable Modules

Scene Control Macro / Communication Command Macro

Supported Versions

Version 3.50 or later

Related Items

ChangeSceneGroup (Reference: ▶ Details (p.424))
MeasureStart (Reference: ▶ Details (p.617))
MeasureStop (Reference: ▶ Details (p.619))
SceneCount (Reference: ▶ Details (p.683))
SceneNo (Reference: ▶ Details (p.691))
SceneGroupNo (Reference: ▶ Details (p.687))

ClearSceneGroup

Clears scene group data.

Format

ClearSceneGroup <sceneGroupNo>

Parameter

Parameter name	Data type	Description
<sceneGroupNo>	Integer type	Scene group whose scene group data is to be cleared 0 to 31: scene group number 0 to 31 -1: current scene group number

Return value

None.

Description

Clears the settings of the scene group specified in the <sceneGroupNo> parameter and restores the factory settings.
If -1 is specified for the <sceneGroupNo> parameter, the settings in the current scene group will be cleared.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a non-existent number, numerical value, or combination of data types or value are specified for a parameter, an "Illegal function call" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- Execute this macro function when the BUSY signal or other measurement in progress signal is ON and measurement is prohibited. (Reference: ▶State Transitions and Execution Timing (p.227))

Example

Clears the scene group data of scene group number 1.

```
ClearSceneGroup 1
```

Useable Modules

Scene Control Macro / Communication Command Macro

Supported Versions

Version 3.50 or later

Related Items

ClearScene (Reference: ▶ Details (p.430))
MeasureStop (Reference: ▶ Details (p.619))
SceneGroupNo (Reference: ▶ Details (p.687))
MeasureStart (Reference: ▶ Details (p.617))
SceneGroupCount (Reference: ▶ Details (p.686))

Close

Closes up the file.

Format

Close [#<fileNo>[, #<fileNo>]...]

Parameter

Parameter name	Data type	Description
<fileNo>	Integer type	File number (0 to 15) of closed file

Return value

None.

Description

Close the file number specified in the <fileNo> parameter.
In the <fileNo> parameter, specify the specified file number in the Open function that has been used to open the file.
If multiple file numbers have been specified in the <fileNo> parameter, the multiple open files are closed.
If the <fileNo> parameter is omitted, all open files are closed.
If a value outside the range of 0 to 15 is specified in the <fileNo> parameter, an "Illegal function call" error will occur.
Be sure to use this macro function to close the opened file with the Open function within the same subroutine as where the Open function is used. File accessing processes such as data writing to a file and data reading from a file may not be completed properly in the following cases.

- This macro function is not executed.
- This macro function is used in a different subroutine from where the Open function is executed.
- This macro function is executed at a different timing from the Open function execution.

To access the files that have been closed by executing this macro function, execute the Open function again to open the closed file.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a value outside the range -2147483648 to 2147483647 is specified as an integer parameter, an "Overflow" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

Opens the file, writes the data in the file, and then closes the file.

```
DATA& = 10

Rem Open the file
Open "E:\input.dat" For Output As #1

Rem Write the data in the opened file
Print #1 DATA&

Rem Close the opened file
Close #1
```

Useable Modules

Unit Calculation Macro / Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 3.50 or later

Related Items

Open For Append As# (Reference: ▶ Details (p.627)) Open For Input As# (Reference: ▶ Details (p.629))
Open For Output As# (Reference: ▶ Details (p.631)) Print (Reference: ▶ Details (p.639))

CloseTextData

Close up a messages file.

Format

CloseTextData [#<textDataNo>[, #<textDataNo>]...]

Parameter

Parameter name	Data type	Description
<textDataNo>	Integer type	Text data number (0 to 15) of the closed message file

Return value

None.

Description

Close the messages file in the text data number specified in the <textDataNo> parameter.
In the <textDataNo> parameter, specify the speficied text data number in the OpenTextData function that has been used to open the message file.
If multiple text data numbers have been specified in the <textDataNo> parameter, the multiple open message files are closed.
Close all the open message file if the <textDataNo> parameter is omitted.
If a value outside the range from 0 to 15 is specified in the <textDataNo> parameter, an "Illegal function call" error will occur.
Be sure to use this macro function to close the opened file with the OpenTextData function within the same subroutine as where the OpenTextData function is used. The message file cannot properly be closed and this macro function may not properly be executed in the subsequent processes in the following cases.

- This macro function is not executed.
- This macro function is used in a different subroutine from where the OpenTextData function is executed.
- This macro function is executed at a different timing from the OpenTextData function execution.

To access the messages file that has been closed with this macro function, execute the OpenTextData function again to open the messages file.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a value outside the range -2147483648 to 2147483647 is specified as an integer parameter, an "Overflow" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

Uses the *MEASUREDISPT subroutine of the Unit Macro processing unit to display the measured correlation value by the search processing unit (Processing Unit number 5), along with the gotten text string from the prepared message file for the processing unit, in the text window. The correlation value can be gotten with External Reference Data number 5.

```
*MEASUREDISPT

  Rem Get the measurement result.
  GetUnitData 5, 5, CR#

  Rem Open the messages file
  OpenTextData "Search" As #1

  Rem Get the text
  TEXT$ = GetText$(#1, "Correlation")

  Rem Draw the gotten text string from the messages file without adding any line break on the text window.
  DrawText TEXT$, UnitJudge(5), 0

  Rem Draw the measurement results on the text window.
  DrawText Str2$(CR#, 4, 4, 0, 0), UnitJudge(5), 1

  Rem Close up the messages file.
  CloseTextData

Return
```

The result is shown below.

```
Correlation value:  90.0000
```

Useable Modules

Unit Calculation Macro / Unit Macro

Supported Versions

Version 5.00 or later

Related Items

DrawText (Reference: ▶ Details (p.500))
GetUnitData (Reference: ▶ Details (p.552))
UnitJudge (Reference: ▶ Details (p.788))
GetText$ (Reference: ▶ Details (p.548))
OpenTextData (Reference: ▶ Details (p.633))

Cont

Resumes execution of the program after it has been stopped.

Format

Cont [<mode>]

Parameter

Parameter name	Data type	Description
<mode>	Integer type	Resuming method of the stopped program This parameter can be omitted. Omitting the parameter resumes the program execution. After the resume, the program runs to the end unless an error occurs. 0: Execute the program by step-in execution. If the current program line calls a subroutine, the subroutine is entered and is executed in steps. Otherwise, the current statement is executed and the program is stopped at the next line. 1: Step-over execution If the current program line calls a subroutine, the entire subroutine is executed and the program stops at the next line after the subroutine call. Otherwise, the current statement is executed and the program is stopped at the next line. 2: Execute the program by step-out execution. If the current program line is a subroutine that was called from a subroutine, the entire subroutine after the current program line is executed, and the program stops at the next line of the subroutine that called the subroutine. Otherwise, the program is executed until it ends or an error occurs.

Return value

None.

Description

With a use of the resuming method specified in the <mode> parameter, resumes the program execution from the statement line where the program has been stopped by the Stop function execution. All the statuses before the Stop function execution is handed to the operation after resuming. (How to Use the Debug Function (p.235))

By specifying the <mode> parameter, the program can be executed in steps of one line at a time after the Stop function is executed.

If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.

Even if a non-existent number, numerical value, or combination of data types or value is specified for the parameter, an error will not occur.

If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

After the execution of the Stop function in line 220 of the Unit Macro processing unit (Processing number 1), executes the next single line (line 230)

```
Macro(U1) 220 Stop
Macro(U1) Stop in 220
Macro(U1) 230 POS.X#=(POS0.X@ + POS1.X@) / 2
Macro(U1)>Cont 1
Macro(U1)>
```

Useable Modules

Unit Calculation Macro / Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 5.20 or later

Related Items

Debug (Reference: ▶ Details (p.456))
List (Reference: ▶ Details (p.597))
SetStop (Reference: ▶ Details (p.723))
Stop (Reference: ▶ Details (p.747))
DebugPrint (Reference: ▶ Details (p.458))
Print (Reference: ▶ Details (p.639))
SetVar (Reference: ▶ Details (p.741))
VarList (Reference: ▶ Details (p.796))

CopyMeasureImage

Copies the measurement image as an image of the Unit Macro processing unit.

Format

CopyMeasureImage <measureImageNo>, <myImageNo>

Parameter

Parameter name	Data type	Description
<measureImageNo>	Integer type	Measurement image number (always 0)
<myImageNo>	Integer type	Image number of copy destination (0 to 31)

Return value

None.

Description

Copies the image with the measurement image number specified in the <measureImageNo> parameter to the image buffer of the image number specified in the <myImageNo> parameter of the Unit Macro processing unit that calls this macro function.
Normally 0 should be specified in the <measureImageNo> parameter.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a non-existent number, numerical value, or combination of data types or values are specified for a parameter, an "Illegal function call" error will occur.
If a value outside the range -2147483648 to 2147483647 is specified as an integer parameter, an "Overflow" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

Stores the measurement image of each measurement in order in the image buffer.

```
Rem Calculate the image buffer number
MYIMAGENO& = MYIMAGENO& + 1

Rem Overwrite the 32nd and following images in order from the oldest image.
If MYIMAGENO& > 31 Then
  MYIMAGENO& = 0
Endif

Rem Store the measurement image in the image buffer of the Unit Macro processing unit.
CopyMeasureImage 0, MYIMAGENO&
```

Useable Modules

Unit macro

Supported Versions

Version 3.50 or later

Related Items

CopyUnitImage (Reference: ▶ Details (p.446))

SetMeasureImage (Reference: ▶ Details (p.707))

CopyScene

Copies scene data.

Format

CopyScene <srcSceneNo>, <destSceneNo>

Parameter

Parameter name	Data type	Description
<srcSceneNo>	Integer type	Scene number (0 to 127) of the scene to be copied.
<destSceneNo>	Integer type	Scene number (0 to 127) of destination scene.

Return value

None.

Description

Copies the scene data of the scene number specified in the <srcSceneNo> parameter to the scene data of the scene number specified in the <destSceneNo> parameter.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If the same scene number is specified in the <destSceneNo> parameter as the <srcSceneNo> parameter, an "Illegal function call" error will occur.
If a non-existent number, numerical value, or combination of data types or values are specified for a parameter, an "Illegal function call" error will occur.
If there is insufficient free working memory to copy the data, an "Illegal function call" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- Execute this macro function when the BUSY signal or other measurement in progress signal is ON and measurement is prohibited. (Reference: ▶State Transitions and Execution Timing (p.227))

Example

Copies the data of scene 2 to scene 3.

```
CopyScene 2, 3
```

Useable Modules

Scene Control Macro / Communication Command Macro

Supported Versions

Version 3.50 or later

Related Items

CopySceneGroup (Reference: ▶ Details (p.441))
MeasureStart (Reference: ▶ Details (p.617))
MeasureStop (Reference: ▶ Details (p.619))
SceneCount (Reference: ▶ Details (p.683))
SceneGroupNo (Reference: ▶ Details (p.687))
SceneNo (Reference: ▶ Details (p.691))

CopySceneGroup

Copies scene group data.

Format

CopySceneGroup <srcSceneGroupNo>, <destSceneGroupNo>

Parameter

Parameter name	Data type	Description
<srcSceneGroupNo>	Integer type	Scene group number (0 to 31) of the scene group to be copied.
<destSceneGroupNo>	Integer type	Scene group number (0 to 31) of the destination scene group.

Return value

None.

Description

Copies the scene group data of the scene group number specified in the <srcSceneGroupNo> parameter to the scene group data of the scene group specified in the <destSceneGroupNo> parameter
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If the same scene number is specified in the <destSceneNo> parameter as the <srcSceneNo> parameter, an "Illegal function call" error will occur.
If a non-existent number, numerical value, or combination of data types or values are specified for a parameter, an "Illegal function call" error will occur.
If there is insufficient free working memory to copy the data, an "Illegal function call" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- Execute this macro function when the BUSY signal or other measurement in progress signal is ON and measurement is prohibited. (Reference: ▶State Transitions and Execution Timing (p.227))

Example

Copies the data of scene group 0 to scene group 1.

```
CopySceneGroup 0, 1
```

Useable Modules

Scene Control Macro / Communication Command Macro

Supported Versions

Version 3.50 or later

Related Items

CopyScene (Reference: ▶ Details (p.440))
MeasureStart (Reference: ▶ Details (p.617))
MeasureStop (Reference: ▶ Details (p.619))
SceneCount (Reference: ▶ Details (p.683))
SceneGroupNo (Reference: ▶ Details (p.687))
SceneNo (Reference: ▶ Details (p.691))

CopyUnit

Copies a processing unit.

Format

CopyUnit [<srcSceneNo>,] <srcUnitNo>, <destUnitNo>, <mode>

Parameter

Parameter name	Data type	Description
<srcSceneNo>	Integer type	Scene number of the origin of copy (0 to 127)
<srcUnitNo>	Integer type	Processing unit number of copy source (0 to (number of scene processing units of copy source minus one))
<destUnitNo>	Integer type	Processing unit number that is to copy to (0 to the enrollment number of the processing units at the current scene -1)
<mode>	Integer type	Processing mode 0: Overwrites the processing unit of the processing unit number of the copy destination 1: Insert the loaded processing unit in front of the processing unit number.

Return value

None.

Description

Copies the processing unit specified in the <srcUnitNo> parameter of the scene number specified in the <srcSceneNo> parameter to the measurement flow position specified in the <destUnitNo> parameter of the current scene, using the mode specified in the <mode> parameter. If the <srcSceneNo> parameter is omitted, the copy source scene is the current scene.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a non-existent number, numerical value, or combination of data types or values are specified for a parameter, an "Illegal function call" error will occur.
When 0 is specified in the <mode> parameter, the processing unit that has the processing unit number of the copy destination is overwritten by the copy source processing unit. When 1 is specified in the <mode> parameter, the processing unit of the copy source is inserted in the processing unit number of the copy destination, and the processing unit number of the processing unit of the copy destination is moved down by 1.
When 0 is specified in the <mode> parameter, specify different values in the <srcUnitNo> parameter and <destUnitNo> parameter. If the same value is specified, an "Illegal function call" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- Execute this macro function when the BUSY signal or other measurement in progress signal is ON and measurement is prohibited. (Reference: ▶State Transitions and Execution Timing (p.227))

Example

Copies processing unit 3 of scene 2 and inserts in front of processing unit 4 of the current scene.

```
CopyUnit 2, 3, 4, 1
```

Useable Modules

Scene Control Macro / Communication Command Macro

Supported Versions

Version 3.50 or later

Related Items

AssignUnit (Reference: ▶ Details (p.415))
DeleteUnit (Reference: ▶ Details (p.460))
MeasureStart (Reference: ▶ Details (p.617))
MoveUnit (Reference: ▶ Details (p.625))
CheckUnit (Reference: ▶ Details (p.425))
InsertUnit (Reference: ▶ Details (p.573))
MeasureStop (Reference: ▶ Details (p.619))
UnitCount (Reference: ▶ Details (p.777))

CopyUnitFigure

Copies figure data to the processing unit.

Format

CopyUnitFigure <srcSceneNo>, <srcUnitNo>, <srcFigureNo>, <destUnitNo>, <destFigureNo>

Parameter

Parameter name	Data type	Description
<srcSceneNo>	Integer type	Scene number of the origin of copy (0 to 127)
<srcUnitNo>	Integer type	Unit number of copy source (0 to (number of scene processing units of copy source minus one))
<srcFigureNo>	Integer type	Figure number of copy source (Reference: ▶Figure Data List (p.364))
<destUnitNo>	Integer type	Unit number of copy destination (0 to (number of scene processing units of copy destination minus one))
<destFigureNo>	Integer type	Figure number of copy destination (Reference: ▶Figure Data List (p.364))

Return value

None.

Description

For the scene specified in the <srcSceneNo> parameter, copies the figure data of the figure number specified in the <srcFigureNo> parameter, of the processing unit specified in the <srcUnitNo> parameter, to the figure data of the figure number specified in the <destFigureNo> parameter, of the processing unit specified in the <destUnitNo> parameter.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a non-existent number, numerical value, or combination of data types or values are specified for a parameter, an "Illegal function call" error will occur.
If a value outside the range -2147483648 to 2147483647 is specified as an integer parameter, an "Overflow" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- Execute this macro function when the BUSY signal or other measurement in progress signal is ON and measurement is prohibited. (Reference: ▶State Transitions and Execution Timing (p.227))
- Use this macro function with the measurement image displayed after one or more measurements, or after the image file is specified and re-measured.
- Set the figure data so that pixels from outside the image are not included in the figure.

Example

Copies the measurement region figure of the Shape Search III processing unit of Processing Unit number 2 in the measurement flow of scene 0 to the measurement region of the Shape Search III processing unit of Processing Unit number 5. The measurement region figure of the Shape Search III processing item is figure 1.

```
Rem Copy the figure of the measurement region
CopyUnitFigure 0, 2, 1, 5, 1
```

Useable Modules

Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 3.50 or later

Related Items

GetUnitFigure (Reference: ▶ Details (p.554))
MeasureStop (Reference: ▶ Details (p.619))
SetUnitFigure (Reference: ▶ Details (p.733))
Ut (Reference: ▶ Details (p.793))
MeasureStart (Reference: ▶ Details (p.617))
SceneNo (Reference: ▶ Details (p.691))
UnitNo (Reference: ▶ Details (p.790))

CopyUnitImage

Copies a processing unit image as a unit macro processing unit image.

Format

CopyUnitImage <unitNo>, <imageNo>, <myImageNo>

Parameter

Parameter name	Data type	Description
<unitNo>	Integer type	Processing unit number that is to be copied (0 to (the number of registered processing units in the current scene -1))
<imageNo>	Integer type	Image number of copy source (Reference: ▶Image Number List (p.371))
<myImageNo>	Integer type	Image number of copy destination (0 to 31)

Return value

None.

Description

Copies the image with the image number specified in the <ImageNo> parameter, of the processing unit specified in the <unitNo> parameter, to the image buffer of the unit macro processing unit that calls this macro function with the image number specified in the <myImageNo> parameter.
Normally 0 should be specified in the <imageNo> parameter.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a non-existent number, numerical value, or combination of data types or values are specified for a parameter, an "Illegal function call" error will occur.
If a value outside the range -2147483648 to 2147483647 is specified as an integer parameter, an "Overflow" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

None.

Example

Stores the image of each measurement in order in the image buffer after filtering by the color gray filter processing unit of processing unit number 1.

```
Rem Calculate the image buffer number
MYIMAGENO& = MYIMAGENO& + 1

Rem Overwrite the 32nd and following images in order from the oldest image.
If MYIMAGENO& > 31 Then
  MYIMAGENO& = 0
Endif

Rem Store the filtered image in the image buffer of the unit macro processing unit.
CopyUnitImage 1, 0, MYIMAGENO&
```

Useable Modules

Unit macro

Supported Versions

Version 3.50 or later

Related Items

CopyMeasureImage (Reference: ▶ Details (p.438))
SetMeasureImage (Reference: ▶ Details (p.707))
UnitNo (Reference: ▶ Details (p.790))
Ut (Reference: ▶ Details (p.793))

CopyUnitModel

Copies the model data of a processing unit.

Format

CopyUnitModel <srcSceneNo>, <srcUnitNo>, <srcModelNo>, <destUnitNo>, <destModelNo>

Parameter

Parameter name	Data type	Description
<srcSceneNo>	Integer type	Scene number of the origin of copy (0 to 127)
<srcUnitNo>	Integer type	Unit number of copy source (0 to (number of scene processing units of copy source minus one))
<srcModelNo>	Integer type	Model number of copy source (Reference: ▶Model Number List (p.369))
<destUnitNo>	Integer type	Unit number of copy destination (0 to (number of scene processing units of copy destination minus one))
<destModelNo>	Integer type	Model number of copy destination (Reference: ▶Model Number List (p.369))

Return value

None.

Description

Copies the model data of the model number specified in the <srcModelNo> parameter, of the processing unit specified in the <srcUnitNo> parameter, of the scene specified in the <srcSceneNo> parameter, to the model data of the model number specified in the <destModelNo> parameter, of the processing unit specified in the <destUnitNo> parameter.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a non-existent number, numerical value, or combination of data types or values are specified for a parameter, an "Illegal function call" error will occur.
If a value outside the range -2147483648 to 2147483647 is specified as an integer parameter, an "Overflow" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.
On models that use a processing item for measurement, there is also the model data, model figure, model parameter, and other information. The model information varies by processing item, and correct operation does not always result from simply copying the model data, but it is also possible to use a variant-type variable in the scene control macro to copy model data. For details, refer to Application Producer (FH-AP1, sold separately).

Usage Cautions

- Execute this macro function when the BUSY signal or other measurement in progress signal is ON and measurement is prohibited. (Reference: ▶State Transitions and Execution Timing (p.227))
- Use this macro function with the measurement image displayed after one or more measurements, or after the image file is specified and re-measured.
- Set the figure data so that pixels from outside the image are not included in the figure.

Example

Copies the Shape Search III processing unit model of Processing Unit number 2 to the Shape Search III processing unit of Processing Unit number 3.

```
Rem Copy the model figure of Shape Search III.
CopyUnitFigure 0, 2, 0, 3, 0

Rem Copy the detection point setting of Shape Search III
GetUnitData 2, "detectionPosX", PosX#
GetUnitData 2, "detectionPosY", PosY#
SetUnitData 3, "detectionPosX", PosX#
SetUnitData 3, "detectionPosY", PosY#

Rem Copy the model data of Shape Search III.
CopyUnitModel 0, 2, 0, 3, 0
```

Useable Modules

Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 3.50 or later

Related Items

CopyUnitFigure (Reference: ▶ Details (p.444))
GetUnitData (Reference: ▶ Details (p.552))
MeasureStart (Reference: ▶ Details (p.617))
MeasureStop (Reference: ▶ Details (p.619))
SceneNo (Reference: ▶ Details (p.691))
SetUnitData (Reference: ▶ Details (p.731))
UnitNo (Reference: ▶ Details (p.790))
Ut (Reference: ▶ Details (p.793))

Cos

Gets the cosine of the specified expression.

Format

Cos(<expression>)

Parameter

Parameter name	Data type	Description
<expression>	Integer type Double precision real number data type	Expression to calculate the cosine

Return value

Returns the cosine as a double precision real value in the range -1 to 1.

Description

Gets the cosine of the expression specified in the <expression> parameter.
In the <expression> parameter, specify the value in radians. To convert an angle value to a radian value, multiply by π/180.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a value is assigned to the return value variable or the variable is not used in an expression, a "Syntax error" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

Gets the cosine of 60°.

```
DATA# = Cos(60/180*3.141592)
```

The result is shown below.

```
DATA# = 0.5
```

Useable Modules

Unit Calculation Macro / Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 3.50 or later

Related Items

Atn (Reference: ▶ Details (p.417))
Sin (Reference: ▶ Details (p.743))
UnitData (Reference: ▶ Details (p.778))
GetUnitData (Reference: ▶ Details (p.552))
Tan (Reference: ▶ Details (p.755))

Crspoint

Gets the intersection between 2 straight lines.

Format

Crspoint <line1()>, <line2()>, <x>, <y>

Parameter

Parameter name	Data type	Description
<line1()>	Double precision real number array	Straight line component of the straight line 1 to get the intersection
<line2()>	Double precision real number array	Straight line component of the straight line 2 to get the intersection
<x>	Double precision real number data type	Variable that holds the X coordinate of the intersection gotten
<y>	Double precision real number data type	Variable that holds the Y coordinate of the intersection gotten

Return value

None.

Description

Gets the intersection point of the line component specified in the <line1()> parameter and the line component specified in the <line2()> parameter. In the <x> parameter and <y> parameter, specify the respective variables that will hold the X coordinate and Y coordinate of the intersection point.
In the <line1()> parameter and in the <line2()> parameter, specify a 1D double precision real number array with "a" in element 0, "b" in element 1, and "c" in element 2, where a, b, and c satisfy the linear equation ax + by + c = 0 , without adding element numbers but adding () to the variables.
This macro function is mainly used to get the intersection point of lines gotten with the Lsqumeth function.
If an undefined array is specified a parameter, an "Undefined label" error will occur.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a non-existent number, numerical value, or combination of data types or values are specified for a parameter, an "Illegal function call" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

Gets the intersection point of two gotten lines. The two lines are respectively gotten using Processing Units 1 to 4 and Processing Units 5 to 8.

```
Dim POS1X#(3), POS1Y#(3), POS2X#(3), POS2Y#(3), PARAM1#(2), PARAM2#(2)

Rem Rem Initialize variables for straight line 1
For I&=0 To 3
  GetUnitData I&+1, "X", POS1X#(I&)
  GetUnitData I&+1, "Y", POS1Y#(I&)
Next

Rem Get the straight line 1st component.
Lsqumeth 4, POS1X#(), POS1Y#(), PARAM1#()

Rem Rem Initialize variables for straight line 2
For I&=0 To 3
  GetUnitData I&+5, "X", POS2X#(I&)
  GetUnitData I&+5, "Y", POS2Y#(I&)
Next

Rem Get the straight line 2nd component.
Lsqumeth 4, POS2X#(), POS2Y#(), PARAM2#()

Rem Get the intersection between 2 straight lines.
Crspoint PARAM1#(), PARAM2#(), CRSX#, CRSY#
Erase POS1X#(), POS1Y#(), POS2X#(), POS2Y#(), PARAM1#(), PARAM2#()
```

Useable Modules

Unit Calculation Macro / Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 3.50 or later

Related Items

Erase (Reference: ▶ Details (p.512))
Lsqumeth (Reference: ▶ Details (p.610))
GetUnitData (Reference: ▶ Details (p.552))
UnitData (Reference: ▶ Details (p.778))

Date$

Reads out the date from the internal clock.

Format

Date$

Parameter

None.

Return value

Returns the date as a character string value.
The date value is a character string of the internal clock date whose year (YY), month (MM), and day (DD) separated by a slash (/). The range of each is indicated below.

- Year (YY): 00 to 80
- Month (MM): 01 to 12
- Day (DD): 01 to 31

Description

Reads the date from the internal clock and returns the date value (YY/MM/DD) in character string format.
The year is expressed as a value from 00 to 80, representing 2000 to 2080.
The internal clock can be adjusted in [Date-time Settings] under [System settings]. (Reference: ▶Setting the Date and Time: [Date-time Setting] (p.291))
If a value is assigned to the return value variable or the variable is not used in an expression, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

Reads the date in the internal clock and outputs the date to the system status console window.

```
Rem Read out the date from the internal clock.
TODAY$ = Date$

Rem Output the read date to the system status console window.
Print "20";TODAY$
```

The result is shown below.

```
2011/03/10
```

Useable Modules

Unit Calculation Macro / Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 3.50 or later

Related Items

GetSystemData (Reference: ▶ Details (p.546))
Piece$ (Reference: ▶ Details (p.637))
SetSystemData (Reference: ▶ Details (p.725))
Mid$ (Reference: ▶ Details (p.621))
Print (Reference: ▶ Details (p.639))
Time$ (Reference: ▶ Details (p.757))

Debug

Set the program execution form and information output method.

Format

Debug <mode>

Parameter

Parameter name	Data type	Description
<mode>	Integer type	Execution form and information output method 0: Release mode, no error description is output when an error occurs 1: Release mode, an error description is output to the system status console window when an error occurs. 2: Release mode, the contents of each line are output to the system status console window when the program is executed. 3: Release mode, an error description is output to the message box when an error occurs. 16: Debug mode, no error description is output when an error occurs 17: Debug mode, an error description is output to the system status console window when an error occurs. 18: Debug mode, the contents of each line are output to the system status console window when the program is executed. 19: Debug mode, an error description is output to the error box when an error occurs.

Return value

None.

Description

Sets the program execution form and information output method specified in the <mode> parameter.
Debug mode can be set to debug the program using macro functions that are only executed in debug mode. Set the mode to release mode after the debug so that there will be no need of removing DebugPring functions and other macro functions that are only used in debug mode from the program. (How to Use the Debug Function (p.235))
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a non-existent number, numerical value, or combination of data types or values are specified for a parameter, an "Illegal function call" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

Uses the MCRINIT subroutine in the unit macro processing unit to set the program execution form to "debug mode" and information output method so as to output error descriptions to the system status console window at an error occurrence.

```
*MCRINIT
  Rem Output an error description to the system status console window when an error occurs in debug mode.
  Debug 17
Return
```

Useable Modules

Unit Calculation Macro / Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 5.20 or later

Related Items

Cont (Reference: ▶ Details (p.436))
List (Reference: ▶ Details (p.597))
SetStop (Reference: ▶ Details (p.723))
Stop (Reference: ▶ Details (p.747))
DebugPrint (Reference: ▶ Details (p.458))
Print (Reference: ▶ Details (p.639))
SetVar (Reference: ▶ Details (p.741))
VarList (Reference: ▶ Details (p.796))

DebugPrint

Outputs debug information to the system status console window.

Format

DebugPrint <expression>[;|, <expression>...]

Parameter

Parameter name	Data type	Description
<expression>	Integer type Double precision real number data type Character string type Array	Numerical expression or character string to be output

Return value

None.

Description

Outputs the numerical expression or character string specified in the <expression> parameter to the system status console window. (Reference: ▶Description of the System Status Console Window (p.185))
If an non-existent number, numerical value, or combination of data types or values are specified for a parameter, an "Illegal function call" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- This macro function is only enabled when specified in debug mode with the Debug function. Specifying other values than the range above will treat the statement with this function in the same manner with the Rem function (i.e., ignores the statement). (Reference: ▶How to Use the Debug Function (p.235))
- After the data output to the system status console window, the window is displayed on top of the Sensor Controller main screen. To display the system status console window on top of the main screen, click [_] on the upper-right of the system status console window or press [Alt] + [Tab] on the connected USB keyboard to the sensor controller.

Example

Outputs a debug information (character string) to the system status console window in debug mode.

```
Rem Set the execution form to debug mode.
Debug 18

Rem Output character string "Result = OK" as the debug information
DebugPrint "Result = " + "OK"

Rem Set the execution form to release mode.
Debug 1
```

Useable Modules

Unit Calculation Macro / Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 5.20 or later

Related Items

Cont (Reference: ▶ Details (p.436))
List (Reference: ▶ Details (p.597))
SetStop (Reference: ▶ Details (p.723))
Stop (Reference: ▶ Details (p.747))
Debug (Reference: ▶ Details (p.456))
Print (Reference: ▶ Details (p.639))
SetVar (Reference: ▶ Details (p.741))
VarList (Reference: ▶ Details (p.796))

DeleteUnit

Deletes a processing unit.

Format

DeleteUnit <unitNo>

Parameter

Parameter name	Data type	Description
<unitNo>	Integer type	Processing unit number (0 to (number of processing units of current scene minus one)) of processing unit to be deleted.

Return value

None.

Description

Deletes the processing unit specified in the <unitNo> parameter of the current scene from the measurement flow.
The processing unit numbers of processing units after the processing unit number specified in the <unitNo> parameter are moved up by 1.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a non-existent number, numerical value, or combination of data types or values are specified for a parameter, an "Illegal function call" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- Execute this macro function when the BUSY signal or other measurement in progress signal is ON and measurement is prohibited. (Reference: ▶State Transitions and Execution Timing (p.227))

Example

Deletes the processing unit of unit number 2.

```
DeleteUnit 2
```

Useable Modules

Scene Control Macro / Communication Command Macro

Supported Versions

Version 3.50 or later

Related Items

AssignUnit (Reference: ▶ Details (p.415))
CopyUnit (Reference: ▶ Details (p.442))
MeasureStart (Reference: ▶ Details (p.617))
MoveUnit (Reference: ▶ Details (p.625))
CheckUnit (Reference: ▶ Details (p.425))
InsertUnit (Reference: ▶ Details (p.573))
MeasureStop (Reference: ▶ Details (p.619))
UnitCount (Reference: ▶ Details (p.777))

Dim

Defines the array variable.

Format

Dim <arrayName>(<maxCount>[, <maxCount>[, <maxCount>[, <maxCount>]]])

Parameter

Parameter name	Data type	Description
<arrayName>	---	Used array variable name
<maxCount>	Integer type	Maximum value of the subscript

Return value

None.

Description

Defines a 1D to 4D array with maximum dimensional length specified in the <maxCount> parameter for each dimension.
Add one of type identifiers to the end of the parameter specified in the <arrayName>. (Reference: ▶Variable (p.209))
Release the array variables defined with this macro function by executing the Erase function.
If the number of array dimension is different, two arrays with the same variable name are treated as the same variable.
An array variable and a variable with the same name are treated as different variables.
If a defined array variable is redefined, the previously defined variable is released before redefinition.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

Defines the array.

```
Dim XY&(3)
Dim XY#(7, 15)
Dim CHARA$(31, 63, 127, 255)
```

Useable Modules

Unit Calculation Macro / Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 3.50 or later

Related Items

Erase (Reference: ▶ Details (p.512))

DisplaySubNo

Get the sub-image number of the displayed sub-image.

Format

DisplaySubNo

Parameter

None.

Return value

Returns the sub-image number as an integer value.

Description

Gets the sub-image number of the displayed sub-image set in the image window on the main screen.
If a value is assigned to the return value variable or the variable is not used in an expression, a "Syntax error" error will occur.

Usage Cautions

- This macro function can only be used in the *MEASUREDISPI subroutine or the *MEASUREDISPG subroutine. If used in another subroutine, an "Illegal function call" error will occur.

Example

Uses the *MEASUREDISPG subroutine in the unit macro processing unit to change the display in the image window according to the set image display sub-number in the image window of the main screen.

```
*MEASUREDISPG

  Rem Get the displayed sub-image number
  SUBNO& = DisplaySubNo

  Rem Change the display on the image window according to the sub-image number of the sub-image to be displayed.
  Select SUBNO&
  Case 1
    Rem If the gotten sub-image number is 1, the title of processing unit 1 is displayed with the color in accordance with the judgment result.
    SetTextStyle 24, TA_LEFT, UnitJudge(1), 0, FONTSTYLE_NORMAL
    TEXT$ = UnitTitle$(1)
  Case 2
    Rem If the gotten sub-image number is 2, the title of processing unit 2 is displayed with the color in accordance with the judgment result.
    SetTextStyle 24, TA_LEFT, UnitJudge(2), 0, FONTSTYLE_NORMAL
    TEXT$ = UnitTitle$(2)
  Case Else
    Rem If the gotten sub-image number is other than 1 and 2, "Error" is displayed in the "unmeasured" color.
    SetTextStyle 24, TA_LEFT, JUDGE_NC, 0, FONTSTYLE_NORMAL
    TEXT$ = "Error"
  End Select

  Rem Displays text on the image window.
  DrawTextG TEXT$, 50, 0, 0, UnitNo
```

Return

Useable Modules

Unit macro

Supported Versions

Version 3.50 or later

Related Items

DrawTextG (Reference: ▶ Details (p.502))
UnitJudge (Reference: ▶ Details (p.788))
UnitTitle$ (Reference: ▶ Details (p.791))
SetTextStyle (Reference: ▶ Details (p.727))
UnitNo (Reference: ▶ Details (p.790))

DisplayUnitNo

Gets the selection state of the processing unit number of the flow window.

Format

DisplayUnitNo

Parameter

None.

Return value

Returns the processing unit number as an integer value.

Description

Gets the processing unit number of the unit selected in the flow window.
If a value is assigned to the return value variable or the variable is not used in an expression, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

In the communication command macro, gets the processing unit number selected in the flow window.

```
Rem Get the processing unit number selected in the flow window.
NO& = DisplayUnitNo

Rem Set the processing unit number in the response data of the communication command.
ResponseString$ = Str$(NO&)
CommandResponse& = 0
```

Useable Modules

Scene Control Macro / Communication Command Macro

Supported Versions

Version 3.50 or later

Related Items

GetImageWindow (Reference: ▶ Details (p.533))
SetDisplayUnitNo (Reference: ▶ Details (p.699))
UnitNo (Reference: ▶ Details (p.790))
Ut (Reference: ▶ Details (p.793))

Do Loop While

Repeatedly executes the statements between Do and Loop while the specified condition meets.

Format

Do
<statement>
Loop While <expression>

Parameter

Parameter name	Data type	Description
<statement>	---	Statement to be executed repeatedly
<expression>	---	Conditional logical expression for which gets a repetition of operation (Reference: ▶Operator (p.213))

Return value

None.

Description

The statement is repeatedly executed if the specified logical expression by the <expression> parameter is true as a result of the specified Do block execution by the <statement> parameter.
If the Exit Do statement is used in the Do block statement, the statement force stops the repeating execution of the program immediately.
If the program process is jumped into or out of the Do block statement using the Goto or Gosub function, unexpected operation may occur.
If neither the Do statement nor the Loop While statement is used, either the "DO without LOOP", "LOOP without DO", or "EXIT without DO" error will occur depending on the statement that is used.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

Repeats the process until the loop counter reaches a constant value.

```
NUM& = 0

Rem Repeat the process
Do
  NUM& = NUM& + 1
Loop While NUM& < 100

Print NUM&
```

Useable Modules

Unit Calculation Macro / Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 3.50 or later

Related Items

For To Step Next (Reference: ▶ Details (p.526))

Print (Reference: ▶ Details (p.639))

Dposline

Gets the shortest distance between the line and point.

Format

Dposline (<x>, <y>, <line()>)

Parameter

Parameter name	Data type	Description
<x>	Double precision real number data type	X coordinate of the points to get the distance
<y>	Double precision real number data type	Y coordinate of the points to get the distance
<line()>	Double precision real number array	Straight line component of the straight line to get the distance

Return value

Returns the shortest distance as a double precision real number value.

Description

Gets the shortest distance between the point that has the X coordinate specified in the <x> parameter and the Y coordinate specified in the <y> parameter, and the line component specified in the <line()> parameter.
In the <line()> parameter, specify a 1D double precision real number array with "a" in element 0, "b" in element 1, and "c" in element 2, where a, b, and c satisfy the linear equation ax + by + c = 0, without adding element numbers but adding () to the variables.
This macro function is mainly used to get the deviation and distribution from an origin point for a line gotten with the Lsqumeth function.
If an undefined array is specified a parameter, an "Undefined label" error will occur.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a non-existent number, numerical value, or combination of data types or values are specified for a parameter, an "Illegal function call" error will occur.
If a value is assigned to the return value variable or the variable is not used in an expression, a "Syntax error" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

Gets the distribution and deviation for a line gotten from four points.

```
Dim POSX#(3), POSY#(3), PARAM#(2), DIST#(3)

Rem Initialize variables for straight line
For I&=0 To 3
  GetUnitData I&+1, "X", POSX#(I&)
  GetUnitData I&+1, "Y", POSY#(I&)
Next

Rem Get the straight line component.
Lsqumeth 4, POSX#(), POSY#(), PARAM#()

SUMDIST# = 0
For I&=0 To 3
  Rem Calculate the shortest distance between the straight line and point.
  DIST#(I&) = Dposline(POSX#(I&), POSY#(I&), PARAM#())
  SUMDIST# = SUMDIST# + DIST#(I&)
Next

Erase POSX#(), POSY#(), PARAM#(), DIST#()
```

Useable Modules

Unit Calculation Macro / Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 3.50 or later

Related Items

Erase (Reference: ▶ Details (p.512))
Lsqumeth (Reference: ▶ Details (p.610))
GetUnitData (Reference: ▶ Details (p.552))
UnitData (Reference: ▶ Details (p.778))

DrawArc

Draw the arc on the image window.

Format

DrawArc <x>, <y>, <radius>, <start>, <end>, <imageNo>[, <unitNo>]

Parameter

Parameter name	Data type	Description
<x>	Integer type	The center X coordinate of the drawn arc
<y>	Integer type	The center Y coordinate of the drawn arc
<radius>	Integer type	Radius of the drawn arc
<start>	Integer type	Starting angle of the drawn arc
<end>	Integer type	Ending angle of the drawn arc
<imageNo>	Integer type	Measurement image number to draw on (always 0)
<unitNo>	Integer type	Processing unit number to display the processing unit (0 to (the number of registered processing units in the current scene minus one))

Return value

None.

Description

On the measurement image whose specified image number is in the <imageNo> parameter, draw an arc of the specified radius by the <radius> parameter and whose starting angle and ending angle are specified in the <start> and <end> parameters respectively at the center coordinates specified in the <x> and <y> parameters.
Specify the corresponding Unit Macro processing unit number in the <unitNo> parameter to draw the image at the position coordinates before applying position compensation. If the <unitNo> parameter is omitted, the image is drawn at the position coordinates after applying position compensation.
In the <X> and <Y> parameters, specify the camera coordinates whose origin is at the upper-left corner of the image.
In the <start> and <end> parameters, specify the angle so that the angle increases in a crockwise direction respect to the positive X-axis of the camera coordinates.
Normally 0 should be specified in the <imageNo> parameter.
In the <unitNo> parameter, normally specify the processing unit number of the Unit Macro processing unit that executes this process.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a non-existent number, numerical value, or combination of data types or values are specified for a parameter, an "Illegal function call" error will occur.
If a value outside the range -2147483648 to 2147483647 is specified as an integer parameter, an "Overflow" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- This macro function can only be used in the *MEASUREDISPI subroutine or the *MEASUREDISPG subroutine. If used in another subroutine, an "Illegal function call" error will occur.

Example

Uses the *MEASUREDISPG subroutine of the Unit Macro processing unit to draw an arc of the measured radius whose starting angle is -90° and ending angle is 180° at the measured center coordinates by the circular scan edge position processing unit (Processing Unit number 5). The measured X and Y coordinates and radius can be gotten with External Reference Data numbers 5 to 7 respectively.
To display the string at the fixed position regardless of the position compensation result, specify the assigned processing unit number to this Unit Macro processing unit (where the *MEASUREDISPG subroutine is used) for the <unitNo> parameter.

```
*MEASUREDISPG

  Rem Get the measurement result.
  GetUnitData 5, 5, X#
  GetUnitData 5, 6, Y#
  GetUnitData 5, 7, R#

  Rem Draw the image
  DrawArc Int(X#), Int(Y#), Int(R#), -90, 180, 0, UnitNo

Return
```

Useable Modules

Unit macro

Supported Versions

Version 3.50 or later

Related Items

GetUnitData (Reference: ▶ Details (p.552))
SetDrawStyle (Reference: ▶ Details (p.700))
UnitNo (Reference: ▶ Details (p.790))
Int (Reference: ▶ Details (p.575))
UnitData (Reference: ▶ Details (p.778))
Ut (Reference: ▶ Details (p.793))

DrawArcW

Draw the wide arc on the image window.

Format

DrawArc <x>, <y>, <radius>, <start>, <end>, <imageNo>[, <unitNo>]

Parameter

Parameter name	Data type	Description
<x>	Integer type	The center X coordinate of the drawn wide arc
<y>	Integer type	The center Y coordinate of the drawn wide arc
<radius>	Integer type	Radius of the drawn wide arc
<start>	Integer type	Starting angle of the drawn wide arc
<end>	Integer type	Ending angle of the drawn wide arc
<width>	Integer type	Width of the drawn wide arc
<imageNo>	Integer type	Measurement image number to draw on (always 0)
<unitNo>	Integer type	Processing unit number to display the processing unit (0 to (the number of registered processing units in the current scene minus one))

Return value

None.

Description

On the measurement image whose specified image number is in the <imageNo> parameter, draw a wide arc of whose radius, width, starting angle, and ending angle are specified in the <radius>, <width>, <start>, and <end> parameters respectively at the center coordinates specified in the <x> and <y> parameters.
Specify the corresponding Unit Macro processing unit number in the <unitNo> parameter to draw the image at the position coordinates before applying position compensation. If the <unitNo> parameter is omitted, the image is drawn at the position coordinates after applying position compensation.
In the <X> and <Y> parameters, specify the camera coordinates whose origin is at the upper-left corner of the image.
In the <start> and <end> parameters, specify the angle so that the angle increases in a crockwise direction respect to the positive X-axis of the camera coordinates.
Normally 0 should be specified in the <imageNo> parameter.
In the <unitNo> parameter, normally specify the processing unit number of the Unit Macro processing unit that executes this process.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a non-existent number, numerical value, or combination of data types or values are specified for a parameter, an "Illegal function call" error will occur.
If a value outside the range -2147483648 to 2147483647 is specified as an integer parameter, an "Overflow" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- This macro function can only be used in the *MEASUREDISPI subroutine or the *MEASUREDISPG subroutine. If used in another subroutine, an "Illegal function call" error will occur.

Example

Uses the *MEASUREDISPG subroutine of the Unit Macro processing unit to draw an arc of the measured radius whose starting angle is -90° and ending angle is 180° at the measured center coordinates by the circular scan edge position processing unit (Processing Unit number 5). The displayed arc is a wide arc whose outer radius is the measured maximum radius and whose inner radius is the measured minimum radius. The measured X/Y coordinates and radius, and maximum/minimum radii can be gotten with External Reference Data numbers 5, 6, 7, 8, and 9 respectively.
To display the arc at the fixed position regardless of the position compensation result, specify the assigned processing unit number to this Unit Macro processing unit (where the *MEASUREDISPG subroutine is used) for the <unitNo> parameter.

```
*MEASUREDISPG

  Rem Get the measurement result.
  GetUnitData 5, 5, X#
  GetUnitData 5, 6, Y#
  GetUnitData 5, 7, R#
  GetUnitData 5, 8, R_MAX#
  GetUnitData 5, 9, R_MIN#

  Rem Determine the width
  W# = R_MAX# - R_MIN#

  Rem Draw the image
  DrawArcW Int(X#),Int(Y#), Int(R#), -90, 180, Int(W#), 0, UnitNo

Return
```

Useable Modules

Unit macro

Supported Versions

Version 3.50 or later

Related Items

GetUnitData (Reference: ▶ Details (p.552))
Int (Reference: ▶ Details (p.575))
SetDrawStyle (Reference: ▶ Details (p.700))
UnitData (Reference: ▶ Details (p.778))
UnitNo (Reference: ▶ Details (p.790))
Ut (Reference: ▶ Details (p.793))

DrawBox

Draws a rectangle on the image window.

Format

DrawBox <x0>, <y0>, <x1>, <y1>, <imageNo>[, <unitNo>]

Parameter

Parameter name	Data type	Description
<x0>	Integer type	The upper-left corner X coordinate of the drawn rectangle
<y0>	Integer type	The upper-left corner Y coordinate of the drawn rectangle
<x1>	Integer type	The lower-right corner X coordinate of the drawn rectangle
<y1>	Integer type	The lower-right corner Y coordinate of the drawn rectangle
<imageNo>	Integer type	Measurement image number to draw on (always 0)
<unitNo>	Integer type	Processing unit number that executes the drawing processing (0 to (the number of registered processing units in the current scene minus one))

Return value

None.

Description

On the measurement image whose specified image number is in the <imageNo> parameter, draws a rectangle that has the specified upper left vertex coordinates by the <x0> and <y0> parameters and the specified lower right vertex coordinates by the <x1> and <y1> parameters.
Specify the corresponding Unit Macro processing unit number in the <unitNo> parameter to draw the image at the position coordinates before applying position compensation. If the <unitNo> parameter is omitted, the image is drawn at the position coordinates after applying position compensation.
In the <x0>, <y0>, <x1>, and <y1> parameters, specify the camera coordinates whose origin is at the upper-left corner of the image.
Normally 0 should be specified in the <imageNo> parameter.
In the <unitNo> parameter, normally specify the processing unit number of the Unit Macro processing unit that executes this process.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a non-existent number, numerical value, or combination of data types or values are specified for a parameter, an "Illegal function call" error will occur.
If a value outside the range -2147483648 to 2147483647 is specified as an integer parameter, an "Overflow" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- This macro function can only be used in the *MEASUREDISPI subroutine or the *MEASUREDISPG subroutine. If used in another subroutine, an "Illegal function call" error will occur.

Example

Uses the *MEASUREDISPG subroutine of the Unit Macro processing unit to display the rectangle whose upper-left and lower-right corner coordinates are the coordinates measured by the Processing Unit numbers 5 and 6 search processing units respectively. The measured X and Y coordinates can be gotten with External Reference Data numbers 6 and 7 respectively.
To display the string at the fixed position regardless of the position compensation result, specify the assigned processing unit number to this Unit Macro processing unit (where the *MEASUREDISPG subroutine is used) for the <unitNo> parameter.

```
*MEASUREDISPG

  Rem Get the measurement result.
  GetUnitData 5, 6, X_LEFTTOP#
  GetUnitData 5, 7, Y_LEFTTOP#
  GetUnitData 6, 6, X_RIGHTBOTTOM#
  GetUnitData 6, 7, Y_RIGHTBOTTOM#

  Rem Draw the image
  DrawBox Int(X_LEFTTOP#), Int(Y_LEFTTOP#), Int(X_RIGHTBOTTOM#), Int(Y_RIGHTBOTTOM#), 0, UnitNo

Return
```

Useable Modules

Unit macro

Supported Versions

Version 3.50 or later

Related Items

GetUnitData (Reference: ▶ Details (p.552))
SetDrawStyle (Reference: ▶ Details (p.700))
UnitNo (Reference: ▶ Details (p.790))
Int (Reference: ▶ Details (p.575))
UnitData (Reference: ▶ Details (p.778))
Ut (Reference: ▶ Details (p.793))

DrawCircle

Draw a circle on the image window.

Format

DrawCircle <x>, <y>, <radius>, <imageNo>[, <unitNo>]

Parameter

Parameter name	Data type	Description
<x>	Integer type	The center X coordinate of the drawn circle
<y>	Integer type	The center Y coordinate of the drawn circle
<radius>	Integer type	Radius of the drawn circle
<imageNo>	Integer type	Measurement image number to draw on (always 0)
<unitNo>	Integer type	Processing unit number to display the processing unit (0 to (the number of registered processing units in the current scene minus one))

Return value

None.

Description

On the measurement image whose specified image number is in the <imageNo> parameter, draw a circle of the specified radius by the <radius> parameter at the center coordinates specified in the <x> and <y> parameters.
Specify the corresponding Unit Macro processing unit number in the <unitNo> parameter to draw the image at the position coordinates before applying position compensation. If the <unitNo> parameter is omitted, the image is drawn at the position coordinates after applying position compensation.
In the <X> and <Y> parameters, specify the camera coordinates whose origin is at the upper-left corner of the image.
Normally 0 should be specified in the <imageNo> parameter.
In the <unitNo> parameter, normally specify the processing unit number of the Unit Macro processing unit that executes this process.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a non-existent number, numerical value, or combination of data types or values are specified for a parameter, an "Illegal function call" error will occur.
If a value outside the range -2147483648 to 2147483647 is specified as an integer parameter, an "Overflow" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- This macro function can only be used in the *MEASUREDISPI subroutine or the *MEASUREDISPG subroutine. If used in another subroutine, an "Illegal function call" error will occur.

Example

Uses the *MEASUREDISPG subroutine of the Unit Macro processing unit to draw a circle of the measured radius by the circular scan edge position processing unit (Processing Unit number 5) at the measured center coordinates by the same circular scan edge position processing unit used for measuring the radius. The measured X and Y coordinates and radius can be gotten with External Reference Data numbers 5 to 7 respectively.
To display the string at the fixed position regardless of the position compensation result, specify the assigned processing unit number to this Unit Macro processing unit (where the *MEASUREDISPG subroutine is used) for the <unitNo> parameter.

```
*MEASUREDISPG

  Rem Get the measurement result.
  GetUnitData 5, 5, X#
  GetUnitData 5, 6, Y#
  GetUnitData 5, 7, R#

  Rem Draw the image
  DrawCircle Int(X#),Int(Y#), Int(R#), 0, UnitNo

Return
```

Useable Modules

Unit macro

Supported Versions

Version 3.50 or later

Related Items

GetUnitData (Reference: ▶ Details (p.552))
SetDrawStyle (Reference: ▶ Details (p.700))
UnitNo (Reference: ▶ Details (p.790))
Int (Reference: ▶ Details (p.575))
UnitData (Reference: ▶ Details (p.778))
Ut (Reference: ▶ Details (p.793))

DrawCircleW

Draw the wide circle on the image window.

Format

DrawCircleW <x>, <y>, <width>, <radius>, <imageNo>[, <unitNo>]

Parameter

Parameter name	Data type	Description
<x>	Integer type	The center X coordinate of the drawn circle
<y>	Integer type	The center Y coordinate of the drawn circle
<width>	Integer type	The width of the drawn wide circle
<radius>	Integer type	Radius of the drawn circle
<imageNo>	Integer type	Measurement image number to draw on (always 0)
<unitNo>	Integer type	Processing unit number to display the processing unit (0 to (the number of registered processing units in the current scene minus one))

Return value

None.

Description

On the measurement image whose specified image number is in the <imageNo> parameter, draw a wide circle of the specified radius by the <radius> parameter and the specified width by the <width> parameter at the center coordinates specified in the <x> and <y> parameters.
Specify the corresponding Unit Macro processing unit number in the <unitNo> parameter to draw the image at the position coordinates before applying position compensation. If the <unitNo> parameter is omitted, the image is drawn at the position coordinates after applying position compensation.
In the <X> and <Y> parameters, specify the camera coordinates whose origin is at the upper-left corner of the image.
Normally 0 should be specified in the <imageNo> parameter.
In the <unitNo> parameter, normally specify the processing unit number of the Unit Macro processing unit that executes this process.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a non-existent number, numerical value, or combination of data types or values are specified for a parameter, an "Illegal function call" error will occur.
If a value outside the range -2147483648 to 2147483647 is specified as an integer parameter, an "Overflow" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- This macro function can only be used in the *MEASUREDISPI subroutine or the *MEASUREDISPG subroutine. If used in another subroutine, an "Illegal function call" error will occur.

Example

Uses the *MEASUREDISPG subroutine of the Unit Macro processing unit to draw a circle of the measured radius by the circular scan edge position processing unit (Processing Unit number 5) at the measured center coordinates by the same circular scan edge position processing unit used for measuring the radius. This wide circle has an outer radius of the measured maximum radius and an inner radius of the measured minimum radius. The measured X/Y coordinates and radius, and maximum/minimum radii can be gotten with External Reference Data numbers 5, 6, 7, 8, and 9 respectively.
To display the string at the fixed position regardless of the position compensation result, specify the assigned processing unit number to this Unit Macro processing unit (where the *MEASUREDISPG subroutine is used) for the <unitNo> parameter.

```
*MEASUREDISPG

  Rem Get the measurement result.
  GetUnitData 5, 5, X#
  GetUnitData 5, 6, Y#
  GetUnitData 5, 7, R#
  GetUnitData 5, 8, R_MAX#
  GetUnitData 5, 9, R_MIN#

  Rem Determine the width
  W# = R_MAX# - R_MIN#

  Rem Draw the image
  DrawCircleW Int(X#), Int(Y#), Int(W#), Int(R#), 0, UnitNo

Return
```

Useable Modules

Unit macro

Supported Versions

Version 3.50 or later

Related Items

GetUnitData (Reference: ▶ Details (p.552))
SetDrawStyle (Reference: ▶ Details (p.700))
UnitNo (Reference: ▶ Details (p.790))
Int (Reference: ▶ Details (p.575))
UnitData (Reference: ▶ Details (p.778))
Ut (Reference: ▶ Details (p.793))

DrawCursor

Draw the cross-hair cursor on the image window.

Format

DrawCursor <x>, <y>, <imageNo>[, <unitNo>]

Parameter

Parameter name	Data type	Description
<x>	Integer type	The center X coordinate of the drawn cross-hair cursor
<y>	Integer type	The center Y coordinate of the drawn cross-hair cursor
<imageNo>	Integer type	Measurement image number to draw on (always 0)
<unitNo>	Integer type	Processing unit number to display the processing unit (0 to (the number of registered processing units in the current scene minus one))

Return value

None.

Description

On the measurement image whose specified image number is in the <imageNo> parameter, draw a cross-hair cursor at the center coordinates specified in the <x> and <y> parameters.
Specify the corresponding Unit Macro processing unit number in the <unitNo> parameter to draw the image at the position coordinates before applying position compensation. If the <unitNo> parameter is omitted, the image is drawn at the position coordinates after applying position compensation.
In the <X> and <Y> parameters, specify the camera coordinates whose origin is at the upper-left corner of the image.
Normally 0 should be specified in the <imageNo> parameter.
In the <unitNo> parameter, normally specify the processing unit number of the Unit Macro processing unit that executes this process.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a non-existent number, numerical value, or combination of data types or values are specified for a parameter, an "Illegal function call" error will occur.
If a value outside the range -2147483648 to 2147483647 is specified as an integer parameter, an "Overflow" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- This macro function can only be used in the *MEASUREDISPI subroutine or the *MEASUREDISPG subroutine. If used in another subroutine, an "Illegal function call" error will occur.

Example

Uses the *MEASUREDISPG subroutine of the Unit Macro processing unit to draw a cross-hair cursor at the measured coordinates by the search processing unit (Processing Unit number 5). The measured X and Y coordinates can be gotten with External Reference Data numbers 6 and 7 respectively.
To display the string at the fixed position regardless of the position compensation result, specify the assigned processing unit number to this Unit Macro processing unit (where the *MEASUREDISPG subroutine is used) for the <unitNo> parameter.

```
*MEASUREDISPG

  Rem Get the measurement result.
  GetUnitData 5, 6, X#
  GetUnitData 5, 7, Y#

  Rem Draw the image
  DrawCursor Int(X#), Int(Y#), 0, UnitNo

Return
```

Useable Modules

Unit macro

Supported Versions

Version 3.50 or later

Related Items

GetUnitData (Reference: ▶ Details (p.552))
SetDrawStyle (Reference: ▶ Details (p.700))
UnitNo (Reference: ▶ Details (p.790))
Int (Reference: ▶ Details (p.575))
UnitData (Reference: ▶ Details (p.778))
Ut (Reference: ▶ Details (p.793))

DrawEllipse

Draw the ellipse on the image window.

Format

DrawCircle <x>, <y>, <radiusX>, <radiusY>, <imageNo>[, <unitNo>]

Parameter

Parameter name	Data type	Description
<x>	Integer type	The center X coordinate of the drawn ellipse
<y>	Integer type	The center Y coordinate of the drawn ellipse
<radiusX>	Integer type	Radius in the X direction of the drawn ellipse
<radiusY>	Integer type	Radius in the Y direction of the drawn ellipse
<imageNo>	Integer type	Measurement image number to draw on (always 0)
<unitNo>	Integer type	Processing unit number to display the processing unit (0 to (the number of registered processing units in the current scene minus one))

Return value

None.

Description

On the measurement image whose specified image number is in the <imageNo> parameter, draw an ellipse of the specified radius in the X direction by the <radiusX> parameter and the specified radius in the Y direction by the <radius Y> parameter at the center coordinates specified in the <x> and <y> parameters.
Specify the corresponding Unit Macro processing unit number in the <unitNo> parameter to draw the image at the position coordinates before applying position compensation. If the <unitNo> parameter is omitted, the image is drawn at the position coordinates after applying position compensation.
In the <X> and <Y> parameters, specify the camera coordinates whose origin is at the upper-left corner of the image.
Normally 0 should be specified in the <imageNo> parameter.
In the <unitNo> parameter, normally specify the processing unit number of the Unit Macro processing unit that executes this process.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a non-existent number, numerical value, or combination of data types or values are specified for a parameter, an "Illegal function call" error will occur.
If a value outside the range -2147483648 to 2147483647 is specified as an integer parameter, an "Overflow" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- This macro function can only be used in the *MEASUREDISPI subroutine or the *MEASUREDISPG subroutine. If used in another subroutine, an "Illegal function call" error will occur.

Example

Uses the *MEASUREDISPG subroutine of the Unit Macro processing unit to display the ellipse whose radii in the X and Y directions are the measured coordinates of the center of gravity by the labeling processing unit (Processing Unit number 5). In this example, set the judgement conditions for the labeling processing unit to "Gravity X", "Gravity Y", "Elliptic major axis", and "Elliptic minor axis" from label number 0. Therefore, the assigned external reference data numbers to the gravity X, gravity Y, elliptic major axis, and elliptic minor axis parameters are 1000, 1100, 1200, and 1300 respectively.
To display the string at the fixed position regardless of the position compensation result, specify the assigned processing unit number to this Unit Macro processing unit (where the *MEASUREDISPG subroutine is used) for the <unitNo> parameter.

```
*MEASUREDISPG

  Rem Get the measurement result.
  GetUnitData 5, 1000, X#
  GetUnitData 5, 1100, Y#
  GetUnitData 5, 1200, XDIAMETER#
  GetUnitData 5, 1300, YDIAMETER#

  Rem Draw the image
  DrawEllipse Int(X#), Int(Y#), Int(XDIAMETER# / 2), Int(YDIAMETER# / 2), 0, UnitNo

Return
```

Useable Modules

Unit macro

Supported Versions

Version 3.50 or later

Related Items

GetUnitData (Reference: ▶ Details (p.552))
SetDrawStyle (Reference: ▶ Details (p.700))
UnitNo (Reference: ▶ Details (p.790))
Int (Reference: ▶ Details (p.575))
UnitData (Reference: ▶ Details (p.778))
Ut (Reference: ▶ Details (p.793))

DrawFigure

Draw a figure on the image window.

Format

DrawFigure <figure()>, <imageNo>[, <unitNo>]

Parameter

Parameter name	Data type	Description
<figure()>	Integer array	1D array that stores the drawn figure data (Reference: ▶Figure Data Structure List (p.364))
<imageNo>	Integer type	Measurement image number to draw on (always 0)
<unitNo>	Integer type	Processing unit number to display the processing unit (0 to (the number of registered processing units in the current scene minus one))

Return value

None.

Description

On the measurement image whose specified image number is in the <imageNo> parameter, draw a figure specified in the <figure()> parameter.
Specify the corresponding Unit Macro processing unit number in the <unitNo> parameter to draw the image at the position coordinates before applying position compensation. If the <unitNo> parameter is omitted, the image is drawn at the position coordinates after applying position compensation.
In the <figure()> parameter, specify the 1D integer array variable that will hold the figure data by adding only () without specifying an element number.
Normally 0 should be specified in the <imageNo> parameter.
In the <unitNo> parameter, normally specify the processing unit number of the Unit Macro processing unit that executes this process.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a non-existent number, numerical value, or combination of data types or values are specified for a parameter, an "Illegal function call" error will occur.
If a value outside the range -2147483648 to 2147483647 is specified as an integer parameter, an "Overflow" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- This macro function can only be used in the *MEASUREDISPI subroutine or the *MEASUREDISPG subroutine. If used in another subroutine, an "Illegal function call" error will occur.

Example

Uses the *MEASUREDISPG subroutine of the Unit Macro processing unit to simultaneously display registered model figures for the search processing units (Processing Unit numbers 5, 6, 7, and 8). The model registration figure of the Search processing item is figure 0. (Reference: ▶List of Figure Numbers (p.366)) Use the Dim function to define an array with element number that is larger than the element number used for the figure data.

```
*MEASUREDISPG

  Dim FIGURE1&(255), FIGURE2&(255), FIGURE3&(255), FIGURE4&(255)

  Rem Get the figure data of the model figure.
  GetUnitFigure 5, 0, FIGURE1&()
  GetUnitFigure 6, 0, FIGURE2&()
  GetUnitFigure 7, 0, FIGURE3&()
  GetUnitFigure 8, 0, FIGURE4&()

  Rem Draw the image
  DrawFigure FIGURE1&(), 0, UnitNo
  DrawFigure FIGURE2&(), 0, UnitNo
  DrawFigure FIGURE3&(), 0, UnitNo
  DrawFigure FIGURE4&(), 0, UnitNo

Return
```

Useable Modules

Unit macro

Supported Versions

Version 3.50 or later

Related Items

GetUnitData (Reference: ▶ Details (p.552))
SetDrawStyle (Reference: ▶ Details (p.700))
UnitNo (Reference: ▶ Details (p.790))
Int (Reference: ▶ Details (p.575))
UnitData (Reference: ▶ Details (p.778))
Ut (Reference: ▶ Details (p.793))

DrawFillImage

Draw the fill image on the image window.

Format

DrawFillImage <color>

Parameter

Parameter name	Data type	Description
<color>	Integer type	Color value of the color to fill with (Reference: ▶RGB (p.662))

Return value

None.

Description

Draw the color filled image specified in the <color> parameter.
In <color> parameter, specify the color value gotten with the RGB function.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a value outside the range -2147483648 to 2147483647 is specified as an integer parameter, an "Overflow" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- This macro function can only be used in the *MEASUREDISPI subroutine or the *MEASUREDISPG subroutine. If used in another subroutine, an "Illegal function call" error will occur.

Example

Uses the *MEASUREDISPI subroutine in the Unit Macro processing unit to display a white-filled figure.

```
*MEASUREDISPI

  DrawFillImage RGB(255, 255, 255)

Return
```

Useable Modules

Unit macro

Supported Versions

Version 3.50 or later

Related Items

DrawMeasureImage (Reference: ▶ Details (p.492)) DrawUnitImage (Reference: ▶ Details (p.504))
RGB (Reference: ▶ Details (p.662))

DrawJudgeText

Draws the judgement result of the character string on the text display screen.

Format

DrawJudgeText <judge>

Parameter

Parameter name	Data type	Description
<judge>	Integer type	Judgement results to be drawn JUDGE_NC: "No judgement (unmeasured)" JUDGE_OK: Judgement: OK JUDGE_NG: Judgement: NG JUDGE_IMAGEERROR: Judgement: NG (Image mismatch) JUDGE_MODELERROR: Judgement: NG (model not register) JUDGE_MEMORYERROR: Judgement: NG (Out of memory) JUDGE_ERRORJudgement: NG (immeasurable)

Return value

None.

Description

Draw the specified judgement result string by the <judge> parameter in the text window.
Gotten value with the UnitJudge function can be specified in the <judge> parameter. (Reference: ▶UnitJudge (p.788))
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
Even if a non-existent number, numerical value, or combination of data types or values is specified for the parameter, an error will not occur.
If a value outside the range -2147483648 to 2147483647 is specified as an integer parameter, an "Overflow" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- This macro function can only be used in the *MEASUREDISPT subroutine. If used in another subroutine, an "Illegal function call" error will occur.

Example

Uses the *MEASUREDISPT subroutine in the Unit Macro processing unit to draw the judgement result text string for the processing unit in the text window.

```
*MEASUREDISPT
  DrawJudgeText UnitJudge(UnitNo)
Return
```

Useable Modules

Unit macro

Supported Versions

Version 3.50 or later

Related Items

DrawText (Reference: ▶ Details (p.500))
UnitData (Reference: ▶ Details (p.778))
UnitNo (Reference: ▶ Details (p.790))
GetUnitData (Reference: ▶ Details (p.552))
UnitJudge (Reference: ▶ Details (p.788))

DrawLine

Draw a straight line on the image window.

Format

DrawBox <x0>, <y0>, <x1>, <y1>, <imageNo>[, <unitNo>]

Parameter

Parameter name	Data type	Description
<x0>	Integer type	The starting point X coordinate of the drawn straight line
<y0>	Integer type	The starting point Y coordinate of the drawn straight line
<x1>	Integer type	The ending point X coordinate of the drawn straight line
<y1>	Integer type	The ending point Y coordinate of the drawn straight line
<imageNo>	Integer type	Measurement image number to draw on (always 0)
<unitNo>	Integer type	Processing unit number to display the processing unit (0 to (the number of registered processing units in the current scene minus one))

Return value

None.

Description

On the measurement image whose specified image number is in the <imageNo> parameter, draws a line that starts at the specified X- and Y-coordinates by the <x0> and <y0> parameters and ends at the specified X- and Y-coordinates by the <x1> and <y1> parameters.
Specify the corresponding Unit Macro processing unit number in the <unitNo> parameter to draw the image at the position coordinates before applying position compensation. If the <unitNo> parameter is omitted, the image is drawn at the position coordinates after applying position compensation.
In the <x0>, <y0>, <x1>, and <y1> parameters, specify the camera coordinates whose origin is at the upper-left corner of the image.
Normally 0 should be specified in the <imageNo> parameter.
In the <unitNo> parameter, normally specify the processing unit number of the Unit Macro processing unit that executes this process.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a non-existent number, numerical value, or combination of data types or values is specified for a parameter, an "Illegal function call" error will occur.
If a value outside the range -2147483648 to 2147483647 is specified as an integer parameter, an "Overflow" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- This macro function can only be used in the *MEASUREDISPI subroutine or the *MEASUREDISPG subroutine. If used in another subroutine, an "Illegal function call" error will occur.

Example

Uses the *MEASUREDISPG subroutine of the Unit Macro processing unit to display the line whose starting and ending point coordinates are the coordinates measured by the Processing Unit numbers 5 and 6 search processing units respectively. The measured X and Y coordinates can be gotten with External Reference Data numbers 6 and 7 respectively.
To display the string at the fixed position regardless of the position compensation result, specify the assigned processing unit number to this Unit Macro processing unit (where the *MEASUREDISPG subroutine is used) for the <unitNo> parameter.

```
*MEASUREDISPG

  Rem Get the measurement result.
  GetUnitData 5, 6, X_START#
  GetUnitData 5, 7, Y_START#
  GetUnitData 6, 6, X_END#
  GetUnitData 6, 7, Y_END#

  Rem Draw the image
  DrawLine Int(X_START#), Int(Y_START#), Int(X_END#), Int(Y_END#), 0, UnitNo

Return
```

Useable Modules

Unit macro

Supported Versions

Version 3.50 or later

Related Items

GetUnitData (Reference: ▶ Details (p.552))
SetDrawStyle (Reference: ▶ Details (p.700))
UnitNo (Reference: ▶ Details (p.790))
Int (Reference: ▶ Details (p.575))
UnitData (Reference: ▶ Details (p.778))
Ut (Reference: ▶ Details (p.793))

DrawLineW

Draw the wide straight line on the image window.

Format

DrawBox <x0>, <y0>, <x1>, <y1>, <imageNo>[, <unitNo>]

Parameter

Parameter name	Data type	Description
<x0>	Integer type	The starting point X coordinate of the drawn straight line
<y0>	Integer type	The starting point Y coordinate of the drawn straight line
<x1>	Integer type	The ending point X coordinate of the drawn straight line
<y1>	Integer type	The ending point Y coordinate of the drawn straight line
<width>	Integer type	Width of the drawn straight line
<imageNo>	Integer type	Measurement image number to draw on (always 0)
<unitNo>	Integer type	Processing unit number to display the processing unit (0 to (the number of registered processing units in the current scene minus one))

Return value

None.

Description

On the measurement image whose specified image number is in the <imageNo> parameter, draws a wide line that starts at the specified X and Y coordinates by the <x0> and <y0> parameters and ends at the specified X and Y coordinates by the <x1> and <y1> parameters with the specified width by the <width> parameter.
Specify the corresponding Unit Macro processing unit number in the <unitNo> parameter to draw the image at the position coordinates before applying position compensation. If the <unitNo> parameter is omitted, the image is drawn at the position coordinates after applying position compensation.
In the <x0>, <y0>, <x1>, and <y1> parameters, specify the camera coordinates whose origin is at the upper-left corner of the image.
Normally 0 should be specified in the <imageNo> parameter.
In the <unitNo> parameter, normally specify the processing unit number of the Unit Macro processing unit that executes this process.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a non-existent number, numerical value, or combination of data types or values is specified for a parameter, an "Illegal function call" error will occur.
If a value outside the range -2147483648 to 2147483647 is specified as an integer parameter, an "Overflow" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- This macro function can only be used in the *MEASUREDISPI subroutine or the *MEASUREDISPG subroutine. If used in another subroutine, an "Illegal function call" error will occur.

Example

Uses the *MEASUREDISPG subroutine of the Unit Macro processing unit to display the wide line with width of 10 whose starting and ending point coordinates are the coordinates measured by the Processing Unit numbers 5 and 6 search processing units respectively. The measured X and Y coordinates can be gotten with External Reference Data numbers 6 and 7 respectively.
To display the string at the fixed position regardless of the position compensation result, specify the assigned processing unit number to this Unit Macro processing unit (where the *MEASUREDISPG subroutine is used) for the <unitNo> parameter.

```
*MEASUREDISPG

  Rem Get the measurement result.
  GetUnitData 5, 6, X_START#
  GetUnitData 5, 7, Y_START#
  GetUnitData 6, 6, X_END#
  GetUnitData 6, 7, Y_END#

  Rem Draw the image
  DrawLineW Int(X_START#), Int(Y_START#), Int(X_END#), Int(Y_END#), 10, 0, UnitNo

Return
```

Useable Modules

Unit macro

Supported Versions

Version 3.50 or later

Related Items

GetUnitData (Reference: ▶ Details (p.552))
SetDrawStyle (Reference: ▶ Details (p.700))
UnitNo (Reference: ▶ Details (p.790))
Int (Reference: ▶ Details (p.575))
UnitData (Reference: ▶ Details (p.778))
Ut (Reference: ▶ Details (p.793))

DrawMeasureImage

Draw the measurement image on the image window.

Format

DrawMeasureImage <imageNo>

Parameter

Parameter name	Data type	Description
<imageNo>	Integer type	Number of the measurement image to display (always 0)

Return value

None.

Description

Display an image that has been registered to the Unit Macro processing unit where this macro function is executed and whose image number is specified in the <imageNo> parameter.
Even if a value that does not exist is specified for the <imageNo> parameter, an error will not occur.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a value outside the range -2147483648 to 2147483647 is specified as an integer parameter, an "Overflow" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- This macro function can only be used in the *MEASUREDISPI subroutine or the *MEASUREDISPG subroutine. If used in another subroutine, an "Illegal function call" error will occur.

Example

Uses the *MEASUREDISPI subroutine in the Unit Macro processing unit to display a measurement image.

```
*MEASUREDISPI

  DrawMeasureImage 0

Return
```

Useable Modules

Unit macro

Supported Versions

Version 3.50 or later

Related Items

DrawFillImage (Reference: ▶ Details (p.485))
DrawUnitImage (Reference: ▶ Details (p.504))

DrawPoint

Draw a point on the image window.

Format

DrawPoint <x>, <y>, <imageNo>[, <unitNo>]

Parameter

Parameter name	Data type	Description
<x>	Integer type	The X coordinate of the drawn point
<y>	Integer type	The Y coordinate of the drawn point
<imageNo>	Integer type	Measurement image number to draw on (always 0)
<unitNo>	Integer type	Processing unit number to display the processing unit (0 to (the number of registered processing units in the current scene minus one))

Return value

None.

Description

On the measurement image whose specified image number is in the <imageNo> parameter, draw a point at the coordinates specified in the <x> and <y> parameters.
Specify the corresponding Unit Macro processing unit number in the <unitNo> parameter to draw the image at the position coordinates before applying position compensation. If the <unitNo> parameter is omitted, the image is drawn at the position coordinates after applying position compensation.
In the <X> and <Y> parameters, specify the camera coordinates whose origin is at the upper-left corner of the image.
Normally 0 should be specified in the <imageNo> parameter.
In the <unitNo> parameter, normally specify the processing unit number of the Unit Macro processing unit that executes this process.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a non-existent number, numerical value, or combination of data types or values is specified for a parameter, an "Illegal function call" error will occur.
If a value outside the range -2147483648 to 2147483647 is specified as an integer parameter, an "Overflow" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- This macro function can only be used in the *MEASUREDISPI subroutine or the *MEASUREDISPG subroutine. If used in another subroutine, an "Illegal function call" error will occur.

Example

Uses the *MEASUREDISPG subroutine of the Unit Macro processing unit to draw a point at the measured coordinates by the search processing unit (Processing Unit number 5). The measured X and Y coordinates can be gotten with External Reference Data numbers 6 and 7 respectively.
To display the string at the fixed position regardless of the position compensation result, specify the assigned processing unit number to this Unit Macro processing unit (where the *MEASUREDISPG subroutine is used) for the <unitNo> parameter.

```
*MEASUREDISPG

  Rem Get the measurement result.
  GetUnitData 5, 6, X#
  GetUnitData 5, 7, Y#

  Rem Draw the image
  DrawPoint Int(X#), Int(Y#), 0, UnitNo

Return
```

Useable Modules

Unit macro

Supported Versions

Version 3.50 or later

Related Items

GetUnitData (Reference: ▶ Details (p.552))
SetDrawStyle (Reference: ▶ Details (p.700))
UnitNo (Reference: ▶ Details (p.790))
Int (Reference: ▶ Details (p.575))
UnitData (Reference: ▶ Details (p.778))
Ut (Reference: ▶ Details (p.793))

DrawPolygon

Draw a polygon on the image window.

Format

DrawPolygon <count>, <x()>, <y()>, <imageNo>[, <unitNo>]

Parameter

Parameter name	Data type	Description
<count>	Integer type	Number of polygon vertices (0 or larger) to be drawn
<x()>	Integer array	1D array that stores the X coordinate of the drawn polygon
<y()>	Integer array	1D array that stores the Y coordinate of the drawn polygon
<imageNo>	Integer type	Measurement image number to draw on (always 0)
<unitNo>	Integer type	Processing unit number to display the processing unit (0 to (the number of registered processing units in the current scene minus one))

Return value

None.

Description

On the measurement image whose specified image number is in the <imageNo> parameter, draw a polygon having the specified number of vertices by the <count> parameter and whose vertices are at the specified coordinates by the <x()> and <y()> parameters.
Specify the corresponding Unit Macro processing unit number in the <unitNo> parameter to draw the image at the position coordinates before applying position compensation. If the <unitNo> parameter is omitted, the image is drawn at the position coordinates after applying position compensation.
In the <x()> parameter and in the <y()> parameter, specify a 1D integer number array variable that stores a number of coordinate values greater than or equal to the number specified in the <count> parameter, without adding element numbers but adding () to the variables.
In the <x()> and <y()> parameters, specify the camera coordinates whose origin is at the upper-left corner of the image.
Normally 0 should be specified in the <imageNo> parameter.
In the <unitNo> parameter, normally specify the processing unit number of the Unit Macro processing unit that executes this process.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a non-existent number, numerical value, or combination of data types or values is specified for a parameter, an "Illegal function call" error will occur.
If a value outside the range -2147483648 to 2147483647 is specified as an integer parameter, an "Overflow" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- This macro function can only be used in the *MEASUREDISPI subroutine or the *MEASUREDISPG subroutine. If used in another subroutine, an "Illegal function call" error will occur.

Example

Uses the *MEASUREDISPG subroutine of the Unit Macro processing unit to draw a polygon whose vertices are at the measured positions by the search processing units (Processing unit numbers 5 to 7). The measured X and Y coordinates can be gotten with External Reference Data numbers 6 and 7 respectively. To display the string at the fixed position regardless of the position compensation result, specify the assigned processing unit number to this Unit Macro processing unit (where the *MEASUREDISPG subroutine is used) for the <unitNo> parameter.

```
*MEASUREDISPG

Dim X&(2), Y&(2)

  Rem Get the measurement result.
  GetUnitData 5, 6, X&(0)
  GetUnitData 5, 7, Y&(0)
  GetUnitData 6, 6, X&(1)
  GetUnitData 6, 7, Y&(1)
  GetUnitData 7, 6, X&(2)
  GetUnitData 7, 7, Y&(2)

  Rem Draw the image
  DrawPolygon 3, X&(), Y&(), 0, UnitNo

Return
```

Useable Modules

Unit macro

Supported Versions

Version 3.50 or later

Related Items

GetUnitData (Reference: ▶ Details (p.552))
SetDrawStyle (Reference: ▶ Details (p.700))
UnitNo (Reference: ▶ Details (p.790))
Int (Reference: ▶ Details (p.575))
UnitData (Reference: ▶ Details (p.778))
Ut (Reference: ▶ Details (p.793))

DrawSearchFigure

Draw the search figure on the image window.

Format

DrawSearchFigure <figure()>, <referenceX>, <referenceY>, <measureX>, <measureY>, <measureAngle>, <imageNo>, <unitNo>

Parameter

Parameter name	Data type	Description
<figure()>	Integer array	1D array that stores the drawn figure data (Reference: ▶Figure Data List (p.364))
<referenceX>	Double precision real number data type	Detection point X coordinate used for the drawn figure position specification
<referenceY>	Double precision real number data type	Detection point Y coordinate used for the drawn figure position specification
<measureX>	Double precision real number data type	Measurement X coordinate used for the drawn figure position specification
<measureY>	Double precision real number data type	Measurement Y coordinate used for the drawn figure position specification
<measureAngle>	Double precision real number data type	Measured angle used for the drawn figure angle specification
<imageNo>	Integer type	Measurement image number to draw on (always 0)
<unitNo>	Integer type	Processing unit number to display the processing unit (0 to (the number of registered processing units in the current scene minus one))

Return value

None.

Description

On the measurement image whose specified image number is in the <imageNo> parameter, draw a figure specified in the <figure()> parameter at the specified position with the <referenceX>, <referenceY>, <measureX>, and <measureY> parameter at an angle specified in the <measureAngle>.
With the use of this macro function, drawing of a figure composed of registered model figures and detected points gotten mainly from search processing units and shape search III processing units. This macro function cannot draw images properly if the referenced measurement results from processing units where the model registration or detection point specification is not implemented are specified in the function arguments.
In the <referenceX> parameter and the <referenceY> parameter, normally specify detection points X and Y of the referenced search processing unit.
In the <measureX> parameter, the <measureY> parameter, and the <measureAngle> parameter, normally specify the measurement coordinates X and Y and the measurement angle of a referenced search

processing unit.
Specify the corresponding Unit Macro processing unit number in the <unitNo> parameter to draw the image at the position coordinates before applying position compensation. If the <unitNo> parameter is omitted, the image is drawn at the position coordinates after applying position compensation.
In the <figure()> parameter, specify the 1D integer array variable that will hold the figure data by adding only () without specifying an element number.
Normally 0 should be specified in the <imageNo> parameter.
In the <unitNo> parameter, normally specify the processing unit number of the Unit Macro processing unit that executes this process.
This macro function is mainly used for drawing the model figure detected by the search processing unit.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a non-existent number, numerical value, or combination of data types or values is specified for a parameter, an "Illegal function call" error will occur.
If a value outside the range -2147483648 to 2147483647 is specified as an integer parameter, an "Overflow" error will occur.
If a value outside the range -1.0E30 to 1.0E30 is specified for a double precision real number parameter, an "Overflow" error might occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- This macro function can only be used in the *MEASUREDISPI subroutine or the *MEASUREDISPG subroutine. If used in another subroutine, an "Illegal function call" error will occur.

Example

Uses the *MEASUREDISPG subroutine of the Unit Macro processing unit to display a registered model figure for the search processing unit (Processing Unit number 5). The measured X/Y coordinates and radius, and the detection point X/Y coordinates can be gotten with External Reference Data numbers 6, 7, 8, 132, and 133 respectively.
To display the string at the fixed position regardless of the position compensation result, specify the assigned processing unit number to this Unit Macro processing unit (where the *MEASUREDISPG subroutine is used) for the <unitNo> parameter.

```
*MEASUREDISPG

  Dim FIGURE&(255)

  Rem Get the model figure
  GetUnitFigure 5, 0, FIGURE&()

  Rem Get the measurement result.
  GetUnitData 5, 6, X#
  GetUnitData 5, 7, Y#
  GetUnitData 5, 8, TH#
  GetUnitData 5, 132, RX#
  GetUnitData 5, 133, RY#

  Rem Draw the image
  DrawSearchFigure FIGURE&(), RX#, RY#, X#, Y#, TH#, 0, UnitNo

Return
```

Useable Modules

Unit macro

Supported Versions

Version 3.50 or later

Related Items

GetUnitData (Reference: ▶ Details (p.552))
SetDrawStyle (Reference: ▶ Details (p.700))
UnitNo (Reference: ▶ Details (p.790))
Int (Reference: ▶ Details (p.575))
UnitData (Reference: ▶ Details (p.778))
Ut (Reference: ▶ Details (p.793))

DrawText

Draw a character string on the text window.

Format

DrawText <string>, <color>, <newLine>

Parameter

Parameter name	Data type	Description
<string>	Character string type	Character string to display
<color>	Integer type	Color value of character string color to be drawn JUDGE_NC: Unmeasured color (Grey) JUDGE_OK: OK judgement color (Green) JUDGE_NG: NG judgement color (Red) RGB Function: Any color
<newLine>	Integer type	Line break after display 0: Do not break a line 1: Break a line

Return value

None.

Description

Use the color specified in the <color> parameter and with the line break method specified in the <newLine> parameter to draw the character string specified in the <string> parameter in the text window.
The gotten color value by the RGB function can be set for the <color> parameter. (Reference: ▶RGB (p.662))
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
Even if a value that does not exist is specified for a parameter, an error will not occur.
If a value outside the range -2147483648 to 2147483647 is specified as an integer parameter, an "Overflow" error will occur.
If a character string longer than 255 characters is specified for a character string parameter, a "String too long" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- This macro function can only be used in the *MEASUREDISPT subroutine. If used in another subroutine, an "Illegal function call" error will occur.

Example

Uses the *MEASUREDISPT subroutine of the Unit Macro processing unit to display the provided measurement results by the search processing unit (Processing Unit number 5) in the text window. The measured X and Y coordinates and angle can be gotten with External Reference Data numbers 6, 7, and 8 respectively.

```
*MEASUREDISPT

  Rem Get the measurement result.
  GetUnitData 5, 6, X#
  GetUnitData 5, 7, Y#
  GetUnitData 5, 8, TH#

  Rem Draw the delimiter after drawing the character string with the "OK" judgement color without adding any line
  break.
  DrawText Str2$(X#, 4, 4, 0, 0), JUDGE_OK, 0
  DrawText ", ", JUDGE_OK, 0

  Rem Draw the character string with the "OK" judgement color and add a line break.
  DrawText Str2$(Y#, 4, 4, 0, 0), JUDGE_OK, 1

  Rem Draw the character string with the "OK" judgement color without adding any line break.
  DrawText Str2$(TH#, 4, 4, 0, 0), JUDGE_OK, 0

Return
```

The result is shown below.

```
123.4567,   10.5000
90.0000
```

Useable Modules

Unit macro

Supported Versions

Version 3.50 or later

Related Items

DrawJudgeText (Reference: ▶ Details (p.486))
RGB (Reference: ▶ Details (p.662))
Str2$ (Reference: ▶ Details (p.751))
GetUnitData (Reference: ▶ Details (p.552))
Str$ (Reference: ▶ Details (p.749))
UnitData (Reference: ▶ Details (p.778))

DrawTextG

Draw a character string on the image window.

Format

DrawTextG <string>, <x>, <y>, <imageNo>[, <unitNo>]

Parameter

Parameter name	Data type	Description
<string>	Character string type	Character string to display
<x>	Integer type	Upper left X coordinate value of the drawn area
<y>	Integer type	Upper left Y coordinate value of the drawn area
<imageNo>	Integer type	Measurement image number to draw on (always 0)
<unitNo>	Integer type	Processing unit number to display the processing unit (0 to (the number of registered processing units in the current scene minus one))

Return value

None.

Description

On the measurement image whose specified image number is in the <imageNo> parameter, draw a character string specified in the <string> parameter at the position coordinates specified in the <x> and <y> parameters.
Specify the corresponding Unit Macro processing unit number in the <unitNo> parameter to draw the image at the position coordinates before applying position compensation. If the <unitNo> parameter is omitted, the image is drawn at the position coordinates after applying position compensation.
In the <X> and <Y> parameters, specify the camera coordinates whose origin is at the upper-left corner of the image.
Normally 0 should be specified in the <imageNo> parameter.
In the <unitNo> parameter, normally specify the processing unit number of the Unit Macro processing unit that executes this process.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a non-existent number, numerical value, or combination of data types or values is specified for a parameter, an "Illegal function call" error will occur.
If a value outside the range -2147483648 to 2147483647 is specified as an integer parameter, an "Overflow" error will occur.
If a character string longer than 255 characters is specified for a character string parameter, a "String too long" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- This macro function can only be used in the *MEASUREDISPI subroutine or the *MEASUREDISPG subroutine. If used in another subroutine, an "Illegal function call" error will occur.

Example

Uses the *MEASUREDISPG subroutine of the Unit Macro processing unit to display the string "OK" at fixed position coordinates (100, 100). To display the string at the fixed position regardless of the position compensation result, specify the assigned processing unit number to this Unit Macro processing unit (where the *MEASUREDISPG subroutine is used) for the <unitNo> parameter.

```
*MEASUREDISPG
  DrawTextG "OK", 100, 100, 0, UnitNo
Return
```

Useable Modules

Unit macro

Supported Versions

Version 3.50 or later

Related Items

GetUnitData (Reference: ▶ Details (p.552))
Str$ (Reference: ▶ Details (p.749))
UnitData (Reference: ▶ Details (p.778))
Ut (Reference: ▶ Details (p.793))
SetTextStyle (Reference: ▶ Details (p.727))
Str2$ (Reference: ▶ Details (p.751))
UnitNo (Reference: ▶ Details (p.790))

DrawUnitImage

Display the "other unit image" on the image window. The "other unit image" refers to the held image by a processing unit for the judgement reference.

Format

DrawUnitImage <unitNo>, <imageNo>

Parameter

Parameter name	Data type	Description
<unitNo>	Integer type	Processing unit number that holds the displayed image (0 to (the number of processing units in the current scene minus one))
<imageNo>	Integer type	Drawn image number (Reference: ▶List of Figure Numbers (p.366))

Return value

None.

Description

Draw the image of the specified image number by the <imageNo> parameter that is held by the specified processing unit number by the <unitNo> parameter.
If a non-existent number, numerical value, or combination of data types or values is specified for the <unitNo> parameter, an "Illegal function call" error will occur.
Even if a value that does not exist is specified for the <imageNo> parameter, an error will not occur.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a value outside the range -2147483648 to 2147483647 is specified as an integer parameter, an "Overflow" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- This macro function can only be used in the *MEASUREDISPI subroutine or the *MEASUREDISPG subroutine. If used in another subroutine, an "Illegal function call" error will occur.

Example

Uses the *MEASUREDISPG subroutine of the Unit Macro processing unit to display image number 1 that is held in advance by the advanced filter processing unit (Processing Unit number 5) and is allocated to output image 1.

```
*MEASUREDISPG

  Rem Display the "other unit image"
  DrawUnitImage 5, 1

Return
```

Useable Modules

Unit macro

Supported Versions

Version 3.50 or later

Related Items

DrawFillImage (Reference: ▶ Details (p.485))
UnitNo (Reference: ▶ Details (p.790))
DrawMeasureImage (Reference: ▶ Details (p.492))
Ut (Reference: ▶ Details (p.793))

Dskf

Gets the free space on disk drives.

Format

Dskf(<driveName>)

Parameter

Parameter name	Data type	Description
<driveName>	Character string type	Drive name whose free space is to be gotten

Return value

Returns free space (in bytes) on the disk drive as an integer value.
Returns -1 when the specified disk drive does not exist.

Description

Determinate the free space (in bytes) on the disk drive specified in the <driveName> parameter.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a character string longer than 255 characters is specified for a character string parameter, a "String too long" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

Turns ON the ERROR signal when the free space on the disk drive reaches less than KB (1,024 bytes).

```
Rem Check the free space on the disk drive
If Dskf("E:\") < 1024 Then

  Rem Turn ON the ERROR Signal
  PutPort "ParallelIo", 103, 1

Endif
```

Useable Modules

Unit Calculation Macro / Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 3.50 or later

Related Items

Fcopy (Reference: ▶ Details (p.522))
Kill (Reference: ▶ Details (p.587))
PutPort (Reference: ▶ Details (p.645))
IsFile (Reference: ▶ Details (p.577))
Mkdir (Reference: ▶ Details (p.623))
Rmdir (Reference: ▶ Details (p.666))

ElapsedTime

Gets the elapsed time since starting the measurement.

Format

ElapsedTime(<mode>)

Parameter

Parameter name	Data type	Description
<mode>	Integer type	Unit of the elapse time to get 0: ms unit 1: μ unit

Return value

Returns the elapsed time as an integer value.

Description

Gets the elapsed time since starting the measurement with the unit specified in the <mode> parameter.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a value is assigned to the return value variable or the variable is not used in an expression, a "Syntax error" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

Uses the MEASUREPROC subroutine of the Unit Macro processing unit to get the elapsed time until this macro function is executed. If the elapsed time is 1,000 ms or longer, the error character string is displayed in the text window.

```
*MEASUREPROC

  Rem Get the elapsed time.
  TIME& = ElapsedTime(0)

Return



*MEASUREDISPT

  Rem If the elapsed time is 1,000 ms or longer, the error character string is displayed in the NG color.
  If TIME& > 999 Then
    DrawText "Error", JUDGE_NG, 1
  Endif
```

Useable Modules

Unit Calculation Macro / Unit Macro

Supported Versions

Version 3.50 or later

Related Items

DrawText (Reference: ▶ Details (p.500))
Timer (Reference: ▶ Details (p.759))
StartTimer (Reference: ▶ Details (p.746))
Wait (Reference: ▶ Details (p.806))

Eof

Examines the end of the file.

Format

Eof(<fileNo>)

Parameter

Parameter name	Data type	Description
<fileNo>	Integer type	File number (0 to 15) of the examined file end.

Return value

Returns an integer value that notifies if the end of the file is reached.

- 0: The end of the file is reached
- -1: The end of file is not reached

Description

Check if the end of the file of the file number specified in the <fileNo> is reached.
If an unopened file number is specified in the <fileNo> parameter, an "Illegal function call" error will occur.
If a value outside the range of 0 to 15 is specified in the <fileNo> parameter, an "Illegal function call" error will occur.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a value outside the range -2147483648 to 2147483647 is specified as an integer parameter, an "Overflow" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

Reads the data until the end of the file.

```
Dim ALLDATA$(255)

Rem Open the file
Open "E:\input.dat" For Input As #1

For I&=0 to 255

  Rem Read line by line from the top of the file
  Input #1, DATA$
  ALLDATA$(I&) = DATA$

  Rem Check if the end of the file is reached
  If Eof(1) <> 0 Then
    Exit For
  Endif
```

```
Next

Rem Close up the file.
Close #1
```

Useable Modules

Unit Calculation Macro / Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 3.50 or later

Related Items

Close (Reference: ▶ Details (p.432))
Input# (Reference: ▶ Details (p.569))
Open For Append As# (Reference: ▶ Details (p.627))
Open For Input As# (Reference: ▶ Details (p.629))
Open For Output As# (Reference: ▶ Details (p.631))

Erase

Releases array variable.

Format

Erase <array>[, <array>...]

Parameter

Parameter name	Data type	Description
<array>	---	Released array variable

Return value

None.

Description

Releases the allocated memory area of the predefined array variable with the Dim function that is specified in the <array> parameter. By releasing the temporarily used array variables with this function, the allocated memory areas of the variables can be released so that the released memory areas can be efficiently utilized.
If an array variable is redefined without being released, its allocated memory area is released before the redefinition.
If variables other than array variables are specified in the <array> parameter, "Syntax error" will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

Releases defined array.

```
Dim XY&(3)
Dim XY#(7, 15)
Dim CHARA$(31, 63, 127, 255)

Rem Releases array variable.
Erase XY&, XY#(), CHARA$()
```

Useable Modules

Unit Calculation Macro / Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 3.50 or later

Related Items

Dim (Reference: ▶ Details (p.461))

Errcmnd$

Get the function name of the macro where an error occurred.

Format

Errcmnd$

Parameter

None.

Return value

Returns the character string type value of the upper case letters that represents the macro function name where the error occurred.

Description

Gets the character string of the macro function name where the error occurred at error occurrence in the program.
If there is no error in the program or an error occurred as a result of the unrelated operations to the macro functions such as a division by zero operation, a null string is returned.
Use this macro function in the Try-Catch-End Try statement.
If a value is assigned to the return value variable or the variable is not used in an expression, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

Outputs the error information to the system status console window only if the error is occurred in the GetUnitData statement process in the *MEASUREPROC subroutine within the Unit Macro processing unit. If an error is occurred in other macro function statement in the subroutine, the measurement processing ends without taking any action.

```
*MEASUREPROC

  Try
    GetUnitData 5, 5, CR#

    SetUnitData 6, 143, CR#
  Catch
    If Errcmnd$ = "GetUnitData" Then
      Print Errcmnd$
    Endif
  End Try

Return
```

Useable Modules

Unit Calculation Macro / Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 3.50 or later

Related Items

Errno (Reference: ▶ Details (p.515))
Print (Reference: ▶ Details (p.639))
Try Catch End Try (Reference: ▶ Details (p.773))
GetUnitData (Reference: ▶ Details (p.552))
SetUnitData (Reference: ▶ Details (p.731))

Errno

Gets the error number.

Format

Errno

Parameter

None.

Return value

Returns the error number as an integer value. (Reference: ▶Error List (p.320))

Description

Gets the error number of the error which occurred in the program.
Use this macro function in the Try-Catch-End Try statement
If a value is assigned to the return value variable or the variable is not used in an expression, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

Uses the Try Catch-End Try statement in the *MEASUREPROC subroutine of the Unit Macro processing unit to detect the error occurrence and get the detected error number.

```
*MEASUREPROC

  Try
    WORK& = 0
    SUMM& = 100 + 200 + 300

    ANS& = SUMM& / WORK&
  Catch
    If Errno = 11 Then
      Rem Output the error number and the error content on the system status console window
      Print "Error Number = " + Str$(Errno) + ", Division by Zero"
    Endif
  End Try

Return
```

Useable Modules

Unit Calculation Macro / Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 3.50 or later

Related Items

Errcmnd$ (Reference: ▶ Details (p.513))
Try Catch End Try (Reference: ▶ Details (p.773))
Print (Reference: ▶ Details (p.639))

ExecuteImageLogging

Executes image logging.

Format

ExecuteImageLogging <directory>, <header>

Parameter

Parameter name	Data type	Description
<directory>	Character string type	Subdirectory name that saves the logging images
<header>	Character string type	Header string added to the file name of the logged image

Return value

None.

Description

Executes image logging and saves the logged images using the file name (composed from the header character string specified in the <header> parameter and a measurement identification) in the subdirectory specified in the <directory> parameter.
Logged images are saved in the destination folder specified in [System setting] - [Logging setting].
(Reference: ▶Setting Logging Conditions [Logging Setting] (p.115))
If a subdirectory name is specified in the <directory> parameter, the logged file is saved in a subdirectory created under the logged image files destination folder.
If any empty string "" is specified in the <directory> parameter, the subdirectory is not created.
Logged images are saved as a file with the file name composed of measurement identification number and logging file extension .ifz (i.e., YYYY-MM-DD_HH-MM-SS-SSSS.ifz). If a header character string is specified in the <header> parameter, the specified character string is added to the saved file name. If an empty string "" is specified in the <header> parameter, no header character string is added to the file name.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a character string longer than 255 characters is specified for a character string parameter, a "String too long" error will occur.
If a character string longer than 63 characters is specified in the <directory> parameter, the 63-character string before the 64th character is used for the macro function processing. Characters after the 64th character will be discarded.
If a character string longer than 31 characters is specified in the <header> parameter, the 31-character string before the 32nd character is used for the macro function processing. Characters after the 32nd character will be discarded.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- This macro function can only be used in the *MEASUREPROC subroutine. If used in any other subroutines, an error will occur and the function will not be executed.

Example

Uses the MEASUREPROC subroutine in the Unit Macro processing unit to save the logged images using the file names with a header "new_" in the subdirectory named "Image" under the "C:\temp" directory set as the logged files destination in the system settings.

```
*MEASUREPROC

  Rem Execute image logging.
  ExecuteImageLogging "Image", "new_"

  Return
```

After the measurement, a file with file name "new_2012-11-01_13-11-25-0025.ifz" is saved under "C:\temp\Image".

Useable Modules

Unit macro

Supported Versions

Version 3.50 or later

Related Items

GetSystemData (Reference: ▶ Details (p.546))
GetUnitData (Reference: ▶ Details (p.552))
Str$ (Reference: ▶ Details (p.749))
Str2$ (Reference: ▶ Details (p.751))

ExitFzProcess

Terminate the Sensor Controller.

Format

ExitFzProcess

Parameter

None.

Return value

None.

Description

Terminate the FH/FZ5 process execution.
If this macro function macro is executed on the sensor controller, turn OFF the power to the sensor controller after the execution.

Usage Cautions

- None.

Example

After saving the data to the controller, terminates the controller.

```
Rem Carry out the 'Data save'.
SaveData

Rem Terminate the Sensor Controller.
ExitFzProcess
```

Useable Modules

Scene Control Macro / Communication Command Macro

Supported Versions

Version 3.50 or later

Related Items

SaveData (Reference: ▶ Details (p.670))

SystemReset (Reference: ▶ Details (p.754))

Exp

Gets the value of the exponential function of the base e natural logarithm.

Format

Exp(<expression>)

Parameter

Parameter name	Data type	Description
<expression>	Integer type Double precision real number data type	Expression to get the exponential value

Return value

Returns the calculated exponent (power) as a double precision real number value.

Description

Calculates the exponent (power) of the base e natural logarithm of the expression specified in the <expression> parameter.
The Exp function is the inverse function of the Log function. The Exp function can be used to derive other mathematical functions, such as the hyperbolic sine function.
In the <expression> parameter, specify a value no greater than 21.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a non-existent number, numerical value, or combination of data types or values is specified for a parameter, an "Illegal function call" error will occur.
If a value is assigned to the return value variable or the variable is not used in an expression, a "Syntax error" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

Gets the values of the hyperbolic sine function and hyperbolic cosine function of the value TH&.

```
SINH& = (Exp(TH&) - Exp(-TH&)) / 2
COSH& = (Exp(TH&) + Exp(-TH&)) / 2
```

Useable Modules

Unit Calculation Macro / Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 3.50 or later

Related Items

Cos (Reference: ▶ Details (p.450))
Log (Reference: ▶ Details (p.609))
UnitData (Reference: ▶ Details (p.778))
GetUnitData (Reference: ▶ Details (p.552))
Sin (Reference: ▶ Details (p.743))

Fcopy

Copies the file.

Format

Fcopy <srcPath>, <dstPath>

Parameter

Parameter name	Data type	Description
<srcPath>	Character string type	Absolute path of the original file to be copied
<dstPath>	Character string type	Absolute path for the copy destination file

Return value

None.

Description

Copies the file of the file name specified in the <srcPath> parameter as file of the file name specified in the <dstPath> parameter.
In the <srcPath> parameter and the <dstPath> parameter, specify with the absolute path the copy source file and the file name of the copy destination file.
Overwrite if the copy destination file already exists. If it does not exist, create a new one.
In the following cases, the file cannot be copied.

- The original file to copy from does not exist.
- The destination directory does not exist.
- The external memory has not been inserted.
- There is insufficient free space on the external memory.

If only the directory name is specified in the <srcPath> parameter or the <dstPath> parameter, an "Illegal function call" error will occur.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a character string longer than 255 characters is specified for a character string parameter, a "String too long" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

Copies a file named "1280-720.bmp" under the directory "E:\" to the directory "F:\"

```
Fcopy "E:\1280-720.bmp", "F:\1280-720.bmp"
```

Useable Modules

Unit Calculation Macro / Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 3.50 or later

Related Items

Dskf (Reference: ▶ Details (p.506))
Kill (Reference: ▶ Details (p.587))
Rmdir (Reference: ▶ Details (p.666))
IsFile (Reference: ▶ Details (p.577))
Mkdir (Reference: ▶ Details (p.623))

Fix

Gets the integer of a value by rounding off digits to the right of the decimal point.

Format

Fix(<expression>)

Parameter

Parameter name	Data type	Description
<expression>	Double precision real number data type	Expression to truncate after the decimal point

Return value

Returns an integer value gotten by rounding off digits to the right of the decimal point.

Description

Gets the integer value of the expression specified in the <expression> parameter by rounding off digits to the right of the decimal point.
If a negative value is specified in the <expression> parameter, the Fix function will return the least negative integer value greater than the specified negative value. This contrasts with the Int function that returns the greatest negative value that does not exceed the specified negative value. For example, Int(-7.2) returns -8 and Fix(-7.2) returns -7.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a value is assigned to the return value variable or the variable is not used in an expression, a "Syntax error" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

Changes the double precision real number value of a measurement result to an integer by rounding off digits to the right of the decimal point.

```
NUMBER1& = Fix(9.7)
NUMBER2& = Fix(-9.7)
NUMBER3& = Fix(-9.2)
```

The result is shown below.

```
NUMBER1& = 9
NUMBER2& = -9
NUMBER3& = -9
```

Useable Modules

Unit Calculation Macro / Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 3.50 or later

Related Items

Int (Reference: ▶ Details (p.575))
UnitData (Reference: ▶ Details (p.778))
GetUnitData (Reference: ▶ Details (p.552))

For To Step Next

Repeats the statements between the For and Next statements.

Format

For <variable> = <startValue> To <endValue>[Step <increment>]
<statement>
Next[<variable>]

Parameter

Parameter name	Data type	Description
<variable>	Integer type	Loop counter variable of repetition process
<startValue>	Integer type	Initial value of loop counter variable
<endValue>	Integer type	Loop counter variable value that end up repetition process
<increment>	Integer type	Increment of the loop counter variable
<statement>	Integer type	Statement to be executed repeatedly

Return value

None.

Description

Repeats the specified For block statement in the <statement> parameter until the loop counter variable specified in the <variable> parameter reaches the <endValue> parameter value. The loop counter value starts from the <startValue> parameter value. Every repeating process increments the loop counter value by the <increment> parameter value.
If the <increment> parameter is omitted, the every repeating process increments the loop counter variable value by one.
If the Exit For statement is used in the For block statement, the statement force stops the repeating operation immediately.
If the program process is jumped into or out of the For block statement using the Goto or Gosub function, the resulting operation may be unpredictable.
If neither the For statement nor the Next statement is used, either the "NEXT without FOR", "FOR without NEXT", or "EXIT without FOR" error will occur depending on the statement that is used.

If the Next statement is not followed by the For statement, the "NEXT without FOR" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

Uses the *MEASUREPROC subroutine in the Unit Macro processing unit to set the gotten the edge position X values with the edge position processing units (Processing Unit numbers 1 to 4) for the upper limits of measure X of the search processing units (Processing Unit numbers 6 to 9) respectively.

```
*MEASUREPROC

  Dim POS#(3)

  Rem Get the measurement result.
  GetUnitData 1, 5, POS#(0)
  GetUnitData 2, 5, POS#(1)
  GetUnitData 3, 5, POS#(2)
  GetUnitData 4, 5, POS#(3)

  For NUM& = 0 To 3
    Rem Set the setting data
    SetUnitData NUM&+6, 136, POS#(NUM&)
  Next

Return
```

Useable Modules

Unit Calculation Macro / Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 3.50 or later

Related Items

Do Loop While (Reference: ▶ Details (p.465))
GetUnitData (Reference: ▶ Details (p.552))
SetUnitData (Reference: ▶ Details (p.731))

GetAll

Gets the input states of all input terminals.

Format

GetAll(<ioIdent>)

Parameter

Parameter name	Data type	Description
<ioIdent>	Character string type	Identification name of the communication module to be used (always "ParallelIo") (Reference: ▶List of I/O Modules (p.341))

Return value

Returns the input states of all input terminals as integer values.
The input state of each input terminal is expressed as an integer value (OFF (0) or ON (1)) in each digit of a character string in binary notation.
In parallel I/O, integer values are returned expressing DI0 to DI7 in the 1st digit to the 8th digit.
Example: When DI0 to DI5 are ON and DI6 to DI7 are OFF

- Binary notation: 0011 1111
- Value of input states that can be gotten: 63

Description

Gets the input states of all input terminals of the communication module specified in the <ioIdent> parameter.
Normally "ParallelIo" should be specified in the <ioIdent> parameter.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a value is assigned to the return value variable or the variable is not used in an expression, a "Syntax error" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

In the communication command macro, gets the input state of DI in parallel I/O.

```
IOMODULE$ = "ParallelIo"

Rem Get the input state.
STATE& = GetAll(IOMODULE$)
```

Useable Modules

Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 3.50 or later

Related Items

BusyOut (Reference: ▶ Details (p.419))
JudgeOut (Reference: ▶ Details (p.585))
PutPort (Reference: ▶ Details (p.645))
GetPort (Reference: ▶ Details (p.542))
PutAll (Reference: ▶ Details (p.643))
RunOut (Reference: ▶ Details (p.667))

GetGlobalData

Gets the global data.

Format

GetGlobalData <dataIdent>, <data>

Parameter

Parameter name	Data type	Description
<dataIdent>	Character string type	Identification name of the global data to get
<data>	Integer type Double precision real number data type Character string type	Gotten data

Return value

None.

Description

Gets the value of the global data that has the identification name specified in the <dataIdent> parameter. The value of the gotten global data is converted to the specified variable type and stored in the <data> parameter.
If a character string that cannot be converted to a numerical value is gotten and an integer or double precision variable is specified for the <data> parameter, 0 is stored in the gotten data.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a non-existent number, numerical value, or combination of data types or values is specified for a parameter, an "Illegal function call" error will occur.
If a character string longer than 255 characters is specified in the <dataIdent> parameter, a "String too long" error will occur.
If a constant is specified for the <data> parameter, a "Syntax error" error will result.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

Gets the integer value set in the global data that has the identification name "ABC".

```
Rem Set the integer value 1 in the "ABC" global data value.
SetGlobalData "ABC", 1

Rem Get the value of the global data "ABC" and store in the integer variable DATA&.
GetGlobalData "ABC", DATA&
```

Useable Modules

Unit Calculation Macro / Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 3.50 or later

Related Items

AddGlobalData (Reference: ▶ Details (p.405))

SetGlobalData (Reference: ▶ Details (p.702))

GetImageSize

Gets the image size of the processing unit image.

Format

GetImageSize <unitNo>, <measureImageNo>, <sizeX>, <sizeY>

Parameter

Parameter name	Data type	Description
<unitNo>	Integer type	Processing unit number (0 to (the number of registered processing units in the current scene minus one))
<measureImageNo>	Integer type	Image number of the current image to be gotten (Reference: ▶List of Figure Numbers (p.366))
<sizeX>	Integer type	X size of the gotten image
<sizeY>	Integer type	Y size of the gotten image

Return value

None.

Description

Gets the size of the image data of the image number specified in the <measureImageNo> parameter, held by the processing unit specified in the <unitNo> parameter.
In <sizeX> parameter and <sizeY> parameter, specify variables to store the gotten image size.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a non-existent number, numerical value, or combination of data types or values is specified for a parameter, an "Illegal function call" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- Use this macro function with the measurement image displayed after one or more measurements, or after the image file is specified and re-measured.

Example

Gets the size of the image of Image number 0 in Processing Unit number 2.

```
GetImageSize 2, 0, SIZEX&, SIZEY&
```

Useable Modules

Unit Calculation Macro / Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 3.50 or later

Related Items

SaveMeasureImage (Reference: ▶ Details (p.673)) UnitNo (Reference: ▶ Details (p.790))
Ut (Reference: ▶ Details (p.793))

GetImageWindow

Get the state of the image window.

Format

[Scene Control Macro / Communication Command Macro]
GetImageWindow <windowNo>, <locationX>, <locationY>, <width>, <height>, <unitNo>, <subNo>, <magnification>, <originX>, <originY>, <update>, <visible>

[Unit Macro]
GetImageWindow <locationX>, <locationY>, <width>, <height>, <unitNo>, <subNo>, <magnification>, <originX>, <originY>, <update>, <visible>

Parameter

Parameter name	Data type	Description
<windowNo>	Integer type	Image window number to get the status
<locationX>	Integer type	Upper left X coordinate value of the image window
<locationY>	Integer type	Upper left Y coordinate value of the image window
<width>	Integer type	Width of the image window
<height>	Integer type	Height of the image window
<unitNo>	Integer type	Processing unit number of the target processing unit to display
<subNo>	Integer type	Sub-image number of the target image to display
<magnification>	Double precision real number data type	Display magnification
<originX>	Integer type	Upper left X coordinate of the image display relative to the upper left coordinate of the image window.
<originY>	Integer type	Upper left Y coordinate of the image display relative to the upper left coordinate of the image window
<update>	Integer type	Update timing of image window 0: Every measurement (Image mode Freeze) 1: Only when an overall judgement result is NG at the time of measurement (Last NG image). 2: Only when a target processing unit is NG at the time of measurement. 3: Always updated (through display)
<visible>	Integer type	Setting of whether to display 0: Window invisible 1: Window visible

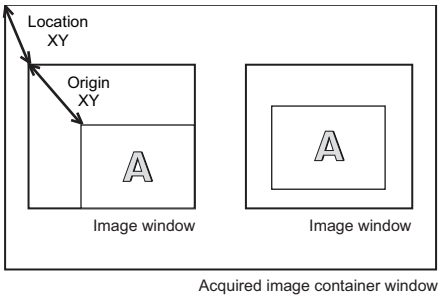
Return value

None.

Description

Gets the state of the image window specified in the <windowNo> parameter. When this macro function is used with the unit macro, the state of the image window displayed using the MEASUREDISPI subroutine is set.
In the <locationX> parameter and <locationY> parameter, specify the variables that store the relative coordinate values from the upper left coordinates of the gotten image container window to the upper left

coordinates of the image window.
In the <width> parameter and <height> parameter, specify the variables that store the values of the gotten image window width and height.
In the <unitNo> parameter, specify the variable that stores the value of the gotten processing unit number that is displayed. When the processing unit displayed in the image window is linked to flow display, -1 is stored.
In the <subNo> parameter, specify the variable that stores the value of the gotten sub image number that is displayed. When the content displayed in the image window is the position list, -1 is stored.
In the <magnification> parameter, specify the variable that stores the value of the display zoom of the gotten image window. When the display zoom is auto, -1 is stored.
In the <originX> parameter and the <originY> parameter, specify the variables that store the values of the relative coordinates from the upper left coordinates of the gotten image window to the upper left coordinates of the displayed image.



In the <update> parameter, specify the variable that stores the value of the image mode of the gotten image window.
In the <visible> parameter, specify the variable that stores the value of the display state of the gotten image window.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- [Unit Macro]
  This macro function can only be used in the *MEASUREDISPI subroutine or the *MEASUREDISPG subroutine. If used in another subroutine, an "Illegal function call" error will occur.

Example

In the communication command macro, changes the image mode of image windows 0 to 3 to Through.
Setting the BusyOn flag to ON in advance in the communication command macro.

```
For I& = 0 To 3

    Rem Get the state of the image window.
    GetImageWindow I&, LOCATIONX&, LOCATIONY&, WIDTH&, HEIGHT&, UNITNO&, SUBNO&,MAG#,
    ORIGINX&, ORIGINY&, UPDATE&, VISIBLE&


    Rem Change the update timing to Through.
    UPDATE& = 3

    Rem Set the state of the image window.
    SetImageWindow I&, LOCATIONX&, LOCATIONY&, WIDTH&, HEIGHT&, UNITNO&, SUBNO&MAG#, ORIGINX&,
    ORIGINY&, UPDATE&, VISIBLE&,

Next
```

Useable Modules

Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 3.50 or later

Related Items

DisplayUnitNo (Reference: ▶ Details (p.464))
SetDisplayUnitNo (Reference: ▶ Details (p.699))
UnitNo (Reference: ▶ Details (p.790))
GetTextWindow (Reference: ▶ Details (p.550))
SetImageWindow (Reference: ▶ Details (p.704))
Ut (Reference: ▶ Details (p.793))

GetMeasureOut

Gets the external output setting for measurement results.

Format

GetMeasureOut

Parameter

None.

Return value

Returns the external output setting as an integer value.
0: Not output externally
1: Output externally

Description

Gets the "External output" setting in the layout settings as the external output setting for measurement results. (Reference: ▶Setting the Behavior of Output Signals for Each Layout (Layout Settings) (p.90))
Even when the measurement result external output setting is 0, data can be output using the SendData function or SendString function in the macro customize function.
If a value is assigned to the return value variable or the variable is not used in an expression, a "Syntax error" error will occur.

Usage Cautions

- Execute this macro function when the BUSY signal or other measurement in progress signal is ON and measurement is prohibited. (Reference: ▶State Transitions and Execution Timing (p.227))

Example

In the scene control macro, outputs the measurement results to an external device when "External Output" is ON, and outputs the measurement results to the system status console window when OFF.

```
Rem Get the read character string of the 2D code processing unit of Processing Unit number 3.
GetUnitData 3, "decodeCharStr", RESULT$

Rem Branch the processing based on the external output setting
If GetMeasureOut = 1 Then
  SendString "TcpNormal", RESULT$
Else
  Print RESULT$
Endif
```

Useable Modules

Scene Control Macro / Communication Command Macro

Supported Versions

Version 3.50 or later

Related Items

SendData (Reference: ▶ Details (p.696))
SendString (Reference: ▶ Details (p.698))
SetMeasureOut (Reference: ▶ Details (p.708))

GetPlcData

Gets data read with the ReadPlcMemory function.

Format

GetPlcData <ioIdent>, <readData()>, <offset>, <size>, <data>

Parameter

Parameter name	Data type	Description
<ioIdent>	Character string type	Identification name of the communication module to be used (Reference: ▶List of I/O Modules (p.341))
<readData()>	Integer array	Loaded data
<offset>	Integer type	Offset from the top of the loaded data to the head of the loaded data to read in (byte unit)
<size>	Integer type	Size of the data to get (byte unit)
<data>	Integer type Double precision real number data type Character string type	Gotten data

Return value

None.

Description

Using the communication module specified in the <ioIdent> parameter, the data size specified in the <size> parameter is gotten from the position that is offset by the value specified in the <offset> parameter from the start of the data array specified in the <readData>parameter.
Use this macro function to get the data after executing the ReadPlcMemory function to read the value in the PLC memory area.
In the <readData()> parameter, specify the 1D integer array variable that stores the data read with the ReadPlcMemory function. Add () without specifying element numbers.
In the <offset> parameter and <size> parameter, specify the offset and size in units of bytes. These units are different from the units specified in the ReadPlcMemory function (channel units).
Specify 2, 4, or 8 in the <size> parameter. These will respectively get a 2 byte integer, 4 byte integer, or 8 byte real number.
In the <data> parameter, specify the variable that will store the gotten data.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a non-existent number, numerical value, or combination of data types or values is specified for a parameter, an "Illegal function call" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- If the value in the PLC memory area is read using the ReadPlcMemory function in PLC link communication, always use this macro function to get the value from the data that is read. If the value is directly gotten from the ReadPlcMemory function parameter without using this macro function, the correct value may not be gotten.

Example

In the communication command macro, reads multiple data from the PLC connected by PLC link.

```
IOMODULE$ = "UdpPlcLink"

Rem Get the settings of the output data area.
GetSystemData IOMODULE$, "outputArea", AREA&
GetSystemData IOMODULE$, "outputMemoryAddress", ADDRESS&

Rem Create the integer array variable to store the read data.
Dim DATA&(1)

Rem Load the data (4ch) from data output area.
ReadPlcMemory IOMODULE$, AREA&, ADDRESS&, 4, DATA&()

Get the values from the read data.
GetPlcData IOMODULE$, DATA&(), 0, 4, VALUE0&
GetPlcData IOMODULE$, DATA&(), 4, 4, VALUE1&
```

Useable Modules

Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 4.20 or later

Related Items

ReadPlcMemory (Reference: ▶ Details (p.649))
WritePlcMemory (Reference: ▶ Details (p.807))
SetPlcData (Reference: ▶ Details (p.709))

GetPollingState

Gets the polling state of the communication module.

Format

GetPollingState(<ioIdent>)

Parameter

Parameter name	Data type	Description
<ioIdent>	Character string type	Identification name of communication module whose polling state is to be gotten (Reference: ▶List of I/O Modules (p.341))

Return value

Returns the polling state as an integer value.

- False: Stopped
- True: Operating

Description

Gets the polling state of the communication module specified in the <ioIdent> parameter.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a non-existent number, numerical value, or combination of data types or values is specified for a parameter, an "Illegal function call" error will occur.
If a value is assigned to the return value variable or the variable is not used in an expression, a "Syntax error" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

Receives normal TCP communication data in the MEASUREPROC subroutine of the Unit Macro processing unit.

```
Rem Prepare a buffer that can receive 12 bytes of data.
Dim BUFFER&(11)
IOMODULE$ = "TcpNormal"

Rem Set the polling state of the communication module to stopped in order to receive the data.
SetPollingState IOMODULE$, False

Rem Executing the initialization of the reception data size.
SIZE& = 0

Repeat the reception process until the data has been received.
Try

  Do
```

```
    Rem Attempting the data reception.
    ReceiveData IOMODULE$, BUFFER&(), 12, SIZE&

    Rem Once the data has been received, display the data size in the system status console window.
    If(SIZE& > 0) Then
      Print "Received data size = " + Str$(SIZE&)
    Endif

  Loop While SIZE& = 0

  Rem Data has been received, so set the polling state of the communication module to running.
  SetPollingState IOMODULE$, True

Catch

  Rem Return the polling state of the stopped communication module to running.
  If GetPollingState(IOMODULE$) = False Then
    SetPollingState IOMODULE$, True
  Endif

End Try
```

Useable Modules

Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 4.20 or later

Related Items

SendData (Reference: ▶ Details (p.696))
SetPollingState (Reference: ▶ Details (p.711))
SendString (Reference: ▶ Details (p.698))
ReceiveData (Reference: ▶ Details (p.651))

GetPort

Gets the input state of the specified input terminal.

Format

GetPort(<ioIdent>, <portNo>)

Parameter

Parameter name	Data type	Description
<ioIdent>	Character string type	Identification name of the communication module to be used (always "ParallelIo") (Reference: ▶List of I/O Modules (p.341))
<portNo>	Integer type	Terminal number of input terminal whose input state is to be gotten. Parallel I/O • FH DI0 to DI7: 0 to 7 DSA N: 100 + N x 8 (N: Line number (0 to 7)) STEP N: 101+N x 8 DILINE0 to DILINE2: 200 to 202 • FZ5 DI0 to DI7: 0 to 7 DSA0: 100 STEP0: 101 DSA1: 102 STEP1: 103

Return value

Returns the input state of the input terminal as an integer value.

- 0: Input OFF state
- 1: Input ON state

Description

Gets the state of the input terminal of the terminal number specified in the <portNo> parameter of the communication module specified in the <ioIdent> parameter.
Normally "ParallelIo" should be specified in the <ioIdent> parameter.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a value is assigned to the return value variable or the variable is not used in an expression, a "Syntax error" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

In the communication command macro, gets the input state of DI7 of parallel I/O.

```
IOMODULE$ = "ParallelIo"

Rem Get the input state.
STATE& = GetPort(IOMODULE$, 7)
```

Useable Modules

Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 3.50 or later

Related Items

BusyOut (Reference: ▶ Details (p.419))
JudgeOut (Reference: ▶ Details (p.585))
PutPort (Reference: ▶ Details (p.645))
GetPort (Reference: ▶ Details (p.542))
PutAll (Reference: ▶ Details (p.643))
RunOut (Reference: ▶ Details (p.667))

GetSceneData

Gets data related to the scene control macro.

Format

GetSceneData<dataIdent>,<data>

Parameter

Parameter name	Data type	Description
<dataIdent>	Character string type	Identification name of data to be gotten
<data>	Integer type Double precision real number data type Character string type	Gotten data

Return value

None.

Description

Gets the data identified by the identification name specified in the <dataIdent> parameter.
In the <data> parameter, specify the variable that will store the gotten data.
If the data type of the data to be gotten is different from the data type of the variable specified in the <data> parameter, the gotten data will be converted to the data type of the <data> parameter.
In the <dataIdent> parameter, specify the variable name to be used in the scene control macro program.
If an identification name that does not exist is specified as the parameter, an "Illegal function call" error will occur.
Even if a combination of different data types is specified as parameters, an error will not occur.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

Uses the unit macro to get the integer variables "SearchResult0&" and "SEARCHRESULT1&" defined in the scene control macro

```
Rem Get the value of the variable that has been defined in the scene control macro.
GetSceneData "SEARCHRESULT0&", RET0&
GetSceneData "SEARCHRESULT1&", RET1&

Rem Calculate the result based on the gotten variable value
RESULTDATA& = (RET0& + RET1&) / 2
```

Useable Modules

Unit Calculation Macro / Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 5.20 or later

Related Items

SetSceneData (Reference: ▶ Details (p.713))

GetSystemData

Gets the system data.

Format

GetSystemData <dataIdent0>, <dataIdent1>, <data>

Parameter

Parameter name	Data type	Description
<dataIdent0>	Character string type	Data identification name of identification information 0 of system data to be gotten
<dataIdent1>	Character string type	Data identification name of identification information 1 of system data to be gotten
<data>	Integer type Double precision real number data type Character string type	Value of the gotten system data

Return value

None.

Description

Gets the system data of identification information 1 specified in the <dataIdent1> parameter, which belongs to identification information 0 specified in the <datdaIdent0> parameter. In the <data> parameter, specify the variable that will hold the gotten system data.
For the identification information list, refer to the system data list. (Reference: ▶System Data List (p.327))
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If an identification name that does not exist is specified as the parameter, an "Illegal function call" error will occur.
If a character string longer than 255 characters is specified in the <dataIdent1> parameter, a "String too long" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

Gets the value set for the screen capture destination folder of identification information 1, "captureDirectory", which belongs to the measurement control settings of identification information 0, "Measure", and copies the Sample.bmp file to "C:\temp\bmp".

```
Rem Get the screen capture destination folder that belongs to the measurement control settings.
GetSystemData "Measure", "captureDirectory", DIRNAME$

Rem Get the file name, including the copy destination path, of the file to be copied.
FILE$ = DIRNAME$ + "/Sample.bmp"

Rem Copy the file.
Fcopy FILE$, "C:\temp\bmp\Sample.bmp"
```

Useable Modules

Unit Calculation Macro / Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 3.50 or later

Related Items

AddSystemData (Reference: ▶ Details (p.407))
GetGlobalData (Reference: ▶ Details (p.530))
SetSystemData (Reference: ▶ Details (p.725))
Fcopy (Reference: ▶ Details (p.522))
GetUnitData (Reference: ▶ Details (p.552))

GetText$

Get a text data from a messages file.

Format

GetText$(#<textDataNo>, <textIdent>)

Parameter

Parameter name	Data type	Description
<textDataNo>	Integer type	Text data number of the messages file (0 to 15) that contains the gotten text
<textIdent>	Character string type	Identification name of the text data to be gotten

Return value

Returns an gotten text data value in character string type.

Description

Get the text data of the identification name specified in the <textIdent> parameter, from the message file of the text data number specified in the <textDataNo> parameter.
In the <textDataNo> parameter, specify the speficied text data number in the OpenTextData function that has been used to open the message file.
If a value outside the range from 0 to 15 is specified in the <textDataNo> parameter, an "Illegal function call" error will occur.
If the text data number that has not been opened is specified in the <textDataNo> parameter, an "Illegal function call" error will occur.
Even if an identification name that does not exist is specified in the <textIdent> parameter, an error will not occur. If an identification name that does not exist is specified for a parameter, a text string of "#ERROR" will be returned.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a value outside the range -2147483648 to 2147483647 is specified as an integer parameter, an "Overflow" error will occur.
If a character string longer than 255 characters is specified for a character string parameter, a "String too long" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

Uses the *MEASUREDISPT subroutine of the Unit Macro processing unit to display the measured correlation value by the search processing unit (Processing Unit number 5), along with the gotten text string from the prepared message file for the processing unit, in the text window. The correlation value can be gotten with External Reference Data number 5.

```
*MEASUREDISPT

  Rem Get the measurement result.
  GetUnitData 5, 5, CR#

  Rem Open the messages file
  OpenTextData "Search" As #1

  Rem Get the text
  TEXT$ = GetText$(#1, "Correlation")

  Rem Draw the gotten text string from the messages file without adding any line break on the text window.
  DrawText TEXT$, UnitJudge(5), 0

  Rem Draw the measurement results on the text window.
  DrawText Str2$(CR#, 4, 4, 0, 0), UnitJudge(5), 1

  Rem Close up the messages file.
  CloseTextData

Return
```

The result is shown below.

```
Correlation value:  90.0000
```

Useable Modules

Unit Calculation Macro / Unit Macro

Supported Versions

Version 5.00 or later

Related Items

CloseTextData (Reference: ▶ Details (p.434))
DrawText (Reference: ▶ Details (p.500))
GetUnitData (Reference: ▶ Details (p.552))
OpenTextData (Reference: ▶ Details (p.633))
UnitJudge (Reference: ▶ Details (p.788))

GetTextWindow

Gets the state of the text window.

Format

GetTextWindow <unitNo>, <subNo>, <update>, <visible>

Parameter

Parameter name	Data type	Description
<unitNo>	Integer type	Processing unit number of the target processing unit to display
<subNo>	Integer type	Sub number of the target image to display
<update>	Integer type	Updated timing (always 0)
<visible>	Integer type	Setting of whether to display 0: Window invisible 1: Window visible

Return value

None.

Description

Gets the state of the text window.
In the <unitNo> parameter, specify the variable that stores the value of the gotten processing unit number that is displayed. When the processing unit displayed in the text window is linked to flow display, -1 is stored.
In the <subNo> parameter, specify the variable that stores the value of the gotten sub image number that is displayed.
In the <update> parameter, specify the variable that stores the value of the gotten update timing.
In the <visible> parameter, specify the variable that stores the display state of the gotten text window.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

In the communication command macro, changes the processing unit number of the processing unit displayed in the text display window to the number specified in the communication command argument.

```
Rem Get the state of the text window.
GetTextWindow UNITNO&, SUBNO&, UPDATE&, VISIBLE&

Rem Set the number specified in the command argument in the processing unit number that is displayed.
SetTextWindow argumentValue#(0), SUBNO&, UPDATE&, VISIBLE&
```

Useable Modules

Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 3.50 or later

Related Items

DisplayUnitNo (Reference: ▶ Details (p.464))
SetDisplayUnitNo (Reference: ▶ Details (p.699))
UnitNo (Reference: ▶ Details (p.790))
GetImageWindow (Reference: ▶ Details (p.533))
SetTextWindow (Reference: ▶ Details (p.729))
Ut (Reference: ▶ Details (p.793))

GetUnitData

Gets the data of a processing unit.

Format

GetUnitData <unitNo>, <dataNo>, <data>
GetUnitData <unitNo>, <dataIdent>, <data>

Parameter

Parameter name	Data type	Description
<unitNo>	Integer type	Processing unit number (0 to (the number of registered processing units in the current scene minus one))
<dataNo>	Integer type	External reference data of the processing unit data to get (reference: ▶Vision System FH/FZ5 Series Processing Item Function Reference Manual (Cat. No. Z341)"
<dataIdent>	Character string type	Data identification name of processing unit data to be gotten.
<data>	Integer type Double precision real number data type Character string type	Gotten processing unit data

Return value

None.

Description

Gets the data of the external reference data number specified in the <dataNo> parameter, held by the processing unit specified in the <unitNo> parameter. In the <data> parameter, specify the variable that will store the gotten data.
The data can also be gotten by specifying the <dataIdent> parameter instead of the <dataNo> parameter.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a non-existent number, numerical value, or combination of data types or values is specified for a parameter, an "Illegal function call" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

Gets the judgement result of the processing unit of Processing Unit number 2. The judgement result is external reference data number 0 and external reference data identification name "JG".

```
GetUnitData 2, 0, JUDGE&

Rem The same result can be gotten by specifying "JG" instead of 0.
GetUnitData 2, "JG", JUDGE&
```

Useable Modules

Unit Calculation Macro / Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 3.50 or later

Related Items

GetUnitFigure (Reference: ▶ Details (p.554))
SetUnitFigure (Reference: ▶ Details (p.733))
UnitData$ (Reference: ▶ Details (p.780))
UnitNo (Reference: ▶ Details (p.790))
SetUnitData (Reference: ▶ Details (p.731))
UnitData (Reference: ▶ Details (p.778))
UnitData2 (Reference: ▶ Details (p.782))
Ut (Reference: ▶ Details (p.793))

GetUnitFigure

Gets figure data to the processing unit.

Format

GetUnitFigure <unitNo>, <figureNo>, <figure()>

Parameter

Parameter name	Data type	Description
<unitNo>	Integer type	Processing unit number (0 to (the number of registered processing units in the current scene minus one))
<figureNo>	Integer type	Figure data to get (Reference: ▶List of Figure Numbers (p.366))
<figure()>	Integer array	Gotten figure data (Reference: ▶Figure Data List (p.364))

Return value

None.

Description

Gets the figure data of the figure specified in the <figureNo> parameter, of the processing unit specified in the <unitNo> parameter.
In the <figure()> parameter, specify the 1D integer array variable that will hold the figure data by adding only () without specifying an element number.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a non-existent number, numerical value, or combination of data types or values is specified for a parameter, an "Illegal function call" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

Changes the value of wide arc line thickness if the figure of the edge processing unit of Processing Unit number 2 is a wide arc

```
Dim FIGURE&(10)

Rem Get the figure data of the processing unit.
GetUnitFigure 2, 0, FIGURE&()

Rem Case of width of the wide arc.
If FIGURE&(1) = 256 Then
  Rem Set the thickness value
  FIGURE&(7) = 64
  SetUnitFigure 2, 0, FIGURE&()
Endif
```

Useable Modules

Unit Calculation Macro / Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 3.50 or later

Related Items

GetUnitFigure (Reference: ▶ Details (p.554))
SetUnitFigure (Reference: ▶ Details (p.733))
UnitNo (Reference: ▶ Details (p.790))
SetUnitData (Reference: ▶ Details (p.731))
UnitData (Reference: ▶ Details (p.778))
Ut (Reference: ▶ Details (p.793))

Gosub

Operate the specified subroutine.

Format

Gosub <label>

Parameter

Parameter name	Data type	Description
<label>	Character string type	Label name for the executing subroutine

Return value

None.

Description

Execute the subroutine whose label name is specified in the <label> parameter.
At the Return statement in the end of a subroutine, the program resumes the operation from the next statement of the Gosub statement in the calling subroutine.
If the Return statement will not resume the operation in the original subroutine, use the Goto function.
If the specified label does not exist for a parameter, an "Undefined label" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

Uses the *MCRINIT subroutine in the Unit Macro processing unit to execute the process in the *INITPROC subroutine (defined separately from *MCRINIT).

```
*MCRINIT

  Rem Execute another subroutine
  Gosub *INITPROC

Return

*INITPROC

  Dim DATA$(255)

Return
```

Useable Modules

Unit Calculation Macro / Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 3.50 or later

Related Items

Goto (Reference: ▶ Details (p.558))

Goto

Move the process to the statement line with a specified label.

Format

Goto <label>

Parameter

Parameter name	Data type	Description
<label>	Integer type Character string type	Move destination line number or move destination label name

Return value

None.

Description

Move the process to the specified line number or label name in the <label> parameter.
To return to the calling subroutine where the Return statement is used, use the Gosub function.
If the specified label does not exist for a parameter, an "Undefined label" error will occur.
If the specified line number does not exist for a parameter, an "Undefined line number" error will occur.
If the specified data type in the parameter is incorrect, a "Syntax error" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

Uses the *MEASUREPROC subroutine in the Unit Macro processing unit to move the programming process to the *PROC2 labeled line and output the text string to the system status console window. If the first Goto statement is rewritten to jump the process to the *PROC1 labeled line or the *PROC3 labeled line, the process ends without taking any action. Similarly, if the first Goto statement is skipped, the process moves to the *PROC3 labeled line and the process ends without taking any action.

```
*MEASUREPROC

  Rem Move the process to another line
  Goto *PROC2

  *PROC1
  Goto *PROC3

  *PROC2
  Print "PROC1"

  *PROC3

Return
```

Useable Modules

Unit Calculation Macro / Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 3.50 or later

Related Items

Gosub (Reference: ▶ Details (p.556))

Hex$

Converts the value in the expression to the hexadecimal value in character string format.

Format

Hex$(<expression>)

Parameter

Parameter name	Data type	Description
<expression>	Integer type	Expression to be converted to a character string

Return value

Returns the character string type hexadecimal value.
If the value in the expression is negative, the two's complement hexadecimal value is returned.
(&H) to express a hexadecimal number is not added to the return value.

Description

Converts the value specified in the <expression> parameter to the hexadecimal value in character string format.
To specify a double precision real number type expression in the <expression> parameter, convert the expression to the integer type with Fix or Int in advance.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a value outside the range -2147483648 to 2147483647 is specified as an integer parameter, an "Overflow" error will occur.
If a value is assigned to the return value variable or the variable is not used in an expression, a "Syntax error" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

Gets the read letter "A" with the OCR processing unit (Processing Unit number 5) using the unit calculation macro processing unit and converts the letter into a hexadecimal value. The read character string can be gotten with External Reference Data number 20

```
Rem Get the measurement result of the processing unit.
GetUnitData 5, 20, CHARA1$

Rem Convert the character to the character code.
CODE& = Asc(CHARA1$)

Rem Convert the character code to the hexadecimal value in character string format.
CHARA2$ = Hex$(CODE&)
```

The result is shown below.

```
CHARA1$ = "A"
CODE& = 65
CHARA2$ = 41
```

Useable Modules

Unit Calculation Macro / Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 3.50 or later

Related Items

Asc (Reference: ▶ Details (p.413))
GetUnitData (Reference: ▶ Details (p.552))
Left$ (Reference: ▶ Details (p.591))
Mid$ (Reference: ▶ Details (p.621))
Right$ (Reference: ▶ Details (p.664))
Str2$ (Reference: ▶ Details (p.751))
Val (Reference: ▶ Details (p.794))
Chr$ (Reference: ▶ Details (p.427))
LCase$ (Reference: ▶ Details (p.589))
Len (Reference: ▶ Details (p.593))
Piece$ (Reference: ▶ Details (p.637))
Str$ (Reference: ▶ Details (p.749))
UCase$ (Reference: ▶ Details (p.775))

If Then Else

Controls the process flow according to the specified condition.

Format

If <expression> Then <statement>|<label>[Else <statement>|<label>]

Parameter

Parameter name	Data type	Description
<expression>	---	Logical expression that controls the process flow. (Reference: ▶Operator (p.213))
<statement>	---	Processed statement
<label>	Character string type	Label name for the jump destination

Return value

None.

Description

If the specified condition by the <expression> parameter is true, the specified If block statement by the <statement> parameter is executed. If a label name is specified in the "Then" part statement, the process is jumped to the line with a label name specified in the <label> parameter. If the condition specified in the <expression> parameter is false, the specified Else block with the <statement> parameter in the "Else" part statement is executed. In the similar way as the "Then" part, if a label name is specified in the "Else" part statement, the process is jumped to the line with a label name specified in the <label> parameter.
If the "Else" part statement is omitted, the process flow is controlled only when the specified condition by the <expression> parameter is true.
The Else statement line cannot be broken into multiple parts. Write the If-Else statement in a single line.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

Uses the *MEASUREPROC subroutine in the Unit Macro processing unit to branch the processing depending on the gotten judgement result by the Processing Unit number 0.

```
*MEASUREPROC

  If UnitJudge(0)=JUDGE_OK Then Gosub *OKOUT Else Gosub *NGOUT

Return

*OKOUT
  Print "OK"
Return

*NGOUT
```

```
    Print "NG"
Return
```

Useable Modules

Unit Calculation Macro / Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 3.50 or later

Related Items

Gosub (Reference: ▶ Details (p.556))

Print (Reference: ▶ Details (p.639))

Select Case Case Else End Select (Reference: ▶ Details (p.694))

If Then Elseif Else EndIf (Reference: ▶ Details (p.564))

UnitJudge (Reference: ▶ Details (p.788))

If Then Elseif Else EndIf

Controls the process flow according to the specified condition.

Format

If <expression> Then
<ifStatement>
[Elseif <expression> Then
<elseifStatement>]
:
:
[Else
<elseStatement>]
EndIf

Parameter

Parameter name	Data type	Description
<expression>	---	Logical expression that controls the process flow. (Reference: ▶Operator (p.213))
<ifStatement>	---	Executed statement if the following logical expression after the If statement is true
<elseifStatement>	---	Executed statement if the following logical expression after the Elseif statement is true
<elseStatement>	---	Executed statement if all logical expressions in the statement are false.

Return value

None.

Description

If the logical expression specified in the <expression> parameter is true, the specified statement by the <statement> parameter in the If block or multiple of Elseif blocks is executed.
If the logical expression specified in the <expression> parameter in the "If" part statement is true, the If block statement specified in the <ifStatement> parameter is executed.
If the logical expression specified in the <expression> parameter in the "If" part statement is false, and the logical expression specified in the <expression> parameter in the "Elseif" part statement is true, the Elseif block statement specified in the <elseifStatement> parameter is executed.
If all logical expressions specified in the <expression> parameter are false, the specified Else block statement by the <elseStatement> parameter is executed.
If there are multiple of If, Elseif, and Else statements having a true logical expression within the statement, only the first statement from the beginning of the block statement having a true logical expression is executed.
ElseIf block statements and Else block statement are optional.
If the program process is jumped into or out of the Do block statement using the Goto function in a statement, unexpected operation may occur.
If neither the If statement nor the EndIf statement is used, either the "ELSEIF without IF", "ELSE without IF", "ENDIF without IF", "IF without ENDIF", "ELSEIF without ENDIF", or "ELSE without ENDIF" error will occur depending on the statement that is used.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

Uses the *MEASUREDISPG subroutine in the Unit Macro processing unit to output the measured correlation value with the search processing unit (Processing Unit number 1) to the system status console window.

```
*MEASUREDISPT

  Rem Get the measurement result.
  GetUnitData 1, 5, RESULT&

  Rem Branch the process according the measured value
  If RESULT&>=80 Then
    DrawTextG "Excellent", 100, 100, 0
  Elseif RESULT&>=60 Then
    DrawTextG "Good", 100, 100, 0
  Else
    DrawTextG "Bad", 100, 100, 0
  EndIf

Return
```

Useable Modules

Unit Calculation Macro / Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 3.50 or later

Related Items

DrawTextG (Reference: ▶ Details (p.502))
If Then Else (Reference: ▶ Details (p.562))
Select Case Case Else End Select
(Reference: ▶ Details (p.694))
GetUnitData (Reference: ▶ Details (p.552))

ImageFormat

Gets the image format of the image in the processing unit.

Format

ImageFormat(<unitNo>, <measureImageNo>)

Parameter

Parameter name	Data type	Description
<unitNo>	Integer type	Processing unit number (0 to (the number of registered processing units in the current scene minus one))
<measureImageNo>	Integer type	Measurement image number of image format to be gotten (always 0)

Return value

Returns the image format as an integer value.

- 0: Binary image
- 1: Monochrome image
- 2: Color image
- -1: Invalid image

Description

Gets the format of the image of the image number specified in the <measureImageNo> parameter, of the processing unit specified in the <unitNo> parameter.
Normally 0 should be specified in the <measureImageNo> parameter.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a non-existent number, numerical value, or combination of data types or values is specified for a parameter, an "Illegal function call" error will occur.
If a value outside the range -2147483648 to 2147483647 is specified as an integer parameter, an "Overflow" error will occur.
If a value is assigned to the return value variable or the variable is not used in an expression, a "Syntax error" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

In the *MEASUREDISPT subroutine of the unit macro, displays the character string corresponding to the image format in the text window.

```
*MEASUREDISPT
  Rem Get the judgement result of the processing unit.
  JUDGE& = UnitJudge(UnitNo)

  Rem Get the image format.
  FORMAT& = ImageFormat(UnitNo, 0)

  If FORMAT& = 2 Then
    Rem If the format is a color image, display "Color".
    DrawText "Color", JUDGE&, 1

  Elseif FORMAT& = 1 Then
    Rem If the format is a monochrome image, display "Monochrome".
    DrawText "Monochrome", JUDGE&, 1
  EndIf
Return
```

Useable Modules

Unit Calculation Macro / Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 3.50 or later

Related Items

DrawText (Reference: ▶ Details (p.500))
UnitNo (Reference: ▶ Details (p.790))
UnitJudge (Reference: ▶ Details (p.788))
Ut (Reference: ▶ Details (p.793))

ImageUpdate

Updates the image input from the camera.

Format

ImageUpdate

Parameter

None.

Return value

None.

Description

Updates the images of processing units related to image input in the measurement flow and processing units related to image conversion to the image that uses the most recent image from the camera.

Usage Cautions

- None.

Example

In the communication command macro, updates an image displayed in "Camera Image Freeze" image mode to the most recent image.

```
Rem Update the freeze image to the most recent image.
ImageUpdate

Rem Apply the updated image to the display in the image window.
RefreshImageWindow
```

Useable Modules

Scene Control Macro / Communication Command Macro

Supported Versions

Version 3.50 or later

Related Items

GetImageWindow (Reference: ▶ Details (p.533))
SetImageWindow (Reference: ▶ Details (p.704))
RefreshImageWindow (Reference: ▶ Details (p.653))

Input#

Reads data from the file.

Format

Input# <fileNo>, <data>[, <data>...]

Parameter

Parameter name	Data type	Description
<fileNo>	Integer type	Read file number (0 to 15)
<data>	Integer type Double precision real number data type Character string type Array	Loaded data

Return value

None.

Description

Reads the comma separated data in a separated line by a line break code within the specified file number by the <fileNo> parameter.
Specify the variable to store the read data in the <data> parameter.
The value of the read global data is converted to the specified variable type and stored in the <data> parameter. If a character string that cannot be converted to a numerical value is read and an integer or double precision variable is specified for the <data> parameter, 0 is stored in the read data.
If the read data contains a double quoted character string, the double quotation marks (") are also read as characters.
If an unopened file number is specified in the <fileNo> parameter, an "Illegal function call" error will occur.
If a file number of the opened file by a macro function other than the Open For Input As# function is specified in the <fileNo> parameter, an "Illegal function call" error will occur.
If a value outside the range of 0 to 15 is specified in the <fileNo> parameter, an "Illegal function call" error will occur.
If the number of variables specified in the <data> parameter and the number of data to be read do not match, an "Illegal function call" error will occur.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a value outside the range -2147483648 to 2147483647 is specified as an integer parameter, an "Overflow" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

Reads the data until the end of the file.

```
Dim ALLDATA$(255)

Rem Open the file
Open "E:\input.dat" For Input As #1

For I&=0 to 255

  Rem Read line by line from the top of the file
  Input #1, DATA$
  ALLDATA$(I&) = DATA$

  Rem Check if the end of the file is reached
  If Eof(1) <> 0 Then
    Exit For
  Endif

Next

Rem Close up the file.
Close #1
```

Useable Modules

Unit Calculation Macro / Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 3.50 or later

Related Items

Close (Reference: ▶ Details (p.432))
Input$ (Reference: ▶ Details (p.571))
Open For Input As# (Reference: ▶ Details (p.629))
Eof (Reference: ▶ Details (p.510))
Line Input# (Reference: ▶ Details (p.595))

Input$

Reads binary data from the file.

Format

Input$(<length>[, #<fikeNo>])

Parameter

Parameter name	Data type	Description
<length>	Integer type	Bytes number (0 to 255) of read data
<fileNo>	Integer type	Read file number (0 to 15)

Return value

Returns the read binary data value in the character string format.

Description

Reads specified size of binary data whose read size is specified in the <length> parameter from the file with the specified file number in the <fileNo> parameter.
If a larger byte number than 255 is specified in the <length> parameter, the 255 byte data from the beginning is read.
If a less byte number is specified in the <length> parameter than the byte number of data in the file to be read, only the specified inte number of data is read. Unread data after the specified read range is read next time the Input$ function is executed.
If the read data contains a double quoted character string, the double quotation marks (") are also read as characters.
If a larger value than the byte number of all read data is specified in the <length> parameter, an "Illegal function call" error will occur.
If an unopened file number is specified in the <fileNo> parameter, an "Illegal function call" error will occur.
If a file number of the opened file by a macro function other than the Open For Input As# function is specified in the <fileNo> parameter, an "Illegal function call" error will occur.
If a value outside the range of 0 to 15 is specified in the <fileNo> parameter, an "Illegal function call" error will occur.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a value outside the range -2147483648 to 2147483647 is specified as an integer parameter, an "Overflow" error will occur.
If a character string longer than 255 characters is specified for a character string parameter, a "String too long" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

Reads 6 bytes of binary data from the file

```
Rem Open the file
Open "E:\input.dat" For Input As #1

Rem Read 6 bytes of data from the file.
DATA$ = Input$(6, #1)

Rem Close up the file.
Close #1
```

Useable Modules

Unit Calculation Macro / Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 3.50 or later

Related Items

Close (Reference: ▶ Details (p.432))
Line Input# (Reference: ▶ Details (p.595))
Input$ (Reference: ▶ Details (p.571))
Open For Input As# (Reference: ▶ Details (p.629))

InsertUnit

Inserts a processing unit.

Format

InsertUnit <unitNo>, <itemIdent>

Parameter

Parameter name	Data type	Description
<unitNo>	Integer type	Processing unit number on the measurement flow to insert the processing unit (0 to the number of registered processing units in the current scene)
<itemIdent>	Character string type	Identification name of the processing item to insert the processing unit

Return value

None.

Description

Registers the processing item with the identification name specified in the <itemIdent> parameter as a processing unit in the position in the measurement flow specified in the <unitNo> parameter.
The processing unit numbers of processing units after the processing unit number specified in the <unitNo> parameter are moved down by 1.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a non-existent number, numerical value, or combination of data types or values is specified for a parameter, an "Illegal function call" error will occur.
If an identification name that does not exist is specified as the <itemIdent> parameter, an "Illegal function call" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- Execute this macro function when the BUSY signal or other measurement in progress signal is ON and measurement is prohibited. (Reference: ▶State Transitions and Execution Timing (p.227))

Example

Inserts the search processing unit between Processing Unit number 2 and Processing Unit number 3.

```
InsertUnit 3, "Search"
```

Useable Modules

Scene Control Macro / Communication Command Macro

Supported Versions

Version 3.50 or later

Related Items

AssignUnit (Reference: ▶ Details (p.415))
CopyUnit (Reference: ▶ Details (p.442))
MeasureStart (Reference: ▶ Details (p.617))
MoveUnit (Reference: ▶ Details (p.625))
CheckUnit (Reference: ▶ Details (p.425))
DeleteUnit (Reference: ▶ Details (p.460))
MeasureStop (Reference: ▶ Details (p.619))
UnitCount (Reference: ▶ Details (p.777))

Int

Converts numeric value to integer value.

Format

Int(<expression>)

Parameter

Parameter name	Data type	Description
<expression>	Double precision real number data type	Expression to get the integer value

Return value

Returns an integer value.

Description

Rounds off digits to the right of the decimal point in the expression specified in the <expression> parameter, and converts the value to the maximum integer value that does not exceed the value of the specified expression.
If a negative value is specified in the <expression> parameter, the Int function will return the greatest negative value that does not exceed the specified negative value. This contrasts with the Fix function that returns the least negative integer value greater than the specified negative value. For example, Fix(-7.2) returns -7 and Int(-7.2) returns -8.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a value is assigned to the return value variable or the variable is not used in an expression, a "Syntax error" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

Changes the double precision real number value of a measurement result to an integer by rounding off digits to the right of the decimal point.

```
NUMBER1& = Int(9.7)
NUMBER2& = Int(-9.7)
NUMBER3& = Int(-9.2)
```

The result is shown below.

```
NUMBER1& = 9
NUMBER2& = -10
NUMBER3& = -10
```

Useable Modules

Unit Calculation Macro / Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 3.50 or later

Related Items

Fix (Reference: ▶ Details (p.524))
GetUnitData (Reference: ▶ Details (p.552))
UnitData (Reference: ▶ Details (p.778))

Isfile

Checks the attribute and the existence of the file.

Format

Isfile(<fileName>)

Parameter

Parameter name	Data type	Description
<fileName>	Character string type	Absolute path of the file to be checked

Return value

Returns integer values representing the attribute and the existence of the file.

- 0: There is no file
- 1: File attribute is "file"
- 2: File attribute is "directory"

Description

Checks the attribute and the existence of the file specified in the <fileName> parameter.
In the <fileName> parameter, use an absolute path to specify the file name of the file to be checked.
If the file in the external memory is specified in the <fileName> parameter without the external memory being inserted to the sensor controller, "0" (There is no file) is returned.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
Even if a character string longer than 255 characters is specified for a character string parameter, an error will not occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

Checks if the file exists before opening it.

```
Rem Check the file existence
If Isfile("E:\input.dat") <> 1 Then

  Print "There is no file"

Endif
```

Useable Modules

Unit Calculation Macro / Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 3.50 or later

Related Items

Dskf (Reference: ▶ Details (p.506))
Fcopy (Reference: ▶ Details (p.522))
GetSystemData (Reference: ▶ Details (p.546))
Kill (Reference: ▶ Details (p.587))
Open For Append As# (Reference: ▶ Details (p.627))
Open For Input As# (Reference: ▶ Details (p.629))
Open For Output As# (Reference: ▶ Details (p.631))
Print (Reference: ▶ Details (p.639))

ItemCount

Gets the number of useable processing item types.

Format

ItemCount

Parameter

None.

Return value

Returns the number of usable processing item types as an integer value.

Description

Gets the number of processing item types that can be used on the sensor controller.
If a value is assigned to the return value variable or the variable is not used in an expression, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

Gets the number of processing item types, and by repeated processing, searches for search processing items and gets the processing item type of each search processing item.

```
Rem Get the number of the available processing items.
INUM& = ItemCount

Rem Search for search processing items a number of times equal to the number of processing items
For I&=0 To INUM&-1
  If ItemIdent$(I&) = "Search" Then
    Rem Get the processing item type of the search processing item
    FIGNAME& = ItemInfo(I&, 0)
  Endif
Next
```

Useable Modules

Scene Control Macro / Communication Command Macro

Supported Versions

Version 3.50 or later

Related Items

ItemIdent$ (Reference: ▶ Details (p.580))
ItemInfo (Reference: ▶ Details (p.582))
ItemTitle$ (Reference: ▶ Details (p.584))

ItemIdent$

Gets the identification name of the processing item.

Format

ItemIdent$(<itemNo>)

Parameter

Parameter name	Data type	Description
<itemNo>	Integer type	Processing item number to get the identification name of the processing item

Return value

Returns the value of the processing item identification name as a character string.

Description

Gets the identification name of the processing item with the processing item number specified in the <itemNo> parameter.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
Even if a non-existent number, numerical value, or combination of data types or values is specified for the parameter, an error will not occur.
If a value is assigned to the return value variable or the variable is not used in an expression, a "Syntax error" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

Gets the number of processing item types, and by repeated processing, searches for search processing items and gets the processing item type of each search processing item.

```
Rem Get the number of the available processing items.
INUM& = ItemCount

Rem Search for search processing items a number of times equal to the number of processing items
For I&=0 To INUM&-1
  If ItemIdent$(I&) = "Search" Then
    Rem Get the processing item type of the search processing item
    FIGNAME& = ItemInfo(I&, 0)
  Endif
Next
```

Useable Modules

Scene Control Macro / Communication Command Macro

Supported Versions

Version 3.50 or later

Related Items

ItemCount (Reference: ▶ Details (p.579))
ItemInfo (Reference: ▶ Details (p.582))
ItemTitle$ (Reference: ▶ Details (p.584))

ItemInfo

Gets the processing item information.

Format

ItemInfo(<itemNo>, <kind>)

Parameter

Parameter name	Data type	Description
<itemNo>	Integer type	Processing item number to get the information
<kind>	Integer type	Type of information 0: Processing item type Number that indicates the processing item type. The values below can be gotten. 0: Inspect and Measure (measurement) 1: Input image (Image input) 2: Compensate image (image correction) 3: Support Inspection and Measurement (supplementary measurement) 4: Branch (branch control) 5: Output result (result output) 6: Display result (result display)

Return value

Returns processing item information as an integer. Returns -1 if information does not exist.

Description

Gets the information specified in the <kind> parameter of the processing item specified in the <itemNo> parameter.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
Even if a non-existent number, numerical value, or combination of data types or values is specified for the parameter, an error will not occur.
If a value is assigned to the return value variable or the variable is not used in an expression, a "Syntax error" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

Gets the number of processing item types, and by repeated processing, searches for search processing items and gets the processing item type of each search processing item.

```
Rem Get the number of the available processing items.
INUM& = ItemCount

Rem Search for search processing items a number of times equal to the number of processing items
For I&=0 To INUM&-1
  If ItemIdent$(I&) = "Search" Then
    Rem Get the processing item type of the search processing item
    FIGNAME& = ItemInfo(I&, 0)
  Endif
Next
```

Useable Modules

Scene Control Macro / Communication Command Macro

Supported Versions

Version 3.50 or later

Related Items

ItemCount (Reference: ▶ Details (p.579))
ItemIdent$ (Reference: ▶ Details (p.580))
ItemTitle$ (Reference: ▶ Details (p.584))
UnitInfo (Reference: ▶ Details (p.784))

ItemTitle$

Gets the processing item title.

Format

ItemTitle$(<itemNo>)

Parameter

Parameter name	Data type	Description
<itemNo>	Integer type	Processing item number to get the title name

Return value

Returns the title as a character string.

Description

Gets the title of the processing item specified in the <itemNo> parameter.
The title can be gotten in a language based on the language setting.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
Even if a non-existent number, numerical value, or combination of data types or values is specified for the parameter, an error will not occur.
If a value is assigned to the return value variable or the variable is not used in an expression, a "Syntax error" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

Gets the number of processing item types, and gets the title of the processing item of each processing item number by repeated processing.

```
Rem Get the number of the available processing items.
INUM& = ItemCount

Rem Repeat the process up to the number of the processing items.
For I&=0 To INUM&-1
  Rem Get the title of the processing item
  TITLE$ = ItemTitle$(I&)
Next
```

Useable Modules

Scene Control Macro / Communication Command Macro

Supported Versions

Version 3.50 or later

Related Items

ItemCount (Reference: ▶ Details (p.579))
ItemIdent$ (Reference: ▶ Details (p.580))
ItemInfo (Reference: ▶ Details (p.582))

JudgeOut

Sets the output state of the overall judgement signal.

Format

JudgeOut <ioIdent>, <state>

Parameter

Parameter name	Data type	Description
<ioIdent>	Character string type	Identification name of the communication module to be used ("ParallelIo" or "EtherCAT") (Reference: ▶List of I/O Modules (p.341))
<state>	Integer type	Total judgement result to be output JUDGE_OK: Total judgement result OK JUDGE_NG: Total judgement result NG The actual signal output depends on this parameter setting and the total judgement (Total Judgement or OR) signal output polarity setting. Specify "ON at NG" for the total judgement signal polarity when using the communication modules other than ParallelIO.

Return value

None.

Description

Sets the output state specified in the <state> parameter of the OR signal or other overall judgement signal of the communication module specified in the <ioIdent> parameter.
Normally "ParallelIo" or "EtherCAT" should be specified in the <ioIdent> parameter.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If an identification name that does not exist is specified in the <ioIdent> parameter, an "Illegal function call" error will occur.
Even if an output status parameter value that does not exist (i.e., other than 0 and 1) is specified in the <state> parameter, an error will not occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

In the communication command macro, sets the OR signal of parallel I/O to ON.

```
IOMODULE$ = "ParallelIo"

Rem Set the output state.
JudgeOut IOMODULE$, 1
```

Useable Modules

Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 3.50 or later

Related Items

BusyOut (Reference: ▶ Details (p.419))
GetPort (Reference: ▶ Details (p.542))
PutAll (Reference: ▶ Details (p.643))
TotalJudge (Reference: ▶ Details (p.761))
GetAll (Reference: ▶ Details (p.528))
RunOut (Reference: ▶ Details (p.667))
PutPort (Reference: ▶ Details (p.645))

Kill

Deletes a file.

Format

Kill <fileName>

Parameter

Parameter name	Data type	Description
<fileName>	Character string type	Absolute path of the file to delete

Return value

None.

Description

Deletes the file of the file name specified in the <fileName> parameter.
In the <fileName> parameter, use an absolute path to specify the file name including the drive name.
Characters * (character string wildcard operator) and ? (single character wildcard operator) can be used as wildcards for the <fileName> specification. These wildcards can be used to specify file names and cannot be used for the directory name specification. Directory names that match the wildcard pattern will not be deleted.
For example, the "Kill "E:\img*" statement will not delete the "M:\img\caputure\" directory.
Wildcards can be used to specify file names in the following manner.

.	Specify all file with the extension.
*	Specify all files.
???.*	Specify the files with file name having an extension with a three-character file root name.
???????	Specify the files with file name having a seven-character file name including the file name extension separator (.).
A*.*	Specify the files with file name having an extension with file name starting with "A".
A*A*A	Specify the files with file name (including file extension) having three and more "A"s.
????*	Specify the files with file name (including file extension) having four and more characters.
????*.*	Specify the files with a file name having a file root name with four or more characters and an extension.
*.??	Specify the file with 2 characters extension name.

In the following cases, the file cannot be deleted.
- The specified file does not exist.
- The external memory has not been inserted.

If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a character string longer than 255 characters is specified for a character string parameter, a "String too long" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

Deletes a file named "1280-720.bmp" under the directory "M:\"

```
Kill "M:\1280-720.bmp"
```

Useable Modules

Unit Calculation Macro / Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 3.50 or later

Related Items

Dskf (Reference: ▶ Details (p.506))
IsFile (Reference: ▶ Details (p.577))
Fcopy (Reference: ▶ Details (p.522))

LCase$

Converts an upper case letter to a lower case letter.

Format

LCase$(<string>)

Parameter

Parameter name	Data type	Description
<string>	Character string type	Character string contains an alphabet to be converted to lower case.

Return value

Returns the case converted character string type value.

Description

Converts the upper case letters in the character strings specified in the <string> parameter to lower case.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a character string longer than 255 characters is specified for a character string parameter, the 255-character string before the 256th character is used for the macro function processing. Characters after the 256th character will be discarded.
If a value is assigned to the return value variable or the variable is not used in an expression, a "Syntax error" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

Converts an upper case letter to a lower case letter.

```
CHARA1$ = "Measurement Result = 100.0(OK)"

Rem Convert the upper case letters in the character strings to lower case.
CHARA2$ = LCase$(CHARA1$)
```

The result is shown below.

```
CHARA2$ = "measurement result = 100.0(ok)"
```

Useable Modules

Unit Calculation Macro / Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 4.20 or later

Related Items

Asc (Reference: ▶ Details (p.413))
Hex$ (Reference: ▶ Details (p.560))
Len (Reference: ▶ Details (p.593))
Piece$ (Reference: ▶ Details (p.637))
Str$ (Reference: ▶ Details (p.749))
UCase$ (Reference: ▶ Details (p.775))
Chr$ (Reference: ▶ Details (p.427))
Left$ (Reference: ▶ Details (p.591))
Mid$ (Reference: ▶ Details (p.621))
Right$ (Reference: ▶ Details (p.664))
Str2$ (Reference: ▶ Details (p.751))
Val (Reference: ▶ Details (p.794))

Left$

Extracts the specified length of characters from the left side of character string.

Format

Left$(<string>, <length>)

Parameter

Parameter name	Data type	Description
<string>	Character string type	Extraction target character string
<length>	Integer type	Length of characters to be extracted (1 to the length of the target character string)

Return value

Returns the character string type value of the extracted character string.

Description

Extracts the specified length in the <length> parameter from the left side of specified character string in the <string> parameter.
Specify the length of characters to be extracted in bytes for the <length> parameter. Each single-byte character (i.e., half-width alphanumeric character and symbol) consumes one byte, whereas each double-byte character consumes two bytes.
If the length specified in the <length> parameter is longer than the length of the character string specified in the <string> parameter, the whole character string in the parameter is extracted.
If 0 or less number is specified in the <length> parameter, an "Illegal function call" error will occur.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a value outside the range -2147483648 to 2147483647 is specified as an integer parameter, an "Overflow" error will occur.
If a character string longer than 255 characters is specified for a character string parameter, the 255-character string before the 256th character is used for the macro function processing. Characters after the 256th character will be discarded.
If a value is assigned to the return value variable or the variable is not used in an expression, a "Syntax error" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

Extracts 11-byte length of characters from the left side of the character string. Because one half-width alphabet consumes single byte, this example extracts 11 characters from the character string.

```
CHARA$ = "Measurement Result"

Rem Extract 11-byte length of characters from the left side of the character string.
TITLE$ = Left$(CHARA$, 11)
```

The result is shown below.

```
TITLE$ = "Measurement"
```

Useable Modules

Unit Calculation Macro / Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 3.50 or later

Related Items

Asc (Reference: ▶ Details (p.413))
Chr$ (Reference: ▶ Details (p.427))
Hex$ (Reference: ▶ Details (p.560))
LCase$ (Reference: ▶ Details (p.589))
Len (Reference: ▶ Details (p.593))
Mid$ (Reference: ▶ Details (p.621))
Piece$ (Reference: ▶ Details (p.637))
Right$ (Reference: ▶ Details (p.664))
Str$ (Reference: ▶ Details (p.749))
Str2$ (Reference: ▶ Details (p.751))
UCase$ (Reference: ▶ Details (p.775))
Val (Reference: ▶ Details (p.794))

Len

Gets the length of the specified character string.

Format

Len(<string>)

Parameter

Parameter name	Data type	Description
<string>	Character string type	Character string to be gotten the length of

Return value

Returns the character string length in bytes as an integer value.

Description

Gets the length of the specified character string in the <string> parameter in bytes.
Each single-byte character (i.e., half-width alphanumeric character and symbol) consumes one byte, whereas each double-byte character consumes two bytes.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
Even if a character string longer than 255 characters is specified for a character string parameter, an error will not occur.
If a value is assigned to the return value variable or the variable is not used in an expression, a "Syntax error" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

Gets the length of the character string "OMRON".

```
CHRLEN& = Len("OMRON")
```

The result is shown below.

```
CHRLEN& = 5
```

Useable Modules

Unit Calculation Macro / Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 3.50 or later

Related Items

Asc (Reference: ▶ Details (p.413))
Hex$ (Reference: ▶ Details (p.560))
Left$ (Reference: ▶ Details (p.591))
Piece$ (Reference: ▶ Details (p.637))
Str$ (Reference: ▶ Details (p.749))
UCase$ (Reference: ▶ Details (p.775))
Chr$ (Reference: ▶ Details (p.427))
LCase$ (Reference: ▶ Details (p.589))
Mid$ (Reference: ▶ Details (p.621))
Right$ (Reference: ▶ Details (p.664))
Str2$ (Reference: ▶ Details (p.751))
Val (Reference: ▶ Details (p.794))

Line Input#

Reads the data of one line from the file.

Format

Line Input #<fileNo>, <data>

Parameter

Parameter name	Data type	Description
<fileNo>	Integer type	Read file number (0 to 15)
<data>	Character string type	Loaded data

Return value

None.

Description

Read the data of one line separated by the line break code from the file of the file number specified in the <fileNo> parameter. The Input# function operates in similar way. This Line Input# function reads all data within a single line as a character string type data, whereas the Input# function reads multiple data within a single line and data type specifications for each piece of read data are required in advance.
Specify the variable to store the read data in the <data> parameter.
When a read line only has a line break code, a null character string is stored in the read data.
If the read data contains a double quoted character string, the double quotation marks (") are also read as characters.
If an unopened file number is specified in the <fileNo> parameter, an "Illegal function call" error will occur.
If a file number of the opened file by a macro function other than the Open For Input As# function is specified in the <fileNo> parameter, an "Illegal function call" error will occur.
If a value outside the range of 0 to 15 is specified in the <fileNo> parameter, an "Illegal function call" error will occur.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a value outside the range -2147483648 to 2147483647 is specified as an integer parameter, an "Overflow" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

Reads the data until the end of the file.

```
Dim ALLDATA$(255)

Rem Open the file
Open "E:\input.dat" For Input As #1

For I&=0 to 255

  Rem Read line by line from the top of the file
  Line Input #1, DATA$
  ALLDATA$(I&) = DATA$

  Rem Check if the end of the file is reached
  If Eof(1) <> 0 Then
    Exit For
  Endif

Next

Rem Close up the file.
Close #1
```

Useable Modules

Unit Calculation Macro / Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 3.50 or later

Related Items

Close (Reference: ▶ Details (p.432))
Input# (Reference: ▶ Details (p.569))
Open For Input As# (Reference: ▶ Details (p.629))
Eof (Reference: ▶ Details (p.510))
Input$ (Reference: ▶ Details (p.571))

List

Outputs all or a part of program list in the system status console window.

Format

List [<lineNo1>][-<lineNo2>]

Parameter

Parameter name	Data type	Description
<lineNo1>	Integer type	Line number where the output starts
<lineNo2>	Integer type	Line number where the output ends

Return value

None.

Description

Program list from the specified line number in the <lineNo1> parameter to the specified line number in the <lineNo2> parameter is output to the system status console window.
If the <lineNo1> and <lineNo2> parameters are omitted, all lines in the program are output. If only the <lineNo1> parameter is omitted, program lines from the beginning to the specified line number by the <lineNo2> parameter are output.
If only the <lineNo2> parameter is omitted, only the program line specified in the <lineNo1> parameter is output.
If a non-existent number, numerical value, or combination of data types or values is specified for a parameter, an "Undefined line number" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- This macro function is only enabled when specified in debug mode with the Debug function. Otherwise, this macro function statement is ignored. (Reference: ▶How to Use the Debug Function (p.235))

Example

After stopping the program execution with the Stop function used in the Unit Macro processing unit, executes the List function in the system status console window to output the program line numbers from 230 to 240 to the window.

```
Macro(U1) Stop in 220
Macro(U1) List 230-240
230 POS.X#=(POS0.X@ + POS1.X@) / 2
240 POS.Y#=(POS0.Y@ + POS1.Y@) / 2
Macro(U1)>
```

Useable Modules

Unit Calculation Macro / Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 3.50 or later

Related Items

Cont (Reference: ▶ Details (p.436))
DebugPrint (Reference: ▶ Details (p.458))
SetStop (Reference: ▶ Details (p.723))
Stop (Reference: ▶ Details (p.747))
Debug (Reference: ▶ Details (p.456))
Print (Reference: ▶ Details (p.639))
SetVar (Reference: ▶ Details (p.741))
VarList (Reference: ▶ Details (p.796))

LoadBackupData

Loads the system + scene group 0 data.

Format

LoadBackupData(<fileName>)

Parameter

Parameter name	Data type	Description
<fileName>	Character string type	File name of bkd file to read in (System data + scene group 0 data (*.bkd))

Return value

None.

Description

Loads the system + scene group 0 data file specified in the <fileName> parameter.
In the <fileName> parameter, use an absolute path to specify the file name of the file to be loaded.
In the <fileName> parameter, specify a system + scene group 0 data file name. If a file other than a system + scene group 0 data file is specified, an "Illegal function call" error will occur.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a non-existent number, numerical value, or combination of data types or values is specified for a parameter, an "Illegal function call" error will occur.
If a character string longer than 255 characters is specified for a character string parameter, a "String too long" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- Execute this macro function when the BUSY signal or other measurement in progress signal is ON and measurement is prohibited. (Reference: ▶State Transitions and Execution Timing (p.227))
- After loading a file, execute "Save data". To apply the loaded settings in the sensor controller, restart the sensor controller.

Example

Loads a system + scene group 0 data file, and execute "Save data". To apply the loaded settings in the sensor controller, restart the sensor controller.

```
Rem Load the system + scene group 0 data to a file.
LoadBackupData "C:\BACKDIR\BackupData.bkd"

Rem Save to the controller.
SaveData

Rem Reboot the Sensor Controller.
SystemReset
```

Useable Modules

Scene Control Macro / Communication Command Macro

Supported Versions

Version 3.50 or later

Related Items

GetSystemData (Reference: ▶ Details (p.546))
LoadSceneGroup (Reference: ▶ Details (p.603))
LoadUnitData (Reference: ▶ Details (p.607))
SaveData (Reference: ▶ Details (p.670))
LoadScene (Reference: ▶ Details (p.601))
LoadSystemData (Reference: ▶ Details (p.605))
SaveBackupData (Reference: ▶ Details (p.669))
SystemReset (Reference: ▶ Details (p.754))

LoadScene

Loads the scene data.

Format

LoadScene <sceneNo>, <fileName>

Parameter

Parameter name	Data type	Description
<sceneNo>	Integer type	Scene number of the destination to read in (0 to 127)
<fileName>	Character string type	File name of the scene data to read in (*.scn)

Return value

None.

Description

The scene data file specified in the <fileName> parameter is loaded into the scene number specified in the <sceneNo> parameter.
In the <fileName> parameter, use an absolute path to specify the file name of the file to be loaded.
In the <fileName> parameter, specify a scene data file name. If a file other than a scene data file is specified, an "Illegal function call" error will occur.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a non-existent number, numerical value, or combination of data types or values is specified for a parameter, an "Illegal function call" error will occur.
If a character string longer than 255 characters is specified for a character string parameter, a "String too long" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- Execute this macro function when the BUSY signal or other measurement in progress signal is ON and measurement is prohibited. (Reference: ▶State Transitions and Execution Timing (p.227))

Example

Loads scene data into scene 2 and then changes the scene used to scene 2.

```
Rem Load the file of the scene data.
LoadScene 2, "C:\BACKDIR\scene02.scn"

Rem Switch the scene.
ChangeScene 2
```

Useable Modules

Scene Control Macro / Communication Command Macro

Supported Versions

Version 3.50 or later

Related Items

ChangeScene (Reference: ▶ Details (p.423))
LoadBackupData (Reference: ▶ Details (p.599))
LoadSystemData (Reference: ▶ Details (p.605))
SaveScene (Reference: ▶ Details (p.675))
GetSystemData (Reference: ▶ Details (p.546))
LoadSceneGroup (Reference: ▶ Details (p.603))
LoadUnitData (Reference: ▶ Details (p.607))
SceneNo (Reference: ▶ Details (p.691))

LoadSceneGroup

Loads the scene group data.

Format

LoadSceneGroup <sceneGroupNo>, <fileName>

Parameter

Parameter name	Data type	Description
<sceneGroupNo>	Integer type	Scene group number of the destination to read in (0 to 31)
<fileName>	Character string type	File name of the scene group data to read in (*.sgp)

Return value

None.

Description

Loads the scene group data file specified in the <fileName> parameter into the scene group number specified in the <sceneGroupNo> parameter.
In the <fileName> parameter, use an absolute path to specify the file name of the file to be loaded.
In the <fileName> parameter, specify the file name of the scene group data. If a file other than a scene group data file is specified, an "Illegal function call" error will occur.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a non-existent number, numerical value, or combination of data types or values is specified for a parameter, an "Illegal function call" error will occur.
If a character string longer than 255 characters is specified for a character string parameter, a "String too long" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- Execute this macro function when the BUSY signal or other measurement in progress signal is ON and measurement is prohibited. (Reference: ▶State Transitions and Execution Timing (p.227))

Example

Loads scene data into scene group 2 and then chenges the scene used to scene 0.

```
Rem Load the file of the scene group data.
LoadSceneGroup 2, "C:\BACKDIR\scenegroup02.sgp"

Rem Switch the scene.
ChangeScene 0
```

Useable Modules

Scene Control Macro / Communication Command Macro

Supported Versions

Version 3.50 or later

Related Items

ChangeScene (Reference: ▶ Details (p.423))
LoadBackupData (Reference: ▶ Details (p.599))
LoadSystemData (Reference: ▶ Details (p.605))
SaveSceneGroup (Reference: ▶ Details (p.677))
GetSystemData (Reference: ▶ Details (p.546))
LoadScene (Reference: ▶ Details (p.601))
LoadUnitData (Reference: ▶ Details (p.607))
SceneGroupNo (Reference: ▶ Details (p.687))

LoadSystemData

Loads the system data.

Format

LoadSystemData <fileName>

Parameter

Parameter name	Data type	Description
<fileName>	Character string type	File name of the system data to read in (*.ini)

Return value

None.

Description

Loads the system data file specified in the <fileName> parameter.
In the <fileName> parameter, use an absolute path to specify the file name of the file to be loaded.
In the <fileName> parameter, specify a system data file name. If a file other than a system data file is specified, and "Illegal function call" error will occur.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a non-existent number, numerical value, or combination of data types or values is specified for a parameter, an "Illegal function call" error will occur.
If a character string longer than 255 characters is specified for a character string parameter, a "String too long" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- Execute this macro function when the BUSY signal or other measurement in progress signal is ON and measurement is prohibited. (Reference: ▶State Transitions and Execution Timing (p.227))
- After loading a file, execute "Save data". To apply the loaded settings in the sensor controller, restart the sensor controller.

Example

After loading the system data, executes "Save data". To apply the loaded settings in the sensor controller, restart the sensor controller.

```
Rem Load the file of the system data
LoadSystemData "C:\BACKDIR\backupsysset.ini"

Rem Save to the controller.
SaveData

Rem Reboot the Sensor Controller.
SystemReset
```

Useable Modules

Scene Control Macro / Communication Command Macro

Supported Versions

Version 3.50 or later

Related Items

GetSystemData (Reference: ▶ Details (p.546))
LoadScene (Reference: ▶ Details (p.601))
LoadUnitData (Reference: ▶ Details (p.607))
SaveSystemData (Reference: ▶ Details (p.679))
LoadBackupData (Reference: ▶ Details (p.599))
LoadSceneGroup (Reference: ▶ Details (p.603))
SaveData (Reference: ▶ Details (p.670))
SystemReset (Reference: ▶ Details (p.754))

LoadUnitData

Loads the processing unit data.

Format

LoadUnitData <sceneNo>, <unitNo>, <unitCount>, <mode>, <fileName>

Parameter

Parameter name	Data type	Description
<sceneNo>	Integer type	Scene number of the destination to read in (-1 to 127)
<unitNo>	Integer type	Processing unit number to begin to read in (0 to (the number of registered processing units in the current scene minus one))
<unitCount>	Integer type	Number of pieces of the processing unit to read in (-1, 1 to (the number of registered processing units in the current scene) - (the processing unit number to begin to read in))
<mode>	Integer type	Read mode 0: Overwrite the processing unit by the loaded processing unit. 1: Insert the loaded processing unit in front of the processing unit.
<fileName>	Character string type	File name of the processing unit to read in (*unt)

Return value

None.

Description

Loads the processing unit data file specified in the <fileName> parameter in the mode specified in the <mode> parameter from processing units whose processing unit numbers are specified in the <unitNo> parameter, with the number of processing units specified in the <unitCount> parameter, of the scene number specified in the <sceneNo> parameter.
If -1 is specified in the <sceneNo> parameter, the scene number of the current scene is specified in the scene number of the destination scene.
If -1 is specified in the <unitCount> parameter, all processing unit data included in the processing unit data file is loaded.
In the <fileName> parameter, use an absolute path to specify the file name of the file to be loaded.
In the <fileName> parameter, specify the file name of a processing unit data file. If a file other than a processing unit data file is specified, an "Illegal function call" error will occur, and the scene data in the load destination will be cleared.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a non-existent number, numerical value, or combination of data types or values is specified for a parameter, an "Illegal function call" error will occur.
If a character string longer than 255 characters is specified for a character string parameter, a "String too long" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- Execute this macro function when the BUSY signal or other measurement in progress signal is ON and measurement is prohibited. (Reference: ▶State Transitions and Execution Timing (p.227))

Example

Loads a processing unit data file and inserting five of the same processing units between Processing Unit number 3 and Processing Unit number 4 of scene 2.

```
Rem Load the file of the scene data.
LoadUnitData 2, 4, 5, 1, "C:\BACKDIR\unitsave.unt"
```

Useable Modules

Scene Control Macro / Communication Command Macro

Supported Versions

Version 3.50 or later

Related Items

GetSystemData (Reference: ▶ Details (p.546))
LoadScene (Reference: ▶ Details (p.601))
LoadSystemData (Reference: ▶ Details (p.605))
SceneNo (Reference: ▶ Details (p.691))
Ut (Reference: ▶ Details (p.793))
LoadBackupData (Reference: ▶ Details (p.599))
LoadSceneGroup (Reference: ▶ Details (p.603))
SaveUnitData (Reference: ▶ Details (p.681))
UnitNo (Reference: ▶ Details (p.790))

Log

Gets the natural logarithm.

Format

Log(<expression>)

Parameter

Parameter name	Data type	Description
<expression>	Integer type Double precision real number data type	Expression to get the natural logarithm

Return value

Returns the natural logarithm value as a double precision real number.

Description

Gets the value of the natural logarithm of the expression specified in the <expression> parameter.
In the <expression> parameter, specify a positive number.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a non-existent number, numerical value, or combination of data types or values is specified for a parameter, an "Illegal function call" error will occur.
If a value is assigned to the return value variable or the variable is not used in an expression, a "Syntax error" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

Gets the natural logarithm of the integer 25 assigned to the variable X&.

```
X& = 25
XLOG# = Log(X&)
```

The result is shown below.

```
XLOG# = 3.21887582487
```

Useable Modules

Unit Calculation Macro / Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 3.50 or later

Related Items

Exp (Reference: ▶ Details (p.520))
UnitData (Reference: ▶ Details (p.778))
GetUnitData (Reference: ▶ Details (p.552))

Lsqumeth

Gets the approximate straight line from the coordinates of multiple points using the least squares method.

Format

Lsqumeth <count>, <x()>, <y()>, <line()>

Parameter

Parameter name	Data type	Description
<count>	Integer type	Number of coordinates calculated for the approximate line
<x()>	Array of double precision real number data type	1D array that stores the X coordinates of the points from which the approximate straight line is to be gotten.
<y()>	Array of double precision real number data type	1D array that stores the Y coordinates of the points from which the approximate straight line is to be gotten.
<line()>	Array of double precision real number data type	Straight line component gotten by the approximate line

Return value

None.

Description

Gets the approximate straight line from the coordinates specified in the <x()> parameter and <y()> parameter. The number of coordinates is specified in the <count> parameter.
In the <line()> parameter, specify the variable that will hold the line components of the gotten approximate straight line, without adding element numbers but adding () to the variables. In the <line()> parameter, the parameters a, b, and c that satisfy the linear equation ax + by + c = 0 are stored in the array elements of the double precision real number array variable, with "a" in element 0, "b" in element 1, and "c" in element 2.
In the <count> parameter, specify an integer value of at least 2.
In the <x()> parameter and in the <y()> parameter, specify a 1D double precision real number array variable that stores a number of coordinate values greater than or equal to the number specified in the <count> parameter, without adding element numbers but adding () to the variables.
This macro function is mainly used to get the straight line of an edge from the measurement points gotten in multiple edge measurements.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a non-existent number, numerical value, or combination of data types or values is specified for a parameter, an "Illegal function call" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

Gets the straight line of the edge of a workpiece from four edge points.

```
Dim POSX#(3), POSY#(3), PARAM#(2)

Rem Calculate the edge points from the measurement result.
For I&=0 To 3
  GetUnitData I&+1, "X", POSX#(I&)
  GetUnitData I&+1, "Y", POSY#(I&)
Next

Rem Get the straight line component.
Lsqumeth 4, POSX#(), POSY#(), PARAM#()

Erase POSX#(), POSY#(), PARAM#(), DIST#()
```

Useable Modules

Unit Calculation Macro / Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 3.50 or later

Related Items

ApproximationCircle (Reference: ▶ Details (p.411))
Crspoint (Reference: ▶ Details (p.452))
Dposline (Reference: ▶ Details (p.467))
Erase (Reference: ▶ Details (p.512))
GetUnitData (Reference: ▶ Details (p.552))
UnitData (Reference: ▶ Details (p.778))

Measure

Executes measurement processing.

Format

Measure[<wait>]

Parameter

Parameter name	Data type	Description
<wait>	Integer type	Recovery timing of macro function 0: Executes subsequent program lines without waiting for measurement to end. 1: Waits for measurement to end and then executes subsequent program lines. 2: Waits for measurement to end and measurement result display to end and then executes subsequent program lines.

Return value

None.

Description

Executes measurement processing at the recovery timing specified in the <wait> parameter.
If the <wait> parameter is omitted, operation is the same as when the <wait> parameter is set to 0.
When 0 is specified for the <wait> parameter, there is a possibility that the measurement processing executed immediately after execution of this macro function will not have ended and the measurement result cannot be properly gotten. If you want to get the measurement result, specify 1 or 2 for the <wait> parameter.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
Even if a non-existent number, numerical value, or combination of data types is specified for the parameter, an "Illegal function call" error will not occur.
If a value outside the range -2147483648 to 2147483647 is specified as an integer parameter, an "Overflow" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- Execute this macro function when a measurement-in-progress signal such as the BUSY signal is OFF and measurement is allowed. (Reference: ▶State Transitions and Execution Timing (p.227))

Example

In the communication command macro, gets the measurement X coordinate and measurement Y coordinate of the search processing unit of Processing Unit number 2 after measurement is executed. The measured X and Y coordinates can be gotten with External Reference Data numbers 6 and 7 respectively.

```
Rem Execute measurement and wait until measurement ends.
Measure 1

Rem Get the measurement result.
GetUnitData 2, 6, POSX#
GetUnitData 2, 7, POSY#
```

Useable Modules

Scene Control Macro / Communication Command Macro

Supported Versions

Version 3.50 or later

Related Items

MeasureStart (Reference: ▶ Details (p.617))
MeasureStop (Reference: ▶ Details (p.619))
Remeasure (Reference: ▶ Details (p.658))

MeasureDispG

Executes display of the measurement result of the processing unit.

Format

MeasureDispG <unitNo>, <subNo>

Parameter

Parameter name	Data type	Description
<unitNo>	Integer type	Processing unit number (0 to (the number of registered processing units in the current scene minus one))
<subNo>	Integer type	Image display sub-number of the image window to be used for display (Reference: ▶List of Sub-Image Numbers (p.374))

Return value

None.

Description

Graphically displays the judgement result in the image of the image display sub-number specified in the <subNo> parameter, of the processing unit specified in the <unitNo> parameter.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a non-existent number, numerical value, or combination of data types or values is specified for the <unitNo> parameter, an "Illegal function call" error will occur.
Even if a non-existent number, numerical value, or combination of data types or values is specified for the <subNo> parameter, an error will not occur.
If a value outside the range -2147483648 to 2147483647 is specified as an integer parameter, an "Overflow" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- This macro function can only be used in the *MEASUREDISPG subroutine. If used in another subroutine, an "Illegal function call" error will occur.

Example

In the*MEASUREDISPG subroutine of the Unit Macro processing unit, executes the analysis result display process (graphic display) of sub number 0 of Processing Unit number 2.

```
*MEASUREDISPG
  MeasureDispG 2, 0
Return
```

Useable Modules

Unit macro

Supported Versions

Version 3.50 or later

Related Items

UnitNo (Reference: ▶ Details (p.790))

Ut (Reference: ▶ Details (p.793))

MeasureId$

Gets the measurement identification.

Format

MeasureId$

Parameter

None.

Return value

Returns the measurement identification as a character string.
The measurement identification is a character string in the format (YYYY-MM-DD_HH-MM-SS-XXXX) that consists of the date and time of the internal clock at the time measurement was executed. The date consists of the year (YYYY), month (MM), and day (DD), separated by hyphens (-), and the time consists of the hour (HH), minute (MM), second (SS), and 100-µsec (XXXX), separated by hyphens (-), with the two segments joined by an underline.

Description

Gets the measurement identification.
If a value is assigned to the return value variable or the variable is not used in an expression, a "Syntax error" error will occur.

Usage Cautions

- This macro function can only be used in the *MEASUREPROC subroutine. If used in any other subroutines, an error will occur and the function will not be executed.

Example

In the *MEASUREPROC subroutine of the Unit Macro processing unit, gets the measurement identification and saves the processing unit measurement image with the measurement identification as the file name.

```
*MEASUREPROC

  Rem Get the measurement identification to set the character string for the file name.
  FILENAME$ = MeasureId$ + ".bmp"

  Rem Output measurement image 0 in bitmap format.
  SaveMeasureImage 0, FILENAME$, 0

Return
```

Useable Modules

Unit Macro / Unit Calculation Macro

Supported Versions

Version 3.50 or later

Related Items

SaveMeasureImage (Reference: ▶ Details (p.673))

MeasureProc

Executes measurement processing in a processing unit.

Format

MeasureProc <unitNo>

Parameter

Parameter name	Data type	Description
<unitNo>	Integer type	Processing unit number (0 to (the number of registered processing units in the current scene minus one))

Return value

None.

Description

Executes measurement processing in the processing unit specified in the <unitNo> parameter.
If a processing unit related to image input is specified in the <unitNo> parameter, processing may not take place correctly.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a non-existent number, numerical value, or combination of data types or values is specified for a parameter, an "Illegal function call" error will occur.
If a value outside the range -2147483648 to 2147483647 is specified as an integer parameter, an "Overflow" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

Executes measurement processing in the search processing unit of Processing Unit number 2.

```
MeasureProc 2
```

Useable Modules

Unit macro

Supported Versions

Version 3.50 or later

Related Items

UnitNo (Reference: ▶ Details (p.790))
Ut (Reference: ▶ Details (p.793))

MeasureStart

Allows input of the measurement trigger.

Format

MeasureStart

Parameter

None.

Return value

None.

Description

Allows input of the measurement trigger and input of communication commands, and changes to the measurement allowed state.
After setting the measurement prohibited state with the MeasureStop function, execute this macro function to return to the measurement allowed state.
After executing the MeasureStart function, be sure to execute the MeasureStop function. If the MeasureStop function is not executed or the MeasureStart function is executed without executing the MeasureStop function, unexpected operation may occur.
There are macro functions that can and cannot be executed in the measurement allowed state and in the measurement prohibited state. For details, refer to the explanations of the macro functions and usage cautions.
If input of the measurement trigger is allowed, measurement-in-progress signals such as the BUSY signal turn OFF.

Usage Cautions

- Execute this macro function when the BUSY signal or other measurement in progress signal is ON and measurement is prohibited. (Reference: ▶State Transitions and Execution Timing (p.227))

Example

In the communication command macro, executes measurement on the Shape Search III processing unit of Processing Unit number 1 after input of the measurement trigger is allowed, and gets the measurement result. Sets the "BUSY ON" setting to ON in advance. Multi-point output is external reference data number 168, the measurement X coordinate is external reference data number 6, and the measurement Y coordinate is external reference data number 7.

```
Rem Set multi-point output to OFF on the Shape Search III processing unit.
SetUnitData 1, 168, 0

Rem Allow input of the measurement trigger.
MeasureStart

Rem Execute measurement.
Measure 1

Rem Prohibit input of the measurement trigger.
MeasureStop

Rem Get the measurement result of the Shape Search processing unit.
GetUnitData 1, 6, POSX#
GetUnitData 1, 7, POSY#
```

Useable Modules

Scene Control Macro / Communication Command Macro

Supported Versions

Version 3.50 or later

Related Items

GetUnitData (Reference: ▶ Details (p.552))
Measure (Reference: ▶ Details (p.612))
MeasureStop (Reference: ▶ Details (p.619))
SetUnitData (Reference: ▶ Details (p.731))

MeasureStop

Prohibits input of the measurement trigger.

Format

MeasureStop[<mode>]

Parameter

Parameter name	Data type	Description
<mode>	Integer type	Measurement trigger input prohibited mode (always 0)

Return value

None.

Description

In the measurement trigger input prohibited mode specified in the <mode> parameter, prohibits measurement trigger input and communication command input, and sets the measurement prohibited state. After executing this macro function to prohibit measurement trigger input, execute the MeasureStart function to return to the measurement allowed state.
After executing the MeasureStop function, be sure to execute the MeasureStart function. If the MeasureStart function is not executed or the MeasureStop function is executed without executing the MeasureStart function, unexpected operation may occur.
If the <mode> parameter is omitted, operation is the same as when 0 is specified for the <mode> parameter.
There are macro functions that can and cannot be executed in the measurement allowed state and in the measurement prohibited state. For details, refer to the explanations of the macro functions and usage cautions.
When measurement trigger input is prohibited, measurement in progress signals such as the BUSY signal turn ON.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
Even if a non-existent number, numerical value, or combination of data types is specified for the parameter, an "Illegal function call" error will not occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- Execute this macro function when a measurement-in-progress signal such as the BUSY signal is OFF and measurement is allowed. (Reference: ▶State Transitions and Execution Timing (p.227))

Example

In the communication command macro, executes measurement on the Shape Search III processing unit of Processing Unit number 1 after input of the measurement trigger is allowed, and gets the measurement result. Sets the "BUSY ON" setting to ON in advance. Multi-point output is external reference data number 168, the measurement X coordinate is external reference data number 6, and the measurement Y coordinate is external reference data number 7.

```
Rem Set multi-point output to OFF on the Shape Search III processing unit.
SetUnitData 1, 168, 0

Rem Allow input of the measurement trigger.
MeasureStart

Rem Execute measurement.
Measure 1

Rem Prohibit input of the measurement trigger.
MeasureStop

Rem Get the measurement result of the Shape Search processing unit.
GetUnitData 1, 6, POSX#
GetUnitData 1, 7, POSY#
```

Useable Modules

Scene Control Macro / Communication Command Macro

Supported Versions

Version 3.50 or later

Related Items

GetUnitData (Reference: ▶ Details (p.552))
MeasureStart (Reference: ▶ Details (p.617))
Measure (Reference: ▶ Details (p.612))
SetUnitData (Reference: ▶ Details (p.731))

Mid$

Extract a part from the character string.

Format

Mid$(<string>, <start>, <length>)

Parameter

Parameter name	Data type	Description
<string>	Character string type	Extraction target character string
<start>	Integer type	Starting position of extraction (1 to the length of the target character string)
<length>	Integer type	Length of characters to be extracted (1 to the remaining length from specified starting position)

Return value

Returns the character string type value of the extracted character string.

Description

Extracts the specified length of the character string in the <length> parameter after the positon specified in the <start> parameter from the <string> parameter.
Specify the length of characters to be extracted in bytes for the <length> parameter. Each single-byte character (i.e., half-width alphanumeric character and symbol) consumes one byte, whereas each double-byte character consumes two bytes.
If the length specified in the <length> parameter is longer than the length of the character string between the <start> parameter position and the end of the character string, the all characters from the <start> parameter position to the end of the character string are extracted.
If a larger value than the length of the target character string is specified in the <start> parameter, an "Illegal function call" error will occur.
If 0 or smaller value is specified in the <start> parameter or <length> parameter, an "Illegal function call" error will occur.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a value outside the range -2147483648 to 2147483647 is specified as an integer parameter, an "Overflow" error will occur.
If a character string longer than 255 characters is specified for a character string parameter, the 255-character string before the 256th character is used for the macro function processing. Characters after the 256th character will be discarded.
If a value is assigned to the return value variable or the variable is not used in an expression, a "Syntax error" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

Extracts four characters from the second character and eight characters from the third character in the half-width alphanumeric character string. For the latter operation, because the specified length exceeds the target character string length of eight in this example, only five characters from the third to the end of the character string.

```
INPUTSTR$ = "ABCDEFG"

Rem Extract four characters from the second character of the character string.
OUTPUTSTR1$ = Mid$(INPUTSTR$, 2, 4)
Rem Extract eight characters from the third character of the character string
OUTPUTSTR2$ = Mid$(INPUTSTR$, 3, 8)
```

The result is shown below.

```
OUTPUTSTR1$ = "BCDE"
OUTPUTSTR2$ = "CDEFG"
```

Useable Modules

Unit Calculation Macro / Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 3.50 or later

Related Items

Asc (Reference: ▶ Details (p.413))
Chr$ (Reference: ▶ Details (p.427))
Hex$ (Reference: ▶ Details (p.560))
LCase$ (Reference: ▶ Details (p.589))
Left$ (Reference: ▶ Details (p.591))
Len (Reference: ▶ Details (p.593))
Piece$ (Reference: ▶ Details (p.637))
Right$ (Reference: ▶ Details (p.664))
Str$ (Reference: ▶ Details (p.749))
Str2$ (Reference: ▶ Details (p.751))
UCase$ (Reference: ▶ Details (p.775))
Val (Reference: ▶ Details (p.794))

Mkdir

Build a directory

Format

Mkdir <directoryName>

Parameter

Parameter name	Data type	Description
<directoryName>	Character string type	Directory name of built directory

Return value

None.

Description

Build the directory specified in the <directoryName> parameter.
In the <directoryName> parameter, use an absolute path to specify the directory name of the directory to be built.
In the following case, a "Illegal function call" error occurs without building a directory.

- If the specified directory already exists
- If the external memory is specified for where to build a directory in with no external memory inserted
- Free space in the disk drive specified for the directory building is insufficient for building a new directory

If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a character string longer than 255 characters is specified for a character string parameter, a "String too long" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

Builds a directory named "IMAGE2" under the root folder of the E drive.

```
Mkdir "E:\IMAGE2"
```

Useable Modules

Unit Calculation Macro / Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 3.50 or later

Related Items

Dskf (Reference: ▶ Details (p.506))
GetSystemData (Reference: ▶ Details (p.546))
Kill (Reference: ▶ Details (p.587))
Fcopy (Reference: ▶ Details (p.522))
IsFile (Reference: ▶ Details (p.577))
Rmdir (Reference: ▶ Details (p.666))

MOD

Gets the remainder.

Format

<expression1> MOD <expression2>

Parameter

Parameter name	Data type	Description
<expression1>	Integer type	Expression of the dividend to calculate the remainder
<expression2>	Integer type	Expression of the divisor to calculate the remainder

Return value

Returns the remainder as an integer value.

Description

Gets the remainder from division of the expression specified in the <expression1> parameter by the expression specified in the <expression2> parameter.
If double precision real number expressions are specified, the values are treated as values with digits to the right of the decimal point rounded off.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a value outside the range -2147483648 to 2147483647 is specified as an integer parameter, an "Overflow" error will occur.
If 0 is specified in the <expression2> parameter, a "Division by zero" error will occur.
If a value is assigned to the return value variable or the variable is not used in an expression, a "Syntax error" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

Increments a counter from 0 to 100.

```
I& = (I&+1) MOD 100
```

Useable Modules

Unit Calculation Macro / Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 3.50 or later

Related Items

Fix (Reference: ▶ Details (p.524))
GetUnitData (Reference: ▶ Details (p.552))
UnitData (Reference: ▶ Details (p.778))

MoveUnit

Moves a processing unit.

Format

MoveUnit <srcUnitNo>, <destUnitNo>

Parameter

Parameter name	Data type	Description
<srcUnitNo>	Integer type	Processing unit number (0 to (number of processing units of current scene minus one)) of source
<destUnitNo>	Integer type	Processing unit number (0 to (number of processing units of current scene minus one)) indicating the position of the destination in the measurement flow

Return value

None.

Description

Moves the processing unit specified in the <srcUnitNo> parameter to the position in the measurement flow specified in the <destUnitNo> parameter.
The processing unit numbers of processing units after the processing unit number specified in the <srcUnitNo> parameter are moved up by 1. The processing unit numbers of processing units after the processing unit number specified in the <destUnitNo> parameter are moved down by 1.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a non-existent number, numerical value, or combination of data types or values is specified for a parameter, an "Illegal function call" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- Execute this macro function when the BUSY signal or other measurement in progress signal is ON and measurement is prohibited. (Reference: ▶State Transitions and Execution Timing (p.227))

Example

Moves the processing unit of Processing Unit number 2 between Processing Unit number 5 and Processing Unit number 6.

```
MoveUnit 2, 5
```

Useable Modules

Scene Control Macro / Communication Command Macro

Supported Versions

Version 3.50 or later

Related Items

AssignUnit (Reference: ▶ Details (p.415))
CopyUnit (Reference: ▶ Details (p.442))
InsertUnit (Reference: ▶ Details (p.573))
MeasureStop (Reference: ▶ Details (p.619))
CheckUnit (Reference: ▶ Details (p.425))
DeleteUnit (Reference: ▶ Details (p.460))
MeasureStart (Reference: ▶ Details (p.617))
UnitCount (Reference: ▶ Details (p.777))

NOT

Gets the "not" result (negation) of the expression.

Format

NOT(<expression>)

Parameter

Parameter name	Data type	Description
<expression>	Integer type	Expression to calculate the negation

Return value

Returns an integer "not" value.

Description

Gets the "not" result of the 32-digit binary value specified in the <expression> parameter by inverting each bit.
If a double precision real number expression is specified in the <expression> parameter, the value is treated as a value with digits to the right of the decimal point rounded off.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a value outside the range -2147483648 to 2147483647 is specified as an integer parameter, an "Overflow" error will occur.
If a value is assigned to the return value variable or the variable is not used in an expression, a "Syntax error" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

Gets the "not" value of the integer 0 assigned to the variable X&.

```
X& = 0
XX& = NOT(X&)
```

The result is shown below.

```
XX& = -1
```

Useable Modules

Unit Calculation Macro / Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 3.50 or later

Related Items

AND (Reference: ▶ Details (p.409))
UnitData (Reference: ▶ Details (p.778))
XOR (Reference: ▶ Details (p.809))
GetUnitData (Reference: ▶ Details (p.552))
OR (Reference: ▶ Details (p.635))

Open For Append As#

Open the file in append mode.

Format

Open <fileName> For Append As #<fileNo>

Parameter

Parameter name	Data type	Description
<fileName>	Character string type	Absolute path of the file to be opened
<fileNo>	Integer type	Assigned file number (0 to 15) to the opened file

Return value

None.

Description

Assigns the specified file number in the <fileNo> parameter to the specified file by the <fileName> parameter and open the file in append mode.
Be sure to execute the Close function to close the file opended with this macro function within the same subroutine. File accessing processes such as data writing to a file and data reading from a file may not be completed properly in the following cases.

- The Close function is not executed. The Close function is used in a different subroutine from where this macro function is executed.
- This macro function is executed at a different timing from the Open function execution.

In the <fileName> parameter, use an absolute path to specify the file name of the file to be opened.
If the file of the file name specified in the <fileName> parameter does not exist, the file is newly created.
If the file of the file name specified in the <fileName> parameter already exists, open the existing file so that the additional writing in the file is possible.
If a value outside the range of 0 to 15 is specified in the <fileNo> parameter, an "Illegal function call" error will occur.
If a value outside the range -2147483648 to 2147483647 is specified as an integer parameter, an "Overflow" error will occur.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a character string longer than 255 characters is specified for a character string parameter, a "String too long" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

Opens the file named "input.txt" under the E drive and writes the data to the file.

```
STRING$ = "Sample"

Rem Open the file
Open "E:\input.txt" For Append As #1

Print #1, STRING$

Rem Close up the file.
Close #1
```

Useable Modules

Unit Calculation Macro / Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 3.50 or later

Related Items

Close (Reference: ▶ Details (p.432))
GetSystemData (Reference: ▶ Details (p.546))
Input$ (Reference: ▶ Details (p.571))
Line Input# (Reference: ▶ Details (p.595))
Open For Output As# (Reference: ▶ Details (p.631))
Eof (Reference: ▶ Details (p.510))
Input# (Reference: ▶ Details (p.569))
IsFile (Reference: ▶ Details (p.577))
Open For Input As# (Reference: ▶ Details (p.629))
Print# (Reference: ▶ Details (p.641))

Open For Input As#

Open the file in reading mode.

Format

Open <fileName> For Input As #<fileNo>

Parameter

Parameter name	Data type	Description
<fileName>	Character string type	Absolute path of the file to be opened
<fileNo>	Integer type	Assigned file number (0 to 15) to the opened file

Return value

None.

Description

Assigns the specified file number in the <fileNo> parameter to the specified file by the <fileName> parameter and open the file in reading mode.
Be sure to execute the Close function to close the file opended with this macro function within the same subroutine. File accessing processes such as data writing to a file and data reading from a file may not be completed properly in the following cases.

- The Close function is not executed.
- The Close function is used in a different subroutine from where this macro function is executed.
- This macro function is executed at a different timing from the Open function execution.

In the <fileName> parameter, use an absolute path to specify the file name of the file to be opened.
If the file of the file name specified in the <fileName> parameter does not exist, an "Illegal function call" error will occur.
If a value outside the range of 0 to 15 is specified in the <fileNo> parameter, an "Illegal function call" error will occur.
If a value outside the range -2147483648 to 2147483647 is specified as an integer parameter, an "Overflow" error will occur.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a character string longer than 255 characters is specified for a character string parameter, a "String too long" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

Reads the data until the end of the file.

```
Dim ALLDATA$(255)

Rem Open the file
Open "E:\input.dat" For Input As #1

For I&=0 to 255

  Rem Read line by line from the top of the file
  Input #1, DATA$
  ALLDATA$(I&) = DATA$

  Rem Check if the end of the file is reached
  If Eof(1) <> 0 Then
    Exit For
  Endif

Next

Rem Close up the file.
Close #1
```

Useable Modules

Unit Calculation Macro / Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 3.50 or later

Related Items

Close (Reference: ▶ Details (p.432))
Eof (Reference: ▶ Details (p.510))
GetSystemData (Reference: ▶ Details (p.546))
Input# (Reference: ▶ Details (p.569))
Input$ (Reference: ▶ Details (p.571))
IsFile (Reference: ▶ Details (p.577))
Line Input# (Reference: ▶ Details (p.595))
Open For Append As# (Reference: ▶ Details (p.627))
Open For Output As# (Reference: ▶ Details (p.631))
Print# (Reference: ▶ Details (p.641))

Open For Output As#

Opens the file in writing mode.

Format

Open <fileName> For Output As #<fileNo>

Parameter

Parameter name	Data type	Description
<fileName>	Character string type	Absolute path of the file to be opened
<fileNo>	Integer type	Assigned file number (0 to 15) to the opened file

Return value

None.

Description

Assigns the specified file number in the <fileNo> parameter to the specified file by the <fileName> parameter and open the file in writing mode.
Be sure to execute the Close function to close the file opended with this macro function within the same subroutine. File accessing processes such as data writing to a file and data reading from a file may not be completed properly in the following cases.

- The Close function is not executed.
- The Close function is used in a different subroutine from where this macro function is executed.
- This macro function is executed at a different timing from the Open function execution.

In the <fileName> parameter, use an absolute path to specify the file name of the file to be opened.
If the file of the file name specified in the <fileName> parameter does not exist, an "Illegal function call" error will occur.
If the file of the file name specified in the <fileName> parameter already exists, the existing file content is once deleted, then opens the existing file so that the writing in the file is possible.
If a value outside the range of 0 to 15 is specified in the <fileNo> parameter, an "Illegal function call" error will occur.
If a value outside the range -2147483648 to 2147483647 is specified as an integer parameter, an "Overflow" error will occur.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a character string longer than 255 characters is specified for a character string parameter, a "String too long" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

Opens the file, writes the data in the file, and then closes the file.

```
DATA& = 10

Rem Open the file
Open "E:\input.dat" For Output As #1

Rem Write the data in the opened file
Print #1 DATA&

Rem Close the opened file
Close #1
```

Useable Modules

Unit Calculation Macro / Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 3.50 or later

Related Items

Close (Reference: ▶ Details (p.432))
GetSystemData (Reference: ▶ Details (p.546))
Input$ (Reference: ▶ Details (p.571))
Line Input# (Reference: ▶ Details (p.595))
Open For Input As# (Reference: ▶ Details (p.629))
Eof (Reference: ▶ Details (p.510))
Input# (Reference: ▶ Details (p.569))
IsFile (Reference: ▶ Details (p.577))
Open For Append As# (Reference: ▶ Details (p.627))
Print# (Reference: ▶ Details (p.641))

OpenTextData

Opens a messages file.

Format

OpenTextData <ident> As #<textDataNo>

Parameter

Parameter name	Data type	Description
<ident>	Character string type	Identification name for the message file that is opened as text data
<textDataNo>	Integer type	Text data number (0 to 15) that is assigned to the message file opened as text data.

Return value

None.

Description

Assigns the specified text data number in the <textDataNo> parameter to the specified file by the <ident> parameter and opens the file.

Be sure to execute the CloseTextData function to close the file opended with this macro function within the same subroutine. The message file cannot properly be closed and this macro function may not properly be executed in the subsequent processes in the following cases.

- The CloseTextData function is not executed.
- This CloseTextData function is used in a different subroutine from where this function is executed.
- This CloseTextData function is executed at a different timing from this function execution.

If a value outside the range from 0 to 15 is specified in the <textDataNo> parameter, an "Illegal function call" error will occur.

If the text data number that is already opened is specified in the <textDataNo> parameter, an "Illegal function call" error will occur.

If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.

If a non-existent number, numerical value, or combination of data types or values is specified for a parameter, an "Illegal function call" error will occur.

If a value outside the range -2147483648 to 2147483647 is specified as an integer parameter, an "Overflow" error will occur.

If a character string longer than 255 characters is specified for a character string parameter, a "String too long" error will occur.

If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

Uses the *MEASUREDISPT subroutine of the Unit Macro processing unit to display the measured correlation value by the search processing unit (Processing Unit number 5), along with the gotten text string from the prepared message file for the processing unit, in the text window. The correlation value can be gotten with External Reference Data number 5.

```
*MEASUREDISPT

  Rem Get the measurement result.
  GetUnitData 5, 5, CR#

  Rem Open the messages file
  OpenTextData "Search" As #1

  Rem Get the text
  TEXT$ = GetText$(#1, "Correlation")

  Rem Draw the gotten text string from the messages file without adding any line break on the text window.
  DrawText TEXT$, UnitJudge(5), 0

  Rem Draw the measurement results on the text window.
  DrawText Str2$(CR#, 4, 4, 0, 0), UnitJudge(5), 1

  Rem Close up the messages file.
  CloseTextData

Return
```

The result is shown below.

```
Correlation value:  90.0000
```

Useable Modules

Unit Calculation Macro / Unit Macro

Supported Versions

Version 5.00 or later

Related Items

CloseTextData (Reference: ▶ Details (p.434))
DrawText (Reference: ▶ Details (p.500))
GetText$ (Reference: ▶ Details (p.548))
GetUnitData (Reference: ▶ Details (p.552))
UnitJudge (Reference: ▶ Details (p.788))

OR

Gets the logical sum of two expressions.

Format

<expression1> OR <expression2>

Parameter

Parameter name	Data type	Description
<expression1>	Integer type	Expression to calculate the logical sum
<expression2>	Integer type	Expression to calculate the logical sum

Return value

Returns the logical sum as an integer value.

Description

Gets the logical sum by bit of the expression specified in the <expression1> parameter and the expression specified in the <expression2> parameter.
When the values of the <expression1> parameter and <expression2> parameter are double precision real values, the decimal part is rounded off.
This can also be used as an Or condition in an If statement. For details on the logical expression, refer to "Operators".
Reference: ▶Operator (p.213)
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a value outside the range -2147483648 to 2147483647 is specified as an integer parameter, an "Overflow" error will occur.
If a value is assigned to the return value variable or the variable is not used in an expression, a "Syntax error" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

Gets the logical sum of 1 assigned to the variable EXP1& and 4 assigned to the variable EXP2&.

```
EXP1& = 1
EXP2& = 4

EXPALL& = EXP1& OR EXP2&
```

The result is shown below.

```
EXPALL& = 5
```

Useable Modules

Unit Calculation Macro / Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 3.50 or later

Related Items

AND (Reference: ▶ Details (p.409))
UnitData (Reference: ▶ Details (p.778))
XOR (Reference: ▶ Details (p.809))
GetUnitData (Reference: ▶ Details (p.552))
NOT (Reference: ▶ Details (p.626))

Piece$

Extract the part of the character string which was separated by delimiter from the string.

Format

Piece$(<string>, <delimiter>, <start>, <end>)

Parameter

Parameter name	Data type	Description
<string>	Character string type	Extraction target character string
<delimiter>	Character string type	Character string delimiter
<start>	Integer type	Index number of the character string that the extraction is started (Number 1 to number of substrings)
<end>	Integer type	Index number of the character string that the extraction is finished (Number 1 to number of substrings)

Return value

Returns the character string type value of the extracted character string.

Description

Extracts the character string portions from the starting index number specified in the <start> parameter to the ending index number specified in the <end> parameter after assigning index numbers to the portions separated with the separator string specified in the <delimiter> parameter.
If the character string in the <string> parameter cannot be separated with the character string in the <delimiter> parameter, all characters in the character string is extracted as a portion.
Specify the starting index number to be extracted in the <start> parameter. The index numbers are assigned to the portions in ascending order starting with 1 to the first portion.
If a larger value than the number of separated portions is specified in the <start> parameter, an "Illegal function call" error will occur.
If the index number specified in the <end> parameter is larger than the index number specified in the <start> parameter, an "Illegal function call" error will occur.
If 0 or smaller value is specified in the <start> parameter or <end> parameter, an "Illegal function call" error will occur.
If a larger value than the number of separated portions is specified in the <end> parameter, all portions from the starting index number in the <start> parameter to the end of the character string are extracted.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a value outside the range -2147483648 to 2147483647 is specified as an integer parameter, an "Overflow" error will occur.
If a character string longer than 255 characters is specified in the <string> parameter, the 255-character string before the 256th character is used for the macro function processing. Characters after the 256th character will be discarded.
If a value is assigned to the return value variable or the variable is not used in an expression, a "Syntax error" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

Gets the part of the character string which was separated by a semicolon (;) delimiter from the string.

```
INPUTSTR$ = "PIECE1;PIECE2;PIECE3;PIECE4"
DELIMITER$ = ";"

Rem Extract the first substring of the character string
OUTPUTSTR1$ = Piece$(INPUTSTR$, DELIMITER$, 1, 1)
Rem Extract the third and forth substrings from the character string
OUTPUTSTR2$ = Piece$(INPUTSTR$, DELIMITER$, 3, 4)
```

The result is shown below.

```
OUTPUTSTR1$ = "PIECE1"
OUTPUTSTR2$ = "PIECE3;PIECE4"
```

Useable Modules

Unit Calculation Macro / Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 3.50 or later

Related Items

Asc (Reference: ▶ Details (p.413))
Chr$ (Reference: ▶ Details (p.427))
Hex$ (Reference: ▶ Details (p.560))
LCase$ (Reference: ▶ Details (p.589))
Left$ (Reference: ▶ Details (p.591))
Len (Reference: ▶ Details (p.593))
Mid$ (Reference: ▶ Details (p.621))
Right$ (Reference: ▶ Details (p.664))
Str$ (Reference: ▶ Details (p.749))
Str2$ (Reference: ▶ Details (p.751))
UCase$ (Reference: ▶ Details (p.775))
Val (Reference: ▶ Details (p.794))

Print

Outputs data in the system status console window.

Format

Print <expression>[;|, <expression>...][;|,]

Parameter

Parameter name	Data type	Description
<expression>	Integer type Double precision real number data type Character string type Array	Numerical expression or character string to be output

Return value

None.

Description

Outputs the numerical expression or character string specified in the <expression> parameter to the system status console window. (Reference: ▶Description of the System Status Console Window (p.185))
If the parameters are separated with commas (,), the specified expressions and character strings in the <expression> parameters are separated by tab delimiters and output.
If the parameters are separated with semicolons (;), the specified expressions and character strings in the <expression> parameters are output subsequently to the output expression or text string immediately before.
If any of semicolon (;) and comma (,) are specified at the end of parameters, a line break is added after all the parameters are output to the system status console window.
If a non-existent number, numerical value, or combination of data types or values is specified for a parameter, an "Illegal function call" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- After the data output to the system status console window, the window is displayed on top of the Sensor Controller main screen. To display the system status console window on top of the main screen, click [_] on the upper-right of the system status console window or press [Alt] + [Tab] on the connected USB keyboard to the sensor controller.

Example

Gets the gotten values of "correlation value", "measurement coordinate X", and "measurement coordinate Y" by the search processing unit (Processing Unit number 2) and outputs to the system status console window. The correlation value, measured position coordinates X and Y can be gotten with External Reference Data numbers 5, 6, and 7 respectively.

```
Rem Get the measurement result.
GetUnitData 2, 5, CR#
GetUnitData 2, 6, X#
GetUnitData 2, 7, Y#

Rem Output the gotten measurement results to the system status console window.
Print CR#; ","; X#; ","; Y#
```

Useable Modules

Unit Calculation Macro / Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 3.50 or later

Related Items

Cont (Reference: ▶ Details (p.436))
DebugPrint (Reference: ▶ Details (p.458))
List (Reference: ▶ Details (p.597))
SetVar (Reference: ▶ Details (p.741))
VarList (Reference: ▶ Details (p.796))
Debug (Reference: ▶ Details (p.456))
GetUnitData (Reference: ▶ Details (p.552))
SetStop (Reference: ▶ Details (p.723))
Stop (Reference: ▶ Details (p.747))

Print#

Outputs data in a file.

Format

Print #<fileNo>[, <expression>[;|, <expression>...][;|,]

Parameter

Parameter name	Data type	Description
<fileNo>	Integer type	File number (0 to 15) of the output destination file
<expression>	Integer type Double precision real number data type Character string type Array	Numerical expression or character string to be output

Return value

None.

Description

Outputs an expression or a character string specified in the <expression> parameter in the file with the file number specified in the <fileNo> parameter.
If the parameters are separated with commas (,), the specified expressions and character strings in the <expression> parameters are separated by tab delimiters and output.
If the parameters are separated with semicolons (;), the specified expressions and character strings in the <expression> parameters are output subsequently to the output expression or text string immediately before.
If any of semicolon (;) and comma (,) are specified at the end of parameters, a line break is added after all the parameters are output.
If an unopened file number is specified in the <fileNo> parameter, an "Illegal function call" error will occur.
If a file number of the opened file by a macro function other than the Open For Append As# function and the Open For Output As# function is specified in the <fileNo> parameter, an "Illegal function call" error will occur.
If a value outside the range of 0 to 15 is specified in the <fileNo> parameter, an "Illegal function call" error will occur.
If a character string longer than 127 characters is specified in the <expression> parameter, an "Illegal function call" error will occur.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

Outputs a character string to the file "E:\input.txt".

```
STRING$ = "Sample"

Rem Open the file
Open "E:\input.txt" For Append As #1

Print #1, STRING$

Rem Close up the file.
Close #1
```

Useable Modules

Unit Calculation Macro / Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 3.50 or later

Related Items

Close (Reference: ▶ Details (p.432))
Open For Append As# (Reference: ▶ Details (p.627))
Open For Output As# (Reference: ▶ Details (p.631))

PutAll

Sets the output state of all output terminals.

Format

PutAll <ioIdent>, <state>

Parameter

Parameter name	Data type	Description
<ioIdent>	Character string type	Identification name of the communication module to be used (always "ParallelIo") (Reference: ▶List of I/O Modules (p.341))
<state>	Integer type	Output state of terminal

Return value

None.

Description

Sets the output state of all output terminals of the communication module specified in the <ioIdent> parameter to the state specified in the <state> parameter.
Normally "ParallelIo" should be specified in the <ioIdent> parameter.
The output state of each output terminal is expressed as an integer value (OFF (0) or ON (1)) in each digit of a character string in binary notation.
In parallel I/O, integer values are returned expressing DO0 to DO15 in the 1st digit to the 16th digit.
Example: When DO0 to DO5 are ON and DO6 to DO15 are OFF

- Binary notation: 0000 0000 0011 1111
- Value of output states that are set: 63

If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a non-existent number, numerical value, or combination of data types or values is specified for a parameter, an "Illegal function call" error will occur.
If a value outside the range -2147483648 to 2147483647 is specified as an integer parameter, an "Overflow" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- When the operation mode is multi-line random trigger and there are three or more lines, the output states of the DO signal cannot be set. Even if this macro function is used when there are three or more lines, an "Illegal function call" error will not occur.

Example

In the communication command macro, sets all DO output states of parallel I/O to ON.

```
IOMODULE$ = "ParallelIo"

Rem Set the output state.
PutAll IOMODULE$, 65535
```

Useable Modules

Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 3.50 or later

Related Items

BusyOut (Reference: ▶ Details (p.419))
GetPort (Reference: ▶ Details (p.542))
PutPort (Reference: ▶ Details (p.645))
GetAll (Reference: ▶ Details (p.528))
JudgeOut (Reference: ▶ Details (p.585))
RunOut (Reference: ▶ Details (p.667))

PutPort

Sets the output state of the specified output terminal.

Format

PutPort <ioIdent>, <portNo>, <state>

Parameter

Parameter name	Data type	Description
<ioIdent>	Character string type	Identification name of the communication module to be used (always "ParallelIo") (Reference: ▶List of I/O Modules (p.341))
<portNo>	Integer type	Terminal number of output terminal whose output state is to be set. Parallel I/O • FH DO0 to DO15: 0 to 15 GATE N: 100 + N x 8 (N: Line number (0 to 7)) BUSY N: 101+N x 8 OR N: 102+N x 8 ERROR N: 103+N x 8 RUN N: 104+N x 8 READY N: 105 + N x 8 ACK: 200 • FZ5 DO0 to DO15: 0 to 15 GATE0: 100 BUSY: 101 OR0: 102 ERROR: 103 RUN: 104 READY0: 105 GATE1: 108 OR1: 110 READY1: 113
<state>	Integer type	Output state of output terminal 0: Output OFF 1: Output ON

Return value

None.

Description

Sets the state of the output terminal of the terminal number specified in the <portNo> parameter of the communication module specified in the <ioIdent> parameter to the output state specified in the <state> parameter.
Normally "ParallelIo" should be specified in the <ioIdent> parameter.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a non-existent number, numerical value, or combination of data types or values is specified in any of the <ioIdent> and <portNo> parameters, an "Illegal function call" error will occur.
Even if an output status parameter value that does not exist (i.e., other than 0 and 1) is specified in the <state> parameter, an error will not occur.
If a value outside the range -2147483648 to 2147483647 is specified as an integer parameter, an "Overflow" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

In the communication command macro, sets the output state of DO3 of parallel I/O to ON.

```
IOMODULE$ = "ParallelIo"

Rem Set the output state.
PutPort IOMODULE$, 3, 1
```

Useable Modules

Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 3.50 or later

Related Items

BusyOut (Reference: ▶ Details (p.419))
GetPort (Reference: ▶ Details (p.542))
PutAll (Reference: ▶ Details (p.643))
GetAll (Reference: ▶ Details (p.528))
JudgeOut (Reference: ▶ Details (p.585))
RunOut (Reference: ▶ Details (p.667))

RaiseOptionEvent

Notifies option events to the UI screen.

Format

RaiseOptionEvent <eventNo>, <parameter>

Parameter

Parameter name	Data type	Description
<eventNo>	Integer type	Event number to be notified 0xFF: Layout switch in the main screen of both sensor controller and remote operation tool 0x1FF: Layout switch in the main screen of the sensor controller 0x2FF: Layout switch in the main screen of the remote operation tool
<parameter>	Integer type	Event parameter to be notified If the event is layout switch 0 to 8: New layout number

Return value

None.

Description

Notifies the UI screen of the event specified in the <eventNo> parameter and the parameter specified in the <parameter> parameter.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
Even if a non-existent number, numerical value, or combination of data types or values is specified for the parameter, an error will not occur.
If a value outside the range -2147483648 to 2147483647 is specified as an integer parameter, an "Overflow" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.
Through the use of the FH-AP1 Application Producer (sold separately), it is possible to create a new event in the UI screen.

Usage Cautions

None.

Example

Switches the main screen layout of the sensor controller with the communications command macro.

```
Rem switch to layout 1
RaiseOptionEvent 255, 1
```

Useable Modules

Scene Control Macro / Communication Command Macro

Supported Versions

Version 3.50 or later

Related Items

None.

ReadPlcMemory

Reads a value from the PLC memory area.

Format

ReadPlcMemory <ioIdent>, <area>, <channelOffset>, <channelCount>, <readData()>

Parameter

Parameter name	Data type	Description
<ioIdent>	Character string type	Identification name of the communication module to be used (Reference: ▶List of I/O Modules (p.341))
<area>	Integer type	Area classification number of data output area to read in the target
<channelOffset>	Integer type	Offset from beginning of data output area to address where reading is to start.
<channelCount>	Integer type	Data size to read in (channel unit)
<readData()>	Integer array	Loaded data

Return value

None.

Description

Using the communication module specified in the <ioIdent> parameter, the data size specified in the <channelCount> parameter is read from the address that is offset by the value specified in the <channelOffset> parameter, of the PLC area type specified in the <area> parameter.
After reading the value from the PLC memory area using this macro function, execute the GetPlcData function to get the read value.
In the <readData()> parameter, specify the 1D integer array variable that stores the data that was read. Add () without specifying element numbers.
This macro function cannot be used to read data from a PLC that is connected by other than the PLC link communication module.
In the <area> parameter, specify the Identification of the register that is set with the PLC link setting in the system settings.
In the <channelCount> parameter, specify the size in channel units. The size of one integer type data item is two channels (four bytes), and thus to read one integer value, a one-element array should be prepared with the <readData()> parameter, and 2 should be specified in the <channelCount> parameter.
If a size larger than the array size specified in the <readData()> parameter is specified in the <channelCount> parameter, a "Subscript out of range" error will occur.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a non-existent number, numerical value, or combination of data types or values is specified for a parameter, an "Illegal function call" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- After using this macro function to read data, always use the GetPlcData function to get the value from the data that was read. If the value is gotten directly from the <readData()> parameter without using the GetPlcData function, the correct value may not be gotten.

Example

In the communication command macro, reads multiple data from the PLC connected by PLC link.

```
IOMODULE$ = "UdpPlcLink"

Rem Get the settings of the output data area.
GetSystemData IOMODULE$, "outputArea", AREA&
GetSystemData IOMODULE$, "outputMemoryAddress", ADDRESS&

Rem Create the integer array variable to store the read data.
Dim DATA&(1)

Rem Load the data (4ch) from data output area.
ReadPlcMemory IOMODULE$, AREA&, ADDRESS&, 4, DATA&()

Get the values from the read data.
GetPlcData IOMODULE$, DATA&(), 0, 4, VALUE0&
GetPlcData IOMODULE$, DATA&(), 4, 4, VALUE1&
```

Useable Modules

Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 4.20 or later

Related Items

GetPlcData (Reference: ▶ Details (p.538))
SetPlcData (Reference: ▶ Details (p.709))
WritePlcMemory (Reference: ▶ Details (p.807))

ReceiveData

Receives data.

Format

ReceiveData <ioIdent>, <inputData()>, <inputMaxSize>, <inputSize>[, <parameter()>, <parameterSize>]

Parameter

Parameter name	Data type	Description
<ioIdent>	Character string type	Identification name of the communication module to be used (Reference: ▶List of I/O Modules (p.341))
<inutData()>	Integer array	Received data
<inputMaxSize>	Integer type	Maximum size of data to be received
<inputSize>	Integer type	Received data size
<parameter()>	Integer array	The parameter data specified in the option
<parameterSize>	Integer type	Size of parameter data specified optionally.

Return value

None.

Description

Receive data up to the size specified in the <inputMaxSize> parameter by using the communication module specified in the <ioIdent> parameter.
In the <inputData()> parameter, specify the 1D integer array variable that will hold the data to be received, without adding element numbers but adding () to the variables. In the <inputSize> parameter, specify the integer variable that will hold the size of the received data.
Values that can be set in the <parameter()> parameter and <parameterSize> parameter depend on the communication module specified in the <ioIdent> parameter. The <parameter()> parameter and <parameterSize> parameter can be omitted.
With this macro function, data that has arrived at the sensor controller from an external device is received when the macro function is executed. If no data has arrived, 0 is stored in the <inputSize> parameter. If the desired data is not received the first time the macro function is executed, execute repeatedly until all data has been received.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a non-existent number, numerical value, or combination of data types or values is specified for a parameter, an "Illegal function call" error will occur.
If Handshake is on and data reception fails due to a communication timeout or other reason, an "Illegal function call" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- Execute this macro function when the target communication module is stopped. (Reference: ▶Exclusive Control in a Process (p.234))

Example

Receives data from an external device in the measurement process of the unit macro.

```
Rem Prepare a buffer that can receive 12 bytes of data.
Dim BUFFER&(11)
IOMODULE$ = "TcpNormal"

Rem Set the polling state of the communication module to stopped in order to receive the data.
SetPollingState IOMODULE$, False

Rem Executing the initialization of the reception data size.
SIZE& = 0

Repeat the reception process until the data has been received.
Try

  Do

    Rem Attempting the data reception.
    ReceiveData IOMODULE$, BUFFER&(), 12, SIZE&

    Rem Once the data has been received, display the data size in the system status console window.
    If(SIZE& > 0) Then
      Print "Received data size = " + Str$(SIZE&)
    Endif

  Loop While SIZE& = 0

Rem Data has been received, so set the polling state of the communication module to running.
  SetPollingState IOMODULE$, True

Catch

Rem Return the polling state of the stopped communication module to running.
If GetPollingState(IOMODULE$) = False Then
  SetPollingState IOMODULE$, True
Endif

End Try
```

Useable Modules

Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 3.50 or later

Related Items

GetPollingState (Reference: ▶ Details (p.540))
SendData (Reference: ▶ Details (p.696))
SendString (Reference: ▶ Details (p.698))
SetPollingState (Reference: ▶ Details (p.711))

RefreshImageWindow

Updates the image window.

Format

RefreshImageWindow

Parameter

None.

Return value

None.

Description

Updates the image window.
Execute this macro function to update the display after executing the ImageUpdate function when a graphic is redrawn in the image window or the image is redrawn.

Usage Cautions

- None.

Example

In the communication command macro, updates an image displayed in "Camera Image Freeze" image mode to the most recent image.

```
Rem Update the freeze image to the most recent image.
ImageUpdate

Rem Apply the updated image to the display in the image window.
RefreshImageWindow
```

Useable Modules

Communication Command Macro / Scene Control Macro

Supported Versions

Version 3.50 or later

Related Items

ImageUpdate (Reference: ▶ Details (p.568))
RefreshTextWindow (Reference: ▶ Details (p.655))
SetImageWindow (Reference: ▶ Details (p.704))
RefreshJudgeWindow (Reference: ▶ Details (p.654))
RefreshTimeWindow (Reference: ▶ Details (p.656))

RefreshJudgeWindow

Updates the judgement window.

Format

RefreshJudgeWindow

Parameter

None.

Return value

None.

Description

Updates the judgement window.

Usage Cautions

- None.

Example

Updates the judgement window in the communication command macro.

```
RefreshJudgeWindow
```

Useable Modules

Communication Command Macro / Scene Control Macro

Supported Versions

Version 3.50 or later

Related Items

RefreshImageWindow (Reference: ▶ Details (p.653)) RefreshTextWindow (Reference: ▶ Details (p.655)) RefreshTimeWindow (Reference: ▶ Details (p.656))

RefreshTextWindow

Updating the text display window.

Format

RefreshTextWindow

Parameter

None.

Return value

None.

Description

Updating the text display window.
Execute this macro function to update the display after redrawing detailed results in the text window.

Usage Cautions

- None.

Example

In the communication macro, sets the value of the measurement result of calculation expression 0 of the calculation processing unit of Processing Unit number 5 of the current scene, and then updating the display of the text window."Calculation result of calculation expression 0" is external reference data number 5.

```
Rem Set the value as the measurement result.
SetUnitData 5, 5, 100

Rem Update the text window display.
RefreshTextWindow
```

Useable Modules

Communication Command Macro / Scene Control Macro

Supported Versions

Version 3.50 or later

Related Items

RefreshImageWindow (Reference: ▶ Details (p.653)) RefreshJudgeWindow (Reference: ▶ Details (p.654))
RefreshTimeWindow (Reference: ▶ Details (p.656)) SetTextWindow (Reference: ▶ Details (p.729))

RefreshTimeWindow

Updates the display of the information window.

Format

RefreshTimeWindow

Parameter

None.

Return value

None.

Description

Updates the display of the information window.

Usage Cautions

- None.

Example

Updates the display of the information window in the communication macro.

```
RefreshTimeWindow
```

Useable Modules

Communication Command Macro / Scene Control Macro

Supported Versions

Version 3.50 or later

Related Items

RefreshImageWindow (Reference: ▶ Details (p.653)) RefreshJudgeWindow (Reference: ▶ Details (p.654))
RefreshTextWindow (Reference: ▶ Details (p.655)) SetTextWindow (Reference: ▶ Details (p.729))

Rem

Put a comment in the program.

Format

Rem

Parameter

None.

Return value

None.

Description

Add a comment or a description in the program. The readability of the program is improved by adding comments.

For details on comment, refer to the "Comment" section. (Reference: ▶Comment (p.207))

Usage Cautions

- Do not mix a non-comment statement together with a comment on the same line. If a comment and another type of statement are written in one same line, the comment may not be correctly recognized and the program may not operate properly.

Example

Inserts a comment statement in a program to describe the program process.

```
Rem Display the judgement result provided by the latest executed processing unit on the system status console window.
Print UnitJudge(UnitNo - 1)
```

Useable Modules

Unit Calculation Macro / Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 3.50 or later

Related Items

Print (Reference: ▶ Details (p.639))
UnitJudge (Reference: ▶ Details (p.788))
UnitNo (Reference: ▶ Details (p.790))

Remeasure

Executes remeasurement.

Format

Remeasure <preImageNo>[, <wait>]
Remeasure <fileName>[, <wait>]

Parameter

Parameter name	Data type	Description
<preImageNo>	Integer type	Image logging number (-1 to (number of images in main unit already logged minus one)) of image to be remeasured.
<fileName>	Character string type	Image file name of image to be remeasured.
<wait>	Integer type	Recovery timing of macro function 0: Executes subsequent program lines without waiting for measurement to end. 1: Waits for measurement to end and then executes subsequent program lines. 2: Waits for measurement to end and measurement result display to end and then executes subsequent program lines.

Return value

None.

Description

Executes remeasurement at the recovery timing specified in the <wait> parameter for the image of the image logging number specified in the <preImageNo> parameter or the image of the image file name specified in the <fileName> parameter.
In the <preImageNo> parameter, specify the number of the logging image already logged as a logging image in the main unit. (Reference: ▶Setting Logging Conditions [Logging Setting] (p.115))
If -1 is specified for the <preImageNo> parameter, remeasurement is executed using the most recent input image.
If 0 is specified for the <preImageNo> parameter, remeasurement is executed using the most recent logged image of the main unit.
If the <wait> parameter is omitted, operation is the same as when the <wait> parameter is set to 0.
When 0 is specified for the <wait> parameter, there is a possibility that the measurement processing executed immediately after execution of this macro function will not have ended and the measurement result cannot be properly gotten. If you want to get the measurement result, specify 1 or 2 for the <wait> parameter.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
Even if a non-existent number, numerical value, or combination of data types is specified for the parameter, an "Illegal function call" error will not occur.
If a value outside the range -2147483648 to 2147483647 is specified as an integer parameter, an "Overflow" error will occur.
If a character string longer than 255 characters is specified for a character string parameter, a "String too long" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- Execute this macro function when a measurement-in-progress signal such as the BUSY signal is OFF and measurement is allowed. (Reference: ▶State Transitions and Execution Timing (p.227))

Example

In the communication command macro, gets the measurement X coordinate and measurement Y coordinate of the search processing unit of Processing Unit number 2 after remeasurement is executed using the most recent logging image. The measured X and Y coordinates can be gotten with External Reference Data numbers 6 and 7 respectively.

```
Rem Execute remeasurement and wait until measurement ends.
Remeasure 0, 1

Rem Get the measurement result.
GetUnitData 2, 6, POSX#
GetUnitData 2, 7, POSY#
```

Useable Modules

Scene Control Macro / Communication Command Macro

Supported Versions

Version 3.50 or later

Related Items

MeasureStart (Reference: ▶ Details (p.617))
Measure (Reference: ▶ Details (p.612))
MeasureStop (Reference: ▶ Details (p.619))

RenumUnitNo

Gets the processing unit number after flow edit.

Format

RenumUnitNo(<oldUnitNo>)

Parameter

Parameter name	Data type	Description
<oldUnitNo>	Integer type	Processing unit number before edit (0 to (Processing unit number of current scene minus one))

Return value

Returns the processing unit number after update of measurement flow as an integer.

Description

Gets the processing unit number specified in the <oldUnitNo> parameter after editing the measurement flow.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
Even if a non-existent number, numerical value, or combination of data types or values is specified for the parameter, an error will not occur.
If a value outside the range -2147483648 to 2147483647 is specified as an integer parameter, an "Overflow" error will occur.
If a value is assigned to the return value variable or the variable is not used in an expression, a "Syntax error" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.
Besides using this macro function, using reference variables follows the changes of the processing unit numbers resulting from editing the measurement flow. (Reference: ▶Variable (p.209))

Usage Cautions

- This macro function can only be used in the *RENUMPROC subroutine. If used in another subroutine, an "Illegal function call" will occur.

Example

Uses the *RENUMPROC subroutine in the Unit Macro processing unit to get the processing unit number (whose original processing unit number was 5) after editing the measurement flow.

```
*RENUMPROC

  LATESTUNITNO& = 5

  Rem Get the processing unit number after flow edit.
  LATESTUNITNO& = RenumUnitNo(LATESTUNITNO&)

Return
```

Useable Modules

Unit Calculation Macro / Unit Macro

Supported Versions

Version 3.50 or later

Related Items

DeleteUnit (Reference: ▶ Details (p.460))
UnitNo (Reference: ▶ Details (p.790))
InsertUnit (Reference: ▶ Details (p.573))
Ut (Reference: ▶ Details (p.793))

RGB

Gets the color value.

Format

RGB(<red>, <green>, <blue>)

Parameter

Parameter name	Data type	Description
<red>	Integer type	Red component of the color value being gotten (0 to 255)
<green>	Integer type	Green component of the color value being gotten (0 to 255)
<blue>	Integer type	Blue component of the color value being gotten (0 to 255)

Return value

Returns the color value as an integer value.
The red component is stored in the lower byte of the color value, green component is stored is stored in the middle byte of the color value, and blue comonent is stored in the upper byte of the color value.

Description

Gets the color value of the color that has the red component specified in the <red> parameter, the green component specified in the <green> parameter, and the blue component specified in the <blue> parameter. A drawing color can be specified in specifying a color value in a macro function that sets a drawing style and in some drawing macro functions.
If a value that exceeds the allowed setting range is specified in the <red> parameter, <green> parameter, or <blue> parameter, the value is handled as being 255.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
Even if a non-existent number, numerical value, or combination of data types or values is specified for the parameter, an error will not occur.
If a value is assigned to the return value variable or the variable is not used in an expression, a "Syntax error" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

Displays a green character string in the text window.

```
Rem Get the color value of green.
COLOR& = RGB(0, 255, 0)

Rem Draw the character string in the text window.
DrawText "Processing OK", COLOR&, 1
```

Useable Modules

Unit Calculation Macro / Unit Macro

Supported Versions

Version 3.50 or later

Related Items

DrawFillImage (Reference: ▶ Details (p.485))
SetDrawStyle (Reference: ▶ Details (p.700))
DrawText (Reference: ▶ Details (p.500))
SetTextStyle (Reference: ▶ Details (p.727))

Right$

Extracts the specified length of characters from the right side of character string.

Format

Right$(<string>, <length>)

Parameter

Parameter name	Data type	Description
<string>	Character string type	Extraction target character string
<length>	Integer type	Length of characters to be extracted (1 to the length of the target character string)

Return value

Returns the character string type value of the extracted character string.

Description

Extracts the specified length in the <length> parameter from the right side of specified character string in the <string> parameter.
Specify the length of characters to be extracted in bytes for the <length> parameter. Each single-byte character (i.e., half-width alphanumeric character and symbol) consumes one byte, whereas each double-byte character consumes two bytes.
If the length specified in the <length> parameter is longer than the length of the character string specified in the <string> parameter, the whole character string in the parameter is extracted.
If 0 or less number is specified in the <length> parameter, an "Illegal function call" error will occur.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a value outside the range -2147483648 to 2147483647 is specified as an integer parameter, an "Overflow" error will occur.
Even if a character string longer than 255 characters is specified for a character string parameter, an error will not occur.
If a value is assigned to the return value variable or the variable is not used in an expression, a "Syntax error" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

Extracts 6-byte length of characters from the right side of the character string. Because one half-width alphabet consumes single byte, this example extracts 6 characters from the character string.

```
CHARA$ = "Measurement Result"

Rem Extract 6-byte length of characters from the right side of the character string.
TITLE$ = Right$(CHARA$, 6)
```

The result is shown below.

```
TITLE$ = "Result"
```

Useable Modules

Unit Calculation Macro / Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 3.50 or later

Related Items

Asc (Reference: ▶ Details (p.413))
Chr$ (Reference: ▶ Details (p.427))
Hex$ (Reference: ▶ Details (p.560))
LCase$ (Reference: ▶ Details (p.589))
Left$ (Reference: ▶ Details (p.591))
Len (Reference: ▶ Details (p.593))
Mid$ (Reference: ▶ Details (p.621))
Piece$ (Reference: ▶ Details (p.637))
Str$ (Reference: ▶ Details (p.749))
Str2$ (Reference: ▶ Details (p.751))
UCase$ (Reference: ▶ Details (p.775))
Val (Reference: ▶ Details (p.794))

Rmdir

Deletes a directory.

Format

Rmdir <directoryName>

Parameter

Parameter name	Data type	Description
<directoryName>	Character string type	Name of a directory to be deleted

Return value

None.

Description

Deletes the directory specified in the <directoryName> parameter.
In the <directoryName> parameter, use an absolute path to specify the directory name of the directory to be deleted.
In the following case, an "Illegal function call" error occurs without deleting a directory.

- The specified directory does not exist
- If the external memory is specified for where to delete a directory from with no external memory inserted
- If more than one file is in the specified directory

If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a character string longer than 255 characters is specified for a character string parameter, a "String too long" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

Deletes a directory named "IMAGE2" under the root folder of the E drive.

```
Rmdir "E:\IMAGE2"
```

Useable Modules

Unit Calculation Macro / Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 3.50 or later

Related Items

Dskf (Reference: ▶ Details (p.506))
GetSystemData (Reference: ▶ Details (p.546))
Kill (Reference: ▶ Details (p.587))
Fcopy (Reference: ▶ Details (p.522))
IsFile (Reference: ▶ Details (p.577))
Mkdir (Reference: ▶ Details (p.623))

RunOut

Sets the output state of the RUN signal.

Format

RunOut <ioIdent>, <state>

Parameter

Parameter name	Data type	Description
<ioIdent>	Character string type	Identification name of the communication module to be used ("ParallelIo" or "EtherCAT") (Reference: ▶List of I/O Modules (p.341))
<state>	Integer type	Output state of terminal 0: Output OFF 1: Output ON

Return value

None.

Description

Sets the RUN signal of the communication module specified in the <ioIdent> parameter to the output state specified in the <state> parameter.
Normally "ParallelIo" or "EtherCAT" should be specified in the <ioIdent> parameter.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If an identification name that does not exist is specified in the <ioIdent> parameter, an "Illegal function call" error will occur.
Even if an output status parameter value that does not exist (i.e., other than 0 and 1) is specified in the <state> parameter, an error will not occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

In the communication command macro, sets the BUSY signal of parallel I/O to ON.

```
IOMODULE$ = "ParallelIo"

Rem Set the output state.
RunOut IOMODULE$, 1
```

Useable Modules

Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 3.50 or later

Related Items

BusyOut (Reference: ▶ Details (p.419))
GetPort (Reference: ▶ Details (p.542))
PutAll (Reference: ▶ Details (p.643))
GetAll (Reference: ▶ Details (p.528))
JudgeOut (Reference: ▶ Details (p.585))
PutPort (Reference: ▶ Details (p.645))

SaveBackupData

Saves the system + scene group 0 data.

Format

SaveBackupData(<fileName>)

Parameter

Parameter name	Data type	Description
<fileName>	Character string type	File name of bkd file to save (system + scene group 0 data (*.bkd))

Return value

None.

Description

Saves the system + scene group 0 in the file with the file name specified in the <fileName> parameter.
In the <fileName> parameter, use an absolute path to specify the file name of the file to be saved.
Specify the file extension ".bkd" in the file name specified in the <fileName> parameter.
If the file specified in the <fileName> parameter already exists, it is overwritten.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a character string longer than 255 characters is specified for a character string parameter, a "String too long" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- Execute this macro function when the BUSY signal or other measurement in progress signal is ON and measurement is prohibited. (Reference: ▶State Transitions and Execution Timing (p.227))

Example

Saves the system + scene group 0 data to a file.

```
Rem Save the system + scene group 0 data to a file.
SaveBackupData "C:\BACKDIR/BackupData.bkd"
```

Useable Modules

Scene Control Macro / Communication Command Macro

Supported Versions

Version 3.50 or later

Related Items

GetSystemData (Reference: ▶ Details (p.546))
SaveData (Reference: ▶ Details (p.670))
SaveSceneGroup (Reference: ▶ Details (p.677))
SaveUnitData (Reference: ▶ Details (p.681))
LoadBackupData (Reference: ▶ Details (p.599))
SaveScene (Reference: ▶ Details (p.675))
SaveSystemData (Reference: ▶ Details (p.679))

SaveData

Saves the data to the controller.

Format

SaveData

Parameter

None.

Return value

None.

Description

Saves the current system group data and the system data to the sensor controller.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- Execute this macro function when the BUSY signal or other measurement in progress signal is ON and measurement is prohibited. (Reference: ▶State Transitions and Execution Timing (p.227))

Example

Saves the current system group data and the system data to the sensor controller.

```
Rem Save the data to the controller.
SaveData
```

Useable Modules

Scene Control Macro / Communication Command Macro

Supported Versions

Version 3.50 or later

Related Items

SaveBackupData (Reference: ▶ Details (p.669))
SaveSceneGroup (Reference: ▶ Details (p.677))
SaveUnitData (Reference: ▶ Details (p.681))
SaveScene (Reference: ▶ Details (p.675))
SaveSystemData (Reference: ▶ Details (p.679))

SaveImage

Saves image data.

Format

SaveImage <preImageNo>, <fileName>

Parameter

Parameter name	Data type	Description
<preImageNo>	Integer type	Number (-1 to (number of images already logged in main unit minus one)) of main unit logging image to be saved
<fileName>	Character string type	File name of file to be saved

Return value

None.

Description

Saves the image that has the image logging number specified in the <preImageNo> parameter, using the file name specified in the <fileName> parameter, in ifz format.
In the <preImageNo> parameter, specify the number of the logging image already logged as a logging image in the main unit. (Reference: ▶Setting Logging Conditions [Logging Setting] (p.115))
If -1 is specified for the <preImageNo> parameter, the most recent input image is saved.
In the <fileName> parameter, use an absolute path to specify the file name of the file to be saved.
Specify the file extension ".ifz" in the file name specified in the <fileName> parameter.
If the file of the file name specified in the <fileName> parameter already exists, it is overwritten.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a non-existent number, numerical value, or combination of data types or values is specified for a parameter, an "Illegal function call" error will occur.
If a value outside the range -2147483648 to 2147483647 is specified as an integer parameter, an "Overflow" error will occur.
If a character string longer than 255 characters is specified for a character string parameter, a "String too long" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

Saves the most recent measurement image.

```
Rem Save the most recent input image as a file.
SaveImage -1, "C:\IMAGE\sample.ifz"
```

Useable Modules

Scene Control Macro / Communication Command Macro

Supported Versions

Version 3.50 or later

Related Items

GetSystemData (Reference: ▶ Details (p.546))

Remeasure (Reference: ▶ Details (p.658))

SaveMeasureImage

Saves the measurement image of the processing unit.

Format

SaveMeasureImage <measureImageNo>, <fileName>, <imageFormat>[, <startX>, <startY>, <sizeX>, <sizeY>]

Parameter

Parameter name	Data type	Description
<measureImageNo>	Integer type	Measurement image number of the measurement image to be saved (always 0)
<fileName>	Character string type	File name of file to be saved
<imageFormat>	Integer type	Image format of image to be saved 0: BMP format 10000 to 10100: JPEG format (10000 + JPEG image quality (0 to 100))
<startX>	Integer type	Start point X of image region to be saved
<startY>	Integer type	Start point Y of image region to be saved
<sizeX>	Integer type	X dimension of image to be saved (at least 1)
<sizeY>	Integer type	Y dimension of image to be saved (at least 1)

Return value

None.

Description

Saves the measurement image specified in the <measureImageNo> parameter in a file with the file name specified in the <fileName> parameter, the image format specified in the <imageFormat> parameter, and the pixel size specified in the <sizeX> and <sizeY> parameters cut off from the position in camera coordinates that starts from the upper left point specified in the <startX> parameter and <startY> parameter.
If the <start X>, <start Y>, <size X>, and <size Y> parameters are omitted, the entire image is saved.
Always specify 0 in the <measureImageNo> parameter.
In the <fileName> parameter, use an absolute path to specify the file name of the file to be saved.
In the file name specified in the <fileName> parameter, specify the file extension ".bmp" or ".jpg/jpeg".
If the file of the file name specified in the <fileName> parameter already exists, it is overwritten.
Specify a value of at least 1 in the <sizeX> and <sizeY> parameters.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a non-existent number, numerical value, or combination of data types or values is specified for a parameter, an "Illegal function call" error will occur.
If a value outside the range -2147483648 to 2147483647 is specified as an integer parameter, an "Overflow" error will occur.
If a character string longer than 255 characters is specified for a character string parameter, a "String too long" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

Saves the entire image of measurement image 0 in a file in BMP format.

```
SaveMeasureImage 0, "C:\IMAGE\sample.bmp", 0
```

Useable Modules

Unit macro

Supported Versions

Version 4.00 or later

Related Items

GetImageSize (Reference: ▶ Details (p.532))
GetSystemData (Reference: ▶ Details (p.546))
GetUnitData (Reference: ▶ Details (p.552))

SaveScene

Saves the scene data.

Format

SaveScene <sceneNo>, <fileName>

Parameter

Parameter name	Data type	Description
<sceneNo>	Integer type	Scene number to save the scene (0 to 127)
<fileName>	Character string type	File name of the scene data to save (*.scn)

Return value

None.

Description

Saves the scene data of the scene number specified in the <sceneNo> parameter in the file with the file name specified in the <fileName> parameter.
In the <fileName> parameter, use an absolute path to specify the file name of the file to be saved.
Specify the file extension ".scn" in the file name specified in the <fileName> parameter.
If the file specified in the <fileName> parameter already exists, it is overwritten.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a non-existent number, numerical value, or combination of data types or values is specified for a parameter, an "Illegal function call" error will occur.
If a character string longer than 255 characters is specified for a character string parameter, a "String too long" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- Execute this macro function when the BUSY signal or other measurement in progress signal is ON and measurement is prohibited. (Reference: ▶State Transitions and Execution Timing (p.227))

Example

Saves the scene data of scene 2 in a file

```
Rem Save the scene data of scene 2 in a file.
SaveScene 2, "C:\BACKDIR\scene02.scn"
```

Useable Modules

Scene Control Macro / Communication Command Macro

Supported Versions

Version 3.50 or later

Related Items

GetSystemData (Reference: ▶ Details (p.546))
SaveBackupData (Reference: ▶ Details (p.669))
SaveSceneGroup (Reference: ▶ Details (p.677))
SaveUnitData (Reference: ▶ Details (p.681))
LoadScene (Reference: ▶ Details (p.601))
SaveData (Reference: ▶ Details (p.670))
SaveSystemData (Reference: ▶ Details (p.679))
SceneNo (Reference: ▶ Details (p.691))

SaveSceneGroup

Saves the scene group data.

Format

SaveSceneGroup <sceneGroupNo>, <fileName>

Parameter

Parameter name	Data type	Description
<sceneGroupNo>	Integer type	Scene group number of the scene group to save (0 to 31)
<fileName>	Character string type	File name of the scene group data to save (*.sgp)

Return value

None.

Description

Saves the scene group data of the scene group number specified in the <sceneNo> parameter in the file with the file name specified in the <fileName> parameter.
In the <fileName> parameter, use an absolute path to specify the file name of the file to be saved.
Specify the file extension ".sgp" in the file name specified in the <fileName> parameter.
If the file specified in the <fileName> parameter already exists, it is overwritten.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a non-existent number, numerical value, or combination of data types or values is specified for a parameter, an "Illegal function call" error will occur.
If a character string longer than 255 characters is specified for a character string parameter, a "String too long" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- Execute this macro function when the BUSY signal or other measurement in progress signal is ON and measurement is prohibited. (Reference: ▶State Transitions and Execution Timing (p.227))

Example

Saves the scene group data of scene group 2 in a file

```
Rem Save the scene group data of scene group 2 in a file.
SaveSceneGroup 2, "C:\BACKDIR\scenegroup02.sgp"
```

Useable Modules

Scene Control Macro / Communication Command Macro

Supported Versions

Version 3.50 or later

Related Items

GetSystemData (Reference: ▶ Details (p.546))
SaveBackupData (Reference: ▶ Details (p.669))
SaveScene (Reference: ▶ Details (p.675))
SaveUnitData (Reference: ▶ Details (p.681))
LoadSceneGroup (Reference: ▶ Details (p.603))
SaveData (Reference: ▶ Details (p.670))
SaveSystemData (Reference: ▶ Details (p.679))
SceneGroupNo (Reference: ▶ Details (p.687))

SaveSystemData

Saves the system data.

Format

SaveSystemData <fileName>

Parameter

Parameter name	Data type	Description
<fileName>	Character string type	File name of the system data to save (*.ini)

Return value

None.

Description

Saves the system data in the file with the file name specified in the <fileName> parameter.
In the <fileName> parameter, use an absolute path to specify the file name of the file to be saved.
Specify the file extension ".ini" in the file name specified in the <fileName> parameter.
If the file specified in the <fileName> parameter already exists, it is overwritten.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a character string longer than 255 characters is specified for a character string parameter, a "String too long" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- Execute this macro function when the BUSY signal or other measurement in progress signal is ON and measurement is prohibited. (Reference: ▶State Transitions and Execution Timing (p.227))

Example

Saves the system data in a file

```
Rem Save the system data in a file.
SaveSystemData "C:\BACKDIR\backupsysset.ini"
```

Useable Modules

Scene Control Macro / Communication Command Macro

Supported Versions

Version 3.50 or later

Related Items

GetSystemData (Reference: ▶ Details (p.546))
SaveBackupData (Reference: ▶ Details (p.669))
SaveScene (Reference: ▶ Details (p.675))
SaveUnitData (Reference: ▶ Details (p.681))
LoadSystemData (Reference: ▶ Details (p.605))
SaveData (Reference: ▶ Details (p.670))
SaveSceneGroup (Reference: ▶ Details (p.677))

SaveUnitData

Saves a processing unit.

Format

SaveUnitData <sceneNo>, <unitNo>, <unitCount>, <fileName>

Parameter

Parameter name	Data type	Description
<sceneNo>	Integer type	Scene number to save the scene (-1 to 127)
<unitNo>	Integer type	Processing unit number to begin to save (0 to (the number of registered processing units in the current scene minus one))
<unitCount>	Integer type	Number of pieces of the processing unit to save (-1, 1 to (the number of registered processing units in the current scene) - (the processing unit number to begin to save))
<fileName>	Character string type	File name of the processing unit to save (*unt)

Return value

None.

Description

Saves processing unit data in the file with the file name specified in the <fileName> parameter from processing units whose processing unit numbers are specified in the <unitNo> parameter, with the number of processing units specified in the <unitCount> parameter, of the scene number specified in the <sceneNo> parameter.
When -1 is specified in the <sceneNo> parameter, the scene number of the current scene is specified in the scene number of the scene to be saved.
If -1 is specified in the <unitCount> parameter, all processing unit data included in the processing unit data file is saved.
In the <fileName> parameter, use an absolute path to specify the file name of the file to be saved.
Specify the file extension ".unt" in the file name specified in the <fileName> parameter.
If the file specified in the <fileName> parameter already exists, it is overwritten.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a non-existent number, numerical value, or combination of data types or values is specified for a parameter, an "Illegal function call" error will occur.
If a character string longer than 255 characters is specified for a character string parameter, a "String too long" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- Execute this macro function when the BUSY signal or other measurement in progress signal is ON and measurement is prohibited. (Reference: ▶State Transitions and Execution Timing (p.227))

Example

Saves the processing units of Processing Unit number 2 to Processing Unit number 4 of the current scene in a processing unit data file.

```
Rem Save Processing Unit number 2 to Processing Unit number 4 of the current scene in a processing unit data file.
SaveUnitData -1, 2, 3, "C:\BACKDIR\unitsave.unt"
```

Useable Modules

Scene Control Macro / Communication Command Macro

Supported Versions

Version 3.50 or later

Related Items

GetSystemData (Reference: ▶ Details (p.546))
SaveBackupData (Reference: ▶ Details (p.669))
SaveScene (Reference: ▶ Details (p.675))
SaveSystemData (Reference: ▶ Details (p.679))
UnitNo (Reference: ▶ Details (p.790))
LoadUnitData (Reference: ▶ Details (p.607))
SaveData (Reference: ▶ Details (p.670))
SaveSceneGroup (Reference: ▶ Details (p.677))
SceneNo (Reference: ▶ Details (p.691))
Ut (Reference: ▶ Details (p.793))

SceneCount

Gets the number of scenes that can be used.

Format

SceneCount

Parameter

None.

Return value

Returns the number of scenes that can be used as an integer.

Description

Gets the number of scenes that can be used.
If a value is assigned to the return value variable or the variable is not used in an expression, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

Gets the number of scenes that can be used.

```
NUM& = SceneCount
```

Useable Modules

Scene Control Macro / Communication Command Macro

Supported Versions

Version 3.50 or later

Related Items

ChangeScene (Reference: ▶ Details (p.423))
CopyScene (Reference: ▶ Details (p.440))
ClearScene (Reference: ▶ Details (p.430))
SceneNo (Reference: ▶ Details (p.691))

SceneDescription$

Gets the scene description.

Format

SceneDescription$(<sceneNo>)

Parameter

Parameter name	Data type	Description
<sceneNo>	Integer type	Scene number (0 to 127) of scene whose description is to be gotten.

Return value

Returns the scene description as a character string.

Description

Gets the description set in the scene of the scene number specified in the <sceneNo> parameter.
If a description is not set, the null character string ("") is returned.
The scene description can be set in the maintenance screen or by executing the SetSceneDescription function. (Reference: ▶SetSceneDescription (p.715)) (Reference: ▶Editing Scenes (p.62))
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a non-existent number, numerical value, or combination of data types or values is specified for a parameter, an "Illegal function call" error will occur.
If a value is assigned to the return value variable or the variable is not used in an expression, a "Syntax error" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

Gets the description of scene 1, and if a description is not set, setting the description.

```
Rem Get the scene description
DESCRIPTION$ = SceneDescription$(1)

If DESCRIPTION$ = "" Then
  Rem Set the scene description
  SetSceneDescription 1, "Description 1"
Endif
```

Useable Modules

Control Macro / Communication Command Macro

Supported Versions

Version 3.50 or later

Related Items

SceneMaker$ (Reference: ▶ Details (p.689))
SceneTitle$ (Reference: ▶ Details (p.692))
SetSceneDescription (Reference: ▶ Details (p.715))
SetSceneMaker (Reference: ▶ Details (p.719))
SetSceneTitle (Reference: ▶ Details (p.721))

SceneGroupCount

Gets the number of useable scene groups.

Format

SceneGroupCount

Parameter

None.

Return value

The number of useable scene groups is returned as an integer value.

Description

Gets the number of useable scene groups.
If a value is assigned to the return value variable or the variable is not used in an expression, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

Gets the number of useable scene groups.

```
NUM& = SceneGroupCount
```

Useable Modules

Scene Control Macro / Communication Command Macro

Supported Versions

Version 3.50 or later

Related Items

ChangeSceneGroup (Reference: ▶ Details (p.424))
ClearSceneGroup (Reference: ▶ Details (p.431))
CopySceneGroup (Reference: ▶ Details (p.441))
SceneGroupNo (Reference: ▶ Details (p.687))

SceneGroupNo

Gets the scene group number of the current scene group.

Format

SceneGroupNo

Parameter

None.

Return value

The scene group number of the current scene group is returned as an integer value.

Description

Gets the scene group number of the current scene group.
If a value is assigned to the return value variable or the variable is not used in an expression, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

Gets the scene group number of the current scene group, and if the scene group number is 2, change to scene 3.

```
Rem Get the scene group number of the current scene group.
NO& = SceneGroupNo

Rem Get the scene group number, and if 2, change to scene 3
If NO& = 2 Then
  ChangeScene3
Endif
```

Useable Modules

Scene Control Macro / Communication Command Macro

Supported Versions

Version 3.50 or later

Related Items

ChangeScene (Reference: ▶ Details (p.423))
ChangeSceneGroup (Reference: ▶ Details (p.424))
ClearSceneGroup (Reference: ▶ Details (p.431))
CopySceneGroup (Reference: ▶ Details (p.441))
SceneGroupCount (Reference: ▶ Details (p.686))
SceneNo (Reference: ▶ Details (p.691))

SceneGroupTitle$

Gets the title of the scene group.

Format

SceneGroupTitle$(<sceneGroupNo>)

Parameter

Parameter name	Data type	Description
<sceneGroupNo>	Integer type	Scene group number (0 to 31) of the scene group whose scene group title is gotten.

Return value

Returns the title value of the character string scene group.

Description

Gets the title set in the scene group that has the scene group number specified in the <sceneGroupNo> parameter.
If the title is not set, returns the default character string such as "scene group 0".
The scene group title can be set by executing the SetSceneGroupTitle function, or in the scene maintenance screen. (Reference: ▶SetSceneMaker (p.719)) (Reference: ▶Editing Scenes (p.62))
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a non-existent number, numerical value, or combination of data types or values is specified for a parameter, an "Illegal function call" error will occur.
If a value is assigned to the return value variable or the variable is not used in an expression, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

Gets the title of scene group 2.

```
TITLE$ = SceneGroupTitle$(2)
```

Useable Modules

Scene Control Macro / Communication Command Macro

Supported Versions

Version 3.50 or later

Related Items

SceneTitle$ (Reference: ▶ Details (p.692))

SetSceneGroupTitle (Reference: ▶ Details (p.717))

SceneMaker$

Gets the scene creator.

Format

SceneMaker$(<sceneNo>)

Parameter

Parameter name	Data type	Description
<sceneNo>	Integer type	Scene number (0 to 127) of scene whose creator is to be gotten.

Return value

Returns the value of the scene creator as a character string.

Description

Gets the name of the creator set in the scene that has the screen number specified in the <sceneNo> parameter.
If a creator name is not set, returns the null character string ("").
The scene creator can be set in the scene maintenance screen, or by executing the SetSceneMaker function. (Reference: ▶SetSceneMaker (p.719)) (Reference: ▶Editing Scenes (p.62))
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a non-existent number, numerical value, or combination of data types or values is specified for a parameter, an "Illegal function call" error will occur.
If a value is assigned to the return value variable or the variable is not used in an expression, a "Syntax error" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

Gets the creator of scene 3, and if not set, setting the creator.

```
Rem Get the creator of the scene.
NAME$ = SceneMaker$(3)

If NAME$ = "" Then
  Rem Set the creator of the scene.
  SetSceneMaker 3, "Maker"
Endif
```

Useable Modules

Scene Control Macro / Communication Command Macro

Supported Versions

Version 3.50 or later

Related Items

SceneDescription$ (Reference: ▶ Details (p.684))
SceneTitle$ (Reference: ▶ Details (p.692))
SetSceneDescription (Reference: ▶ Details (p.715))
SetSceneMaker (Reference: ▶ Details (p.719))
SetSceneTitle (Reference: ▶ Details (p.721))

SceneNo

Gets the scene number of the current scene.

Format

SceneNo

Parameter

None.

Return value

Returns the scene number of the current scene as an integer value.

Description

Gets the scene number of the current scene.
If a value is assigned to the return value variable or the variable is not used in an expression, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

Gets the scene number of the current scene, and if not 2, changing to scene 2.

```
Rem Gets the scene number of the current scene
NO& = SceneNo

Rem If the scene number is not 2, change to scene 2
If NO& <> 2 Then
  ChangeScene 2
Endif
```

Useable Modules

Scene Control Macro / Communication Command Macro

Supported Versions

Version 3.50 or later

Related Items

ChangeScene (Reference: ▶ Details (p.423))
ClearScene (Reference: ▶ Details (p.430))
CopyScene (Reference: ▶ Details (p.440))
SceneCount (Reference: ▶ Details (p.683))

SceneTitle$

Gets the scene title.

Format

SceneTitle$(<sceneNo>)

Parameter

Parameter name	Data type	Description
<sceneNo>	Integer type	Scene number (0 to 127) of scene whose title is to be gotten

Return value

Returns the scene title as a character string.

Description

Gets the title set in the scene that has the scene number specified in the <sceneNo> parameter.
If a title is not set, returns the default character string, such as "Scene0".
The scene title can be set in the scene maintenance screen or flow edit screen, or by executing the SetSceneTitle function. (Reference: ▶SetSceneMaker (p.719)) (Reference: ▶Editing Scenes (p.62)) (Reference: ▶Editing Processing Units in Scenes (p.53))
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a non-existent number, numerical value, or combination of data types or values is specified for a parameter, an "Illegal function call" error will occur.
If a value is assigned to the return value variable or the variable is not used in an expression, a "Syntax error" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

Gets the title of scene 2.

```
TITLE$ = SceneTitle$(2)
```

Useable Modules

Scene Control Macro / Communication Command Macro

Supported Versions

Version 3.50 or later

Related Items

SceneDescription$ (Reference: ▶ Details (p.684))
SceneMaker$ (Reference: ▶ Details (p.689))
SetSceneDescription (Reference: ▶ Details (p.715))
SetSceneMaker (Reference: ▶ Details (p.719))
SetSceneTitle (Reference: ▶ Details (p.721))

ScreenCapture

Saves the capture of the screen.

Format

ScreenCapture <fileName>

Parameter

Parameter name	Data type	Description
<fileName>	Character string type	File name that saves the capture of the screen.

Return value

None.

Description

Takes a screen capture of the sensor controller screen and saves it in BMP format with the file name specified in the <fileName> parameter.
In the <fileName> parameter, use an absolute path to specify the file name of the file to be saved.
In the <fileName> parameter, specify the file name with the file extension ".bmp" to save as BMP. It is not possible to capture the screen correctly if an extension other than ".bmp" is specified.
If the file of the file name specified in the <fileName> parameter already exists, it is overwritten.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a character string longer than 255 characters is specified for a character string parameter, a "String too long" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

Captures the screen and save the screen capture to a file with a file name "E:\IMAGE\samplecapture.bmp".

```
ScreenCapture "E:\IMAGE\samplecapture.bmp"
```

Useable Modules

Scene Control Macro / Communication Command Macro

Supported Versions

Version 3.50 or later

Related Items

GetSystemData (Reference: ▶ Details (p.546))
Str$ (Reference: ▶ Details (p.749))
Str2$ (Reference: ▶ Details (p.751))

Select Case Case Else End Select

Controls the process flow according to the specified condition.

Format

Select <expression>
[Case <value>
<caseStatement>]
:
:
[Case Else
<elseStatement>]
End Select

Parameter

Parameter name	Data type	Description
<expression>	Integer type	Expression that controls the process flow.
<value>	Integer type	Numeric value that is compared with the expression value
<caseStatement>	---	Statement that is executed when a result value of the expression and the numeric value match
<elseStatement>	---	Statement that is executed when any of numeric values did not match a result value of the expression.

Return value

None.

Description

Among the multiple Case block statement in the statement, executes the statements whose specified value in the <value> parameter match the value of the specified expression in the <expression> parameter.
If any of values specified in the <value> parameters did not match the result value of the specified expression in the <expression> parameter, the Case Else block statement specified in the <elseStatement> parameter is executed.
If there are multiple of <value> parameters having a value that matches a result value of the expression, only the first statement from the beginning of the Case block statement having a value that matches a result value of the expression is executed.
Case block statements and Case Else block statement are optional.
If the program process is jumped into or out of the Case and Case Else block statements using the Goto function in a statement, unexpected operation may occur.
If neither the Select statement nor the End Select statement is used, either the "CASE without SELECT", "END SELECT without SELECT", "SELECT without END SELECT", or "CASE without END SELECT" error will occur depending on the statement that is used.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

Uses the *MEASUREDISPG subroutine in the Unit Macro processing unit to change the display in the image window according to the set image display sub-number in the image window of the main screen.

```
*MEASUREDISPG

  Rem Get the displayed sub-image number
  SUBNO& = DisplaySubNo

  Rem Change the display on the image window according to the sub-image number of the sub-image to be displayed.
  Select SUBNO&
  Case 1
    Rem If the gotten sub-image number is 1, the title of processing unit 1 is displayed with the color in accordance with
    the judgment result.
    SetTextStyle 24, TA_LEFT, UnitJudge(1), 0, FONTSTYLE_NORMAL
    TEXT$ = UnitTitle$(1)
  Case 2
    Rem If the gotten sub-image number is 2, the title of processing unit 2 is displayed with the color in accordance with
    the judgment result.
    SetTextStyle 24, TA_LEFT, UnitJudge(2), 0, FONTSTYLE_NORMAL
    TEXT$ = UnitTitle$(2)
  Case Else
    Rem If the gotten sub-image number is other than 1 and 2, "Error" is displayed in the "unmeasured" color.
    SetTextStyle 24, TA_LEFT, JUDGE_NC, 0, FONTSTYLE_NORMAL
    TEXT$ = "Error"

  Rem Displays text on the image window.
  DrawTextG TEXT$, 50, 0, 0, UnitNo

Return
```

Useable Modules

Unit Calculation Macro / Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 3.50 or later

Related Items

DisplaySubNo (Reference: ▶ Details (p.462))
Gosub (Reference: ▶ Details (p.556))
If Then Elseif Else EndIf (Reference: ▶ Details (p.564))
UnitJudge (Reference: ▶ Details (p.788))
DrawTextG (Reference: ▶ Details (p.502))
If Then Else (Reference: ▶ Details (p.562))
SetTextStyle (Reference: ▶ Details (p.727))
UnitNo (Reference: ▶ Details (p.790))

SendData

Sends data.

Format

SendData <ioIdent>, <outputData()>, <outputSize>[, <parameter()>, <parameterSize>]

Parameter

Parameter name	Data type	Description
<ioIdent>	Character string type	Identification name of the communication module to be used (Reference: ▶List of I/O Modules (p.341))
<outputData()>	Integer array	Data to send
<outputSize>	Integer type	Data size to send
<parameter()>	Integer array	The parameter data specified in the option (Reference: ▶List of I/O Modules (p.341))
<parameterSize>	Integer type	Size of parameter data specified optionally (Reference: ▶List of I/O Modules (p.341))

Return value

None.

Description

Sends the amount, specified in the <outputSize> parameter, of the data specified in the <outputData()> parameter by using the communication module specified in the <ioIdent> parameter.
In the <outputData()> parameter, specify the 1D integer array variable that stores the data to be sent, without adding element numbers but adding () to the variables.
Values that can be set in the <parameter()> parameter and <parameterSize> parameter depend on the communication module specified in the <ioIdent> parameter. For details, refer to (Reference: ▶List of I/O Modules (p.341)) The <parameter()> parameter and <parameterSize> parameter can be omitted.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a non-existent number, numerical value, or combination of data types or values is specified for a parameter, an "Illegal function call" error will occur.
If Handshake is on and data sending fails due to a communication timeout or other reason, an "Illegal function call" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

In normal UDP communication, specifies the destination and sends data.

```
Rem Create the destination address information (10.1.1.101)
Dim IPADDR&(4)
IPADDR&(0) = 10
IPADDR&(1) = 1
IPADDR&(2) = 1
IPADDR&(3) = 101

Rem Make the transmit data.
Dim BUFFER&(4)
BUFFER&(0) = 1
BUFFER&(1) = 2
BUFFER&(2) = 3
BUFFER&(3) = 4
BUFFER&(4) = 5

Rem Transmit the data selected address.
SendData "UdpNormal", BUFFER&(), 4 * 5, IPADDR&(), 4 * 4
```

In PLC link, specifying the offset value and writing data to the data output area.

```
Rem Create the offset data.
Dim OFFSET&(0)
OFFSET&(0) = 2

Rem Make the transmit data.
Dim BUFFER&(4)
BUFFER&(0) = 1
BUFFER&(1) = 2
BUFFER&(2) = 3
BUFFER&(3) = 4
BUFFER&(4) = 5

Rem Use the offset value to send the data.
SendData "SerialPlcLink", BUFFER&(), 4 * 5, OFFSET&(), 4 * 1
```

Useable Modules

Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 3.50 or later

Related Items

ReceiveData (Reference: ▶ Details (p.651))

SendString (Reference: ▶ Details (p.698))

SendString

Sends the character string data.

Format

SendString <ioIdent>, <outputString>

Parameter

Parameter name	Data type	Description
<ioIdent>	Character string type	Identification name of the communication module to be used (Reference: ▶List of I/O Modules (p.341))
<outputString>	Character string type	Character string to send

Return value

None.

Description

Sends the character string specified in the <outputString()> parameter by using the communication module specified in the <ioIdent> parameter.
Some communication modules do not support this macro function. (Reference: ▶List of I/O Modules (p.341))
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
Even if a non-existent number, numerical value, or combination of data types or values is specified for the parameter, an error will not occur.
If a character string longer than 255 characters is specified for a character string parameter, a "String too long" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

Sends the character string in TCP normal communication

```
Rem Make the transmit characters.
DATA$ = "Test string"

Rem Send a character string.
SendString "TcpNormal", DATA$
```

Useable Modules

Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 3.50 or later

Related Items

ReceiveData (Reference: ▶ Details (p.651))　　SendData (Reference: ▶ Details (p.696))

SetDisplayUnitNo

Sets the processing unit number in the flow window to the selected state.

Format

SetDisplayUnitNo <unitNo>

Parameter

Parameter name	Data type	Description
<unitNo>	Integer type	Processing unit number to be selected

Return value

None.

Description

Sets the processing unit of the processing unit number specified in the <unitNo> parameter to the selected state in the flow window.
When the processing unit displayed in the image window and the text window is set to "Link to Flow Display", the information of the processing unit selected in the flow window is displayed.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
Even if a non-existent number, numerical value, or combination of data types or values is specified for the parameter, an error will not occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- Execute this macro function when the BUSY signal or other measurement in progress signal is ON and measurement is prohibited. (Reference: ▶State Transitions and Execution Timing (p.227))

Example

Sets the unit specified in the command argument of the communication command macro to the selected state in the flow window.

```
Rem Select the processing unit that has the number specified in the argument of the communication command.
SetDisplayUnitNo argumentValue#(0)
```

Useable Modules

Scene Control Macro / Communication Command Macro

Supported Versions

Version 3.50 or later

Related Items

DisplayUnitNo (Reference: ▶ Details (p.464))
UnitNo (Reference: ▶ Details (p.790))
SetImageWindow (Reference: ▶ Details (p.704))
Ut (Reference: ▶ Details (p.793))

SetDrawStyle

Set the drawing attributes of the graphic figure.

Format

SetDrawStyle <style>, <width>, <color>

Parameter

Parameter name	Data type	Description
<style>	Integer type	Type of the drawn line PS_SOLID: Solid line PS_DASH: Dashed line (This selection is valid only when the specified line width is 1) PS_DOT: Dotted line PS_DASHDOT: One-dot chain line (This selection is valid only when the specified line width is 1) PS_DASHDOTDOT: Two-dot chain line (This selection is valid only when the specified line width is 1) PS_NULL: No line PS_INSIDEFRAME: Solid line (This selection is only valid for circle, wide circle, ellipse, arc, and wide arc)
<width>	Integer type	Line width of the drawn graphic line
<color>	Integer type	Line color value of the drawn graphic line JUDGE_NC: Unmeasured color (Grey) JUDGE_OK: OK judgement color (Green) JUDGE_NG: NG judgement color (Red) RGB Function: Any color

Return value

None.

Description

Sets the specified line type by the <style> parameter, the specified line width by the <width> parameter, and specified line color by the <color> parameter as the drawing attributes. Before executing the image screen window control macro function that draws graphic figure, execute this macro function to draw the graphic figure using the set drawing attribute. Use the SetTextStyle function to set the drawing attribute used for the DrawTextG function. (Reference: ▶SetTextStyle (p.727))

If any of "PS_DASH", "PS_DASHDOT", and "PS_DASHDOTDOT" is specified in the <style> parameter, specify 1 in the <width> parameter. If other than 1 is specified, a solid line will be drawn.

If circle, wide circle, ellipse, arc, wide arc is drawn with specification of "PS_INSIDEFRAME" for the <style> parameter, the figure with specified line width by the <width> parameter is drawn and diminished so that the drawn figure is within the figure. Other figure types than ones mentioned above are drawn with a solid line (i.e., the same line type as when "PS_SOLID" is specified for the <style> parameter).

The gotten color value by the RGB function can be set for the <color> parameter. (Reference: ▶RGB (p.662))

If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.

If a value outside the range -2147483648 to 2147483647 is specified as an integer parameter, an "Overflow" error will occur.

If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- This macro function can only be used in the *MEASUREDISPI subroutine or the *MEASUREDISPG subroutine. If used in another subroutine, an "Illegal function call" error will occur.

Example

Uses the *MEASUREDISPG subroutine of the Unit Macro processing unit to draw a straight line whose line type is "dashed line" and whose color is "OK Color".

```
*MEASUREDISPG

  Rem Set the draw attibutes
  SetDrawStyle PS_DASH, 1, JUDGE_OK

  Rem Draw the image
  DrawLine 100, 100, 500, 400, 0, UnitNo

Return
```

Useable Modules

Unit macro

Supported Versions

Version 3.50 or later

Related Items

DrawArc (Reference: ▶ Details (p.469))
DrawArcW (Reference: ▶ Details (p.471))
DrawBox (Reference: ▶ Details (p.473))
DrawCircle (Reference: ▶ Details (p.475))
DrawCircleW (Reference: ▶ Details (p.477))
DrawCursor (Reference: ▶ Details (p.479))
DrawEllipse (Reference: ▶ Details (p.481))
DrawFigure (Reference: ▶ Details (p.483))
DrawLine (Reference: ▶ Details (p.488))
DrawLineW (Reference: ▶ Details (p.490))
DrawPoint (Reference: ▶ Details (p.493))
DrawPolygon (Reference: ▶ Details (p.495))
DrawSearchFigure (Reference: ▶ Details (p.497))
RGB (Reference: ▶ Details (p.662))
SetTextStyle (Reference: ▶ Details (p.727))

SetGlobalData

Sets the global data.

Format

SetGlobalData <dataIdent>, <data>

Parameter

Parameter name	Data type	Description
<dataIdent>	Character string type	Identification name of the global data to set the value
<data>	Integer type Double precision real number data type Character string type	Value set in the global data

Return value

None.

Description

Sets the value specified in the <data> parameter in the global data with the identification name specified in the <dataIdent> parameter.
If global data with the specified identification name does not exist, global data with the identification name specified in the <dataIdent> parameter is added, and the value specified in the <data> parameter is set in the added data.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a character string longer than 255 characters is specified in the <dataIdent> parameter, a "String too long" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

Sets 1 as the value in the global data with the identification name "ABC".

```
Rem Set 1 in the value of the global data "ABC".
SetGlobalData "ABC", 1

Rem Get the value (integer value) set in the global data "ABC", and store in the variable DATA&.
GetGlobalData "ABC", DATA&
```

Useable Modules

Unit Calculation Macro / Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 3.50 or later

Related Items

AddGlobalData (Reference: ▶ Details (p.405))

GetGlobalData (Reference: ▶ Details (p.530))

SetImageWindow

Sets the state of the image window.

Format

[Scene Control Macro / Communication Command Macro]
SetImageWindow <windowNo>, <locationX>, <locationY>, <width>, <height>, <unitNo>, <subNo>, <magnification>, <originX>, <originY>, <update>, <visible>

[Unit Macro]
SetImageWindow <magnification>, <originX>, <originY>

Parameter

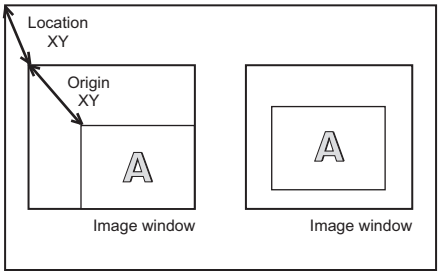
Parameter name	Data type	Description
<windowNo>	Integer type	Number of the image window whose state is to be set (0 to 23)
<locationX>	Integer type	Upper left X coordinate value of the image window
<locationY>	Integer type	Upper left Y coordinate value of the image window
<width>	Integer type	Width of the image window
<height>	Integer type	Height of the image window
<unitNo>	Integer type	Processing unit number of the target processing unit to display (-1 to (the number of registered processing units in the current scene minus one))
<subNo>	Integer type	Sub-image number of the target image to display (-1 to 100)
<magnification>	Double precision real number data type	Display magnification (-1, 0 to 16)
<originX>	Integer type	Upper left X coordinate of the image display relative to the upper left coordinate of the image window.
<originY>	Integer type	Upper left Y coordinate of the image display relative to the upper left coordinate of the image window
<update>	Integer type	Image mode 0: Every measurement (Image mode Freeze) 1: Only when an overall judgement result is NG at the time of measurement (Last NG image). 2: Only when a target processing unit is NG at the time of measurement. 3: Always updated (through display)
<visible>	Integer type	Setting of whether to display 0: Window invisible 1: Window visible

Return value

- None.

Description

Sets the state of the image window specified in the <windowNo> parameter. When this macro function is used with the unit macro, the state of the image window displayed using the MEASUREDISPI subroutine is set.
In the <locationX> parameter and <locationY> parameter, specify the relative coordinate values from the upper left coordinates of the gotten image container window to the upper left coordinates of the image window.
In the <width> parameter and <height> parameter, set the values of the image window width and height.
Specify the value of the displayed processing unit number in the <unitNo> parameter. To link the processing unit displayed in the image window to the flow display, specify -1.
Specify the value of the displayed sub image number in the <subNo> parameter. To display the contents of the image window as a position list, specify -1.
Specify the zoom of the image window in the <magnification> parameter. To set the zoom to auto, specify -1.
In the <originX> parameter and the <originY> parameter, specify the values of the relative coordinates from the upper left coordinates of the image window to the upper left coordinates of the displayed image.



In the <update> parameter, specify the value of the image mode of the image window.
In the <visible> parameter, specify the value of the display state of the image window.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
Even if a non-existent number, numerical value, or combination of data types or values is specified for the parameter, an error will not occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- [Scene Control Macro / Communication Command Macro]
  Execute this macro function when the BUSY signal or other measurement in progress signal is ON and measurement is prohibited. (Reference: ▶State Transitions and Execution Timing (p.227))
- [Unit Macro]
  This macro function can only be used in the *MEASUREDISPI subroutine or the *MEASUREDISPG subroutine. If used in another subroutine, an "Illegal function call" error will occur.

Example

In the communication command macro, changes the image mode of image windows 0 to 3 to Through. Sets the BusyOn flag to ON in advance in the communication command macro.

```
For I& = 0 To 3
  Rem Get the state of the image window.
  GetImageWindow I&, LOCATIONX&, LOCATIONY&, WIDTH&, HEIGHT&, UNITNO&, SUBNO&,
MAG#, ORIGINX&, ORIGINY&, UPDATE&, VISIBLE&

  Rem Change the update timing to Through.
  UPDATE& = 3

  Rem Set the state of the image window.
  SetImageWindow I&, LOCATIONX&, LOCATIONY&, WIDTH&, HEIGHT&, UNITNO&, SUBNO&,
MAG#, ORIGINX&, ORIGINY&, UPDATE&, VISIBLE&
Next
```

Useable Modules

Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 3.50 or later

Related Items

DisplayUnitNo (Reference: ▶ Details (p.464))
SetDisplayUnitNo (Reference: ▶ Details (p.699))
UnitNo (Reference: ▶ Details (p.790))
GetTextWindow (Reference: ▶ Details (p.550))
GetImageWindow (Reference: ▶ Details (p.533))
Ut (Reference: ▶ Details (p.793))

SetMeasureImage

Sets the measurement image of the processing unit.

Format

SetMeasureImage <measureImageNo>, <unitNo>, <imageNo>

Parameter

Parameter name	Data type	Description
<measureImageNo>	Integer type	Measurement image number to set to the target image of the measurement (always 0)
<unitNo>	Integer type	Processing unit number of the processing unit that holds the image to be set as the measurement image.
<imageNo>	Integer type	Image number of the image that is set to the measurement image

Return value

None.

Description

Sets the image of the image number specified in the <imageNo> parameter, which is held by the processing unit specified in the <unitNo> parameter, as the measurement image specified in the <measureImageNo> parameter. The measurement image is an image that can be used for measurement or filtering by a succeeding processing unit in the measurement flow.
Normally 0 should be specified in the <measureImageNo> parameter.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a non-existent number, numerical value, or combination of data types or values is specified for a parameter, an "Illegal function call" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- This macro function can only be used in the *MEASUREPROC subroutine. If used in any other subroutines, an error will occur and the function will not be executed.

Example

Changes the measurement image used by a succeeding processing unit to the camera change image of processing unit 4.

```
Rem Set camera change measurement image 0 of processing unit 4 as the measurement image.
SetMeasureImage 0, 4, 0
```

Useable Modules

Unit macro

Supported Versions

Version 3.50 or later

Related Items

CopyMeasureImage (Reference: ▶ Details (p.438))
CopyUnitImage (Reference: ▶ Details (p.446))
UnitNo (Reference: ▶ Details (p.790))
Ut (Reference: ▶ Details (p.793))

SetMeasureOut

Sets the external output setting for the measurement result.

Format

SetMeasureOut <mode>

Parameter

Parameter name	Data type	Description
<mode>	Integer type	External output setting 0: Not output externally 1: Output externally

Return value

None.

Description

Sets the "External Output" value in the layout settings to the external output setting value specified in the <mode> parameter. (Reference: ▶Setting the Behavior of Output Signals for Each Layout (Layout Settings) (p.90))

Even if 0 is set for the measurement result external output setting value, data is output if the SendData function or SendString function is used in the macro customize functions.

If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.

Even if a non-existent number, numerical value, or combination of data types is specified for the parameter, an "Illegal function call" error will not occur.

If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

If a value outside the range -2147483648 to 2147483647 is specified as an integer parameter, an "Overflow" error will occur.

Usage Cautions

- Execute this macro function when the BUSY signal or other measurement in progress signal is ON and measurement is prohibited. (Reference: ▶State Transitions and Execution Timing (p.227))
  Even when "External Output" is OFF, data output that uses a macro function is executed.

Example

Sets "External Output" to ON in the communication command macro.

```
SetMeasureOut 1
```

Useable Modules

Scene Control Macro / Communication Command Macro

Supported Versions

Version 3.50 or later

Related Items

GetMeasureOut (Reference: ▶ Details (p.536))
SendData (Reference: ▶ Details (p.696))
SendString (Reference: ▶ Details (p.698))

SetPlcData

Creates the data that is written with the WritePlcMemory function.

Format

SetPlcData <ioIdent>, <writeData()>, <offset>, <size>, <data>

Parameter

Parameter name	Data type	Description
<ioIdent>	Character string type	Identification name of the communication module to be used (Reference: ▶List of I/O Modules (p.341))
<writeData()>	Integer array	Data to write
<offset>	Integer type	Offset to address from which the beginning of the data is to be written (byte units).
<size>	Integer type	Data size to set (byte unit)
<data>	Integer type Double precision real number data type Character string type	Data to set

Return value

None.

Description

Sets the amount, specified in the <size> parameter, of the data specified in the <data> parameter by using the communication module specified in the <ioIdent> parameter. The data is set from the beginning of the data array specified in the <writeData()> parameter, in the position offset by the amount of the value specified in the <offset> parameter.
After creating data with this macro function, execute the WritePlcMemory function to write the data to the PLC memory area.
In the <writeData()> parameter, specify the 1D integer array variable that stores the data to be written, without adding element numbers but adding () to the variables.
In the <offset> parameter and <size> parameter, specify the offset and size in units of bytes. These units are different from the units used in the WritePlcMemory function (units of channels).
Specify 2, 4, or 8 in the <size> parameter. These respectively set a 2-byte integer, 4-byte integer, or 8-byte real number.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a non-existent number, numerical value, or combination of data types or values is specified for a parameter, an "Illegal function call" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- Before using the WritePlcMemory function in PLC link communication to write data to the PLC memory area, always use this macro function to create the data to be written. If the data is directly set in the WritePlcMemory parameter without using this macro function, the correct data may not be set.

Example

In the communication macro, writes measurement coordinate X and measurement coordinate Y of the search processing unit of Processing Unit number 2 to the PLC connected by PLC link. Measurement coordinate X is external data number 6, and measurement coordinate Y is external data number 7.

```
IOMODULE$ = "UdpPlcLink"

Rem Get the measurement result.
GetUnitData 2, 6, X#
GetUnitData 2, 7, Y#

Rem Convert the real number value multiplied by 1,000 to the integer value.
VALUE0& = Int(X# * 1000)
VALUE1& = Int(Y# * 1000)

Rem Get the settings of the output data area.
GetSystemData IOMODULE$, "outputArea", AREA&
GetSystemData IOMODULE$, "outputMemoryAddress", ADDRESS&

Rem Store the data to be written in an integer array variable.
Dim DATA&(1)
SetPlcData IOMODULE$, DATA&(), 0, 4, VALUE0&
SetPlcData IOMODULE$, DATA&(), 4, 4, VALUE1&

Rem Write the data (4ch) in data output area.
WritePlcMemory IOMODULE$, AREA&, ADDRESS&, 4, DATA&()
```

Useable Modules

Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 4.20 or later

Related Items

GetPlcData (Reference: ▶ Details (p.538))
ReadPlcMemory (Reference: ▶ Details (p.649))
WritePlcMemory (Reference: ▶ Details (p.807))

SetPollingState

Sets the execution status of the communication module.

Format

SetPollingState <ioIdent>, <state>

Parameter

Parameter name	Data type	Description
<ioIdent>	Character string type	Identification name of communication module whose polling state is to be set (Reference: ▶List of I/O Modules (p.341))
<state>	Integer type	Execution status of the communication module to set False: Stopped True: Operating.

Return value

None.

Description

Sets the polling state specified in the <state> parameter in the communication module specified in the <ioIdent> parameter.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a non-existent number, numerical value, or combination of data types or values is specified for a parameter, an "Illegal function call" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- After using this macro function to set the polling state of the communication module to the stopped state, always return the polling state to the running state. If the polling state of the communication module is left in the stopped state, the communication module will not be able to receive communication commands.

Example

Receives normal TCP communication data in the MEASUREPROC subroutine of the Unit Macro processing unit.

```
Rem Prepare a buffer that can receive 12 bytes of data.
Dim BUFFER&(11)
IOMODULE$ = "TcpNormal"

Rem Set the polling state of the communication module to stopped in order to receive the data.
SetPollingState IOMODULE$, False

Rem Executing the initialization of the reception data size.
SIZE& = 0

Repeat the reception process until the data has been received.
Try

  Do

    Rem Attempting the data reception.
    ReceiveData IOMODULE$, BUFFER&(), 12, SIZE&

    Rem Once the data has been received, display the data size in the system status console window.
    If(SIZE& > 0) Then
      Print "Received data size = " + Str$(SIZE&)
    Endif

  Loop While SIZE& = 0

  Rem Data has been received, so set the polling state of the communication module to running.
  SetPollingState IOMODULE$, True

Catch

  Rem Return the polling state of the stopped communication module to running.
  If GetPollingState(IOMODULE$) = False Then
    SetPollingState IOMODULE$, True
  Endif

End Try
```

Useable Modules

Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 4.20 or later

Related Items

GetPollingState (Reference: ▶ Details (p.540))
ReceiveData (Reference: ▶ Details (p.651))
SendData (Reference: ▶ Details (p.696))
SendString (Reference: ▶ Details (p.698))

SetSceneData

Sets data for the scene control macro.

Format

SetSceneData <dataIdent>, <data>

Parameter

Parameter name	Data type	Description
<dataIdent>	Character string type	Identification name of data to be set
<data>	Integer type Double precision real number data type Character string type	Data to set

Return value

None.

Description

Executes the process associated with the identification name specified in the <dataIdent> parameter.
In addition to the variable name used in the scene control macro program, the following identification names can be specified in the <dataIdent> parameter.

- "direct": Executes the scene control macro specified in the <data> parameter.
- "gosub": Executes the subroutine of the scene control macro specified in the <data>.

If an error occurs during execution of the process when "direct" or "gosub" is specified in the <dataIdent> parameter, the error will occur in the corresponding location as a scene control macro error, and then an "Illegal function call" error will occur in this macro function as an error of the module that called the macro function.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
Even if a non-existent number, numerical value, or combination of data types or values is specified for the parameter, an error will not occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- Execute this macro function when the BUSY signal or other measurement in progress signal is ON and measurement is prohibited. (Reference: ▶State Transitions and Execution Timing (p.227))

Example

Sets the communication command parameter received by the communication command macro in the variable of the scene control macro, and executes the subroutine that applies this value in the appropriate unit.

```
Rem Set the parameter received by the communication command in the variable of the scene control macro
SetSceneData "PARAM0&", ARGUMENTVALUE#(0)
SetSceneData "PARAM1&", ARGUMENTVALUE#(1)

Rem Execute the UPDATE_PARAM subroutine that has been defined in the scene control macro
SetSceneData "gosub", "*UPDATE_PARAM"
```

Useable Modules

Unit Calculation Macro / Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 5.20 or later

Related Items

GetSceneData (Reference: ▶ Details (p.544))

SetSceneDescription

Sets the scene description.

Format

SetSceneDescription <sceneNo>, <sceneDescription>

Parameter

Parameter name	Data type	Description
<sceneNo>	Integer type	Scene number (0 to 127) of the scene for which the description is to be set.
<sceneDescription>	Character string type	Scene description

Return value

None.

Description

Sets the description specified in the <sceneDescription> parameter in the description of the scene of the scene number specified in the <sceneNo> parameter.
The scene description can be set by executing this macro function, or in the scene maintenance screen. (Reference: ▶Editing Scenes (p.62))
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a non-existent number, numerical value, or combination of data types or values is specified for a parameter, an "Illegal function call" error will occur.
If a character string longer than 255 characters is specified for a character string parameter, a "String too long" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- Execute this macro function when the BUSY signal or other measurement in progress signal is ON and measurement is prohibited. (Reference: ▶State Transitions and Execution Timing (p.227))

Example

Gets the description of scene 1, and if a description is not set, sets the description.

```
Rem Get the scene description
DESCRIPTION$ = SceneDescription$(1)

If DESCRIPTION$ = "" Then
  Rem Set the scene description
  SetSceneDescription 1, "Description 1"
Endif
```

Useable Modules

Scene Control Macro / Communication Command Macro

Supported Versions

Version 3.50 or later

Related Items

SceneMaker$ (Reference: ▶ Details (p.689))
SceneDescription$ (Reference: ▶ Details (p.684))
SetSceneTitle (Reference: ▶ Details (p.721))
SceneTitle$ (Reference: ▶ Details (p.692))
SetSceneMaker (Reference: ▶ Details (p.719))

SetSceneGroupTitle

Sets the title of the scene group.

Format

SetSceneGroupTitle <sceneGroupNo>, <title>

Parameter

Parameter name	Data type	Description
<sceneGroupNo>	Integer type	Scene group number of the scene group whose title is to be set (-1 only)
<title>	Character string type	Title of scene group

Return value

None.

Description

Sets the title specified in the <title> parameter in the title of the scene group with the scene group number specified in the <sceneGroupNo> parameter.
If a title with 32 or more characters is specified in the <title> parameter, the first 31 characters are set in the title.
The title of the scene group can be set by executing this macro function, or in the scene maintenance screen. (Reference: ▶Editing Scenes (p.62))
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a non-existent number, numerical value, or combination of data types or values is specified for a parameter, an "Illegal function call" error will occur.
If a character string longer than 255 characters is specified for a character string parameter, a "String too long" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- Execute this macro function when the BUSY signal or other measurement in progress signal is ON and measurement is prohibited. (Reference: ▶State Transitions and Execution Timing (p.227))

Example

Sets the title of the current scene group.

```
SetSceneGroupTitle -1, "Title"
```

Useable Modules

Scene Control Macro / Communication Command Macro

Supported Versions

Version 3.50 or later

Related Items

SceneGroupTitle$ (Reference: ▶ Details (p.688))
SceneTitle$ (Reference: ▶ Details (p.692))
SetSceneTitle (Reference: ▶ Details (p.721))

SetSceneMaker

Sets the creator of the scene.

Format

SetSceneMaker <sceneNo>, <sceneMaker>

Parameter

Parameter name	Data type	Description
<sceneNo>	Integer type	Scene number (0 to 127) of the scene whose creator is to be set.
<sceneMaker>	Character string type	Creator of the scene

Return value

None.

Description

Sets the creator specified in the <sceneMaker> parameter in the creator of the scene that has the scene number specified in the <sceneNo> parameter.
When a creator name with 32 or more characters is specified in the <sceneMaker> parameter, the first 31 characters are set in the creator.
The scene creator can be set by executing this macro, or in the scene maintenance screen. (Reference: ▶Editing Scenes (p.62))
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a non-existent number, numerical value, or combination of data types or values is specified for a parameter, an "Illegal function call" error will occur.
If a character string longer than 255 characters is specified for a character string parameter, a "String too long" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- Execute this macro function when the BUSY signal or other measurement in progress signal is ON and measurement is prohibited. (Reference: ▶State Transitions and Execution Timing (p.227))

Example

Gets the creator of scene 3, and if not set, sets the creator.

```
Rem Get the creator of the scene.
NAME$ = SceneMaker$(3)

If NAME$ = "" Then
  Rem Set the creator of the scene.
  SetSceneMaker 3, "Maker"
Endif
```

Useable Modules

Scene Control Macro / Communication Command Macro

Supported Versions

Version 3.50 or later

Related Items

SceneDescription$ (Reference: ▶ Details (p.684))
SceneTitle$ (Reference: ▶ Details (p.692))
SetSceneTitle (Reference: ▶ Details (p.721))
SceneMaker$ (Reference: ▶ Details (p.689))
SetSceneDescription (Reference: ▶ Details (p.715))

SetSceneTitle

Sets the title of a scene.

Format

SetSceneTitle <sceneNo>, <title>

Parameter

Parameter name	Data type	Description
<sceneNo>	Integer type	Scene number (0 to 127) of the scene for which a title is to be set.
<title>	Character string type	Scene title

Return value

None.

Description

Sets the title specified in the <title> parameter in the title of the scene with the scene number specified in the <sceneNo> parameter.
If a title with 32 or more characters is specified in the <title> parameter, the first 31 characters are set in the title.
The scene title can be set by executing this macro function, or in the scene maintenance screen or flow edit screen. (Reference: ▶Editing Scenes (p.62)) (Reference: ▶Editing Processing Units in Scenes (p.53))
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a non-existent number, numerical value, or combination of data types or values is specified for a parameter, an "Illegal function call" error will occur.
If a character string longer than 255 characters is specified for a character string parameter, a "String too long" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- Execute this macro function when the BUSY signal or other measurement in progress signal is ON and measurement is prohibited. (Reference: ▶State Transitions and Execution Timing (p.227))

Example

Sets the title of scene 2

```
SetSceneTitle 2, "Title"
```

Useable Modules

Scene Control Macro / Communication Command Macro

Supported Versions

Version 3.50 or later

Related Items

SceneDescription$ (Reference: ▶ Details (p.684))
SceneMaker$ (Reference: ▶ Details (p.689))
SceneTitle$ (Reference: ▶ Details (p.692))
SetSceneDescription (Reference: ▶ Details (p.715))
SetSceneMaker (Reference: ▶ Details (p.719))

SetStop

Sets the conditions for stopping program execution.

Format

SetStop <string>

Parameter

Parameter name	Data type	Description
<string>	Character string type	Execution stop condition label

Return value

None.

Description

Set the character string specified in the <string> parameter as the stop condition for stopping program execution with the Stop function.
By specifying the execution stop condition character string set with the SetStop function as the parameter for the Stop function, you can stop program execution when the execution form is debug mode. Characters * (character string wildcard operator) and ? (single character wildcard operator) can be used as wildcards for the <string> specification.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a character string longer than 255 characters is specified for a character string parameter, a "String too long" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

Stops the program execution using the specified condition using the SetStop function in debug mode.

```
Rem Set the execution form to debug mode.
Debug 18
SetStop "AA?"
SetStop "B*"

Rem If character string "AAB" specified in the Stop function argument matches pattern "AA?", the program stops.
Stop "AAB"

Rem If character string "AABB" specified in the Stop function argument does not match pattern "AA?", the program does not stop.
Stop "AABB"

Rem If character string "BCDEF" specified in the Stop function argument matches pattern "B*", the program stops.
Stop "BCDEF"

Rem If character string "CDEF" specified in the Stop function argument does not match pattern "B*", the program does not stop.
Stop "CDEF"

Rem Set the execution form to release mode.
Debug 1
```

Useable Modules

Unit Calculation Macro / Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 5.20 or later

Related Items

Cont (Reference: ▶ Details (p.436))
Debug (Reference: ▶ Details (p.456))
DebugPrint (Reference: ▶ Details (p.458))
List (Reference: ▶ Details (p.597))
Print (Reference: ▶ Details (p.639))
SetVar (Reference: ▶ Details (p.741))
Stop (Reference: ▶ Details (p.747))
VarList (Reference: ▶ Details (p.796))

SetSystemData

Sets the system data.

Format

SetSystemData <dataIdent0>, <dataIdent1>, <data>

Parameter

Parameter name	Data type	Description
<dataIdent0>	Character string type	Data identification name of identification information 0 of system data to be set.
<dataIdent1>	Character string type	Data identification name of identification information 1 of system data to be set.
<data>	Integer type Double precision real number data type Character string type	Value of the system data to set

Return value

None.

Description

Sets the value specified in the <data> parameter in the system data of identification information 1 specified in the <dataIdent1> parameter, which belongs to identification information 0 specified in the <datdaIdent0> parameter.
For the identification information list, refer to the system data list. (Reference: ▶System Data List (p.327))
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If an identification name that does not exist is specified as the parameter, an "Illegal function call" error will occur.
If a character string longer than 255 characters is specified in the <dataIdent1> parameter, a "String too long" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

Sets the value "C:\temp\bmp" in the screen capture destination folder of identification information 1, "captureDirectory", which belongs to the measurement control settings of identification information 0, "Measure".

```
Rem Get the destination path to be set as the screen capture destination folder.
DIRNAME$ = "C:\temp\bmp"

Rem Set the screen capture destination folder that belongs to the measurement control settings.
SetSystemData "Measure", "captureDirectory", DIRNAME$
```

Useable Modules

Unit Calculation Macro / Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 3.50 or later

Related Items

AddSystemData (Reference: ▶ Details (p.407))
GetSystemData (Reference: ▶ Details (p.546))
SetGlobalData (Reference: ▶ Details (p.702))
SetUnitData (Reference: ▶ Details (p.731))

SetTextStyle

Set the draw attributes of the character string.

Format

SetTextStyle <fontSize>, <align>, <color>, <angle>, <style>

Parameter

Parameter name	Data type	Description
<fontSize>	Integer type	Font size of the drawn character string
<align>	Integer type	Alignment of the drawn character string TA_BASELINE: Align baseline TA_BOTTOM: Align lower end TA_TOP: Align top TA_CENTER: Align horizontal center TA_LEFT: Align left TA_RIGHT: Align right TA_NOUPDATECP: Current position not update TA_RTLRENDING: Right to left TA_UPDATECP Current position update
<color>	Integer type	Color value of the character string color to be drawn JUDGE_NC: Unmeasured color (Grey) JUDGE_OK: OK judgement color (Green) JUDGE_NG: NG judgement color (Red) RGB Function: Any color
<angle>	Integer type	Rotation angle (0 to 359) of the drawn character string
<style>	Integer type	Font style of the drawn character string FONTSTYLE_NORMAL: Normal FONTSTYLE_BOLD: Bold FONTSTYLE_ITALIC: Italic FONTSTYLE_UNDERLINE: Underline FONTSTYLE_STRIKEOUT: Strike-through

Return value

None.

Description

Sets the following parameters as the drawing attributes: specified font size by the <fontSize> parameter, the specified string alignment by the <align> parameter, the specified string color by the <color> parameter, the specified string rotation angle by the <angle> parameter, and the specified string style by the <style> parameter. Before executing the DrawTextG image screen window control function, execute this macro function to draw the graphic figure using the set drawing attribute. Use the SetDRAWStyle function to set the drawing attribute used for macro functions that draw graphic figures. (Reference: ▶SetDrawStyle (p.700))
Disjunctive specification of TA_BOTTOM, TA_TOP, TA_LEFT, TA_CENTER, and TA_RIGHT in the <align> parameter is possible.
The gotten color value by the RGB function can be set for the <color> parameter. (Reference: ▶RGB (p.662))
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a value outside the range -2147483648 to 2147483647 is specified as an integer parameter, an "Overflow" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- This macro function can only be used in the *MEASUREDISPI subroutine or the *MEASUREDISPG subroutine. If used in another subroutine, an "Illegal function call" error will occur.

Example

Uses the *MEASUREDISPG subroutine of the Unit Macro processing unit to display a character string with its font size of 20, aligned to bottom and horizontally centered, colored by the "OK" judgement color, and inclined by 90 degrees.

```
*MEASUREDISPG

  Rem Set the draw attibutes
  SetTextStyle 20, TA_BOTTOM OR TA_CENTER, JUDGE_OK, 90, FONTSTYLE_ITALIC

  Rem Draw the image
  DrawTextG "Measurement OK", 100, 100, 0, UnitNo

Return
```

Useable Modules

Unit macro

Supported Versions

Version 3.50 or later

Related Items

DrawTextG (Reference: ▶ Details (p.502))
SetDrawStyle (Reference: ▶ Details (p.700))
RGB (Reference: ▶ Details (p.662))

SetTextWindow

Sets the state of the text window.

Format

SetTextWindow <unitNo>, <subNo>, <update>, <visible>

Parameter

Parameter name	Data type	Description
<unitNo>	Integer type	Processing unit number of the target processing unit to display
<subNo>	Integer type	Sub number of the target image to display
<update>	Integer type	Updated timing (always 0)
<visible>	Integer type	Setting of whether to display (This parameter is always invalid i.e., the entered value is always ignored).

Return value

None.

Description

Sets the state of the text window.
Specify the value of the displayed processing unit number in the <unitNo> parameter. To link the processing unit displayed in the text window to the flow display, specify -1.
In the <subNo> parameter, 0 should normally be specified for the value of the displayed sub image number. When used with the unit macro, the text window state of a sub image number other than 0 can be set by including the DisplaySubNo macro function in the MEASUREDISPT subroutine.
In the <update> parameter, 0 should always be specified for the update timing.
The value specified in the <visible> parameter is not reflected to the setting.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- Execute this macro function when the BUSY signal or other measurement in progress signal is ON and measurement is prohibited. (Reference: ▶State Transitions and Execution Timing (p.227))

Example

In the communication command macro, changes the processing unit number of the processing unit displayed in the text display window to the number specified in the communication command argument.

```
Rem Get the state of the text window.
GetTextWindow UNITNO&, SUBNO&, UPDATE&, VISIBLE&

Rem Set the number specified in the command argument in the processing unit number that is displayed.
SetTextWindow argumentValue#(0), SUBNO&, UPDATE&, VISIBLE&
```

Useable Modules

Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 3.50 or later

Related Items

DisplaySubNo (Reference: ▶ Details (p.462))
GetTextWindow (Reference: ▶ Details (p.550))
SetDisplayUnitNo (Reference: ▶ Details (p.699))
UnitNo (Reference: ▶ Details (p.790))
DisplayUnitNo (Reference: ▶ Details (p.464))
RefreshTextWindow (Reference: ▶ Details (p.655))
SetImageWindow (Reference: ▶ Details (p.704))
Ut (Reference: ▶ Details (p.793))

SetUnitData

Sets the data of a processing unit.

Format

SetUnitData <unitNo>, <dataNo>, <data>
SetUnitData <unitNo>, <dataIdent>, <data>

Parameter

Parameter name	Data type	Description
<unitNo>	Integer type	Processing unit number (0 to (the number of registered processing units in the current scene minus one))
<dataNo>	Integer type	External reference data of the processing unit data to set (reference: ▶ Vision System FH/FZ5 Series Processing Item Function Reference Manual (Cat. No. Z341)"
<dataIdent>	Character string type	Data identification name of processing unit data to be set.
<data>	Integer type Double precision real number data type Character string type	Processing unit data to set

Return value

None.

Description

Sets the data specified in the <data> parameter in the data of the external reference data number specified in the <dataNo> parameter, held by the processing unit specified in the <unitNo> parameter.
The data can also be gotten by specifying the <dataIdent> parameter instead of the <dataNo> parameter.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a non-existent number, numerical value, or combination of data types or values is specified for a parameter, an "Illegal function call" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- Execute this macro function when the BUSY signal or other measurement in progress signal is ON and measurement is prohibited. (Reference: ▶State Transitions and Execution Timing (p.227))
- Use this macro function with the measurement image displayed after one or more measurements, or after the image file is specified and re-measured.

Example

Sets "Reflect to overall judgement" of the search processing unit of Processing Unit number 2 to "OFF". "Reflect to overall judgement" is external reference data number 103 and external reference data identification name "overallJudge".

```
SetUnitData 2, 103, 1

Rem The same result will be gotten if "overallJudge" is specified instead of 103.
SetUnitData 2, "overallJudge", 1
```

Useable Modules

Unit Calculation Macro / Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 3.50 or later

Related Items

GetUnitData (Reference: ▶ Details (p.552))
MeasureStart (Reference: ▶ Details (p.617))
UnitNo (Reference: ▶ Details (p.790))
GetUnitFigure (Reference: ▶ Details (p.554))
MeasureStop (Reference: ▶ Details (p.619))
Ut (Reference: ▶ Details (p.793))

SetUnitFigure

Sets the figure data of the processing unit.

Format

SetUnitFigure <unitNo>, <figureNo>, <figure()>

Parameter

Parameter name	Data type	Description
<unitNo>	Integer type	Processing unit number (0 to (the number of registered processing units in the current scene minus one))
<figureNo>	Integer type	Figure number to be set (Reference: ▶List of Figure Numbers (p.366))
<figure()>	Integer array	Figure data to be set (Reference: ▶Figure Data List (p.364))

Return value

None.

Description

Sets the figure data specified in the <figure()> parameter in the figure specified in the <figureNo> parameter of the processing unit specified in the <unitNo> parameter.
In the <figure()> parameter, specify the 1D integer array variable that will hold the figure data by adding only () without specifying an element number.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a non-existent number, numerical value, or combination of data types or values is specified for a parameter, an "Illegal function call" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- Execute this macro function when the BUSY signal or other measurement in progress signal is ON and measurement is prohibited. (Reference: ▶State Transitions and Execution Timing (p.227))
- Use this macro function with the measurement image displayed after one or more measurements, or after the image file is specified and re-measured.
- Set the figure data so that pixels from outside the image are not included in the figure.

Example

Changes the position of the region figure (rectangle) of the search processing unit of Processing Unit number 2.

```
Dim FIGURE&(5)

Rem Get the figure data of the processing unit.
GetUnitFigure 2, 1, FIGURE&()

Rem Based on the gotten figure data, change the values of the upper left point XY coordinates and lower right point XY coordinates of the region figure.
FIGURE&(2) = 100
FIGURE&(3) = 100
FIGURE&(4) = 300
FIGURE&(5) = 300

Rem Set the figure data in which the position of the region figure has been changed in the processing unit.
SetUnitFigure 2, 1, FIGURE&()
```

Useable Modules

Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 3.50 or later

Related Items

CopyUnitFigure (Reference: ▶ Details (p.444))
GetUnitData (Reference: ▶ Details (p.552))
GetUnitFigure (Reference: ▶ Details (p.554))
MeasureStart (Reference: ▶ Details (p.617))
MeasureStop (Reference: ▶ Details (p.619))
SetUnitData (Reference: ▶ Details (p.731))
UnitNo (Reference: ▶ Details (p.790))
Ut (Reference: ▶ Details (p.793))

SetUnitJudge

Sets the judgement result of a processing unit.

Format

SetUnitJudge <unitNo>, <judge>[, <totalJudgeRefrect>]

Parameter

Parameter name	Data type	Description
<unitNo>	Integer type	Processing unit number (0 to (the number of registered processing units in the current scene minus one))
<judge>	Integer type	Judgement result to set JUDGE_NC: No judgement (unmeasured) JUDGE_OK: Judgement result OK JUDGE_NG: Judgement result NG JUDGE_IMAGEERROR: Judgement result error (image format mismatch) JUDGE_MODELERROR: Judgement result error (unregistered model) JUDGE_MEMORYERROR: Judgement result error (insufficient memory) JUDGE_ERROR: Judgement result error (other error)
<totalJudgeRefrect>	Integer type	Setting of whether to reflect the overall judgement result False: not reflected True: reflected

Return value

None.

Description

Sets the judgement result specified in the <judge> parameter in the judgement result of the processing unit specified in the <unitNo> parameter. If the <totalJudgeRefrect> parameter is omitted, the specified judgement is applied to the overall judgement.

When this macro function is used to set the judgement result of another processing unit, operation is as follows:

- If measurement processing is executed on the processing unit after the judgement result for the first measurement is set with this macro function, the judgement result set with this macro function is overwritten with the measurement processing result of the processing unit.
- If the judgement result is set with this macro function after measurement processing is executed on the processing unit for the first measurement, the judgement result of the processing unit is overwritten by the judgement result set with this macro function.

In both cases, the judgement result can be reflected in the overall judgement prior to overwriting by specifying True with the <totakJudgeRefrect> parameter.

If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.

If a non-existent number, numerical value, or combination of data types or values is specified for a parameter, an "Illegal function call" error will occur.

If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

Sets the judgement result of the search processing unit of Processing Unit number 2 in Judgement Result OK.

```
SetUnitJudge 2, JUDGE_OK, True
```

Useable Modules

Unit Calculation Macro / Unit Macro

Supported Versions

Version 3.50 or later

Related Items

SetUnitData (Reference: ▶ Details (p.731))
UnitJudge (Reference: ▶ Details (p.788))
Ut (Reference: ▶ Details (p.793))
TotalJudge (Reference: ▶ Details (p.761))
UnitNo (Reference: ▶ Details (p.790))

SetUnitTitle

Sets the title of a processing unit.

Format

SetUnitTitle <unitNo>, <title>

Parameter

Parameter name	Data type	Description
<unitNo>	Integer type	Processing unit number (0 to (the number of registered processing units in the current scene minus one))
<title>	Character string type	Title of processing unit to be set (31 characters max.)

Return value

None.

Description

Sets the title specified in the <title> parameter in the title name of the processing unit specified in the <unitNo> parameter. If a title with 32 or more characters is specified in the <title> parameter, the first 31 characters are set in the title.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a non-existent number, numerical value, or combination of data types or values is specified for a parameter, an "Illegal function call" error will occur.
If a character string longer than 255 characters is specified for a character string parameter, a "String too long" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- Execute this macro function when the BUSY signal or other measurement in progress signal is ON and measurement is prohibited. (Reference: ▶State Transitions and Execution Timing (p.227))

Example

Sets "bolt search" as the search name in the title of the search processing unit of Processing Unit number 2.

```
SetUnitTitle 2, "bolt search"
```

Useable Modules

Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 3.50 or later

Related Items

MeasureStart (Reference: ▶ Details (p.617))
UnitNo (Reference: ▶ Details (p.790))
Ut (Reference: ▶ Details (p.793))
MeasureStop (Reference: ▶ Details (p.619))
UnitTitle$ (Reference: ▶ Details (p.791))

SetUserSubroutine

Register a user-defined function that has been defined in the external DDL file.

Format

SetUserSubroutine <subroutineIdent>, <dllFileName>, <functionName>

Parameter

Parameter name	Data type	Description
<subroutineIdent>	Character string type	Identification name to the user-defined function to be registered
<dllFileName>	Character string type	DLL file name
<functionName>	Character string type	User-defined function name to be registered

Return value

None.

Description

Registers the function that is defined in the specified DLL file by the <dllFileName> and has a specified function name by the <functionName> parameter as the specified identification name by the <subroutineIdent> parameter. If registered by this macro function, the user-defined functions can be called using the Call function and specify the identification names.
In the <dllFileName> parameter, specify the file name without the file extension ".dll".
Normally use the SetUserSubroutine statement in the *MCRINIT subroutine and execute the subroutine to perform this macro function, or execute this macro function before executing the Call function.
User-defined functions can only be registered by executing this the SetUserSubroutine function using the macro customize function. Execute this command on all processing units, all scene control macros in all scene, and all communications commands in all communications command macros that call and execute user-defined functions.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a non-existent number, numerical value, or combination of data types or values is specified for a parameter, an "Illegal function call" error will occur.
If a character string longer than 255 characters is specified in the <subroutineIdent> parameter or the <functionName> parameter, a "String too long" error will occur.
If a character string of a file name (including the absolute path name) longer than 255 characters is specified in the <dllFileName> parameter, a "String too long" error will occur.
If a registered user-defined function by this macro function has not been programmed with the supported interfaces, an error will not occur. In this case, an error will occur at the execution of the user-defined function processing with this Call function.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- Only the user-defined functions that have been defined in programmed DLL files by the supported interfaces are accepted to This macro function only accepts the user-defined function that have been defined in programmed DLL files by the supported interfaces. Also, the DLL files must be saved in the same directory as the FZ-CoreRA.exe For user-defined functions creation, use the FH-AP1.

Example

With identification name "USR", registers a user-defined function "UserProc0" that has been defined in MacroUserProc.dll. Then, specifies the identification name to call the user-defined function and execute it.

```
Rem Register the user-defined function so that the function can be used in this program
SetUserSubroutine "USR","MacroUserProc","UserProc0"

Rem Call the registered user-defined function and execute it
Call "USR", 0
```

Useable Modules

Unit Calculation Macro / Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 5.20 or later

Related Items

Call (Reference: ▶ Details (p.421))

SetVar

Sets all variables with the specified variable names.

Format

SetVar <variableName>, <value>

Parameter

Parameter name	Data type	Description
<variableName>	Character string type	Name of variable to be set
<value>	Integer type Double precision real number data type Character string type	Value to be set

Return value

None.

Description

This sets the value specified in the <value> parameter in the variable specified in the <variableName> parameter. Set a value of the same data type in the <value> parameter as the variable specified in the <variableName> parameter.
Characters * (character string wildcard operator) and ? (single character wildcard operator) can be used as wildcards for the <variableName> specification. When using a wildcard to specify multiple variables in the <variableName> parameter, make sure that the specified variables are the same data type.
Wildcards can be used to specify file names in the following manner.

*	Specify all variables.
???	Specify variables with a 3-character variable name.
A*	Specify variables with a variable name that starts with "A".
*A*A*A*	Specify variables with a variable name that includes at least three "A" letters.
????*	Specify variables with a variable name that consists of four or more characters.

If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
Even if a non-existent number, numerical value, or combination of data types or values is specified for the parameter, an error will not occur.
If a character string longer than 255 characters is specified in the <variableName> parameter, a "String too long" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- Execute this macro function when the BUSY signal or other measurement in progress signal is ON and measurement is prohibited. (Reference: ▶State Transitions and Execution Timing (p.227))
- Use this macro function with the measurement image displayed after one or more measurements, or after the image file is specified and re-measured.

Example

Among the variables registered as reference variables of the Unit Macro processing unit of Processing Unit number 1, i.e., A01@, A02@, AA01@, AB01@, B01@, and B02@, sets the values of all reference variables that start with "A", i.e., A01@, A02@, AA01@, and AB01@.

```
A01@ = 100
A02@ = 100
AA01@ = 100
AB01@ = 100
B01@ = 100
B02@ = 100

Rem Set "123" only to the variables whose name start with "A".
SetVar "A*@", 123
```

The result is shown below.

```
A01@ = 123
A02@ = 123
AA01@ = 123
AB01@ = 123
B01@ = 100
B02@ = 100
```

Useable Modules

Unit Calculation Macro / Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 5.20 or later

Related Items

Cont (Reference: ▶ Details (p.436))
DebugPrint (Reference: ▶ Details (p.458))
Print (Reference: ▶ Details (p.639))
Stop (Reference: ▶ Details (p.747))
Debug (Reference: ▶ Details (p.456))
List (Reference: ▶ Details (p.597))
SetStop (Reference: ▶ Details (p.723))
VarList (Reference: ▶ Details (p.796))

Sin

Gets the sine of the specified expression.

Format

Sin(<expression>)

Parameter

Parameter name	Data type	Description
<expression>	Integer type Double precision real number data type	Expression to get the sine

Return value

Returns the sine as a double precision real number value in the range -1 to 1.

Description

Gets the sine of the expression specified in the <expression> parameter.
In the <expression> parameter, specify the value in radians. To convert an angle value to a radian value, multiply by π/180.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a value is assigned to the return value variable or the variable is not used in an expression, a "Syntax error" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

Gets the sine of 30°.

```
DATA# = Sin(30/180*3.141592)
```

The result is shown below.

```
DATA# = 0.5
```

Useable Modules

Unit Calculation Macro / Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 3.50 or later

Related Items

Atn (Reference: ▶ Details (p.417))
GetUnitData (Reference: ▶ Details (p.552))
Tan (Reference: ▶ Details (p.755))
Cos (Reference: ▶ Details (p.450))
UnitData (Reference: ▶ Details (p.778))

Sqr

Determining the square root.

Format

Sqr(<expression>)

Parameter

Parameter name	Data type	Description
<expression>	Integer type Double precision real number data type	Expression to get the square root

Return value

Returns the double precision real square root value.

Description

Gets the square root of the expression specified in the <expression> parameter.
Specify 0 or positive number for the <expression> parameter.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a non-existent number, numerical value, or combination of data types or values is specified for a parameter, an "Illegal function call" error will occur.
If a value is assigned to the return value variable or the variable is not used in an expression, a "Syntax error" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

Gets the square root of 256.

```
DATA# = Sqr(256)
```

The result is shown below.

```
DATA# = 16
```

Useable Modules

Unit Calculation Macro / Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 3.50 or later

Related Items

GetUnitData (Reference: ▶ Details (p.552))
UnitData (Reference: ▶ Details (p.778))

StartTimer

Starts the elapsed time measurement.

Format

StartTimer

Parameter

None.

Return value

Returns the elapsed time as a double precision type real number value.

Description

Starts the elapsed time measurement.
After starting measurment of elapsed time with this macro function, get the elapsed time by executing the Timer function.
Execution of the ElapsedTime function is valid only for the processing units such as the unit calculation macro and the Unit Macro processing units. In contrast, executions of this macro function and the Timer function are valid for all macro customize functions.
If a value is assigned to the return value variable or the variable is not used in an expression, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

Measures the elapsed time since the beginning of the StartTimer function until the execution of the Timer function.

```
T# = StartTimer

Rem Executes the process whose the process execution elapsed time is measured.


Rem Gets the elapsed time using the return value of the StartTimer function.
TIME& = Timer(T#, 0)
```

Useable Modules

Unit Calculation Macro / Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 3.50 or later

Related Items

ElapsedTime (Reference: ▶ Details (p.508))
Timer (Reference: ▶ Details (p.759))
Wait (Reference: ▶ Details (p.806))

Stop

Stops program execution.

Format

Stop [<string>]

Parameter

Parameter name	Data type	Description
<string>	Character string type	Execution stop condition label This parameter can be omitted. Without If the parameter is not specified, program execution stops at the point that this function is executed.

Return value

None.

Description

When the character string specified in the <string> parameter matches the character string set as the execution stop condition by executing the SetStop function, program execution stops.
To resume the stopped program, use the Cont function in the system status console window.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a character string longer than 255 characters is specified for a character string parameter, a "String too long" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- This macro function is only enabled when specified in debug mode with the Debug function. Specifying other values than the range above will treat the statement with this function in the same manner with the Rem function (i.e., ignores the statement). (Reference: ▶How to Use the Debug Function (p.235))
- After the data output to the system status console window, the window is displayed on top of the Sensor Controller main screen. To display the system status console window on top of the main screen, click [_] on the upper-right of the system status console window or press [Alt] + [Tab] on the connected USB keyboard to the sensor controller.

Example

Stops the program execution using the specified condition using the SetStop function in debug mode.

```
Rem Set the execution form to debug mode.
Debug 18
SetStop "ABC"

Rem If character string "ABC" specified in the Stop function argument matches pattern "ABC", the program stops.
Stop "ABC"

Rem If character string "ABCD" specified in the Stop function argument does not match pattern "ABC", the program does not stop.
Stop "ABCD"

Rem If the parameter is not specified in the Stop function, program execution stops at the point that the Stop function is executed.
Stop

Rem Set the execution form to release mode.
Debug 1
```

Useable Modules

Unit Calculation Macro / Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 5.20 or later

Related Items

Cont (Reference: ▶ Details (p.436))
DebugPrint (Reference: ▶ Details (p.458))
Print (Reference: ▶ Details (p.639))
SetVar (Reference: ▶ Details (p.741))
Debug (Reference: ▶ Details (p.456))
List (Reference: ▶ Details (p.597))
SetStop (Reference: ▶ Details (p.723))
VarList (Reference: ▶ Details (p.796))

Str$

Converts a numeric value in the numeric character string.

Format

Str$(<expression>)

Parameter

Parameter name	Data type	Description
<expression>	Integer type Double precision real number data type	Expression converted to a numerical character string

Return value

Returns the character string type numeric value.

Description

Converts the specified expression in the <expression> parameter to the numeric character string.
Str$ is the inverse function of Val. Val converts the specified numeric character string in numeric value.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a value outside the range -1.0E30 to 1.0E30 is specified for a double precision real number parameter, an "Overflow" error might occur.
If a value is assigned to the return value variable or the variable is not used in an expression, a "Syntax error" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

Outputs the gotten judgement result and measured correlation value with the search processing unit (Processing Unit number 5) to the system status console window using the MEASUREPROC subroutine in the Unit Macro processing unit. The judgement result and the correlation value can be gotten with External Reference Data numbers 0 and 5 respectively.

```
*MEASUREPROC

  Rem Get the measurement result of the processing unit.
  GetUnitData 5, 0, JG&
  GetUnitData 5, 5, CR#

  Rem Convert the gotten measurement result to the numeric character string and output the character
  string to the system status console window using the Print function.
  Print Str$(JG&) +"," + Str$(CR#)

Return
```

The result is shown below.

1,98.4

Useable Modules

Unit Calculation Macro / Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 3.50 or later

Related Items

Asc (Reference: ▶ Details (p.413))
Chr$ (Reference: ▶ Details (p.427))
GetUnitData (Reference: ▶ Details (p.552))
Hex$ (Reference: ▶ Details (p.560))
LCase$ (Reference: ▶ Details (p.589))
Left$ (Reference: ▶ Details (p.591))
Len (Reference: ▶ Details (p.593))
Mid$ (Reference: ▶ Details (p.621))
Piece$ (Reference: ▶ Details (p.637))
Print (Reference: ▶ Details (p.639))
Right$ (Reference: ▶ Details (p.664))
Str2$ (Reference: ▶ Details (p.751))
UCase$ (Reference: ▶ Details (p.775))
Val (Reference: ▶ Details (p.794))

Str2$

Converts a value to a numeric character string in the specified formats.

Format

Str2$(<expression>, <integral>, <fixed>, <zeroSuppression>, <negative>)

Parameter

Parameter name	Data type	Description
<expression>	Integer type Double precision real number data type	Expression converted to a numerical character string
<integral>	Integer type	Number of digits in integer part (0 to 8)
<fixed>	Integer type	Number of digits in decimal part (0 to 5)
<zeroSuppression>	Integer type	Inserting character to the unused integer places 0: Fill in with spaces 1: Fill in with zeros
<negative>	Integer type	Negative number sign 0: - 1: 8

Return value

Returns the character string type numeric value.

Description

Converts the expression specified in the <expression> parameter to the numeric character string after the following processes:

- Adjust number of integer places to the <integral> parameter and number of decimal places to the <fixed> parameter,
- Insert the character specified in the <zeroSuppression> parameter to the unused places, and
- Replace the negative sign with the sign specified in the <negative> parameter.

Str$ is the inverse function of Val. There is no inverse function for Str2$. Val converts the specified numeric character string in numeric value.

If 0 is specified in the <integral> parameter, all digits in the integer part in the <expression> parameter are converted to a numeric character string.

If a smaller number than number of integer places in the <expression> parameter is specified in the <integral> parameter, a maximum positive number (or a minimum negative number) that can be expressed with the specified number of integer digits in the <expression> parameter is returned as a numeral character string.

Example: Specify the following parameters: Number of digits in the integer part: 2, number of digits in the decimal part: 0

When <expression> parameter is 179.099, the changed numerical character string is "99"

If a larger value than the number of integer places in the <expression> parameter is specified in the <integral> parameter, either zeros or spaces (depending on the <zeroSuppression> parameter) are filled to the higher digit places of the converted numeral character string until the digit place number reaches to the <integral> parameter.

Note that the negative number uses one extra digit in the integer part for the negative number sign (specified

in the <negative> parameter).

Example: Specify the following parameters: Number of digits in the integer part: 3, number of digits in the decimal part: 3

When <expression> parameter is 999.999, the changed numerical character string is "999.999"

When <expression> parameter is -999.999, the changed numerical character string is "-99.999"

If 0 is specified in the <fixed> parameter, the expression in the <expression> parameter is rounded off to the nearest whole number and converted to the numeric character string.

If a smaller number than number of decimal places in the <expression> parameter is specified in the <fixed> parameter, the expression in the <expression> parameter is rounded off to the nearest number that can be expressed with a <fixed> number and converted to the numeral character string.

Example: Specify the following parameters: Number of digits in the integer part: 2, number of digits in the decimal part: 4

When <expression> parameter is 10.12345, the changed numerical character string is "10.1235"

If a larger number than number of decimal places in the <expression> parameter is specified in the <fixed> parameter, zeros are filled to the lower digit places of the converted numeral character string until the digit place number reaches to the <fixed> parameter.

Example: Specify the following parameters: Number of digits in the integer part: 2, number of digits in the decimal part: 5

When <expression> parameter is 10.123, the changed numerical character string is "10.12300"

If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.

If a non-existent number, numerical value, or combination of data types or values is specified for a parameter, an "Illegal function call" error will occur.

If a value outside the range -2147483648 to 2147483647 is specified as an integer parameter, an "Overflow" error will occur.

If a value outside the range -1.0E30 to 1.0E30 is specified for a double precision real number parameter, an "Overflow" error might occur.

If a value is assigned to the return value variable or the variable is not used in an expression, a "Syntax error" error will occur.

If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

Outputs the measured correlation value, measured position coordinates X and Y (Position X and Position Y) with the search processing unit (Processing Unit number 5) to the system status console window. The correlation value, measured position coordinates X and Y can be gotten with External Reference Data numbers 5, 6, and 7 respectively.

```
Rem Get the measurement result of the processing unit.
GetUnitData 5, 6, X#
GetUnitData 5, 7, Y#
GetUnitData 5, 5, CR#

Rem Convert the measurement results to the numeric character strings in the specified format.
RESX$ = Str2$(X#, 3, 3, 0, 0)
RESY$ = Str2$(Y#, 3, 3, 0, 0)
RESCR$ = Str2$(CR#, 3, 0, 0, 0)

Rem Use the Print function to output the strings to the value to the system status console window.
Print RESX$ + "," + RESY$ + "," + RESCR$
```

The result is shown below.

```
150.000,359.000, 97
```

Useable Modules

Unit Calculation Macro / Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 3.50 or later

Related Items

Asc (Reference: ▶ Details (p.413))
Chr$ (Reference: ▶ Details (p.427))
Hex$ (Reference: ▶ Details (p.560))
LCase$ (Reference: ▶ Details (p.589))
Left$ (Reference: ▶ Details (p.591))
Len (Reference: ▶ Details (p.593))
Mid$ (Reference: ▶ Details (p.621))
Piece$ (Reference: ▶ Details (p.637))
Print (Reference: ▶ Details (p.639))
Right$ (Reference: ▶ Details (p.664))
Str$ (Reference: ▶ Details (p.749))
UCase$ (Reference: ▶ Details (p.775))
Val (Reference: ▶ Details (p.794))

SystemReset

Reboots the Sensor Controller.

Format

SystemReset

Parameter

None.

Return value

None.

Description

Reboots the Sensor Controller.
If this command is executed on the Sysmac Studio FH Tools, no action is taken and the process ends.

Usage Cautions

- None.

Example

After loading the system data, executes "Save data". To apply the loaded settings in the sensor controller, restart the sensor controller.

```
Rem Load the file of the system data
LoadSystemData "C:\BACKDIR\backupsysset.ini"

Rem Save to the controller.
SaveData

Rem Reboot the Sensor Controller.
SystemReset
```

Useable Modules

Scene Control Macro / Communication Command Macro

Supported Versions

Version 3.50 or later

Related Items

Date$ (Reference: ▶ Details (p.454))
Left$ (Reference: ▶ Details (p.591))
SaveData (Reference: ▶ Details (p.670))
ExitFzProcess (Reference: ▶ Details (p.519))
LoadSystemData (Reference: ▶ Details (p.605))

Tan

Gets the tangent of the specified expression.

Format

Tan(<expression>)

Parameter

Parameter name	Data type	Description
<expression>	Integer type Double precision real number data type	Expression to get the tangent

Return value

Returns the double precision real tangent value.

Description

Gets the tangent of the expression specified in the <expression> parameter.
In the <expression> parameter, specify the value in radians. To convert an angle value to a radian value, multiply by π/180.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a value is assigned to the return value variable or the variable is not used in an expression, a "Syntax error" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

Gets the tangent of 45°.

```
DATA# = Tan(45/180*3.141592)
```

The result is shown below.

```
DATA# = 0.999999673205
```

Useable Modules

Unit Calculation Macro / Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 3.50 or later

Related Items

Atn (Reference: ▶ Details (p.417))
GetUnitData (Reference: ▶ Details (p.552))
UnitData (Reference: ▶ Details (p.778))
Cos (Reference: ▶ Details (p.450))
Sin (Reference: ▶ Details (p.743))

Time$

Reads out the clock time from the internal clock.

Format

Time$

Parameter

None.

Return value

Returns the character string type time value.
The time value is a character string of the internal clock time whose hour (HH), minute (MM), and second (SS) separated by a colon (:). The ranges of the hour, minute, and second values are as follows.

Hour (HH) : 00 to 23
Minute (MM) : 00 to 59
Second (SS) : 00 to 59

Description

Reads the time from the internal clock and returns the time value (HH, MM, SS) in character string format.
The internal clock can be adjusted in [Date-time Settings] under [System settings]. (Reference: ▶Setting the Date and Time: [Date-time Setting] (p.291))
If a value is assigned to the return value variable or the variable is not used in an expression, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

Outputs to the read internal clock value to the system status console window.

```
Dim NOW$(2)

Rem Read out the clock time from the internal clock.
NOW$ = Time$

Rem Change the format of the read out clock time.
For I&=0 To 2
  NOW$(I&) = Piece$(NOW$, ":", I&+1, I&+1)
Next

Rem Output to the system status console window the time which the format have been changed previously.
Print NOW$(0) + ":" + NOW$(1) + ":" + NOW$(2)
```

The result is shown below.

```
01:23:45
```

Useable Modules

Unit Calculation Macro / Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 3.50 or later

Related Items

Date$ (Reference: ▶ Details (p.454))
Mid$ (Reference: ▶ Details (p.621))
Print (Reference: ▶ Details (p.639))
GetSystemData (Reference: ▶ Details (p.546))
Piece$ (Reference: ▶ Details (p.637))
SetSystemData (Reference: ▶ Details (p.725))

Timer

Gets the elapsed time.

Format

StartTimer(<start>, <mode>)

Parameter

Parameter name	Data type	Description
<start>	Double precision real number data type	Rem Return value of the StartTimer function that started the measurement of the elapsed time.
<mode>	Integer type	Unit of the elapse time to get 0: ms unit 1: μ unit

Return value

Returns the elapsed time after the execution of the StartTimer function as an integer value gotten by rounding off digits to the right of the decimal point.

Description

Gets the elapsed time after the execution of the StartTimer function with the unit specified in the <mode> parameter. (To use this function, specify the return value of the StartTimer function in the <start> parameter.)
Specify the return value of the StartTimer function to be executed prior to this macro function.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a non-existent number, numerical value, or combination of data types or values is specified for a parameter, an "Illegal function call" error will occur.
If a value is assigned to the return value variable or the variable is not used in an expression, a "Syntax error" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- Depending on the processing time of this macro function and the StartTimer function itself, there may be an error in the elapsed time gotten.

Example

Measures the elapsed time since the beginning of the StartTimer function until the execution of the Timer function.

```
T# = StartTimer

Rem Executes the process whose the process execution elapsed time is measured.


Rem Gets the elapsed time using the return value of the StartTimer function.
TIME& = Timer(T#, 0)
```

Useable Modules

Unit Calculation Macro / Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 3.50 or later

Related Items

ElapsedTime (Reference: ▶ Details (p.508))
StartTimer (Reference: ▶ Details (p.746))
Wait (Reference: ▶ Details (p.806))

TotalJudge

Gets the total judgement result.

Format

TotalJudge

Parameter

None.

Return value

Returns the overall judgement result as an integer value.

- 0: No judgement (unmeasured)
- 1: Judgement result OK
- -1: Judgement result NG

Description

Gets the overall judgement result that is the result of execution of the measurement flow.
If a value is assigned to the return value variable or the variable is not used in an expression, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

In the *MEASUREDISPG subroutine of the Unit Macro processing unit, gets the overall judgement result and displays a character string in the image window.

```
CHARSTRING$ = ""

Rem Get the total judgement result.
JG& = TotalJudge

Change the displayed character string depending on the overall judgement result.
If JG& = 1 Then
   CHARSTRING$ = "OK"
ElseIf JG& = -1 Then
   CHARSTRING$ = "NG"
ElseIf JG& = 0 Then
   CHARSTRING$ = "NC"
Endif

Rem Display a character string.
DrawTextG CHARSTRING$, 100, 100, 0
```

Useable Modules

Unit Calculation Macro / Unit Macro

Supported Versions

Version 3.50 or later

Related Items

DrawTextG (Reference: ▶ Details (p.502))

SetUnitJudge (Reference: ▶ Details (p.735))

TransformAngle

Applies the calibration result and position correction amount in the angle value.

Format

TransformAngle <unitNo>, <imageNo>, <mode>, <srcAngle>, <destAngle>

Parameter

Parameter name	Data type	Description
<unitNo>	Integer type	Processing unit number (0 to (number of processing units of current scene minus one)) of the processing unit that holds the data to be converted.
<imageNo>	Integer type	Measurement image number (always 0) of the processing unit that holds the data to be transformed
<mode>	Integer type	Transformation mode 0: After image transformation. -> Before image transformation. 1: Before image transformation. -> After image transformation. 10: Camera coordinates -> After calibration
<srcAngle>	Double precision real number data type	Pre-transformation angle gotten from the processing unit that holds the data to be transformed.
<destAngle>	Double precision real number data type	Angle after the transformation

Return value

None.

Description

Applies the transformation information specified in the <mode> parameter to the angle value specified in the <srcAngle> parameter on the image specified in the <imageNo> parameter of the processing unit specified in the <unitNo> parameter.
Specify 0, 1, or 10 in the <mode> parameter. If a value other than 0, 1, and 10 is specified, operation after the execution of this macro function will be undefined.
Specify the variable that will contain the transformed angle value with the <destAngle> parameter.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a non-existent number, numerical value, or combination of data types or values is specified for a parameter, an "Illegal function call" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

Transforms the value of the measurement angle before calibration measured with the search processing unit of Processing Unit number 2 is applied, to the angle value after calibration is applied. "Measurement angle" is the external reference data identification name "angle".

```
Rem Get the measurement result.
GetUnitData 2, "angle", BEFOREANGLE#

Rem Transform to the value after calibration is applied.
TransformAngle 2, 0, 10, BEFOREANGLE#, AFTERANGLE#
```

Useable Modules

Unit Calculation Macro / Unit Macro

Supported Versions

Version 3.50 or later

Related Items

GetUnitData (Reference: ▶ Details (p.552))
TransformDist (Reference: ▶ Details (p.767))
TransformXY (Reference: ▶ Details (p.771))
Ut (Reference: ▶ Details (p.793))
TransformArea (Reference: ▶ Details (p.765))
TransformLine (Reference: ▶ Details (p.769))
UnitNo (Reference: ▶ Details (p.790))

TransformArea

Applies the calibration result and position correction amount in the area value.

Format

TransformArea <unitNo>, <imageNo>, <mode>, <srcArea>, <destArea>

Parameter

Parameter name	Data type	Description
<unitNo>	Integer type	Processing unit number (0 to (number of processing units of current scene minus one)) of the processing unit that holds the data to be converted.
<imageNo>	Integer type	Measurement image number (always 0) of the processing unit that holds the data to be transformed
<mode>	Integer type	Transformation mode 0: After image transformation. -> Before image transformation. 1: Before image transformation. -> After image transformation. 10: Camera coordinates -> After calibration
<srcArea>	Double precision real number data type	Pre-transformation area gotten from the processing unit that holds the data to be transformed.
<destArea>	Double precision real number data type	Area after the transformation

Return value

None.

Description

Applies the transformation information specified in the <mode> parameter to the area value specified in the <srcArea> parameter on the image specified in the <imageNo> parameter of the processing unit specified in the <unitNo> parameter.
Specify 0, 1, or 10 in the <mode> parameter. If a value other than 0, 1, and 10 is specified, operation after the execution of this macro function will be undefined.
Specify the variable that will contain the transformed area value with the <destArea> parameter.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a non-existent number, numerical value, or combination of data types or values is specified for a parameter, an "Illegal function call" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

Transforms the value of the area before calibration measured with the labeling processing unit of Processing Unit number 2 is applied, to the area value after calibration is applied. "Area" is the external reference data identification name "area".

```
Rem Get the measurement result.
GetUnitData 2, "area", BEFOREAREA#

Rem Transform to the value after calibration is applied.
TransformArea 2, 0, 10, BEFOREAREA#, AFTERAREA#
```

Useable Modules

Unit Calculation Macro / Unit Macro

Supported Versions

Version 3.50 or later

Related Items

GetUnitData (Reference: ▶ Details (p.552))
TransformAngle (Reference: ▶ Details (p.763))
TransformDist (Reference: ▶ Details (p.767))
TransformLine (Reference: ▶ Details (p.769))
TransformXY (Reference: ▶ Details (p.771))
UnitNo (Reference: ▶ Details (p.790))
Ut (Reference: ▶ Details (p.793))

TransformDist

Applies a calibration result and position correction amount to a distance value.

Format

TransformDist <unitNo>, <imageNo>, <mode>, <srcDist>, <destDist>

Parameter

Parameter name	Data type	Description
<unitNo>	Integer type	Processing unit number (0 to (number of processing units of current scene minus one)) of the processing unit that holds the data to be converted.
<imageNo>	Integer type	Measurement image number (always 0) of the processing unit that holds the data to be transformed
<mode>	Integer type	Transformation mode 0: After image transformation. -> Before image transformation. 1: Before image transformation. -> After image transformation. 10: Camera coordinates -> After calibration
<srcDist>	Double precision real number data type	Pre-transformation distance gotten from the processing unit that holds the data to be transformed.
<destDist>	Double precision real number data type	Distance after the transformation

Return value

None.

Description

Applies the transformation information specified in the <mode> parameter to the distance value specified in the <srcDist> parameter on the image specified in the <imageNo> parameter, of the processing unit specified in the <unitNo> parameter.
Specify 0, 1, or 10 in the <mode> parameter. If a value other than 0, 1, and 10 is specified, operation after the execution of this macro function will be undefined.
In the <destDist> parameter, specify the variable that holds the transformed distance value.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a non-existent number, numerical value, or combination of data types or values is specified for a parameter, an "Illegal function call" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

Transforms the distance value of the average width before calibration is applied, which is measured with the scan edge width processing unit of Processing Unit number 2, to the distance value after calibration is applied. The "average width" is external reference data identification name "width ave".

```
Rem Get the measurement result.
GetUnitData 2, "width_ave", BEFOREDIST#

Rem Transform to the value after calibration is applied.
TransformDist 2, 0, 10, BEFOREDIST#, AFTERDIST#
```

Useable Modules

Unit Calculation Macro / Unit Macro

Supported Versions

Version 3.50 or later

Related Items

GetUnitData (Reference: ▶ Details (p.552))
TransformArea (Reference: ▶ Details (p.765))
TransformXY (Reference: ▶ Details (p.771))
Ut (Reference: ▶ Details (p.793))
TransformAngle (Reference: ▶ Details (p.763))
TransformLine (Reference: ▶ Details (p.769))
UnitNo (Reference: ▶ Details (p.790))

TransformLine

Applies the calibration result and position correction amount to a line component value.

Format

TransformLine <unitNo>, <imageNo>, <mode>, <srcA>, <srcB>, <srcC>, <destA>, <destB>, <destC>

Parameter

Parameter name	Data type	Description
<unitNo>	Integer type	Processing unit number (0 to (number of processing units of current scene minus one)) of the processing unit that holds the data to be converted.
<imageNo>	Integer type	Measurement image number (always 0) of the processing unit that holds the data to be transformed
<mode>	Integer type	Transformation mode 0: After image transformation. -> Before image transformation. 1: Before image transformation. -> After image transformation. 10: Camera coordinates -> After calibration
<srcA>	Double precision real number data type	Pre-transformation line component A gotten from the processing unit that holds the data to be transformed.
<srcB>	Double precision real number data type	Pre-transformation line component B gotten from the processing unit that holds the data to be transformed.
<srcC>	Double precision real number data type	Pre-transformation line component C gotten from the processing unit that holds the data to be transformed.
<destA>	Double precision real number data type	Transformed line component A
<destB>	Double precision real number data type	Transformed line component B
<destC>	Double precision real number data type	Transformed line component C

Return value

None.

Description

Applies the transformation information specified in the <mode> parameter to the line components of the lines specified in the <srcA>, <srcB>, and <srcC> parameters on the image specified in the <imageNo> parameter, of the processing unit specified in the <unitNo> parameter.
Specify 0, 1, or 10 in the <mode> parameter. If a value other than 0, 1, and 10 is specified, operation after the execution of this macro function will be undefined.
In the <destA>, <destB>, and <destC> parameters, specify the variables that will hold the transformed line component values.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a non-existent number, numerical value, or combination of data types or values is specified for a parameter, an "Illegal function call" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

Transforms the values of the line components measured with the scan edge position processing unit of Processing Unit number 2 before calibration is applied, to the values of the line components after calibration is applied. "Line component A" is the external reference parameter identification name "coefficientA", "line component B" is the external reference parameter identification name "coefficientB", and "line component C" is the external reference parameter identification name "coefficientC".

```
Rem Get the measurement result.
GetUnitData 2, "coefficientA", BEFOREA#
GetUnitData 2, "coefficientB", BEFOREB#
GetUnitData 2, "coefficientC", BEFOREC#

Rem Transform to the value after calibration is applied.
TransformLine 2, 0, 10, BEFOREA#, BEFOREB#, BEFOREC#, AFTERA#, AFTERB#, AFTERC#
```

Useable Modules

Unit Calculation Macro / Unit Macro

Supported Versions

Version 3.50 or later

Related Items

GetUnitData (Reference: ▶ Details (p.552))
TransformArea (Reference: ▶ Details (p.765))
TransformXY (Reference: ▶ Details (p.771))
Ut (Reference: ▶ Details (p.793))
TransformAngle (Reference: ▶ Details (p.763))
TransformDist (Reference: ▶ Details (p.767))
UnitNo (Reference: ▶ Details (p.790))

TransformXY

Applies the calibration result and position correction amount to coordinate values.

Format

TransformXY <unitNo>, <imageNo>, <mode>, <srcX>, <srcY>, <destX>, <destY>

Parameter

Parameter name	Data type	Description
<unitNo>	Integer type	Processing unit number (0 to (number of processing units of current scene minus one)) of the processing unit that holds the data to be converted.
<imageNo>	Integer type	Measurement image number (always 0) of the processing unit that holds the data to be transformed
<mode>	Integer type	Transformation mode 0: After image transformation. -> Before image transformation. 1: Before image transformation. -> After image transformation. 10: Camera coordinates -> After calibration
<srcX>	Double precision real number data type	Pre-transformation X coordinate gotten from the processing unit that holds the data to be transformed.
<srcY>	Double precision real number data type	Pre-transformation Y coordinate gotten from the processing unit that holds the data to be transformed.
<destX>	Double precision real number data type	Transformed X coordinate
<destY>	Double precision real number data type	Transformed Y coordinate

Return value

None.

Description

Applies the transformation information specified in the <mode> parameter to the coordinate values specified in the <srcX> and <srcY> parameters on the image specified in the <imageNo> parameter, of the processing unit specified in the <unitNo> parameter.
Specify 0, 1, or 10 in the <mode> parameter. If a value other than 0, 1, and 10 is specified, operation after the execution of this macro function will be undefined.
In the <destX> and <destY> parameters, specify the variables that will store the transformed coordinate values.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a non-existent number, numerical value, or combination of data types or values is specified for a parameter, an "Illegal function call" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

Transforms the pre-calibration measurement coordinates measured with the search processing unit of Processing Unit 2 to the measurement coordinates after calibration is applied. "Measurement coordinate X" is the external reference data identification name "positionX", and "measurement coordinate Y" is the external reference data identification name "positionY".

```
Rem Get the measurement result.
GetUnitData 2, "positionX", BEFOREX#
GetUnitData 2, "positionY", BEFOREY#

Rem Transform to the value after calibration is applied.
TransformXY 2, 0, 10, BEFOREX#, BEFOREY#, AFTERX#, AFTERY#
```

Useable Modules

Unit Calculation Macro / Unit Macro

Supported Versions

Version 3.50 or later

Related Items

GetUnitData (Reference: ▶ Details (p.552))
TransformArea (Reference: ▶ Details (p.765))
TransformLine (Reference: ▶ Details (p.769))
Ut (Reference: ▶ Details (p.793))
TransformAngle (Reference: ▶ Details (p.763))
TransformDist (Reference: ▶ Details (p.767))
UnitNo (Reference: ▶ Details (p.790))

Try Catch End Try

Detects an error occurrence and executes an exception process.

Format

Try
<statement>
Catch
<exceptionStatement>]
End Try

Parameter

Parameter name	Data type	Description
<statement>	---	Statement that can make error occurs
<exceptionstatement>	---	Statement that is executed when an error occurred

Return value

None.

Description

Executes the specified Catch block statement in the <exceptionStatement> parameter if an error is occurred as a result of the Try block statement execution specified in the <statement> parameter.
If there is no error occurrence as a result of executing all statements in the Try block statement, the process execution ends without executing the Catch block statement.
Use the Errno function or the Errcmnd$ function in the Catch block statement to get a macro function name and occurred error number in the Try block statement. (Reference: ▶Errno (p.515)) (Reference: ▶Errcmnd$ (p.513))
Errors in the Catch block statement cannot be detected. If statements in the Catch block can cause an error, nest the Try Catch-End Try statement to detect the error occurrence.
If the program process is jumped into or out of the Try and Catch block statements using the Goto function in a statement, unexpected operation may occur.
If neither the Try statement nor the End Try statement is used, either the "CATCH without TRY" or "END TRY without TRY" error will occur depending on the statement that is used.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

Uses the Try Catch-End Try statement in the *MEASUREPROC subroutine of the Unit Macro processing unit to detect the error occurrence and get the detected error number.

```
*MEASUREPROC

  Try
    WORK& = 0
    SUMM& = 100 + 200 + 300

    ANS& = SUMM& / WORK&
  Catch
    If Errno = 11 Then
      Rem Output the error number and the error content on the system status console window
      Print "Error Number = " + Str$(Errno) + ", Division by Zero"
    Endif
  End Try

Return
```

Useable Modules

Unit Calculation Macro / Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 3.50 or later

Related Items

Errcmnd$ (Reference: ▶ Details (p.513))
If Then Elseif Else EndIf (Reference: ▶ Details (p.564))
Str$ (Reference: ▶ Details (p.749))
Errno (Reference: ▶ Details (p.515))
Print (Reference: ▶ Details (p.639))

UCase$

Converts an lower case letter to a upper case letter.

Format

UCase$(<string>)

Parameter

Parameter name	Data type	Description
<string>	Character string type	Character string contains an alphabet to be converted to upper case.

Return value

Returns the case converted character string as a string type value.

Description

Converts the lower case letters in the character strings specified in the <string> parameter to upper case.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a character string longer than 255 characters is specified for a character string parameter, the 255-character string before the 256th character is used for the macro function process. Characters after the 256th character will be discarded.
If a value is assigned to the return value variable or the variable is not used in an expression, a "Syntax error" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

Converts an lower case letter to a upper case letter.

```
CHARA1$ = "Measurement Result = 100.0(OK)"

Rem Convert the lower case letters in the character strings to upper case.
CHARA2$ = UCase$(CHARA1$)
```

The result is shown below.

```
CHARA2$ = "MEASUREMENT RESULT = 100.0(OK)"
```

Useable Modules

Unit Calculation Macro / Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 4.20 or later

Related Items

Asc (Reference: ▶ Details (p.413))
Chr$ (Reference: ▶ Details (p.427))
Hex$ (Reference: ▶ Details (p.560))
LCase$ (Reference: ▶ Details (p.589))
Left$ (Reference: ▶ Details (p.591))
Len (Reference: ▶ Details (p.593))
Mid$ (Reference: ▶ Details (p.621))
Piece$ (Reference: ▶ Details (p.637))
Right$ (Reference: ▶ Details (p.664))
Str$ (Reference: ▶ Details (p.749))
Str2$ (Reference: ▶ Details (p.751))
Val (Reference: ▶ Details (p.794))

UnitCount

Gets the number of registered processing units.

Format

UnitCount

Parameter

None.

Return value

Returns the number of registered processing units as an integer value.

Description

Gets the number of processing units registered in the current scene.
If a value is assigned to the return value variable or the variable is not used in an expression, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

Adds the search processing unit to the end of the measurement flow.

```
Rem Get the number of processing units registered in the current flow.
UNUM& = UnitCount

Rem Specify the processing item identifier.
IDENT$ = "Search"

Rem Add the "Search" processing item to the end of the flow.
AssignUnit UNUM& , IDENT$
```

Useable Modules

Scene Control Macro / Communication Command Macro

Supported Versions

Version 3.50 or later

Related Items

AssignUnit (Reference: ▶ Details (p.415))
CheckUnit (Reference: ▶ Details (p.425))
CopyUnit (Reference: ▶ Details (p.442))
DeleteUnit (Reference: ▶ Details (p.460))
InsertUnit (Reference: ▶ Details (p.573))
MeasureStart (Reference: ▶ Details (p.617))
MeasureStop (Reference: ▶ Details (p.619))
MoveUnit (Reference: ▶ Details (p.625))

UnitData

Gets the numerical data of a processing unit.

Format

UnitData(<unitNo>, <dataNo>)
UnitData(<unitNo>, <dataIdent>)

Parameter

Parameter name	Data type	Description
<unitNo>	Integer type	Processing unit number (0 to (the number of registered processing units in the current scene minus one))
<dataNo>	Integer type	External reference data of the processing unit data to get (reference: ▶ Vision System FH/FZ5 Series Processing Item Function Reference Manual (Cat. No. Z341)"
<dataIdent>	Character string type	Data identification name of processing unit data to be gotten.

Return value

Returns processing unit data as integer or double precision real values.
If non-numerical data is gotten, the data is converted to numerical values and returned.

Description

Gets numerical data of the external reference data number specified in the <dataNo> parameter, in the processing unit specified in the <unitNo> parameter. The data can also be gotten by specifying the <dataIdent> parameter instead of the <dataNo> parameter.
To get data other than numerical data, use the UnitData$ function or the GetUnitData function.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a non-existent number, numerical value, or combination of data types or values is specified for a parameter, an "Illegal function call" error will occur.
If a value is assigned to the return value variable or the variable is not used in an expression, a "Syntax error" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

Gets the value of the measurement X coordinate of the search processing unit of Processing Unit number 5. The measurement X coordinate is external reference data number 6 and external reference data identification name "X".

```
SEARCH# = UnitData(5, 6)

Rem The same result can be gotten by specifying "X" instead of 6.
SEARCH# = UnitData(5, "X")
```

Useable Modules

Unit Calculation Macro / Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 3.50 or later

Related Items

GetUnitData (Reference: ▶ Details (p.552))
UnitData$ (Reference: ▶ Details (p.780))
UnitNo (Reference: ▶ Details (p.790))
SetUnitData (Reference: ▶ Details (p.731))
UnitData2 (Reference: ▶ Details (p.782))
Ut (Reference: ▶ Details (p.793))

UnitData$

Gets the character string data of the specified processing unit.

Format

UnitData$(<unitNo>, <dataNo>)
UnitData$(<unitNo>, <dataIdent>)

Parameter

Parameter name	Data type	Description
<unitNo>	Integer type	Processing unit number (0 to (the number of registered processing units in the current scene minus one))
<dataNo>	Integer type	External reference data of the processing unit data to get (reference: ▶ Vision System FH/FZ5 Series Processing Item Function Reference Manual (Cat. No. Z341)"
<dataIdent>	Character string type	Data identification name of processing unit data to be gotten.

Return value

Returns the character string data of the processing unit as a string type value.
If non character string data is gotten, the data is converted to a character string and returned.

Description

Gets the data of the external reference data number specified in the <dataNo> parameter, held by the processing unit specified in the <unitNo> parameter. The data can also be gotten by specifying the <dataIdent> parameter instead of the <dataNo> parameter.
To get data other than character string data, use the UnitData function or the GetUnitData function.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a non-existent number, numerical value, or combination of data types or values is specified for a parameter, an "Illegal function call" error will occur.
If a value is assigned to the return value variable or the variable is not used in an expression, a "Syntax error" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

Gets the decode character string of the 2D code processing unit of Processing Unit number 5. The decode character string is external reference data number 7 and external reference data identification name "decodeCharStr".

```
DECODECHAR$ = UnitData$(5, 7)

Rem The same result can be gotten by specifying "decodeCharStr" instead of 7.
DECODECHAR$ = UnitData$(5, "decodeCharStr")
```

Useable Modules

Unit Calculation Macro / Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 3.50 or later

Related Items

GetUnitData (Reference: ▶ Details (p.552))
UnitData (Reference: ▶ Details (p.778))
UnitNo (Reference: ▶ Details (p.790))
SetUnitData (Reference: ▶ Details (p.731))
UnitData2 (Reference: ▶ Details (p.782))
Ut (Reference: ▶ Details (p.793))

UnitData2

Gets the drawing coordinate data of a processing unit.

Format

UnitData2(<unitNo>, <dataNo>)
UnitData2(<unitNo>, <dataIdent>)

Parameter

Parameter name	Data type	Description
<unitNo>	Integer type	Processing unit number (0 to (the number of registered processing units in the current scene minus one))
<dataNo>	Integer type	External reference data of the processing unit data to get (reference: ▶ Vision System FH/FZ5 Series Processing Item Function Reference Manual (Cat. No. Z341)"
<dataIdent>	Character string type	Data identification name of processing unit data to be gotten.

Return value

Returns processing unit data as integer or double precision real values.

Description

Gets numerical data of the external reference data number specified in the <dataNo> parameter, in the processing unit specified in the <unitNo> parameter. The data can also be gotten by specifying the <dataIdent> parameter instead of the <dataNo> parameter.
This macro function can be used to get measurement coordinate values prior to transformation by calibration or otherwise. Use to get drawing coordinates for the display of measurement results in the image window.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a non-existent number, numerical value, or combination of data types or values is specified for a parameter, an "Illegal function call" error will occur.
If a value is assigned to the return value variable or the variable is not used in an expression, a "Syntax error" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

In the *MEASUREDISPG subroutine of the Unit Macro processing unit, gets the measurement X and Y coordinate values of the search processing unit of Processing Unit number 5 and displaying the cursor in the image coordinates. The measurement X coordinate is the external reference data number 8 and the external reference data identification name "X", and the measurement Y coordinate is the external reference data number 7 and the external reference data identification name "Y".
Even when the "Calibration" setting of the search processing unit is "ON", this macro function can be used to get drawing coordinates without concern for the calibration settings.

```
SEARCHX& = UnitData2(5, 6)
SEARCHY& = UnitData2(5, 7)

Rem The same result can be gotten by specifying "X" instead of 6.
SEARCHX& = UnitData2(5, "X")
SEARCHY& = UnitData2(5, "Y")

Rem Display the cursor in the coordinates prior to position correction.
DrawCursor SEARCHX&, SEARCHY&, 0, UnitNo
```

Useable Modules

Unit Calculation Macro / Unit Macro

Supported Versions

Version 3.50 or later

Related Items

GetUnitData (Reference: ▶ Details (p.552))
SetUnitData (Reference: ▶ Details (p.731))
UnitData (Reference: ▶ Details (p.778))
UnitData$ (Reference: ▶ Details (p.780))
UnitNo (Reference: ▶ Details (p.790))
Ut (Reference: ▶ Details (p.793))

UnitInfo

Gets the processing unit information.

Format

UnitInfo(<unitNo>, <kind>)

Parameter

Parameter name	Data type	Description
<unitNo>	Integer type	Processing unit number (0 to (number of processing units of current scene minus one)) of the processing unit whose information is to be gotten.
<kind>	Integer type	Type of information 0: Processing item type Number that indicates the processing item type. The values below can be gotten. 0: Inspect and Measure (measurement) 1: Input image (Image input) 2: Compensate image (image correction) 3: Support Inspection and Measurement (supplementary measurement) 4: Branch (branch control) 5: Output result (result output) 6: Display result (result display) 1: Setting data structure size Size of setting data structure. Units are bytes. 2: Measurement data structure size Size of measurement data structure. Units are bytes. 3: Control data structure size Size of control data structure. Units are bytes. 4: The maximum number of figure data Maximum number of figure data items held by processing unit. 5: Maximum number of model data Maximum number of model data items held by processing unit. 6: Maximum number of image data Maximum number of image data items held by processing unit. 7: Maximum number of inner processing unit Maximum number of processing units held (incorporated) by processing unit. 8: Whether camera setting is effective or not Displays whether or not the processing unit updates the camera settings at measurement initialization and other times. "1" is returned when there is an image input processing unit for camera image input. 0: Camera settings invalid 1: Camera settings valid 9: Whether processing unit measure processing can parallel or not Displays whether or not parallel execution of processing units is possible in the measurement flow during measurement. "1" is returned when the processing units support parallel processing. 0: Parallel processing disabled 1: Parallel processing enabled For the parallel processing, refer to Parallel Processing. (Reference: ▶Parallel Processing (p.282))

Return value

Returns processing unit information as an integer value. Returns -1 if information does not exist.

Description

Gets the information specified in the <kind> parameter of the processing unit specified in the <unitNo> parameter.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
Even if a non-existent number, numerical value, or combination of data types or values is specified for the parameter, an error will not occur.
If a value is assigned to the return value variable or the variable is not used in an expression, a "Syntax error" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

Checks if model registration is possible on the processing unit of Processing Unit number 2.

```
If UnitInfo(2, 5) > 0 Then

  Rem Write a model registration processing statements here.

Endifi
```

Useable Modules

Unit Calculation Macro / Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 3.50 or later

Related Items

UnitNo (Reference: ▶ Details (p.790))
Ut (Reference: ▶ Details (p.793))

UnitItemIdent$

Gets the processing item identification name of the specified processing unit.

Format

UnitItemIdent$(<unitNo>)

Parameter

Parameter name	Data type	Description
<unitNo>	Integer type	Processing unit number (0 to (number of processing units of current scene minus one)) of processing unit whose processing item identification name is to be gotten.

Return value

Returns the value of the processing item identification name as a character string.

Description

Gets the processing item identification name of the processing unit specified in the <unitNo> parameter. If the specified processing unit is not registered on the measurement flow, the null character string ("") is returned.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
Even if a non-existent number, numerical value, or combination of data types or values is specified for the parameter, an error will not occur.
If a value is assigned to the return value variable or the variable is not used in an expression, a "Syntax error" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

In the scene macro, searches for the search processing unit registered in the measurement flow, and updating the correlation value lower limit, which is the judgement condition, to "70". The search correlation value lower limit is external reference number 143.

```
Rem Get the enrollment number of the processing unit.
COUNT& = UnitCount

Rem Search the search processing unit.
For I&=0 To COUNT&-1
  If UnitItemIdent$(I&) = "Search" Then
    Rem Update the correlation value lower limit, which is the judgement condition of the search processing unit.
    SetUnitData I&, 143, 70
  Endif
Next
```

Useable Modules

Unit Calculation Macro / Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 3.50 or later

Related Items

SetUnitData (Reference: ▶ Details (p.731))
UnitNo (Reference: ▶ Details (p.790))
UnitCount (Reference: ▶ Details (p.777))
Ut (Reference: ▶ Details (p.793))

UnitJudge

Gets the judgement result of a processing unit.

Format

UnitJudge(<unitNo>)

Parameter

Parameter name	Data type	Description
<unitNo>	Integer type	Processing unit number to get the judgement result of the processing unit (0 to (the number of registered processing units in the current scene minus one))

Return value

Returns the judgement result as an integer value.

0: No judgement (unmeasured)
1: Judgement result OK
-1: Judgement result NG
-10: Judgement result error (image format mismatch)
-11: Judgement result error (unregistered model)
-12: Judgement result error (insufficient memory)
-20: Judgement result error (other errors)

Description

Gets the judgement result of the processing unit specified in the <unitNo> parameter.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a non-existent number, numerical value, or combination of data types or values is specified for a parameter, an "Illegal function call" error will occur.
If a value is assigned to the return value variable or the variable is not used in an expression, a "Syntax error" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

Gets the judgement result of the processing unit of Processing Unit number 5.

```
JUDGE& = UnitJudge(5)
```

Useable Modules

Unit Calculation Macro / Unit Macro

Supported Versions

Version 3.50 or later

Related Items

GetUnitData (Reference: ▶ Details (p.552))
TotalJudge (Reference: ▶ Details (p.761))
Ut (Reference: ▶ Details (p.793))
SetUnitJudge (Reference: ▶ Details (p.735))
UnitNo (Reference: ▶ Details (p.790))

UnitNo

Gets the processing unit number.

Format

UnitNo

Parameter

None.

Return value

Returns the processing unit number as an integer value.

Description

Gets the processing unit number of the processing unit.
If a value is assigned to the return value variable or the variable is not used in an expression, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

Gets the judgement result of the processing unit registered immediately before the current processing unit.

```
Rem Get the processing unit number of this processing unit
UNO& = UnitNo

Rem Get the judgement result of the processing unit registered immediately before this processing unit.
JUDGE& = UnitJudge(UNO& - 1)
```

Useable Modules

Unit Calculation Macro / Unit Macro

Supported Versions

Version 3.50 or later

Related Items

UnitJudge (Reference: ▶ Details (p.788))
Ut (Reference: ▶ Details (p.793))

UnitTitle$

Gets the title of a processing unit.

Format

UnitTitle$(<unitNo>)

Parameter

Parameter name	Data type	Description
<unitNo>	Integer type	Processing unit number (0 to (number of processing units of current scene minus one)) of processing unit whose title is to be gotten.

Return value

Returns the title as a character string.

Description

Gets the title of the processing unit specified in the <unitNo> parameter.
The title can be gotten in a language based on the language setting.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a non-existent number, numerical value, or combination of data types or values is specified for a parameter, an "Illegal function call" error will occur.
If a value is assigned to the return value variable or the variable is not used in an expression, a "Syntax error" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

In the scene control macro, searches for the search processing unit registered in the measurement flow with the title "Bolt search", and updates the correlation value lower limit, which is the judgement condition, to "70". The search correlation value lower limit is external reference number 143.

```
Rem Get the enrollment number of the processing unit.
COUNT& = UnitCount

Rem Search for the processing unit with the title "Bolt search"
For I&=0 To COUNT&-1
  If UnitTitle$(I&) = "Bolt search" Then
    Rem Change the correlation value lower limit, which is the judgement condition
    SetUnitData I&, 143, 70
  Endif
Next
```

Useable Modules

Unit Calculation Macro / Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 3.50 or later

Related Items

SetUnitData (Reference: ▶ Details (p.731))
UnitCount (Reference: ▶ Details (p.777))
Ut (Reference: ▶ Details (p.793))
SetUnitTitle (Reference: ▶ Details (p.737))
UnitNo (Reference: ▶ Details (p.790))

Ut

Gets a processing unit number based on the specified unit label.

Format

Ut(<unitLabel>)

Parameter

Parameter name	Data type	Description
<unitLabel>	Character string type	Unit label of processing unit

Return value

Returns the processing unit number as an integer value.

Description

Gets the processing unit number of the processing unit that has the set unit label specified in the <unitLabel> parameter.
Knowing the unit label allows you to get the processing unit number, and thus even if the measurement flow is changed and the unit number changes, there is no need to change the program.
Set the unit label in advance with the scene control macro tool.
Reference: ▶Description of the Setting Screen of the Scene Control Macro Tool and How to Configure Settings (p.190)
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a value is assigned to the return value variable or the variable is not used in an expression, a "Syntax error" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

In the scene control macro tool, gets the processing unit number of the unit label "Position Search" set in the search processing unit of processing unit number 10, and gets the judgement result.

```
Rem Specify the unit label that was set with the scene control macro tool and get the processing unit number.
UNITNO& = Ut("PositionSearch")

Rem Using the gotten processing unit number, get the judgement result of the processing unit.
JG& = UnitJudge(UNITNO&)
```

Useable Modules

Unit Calculation Macro / Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 5.20 or later

Related Items

UnitJudge (Reference: ▶ Details (p.788))
UnitNo (Reference: ▶ Details (p.790))

Val

Converts a numeric character string to numeric value.

Format

Val(<string>)

Parameter

Parameter name	Data type	Description
<string>	Character string type	Numeric character string converted to numeric value

Return value

Returns the value as the double precision type real number.

Description

Converts the specified numeric character string in the <string> parameter to the numeric value.
Val is the inverse function of Str$. Str$ converts the specified numeric value to the numeric character string.
Specify a character string starting with either "+", "-", ".", or half-width numbers "0" to "9" to the <string> parameter. If other characters than above is in the first character of the specified character string, 0 is returned.
If there are characters that cannot be converted to numeric values in the specified alphanumeric character string with the <string> parameter, characters from the beginning of the string to one character before the inconvertible character are converted to a numeric value.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
Even if a character string longer than 255 characters is specified for a character string parameter, an error will not occur.
If a value is assigned to the return value variable or the variable is not used in an expression, a "Syntax error" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

Converts a numeric character string to numeric value.

```
VALUE1# = Val("123.456")
VALUE2# = Val("-123.456")
VALUE3# = Val(".123")
VALUE4# = Val("-.456")
VALUE5# = Val("123"+"."+"456")
VALUE6# = Val("123+456")
```

The result is shown below.

```
VALUE1# = 123.456
VALUE2# = -123.456
VALUE3# = 0.123
VALUE4# = -0.456
VALUE5# = 123.456
VALUE6# = 123
```

Useable Modules

Unit Calculation Macro / Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 3.50 or later

Related Items

Asc (Reference: ▶ Details (p.413))
Hex$ (Reference: ▶ Details (p.560))
Left$ (Reference: ▶ Details (p.591))
Mid$ (Reference: ▶ Details (p.621))
Right$ (Reference: ▶ Details (p.664))
Str2$ (Reference: ▶ Details (p.751))
Chr$ (Reference: ▶ Details (p.427))
LCase$ (Reference: ▶ Details (p.589))
Len (Reference: ▶ Details (p.593))
Piece$ (Reference: ▶ Details (p.637))
Str$ (Reference: ▶ Details (p.749))
UCase$ (Reference: ▶ Details (p.775))

VarList

Outputs a list of the values of the specified variables in the system status console window.

Format

VarList [<variableName>]

Parameter

Parameter name	Data type	Description
<variableName>	Character string type	Name of variable to be output. This parameter can be omitted. When the parameter is not specified, a list of the values of the variables used in the current scope is output.

Return value

None.

Description

Outputs the values of the variables specified in the <variableName> parameter to the system status console window.
This macro function cannot output the values of the elements of an array variable. To output the value of each element of an array variable, specify each element of the array variable by entering a question mark (?), a half-width space ( ), and then the array variable and the element number (for example, "? AA&(5)").
Characters * (character string wildcard operator) and ? (single character wildcard operator) can be used as wildcards for the <variableName> specification.
Wildcards can be used to specify file names in the following manner.

*	Specify all variables.
???	Specify variables with a 3-character variable name.
A*	Specify variables with a variable name that starts with "A".
*A*A*A*	Specify variables with a variable name that includes at least three "A" letters.
????*	Specify variables with a variable name that consists of four or more characters.

If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a character string longer than 255 characters is specified for a character string parameter, a "String too long" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- Use this macro function when program execution has been stopped by the Stop function.

Example

Among the variables used by the Unit Macro processing unit of Processing Unit number 1 (AA&, AB#, BB&, ABC$, DEF#), outputs a list of the values of variables AA& and AB$, which start with "A" and consist of three characters including the type identifier.

```
Macro(U1)>VarList "A?&"
AA&=123
AB$=123.456
```

Useable Modules

Unit Calculation Macro / Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 5.20 or later

Related Items

Cont (Reference: ▶ Details (p.436))
DebugPrint (Reference: ▶ Details (p.458))
Print (Reference: ▶ Details (p.639))
SetVar (Reference: ▶ Details (p.741))
Debug (Reference: ▶ Details (p.456))
List (Reference: ▶ Details (p.597))
SetStop (Reference: ▶ Details (p.723))
Stop (Reference: ▶ Details (p.747))

VarPop

Restores the value of the variables that are saved temporarily.

Format

VarPop

Parameter

None.

Return value

None.

Description

Restore the values of all variables that were saved by the most recent VarPush function.
If the VarPush function is executed more than once, the saved values are restored from the latest saved variable to the oldest saved variable.
If this function is executed before saving the values with the VarPush function, an "Internal error" will occur.

Usage Cautions

- None.

Example

Uses the variables used in subroutines as local variables.

```
*EXPA

  Rem Display the current value of the variable.
  Print A&, B&, C&, D#, E#

  Rem Use the EXPA subroutine to save the current values of the variables in order to prepare for treating the variable
  as local variables.
  VarPush A&, B&, C&, D#, E#

  Rem Use A&, B&, C&, D#, and E# freely.
  GetUnitData 2, "CR", A&
  GetUnitData 3, "CR", B&
  GetUnitData 4, "CR", C&
  GetUnitData 5, "X", D#
  GetUnitData 6, "Y", E#

  Rem Check the current values of the variables before calling the subroutine.
  Print A&, B&, C&, D#, E#

  Rem Variables named A&, B&, C& are used in the *EXPB subroutine.
  Rem These names are also used for variables in this *EXPA subroutine.
  Rem Although being nested with the Gosub statement in this program example,
  Rem saving and restoration of variable values are performed with the Varpush and Varpop functions within the *EXPB
  subroutine,
  Rem so as to prevent variable values from being unintentionally overwritten.
  Gosub *EXPB

  Rem Check the current value of the variables after calling the subroutine.
  Print A&, B&, C&, D#, E#

  Rem Restore the current value of the variables that were saved at the beginning of the subroutine EXPA.
  VarPop

Return


*EXPB
```

```
Rem Use the EXPB subroutine to save the current values of the variables in order to prepare for treating the variable
as local variables.
Rem Values in variables A&, B&, C&, D#, E# are saved in different areas from
Rem where the Varpush statement in the earlier part of the *EXPA subroutine saves.
Rem This prevents the previously saved values from being overwritten.
Rem VarPush can be executed up to 16 times consecutively.
VarPush A&, B&, C&, D#, E#

Rem Use A&, B&, C&, D#, and E# freely.
GetUnitData 2, "X", A&
GetUnitData 3, "X", B&
GetUnitData 4, "X", C&
D# = 3
E# = 100 / 512

Rem Check the current values of the variables after change.
Print A&, B&, C&, D#, E#

Rem Restore the current value of the variables that were saved at the beginning of the subroutine EXPB.
VarPop

Return
```

Useable Modules

Unit Calculation Macro / Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 3.50 or later

Related Items

VarPush (Reference: ▶ Details (p.801))

VarPush

Saves the value of the variables that are saved temporarily.

Format

VarPush <variable>[, <variable>[,..., <variable>]]

Parameter

Parameter name	Data type	Description
<variable>	Character string type	Variable name of the variable whose value is saved temporarily

Return value

None.

Description

Temporarily save the variable value specified in the <variable> parameter. Execute the Varpush function to restore the saved value. (Reference: ▶VarPop (p.815))

If this macro function is executed 17 times or more without executing the VarPop function, an "Internal error" will occur. Execute the VarPop function to restore the value so that this function execution count is equal to or less than the Varpop execution count + 16.

If variables except array variable are specified as parameter, a "Type mismatch" error will occur.

Usage Cautions

- None.

Example

Uses the variables used in subroutines as local variables.

```
*EXPA

  Rem Display the current value of the variable.
  Print A&, B&, C&, D#, E#

  Rem Use the EXPA subroutine to save the current values of the variables in order to prepare for treating the variable
  as local variables.
  VarPush A&, B&, C&, D#, E#

  Rem Use A&, B&, C&, D#, and E# freely.
  GetUnitData 2, "CR", A&
  GetUnitData 3, "CR", B&
  GetUnitData 4, "CR", C&
  GetUnitData 5, "X", D#
  GetUnitData 6, "Y", E#

  Rem Check the current values of the variables before calling the subroutine.
  Print A&, B&, C&, D#, E#

  Rem Variables named A&, B&, C& are used in the *EXPB subroutine.
  Rem These names are also used for variables in this *EXPA subroutine.
  Rem Although being nested with the Gosub statement in this program example,
  Rem saving and restoration of variable values are performed with the Varpush and Varpop functions within the *EXPB
  subroutine,
  Rem so as to prevent variable values from being unintentionally overwritten.
  Gosub *EXPB

  Rem Check the current value of the variables after calling the subroutine.
  Print A&, B&, C&, D#, E#

  Rem Restore the current value of the variables that were saved at the beginning of the subroutine EXPA.
  VarPop

Return


*EXPB
```

```
Rem Use the EXPB subroutine to save the current values of the variables in order to prepare for treating the variable
as local variables.
Rem Values in variables A&, B&, C&, D#, E# are saved in different areas from
Rem where the VarPush statement in the earlier part of the *EXPA subroutine saves.
Rem This prevents the previously saved values from being overwritten.
Rem VarPush can be executed up to 16 times consecutively.
Varpush A&, B&, C&, D#, E#

Rem Use A&, B&, C&, D#, and E# freely.
GetUnitData 2, "X", A&
GetUnitData 3, "X", B&
GetUnitData 4, "X", C&
D# = 3
E# = 100 / 512

Rem Check the current values of the variables after change.
Print A&, B&, C&, D#, E#

Rem Restore the current value of the variables that were saved at the beginning of the subroutine EXPB.
VarPop

Return
```

Useable Modules

Unit Calculation Macro / Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 3.50 or later

Related Items

VarPop (Reference: ▶ Details (p.798))

VarSave

Saves the values of the variables in the scene data.

Format

VarSave <variableName>

Parameter

Parameter name	Data type	Description
<variableName>	Character string type	Name of the variables to save

Return value

None.

Description

Saves the values of the variables specified in the <variableName> parameter to the scene data.
Characters * (character string wildcard operator) and ? (single character wildcard operator) can be used as wildcards for the <variableName> specification.
Wildcards can be used to specify file names in the following manner.

*	Specify all variables.
???	Specify variables with a 3-character variable name.
A*	Specify variables with a variable name that starts with "A".
*A*A*A*	Specify variables with a variable name that includes at least three "A" letters.
????*	Specify variables with a variable name that consists of four or more characters.

The variable value saved with this macro function will be read when the scene data is loaded.
If this macro function is executed multiple times, values are restored to the original variables in execution order of the VarSave statement (from the oldest saved variable value to the latest saved variable value) at the loading of the scene data.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
Even if a non-existent number, numerical value, or combination of data types or values is specified for the parameter, an error will not occur.
If a character string longer than 255 characters is specified for a character string parameter, a "String too long" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- This macro function can only be used in the *SAVEPROC subroutine. If used in another subroutine, an "Illegal function call" error will occur.

Example

In SAVARPOC subroutine of scene control macro, saves the version information managed independently in scene data.

```
*SAVEPROC

  Rem Create a variable to store the version information and set 100.
  Version& = 100

  Rem Save the version information.
  VarSave "Version*"

Return
```

Useable Modules

Scene Control Macro

Supported Versions

Version 5.20 or later

Related Items

None.

Wait

Pauses the program process for the specified amount of time elapses.

Format

Wait <time>

Parameter

Parameter name	Data type	Description
<time>	Integer type	Standby time (ms)

Return value

None.

Description

Pauses the program process on the period of time specified in the <time> parameter.
When the process is performed in the background while waiting, the background process will be performed without waiting.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a value outside the range -2147483648 to 2147483647 is specified as an integer parameter, an "Overflow" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- An error may occur between the waiting time specified as a parameter and the actual waiting time.

Example

After switching to the scene 2 with the communication command macro, waits for 10 ms.

```
Rem Switch the scene.
ChangeScene 2

Rem Wait 10ms
Wait 10
```

Useable Modules

Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 3.50 or later

Related Items

ChangeScene (Reference: ▶ Details (p.423))
ElapsedTime (Reference: ▶ Details (p.508))
StartTimer (Reference: ▶ Details (p.746))
Timer (Reference: ▶ Details (p.759))

WritePlcMemory

Writes values in the PLC memory area.

Format

WritePlcMemory <ioIdent>, <area>, <channelOffset>, <channelCount>, <writeData()>

Parameter

Parameter name	Data type	Description
<ioIdent>	Character string type	Identification name of the communication module to be used (Reference: ▶List of I/O Modules (p.341))
<area>	Integer type	Area type number of data output area to be written to.
<channelOffset>	Integer type	Offset from beginning of data output area to address where writing is to start.
<channelCount>	Integer type	Data size to write
<writeData()>	Integer array	Data to write

Return value

None.

Description

Writes the amount of data specified in the <channelCount> parameter from the address offset by the amount of the value specified in the <channelOffset> parameter, of the PLC area type specified in the <area> parameter by using the communication module specified in the <ioIdent> parameter.
Before using this macro function to write data to the PLC memory area, execute the SetPlcData function to set the data to be written.
In the <writeData()> parameter, specify the 1D integer array variable that stores the data to be written, without adding element numbers but adding () to the variables.
This macro function cannot be used to write data to a PLC that is connected by other than the PLC link communication module.
In the <area> parameter, specify the identification of the register that is set with the PLC link setting in the system settings.
In the <channelCount> parameter, specify the size in channel units. The size of one integer type data item is two channels (4 bytes), and thus to write one integer value, a one-element array should be prepared with the <writeData()> parameter, and 2 should be specified in the <channelCount> parameter.
If a size larger than the array size specified in the <writeData()> parameter is specified in the <channelCount> parameter, a "Subscript out of range" error will occur.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a non-existent number, numerical value, or combination of data types or values is specified for a parameter, an "Illegal function call" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.
Writes the data to the data output area. Note that if the data is written to the response area, the response data will be overwritten after the command processing.

Usage Cautions

- Before using this macro function to write data, always use the SetPlcData function to set the data to be written. If the data is directly set in the <writeData()> parameter without using the SetPlcData function, the correct data may not be set.

Example

In the communication macro, writes measurement coordinate X and measurement coordinate Y of the search processing unit of Processing Unit number 2 to the PLC connected by PLC link. Measurement coordinate X is external data number 6, and measurement coordinate Y is external data number 7.

```
IOMODULE$ = "UdpPlcLink"

Rem Get the measurement result.
GetUnitData 2, 6, X#
GetUnitData 2, 7, Y#

Rem Convert the real number value multiplied by 1,000 to the integer value.
VALUE0& = Int(X# * 1000)
VALUE1& = Int(Y# * 1000)

Rem Get the settings of the output data area.
GetSystemData IOMODULE$, "outputArea", AREA&
GetSystemData IOMODULE$, "outputMemoryAddress", ADDRESS&

Rem Store the data to be written in an integer array variable.
Dim DATA&(1)
SetPlcData IOMODULE$, DATA&(), 0, 4, VALUE0&
SetPlcData IOMODULE$, DATA&(), 4, 4, VALUE1&

Rem Write the data (4ch) in data output area.
WritePlcMemory IOMODULE$, AREA&, ADDRESS&, 4, DATA&()
```

Useable Modules

Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 4.20 or later

Related Items

GetPlcData (Reference: ▶ Details (p.538))
GetUnitData (Reference: ▶ Details (p.552))
ReadPlcMemory (Reference: ▶ Details (p.649))
GetSystemData (Reference: ▶ Details (p.546))
Int (Reference: ▶ Details (p.575))
SetPlcData (Reference: ▶ Details (p.709))

XOR

Gets the exclusive disjunction (XOR) of two expressions.

Format

<expression1> XOR <expression2>

Parameter

Parameter name	Data type	Description
<expression1>	Integer type	Expression to get the exclusive disjunction
<expression2>	Integer type	Expression to get the exclusive disjunction

Return value

Returns the XOR value as an integer value.

Description

Gets the XOR value of the expressions specified in the <expression 1> and <expression 2> parameters. (Each bit of the two expressions is computed separately.)
When the values of the <expression1> parameter and <expression2> parameter are double precision real values, the decimal part is rounded off.
If an incorrect data type is specified for a parameter, a "Type mismatch" error will occur.
If a value outside the range -2147483648 to 2147483647 is specified as an integer parameter, an "Overflow" error will occur.
If a value is assigned to the return value variable or the variable is not used in an expression, a "Syntax error" error will occur.
If the format is written incorrectly, such as writing the macro function name incorrectly, omitting a comma, or omitting a half-width space, a "Syntax error" error will occur.

Usage Cautions

- None.

Example

Gets the XOR value of 12 and 31.

```
DATA1& = 12
DATA2& = 31

DATA3& = DATA1& XOR DATA2&
```

The result is shown below.

```
DATA3& = 19
```

Useable Modules

Unit Calculation Macro / Scene Control Macro / Communication Command Macro / Unit Macro

Supported Versions

Version 3.50 or later

Related Items

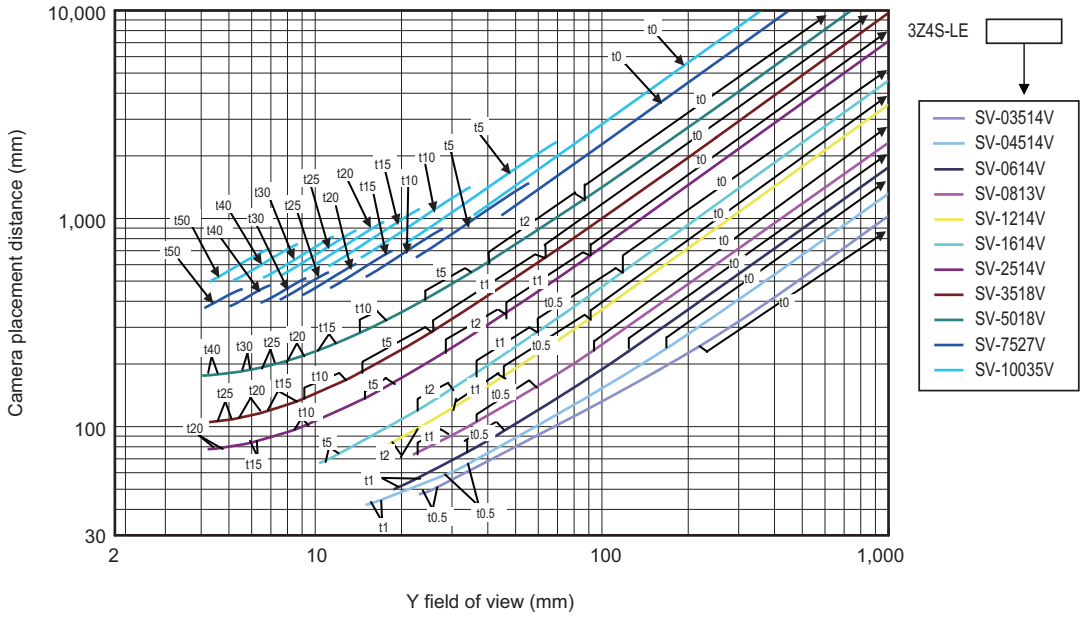
AND (Reference: ▶ Details (p.409))
NOT (Reference: ▶ Details (p.626))
UnitData (Reference: ▶ Details (p.778))
GetUnitData (Reference: ▶ Details (p.552))
OR (Reference: ▶ Details (p.635))

About Lenses

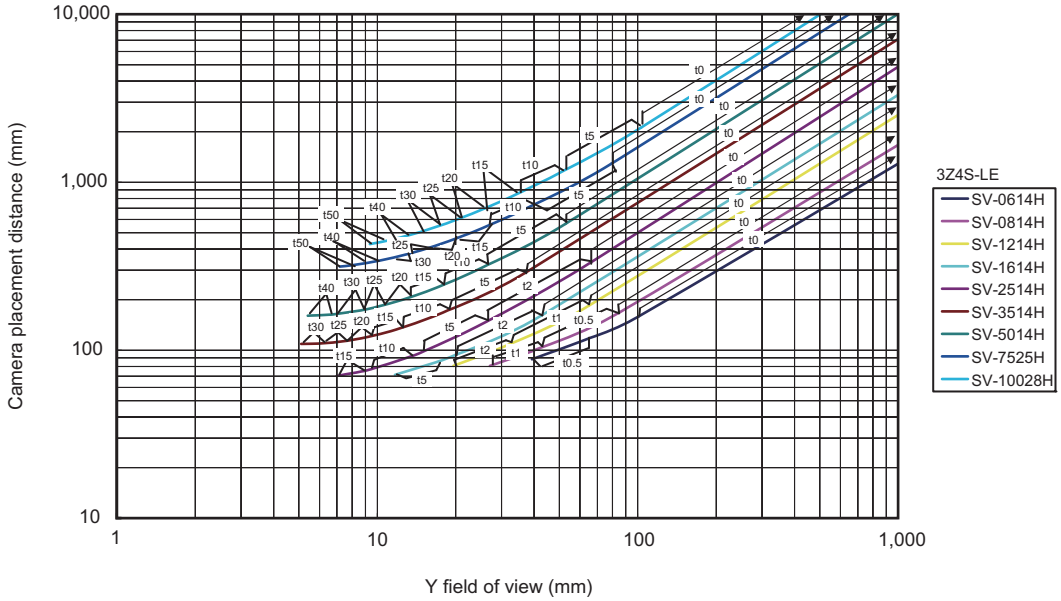
When using a camera (single), refer to the following tables to prepare the lens and extension tube. The lens may vary depending on the size of measurement objects and the camera setting distance.

Normal Lenses (Seven Types)

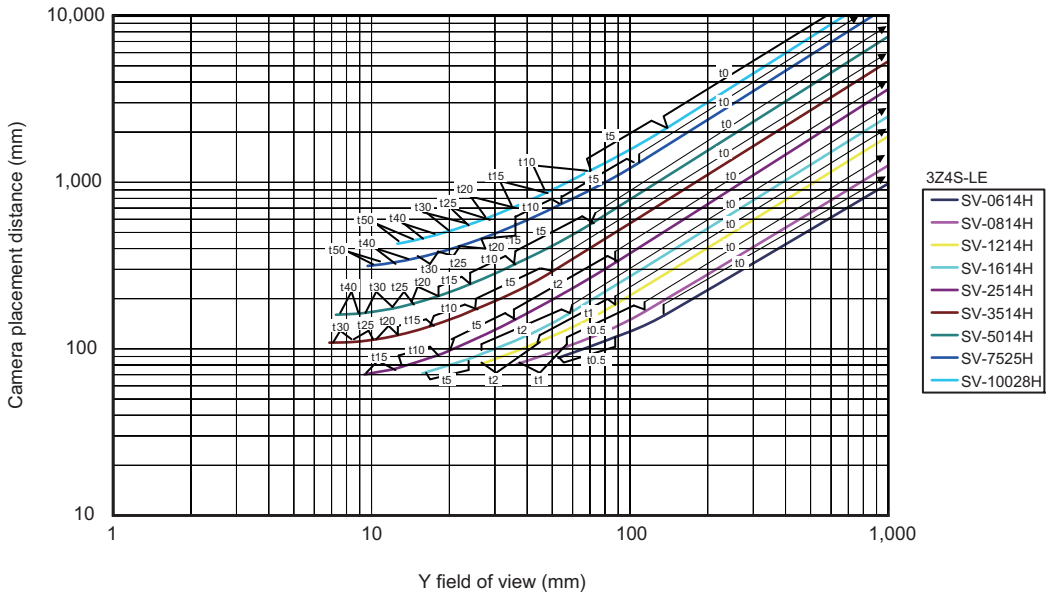
Digital Camera (Standalone) FZ-S□, FZ-SH□, or FH-S□ (Using 3Z4S-LE SV-V)



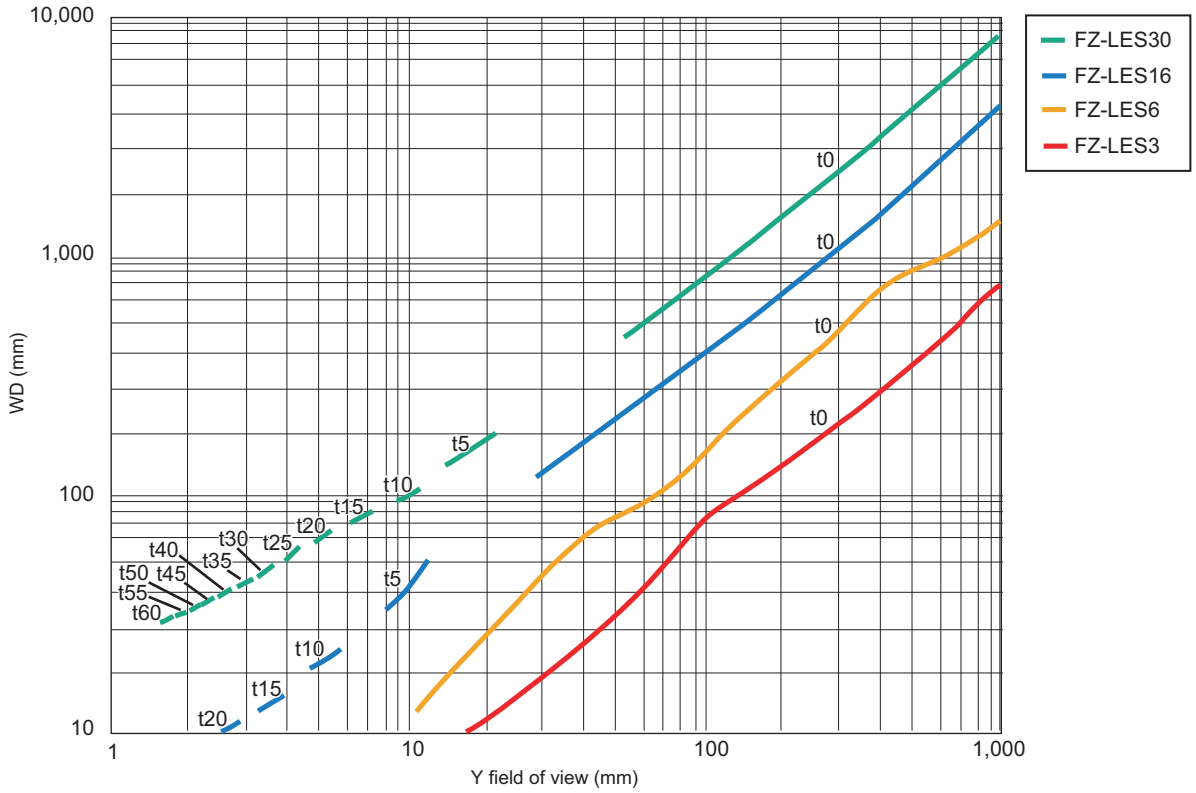
Digital CCD Camera (Standalone) FZ-S□2M (Using 3Z4S-LE SV-H)



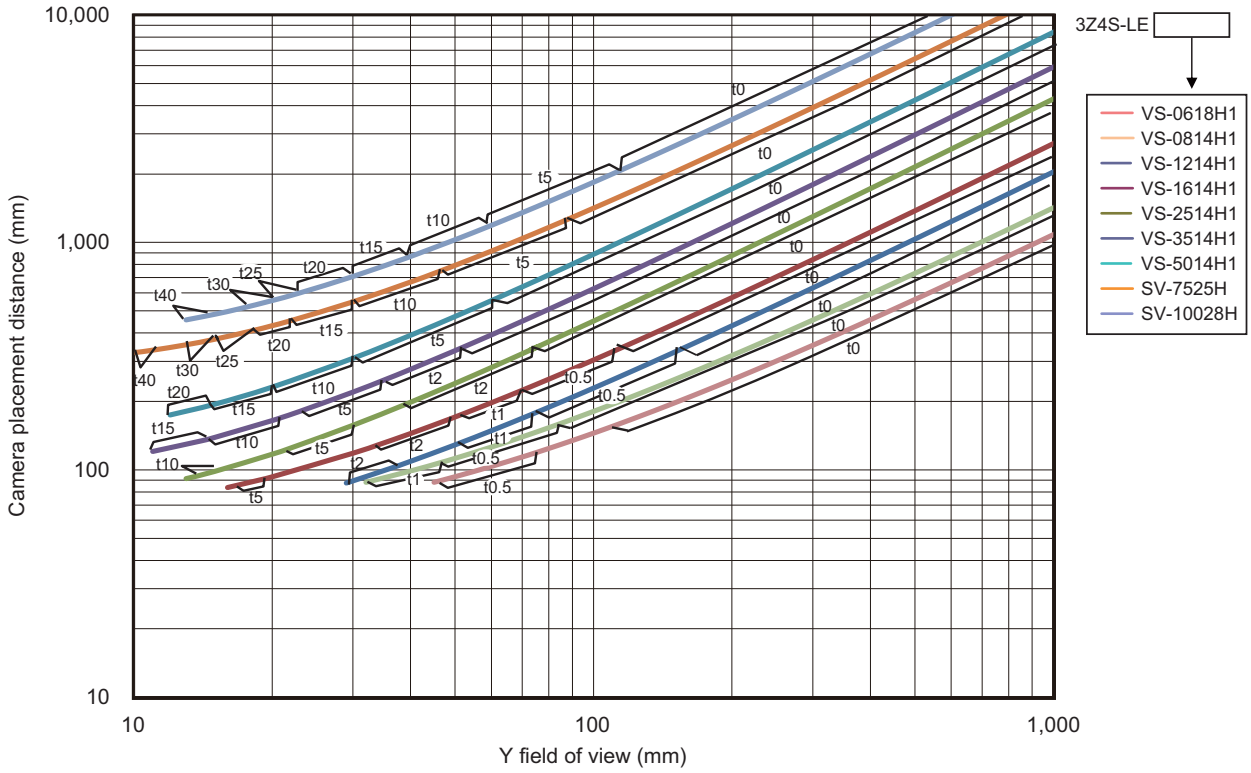
Digital CCD Camera (Standalone) FZ-S□5M2 (Using 3Z4S-LE SV-H)



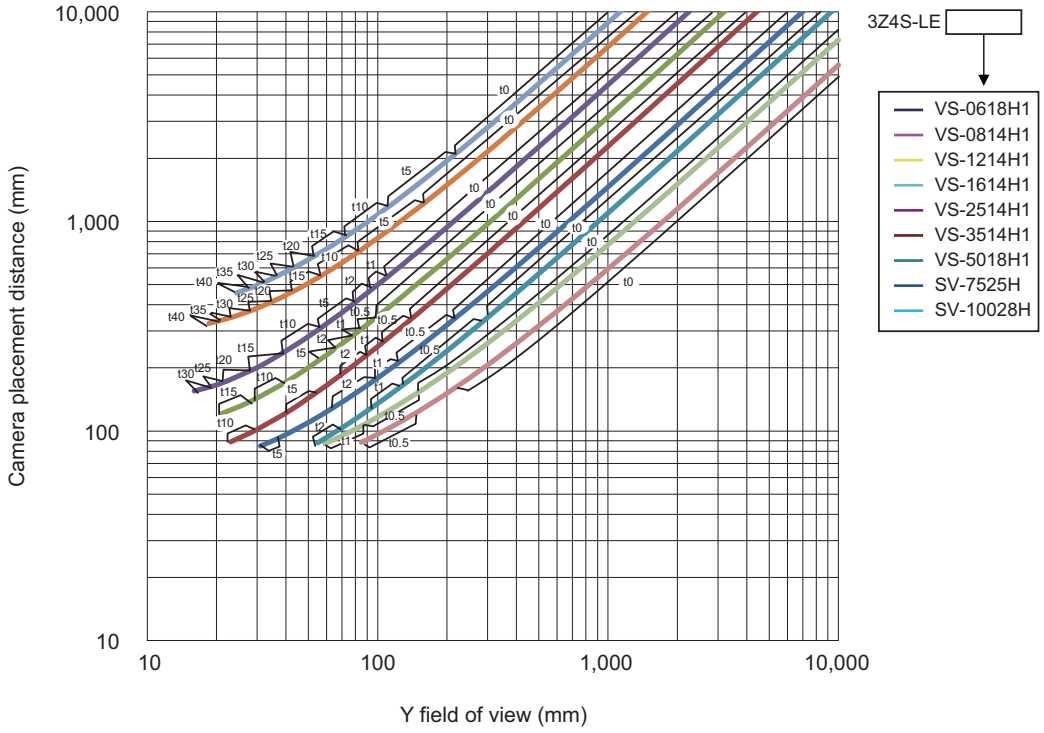
Small Digital CCD Camera FZ-SP□/FZ-SF□ (FZ-LE S Series)



High-speed CMOS Camera (Standalone) FH-S□02 (Using 3Z4S-LE VS-H1)



High-speed CMOS Camera (Standalone) FH-S□04 (Using 3Z4S-LE VS-H1)



High-speed CMOS Camera (Standalone) FH-S□12 (Using 3Z4S-LE VS-L/M42 Series)

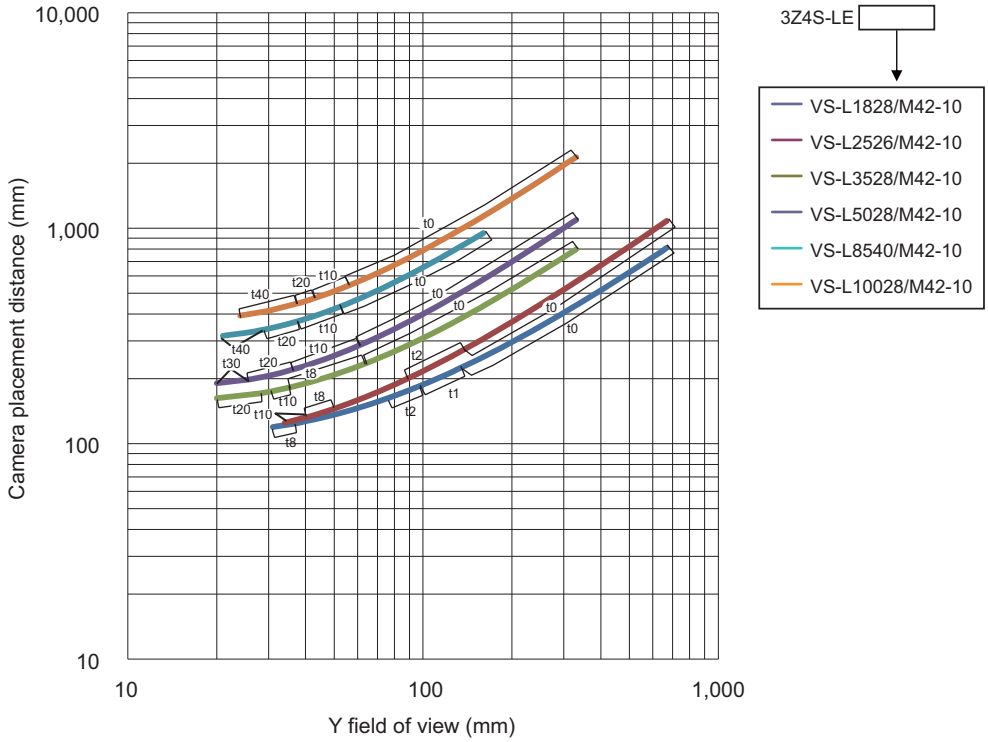
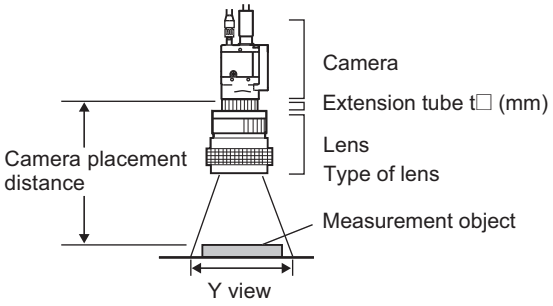


Diagram View

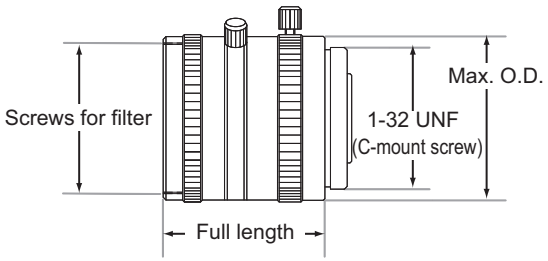
The horizontal axis of the diagram indicates the Y field of view (mm), and vertical axis indicates camera setting distance (mm) or working distance (mm). This diagram shows the relationship between the field of view of lenses and the setting distance for different types. Make sure to verify the lens type when checking the graph as the field of view value is different for each type. Points such as "t5.0" on the graph correspond to the thickness of the extension tube used. "t0" is used if an extension tube is not necessary, and "t5.0 is used if a 5 mm extension tube is used.

(Example)

If the field of view of measurement object is 40 mm, and a 3Z4S-LE SV-5018V Lens is used, the Camera setting distance is set at 500 mm, and a 5-mm extension tube is required.



Lens Types and Outside Diameters



Lens for 3Z4S-LE SV-V Series

Lens type	Focal distance	Aperture (F number)	Max. O.D.	Full length	Filter size
3Z4S-LE SV-0614V	6.20 mm	F1.4	28 mm dia.	30.0 mm	M27 P0.5
3Z4S-LE SV-0813V	8.05 mm	F1.3	28 mm dia.	34.0 mm	M25.5 P0.5
3Z4S-LE SV-1214V	12.43 mm	F1.4	29 mm dia.	29.5 mm	M27 P0.5
3Z4S-LE SV-1614V	16.34 mm	F1.4	29 mm dia.	24.0 mm	M27 P0.5
3Z4S-LE SV-2514V	25.17 mm	F1.4	29 mm dia.	24.5 mm	M27 P0.5
3Z4S-LE SV-3518V	34.75 mm	F1.8	29 mm dia.	33.5 mm (WD: ∞) to 37.5 mm (WD: 300 mm)	M27 P0.5
3Z4S-LE SV-5018V	47.97 mm	F1.8	32 mm dia.	37.0 mm (WD: ∞) to 39.4 mm (WD: 1000 mm)	M30.5 P0.5
3Z4S-LE SV-7527V	76.71 mm	F2.7	32 mm dia.	42.0 mm (WD: ∞) to 44.4 mm (WD: 1000 mm)	M30.5 P0.5
3Z4S-LE SV-10035V	95.4 mm	F3.5	32 mm dia.	43.9 mm (WD: ∞) to 46.3 mm (WD: 1000 mm)	M30.5 P0.5

High Resolution and Low Distortion Lens Model 3Z-4S-LE SV-H Series

Lens type	Focal distance	Aperture (F number)	Max. O.D.	Full length	Filter size
3Z4S-LE SV-0614H	6.1 mm	F1.4	42 mm dia.	57.5 mm	M40.5 P0.5
3Z4S-LE SV-0814H	8.0 mm	F1.4	39 mm dia.	52.5 mm	M35.5 P0.5
3Z4S-LE SV-1214H	12.3 mm	F1.4	30 mm dia.	51.0 mm	M27 P0.5
3Z4S-LE SV-1614H	16.2 mm	F1.4	30 mm dia.	47.5 mm	M27 P0.5
3Z4S-LE SV-2514H	25.0 mm	F1.4	30 mm dia.	36.0 mm	M27 P0.5
3Z4S-LE SV-3514H	35.01 mm	F1.4	44 mm dia.	45.5 mm	M35.5 P0.5
3Z4S-LE SV-5014H	50.0 mm	F1.4	44 mm dia.	57.5 mm	M40.5 P0.5
3Z4S-LE SV-7525H	75 mm	F2.5	36 mm dia.	49.5 mm (WD: ∞) to 54.6 mm (WD: 1200 mm)	M34.0 P0.5
3Z4S-LE SV-10028H	100 mm	F2.8	39 mm dia.	66.5 mm (WD: ∞) to 71.6 mm (WD: 2000 mm)	M37.5 P0.5

3Z4S-LE-VS-H1 Series

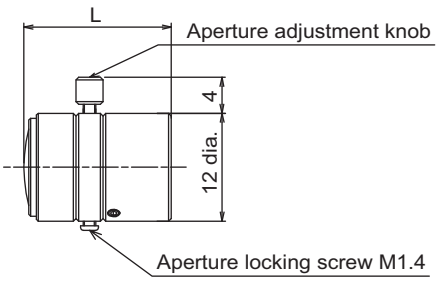
Lens type	Focal distance	Aperture (F number)	Max. O.D.	Full length	Filter size
3Z4S-LE VS-0618H1	6.23 mm	F1.8	64.5 mm dia.	57.2 mm	Filter is not supported for this lens.
3Z4S-LE VS-0814H1	8.24 mm	F1.4	57 mm dia.	59.0 mm	M55 P0.75
3Z4S-LE VS-1214H1	12.05 mm	F1.4	38 mm dia.	48.0 mm (WD: ∞) to 48.5 mm (WD: 300 mm)	M35.5 P0.5
3Z4S-LE VS-1614H1	16.06 mm	F1.4	38 mm dia.	42.5 mm (WD: ∞) to 43.3 mm (WD: 300 mm)	M30.5 P0.5
3Z4S-LE VS-2514H1	25.42 mm	F1.4	38 mm dia.	33.5 mm (WD: ∞) to 35.6 mm (WD: 300 mm)	M30.5 P0.5
3Z4S-LE VS-3514H1	35.10 mm	F1.4	38 mm dia.	35.0 mm (WD: ∞) to 39.1 mm (WD: 300 mm)	M30.5 P0.5
3Z4S-LE VS-5018H1	49.93 mm	F1.8	44 mm dia.	44.5 mm (WD: ∞) to 49.5 mm (WD: 500 mm)	M40.5 P0.5

3Z4S-LE VS-L/M42 Series M42 Mount Lens

Lens type	Focal distance	Aperture (F number)	Max. O.D.	Full length L	Filter size	Optical magnification
3Z4S-LE VS-L1828/M42-10	18 mm	F2.8 to F16	58.5 mm dia	94.0 mm	M55.0 P0.75	0.025x to 0.12x
3Z4S-LE VS-L2526/M42-10	25 mm	F2.6 to F16	58.5 mm dia	80.0 mm	M55.0 P0.75	0.025x to 0.12x
3Z4S-LE VS-L3528/M42-10	35 mm	F2.8 to F16	64.5 mm dia	108.0 mm	M62.0 P0.75	0.05x to 0.3x
3Z4S-LE VS-L5028/M42-10	50 mm	F2.8 to F16	66 mm dia	94.5 mm	M62.0 P0.75	0.05x to 0.3x
3Z4S-LE VS-L8540/M42-10	85 mm	F4.0 to F16	55.5 mm dia	129.5 mm	M52.0 P0.75	0.1x to 0.35x
3Z4S-LE VS-L10028/ M42-10	100 mm	F2.8 to F16	54 mm dia	134.5 mm	M52.0 P0.75	0.05x to 0.3x

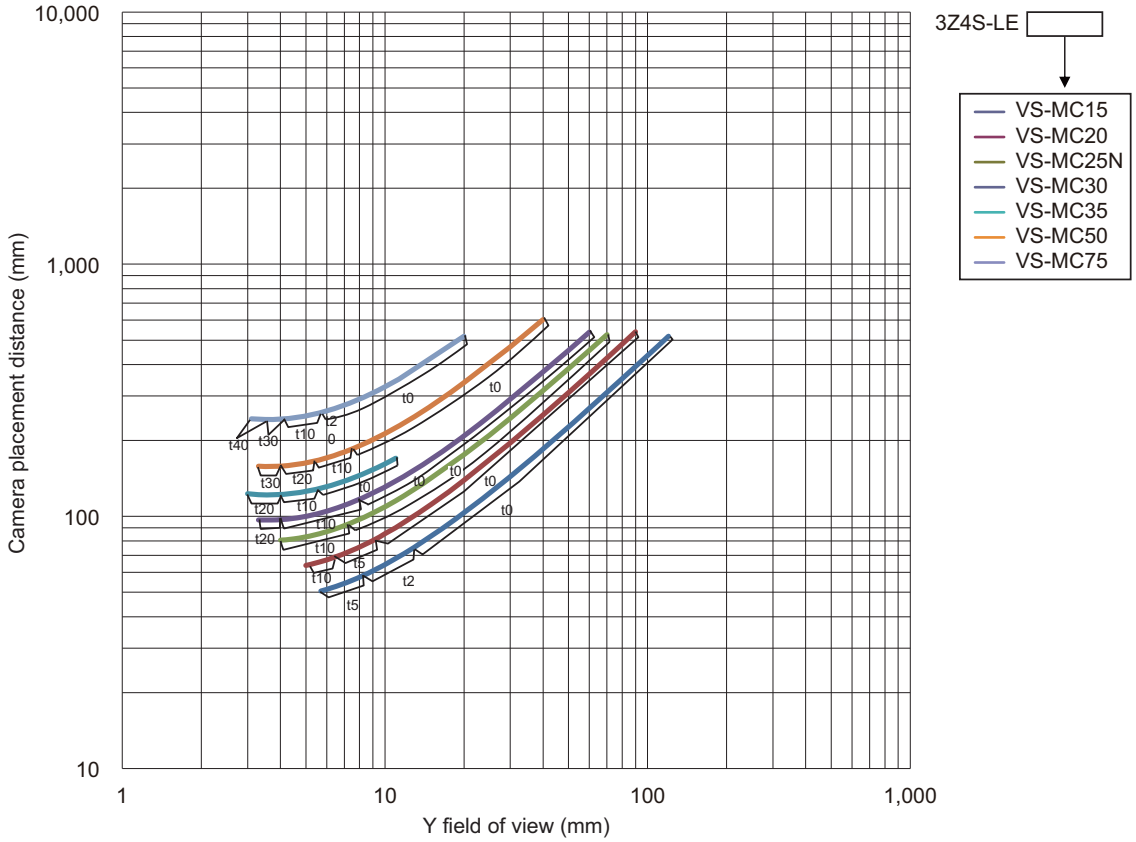
Lens for Small Digital Cameras FZ-LES□ Series

Lens type	Focal distance	Aperture (F number)	Max. O.D.	Full length L
FZ-LES3	3 mm	F2.0	12 mm dia.	16.4 mm
FZ-LES6	6 mm	F2.0	12 mm dia.	19.7 mm
FZ-LES16	16 mm	F3.4	12 mm dia.	23.1 mm
FZ-LES30	30 mm	F3.4	12 mm dia.	25.5 mm

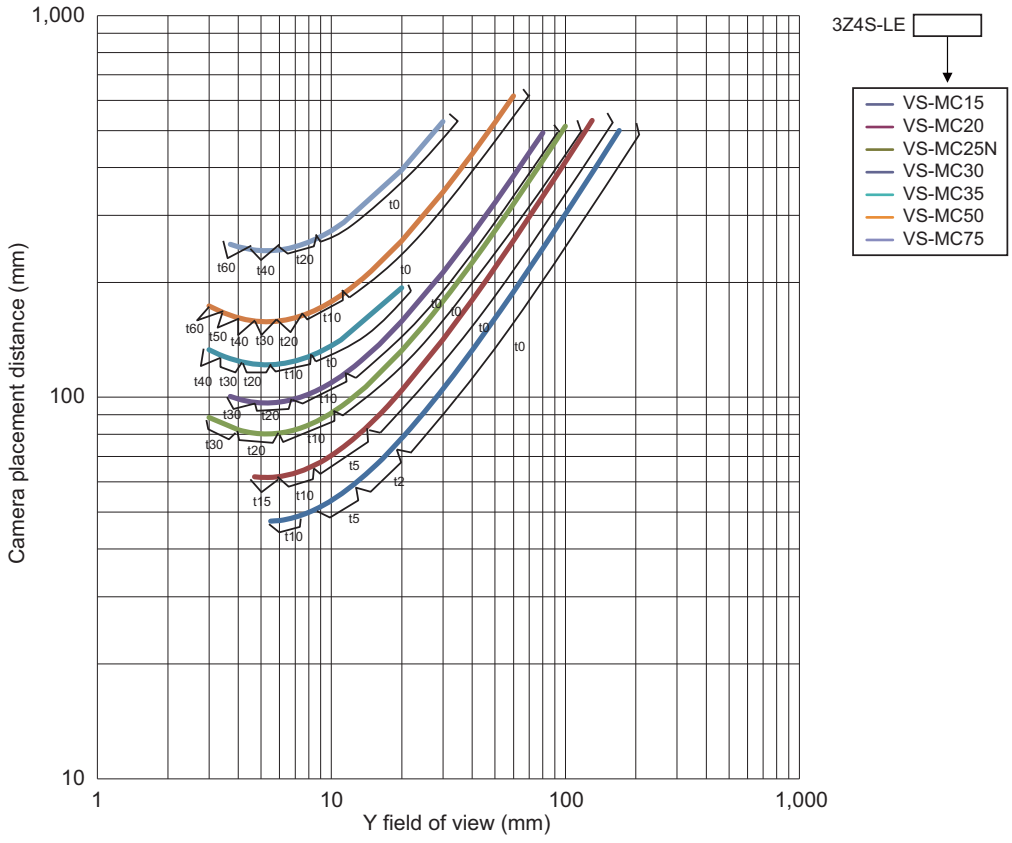


Vibration/Shock-resistance Lens (Six Types)

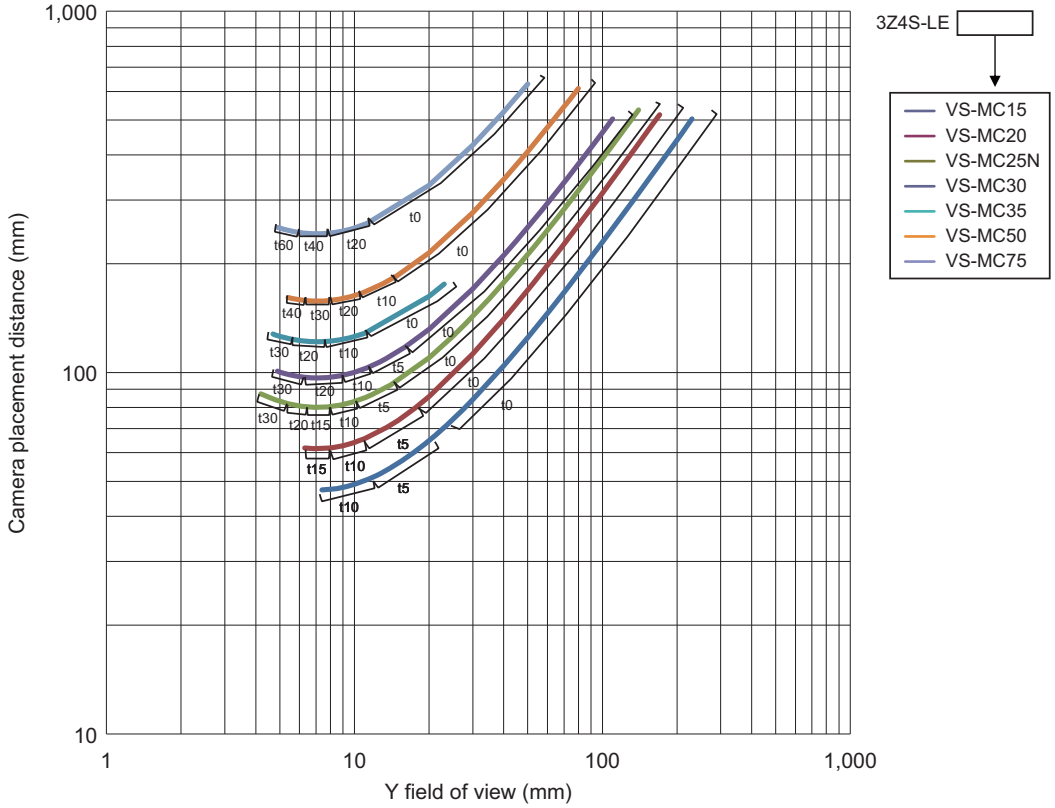
Digital Camera (Standalone) FZ-S□, FZ-SH□, or FH-S□ (Using 3Z4S-LE VS-MC)



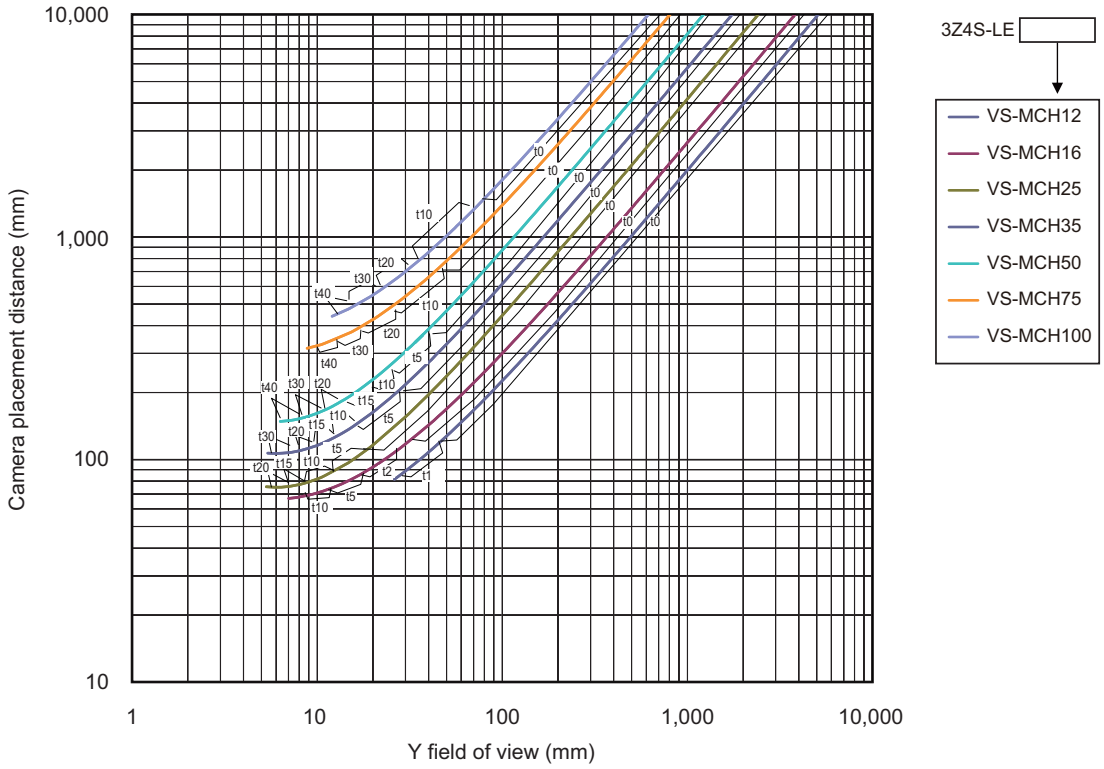
Digital Camera (Standalone) FZ-S□2M (Using 3Z4S-LE VS-MC)



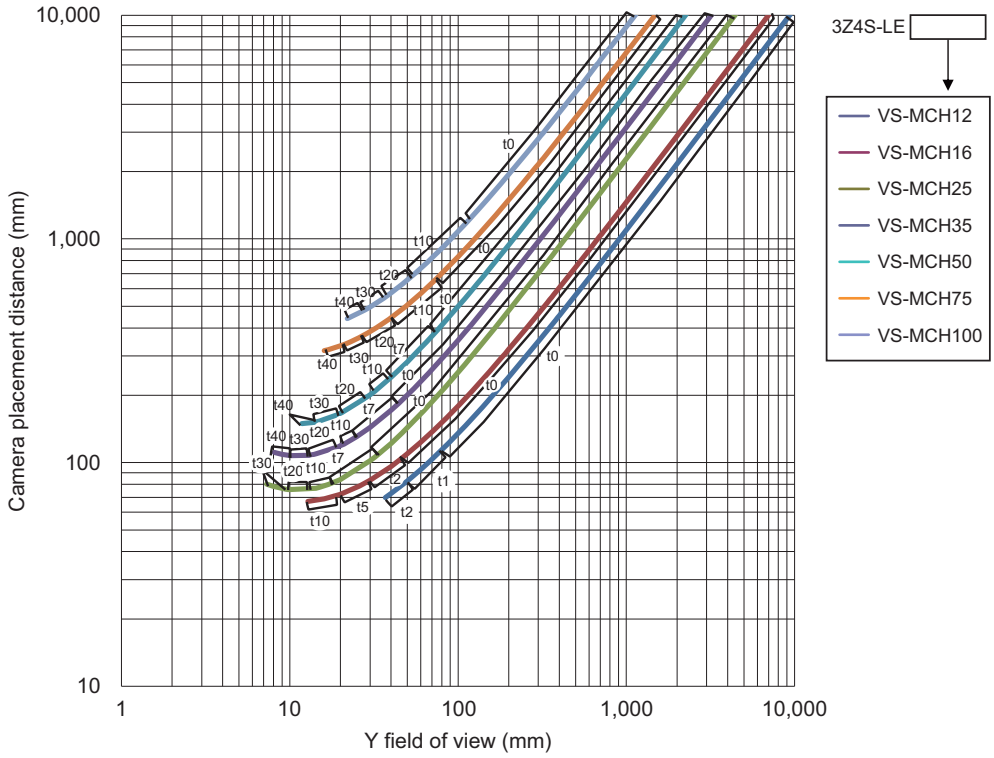
Digital CCD Camera (Standalone) FZ-S□5M2 (Using 3Z4S-LE VS-MC)



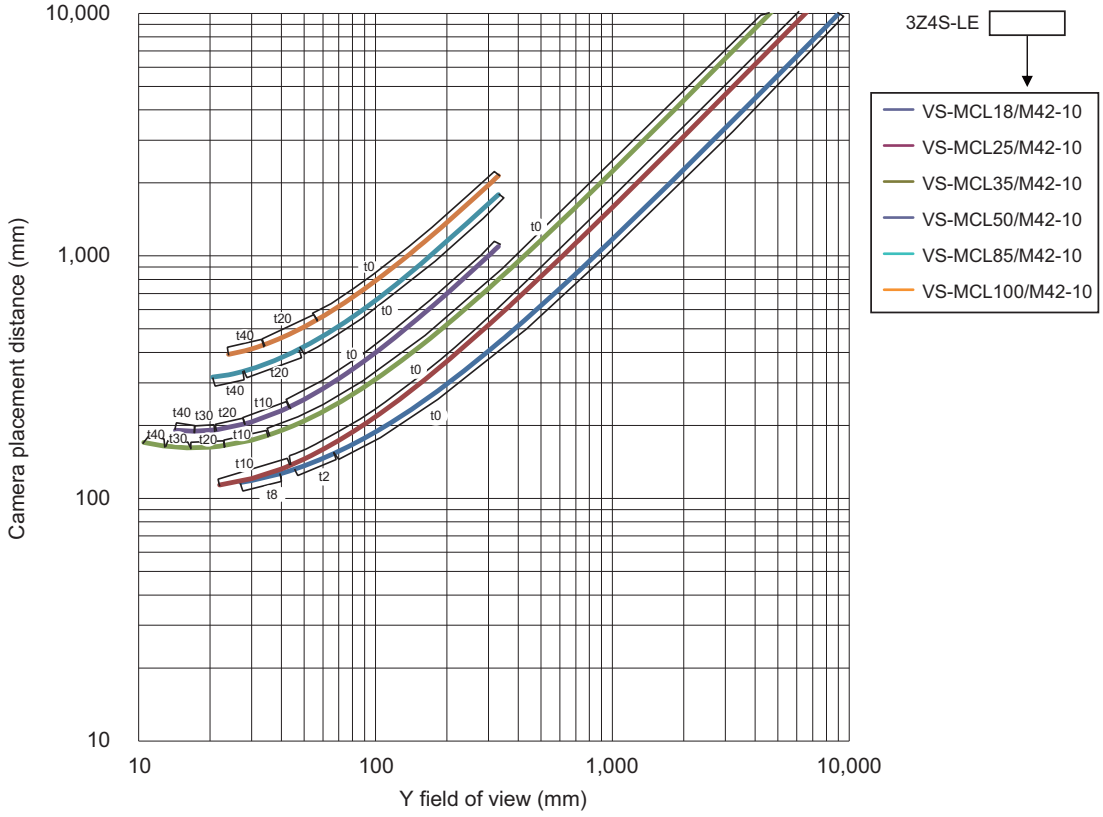
High-speed CMOS Camera (Standalone) FH-S□02 (Using 3Z4S-LE VS-MCH)



High-speed CMOS Camera (Standalone) FH-S□04 (Using 3Z4S-LE VS-MCH Series)



High-speed CMOS Camera (Standalone) FH-S□12 (Using 3Z4S-LE VS-MCL/M42 Series)



Lens Types and Outside Diameters

- **C-mount Lens 3Z4S-LE VS-MC Series for 2/3-inch Image Elements**

Lens type	Focal distance	Aperture (F No.)	Max. O.D.	Full length	Filter size	Optical magnification	Depth of field
3Z4S-LE VS-MC15	15 mm	F2	31 mm dia.	25.4 mm (0.03x) to 29.5 mm (0.30x)	M27.0 P0.5	0.03x	183.1 mm
						0.20x	4.8 mm
						0.30x	2.3 mm
3Z4S-LE VS-MC15-FN056		F5.6	31 mm dia.	25.4 mm (0.03x) to 29.5 mm (0.30x)	M27.0 P0.5	0.03x	512.7 mm
						0.20x	13.4 mm
						0.30x	6.5 mm
3Z4S-LE VS-MC15-FN080		F8	31 mm dia.	25.4 mm (0.03x) to 29.5 mm (0.30x)	M27.0 P0.5	0.03x	732.4 mm
						0.20x	19.2 mm
						0.30x	9.2 mm
3Z4S-LE VS-MC20	20 mm	F2	31 mm dia.	23.0 mm (0.04x) to 30.5 mm (0.40x)	M27.0 P0.5	0.04x	110.8 mm
						0.25x	3.4 mm
						0.40x	1.5 mm
3Z4S-LE VS-MC20-FN056		F5.6	31 mm dia.	23.0 mm (0.04x) to 30.5 mm (0.40x)	M27.0 P0.5	0.04x	291.2 mm
						0.25x	9.0 mm
						0.40x	3.9 mm
3Z4S-LE VS-MC20-FN080		F8	31 mm dia.	23.0 mm (0.04x) to 30.5 mm (0.40x)	M27.0 P0.5	0.04x	416 mm
						0.25x	12.8 mm
						0.40x	5.6 mm
3Z4S-LE VS-MC25	25 mm	F2	31 mm dia.	26.0 mm (0.05x) to 38.0 mm (0.50x)	M27.0 P0.5	0.05x	67.2 mm
						0.25x	3.2 mm
						0.50x	1.0 mm
3Z4S-LE VS-MC25-FN056		F5.6	31 mm dia.	26.0 mm (0.05x) to 38.0 mm (0.50x)	M27.0 P0.5	0.05x	188.2 mm
						0.25x	9.0 mm
						0.50x	2.7 mm
3Z4S-LE VS-MC25-FN080		F8	31 mm dia.	26.0 mm (0.05x) to 38.0 mm (0.50x)	M27.0 P0.5	0.05x	268.8 mm
						0.25x	12.8 mm
						0.50x	3.8 mm

Lens type	Focal distance	Aperture (F No.)	Max. O.D.	Full length	Filter size	Optical magnification	Depth of field
3Z4S-LE VS-MC30	30 mm	F2	31 mm dia.	24.0 mm (0.06x) to 35.7 mm (0.45x)	M27.0 P0.5	0.06x	47.1 mm
						0.15x	8.2 mm
						0.45x	1.1 mm
3Z4S-LE VS-MC30-FN056		F5.6	31 mm dia.	24.0 mm (0.06x) to 35.7 mm (0.45x)	M27.0 P0.5	0.06x	131.9 mm
						0.15x	22.9 mm
						0.45x	3.2 mm
3Z4S-LE VS-MC30-FN080		F8	31 mm dia.	24.0 mm (0.06x) to 35.7 mm (0.45x)	M27.0 P0.5	0.06x	188.4 mm
						0.15x	32.7 mm
						0.45x	4.6 mm
3Z4S-LE VS-MC35	35 mm	F1.9	31 mm dia.	32.0 mm (0.26x) to 45.7 mm (0.65x)	M27.0 P0.5	0.26x	2.8 mm
						0.30x	2.2 mm
						0.65x	0.6 mm
3Z4S-LE VS-MC35-FN056		F5.6	31 mm dia.	32.0 mm (0.26x) to 45.7 mm (0.65x)	M27.0 P0.5	0.26x	8.4 mm
						0.30x	6.5 mm
						0.65x	1.7 mm
3Z4S-LE VS-MC35-FN080		F8	31 mm dia.	32.0 mm (0.26x) to 45.7 mm (0.65x)	M27.0 P0.5	0.26x	11.9 mm
						0.30x	9.2 mm
						0.65x	2.5 mm
3Z4S-LE VS-MC50	50 mm	F2	31 mm dia.	44.5 mm (0.08x) to 63.9 mm (0.48x)	M27.0 P0.5	0.08x	33.8 mm
						0.20x	6.0 mm
						0.48x	1.3 mm
3Z4S-LE VS-MC50-FN056		F5.6	31 mm dia.	44.5 mm (0.08x) to 63.9 mm (0.48x)	M27.0 P0.5	0.08x	75.6 mm
						0.20x	13.4 mm
						0.48x	2.9 mm
3Z4S-LE VS-MC50-FN080		F8	31 mm dia.	44.5 mm (0.08x) to 63.9 mm (0.48x)	M27.0 P0.5	0.08x	108 mm
						0.20x	19.2 mm
						0.48x	4.1 mm
3Z4S-LE VS-MC75	75 mm	F3.8	31 mm dia.	70.0 mm (0.14x) to 105.5 mm (0.62x)	M27.0 P0.5	0.14x	17.7 mm
						0.20x	9.1 mm
						0.62x	1.3 mm
3Z4S-LE VS-MC75-FN056		F5.6	31 mm dia.	70.0 mm (0.14x) to 105.5 mm (0.62x)	M27.0 P0.5	0.14x	26.1 mm
						0.20x	13.4 mm
						0.62x	1.9 mm
3Z4S-LE VS-MC75-FN080		F8	31 mm dia.	70.0 mm (0.14x) to 105.5 mm (0.62x)	M27.0 P0.5	0.14x	37.2 mm
						0.20x	19.2 mm
						0.62x	2.7 mm

- **C-mount Lens 3Z4S-LE VS-MCH Series for 1-inch Image Elements**

Lens type	Focal distance	Aperture (F No.)	Max. O.D.	Full length	Filter size	Optical magnification	Depth of field
3Z4S-LE VS-MCH12	12 mm	F2	38 mm dia.	48.0 mm (0.025x) to 49.8 mm (0.15x)	M35.5 P0.5	0.025x	262.0 mm
						0.10x	17.6 mm
						0.15x	8.2 mm
3Z4S-LE VS-MCH12-FN056		F5.6	38 mm dia.	48.0 mm (0.025x) to 49.8 mm (0.15x)	M35.5 P0.5	0.025x	735.0 mm
						0.10x	49.3 mm
						0.15x	22.9 mm
3Z4S-LE VS-MCH12-FN080		F8	38 mm dia.	48.0 mm (0.025x) to 49.8 mm (0.15x)	M35.5 P0.5	0.025x	1050.0 mm
						0.10x	70.4 mm
						0.15x	32.7 mm
3Z4S-LE VS-MCH16	16 mm	F2	38 mm dia.	42.5 mm (0.025x) to 46.5 mm (0.25x)	M34 P0.5	0.025x	262.0 mm
						0.10x	17.6 mm
						0.25x	3.2 mm
3Z4S-LE VS-MCH16-FN056		F5.6	38 mm dia.	42.5 mm (0.025x) to 46.5 mm (0.25x)	M34 P0.5	0.025x	735.0 mm
						0.10x	49.3 mm
						0.25x	9.0 mm
3Z4S-LE VS-MCH16-FN080		F8	38 mm dia.	42.5 mm (0.025x) to 46.5 mm (0.25x)	M34 P0.5	0.025x	1050.0 mm
						0.10x	70.4 mm
						0.25x	12.8 mm
3Z4S-LE VS-MCH25	25 mm	F2	38 mm dia.	33.5 mm (0.025x) to 42.4 mm (0.35x)	M34 P0.5	0.025x	262.0 mm
						0.10x	17.6 mm
						0.35x	1.8 mm
3Z4S-LE VS-MCH25-FN056		F5.6	38 mm dia.	33.5 mm (0.025x) to 42.4 mm (0.35x)	M34 P0.5	0.025x	735.0 mm
						0.10x	49.3 mm
						0.35x	4.9 mm
3Z4S-LE VS-MCH25-FN080		F8	38 mm dia.	33.5 mm (0.025x) to 42.4 mm (0.35x)	M34 P0.5	0.025x	1050.0 mm
						0.10x	70.4 mm
						0.35x	7.1 mm
3Z4S-LE VS-MCH35	35 mm	F2	38 mm dia.	35.0 mm (0.025x) to 43.8 mm (0.25x)	M34 P0.5	0.025x	262.0 mm
						0.10x	17.6 mm
						0.25x	3.2 mm
3Z4S-LE VS-MCH35-FN056		F5.6	38 mm dia.	35.0 mm (0.025x) to 43.8 mm (0.25x)	M34 P0.5	0.025x	735.0 mm
						0.10x	49.3 mm
						0.25x	9.0 mm
3Z4S-LE VS-MCH35-FN080		F8	38 mm dia.	35.0 mm (0.025x) to 43.8 mm (0.25x)	M34 P0.5	0.025x	1050.0 mm
						0.10x	70.4 mm
						0.25x	12.8 mm

Lens type	Focal distance	Aperture (F No.)	Max. O.D.	Full length	Filter size	Optical magnification	Depth of field
3Z4S-LE VS-MCH50	50 mm	F2	43 mm dia.	44.5 mm (0.025x) to 52.0 mm (0.15x)	M40.5 P0.5	0.025x	262.0 mm
						0.10x	17.6 mm
						0.15x	8.2 mm
3Z4S-LE VS-MCH50-FN056		F5.6	43 mm dia.	44.5 mm (0.025x) to 52.0 mm (0.15x)	M40.5 P0.5	0.025x	735.0 mm
						0.10x	49.3 mm
						0.15x	22.9 mm
3Z4S-LE VS-MCH50-FN080		F8	43 mm dia.	44.5 mm (0.025x) to 52.0 mm (0.15x)	M40.5 P0.5	0.025x	1050.0 mm
						0.10x	70.4 mm
						0.15x	32.7 mm
3Z4S-LE VS-MCH75	75 mm	F2.5	38 mm dia.	49.5 mm (0.025x) to 60.7 mm (0.15x)	M34 P0.5	0.025x	262.0 mm
						0.10x	17.6 mm
						0.15x	8.2 mm
3Z4S-LE VS-MCH75-FN056		F5.6	38 mm dia.	49.5 mm (0.025x) to 60.7 mm (0.15x)	M34 P0.5	0.025x	735.0 mm
						0.10x	49.3 mm
						0.15x	22.9 mm
3Z4S-LE VS-MCH75-FN080		F8	38 mm dia.	49.5 mm (0.025x) to 60.7 mm (0.15x)	M34 P0.5	0.025x	1050.0 mm
						0.10x	70.4 mm
						0.15x	32.7 mm
3Z4S-LE VS-MCH100	100 mm	F2.8	40 mm dia.	66.5 mm (0.025x) to 76.3 mm (0.10x)	M37.5 P0.5	0.025x	262.0 mm
						0.05x	94.1 mm
						0.10x	17.6 mm
3Z4S-LE VS-MCH100-FN056		F5.6	40 mm dia.	66.5 mm (0.025x) to 76.3 mm (0.10x)	M37.5 P0.5	0.025x	735.0 mm
						0.05x	188.2 mm
						0.10x	49.3 mm
3Z4S-LE VS-MCH100-FN080		F8	40 mm dia.	66.5 mm (0.025x) to 76.3 mm (0.10x)	M37.5 P0.5	0.025x	1050.0 mm
						0.05x	268.8 mm
						0.10x	70.4 mm

- **M42 mount Lens 3Z4S-LE VS-MCL/M42 Series for 1.8-inch Image Elements**

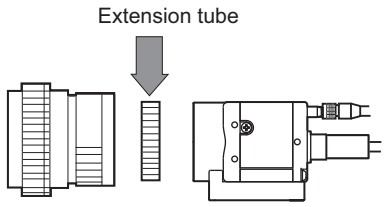
Lens type	Focal distance	Aperture (F No.)	Max. O.D.	Full length	Filter size	Optical magnification	Depth of field
3Z4S-LE VS-MCL18/M42-10	18 mm	F2.8	52 mm dia.	91.5 mm (0.025x) to 96.1 mm (0.25x)	M46 P0.75	0.025x	367.0 mm
						0.10x	24.6 mm
						0.25x	4.5 mm
3Z4S-LE VS-MCL18-FN056/M42-10		F5.6	52 mm dia.	91.5 mm (0.025x) to 96.1 mm (0.25x)	M46 P0.75	0.025x	735.0 mm
						0.10x	49.3 mm
						0.25x	9.0 mm
3Z4S-LE VS-MCL18-FN080/M42-10		F8	52 mm dia.	91.5 mm (0.025x) to 96.1 mm (0.25x)	M46 P0.75	0.025x	1050.0 mm
						0.10x	70.4 mm
						0.25x	12.8 mm
3Z4S-LE VS-MCL25/M42-10	25 mm	F2.6	52 mm dia.	72.0 mm (0.025x) to 82.3 mm (0.40x)	M46 P0.75	0.025x	341.0 mm
						0.10x	22.9 mm
						0.40x	1.8 mm
3Z4S-LE VS-MCL25-FN056/M42-10		F5.6	52 mm dia.	72.0 mm (0.025x) to 82.3 mm (0.40x)	M46 P0.75	0.025x	735.0 mm
						0.10x	49.3 mm
						0.40x	3.9 mm
3Z4S-LE VS-MCL25-FN080/M42-10		F8	52 mm dia.	72.0 mm (0.025x) to 82.3 mm (0.40x)	M46 P0.75	0.025x	1050.0 mm
						0.10x	70.4 mm
						0.40x	5.6 mm
3Z4S-LE VS-MCL35/M42-10	35 mm	F2.8	55 mm dia.	99.5 mm (0.025x) to 117.6 mm (0.50x)	M52 P0.75	0.025x	367.0 mm
						0.10x	24.6 mm
						0.50x	1.3 mm
3Z4S-LE VS-MCL35-FN056/M42-10		F5.6	55 mm dia.	99.5 mm (0.025x) to 117.6 mm (0.50x)	M52 P0.75	0.025x	735.0 mm
						0.10x	49.3 mm
						0.50x	2.7 mm
3Z4S-LE VS-MCL35-FN080/M42-10		F8	55 mm dia.	99.5 mm (0.025x) to 117.6 mm (0.50x)	M52 P0.75	0.025x	1050.0 mm
						0.10x	70.4 mm
						0.50x	3.8 mm
3Z4S-LE VS-MCL50/M42-10	50 mm	F2.8	52 mm dia.	64.0 mm (0.05x) to 82.0 mm (0.40x)	M46 P0.75	0.05x	94.1 mm
						0.10x	24.6 mm
						0.40x	2.0 mm
3Z4S-LE VS-MCL50-FN056/M42-10		F5.6	52 mm dia.	64.0 mm (0.05x) to 82.0 mm (0.40x)	M46 P0.75	0.05x	188.0 mm
						0.10x	49.3 mm
						0.40x	3.9 mm
3Z4S-LE VS-MCL50-FN080/M42-10		F8	52 mm dia.	64.0 mm (0.05x) to 82.0 mm (0.40x)	M46 P0.75	0.05x	269.0 mm
						0.10x	70.4 mm
						0.40x	5.6 mm

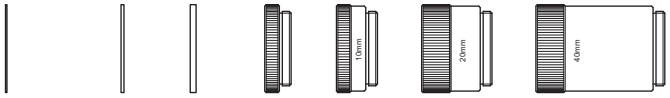
Lens type	Focal distance	Aperture (F No.)	Max. O.D.	Full length	Filter size	Optical magnification	Depth of field
3Z4S-LE VS-MCL85/M42-10	85 mm	F4	52 mm dia.	105.0 mm (0.05x) to 130.2 mm (0.35x)	M46 P0.75	0.05x	134.0 mm
						0.10x	35.2 mm
						0.35x	3.5 mm
3Z4S-LE VS-MCL85-FN056/M42-10		F5.6	52 mm dia.	105.0 mm (0.05x) to 130.2 mm (0.35x)	M46 P0.75	0.05x	188.0 mm
						0.10x	49.3 mm
						0.35x	4.9 mm
3Z4S-LE VS-MCL85-FN080/M42-10		F8	52 mm dia.	105.0 mm (0.05x) to 130.2 mm (0.35x)	M46 P0.75	0.05x	269.0 mm
						0.10x	70.4 mm
						0.35x	7.1 mm
3Z4S-LE VS-MCL100/M42-10	100 mm	F2.5	52 mm dia.	110.0 mm (0.05x) to 135.0 mm (0.30x)	M46 P0.75	0.05x	94.1 mm
						0.10x	24.6 mm
						0.30x	3.2 mm
3Z4S-LE VS-MCL100-FN056/M42-10		F5.6	52 mm dia.	110.0 mm (0.05x) to 135.0 mm (0.30x)	M46 P0.75	0.05x	188.0 mm
						0.10x	49.3 mm
						0.30x	6.5 mm
3Z4S-LE VS-MCL100-FN080/M42-10		F8	52 mm dia.	110.0 mm (0.05x) to 135.0 mm (0.30x)	M46 P0.75	0.05x	269.0 mm
						0.10x	70.4 mm
						0.30x	9.2 mm

Extension Tube

An extension tube, which is installed between lens and camera, is used to adjust focal distance. Different combinations of the 7 tubes can be used to achieve any desired thickness.

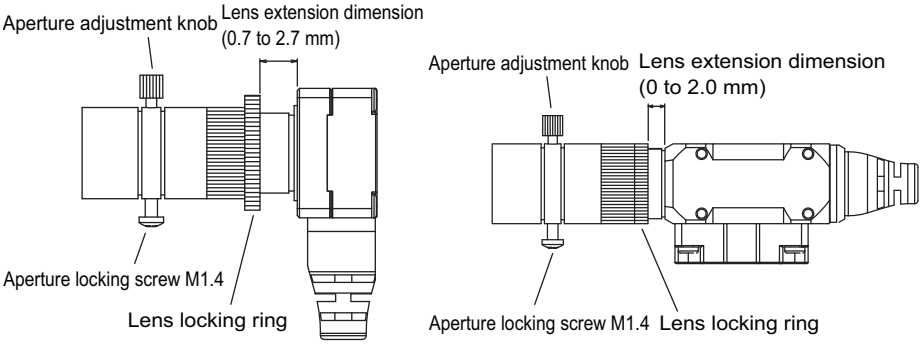
C-mount extension tube

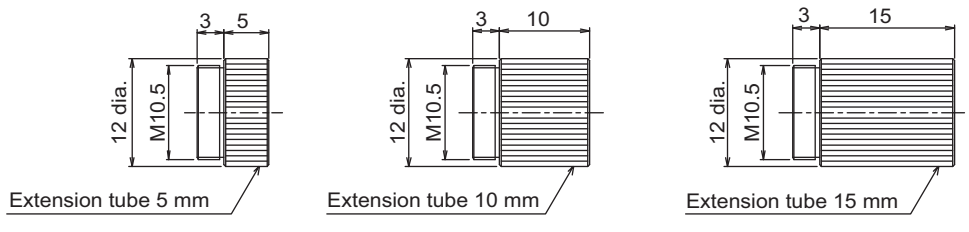


Type	Max. O.D.	Thickness
3Z4S-LE SV-EXR	31 dia.	7 sets Thickness: 0.5 mm 1 mm 2 mm 5 mm 10 mm 20 mm 40 mm 

Type	Max. O.D.	Thickness
3Z4S-LE VS-EXR/M42	47.5 dia.	5 sets Thickness: 1 mm 2mm 8mm 10mm 20mm 

Extension Tubes for Small Digital Cameras FZ-LESR



Max. O.D.	Thickness
12 dia.	 3-piece set

Note

- Do not overlap 2 or more extension tubes (0.5 mm, 1.0 mm, 2.0 mm). Since they are installed between the lens and the threaded section of another extension tube, overlapping them will cause unstable fixation.
- When using an extension tube larger than 30 mm, reinforcement is necessary to avoid being affected by vibration.
- When using the extension tube, please test in the your actual environment before using.

Error Messages and Troubleshooting

This section lists error messages that display on the screen and solutions.

ERR When a message with this symbol is displayed, the ERROR signal on the parallel interface is on.

Error message	Troubleshooting
No value set. Be sure to set up value.	A value must be set for this item. Please set up value.
• Invalid system disk Replace the disk, and then press any key. • Remove disks or other media. Press any key to restart.	An error occurred in USB memory. End the Controller and turn OFF the power supply. Then back up the data from the USB memory device that you removed from the Controller and then physically format it on your computer with a physical formatting tool. If the problem cannot be solved by physically reformatting the USB memory device, it may be damaged.
Error of the expression.	Do the following errors exist in settings for an expression set up? • Different number of open/close parentheses • TJG/unit/value/function/operator/comma is lined next to each other. • Operator is set at beginning or end of expression • The number of function operands is insufficient or there are more than is needed
Cannot add object. Out of range of specified characters of expression.	Incompliant to the conditions of expression. Please check the content. Reference: ▶ *Calculation-Settings (Calculation)* in the *Vision System FH/FZ5 Series Processing Item Function Reference Manual* (Cat No. Z341).
The problem occurred in the camera connection. ERR	Is the camera cable connected to the controller? Is camera cable disconnected? Do [Select camera] of [Camera Image Input] and [Camera Switching] have any errors? Please check the contents of [Camera setting], then switch off and restart. Reference: ▶ *Camera Image Input-Camera Settings* in the *Vision System FH/FZ5 Series Processing Item Function Reference Manual* (Cat No. Z341).
Failed to start the window. Memory is insufficient.	Because memory available for use was insufficient, memory needed to open the window could not be secured. Check the flow by adjusting the measurement region of each processing unit and the sizes of the registered models. • [Shape search] If shape search+ setup window is opened while the RAMDisk memory is insufficient for image logging, etc. on the RAMDisk, a warning message is displayed. In the case that an image logging file is stored in a RAMDisk, please move the content on the RAMDisk to a USBDisk to open up memory and reopen the set up window.
Warning range shall be set within the range from judgement lower to upper limit.	Please redefine the upper and lower warning range limits such that they are within the range between the lower and upper evaluation limits. Reference: ▶ *Trend Monitor-Judgement Conditions (Trend Monitor)* in the *Vision System FH/FZ5 Series Processing Item Function Reference Manual* (Cat No. Z341).
The "Search" measurement result may be "NG (Insufficient memory)" with the current setting.	Insufficient memory may occur during measurement. Reset the model parameters. • [Search] Reference: ▶ *Search - Judgement is NG (Insufficient Memory)* in the *Vision System FH/FZ5 Series Processing Item Function Reference Manual* (Cat No. Z341). • [Flexible search] Reference: ▶ *Flexible search - Judgement is NG (Insufficient Memory)* in the *Vision System FH/FZ5 Series Processing Item Function Reference Manual* (Cat No. Z341). • [Classification] Reference: ▶ *Classification - Judgement is NG (Insufficient Memory)* in the *Vision System FH/FZ5 Series Processing Item Function Reference Manual* (Cat No. Z341).
It failed in preservation because the free space of the flash memory was insufficient.	The scene group data size may be too big. Use external memory and save the data to a scene group other than group 0 or change the measurement flow.

Error message	Troubleshooting
The problem occurred in the system. 	This is displayed when the significant abnormality occurs in the controller system. Please contact one of our branches or regional offices.
The problem occurred in system date. The system battery is insufficient. 	Low battery (for the backup of date and time data) level. The batteries need to be replaced. Please return old batteries to one of our branches or regional offices.
Failed to switch scene.	Is a USB disk set to the controller? A USB disk is needed to read scenes after scene group 1.
Failed to switch scene group or save scene group on switch.	The cause of the failure to switch or to save when switching may be that the USBDisk was removed from the controller. Set a USBDisk to the controller and try again.
Failed to clear scene group.	Confirm that a USBDisk is set to the controller and try again.
Failed to load scene group. Data is corrupted, or memory is insufficient. Scene group data starts with initialized status.	The following causes are suspected: • The power may have been cut off during the last data save. • Because the operation mode was switched, the required memory size may have increased and memory became insufficient.
The camera connected is not the same as the one used for the last save. Please check.	Check if the camera is connected correctly. This message is displayed when the system and the scene group 0 data edited using the simulation software are loaded and the unit is restarted. Save the setting to the controller and then restart.
Cannot read selected file. Confirm selected file once again.	Check the following points. • Have you removed the USBDisk from the controller after selecting files on the USBDisk? • Have you deleted the selected file? (Such as the case when you selected a file on the RAMDisk and the file was deleted via FTP.) In addition, the selected file may be corrupt.
The communication time-out is occurred. 	Switch off controller, verify the following contents and then restart. • Is cable connected correctly? • Does it comply with communication specifications of external devices? • Are external devices functioning normally? If error is not resolved after confirmation, the controller may be damaged. Please discuss this with one of our branches or regional offices. Reference: ▶ *Non-procedure Communications - Communications Specifications Settings* in the *Vision System FH/FZ5 Series User's Manual for Communications Settings* (Cat No. Z342)
Failed to transfer data. The free capacity of RAMDisk may insufficient. Increase the free capacity of RAMDisk and then perform this operation again.	Clean up the content of RAMDisk to increase free space. If this error occurs even though enough free space is available, scene group data may be too large. Review the inspection flow.
Error in input range. Please input using the correct range.	Please verify range for setting and set up again.
File name contains invalid character.	Please confirm that characters such as \ / , : ; * ? " < > |& . SPC (space) are not included in the file name.
Failed to save file. Please check.	Check the following points. • Is the memory of the save destination sufficient? • For cases where the save destination is the folder in the USBDisk, is the USBDisk connected to the controller? Or, the controller may not have detected the USBDisk.

Error message	Troubleshooting
Fan/voltage error. ERR	Switch off controller, and verify if fan is affected. If error message is still shown after restart, the controller may be damaged. Please contact one of our branches or regional offices.
Lighting connection has a problem.	Check the following points. • Is there a light with a power consumption higher than 7.5 W connected without an external power supply? • Is there a light with a power consumption higher than 15 W connected with an external power supply?
The error concerning parallel interface was detected. Turn OFF the power and check the connection state of the parallel interface.	Turn OFF the power and check the connection state of the parallel interface.
ProcItem error xxxx: xxxxerror	This is displayed when the application software detects an abnormality. Please contact one of our branches or regional offices as there may be a software trouble.
Failed to paste. Please check the save source or the save destination.	No more scene can be added due to lack of memory.*1 Review the inspection flow and reduce the memory consumption or switch to another scene group. *1: The display varies depending on the controller. The remaining capacity of the application is not sufficient for the FH/FZ5-□□ series. The remaining data memory capacity is not sufficient for the FZ5-L3□□/FZ5-6□□ series.
A camera outside the guarantee is connected.	A camera outside the guarantee is connected to the controller currently being used.
Destination folder is not found. Please check.	Check the following points. • Have you deleted the destination folder? • For cases where he save destination is the folder in the USBDisk, is the USBDisk connected to the controller? Or, has the USBDisk been detected?
Failed to save the scene group data. The current scene group data in the flash memory is discarded after restarting. Recover or avoid by the following ways.	• Save the data to click "Data Save" after increasing free space of the flash memory. Deleting other scene data, deleting processing units, or setting measurement regions or model regions as small as possible is effective. • Change the save destination for the scene group data to the external memory device. Reference: ▶ Saving Scene Groups in External Memory (p.168) • Save current scene group data in the external memory device. Reference: ▶ Saving Settings Data to the Controller RAM Disk or an External Memory Device (p.258)
PLC link error ERR	PLC link cannot be established. Check the following points. • Are the FZ communication settings correct? • Are the PLC communication settings correct? • Is cable connected correctly?
Failed to register model	Search, classification, flexible search, shape search+, shape search II Please register higher-contrast images as models. If the model is registered at full screen size, re-register at a smaller size.
	For fine matching For the fine matching, the two pixels at the edge of the image cannot be registered as the model.
Available memory is insufficient. It may cause insufficient memory depending on the setting.	Available application memory is low. Memory may become insufficient during operation or an error may occur when the operation mode is switched. Review the inspection flow and reduce the memory consumption.

Error message	Troubleshooting
The free capacity of RAMDisk is insufficient. If nothing is done, measurement cannot be performed correctly. Increase free capacity in RAMDisk.	Clean up the content of RAMDisk to increase free space. If this error occurs even though enough free space is available, scene group data may be too large. Review the inspection flow.
Region size exceeds. Please narrow region.	There are restrictions as to what can be set up based on the camera and processing items being used. Please adjust the region size so that the region is not too large.
Logging error ERR	Image logging failed due to insufficient memory at the save destination. The error message disappears after 10 s. Please delete unneeded files in the save destination or prepare a new USBDisk.
The Logon to Network Server screen is displayed.	The connection was cut off based on the idling/session time set on the network logging destination PC. At the command prompt on the PC, enter net config server /autodisconnect: -1 and disable the automatic disconnection setting.

Although an error message such as the following, is not displayed, the ERROR signal and the ERR indicator will turn ON.

Cause	Troubleshooting
The STEP signal was input when the READY signal was turned OFF.	• Input the STEP signal after the READY signal turns ON. • Noise is superimposed onto the STEP signal. Keep PLC and FH/FZ5 away from the noise source.
A non-existing parallel command was entered.	• Please enter a correct parallel command.
The parallel scene group switch command was executed when no USB memory was installed. (FZ5-L3□□/FZ5-6□□ only)	• Switch the scene group with the USB memory installed.

During Start-up

POWER LCD is not lit

- **Is the power supply connected correctly?**
- **Is the supply voltage low (24 V DC +10%, -15%)?**

Nothing is displayed on the monitor

- **Is the monitor ON?**
- **Is the monitor cable connected correctly?**
- **Has the monitor failed?**
- **Is the power capacity enough (LCD monitor)?**
- **Have you turned off the LCD? If you have, the monitor recovery will occur if you tap on the bottom of the monitor.**

FH/FZ5 RUN and ADJUST windows are not displayed.

Camera connection:

- **When the camera connection is checked and initialized during start-up, the system will not start unless there is a response from the camera side. Check if the system starts with the camera cable disconnected.**

Data corruption:

- **The scene group data and system data saved to the controller have been corrupted.**
  In some cases when the power is shut down or the USB memory was removed while accessing the CF card in the controller, such as when data is being saved to the controller, a scene group is being switched and a scene group is being saved to or loaded from the USB memory, the data file may get corrupted. The file needs to be repaired. Please contact OMRON.
- **Monitor images are disordered**
- **Are the power supply and cable generating electronic noise?**
- **Is the monitor cable connected correctly?**

Input cannot be made

- **Are the cables for input devices (mouse, etc) connected correctly?**
- **Is the angle too big when tapping is done with the touch pen?**

Camera image does not display/Image is blurry

- **Is the lens cap removed?**
- **Is the camera cable connected correctly?**
- **Is the lens aperture the maximum or the minimum?**
- **Is the camera's shutter speed correct?**
- **Is the lighting method correct?**

Start-up is slow

- **Was the system connected to a LAN when started?**
  If the system is started while connected to a LAN, startup may take a longer time.

During Operation

Measurement results do not display on the monitor

- **Are windows other than the Main screen (the Edit Flow window, etc.) displayed?**
- **Are any setting windows open?**
  * A setting window is defined as a window that opens separately such as the Edit Flow window and the Scene Maintenance window.

"NG (memory shortage)" is displayed in the "Detail result" area

- **Does the number of specific processing items, such as camera image input, exceed the limit?**
  Reference: ▶ About Limits on the Number of Image Input Processing Items Used (p.858)

Data cannot be saved

- **The data save may have failed because there was not enough free capacity in the flash memory in the controller. The current scene group data in the controller memory is destroyed after the restart. Immediately perform one of the recovery/avoidance procedures specified below.**
  - For example, reduce the memory usage and then save data again to the controller, delete scenes, delete processing units, adjust the processing unit measurement area, or adjust the sizes of registered models.
  - Evacuate the current scene group data to a USB memory.
  Click [Function] menu - [Save to file] - [Setting data] - [Scene group data].

The window switching speed is slow.

- **When both lines are set to through display in the multi-line random trigger mode, the controller's response time may be slow.**

Remote Operation Tool operation or operation of tools such as the Communications Command Macro tool is unstable.

- **This can occur when there is not enough free application memory space. To free up application memory space in order to resolve this issue, try deleting processing units to reduce the number of processing units in the measurement flow.**
  For FZ5-L35□/6□□ Series models, deleting some of the processing item types registered to the measurement flow and then saving the data to the Sensor Controller directly and restarting the system may work to resolve this issue.

The image will not be displayed when remeasuring saved images

- **You may be using the through image mode in the image display pane on the used layouts. If you are used the through image mode in the image display pane, the image will not be displayed. Check the image mode for all used and unused image display pane.**
  Refer to "Arranging Windows" (Reference: ▶ Image Pane (p.85)) and check the image display mode specifications for all 24 image display panes.

For Measurement

Display is not updated.

- **Measurement is given priority when the STEP signal input interval is short or continuous measurement is being performed. As a result, the measurement results (overall judgment, image, individual judgment in the flow display, detailed results) may not be updated. When continuous measurement ends, the measurement results for the final measurement are displayed.**

The measurement NG (image mismatch) error will result when connecting a monochrome camera

- **Immediately after performing such operation as starting up the FH/FZ5 Sensor Controller, changing scenes, and clearing measurement results, there will be no input image. If you edit the settings or press the OK button in the Properties Dialog Box with no input image, the error will occur at the next measurement.**
- **If you open the Properties Dialog Box and click the OK button to close the dialog box with no input image, the settings will automatically switch to the factory default settings for the color cameras. If measuring an image input from the monochrome camera with these settings, the error will result due to the mismatching color modes between the camera and the settings.**
- **To correct the settings after the error occurrence, input the image from the monochrome camera and open the Properties Dialog Boxes for each processing unit. Click the OK button on each screen to exit the screen.**

Target figure for the processing unit figure setting has disappeared

- **This occurs when image input or image compensation-related units, such as filtering, are set between the processing unit figure setting and the unit to be changed.**
  Processing Items List Manual
  Processing Item List > Supporting Inspection and Measurement > Processing Unit Figure Setting
  Refer to the [Important] section of the aforementioned.

Judgement (JG) value is -10

- **The judgement will be -10 when there is incompatibility between the scene setting and the input image. Could the image be a color image even though it is loaded to a scene that is set for a monochrome camera? Please check the setting.**

About Parallel Interface

Trigger signal (input signal) not accepted

- **Are the cables connected correctly?**
- **Is the signal cable disconnected?**
  You can check the communication status in the Confirmation window.
  Reference: ▶ *Parallel Communications - Testing Communications* in the *Vision System FH/FZ5 Series User's Manual for Communications Settings* (Cat No. Z342)
- **Are windows other than the Main screen (the Edit Flow window, etc.) displayed?**
  Are there any setting windows open?
  * A setting window is defined as a window that opens separately such as the Edit Flow window and the Scene Maintenance window.

Signals cannot be output to external devices

- **Is the trigger signal input?**
- **Are the cables connected correctly?**
- **Is the signal cable disconnected?**
  You can check the communication status in the Confirmation window.
  Reference: ▶ *Parallel Communications - Testing Communications* in the *Vision System FH/FZ5 Series User's Manual for Communications Settings* (Cat No. Z342)
- **Is test measurement being performed?**
  Data cannot be output to external devices during test measurement.

GATE signals are not output

Wiring:

- **Are parallel cables wired correctly?**
  Please check if GATE signals are being recognized by the receiving side (such as the PLC) by turning the GATE signal ON/OFF in the System - Communication - Parallel - Confirmation window.

Output setting:

- **Is parallel judgement output or parallel data output set for the scene? Is an expression input for parallel data output?**
- **GATE signals are not output unless these items are specified.**
- **Is [Output] set to ON in the layout setup?**
  Set whether to produce external outputs for each layout.

Timing:

- **Are the parallel communication settings (output cycle and output time) set to a length sufficient for the PLC to recognize?**

- **Could the System - Communication - Parallel output control be set to handshaking or synchronization output?**
  These settings influence the GATE output timing.
  Reference: ▶ *Parallel Communications - Time Charts* in the *Vision System FH/FZ5 Series User's Manual for Communications Settings* (Cat No. Z342)

Serial Interface (RS-232C/422 Connection)

No communication available

- **Are the cables connected correctly?**
- **Are the communication specifications of the external devices compatible with the controller?**
  You can check the communication status in the Confirmation window.
  Reference: ▶ *Non-procedure Communications - Testing Communications* in the *Vision System FH/FZ5 Series User's Manual for Communications Settings* (Cat No. Z342)

The controller works fine initially, but there is not response after a while

- **Is the buffer memory of the PC full?**
  Please verify if data can be received correctly based on current settings.

Data cannot be saved

- **Are the communication specifications of the external devices compatible with the controller?**
- **Is "Flow control" in communication specifications set to "None"?**
  To save data, set "Flow control" to "None".
  Reference: ▶ *Non-procedure Communications - Communications Specifications Settings* in the *Vision System FH/FZ5 Series User's Manual for Communications Settings* (Cat No. Z342)

Camera with Lighting Controller

The light does not turn ON at all

- **Did you disconnect or connect the Lighting Controller or Light while the power supply was ON?**
- **Are two or more lights connected?**
  If more than one light is connected, it may not be possible to turn them ON simultaneously depending on the total current consumption of the lights. If necessary, set the brightness to 0 for all of the lights except for the light to turn ON.
  Refer to ▶ *Camera Image Input, Camera Image Input FH, or Camera Image Input HDR* in the *Vision System FH/FZ5 Series Processing Items Reference Manual* (Cat No. Z341).

The light will not stay ON all the time

- **Are two or more lights connected?**
  If two or more lights are connected, the always-on lighting mode cannot be used for lighting if the total power consumption of the lights is 7.5 W or higher.

The light malfunctions

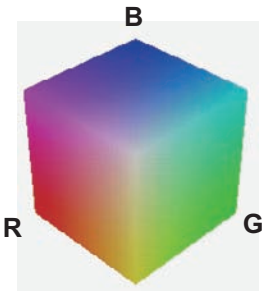
- **Is the sequence for turning ON power correct?**
  If you are using an external power supply, either turn it ON at the same time as the power supply to the Sensor Controller or turn ON the external power supply first.

Measurement Mechanism

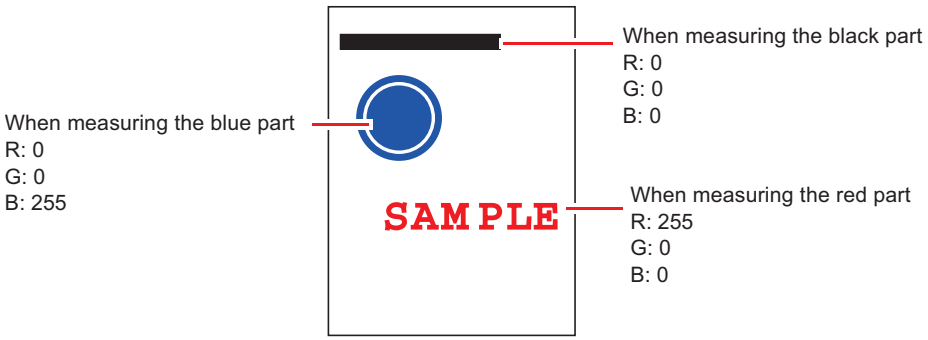
This section describes how to configure measurement in accordance with the images acquired from cameras. This product is prepared with comprehensive processing capabilities for measuring items. Common processes for various processing items are described here.

Color Processing Mechanism

The images acquired from cameras are available with color information such as R(red)/G(green)/B(blue). RGB images can render 16.7 million colors on the screen, and adjustment of color intensity with a range of 0 to 255 can be performed.
For each of RGB, black is rendered with a 0 value and white is rendered with a 255 value.



During measurement of color images, many colors can be measured by adjusting the values for RGB.



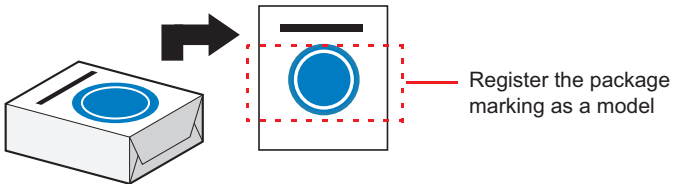
Note

- For monochrome cameras, color processing is not performed.

Search Processing Mechanism

Reference image patterns are registered as models and then search is performed for the parts of input images that most resemble the models. The degree of similarity is represented with a correlation value, inspection for defects and different parts being mixed in can be performed.
The search process is performed over several distinct stages.

1 Register a reference model.



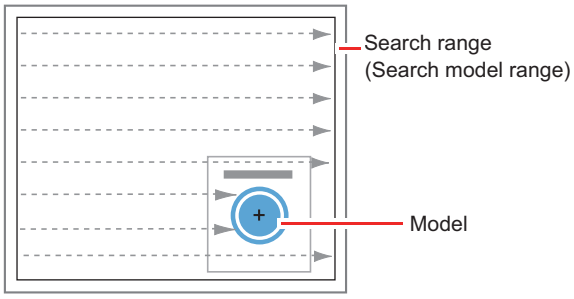
Note

Model Status and Measurement Processing

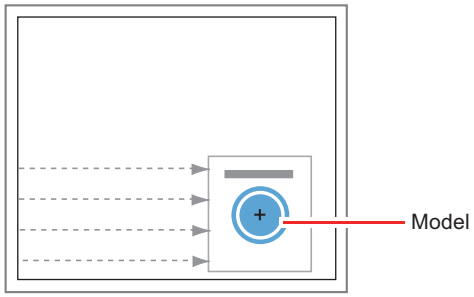
- Measurement time and accuracy may be affected by the status of model in the following ways. Please select measurement objects that are in good condition (clean) for Model Registration.
  - In the case of large or complicated models, processing time is prolonged.
  - With extremely small models or models without features, search processing is unstable.

2 Perform rough search of overall measurement region.

Search for the model over the entire measurement region.



3 Perform additional searching near the model.



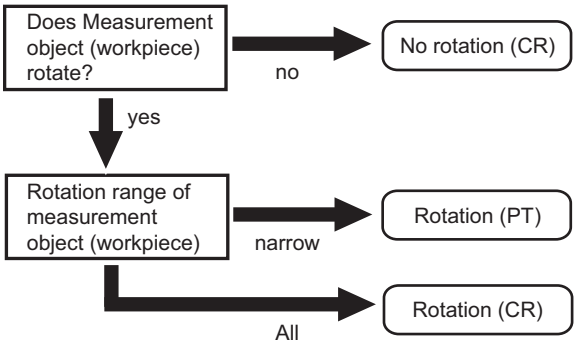
Search Detection Method

With search processing, there are two types of detection methods: Search by "Correlation (CR)" and search by "Shape (PT)".
The detection method can be selected by changing the "Rotation" settings for each processing item.

Setting item	Description
CR	Measurement for locations where there is high correlation (similar areas) between the color of the model registration image and the input image is performed. Since brightness is normalized in this operation, changes in brightness do not affect searching for correct positions.
PT	Measurement for locations where there is a high degree of similarity between the shape of the model registration image and the input image is performed. Since measurement is based on shape consistency (profile of model), positions can be measured accurately even if portions of the model are missing in images. When the rotation angle range is wide (such as full angle), a search that is faster than a "CR" search can be performed. Note, however, that this method may be less reliable than a "CR" search if the image has low contrast or blurred edges.

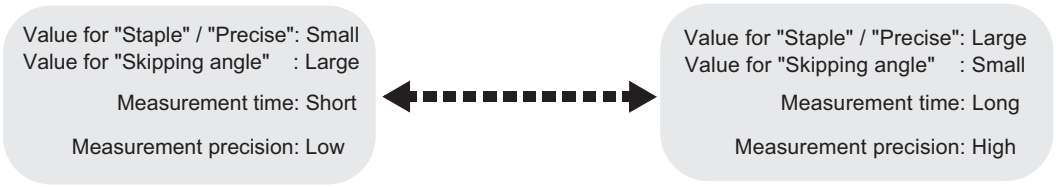
Note

Basic selection flow



Search Speed

For processing items to perform search processing such as [Search] and [Classification], you can specify the search processing speed by through the model parameter items "Stable", "Precise", and/or "Skipping angle". If the value specified for "Stab." or "Prec." is small or if the value for "Skipping angle" is large, the processing speed can be increased since the amount of information for the models will be decreased during the search. In contrast, if the values specified for "Stab." or "Prec." are large or if the value for "Skipping angle" is small, the processing speed is slow because search is performed without the amount of model information being reduced. Specify appropriate values for "Stable", "Precise", and "Skipping angle" according to the measurement conditions.



Correlation value

With processing items that use search processing, judgement is performed through correlation values. Correlation values are used to check the consistency (degree of similarity) between actual measurement images and reference model images. If portions of measured images are missing or if shapes are different, the correlation value is lower.

Model image

For partial defect

Correlation: 54

For wide line

Correlation: 74

For dark image

Correlation: 99

Larger

Correlation: 43

Horizontal

Correlation: 58

Different shapes

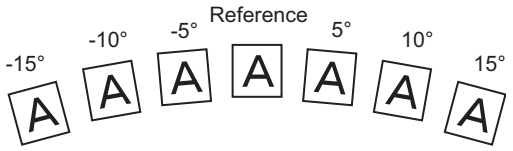
Correlation: 58

Search Angle Range, Skipping Angle

These are values, based on the model registration image, that indicate the allowable rotation interval (skipping angle) and overall maximum rotation range for the model (angle range). Search is performed for objects that most resemble these acceptable models.

Example: When angle range is 15×and skipping angle is 5×

Registration model

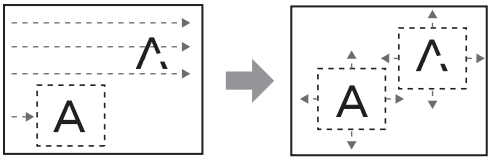


Candidate Point Level

This is the level used for finding models when searching. Images with a correlation value higher than the candidate point level are used to establish candidate points for search inspection.

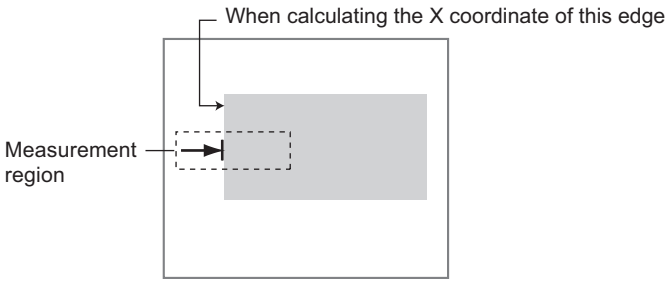
Roughly search within the search region for candidate points.

Perform detailed searches in the areas around each candidate point.



Edge Detection Measurement

This method extracts parts with color changes as edges to perform measurement.
Edges are found through color changes in the measurement region.



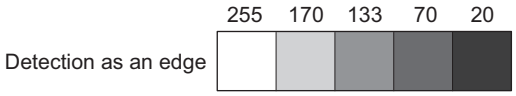
Edge Detection Method

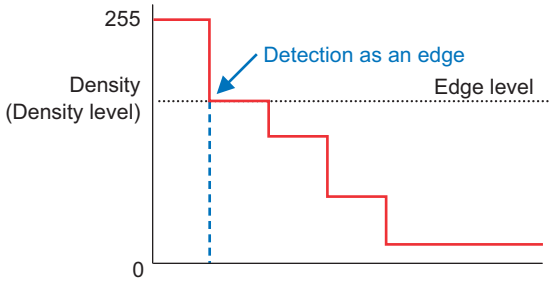
The following two types of edge detection method are available.

Projection (Density method)

A projected waveform is formed relative to the density level. The intersection of the density and the threshold (edge level) is detected as an edge.

With an image with high noise level or image with blurry edges, the density method is suitable as it allows you to specify the density and color of the edges to be detected.

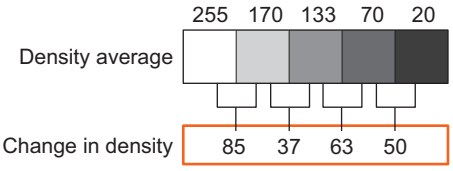


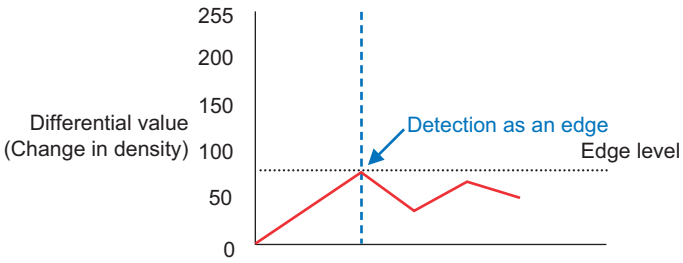


Derivation

A differential processing is performed to calculate the change in density between neighboring pixels. A differential waveform with the largest density set as 100% of the region is created, and the maximum value (peak point) of the differential waveform that exceeds the threshold (edge level) is detected as an edge.

With an image with low contrast, the differential method is suitable as it normalizes the differences between the neighboring pixels for processing.





Edge Level

Indicates the edge color change level (degree of color difference). This level is adjusted if edges cannot be accurately detected.

Note

The value 0 to 100 for the edge level indicates the edge intensity. It is not related to color difference in the original images.

For case of measurement using relative position (%) with regards to width of color difference

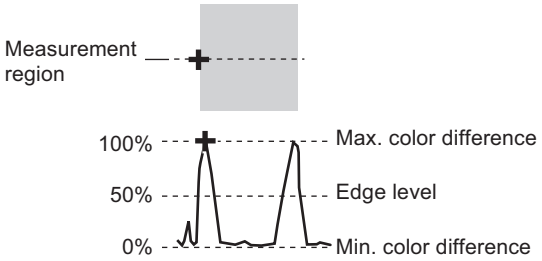
The edges are detected in the following manner.

1 **Calculate the overall distribution of color difference in the measurement region.**

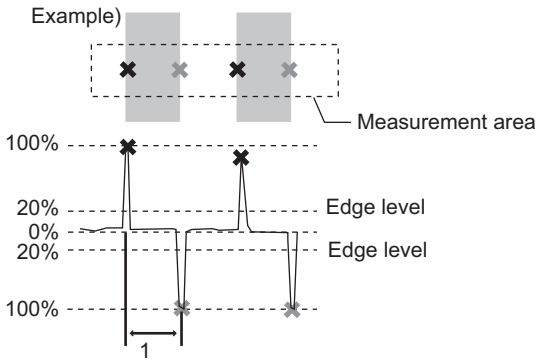
2 **Min. color difference value: 0%; Max. color difference value: 100%**

3 **Locations in which there is an edge level color difference are detected as edges.**

Edge Position

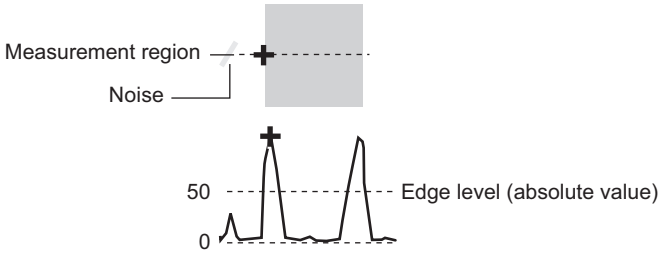


Edge Pitch



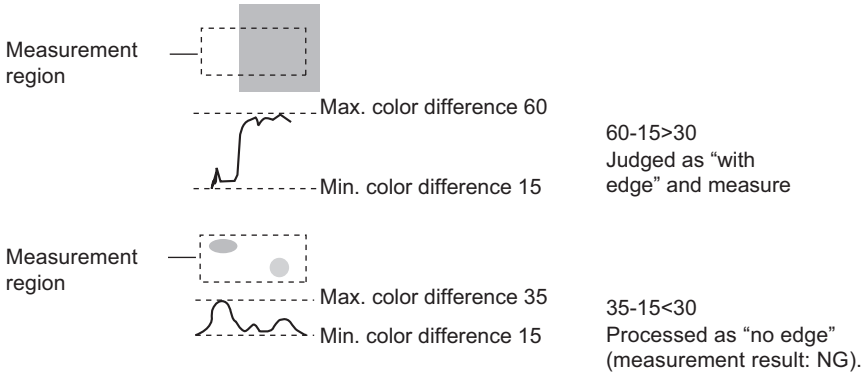
When performing measurement using color difference value

Edge level is set using color difference absolute value.



Noise Level

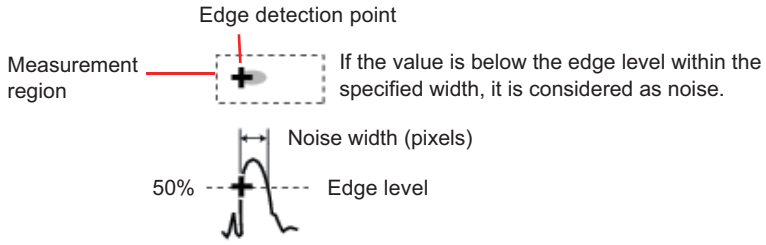
This level judges whether an edge is present or not.
The maximum value and minimum value for color difference in the edge detection area are determined and if the difference is less than the noise level, it is judged that there is no edge. When detection is affected by noise, increase this value.
(within area)
Maximum value - minimum value of color difference < noise level - edge does not exist - measurement result is fail
Maximum value - minimum value of color difference ≥ Noise level - edge exists - targeted for measurement
Example: When noise level is set to 30



Noise Width

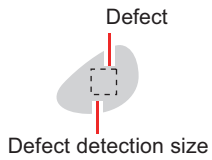
Set the width for judging noise.

- **When no color is specified**
  If another edge is detected within the width range starting from the point where an edge was first detected, the newly detected point is considered noise.
- **When a color is specified**
  If the color difference distributions again falls below the edge level within the width range starting from the point where an edge was first detected, the newly detected point is considered noise.



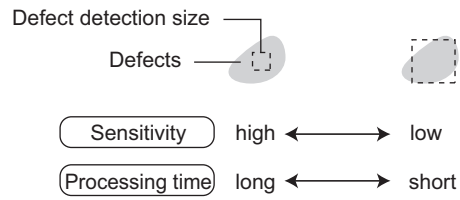
When points are mistakenly detected as edges due to noise, increase the color difference value.

Defect Detection Measurement



Color changes within the measurement region are used to find defects such as scratches,contamination, and chipping.
After measurement region is drawn, a rectangle (defect detection region) is automatically formed in this region. While moving the defect detection region around, calculate average density for each area to determine the difference between the original area and the surrounding area. This difference is called the defect level. Calculate the defect level for all defect detection areas. If the maximum value exceeds the judgement value, it is judged that there are defects in the measurement region.

Increasing "Defect size" allows for shortening of processing time, but this will reduce measurement accuracy.

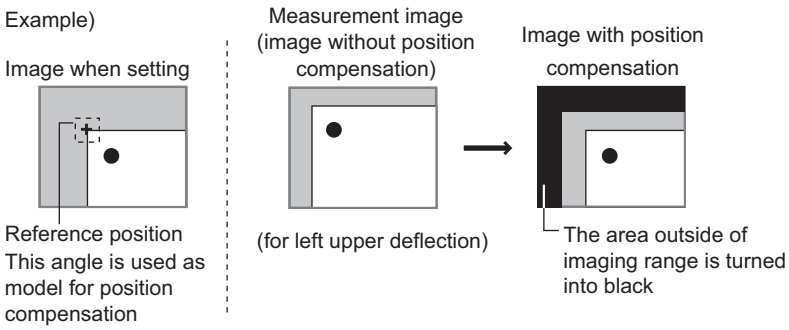
Setting item	Description
Upper Lower	Specify the upper and lower limits of defect detection size based on the size of scratch or contamination to be detected. The larger the difference between upper and lower limits, the easier it is to detect the scratches or contamination of various sizes. For both upper and lower limits, higher values for defect detection size limits leads to weaker detection sensitivity and shorter processing time. 

Handling Coordinates

The processing items for measuring positions have a setting item called "Output parameter" with which you can select how to handle coordinates.
In "Output parameter", you can set "Output coordinates" and "Calibration".

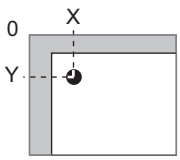
Output Coordinates

Selects coordinate types to be output to external devices.



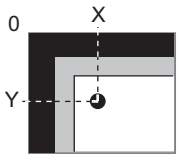
Before position compensation

Coordinate values before position compensation are output.



After position compensation (with factory settings)

Coordinate values after position compensation are output.



Calibration

Select whether or not to perform calibration when selecting output to external devices.

Calibration: ON

Calibration is performed during output and measured values after calibration are output.

Calibration: OFF

Calibration is not performed during output, and coordinate values from camera images are output.

Inputting Values

This section describes how to input values required for setting the judgement conditions and communication specifications. Methods for setting up values include the following, depending on the settings.

- **Specify values directly with the numeric keyboard**
  Used to enter specific values, or to fine-tune the value with the UP and DOWN keys.
- **Set numerical values by dragging the slider**
  Setting values can be done by dragging the slider on the screen.

The method for displaying the numeric keyboard and setting values is explained here. For other methods, refer to individual setting descriptions.

1 **Click [...] in the item in which a value is to be set.**
The numeric keyboard is displayed.

2 **Click the numeric keys to input values.**
The numerical value is input.
To fine-tune the value, click the UP or DOWN key to increase or decrease the value.

3 **Click [OK].**
This verifies the value and closes the numeric keyboard.

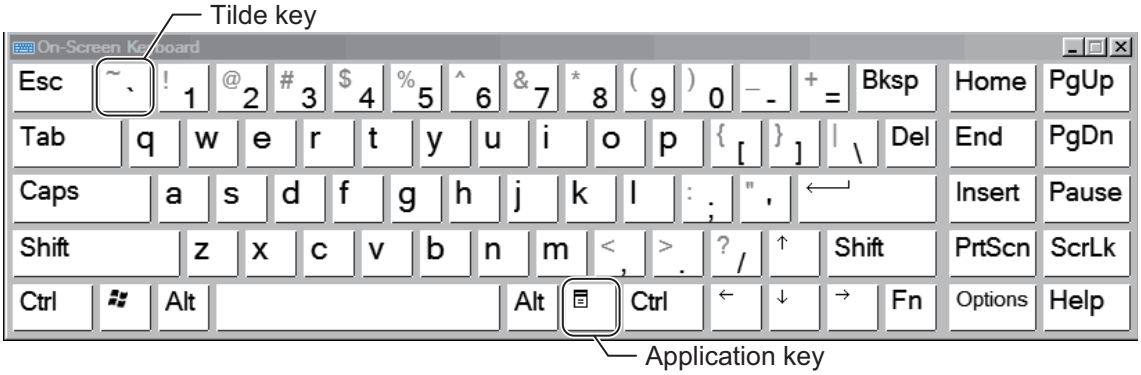
Inputting Text

This section describes methods for inputting file names and descriptive text.
The following software keyboard is displayed in the window for inputting text.
It has the same arrangement as an English keyboard (101/104 keyboard).
Use one of the following procedures to switch to the Japanese input mode.

- **Press the [Alt] + [~] (Tilde) keys.**
- **Press the application key and select [Open IME].**

Note

If you connect a keyboard to the Sensor Controller with a USB connection, the key arrangement will be the same as an English keyboard.



Operation Method

1 **Click [...] in the item in which a character string is to be set.**

The soft keyboard is displayed.

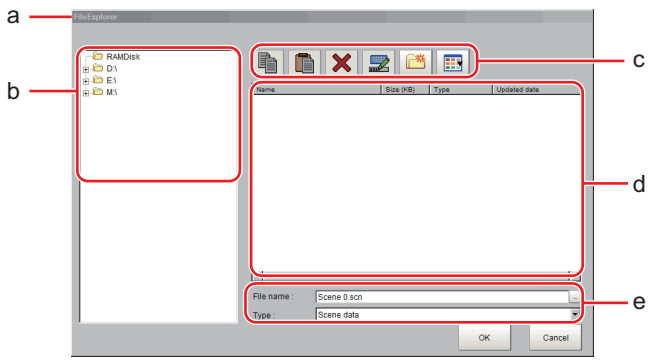
2 **Switch the tabs as needed and Click the character that is to be input.**

3 **Click [OK] after text has been entered.**

The software keyboard is closed.

Selecting Files and Folders

This section describes data save/load methods and operation methods for when selecting a save destination folder for images created during remeasurement, etc.
The following window will appear in the window to select a file or a folder.



a. Window Title

When a file is specified, "FileExplorer" is displayed. If a folder is selected, "Select folder" is displayed.

b.Folder View Area

A list of folders on the RAM disk and folders in the mounted USB memory is displayed. At the root of the tree, the drive names of all accessible USB memory devices are displayed (ex. "USBDisk", "USBDisk2").

c. Toolbar

-  **Copy**

  Reference: ▶Copying/Pasting a File or Folder (p.846)

-  **Paste**

  Reference: ▶Copying/Pasting a File or Folder (p.846)

-  **Delete**

  Reference: ▶Deleting a Folder or File (p.846)

-  **Rename**

  Reference: ▶Renaming a Folder or a File (p.846)

-  **New folder**

  Creates a new folder.

-  **Toggle list view**

  Switches the display format of the List View area.

d. List View Area

A list of files and folders contained in the folder selected from the folder view area is displayed.
In addition, when an extension name is selected from "Kind", only the files with the selected extension name are displayed. The content displayed is switched if [List display toggle] () is clicked.

e. File Name View Area

- **File name**
  Names of files selected in the List View area are displayed.
- **Kind**
  Specifies the types of the file displayed ("Scene data", "System data", etc.) in the List View area.

Available Operations in Select File Window

This section describes the main operations available from the Select File window.

Note

If the target file is not displayed in the list view area when selecting a file, please check that the file type of the target file is selected in "Kind".

Copying/Pasting a File or Folder

1 **Click the folder or file that you want to copy in the List View area.**
The file or folder will be selected.

2 **Click [Copy] ( ).**

3 **Select the target folder, and Click [Paste] ( ).**

Renaming a Folder or a File

1 **Click the name of file or folder to be renamed from the List View area.**
The file or folder will be selected.

2 **Click [Rename] ( ).**
The soft keyboard is displayed.

3 **Enter a new name.**
Character input method: Reference: ▶ Inputting Text (p.844)

Note

When a file or folder with the same name exists within the folder, an error message will display telling you that you cannot change the name.

Deleting a Folder or File

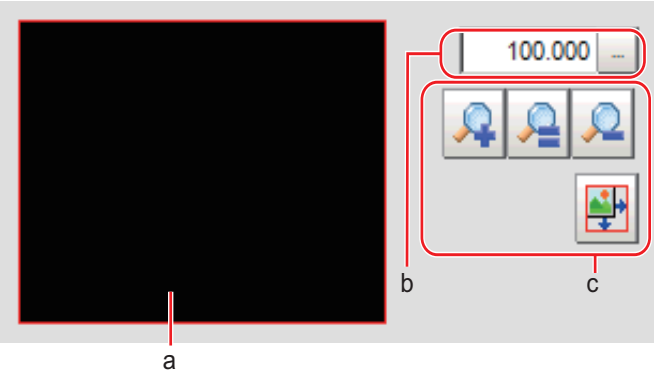
1 **Click the name of file or folder to be deleted from the List View area.**
The file or folder will be selected.

2 **Click [Delete] ( ).**
A confirmation window is displayed.

3 **Click [OK].**
The selected file or folder will be deleted.

Using the Zoom Function

Specifies the magnification settings of the image displayed in the Properties window.



a. Zoom browser

Indicates where the zoom display area is in the original image.

b. Magnification factor

Input the magnification factor A factor of between 25% to 1600% can be indicated.

c. Operation buttons

- **Zoom-in button**

  Enlarges the selected area to twice its size.

- **Original size button**

  Displays the selected area in the original size.

- **Zoom-out button**

  Reduces the selected area by half.

- **Full-screen button**

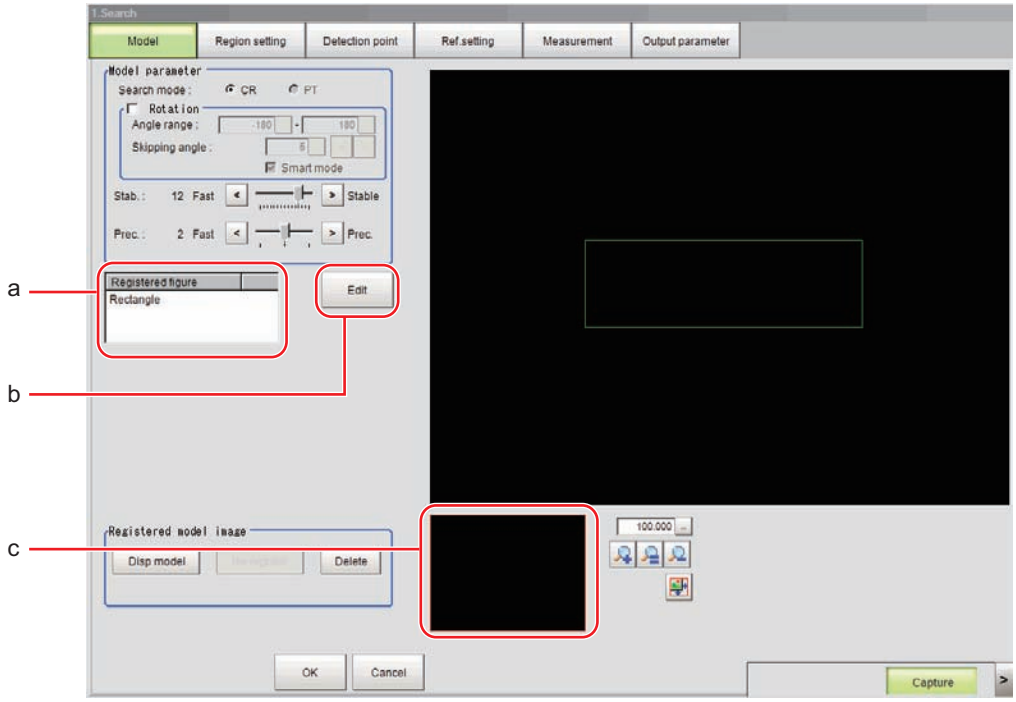
  Enlarges the zoom browser to the entire screen and returns it to its original size.

Setting Figures

This section describes the setting method for objects (figures and text) when registering models or specifying measurement regions.
The type and number of objects varies depending on different setting options.

Layout of Figure Setting Area

Window for registering figures when registering or setting areas or models as measurement objects.

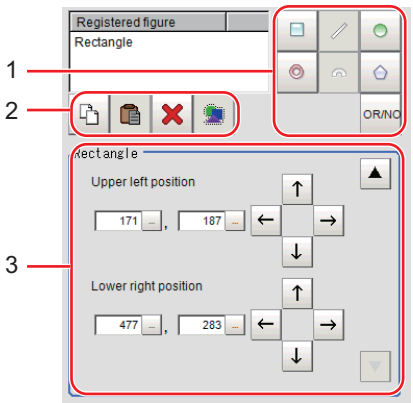


a. Figures

Displays a list of names of objects that have been set. The figure at the bottom of the list is the nearest object in the foreground. The higher the sequence position of the object, the further back in the background it is. When objects are drawn overlapping, the settings for the object set last are valid.

b. [Edit]

Used to edit a figure. The following figure editing tool is displayed.



1. Drawing tool buttons

Sets up objects, such as figures and text. The number and type of objects available is different depending on the applicable setting (ex. "Result display", "Model", "Region setting").

2. Object editing buttons

Buttons for editing objects

3. Details

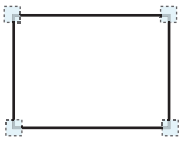
Shows the details of the selected figure. Specify the object coordinates or radius. Tapping [▲] or [▼] will display the items currently not displayed.

c. Zoom Browser Area

Magnifies the Image Display area by the selected magnification factor.

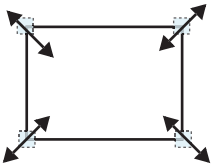
Rectangle

- **Image selection status**

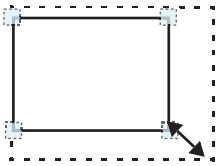


Points are displayed at each of the four corners.

- **Dimension Adjustment**



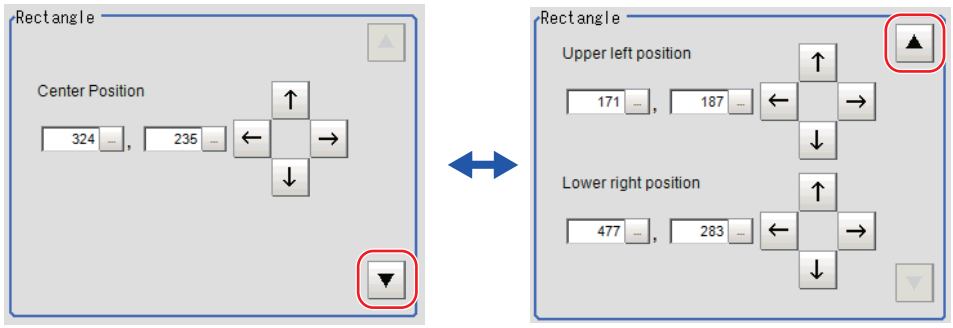
- **Example) When enlarging**



Drag the lower right point down in a diagonal direction.

- **Using numbers for setting**

The window for settings is split into two. Setting is performed through input of numbers or through tapping on the arrows.



Line

- **Image selection status**

Points are displayed at the starting point, ending point, and mid point of lines.

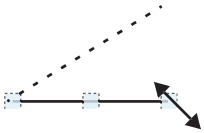
- **Dimension Adjustment**

Drag the points.

- **Example) When changing the length of a line**

Drag the points toward the intended direction.

- **Example) When changing the oblique direction**



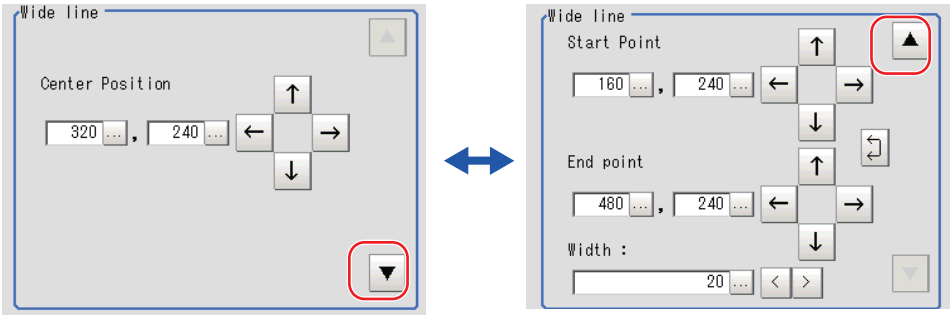
Drag a point in the direction the line is to be changed.

- **Example) When changing the line width**

Drag the center point of the line in a direction perpendicular to the line.

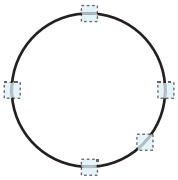
- **Using numbers for setting**

  The window for settings is split into two. Setting is performed through input of numbers or through tapping on the arrows.



Circle/Ellipse

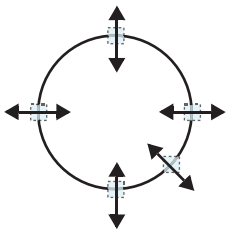
- **Image selection status**



Points are displayed on the top, bottom, left, right, and lower right of the circle.

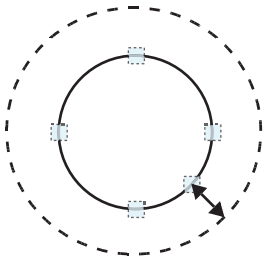
- **Dimension Adjustment**

  Drag the points.



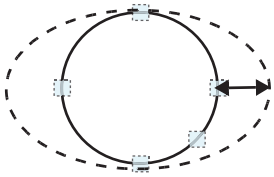
- **Example) When zooming in on a circle**

  Drag the point on the lower right of the circle.



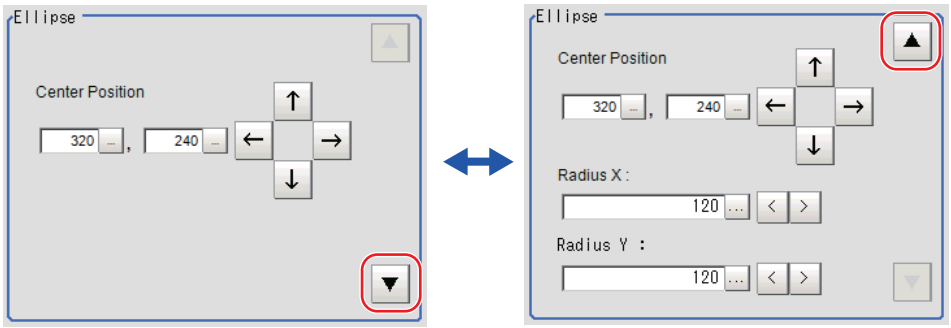
- **Example) When transforming a circle into a long horizontal ellipse**

  Drag the point on the right of the circle to the right.



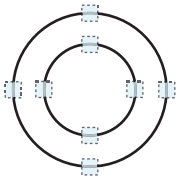
- **Using numbers for setting**

  The window for settings is split into two. Setting is performed through input of numbers or through tapping on the arrows.



Circumference

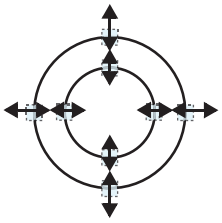
- **Image selection status**



Points are displayed on the top, bottom, left, and right of both the inner and outer circles.

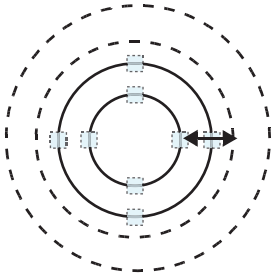
- **Dimension Adjustment**

Drag the points.



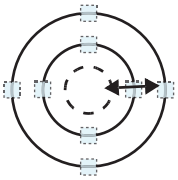
- **Example) When enlarging the entire circumference**

Drag a point on the outer circle.



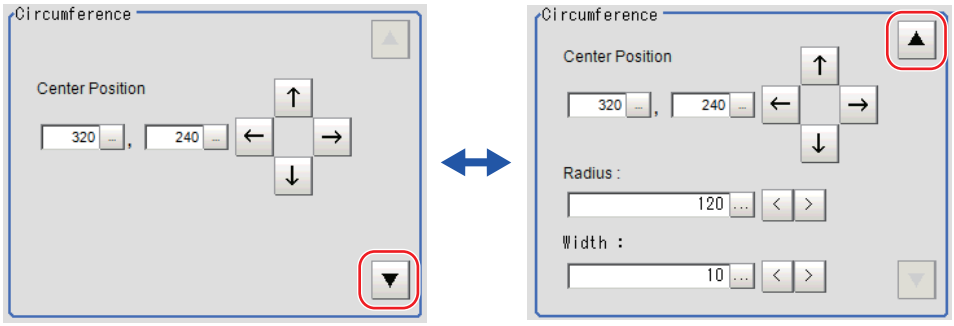
- **Example) When adjusting the width of the circumference**

Drag a point on the inner circle.

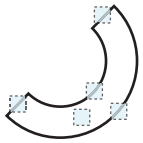


- **Using numbers for setting**

The window for settings is split into two. Setting is performed through input of numbers or through tapping on the arrows.



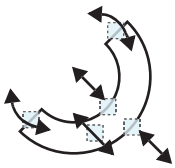
- **Image selection status**



Points are displayed on two lines at both ends of the arcs, on the inner arc, on the outer arc, and inside the

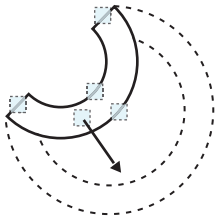
- **Dimension Adjustment**

Drag the points.



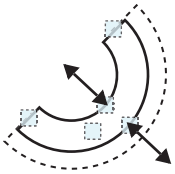
- **Example) When enlarging an arc**

Drag the point inside the arc outward.



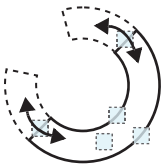
- **Example) When adjusting the width of an arc**

Drag a point on the inner or outer arc inward or outward.



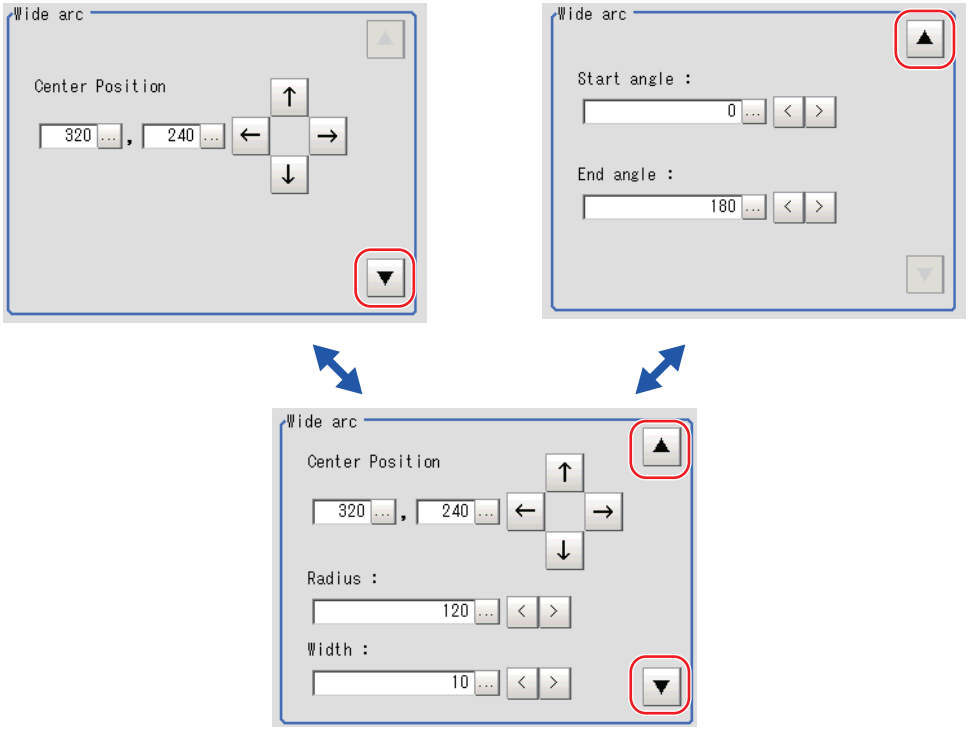
- **Example) When changing the angle of arc (part that is open)**

Drag one of the points at the end of the arc.



- **Using numbers for setting**

The window for settings is split into three. Setting is performed through input of numbers or through tapping on the arrows.



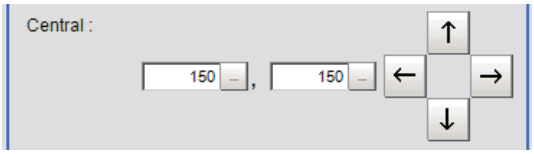
Crosshair Cursor

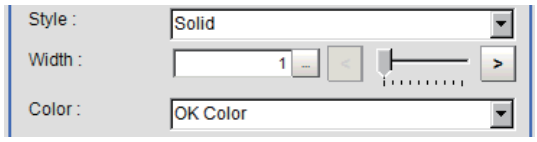
- **Image selection status**

Entire image is selected.

- **Using numbers for setting**

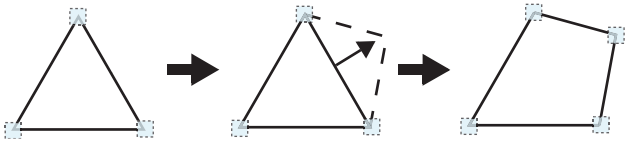
Setting is performed through input of numbers or through tapping on the arrows. The line type and line color can also be changed at this window.





Polygons (Triangle to Decagon)

- **Drawing methods (for drawing a quadrilateral)**

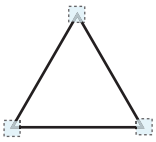


1 **When [Polygon] is specified, a triangle is drawn at first.**

2 **If you drag and drop one of the sides at the point you want to make a new vertex, a new vertex will be created.**

If the number of vertexes is not within 3 to 10, the image cannot be confirmed as a polygon.

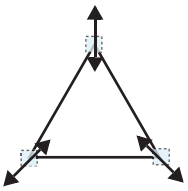
- **Image selection status**



Point is displayed at vertex of figure.

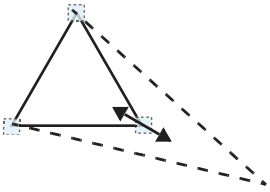
- **Dimension Adjustment**

Drag the points.



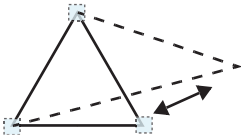
- **Example) When changing the angle of one point**

Drag points (arbitrarily).



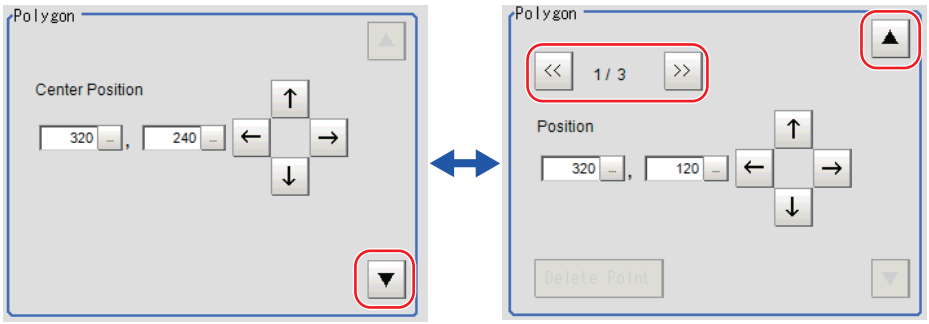
- **Example) When changing the region**

Drag points (arbitrarily).



- **Using numbers for setting**

  The window for settings is split into two. Setting is performed through input of numbers or through tapping on the arrows.



Text

- **Image selection status**

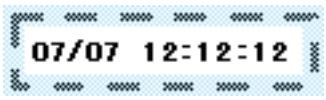
Entire image is selected.

Note

- "Text" can only be used in the [Result display] processing item.

Time

- **Image selection status**



Entire image is selected.

Note

- "Time" can only be used in the [Result display] processing item.

About OR Setting/NOT Setting

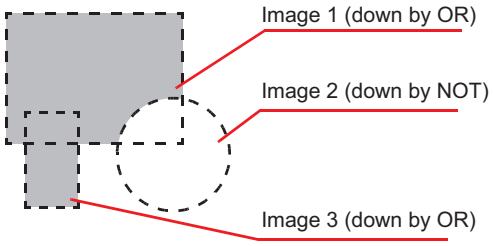
The OR setting/NOT setting is used when multiple images are combined.
Areas with complex shapes can be drawn through combining figures, and unnecessary parts can be excluded form the area. Each time [OR/NOT] (OR/NOT) is tapped, the setting of the selected figure toggles between OR and NOT.

IMPORTANT

- Images with only the NOT setting cannot be drawn.
- At the location that overlaps another figure, OR/NOT of the region set up later will take priority.

Item	Description
OR/NOT (OR/NOT)	Used when drawing a model or a region Switching between OR and NOT occurs each time the button is tapped.
OR	The selected figure is displayed with a dotted line in an OK color. When drawing multiple figures, the entire area is registered as one region.
NOT	The selected figure is displayed with a dotted line in an NG color. The area outside of the NOT image is registered as a region.

(Example) The grey parts are measurement regions.



About Number of Logging Images

The maximum number of logging images (I_MAX) that can be stored in the controller's memory varies depending on the type of the controller in use and the type and number of connected cameras.

For color cameras:

Type of controller	0.3 megapixel camera								Intelligent compact camera							
	1 camera	2 cameras	3 cameras	4 cameras	5 cameras	6 cameras	7 cameras	8 cameras	1 camera	2 cameras	3 cameras	4 cameras	5 cameras	6 cameras	7 cameras	8 cameras
FZ5-L3□□	250	125	83	62	---				214	107	71	53	---			
FZ5-6□□																
FZ5-11□□	270	135	90	67					232	116	77	58				
FH-1□□□ FH-3□□□	270	135	90	67	54	45	38	33	232	116	77	58	46	38	33	29

Type of controller	2 megapixel camera								5 megapixel camera							
	1 camera	2 cameras	3 cameras	4 cameras	5 cameras	6 cameras	7 cameras	8 cameras	1 camera	2 cameras	3 cameras	4 cameras	5 cameras	6 cameras	7 cameras	8 cameras
FZ5-L3□□	40	20	13	10	---				11	5	---		---			
FZ5-6□□																
FZ5-11□□	43	21	14	10					16	8	5	4				
FH-1□□□ FH-3□□□	43	21	14	10	8	7	6	5	16	8	5	4	3	2	2	2

Type of controller	0.3 megapixel camera								2 megapixel camera							
	1 camera	2 cameras	3 cameras	4 cameras	5 cameras	6 cameras	7 cameras	8 cameras	1 camera	2 cameras	3 cameras	4 cameras	5 cameras	6 cameras	7 cameras	8 cameras
FZ5-L3□□	---								---							
FZ5-6□□																
FZ5-11□□																
FH-1□□□ FH-3□□□	270	135	90	67	54	45	38	33	37	18	12	9	7	6	5	4

Type of controller	4 megapixel camera								12 megapixel camera							
	1 camera	2 cameras	3 cameras	4 cameras	5 cameras	6 cameras	7 cameras	8 cameras	1 camera	2 cameras	3 cameras	4 cameras	5 cameras	6 cameras	7 cameras	8 cameras
FZ5-L3□□	---								---							
FZ5-6□□																
FZ5-11□□																
FH-1□□□ FH-3□□□	20	10	6	5	4	3	2	2	6	3	2	2	---			

For monochrome cameras:

Type of controller	0.3 megapixel camera								2 megapixel camera							
	1 camera	2 cameras	3 cameras	4 cameras	5 cameras	6 cameras	7 cameras	8 cameras	1 camera	2 cameras	3 cameras	4 cameras	5 cameras	6 cameras	7 cameras	8 cameras
FZ5-L3□□	252	126	84	63	---				40	20	13	10	---			
FZ5-6□□																
FZ5-11□□	272	136	90	68					43	21	14	10				
FH-1□□□ FH-3□□□	272	136	90	68	54	45	38	34	43	21	14	10	8	7	6	5

Type of controller	5 megapixel camera								0.3 megapixel camera							
	1 camera	2 cameras	3 cameras	4 cameras	5 cameras	6 cameras	7 cameras	8 cameras	1 camera	2 cameras	3 cameras	4 cameras	5 cameras	6 cameras	7 cameras	8 cameras
FZ5-L3□□	11	5	---		---				---							
FZ5-6□□																
FZ5-11□□	16	8	5	4												
FH-1□□□ FH-3□□□	16	8	5	4	3	2	2	2	272	136	90	68	54	45	38	34

Type of controller	2 megapixel camera								4 megapixel camera							
	1 camera	2 cameras	3 cameras	4 cameras	5 cameras	6 cameras	7 cameras	8 cameras	1 camera	2 cameras	3 cameras	4 cameras	5 cameras	6 cameras	7 cameras	8 cameras
FZ5-L3□□	---								---							
FZ5-6□□																
FZ5-11□□																
FH-1□□□ FH-3□□□	37	18	12	9	7	6	5	4	20	10	6	5	4	3	2	2

Type of controller	12 megapixel camera							
	1 camera	2 cameras	3 cameras	4 cameras	5 cameras	6 cameras	7 cameras	8 cameras
FZ5-L3□□	---							
FZ5-6□□								
FZ5-11□□								
FH-1□□□ FH-3□□□	6	3	2	2	---			

About Limits on the Number of Image Input Processing Items Used

For the FZ5-L3□□/6□□ series, The number of image input processing items that can be used within one scene is limited by the combinations of the camera. (Units that are not used in conditional branching, etc. are also targeted.)
If the limit is exceeded, a measurement NG occurs due to a memory shortage during measurement.
Please use within this processing items limitation.

Camera used	Number of image input related processing item restriction
0.3 megapixel color camera	81
0.3 megapixel monochrome camera	245
Intelligent compact camera (FZ-SQ)	69
2 megapixel color camera	13
2 megapixel monochrome camera	39
5 megapixel color camera	6
5 megapixel monochrome camera	19

Image input related processing items

Target processing items are as follows.

Item	Processing item	Item	Processing item
Loading Images	Camera Image Input	Performing image compensation	Anti Color Shading
	Camera Image Input HDR		Stripes Removal Filter II
	Camera Switching		Polar Transformation(*1)
	Camera Image Input FH		Image Subtraction
	Camera Image Input HDR Lite		Advanced Filter(*2)
Performing image compensation	Position Compensation		Machine Simulator
	Filtering		Panorama
	Background Suppression		Trapezoidal Correction
	Brightness Correct Filter		Image Subtraction Panorama Precise Calibration Reference Calib Data
	Color Gray Filter	Display results	Display Image File(*3)
	Extract Color Filter		Display Last NG image(*4)

*1: When using a Polar Transformation, each image input related unit after Polar Transformation may consume up to a maximum of 2 items.

*2: For each unit, 4 items are consumed for Advanced Filter.

*3: Image file display consumes the number of image setting value items per unit.

*4: Display Last NG Image consumes the number of save setting value items per unit.

IMPORTANT

- If there is one or more "Sensitive Search" in the flow, the number of image input items that can be used is only reduced by one.
- In the case of the FH/FZ5-11□□, the number of processing items is not limited and can be registered as long as there is sufficient free memory. Note, however, that a warning message is displayed when the available memory drops to below 1 GB. In this case, adjust the inspection flow to ensure at least 1 G of available memory.

About Max. Number of Loading Images during Multi-input

The function that enables continuous high speed image input is called the multi-input function.
The maximum number of images that can be loaded based on each specification is shown below.

Type of controller	0.3 megapixel camera	2 megapixel camera	5 megapixel camera
FZ5-L3□□	16 (for cameras 0 and 1) 32 (for cameras 2 and 3)	4 (for cameras 0 and 1) 8 (for cameras 2 and 3)	1 (for cameras 0 and 1) 2 (for cameras 2 and 3)
FZ5-6□□	32	8	2
FZ5-11□□	32	8	2
FH-1□□□ FH-3□□□	256	64	25

Type of controller	0.3 megapixel camera	2 megapixel camera	4 megapixel camera	12 megapixel camera
FZ5-L3□□	---	---	---	---
FZ5-6□□	---	---	---	---
FZ5-11□□	---	---	---	---
FH-1□□□ FH-3□□□	256	51	32	10

IMPORTANT

- The multi-input function cannot be used when the built-in lighting of an intelligent compact camera, FZ-SQ□□□□, is used.
- The maximum number of images loaded does not change even if partial reading of camera images is selected.
- When using two camera cables for connection, the maximum number of loaded images during multi-input is twice the number given in the table.

Character Code Table

For the case of character related process items, recognized characters are output to an external device using a character code (base 10).

Calculation method for output value (base 10)
Number of upper level bits x 16 + number of lower level bits = output value

(Examples) If the recognized character is "2", "50" is output.
If the recognized character is "C", "67" is output.

Lower 4 bits \ Upper 4 bits	0	1	2	3	4	5	6	7
0		DE	SP	0	@	P	`	p
1	SH	D1	!	1	A	Q	a	q
2	SX	D2	”	2	B	R	b	r
3	EX	D3	#	3	C	S	c	s
4	ET	D4	$	4	D	T	d	t
5	EQ	NK	%	5	E	U	e	u
6	AK	SN	&	6	F	V	f	v
7	BL	EB	’	7	G	W	g	w
8	BS	CN	(	8	H	X	h	x
9	HT	EM	)	9	I	Y	i	y
A	LF	SB	*	:	J	Z	j	z
B	HM	EC	+	;	K	[	k	{
C	CL	↑	,	<	L	\	l	|
D	CR	↓	-	=	M	]	m	}
E	SO	→	.	>	N	^	n	~
F	SI	←	/	?	O	_	o	

About Memories Usable with FZ Series

The following types of memory can be used with FH/FZ5.

On-board memory

This is the area where images are temporarily stored when logging images using the logging function.
This uses ring memory and if the maximum number of save images has been reached, images are overwritten starting with the oldest. This is cleared when the power is turned OFF.

RAMDisk

Image logging file, data logging file, and capture images can be saved.
As this is memory inside the FZ4, files can be saved and read faster than using USB memory.
However, capacity is a fixed 40 MB (256 MB for FH/FZ5-11□□) . Files saved in RAMDisk are cleared when the power is turned OFF.

Application memory

This is a memory used for all applications.
This is a memory area that is used temporarily by applications. By confirming available memory, this provides a rough standard for confirming status while operating. The user has no access to it.

Data memory (FZ5-L3□□/FZ5-6□□)

This is an area for holding current settings details of scene group data.
If this capacity is exceeded, adding units and copying scenes cannot be performed in edit flow.
Available data memory can be confirmed from the system menu.
Reference: ▶ Checking System Information: [System Information] (p.300)

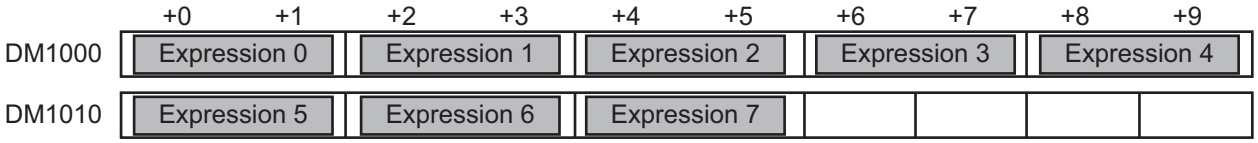
Memory Display Image on PLC I/O

The memory display image on PLC I/O varies depending on the PLC to be used.
Using data output to the Data Output area of the PLC link from the serial data output processing item as an example, this section illustrates how the memory display image varies depending on the model.

Data storage image (Data Output area DM1000)

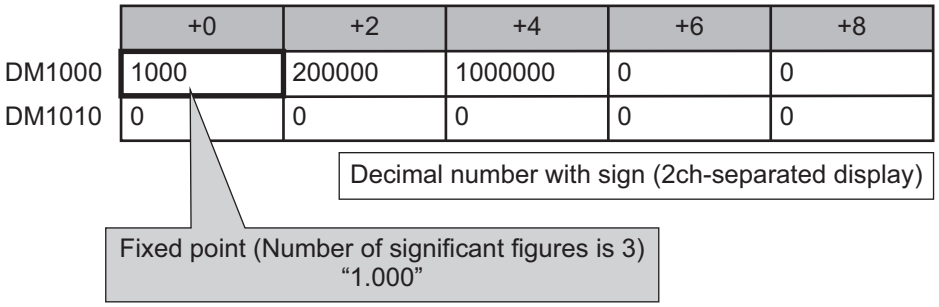
When the PLC link Data Output area is set to DM1000, data is stored as follows in the PLC I/O memory.

*1: Up to 8 expressions can be registered in the serial output flow on the FZ4 side. If 8 expressions are registered, data is stored as follows.



CX-Programmer PLC I/O memory display image

As an example, if 3 data items, expression 0 (DATA0) = 1.000, expression 1 (DATA1) = 200.000, and expression 2 (DATA2) = 1000.000, are output from FZ4, they are stored to the PLC Link area as measurement data as follows.

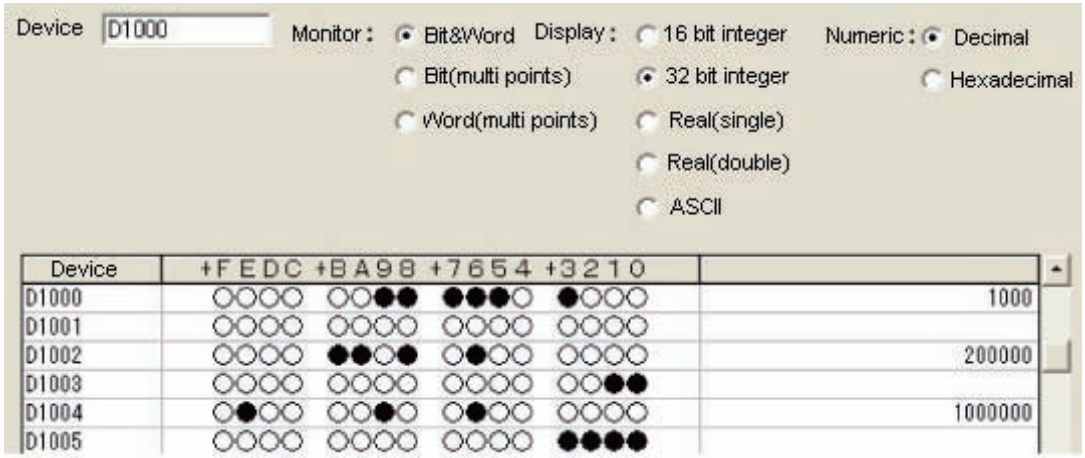


*1: Data is stored to the PLC I/O memory as follows for a decimal number per channel.

	+0	+1	+2	+3
DM1000	1000	0	16960	15

GX-Developer PLC I/O memory display image

As an example, if 3 data items, expression 0 (DATA0) = 1.000, expression 1 (DATA1) = 200.000, and expression 2 (DATA2) = 1000.000, are output from FZ4, they are stored to the PLC Link area as measurement data as follows.



*1: Data is stored to the PLC I/O memory as follows for a decimal number per channel.

	+0	+1	+2	+3	+4	+5
DM1000	1000	0	3392	3	16960	15

Operation log input information list

This is a list of operation log input information. For the operation log format, see below.
Reference: ▶Operation Log Format (p.127)

Name	Description	References
AddGlobalData	Adds global data.	Reference: ▶ Details (p.405)
AddSystemData	Adds system data.	Reference: ▶ Details (p.407)
AssignProc	Assigns units to measurement flows.	---
AssignUnit	Registers a processing unit.	Reference: ▶ Details (p.415)
BeginSetupUnit	Enters into the unit setting screen.	---
ChangeScene	Switches to the specified scene.	Reference: ▶ Details (p.423)
ChangeSceneGroup	Switches to the specified scene group.	Reference: ▶ Details (p.424)
CheckAccessControl	Checks the operation authority.	Reference: ▶ Details (p.867)
ClearImageLogCount	Clears the logging count information.	Reference: ▶ Details (p.867)
ClearMeasureData	Clears the measurement results for the processing unit.	Reference: ▶ Details (p.429)
ClearMeasureData_S	Clears the measurement results for the processing unit.	---
ClearScene	Clears the scene.	Reference: ▶ Details (p.430)
ClearSceneGroup	Clears a scene group.	Reference: ▶ Details (p.431)
ClearSecurityData	Clears security related parameters.	Reference: ▶ Details (p.868)
ClearSettingData	Clears setting data.	Reference: ▶ Details (p.868)
ConvertImageFileF	Converts an image file.	---
ConvertImageFileM	Converts an image file.	---
CopyFile	Copies a file.	Reference: ▶ Details (p.869)
CopyScene	Copies scene data.	Reference: ▶ Details (p.440)
CopySceneGroup	Copies a scene group.	Reference: ▶ Details (p.441)
CopyUnit	Copies a processing unit.	Reference: ▶ Details (p.442)
CopyUnit2	Copies a processing unit.	---
CopyUnitFigure	Copies the figure data for a processing unit.	Reference: ▶ Details (p.444)
CopyUnitModel	Copies the model data for a processing unit.	Reference: ▶ Details (p.448)
CreateDirectory	Creates a directory.	Reference: ▶ Details (p.869)
DeleteUnit	Deletes a processing unit.	Reference: ▶ Details (p.460)
DeleteUserAccount	Deletes a user account.	Reference: ▶ Details (p.869)
EndSetupUnit	Closes the unit setting screen.	---
ExitFzProcess	Exits an FZ process.	Reference: ▶ Details (p.519)
GetAll	Executes batch input of parallel I/O and other terminal states.	Reference: ▶ Details (p.528)
GetDensityProfile	Acquires the image density information.	Reference: ▶ Details (p.870)
GetGlobalData	Acquires global data.	Reference: ▶ Details (p.530)
GetHistgram	Acquires the image histogram.	Reference: ▶ Details (p.870)
GetImageCountM	Acquires the image count.	---
GetImageLogInfo	Acquires the specified image logging information.	Reference: ▶ Details (p.871)
GetIoModuleIdent	Acquires the identifier name for an I/O module.	Reference: ▶ Details (p.871)
GetLanguageIdent	Acquires the language identifier.	---

Name	Description	References
GetMeasureOut	Acquires the measurement result output yes/no.	Reference: ▶ Details (p.536)
GetPort	Executes individual input of parallel I/O terminal states.	Reference: ▶ Details (p.542)
GetSceneCount	Acquires the count of usable scenes.	---
GetSceneDescription	Acquires the count of usable scenes.	---
GetSceneGroupCount	Acquires the count of the valid scene groups on the memory card.	---
GetSceneGroupNo.	Acquires the current scene group No.	---
GetSceneGroupTitle	Acquires the scene group title name.	---
GetSceneMaker	Acquires the scene maker name.	---
GetSceneNo	Acquires the current scene No.	---
GetSceneTitle	Acquires the scene title name.	---
GetSecurityParameter	Acquires security related parameters.	Reference: ▶ Details (p.872)
GetSystemData	Acquires the system data.	Reference: ▶ Details (p.546)
GetUnitDataN	Acquires processing unit data.	---
GetUnitDataN2	Acquires processing unit data.	---
GetUnitDataS	Acquires processing unit data.	---
GetUnitDataS2	Acquires processing unit data.	---
GetUnitFigure	Acquires processing unit figure data.	Reference: ▶ Details (p.554)
GetUnitItemIdent	Acquires the identifier name for a processing item.	---
GetUnitItemIdent2	Acquires the identifier name for a processing item.	---
GetUnitTitle	Acquires a processing unit title name.	---
GetUnitTitle2	Acquires a processing unit title name.	---
ImageLogging	Executes image logging.	---
ImageUpdate	Updates image data.	Reference: ▶ Details (p.568)
InsertUnit	Inserts a processing unit.	Reference: ▶ Details (p.573)
IoInitialize	Executes I/O initialization.	Reference: ▶ Details (p.872)
LoadBackupData	Load the System + Scene group data	Reference: ▶ Details (p.599)
LoadProc	Executes load processing for unit data.	---
LoadScene	Loads scene data.	Reference: ▶ Details (p.601)
LoadSceneGroup	Reads scene group data.	Reference: ▶ Details (p.603)
LoadSceneGroup0	Executes load processing for scene group data.	---
LoadSecurityData	Loads security related settings.	Reference: ▶ Details (p.872)
LoadSystemData	Loads system data.	Reference: ▶ Details (p.605)
LoadUnitData	Loads processing unit data.	Reference: ▶ Details (p.607)
LockMeasureStop	Sets measurement trigger input to disabled.	Reference: ▶ Details (p.873)
Measure	Executes measurement processing.	Reference: ▶ Details (p.612)
MeasureInit	Executes unit measurement initialization processing.	---
MeasureOut	Executes measurement results display processing.	---
MeasureProc	Executes unit measurement processing.	Reference: ▶ Details (p.616)
MeasureStart	Sets measurement trigger input to enabled.	Reference: ▶ Details (p.617)
MeasureStop	Sets measurement trigger input to disabled.	Reference: ▶ Details (p.619)
MeasureStop2	Sets measurement trigger input to disabled.	---

Name	Description	References
MoveUnit	Moves a processing unit.	Reference: ▶ Details (p.625)
NonstopSync	Synchronizes the data for non-stop adjustment settings.	Reference: ▶ Details (p.873)
PutAll	Executes batch input of parallel I/O and other terminal states.	Reference: ▶ Details (p.643)
PutPort	Executes individual input of parallel I/O and other terminal states.	Reference: ▶ Details (p.645)
RaiseOptionEvent	Issues an option event.	Reference: ▶ Details (p.647)
ReceiveData	Executes byte type array data input.	Reference: ▶ Details (p.651)
Remeasure0	Executes remeasurement processing.	---
Remeasure1	Executes remeasurement processing.	---
RemoveFile	Removes a file or directory.	Reference: ▶ Details (p.873)
RenameFile	Changes a file name or directory name.	Reference: ▶ Details (p.873)
RenumProc	Executes processing for unit number change.	---
RunOut	Outputs the RUN state.	Reference: ▶ Details (p.667)
SaveBackupData	Save the System + Scene Group Data	Reference: ▶ Details (p.669)
SaveData	Saves data to the controller.	Reference: ▶ Details (p.670)
SaveImage	Executes save processing for image data.	Reference: ▶ Details (p.671)
SaveProc	Executes save processing for unit data.	---
SaveScene	Saves scene data.	Reference: ▶ Details (p.675)
SaveSceneGroup	Saves scene group data.	Reference: ▶ Details (p.677)
SaveSecurityData	Saves security related settings.	Reference: ▶ Details (p.874)
SaveSystemData	Saves system data.	Reference: ▶ Details (p.679)
SaveSystemData0	Saves system data.	---
SaveUnitData	Saves processing unit data.	Reference: ▶ Details (p.681)
ScreenCapture	Captures the screen.	Reference: ▶ Details (p.693)
SendData	Executes byte type array data output.	Reference: ▶ Details (p.696)
SendString	Sends a character string.	Reference: ▶ Details (p.698)
SetAccessControl	Sets operation restrictions.	Reference: ▶ Details (p.874)
SetDisplayUnitNo	Sets the specified processing unit number used for image/ text display.	Reference: ▶ Details (p.699)
SetGlobalData	Sets global data.	Reference: ▶ Details (p.702)
SetImageWindow	Sets the image display window.	Reference: ▶ Details (p.704)
SetJudgeWindow	Sets the display attributes for the judgment results display window.	Reference: ▶ Details (p.874)
SetMeasureOut	Sets whether or not measurement results are output.	Reference: ▶ Details (p.708)
SetSceneDescription	Sets the scene description.	Reference: ▶ Details (p.715)
SetSceneGroupTitle	Sets the scene group title name.	Reference: ▶ Details (p.717)
SetSceneMaker	Sets the scene maker name.	Reference: ▶ Details (p.719)
SetSceneTitle	Sets the scene title name.	Reference: ▶ Details (p.721)
SetSecurityParameter	Sets security related parameters.	Reference: ▶ Details (p.875)
SetSystemData	Sets the system data.	Reference: ▶ Details (p.725)
SetTextWindow	Sets the text display window.	Reference: ▶ Details (p.729)

Name	Description	References
SetTimeWindow	Sets the display attributes for the measurement processing time display window.	Reference: ▶ Details (p.875)
SetUnitDataN	Sets processing unit data.	---
SetUnitDataN2	Sets processing unit data.	---
SetUnitDataS	Sets processing unit data.	---
SetUnitDataS2	Sets processing unit data.	---
SetUnitFigure	Sets the figure data for a processing unit.	Reference: ▶ Details (p.733)
SetUnitTitle	Sets a processing unit title name.	Reference: ▶ Details (p.737)
SetUserAccount	Sets a user account.	Reference: ▶ Details (p.875)
StartRemoteOperation	Starts remote operation.	---
SyncData	Synchronizes the setting data between Core RA processes.	Reference: ▶ Details (p.876)
SystemReset	Executes a system reset.	Reference: ▶ Details (p.754)
UnLockMeasureStop	Sets measurement trigger input to enabled.	Reference: ▶ Details (p.876)
UserLogin	Logs in as the specified user.	Reference: ▶ Details (p.876)
UserLogout	Logs out.	Reference: ▶ Details (p.877)

Checking Operation Restrictions

CheckAccessControl

Parameters

name	Operation identifier name

Return value

For an operation that is not permitted for the logged in user, returns 0.
For an operation that is permitted for the logged in user, returns value other than 0.

Clearing the Logging Count Information.

ClearImageLogCount

Parameters

None.

Return value

If the image logging count is cleared successfully, returns 0.
Otherwise, returns value other than 0.

Clearing Security Related Parameters

ClearSecurityData

Parameters

_userName	User name executed
_password	User password executed
mode	Data to be cleared Bit 0: User account Bit 1: Operation restriction contents Bit 2: Other

Return value

If it succeeded, returns 0.
Otherwise, returns value other than 0.

Clearing Setting Data in a Batch

ClearSettingData

Parameters

None.

Return value

If it succeeded, returns 0.
Otherwise, returns value other than 0.

Copy File

CopyFile

Parameters

srcFileName	Name of file to copy from
destFileName	Name of destination file

Return value

If the file is copied successfully, returns 0.
Otherwise, returns value other than 0.

Directory Creation

CreateDirectory

Parameters

directoryName	Name of directory to create

Return value

If the directory is created successfully, returns 0.
Otherwise, returns value other than 0.

Deleting a User Account

DeleteUserAccount

Parameters

_userName	User name executed
_password	User password executed
userName	Name of user to be deleted

Return value

If it succeeded, returns 0.
Otherwise, returns value other than 0.

Getting Image Density Information

GetDensityProfile

Parameters

unitNo	Image input related and image conversion related processing item processing unit number
imageNo	Image number
imageChannel	Image channel number (Specifies RGB, R: 0, G:1, B: 2. For monochrome image, any value is OK)
rectangle	Target rectangular area. The maximum size to acquire density is 4096 pixels. Keep the size to no more than 4096.
profile	Density value storage array Resized when functions are executed.

Return value

If the density is acquired successfully, returns 0.
Otherwise, returns value other than 0.

Getting an Image Histogram

GetHistgram

Parameters

unitNo	Image input related and image conversion related processing item processing unit number
imageNo	Image number
imageChannel	Image channel number (Specifies RGB, R: 0, G:1, B: 2. For monochrome image, any value is OK)
rectangle	Target rectangular area. The maximum size to acquire density is 4096 pixels. Keep the size to no more than 4096.
histgram	Histogram array Resized when functions are executed.

Return value

If the density is acquired successfully, returns 0.
Otherwise, returns value other than 0.

Getting the Specified Image Logging Information.

GetImageLogInfo

Parameters

kind	Type of image logging information to acquire • ImageLogMaxCount: Maximum number of images that can be logged in controller memory • ImageLogCount: Count of logging images held in the controller memory

Return value

Returns the specified image logging information.

Getting an I/O Module Identifier Name

GetIoModuleIdent

Parameters

ioModuleNo	I/O module number 0: Registers ParallelIo (Parallel I/O). (Fixed) 1: Registers any of the following I/O modules in the module communication system. • SerialNormal: Serial non-procedure communication • SerialNormal2: Serial non-procedure communication (Fxxx series method) • SerialPlcLink: Serial PLC link (Omron PLC) • SerialPlcLinkM: Serial PLC link (Mitsubishi PLC) 2: Registers any of the following I/O modules for Ethernet communication system • TcpNormal: TCP non-procedure communication • UdpNormal: UDP non-procedure communication • UdpNormal2: UDP non-procedure communication (FXXX series method) • UdpPlcLink: UDP PLC link (Omron PLC) • UdpPlcLinkM: UDP PLC link (Mitsubishi PLC) 3: Registers one of the following I/O modules or no registration. • EtherCAT: EtherCAT communication • EtherNetIP: EtherNet/IP communication

Return value

If the I/O module identifier name is acquired successfully (character string), returns 0.
Otherwise, returns value other than 0.

Getting Security Related Parameters

GetSecurityParameter

Parameters

_userName	User name executed
_password	User password executed
dataIdent	Data identifier name userLoginTimeout: Login timeout time (minutes)
data	Acquisition data storage variable

Return value

If it succeeded, returns 0.
Otherwise, returns value other than 0.

Executing I/O Initialization

IoInitialize

Parameters

ioIdent	Initializing I/O module identifier name • ParallelIo: Parallel I/O • SerialNormal: Serial non-procedure communication • UdpNormal: Ethernet UDP non-protocol communication

Return value

If the specified I/O is initialized successfully, returns 0.
If initialization failed, returns value other than 0.

Loading Security Related Settings

LoadSecurityData

Parameters

_userName	User name executed
_password	User password executed
fileName	Data identifier name userLoginTimeout: Login timeout time (minutes)
data	Path for ID to load

Return value

If it succeeded, returns 0.
Otherwise, returns value other than 0.

Setting Measurement Trigger Input to Disabled

LockMeasureStop

Parameters

timeout	Timeout time

Return value

If it succeeded, returns 0.
Otherwise, returns value other than 0.

Synchronizing the Data for Non-Stop Adjustment Settings

NonstopSync

Parameters

None.

Return value

If non-stop adjustment data is synchronized successfully, returns 0.
Otherwise, returns value other than 0.

File/Directory Name Change

RenameFile

Parameters

oldFileName	Old file/directory name
newFileName	New file/directory name

Return value

If the file/directory name was changed successfully, returns 0.
Otherwise, returns value other than 0.

Deleting File/Directory

RemoveFile

Parameters

pathName	Path name

Return value

If the file/directory is deleted successfully, returns 0.
Otherwise, returns value other than 0.

Saving Security Related Settings

SaveSecurityData

Parameters

_userName	User name executed
_password	User password executed
fileName	File path to save to

Return value

If it succeeded, returns 0.
Otherwise, returns value other than 0.

Setting Operation Restrictions

SetAccessControl

Parameters

_userName	User name executed
_password	User password executed
name	Operation identifier name
userGroup	User group enabled

Return value

If it succeeded, returns 0.
Otherwise, returns value other than 0.

Setting Security Related Parameters

SetJudgeWindow

Parameters

handle	Handle
locationX	X coordinate of upper left of window
locationY	X coordinate of upper right of window
fontSize	Font size

Return value

If the judgment result display window is set successfully, returns 0.
Otherwise, returns value other than 0.

Setting the Measurement Processing Time Display Window

SetSecurityParameter

Parameters

_userName	User name executed
_password	User password executed
dataIdent	Data identifier name userLoginTimeout: Login timeout time (minutes)
data	Data to set

Return value

If it succeeded, returns 0.
Otherwise, returns value other than 0.

Setting the Measurement Processing Time Display Window

SetTimeWindow

Parameters

handle	Handle
locationX	X coordinate of upper left of window
locationY	X coordinate of upper right of window
fontSize	Font size

Return value

If the measurement processing time display window is set successfully, returns 0.
Otherwise, returns value other than 0.

Setting a User Account

SetUserAccount

Parameters

_userName	User name executed
_password	User password executed
userName	Name of user to be set
userGroupNo	User group number to be set
password	Password to be set

Return value

If it succeeded, returns 0.
Otherwise, returns value other than 0.

Synchronization of Setting Data Between CoreRA Processes

SyncData

Parameters

srcCoreRANo	Source CoreRA number
destCoreRANo	Destination CoreRA number

Return value

If setting data between CoreRA processes is synchronized successfully, returns 0.
Otherwise, returns value other than 0.

Setting Measurement Trigger Input to Enabled

UnLockMeasureStop

Parameters

None.

Return value

If it succeeded, returns 0.
Otherwise, returns value other than 0.

Logging in as the Specified User.

UserLogin

Parameters

userName	User name
password	Password

Return value

If it succeeded, returns 0.
Otherwise, returns value other than 0.

Logging Out

UserLogout

Parameters

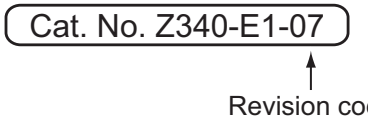
remoteOperation	Logout target

Return value

If it succeeded, returns 0.
Otherwise, returns value other than 0.

Manual Revision History

The manual revision symbol is an alphabet appended at the end of the manual number found in the bottom left-hand corner of the front or back cover.



Rev. No.	Rev. Date	Revision Contents	Software Version
01	Jul. 2013	Original production	Ver.5.00
02	Aug. 2013	Additions for lighting controllers.	Ver.5.10
03	Aug. 2013	Additions for software version upgrade.	Ver.5.10
04	Sep. 2013	Additions for software version upgrade.	Ver.5.12
05	Jan. 2014	Additions for software version upgrade.	Ver.5.20
06	Mar. 2014	Revised explanations for all macro commands.	Ver.5.20
07	Jun. 2014	Additions for software version upgrade.	Ver.5.30

OMRON Corporation **Industrial Automation Company**
Tokyo, JAPAN
Contact: www.ia.omron.com

Regional Headquarters
OMRON EUROPE B.V.
Wegalaan 67-69, 2132 JD Hoofddorp
The Netherlands
Tel: (31)2356-81-300/Fax: (31)2356-81-388

OMRON ELECTRONICS LLC
2895 Greenspoint Parkway, Suite 200
Hoffman Estates, IL 60169 U.S.A
Tel: (1) 847-843-7900/Fax: (1) 847-843-7787

OMRON ASIA PACIFIC PTE. LTD.
No. 438A Alexandra Road # 05-05/08 (Lobby 2),
Alexandra Technopark,
Singapore 119967
Tel: (65) 6835-3011/Fax: (65) 6835-2711

OMRON (CHINA) CO., LTD.
Room 2211, Bank of China Tower,
200 Yin Cheng Zhong Road,
PuDong New Area, Shanghai, 200120, China
Tel: (86) 21-5037-2222/Fax: (86) 21-5037-2200

Authorized Distributor:

Cat. No. Z340-E1-07 0614