MITSUBISHI ELECTRIC

Mitsubishi Electric Industrial Robot

CR800-D/R/Q controllers

MELFA Smart Plus
2D Vision Sensor Enhancement
Function Instruction Manual

2F-DQ510
2F-DQ511
2F-DQ520

MELFA

BFP-A3785-A

⚠ Safety Precautions

Always read the following precautions and the separate "Safety Manual" before starting use of the robot to learn the required measures to be taken.

A. These show precautions based on the Ordinance on Industrial Safety and Health (Articles 36, 104, 150, 151).

⚠CAUTION All teaching work must be carried out by an operator who has received special training. (This also applies to maintenance work with the power source turned ON.)
→Enforcement of safety training

⚠CAUTION For teaching work, prepare a work plan related to the methods and procedures of operating the robot, and to the measures to be taken when an error occurs or when restarting. Carry out work following this plan. (This also applies to maintenance work with the power source turned ON.)
→Preparation of work plan

⚠WARNING Prepare a device that allows operation to be stopped immediately during teaching work. (This also applies to maintenance work with the power source turned ON.)
→Setting of emergency stop switch

⚠CAUTION During teaching work, place a sign indicating that teaching work is in progress on the start switch, etc. (This also applies to maintenance work with the power source turned ON.)
→Indication of teaching work in progress

⚠DANGER Provide a fence or enclosure during operation to prevent contact of the operator and robot.
→Installation of safety fence

⚠CAUTION Establish a set signaling method to the related operators for starting work, and follow this method.
→Signaling of operation start

⚠CAUTION As a principle turn the power OFF during maintenance work. Place a sign indicating that maintenance work is in progress on the start switch, etc. Indication of maintenance work in progress
→Display that maintenance work is underway

⚠CAUTION Before starting work, inspect the robot, emergency stop switch and other related devices, etc., and confirm that there are no errors.
→Inspection before starting work

B. This shows precaution points given in the separate "Safety Manual".
For details, please read the text of the "Safety Manual".

⚠DANGER When automatic operation of the robot is performed using multiple control devices (GOT, programmable controller, push-button switch), the interlocking of operation rights of the devices, etc. must be designed by the customer.

⚠CAUTION Use the robot within the environment given in the specifications. Failure to do so could lead to a drop or reliability or faults. (Temperature, humidity, atmosphere, noise environment, etc.)

⚠CAUTION Transport the robot with the designated transportation posture. Transporting the robot in a non-designated posture could lead to personal injuries or faults from dropping.

⚠CAUTION Always use the robot installed on a secure table. Use in an instable posture could lead to positional deviation and vibration.

⚠CAUTION Wire the cable as far away from noise sources as possible. If placed near a noise source, positional deviation or malfunction could occur.

⚠CAUTION Do not apply excessive force on the connector or excessively bend the cable. Failure to observe this could lead to contact defects or wire breakage.

⚠CAUTION Make sure that the workpiece weight, including the hand, does not exceed the rated load or tolerable torque. Exceeding these values could lead to errors or faults.

⚠WARNING Securely install the hand and tool, and securely grasp the workpiece. Failure to observe this could lead to personal injuries or damage if the object comes off or flies off during operation.

⚠WARNING Securely ground the robot and controller. Failure to observe this could lead to malfunctioning by noise or to electric shock accidents.

⚠CAUTION Indicate the operation state during robot operation. Failure to indicate the state could lead to operators approaching the robot or to incorrect operation.

⚠WARNING When carrying out teaching work in the robot's movement range, always secure the priority right for the robot control. Failure to observe this could lead to personal injuries or damage if the robot is started with external commands.

⚠CAUTION Keep the jog speed as low as possible, and always watch the robot. Failure to do so could lead to interference with the workpiece or peripheral devices.

⚠CAUTION After editing the program, always confirm the operation with step operation before starting automatic operation. Failure to do so could lead to interference with peripheral devices because of programming mistakes, etc.

⚠CAUTION Make sure that if the safety fence entrance door is opened during automatic operation, the door is locked or that the robot will automatically stop. Failure to do so could lead to personal injuries.

⚠CAUTION Never carry out modifications based on personal judgments, or use non- designated maintenance parts.
Failure to observe this could lead to faults or failures.

⚠WARNING When the robot arm has to be moved by hand from an external area, do not place hands or fingers in the openings. Failure to observe this could lead to hands or fingers catching depending on the posture.

⚠CAUTION Do not stop the robot or apply emergency stop by turning the robot controller's main power OFF. If the robot controller main power is turned OFF during automatic operation, the robot accuracy could be adversely affected. Moreover, it may interfere with the peripheral device by drop or move by inertia of the arm.

⚠CAUTION Do not turn off the main power to the robot controller while rewriting the internal information of the robot controller such as the program or parameters. If the main power to the robot controller is turned off while in automatic operation or rewriting the program or parameters, the internal information of the robot controller may be damaged.

⚠DANGER Do not connect the Handy GOT when using the GOT direct connection function of this product. Failure to observe this may result in property damage or bodily injury because the Handy GOT can automatically operate the robot regardless of whether the operation rights are enabled or not.

⚠DANGER Do not connect the Handy GOT to a programmable controller when using an iQ Platform compatible product with the CR800-R/CR800-Q controller. Failure to observe this may result in property damage or bodily injury because the Handy GOT can automatically operate the robot regardless of whether the operation rights are enabled or not.

⚠DANGER Do not remove the SSCNET III cable while power is supplied to the multiple CPU system or the servo amplifier. Do not look directly at light emitted from the tip of SSCNET III connectors or SSCNET III cables of the Motion CPU or the servo amplifier. Eye discomfort may be felt if exposed to the light. (Reference: SSCNET III employs a Class 1 or equivalent light source as specified in JIS C 6802 and IEC60825-1 (domestic standards in Japan).)

⚠DANGER Do not remove the SSCNET III cable while power is supplied to the controller. Do not look directly at light emitted from the tip of SSCNET III connectors or SSCNET III cables. Eye discomfort may be felt if exposed to the light. (Reference: SSCNET III employs a Class 1 or equivalent light source as specified in JIS C 6802 and IEC60825-1 (domestic standards in Japan).)

⚠DANGER Attach the cap to the SSCNET III connector after disconnecting the SSCNET III cable. If the cap is not attached, dirt or dust may adhere to the connector pins, resulting in deterioration connector properties, and leading to malfunction.

⚠CAUTION Make sure there are no mistakes in the wiring. Connecting differently to the way specified in the manual can result in errors, such as the emergency stop not being released. In order to prevent errors occurring, please be sure to check that all functions (such as the teaching box emergency stop, customer emer- gency stop, and door switch) are working properly after the wiring setup is com- pleted.

⚠CAUTION Use the network equipments (personal computer, USB hub, LAN hub, etc) confirmed by manufacturer. The thing unsuitable for the FA environment (related with conformity, temperature or noise) exists in the equipments connected to USB. When using network equipment, measures against the noise, such as measures against EMI and the addition of the ferrite core, may be necessary. Please fully confirm the operation by customer. Guarantee and maintenance of the equipment on the market (usual office automation equipment) cannot be performed.

⚠CAUTION

To maintain the security (confidentiality, integrity, and availability) of the Robot and the system against unauthorized access, DoS *1 attacks, computer viruses, and other cyberattacks from unreliable networks and devices via network, take appropriate measures such as firewalls, virtual private networks (VPNs), and antivirus solutions.

Mitsubishi Electric shall have no responsibility or liability for any problems involving Robot trouble and system trouble by unauthorized access, DoS attacks, computer viruses, and other cyberattacks.

*1 DoS: A denial-of-service (DoS) attack disrupts services by overloading systems or exploiting vulnerabilities, resulting in a denial-ofservice (DoS) state.

■User's Manual Revision History

Printing Date	Manual No.	Revision Contents
2021-10-22	BFP-A3785	･First edition
2023-09-04	BFP-A3785-A	･2.3. Specifications of the vision sensor enhancement function Deleted and added the description of some operating systems. ･Added 2.3.1. Compatibility with former versions. ･Added "How to delete job files" to 3.3.3. ･Added 7.1. Using PatMax. ･Added 8.3.1. When not using PatMax. ･Added 8.3.2. When using PatMax. ･Added 8.3.3. How to use the model registration editor. ･Added 9.3.1. When not using PatMax. ･Added 9.3.2. When using PatMax. ･Changed some of other sections and corrected mistakes.

■ INTRODUCTION

Thank you for purchasing Mitsubishi Electric MELFA industrial robots.
This instruction manual explains the vision sensor enhancement function enabled by the optional MELFA Smart Plus card/card pack.

This function facilitates the start-up of 2D alignment applications with the help of the vision sensor being used.

Before using the MELFA Smart Plus card/card pack, make sure that you have read and fully understood the contents of this manual.

This manual is described on the premise that basic operations and functions of Mitsubishi Electric Industrial Robots are understood. For information on basic robot operations, refer to the "Instruction Manual/Detailed explanations of functions and operations".

■ Notation used in this manual

⚠ **Danger**	Incorrect handling may result in imminent danger, leading to death or serious injury.
⚠ **Warning**	Incorrect handling may lead to death or serious injury.
⚠ **Caution**	Incorrect handling may result in property damage, or danger leading to impairment of the user.

[CONTENTS]

1. BEFORE USE

1.1. Chapters in this manual

This manual is composed of the following chapters to explain how to use the 2D vision sensor enhancement function. For information on the functions and operations of standard robot controllers, refer to the instruction manual included with the robot controller.

Table 1-1 Descriptions in this manual

Chapter	Title	Description
1	BEFORE USE	Explains how this manual is composed.
2	SYSTEM SPECIFICATIONS	Describes system specifications of the vision sensor enhancement function.
3	CHECKING BEFORE USE	Explains product configuration and equipment that should be supplied by customer. Check that required products are present and the versions of the robot controller and RT ToolBox3 you are using.
4	USAGE	Explains steps required to construct a system that incorporates the vision sensor enhancement function. Make sure to complete every step. Details on each step are provided in Chapters 5 to 12.
5	INSTALLATION AND START-UP OF THE PRODUCT	Explains what products must be installed in the system to use the vision sensor enhancement function and start-up procedures.
6	SELECTING AN APPLICATION	Explains robot applications supported by the vision sensor enhancement function and the selection method. The subsequent settings may differ depending on the robot application.
7	CONFIGURING THE VISION SENSOR COMMUNICATION SETTINGS	Explains how to configure the communication and connection settings of the devices required in the system to use the vision sensor enhancement function.
8	CALIBRATION	Describes steps to find the positional relationship between the robot and vision or camera.
9	REGISTERING A WORKPIECE	Explains how to register a model to be recognized by the robot.
10	OPERATION SETTINGS	Explains how to define the operation of the robot.
11	REGISTERING A GRASP POSITION	Explains how to find the positional relationship to grasp (suck) the workpiece correctly when the robot is taught the correct grasp (suck) point. It explains the steps.
12	CHECKING THE OPERATION	Explains how to check the settings using the robot.
13	ACTUAL OPERATION	Explains how to customize a created robot program for the system being constructed.
14	PROGRAM SPECIFICATIONS	Describes detailed specifications of MELFA-BASIC language for the vision sensor enhancement function.
15	MAINTENANCE	Describes the procedures of backing up and restoring settings, and uninstalling the application.
16	PARAMETER SPECIFICATIONS	Describes detailed specifications of the parameters for the vision sensor enhancement function.
17	TROUBLESHOOTING	Provides information on errors and solutions for the vision sensor enhancement function.
18	TIPS	Provides information on cases that may occur during operation and explains how to take countermeasures.
19	APPENDIX	Explains how to display RT ToolBox3 option card information, operate the Operation Panel, perform position jump, and provides information on a calibration mark.

1.2. Terms used in this manual

This manual uses the following terms.

Table 1-2 Terms

Term	Explanation
Vision sensor (abbreviation: vision)	A device that can perform image processing.
Job	Vision sensor-installed image recognition program and data.
Pat Max	The name of the pattern search algorithm based on the outline information of the target object (a registered trademark of Cognex). The processing time sometimes takes longer than the normalized correlation, but it is generally highly accurate and operates stably against illumination light, focus variations, loss of information, and noise.
Hand camera	A vision or camera attached to the end of the robot arm which is used for taking measurements and recognizing workpieces.
Fixed camera	A vision or camera attached to the frame of a device or other parts which is used for taking measurements and recognizing workpieces. In this case, the vision or camera cannot be moved like a hand camera.
Working distance	A distance from the lens attachment flange surface to the target.
Calibration	In this manual, calibration means an operation to calculate the row and column for position calibration of the vision sensor and robot (robot calibration).
MELFA Smart Plus card pack MELFA Smart Plus card	A function enhancement card for CR800 robot controllers. The MELFA Smart Plus card pack allows for all the MELFA Smart Plus functions to be used. The MELFA Smart Plus card allows for one of the MELFA Smart Plus functions to be used. Vision sensor enhancement functions such as automatic robot program creation can be used. For details, refer to the "MELFA Smart Plus User's Manual".

2. SYSTEM SPECIFICATIONS

2.1. 2D vision sensor enhancement function

The 2D vision sensor enhancement function is a generic term for functions added to RT ToolBox3 so that even customers who do not have the know-how of vision alignment can set up various alignment applications easily.

To use the function, insert the optional "MELFA Smart Plus card (A-type)" into the robot controller.

There are nine types of applications available aimed at various vision alignment including pick-and-place operations and grip error correction. Fixed downward-facing, upward-facing, and side-facing vision sensors or cameras, or hand cameras (installed on the robot hand) are available according to your needs. Configuring the settings by following the instructions in the wizard will automatically create a vision program, calibration data, and robot program. When our vision series or Cognex vision sensor is used, the steps require RT ToolBox3, and do not require any other software.

You can go back to the desired setting screen without going through the wizard. This allows you to easily adjust parameters that must be finely adjusted (such as speed and stopping time). Registered mark information and calibration data that directly affect accuracy can be imported and exported. The adjustment can be repeated to improve accuracy in accordance with the actual setup.

2.2. System configuration examples

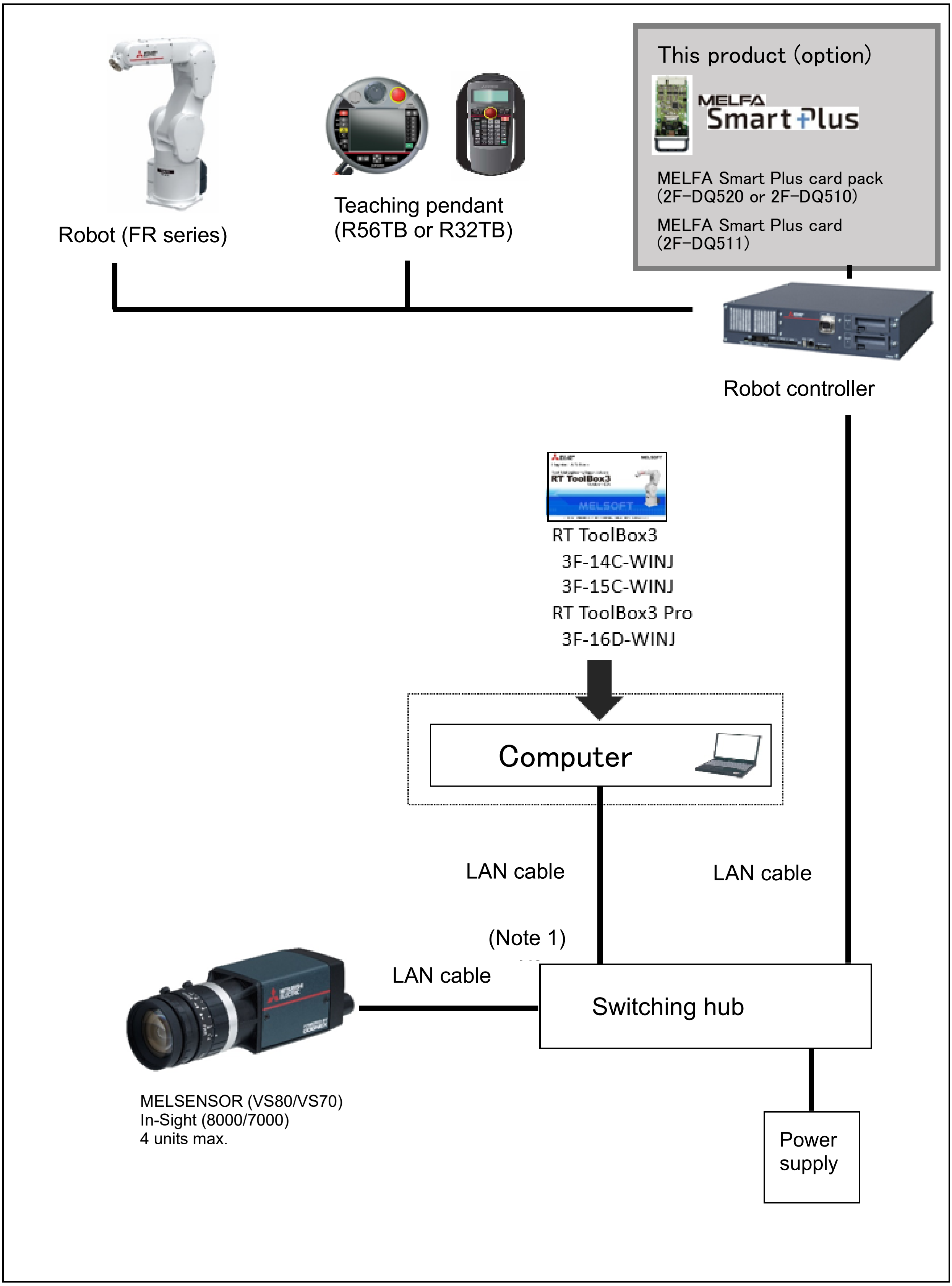


Fig. 2-1 System configuration of the vision sensor enhancement function

Note 1: Use a PoE LAN cable as a LAN cable between the camera and switching hub. When using a hand camera, use a cable that withstands bending. For use in IP65 or IP67 environments, use a cable that meets IP65 or IP67 requirements.

2.3. Specifications of the vision sensor enhancement function

Table 2-1 Specifications of the vision sensor enhancement function

<table>
<tr><th colspan="2">Item</th><th>Description</th></tr>
<tr><td colspan="2">Compatible robot[Note 1]</td><td>RV-FR, RH-FRH series
RV-CR, RH-CRH series
(Excluding the MELFA ASSISTA series)</td></tr>
<tr><td colspan="2">Robot program language</td><td>MELFA-BASIC VI (to use Function procedure)</td></tr>
<tr><td colspan="2">Robot controller</td><td>CR800 with software version C2b or later</td></tr>
<tr><td colspan="2">Option</td><td>Any of the following "MELFA Smart Plus cards" (A-type) is required.
・ 2F-DQ510 (All the A-type functions)
・ 2F-DQ511 (One of the A-type functions)
・ 2F-DQ520 (All the A-type and B-type functions)</td></tr>
<tr><td rowspan="2">Calibration</td><td>Type</td><td>N point movement-type calibration by characteristics registration</td></tr>
<tr><td>Number of calibration points</td><td>9 points max. + one rotational direction</td></tr>
<tr><td>RT ToolBox3</td><td>Supported version</td><td>Ver.1.91V or later</td></tr>
<tr><td>Computer</td><td>Operating system</td><td>Available for the following 64-bit operating systems (Not available for 32-bit operating systems)
Windows 10 Home
Windows 10 Pro
Windows 10 Education
Windows 10 Enterprise
Windows 10 IoT Enterprise 2016 LTSB
Windows 11 Home
Windows 11 Pro
Windows 11 Enterprise</td></tr>
<tr><td rowspan="7">Vision sensor</td><td>Manufacturer (with sample communication)</td><td>Mitsubishi Electric, Cognex</td></tr>
<tr><td>Recommended product</td><td><Mitsubishi Electric: MELSENSOR>
VS70M-□□□(-□□)
VS80M-□□□(-□□)
<Cognex>
IS7□□□M-□□□(-□□)
IS8□□□M-□□□(-□□)</td></tr>
<tr><td>Conditions</td><td>Equipped with a function to recognize the registered mark and report search position information (X, Y, θ).</td></tr>
<tr><td>Connection method</td><td>Ethernet</td></tr>
<tr><td>Number of connections</td><td>4 max.</td></tr>
<tr><td rowspan="2">Installation type</td><td>Fixed camera (downward, upward, sideways)</td></tr>
<tr><td>Hand camera (installed on the robot hand)</td></tr>
<tr><td>Hand [Note 2]</td><td>Compatible hand</td><td>Only a single hand compatible with ON/OFF control of the robot hand dedicated output signal (example 1) or general-purpose output signal (example 2) is available.
For pneumatic hands, both single solenoid type and double solenoid type are available.
・ Example 1
Open/close type pneumatic hands and suction hands used with the optional solenoid valve
・ Example 2
Open/close type pneumatic hands and suction hands that are connected by external wiring with robot's I/O control (parallel I/O, CC Link, etc.)</td></tr>
</table>

Application	Number of supported applications	Basic applications (N types), applied applications (N types)
	Basic function	Pick-up position control, placement position control, grip error correction function, pallet defining function (up to 99×99), waiting position and operating speed adjustment function, calibration data import and export function
	Number of workpiece types	5 max.

Note 1: The RV-AS series is not available.

Note 2: The multifunctional electric gripper option is not available.

2.3.1. Compatibility with former versions

The 2D vision enhancement function for RT ToolBox3 Ver. 2.20W or later is backward compatible with that for 1.91V.

All projects, robot programs, and vision sensor jobs created in Ver. 1.91V to 2.10L can be used as they are.

3. CHECKING BEFORE USE

3.1. Items in package

This product includes the following items as standard. Check if the package includes the following items.
* If any item is missing, please contact the branch office or distributor from which you purchased the product.

Table 3-1 List of standard items for the vision sensor enhancement function

<table>
<tr><th>No.</th><th colspan="2">Item</th><th>Model</th><th>Quantity</th><th>Appearance</th></tr>
<tr><td>1</td><td colspan="2">Instruction manual (this CD-ROM)</td><td>BFP-A3785</td><td>1</td><td></td></tr>
<tr><td rowspan="3">2</td><td rowspan="2">MELFA Smart Plus card pack[Note 1]</td><td>A type</td><td>2F-DQ510</td><td rowspan="3">Any one of the products</td><td rowspan="3"></td></tr>
<tr><td>AB type</td><td>2F-DQ520</td></tr>
<tr><td>MELFA Smart Plus card[Note 1]</td><td>A type</td><td>2F-DQ511</td></tr>
</table>

Note 1: Either the MELFA Smart Plus card pack or the MELFA Smart Plus card is required.

3.2. Devices supplied by customer

Besides this product, there are some devices that should be supplied by customer to configure a system. At least the devices shown in Table 3-2 List of devices supplied by customer are required.

Table 3-2 List of devices supplied by customer

No.	Item	Specifications	Quantity	Remarks
1	Computer to operate RT ToolBox3	RT ToolBox3 has been installed. * RT ToolBox3 can be operated.	1	For settings
2	Switching hub[Note 1]	1000BASE-T or higher PoE supported	1	A device that has been validated ・ Injector manufactured by PHIHONG (POE29U-1AT, POE30U-560(G)) ・ Switching hub manufactured by NETGEAR (GS108P)
3	LAN cable[Note 2] (others)	Category 5e or higher	2	・ Used between items No.1 and 2 ・ Used between the robot controller and item No.2 * Required separately to connect the vision sensor
4	Vision sensor installation jig	-	1 set	A jig to fix the vision (item No.5)
5	Vision sensor[Note 3]	<Mitsubishi Electric: MELSENSOR> VS70/80 series Applicable models: VS70M-□□□(-□□) VS80M-□□□(-□□)	As required	Manufactured by Mitsubishi Electric
		＜Cognex＞ In-Sight 7000/8000 series Applicable models: IS7□□□M-□□□(-□□) IS8□□□M-□□□(-□□)		Manufactured by Cognex
6	Ethernet cable	Cable between the vision sensor and switching hub		-
7	Calibration mark	-		Required for calibration. Prepare a mark that can be recognized easily. The actual workpiece can be used for calibration. For details, refer to "8 CALIBRATION".

Note 1: Use a switching hub that supports Gigabit Ethernet and PoE.
Note 2: It is recommended to use a PoE LAN cable. When using a hand camera, use a cable that withstands bending.
Note 3: Use the vision sensor firmware version 5.07.5 or later (Mitsubishi Electric MELSENSOR) or 5.08.0 or later (Cognex In-Sight). For information on how to check the firmware version, refer to "3.3.3 Vision sensor".

3.3. Software version

To use the vision sensor enhancement function, the relevant software needs to support the function. Check the version before using the function.

3.3.1. Robot controller

Item	Model	Supported version
Robot controller[Note 1]	CR800-D/CR800-R/CR800-Q	Ver.C2b or later[Note 2]

Note 1: For MELFA-BASIC V, the robot program (using Function procedure) that is output by this function is not available. Select "MELFA-BASIC Ⅵ" as a robot language.

Note 2: If the version you are using is earlier than the above, this function cannot be used because the "Vision sensor enhancement function" icon does not appear in the RT ToolBox3 workspace.

3.3.2. Supported software

Item	Model	Supported version
MELSOFT RT ToolBox3	3F-14C-WINJ 3F-14C-WINE	Ver.1.91V or later[Note 1]
MELSOFT RT ToolBox3 mini	3F-15C-WINJ 3F-15C-WINE	
MELSOFT RT ToolBox3 Pro	3F-16D-WINJ 3F-16D-WINE	

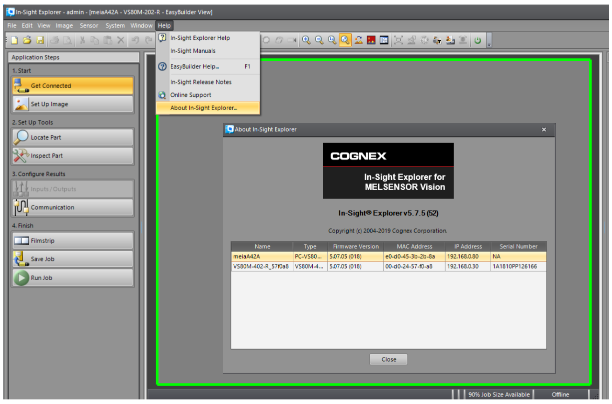
Note 1: If the version you are using is earlier than the above, the "Vision sensor enhancement function" icon will not appear in the RT ToolBox3 workspace.

3.3.3. Vision sensor

・ Vision sensor firmware version

Manufacturer	Supported version
Mitsubishi Electric MELSENSOR	5.07.5 or later
Cognex In-Sight	5.08.0 or later

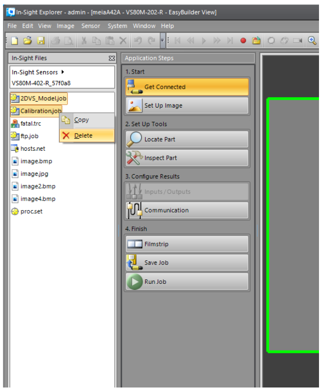
If the firmware has an unsupported version, update the firmware. The firmware version can be checked from Help of In-Sight Explorer.



● How to update the firmware

・ Start [In-Sight Explorer].
・ Establish connection with the vision sensor.
・ Select [Update Firmware] from the [System] menu.
・ Check the Version field of the displayed vision sensor, select the check box of the applicable vision sensor, then [Next] → [Update Firmware] to update the firmware.

Before using the vision sensor, it is recommended to delete job files.



●How to delete job files

・ Start [In-Sight Explorer].
・ Establish connection with the vision sensor.
・ Select [In-Sight Files] from the [View] menu.
・ Double-click the desired vision sensor in the [In-Sight Files] pane.
・ Select the following files from the list.
  2DVS_Model.job
  Calibration.job
・ Right-click on the files slected at previous step to display the context menu and select [Delete].

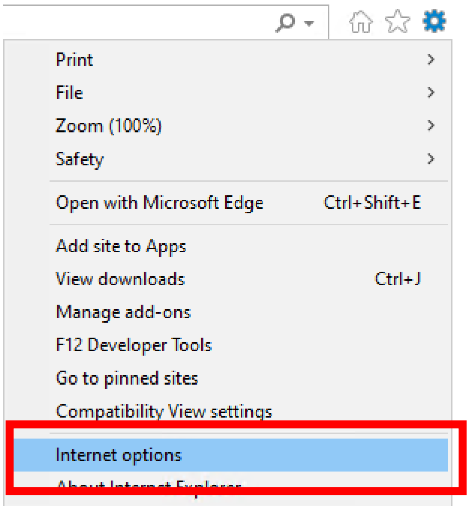
3.4. Computer settings

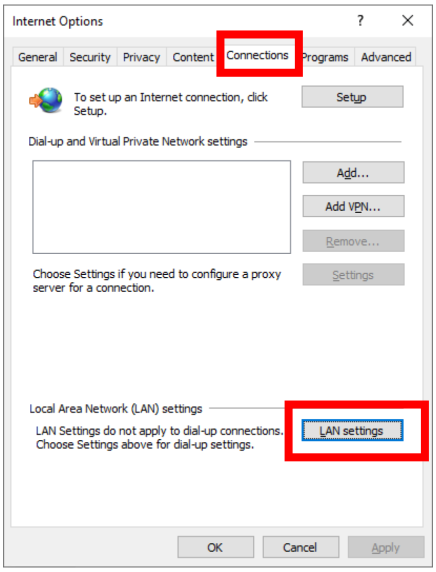
If an internet proxy is configured on the computer you are using, connection to the vision sensor may not be established or live images may not appear. In this case, use the exception settings shown below.
Change the proxy settings in "Internet options" of the computer on Internet Explorer or Microsoft Edge.

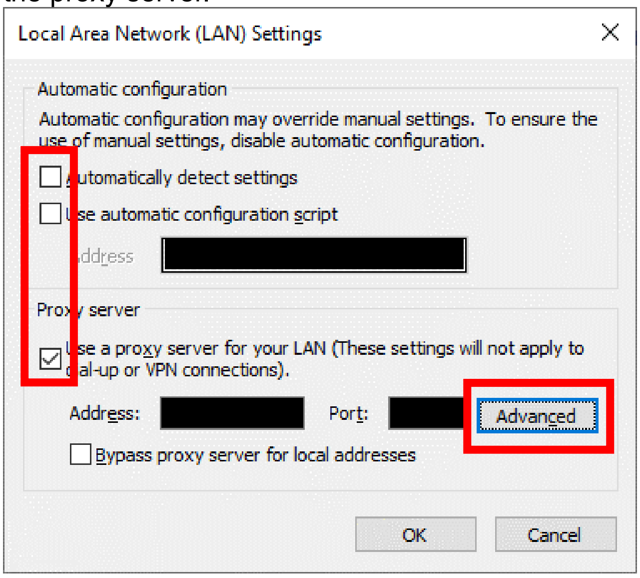
3.4.1. For Internet Explorer

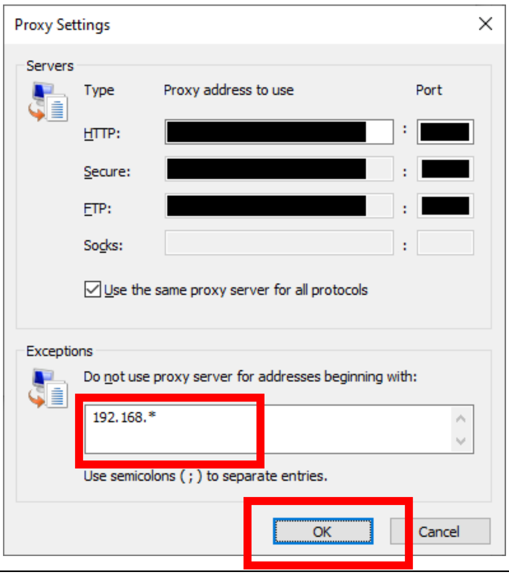
Set the "Exceptions" of the proxy using the following steps:

<table>
<tr><th>[1] Opening the Internet options.</th><th>[2] Configuring the LAN settings.</th></tr>
<tr><td>Click the Internet Explorer settings icon, then [Internet options].


</td><td>Click the [Connections] tab, then the [LAN settings] button.

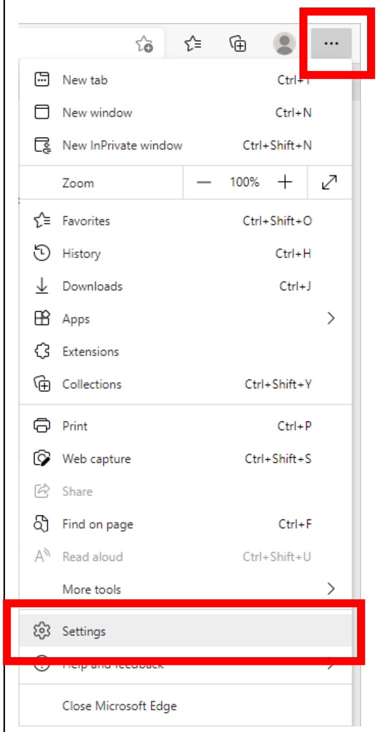
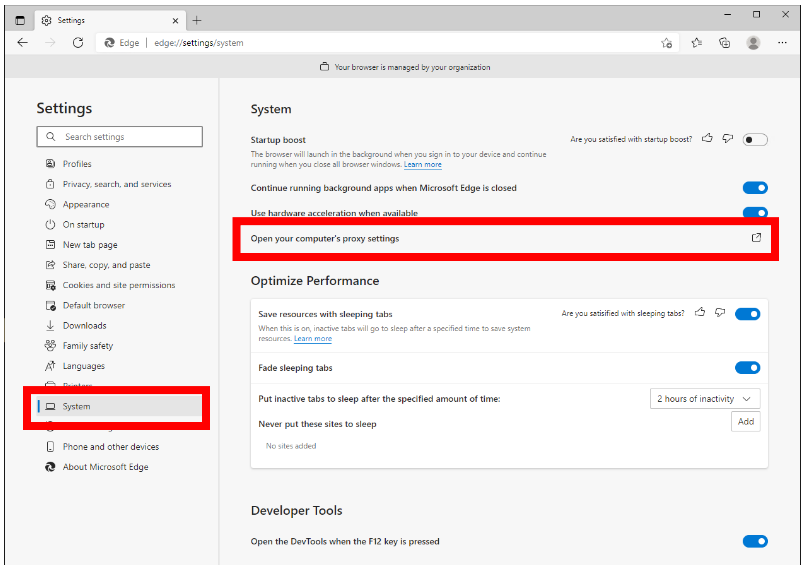

</td></tr>
<tr><th>[3] Configuring the proxy server advanced settings.</th><th>[4] Specifying exception addresses</th></tr>
<tr><td>Select the check box enclosed in the frame as shown below, then click the [Advanced] button of the proxy server.

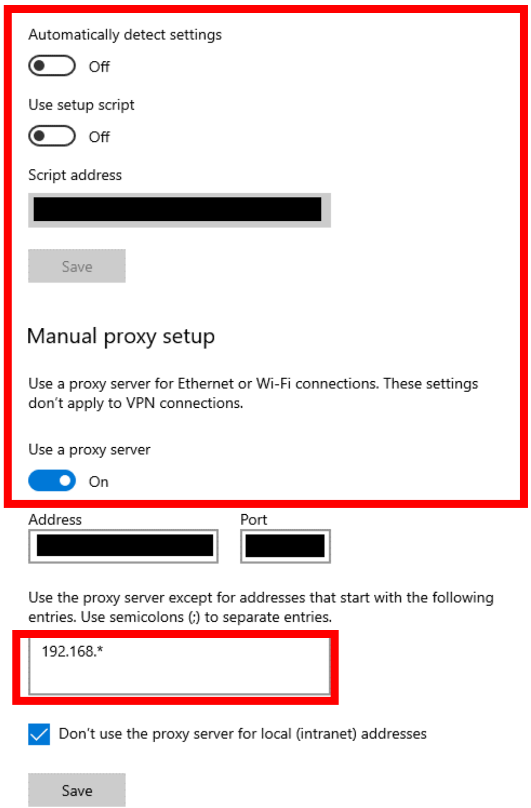

</td><td>Enter exception addresses based on the computer IP address or robot system IP address, then click the [OK] button.


</td></tr>
</table>

3.4.2. For Microsoft Edge

Set the "Exceptions" of the proxy using the following steps:

[1] Opening the Settings	[2] Opening the proxy settings
Click the [･･･] icon, then [Settings].	Click [System], then [Open your computer's proxy settings].
	

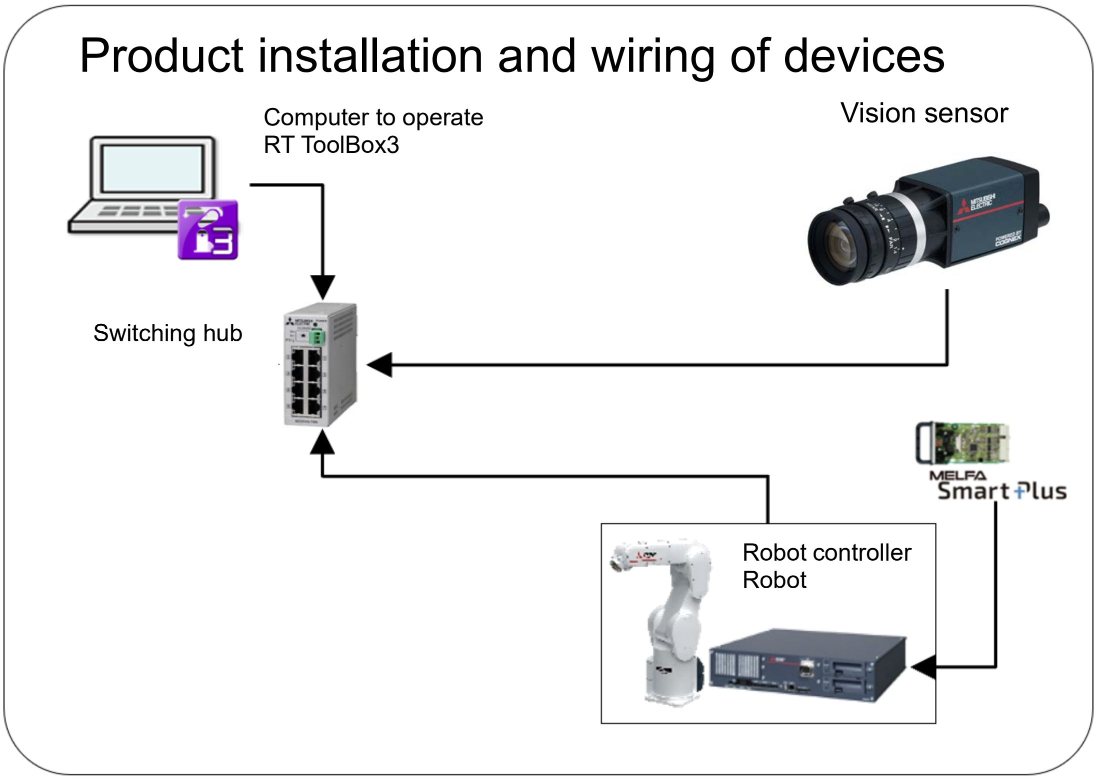
[3] Specifying exception addresses	
Turn on the setting enclosed in the frame as shown in the screen on the right. Enter exception addresses based on the computer IP address or robot system IP address in the field, then click the [Save] button.	

4. USAGE

A vision program, calibration data, and robot program can be automatically created by using the 2D vision sensor enhancement function and performing settings using the following steps.

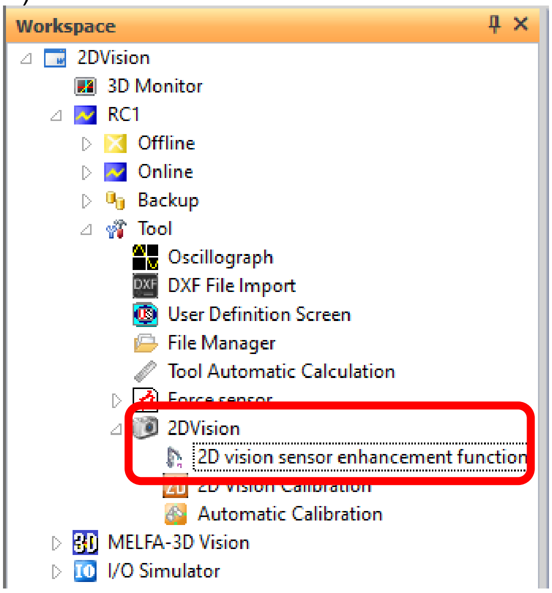
1. INSTALLATION AND START-UP OF THE PRODUCT (5 INSTALLATION AND START-UP OF THE PRODUCT**)**

- Attaching the vision sensor to the robot or stand (5.1.4 Vision sensor installation)
- Installing RT ToolBox3 on a computer to operate the software
- Wiring devices (robot controller, computer to operate RT ToolBox3, vision sensor, and robot) as shown below
- Inserting the MELFA Smart Plus card into the robot controller (5.2 Inserting the MELFA Smart Plus card)



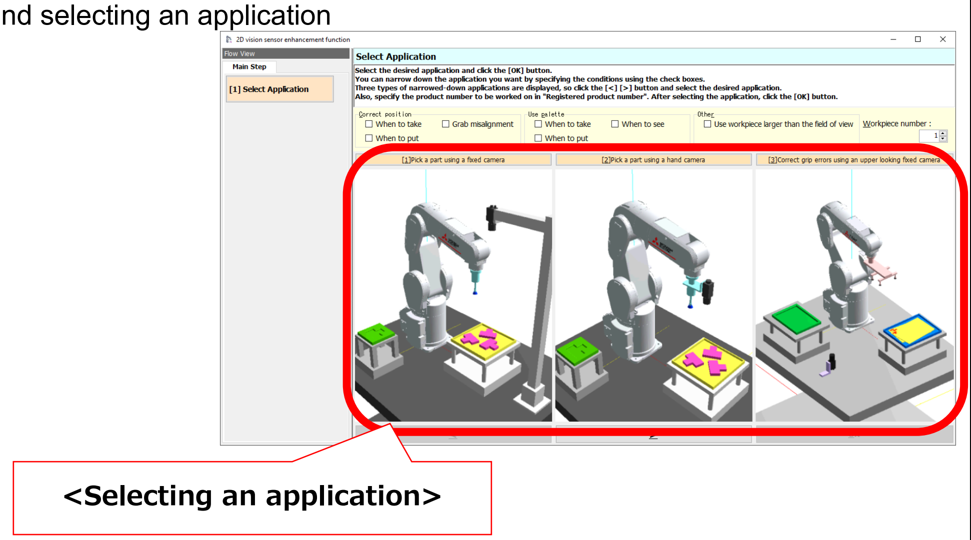
2. STARTING THE VISION SENSOR ENHANCEMENT FUNCTION → (5 INSTALLATION AND START-UP OF THE PRODUCT**)**

- Starting RT ToolBox3 from the desktop shortcut or start menu (5.3 Starting RT ToolBox3)
- Opening the setting screen of the vision sensor enhancement function from the RT ToolBox3 workspace (5.5 Starting the vision sensor enhancement function)

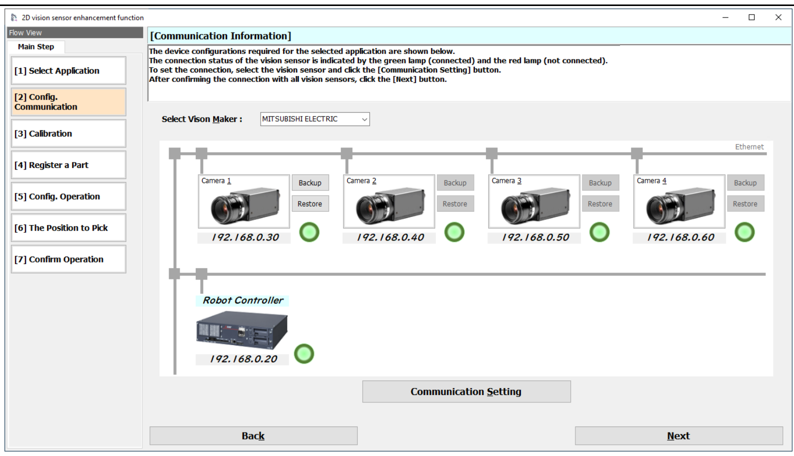


3. SELECTING AN APPLICATION → (6 SELECTING AN APPLICATION)

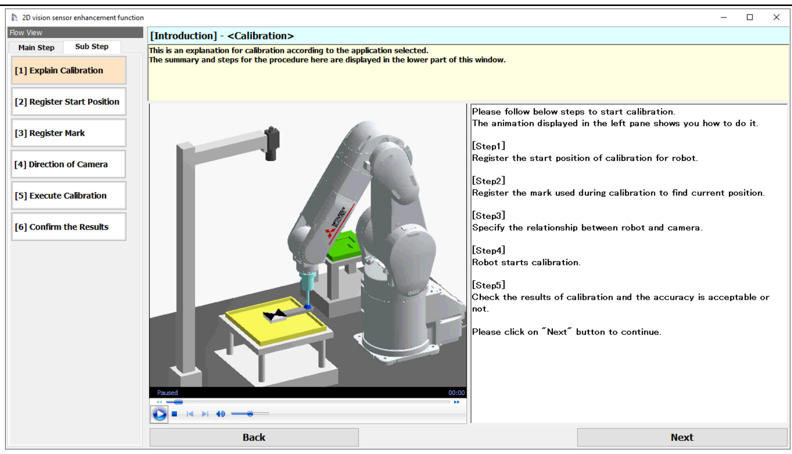
- Checking the animation and selecting an application



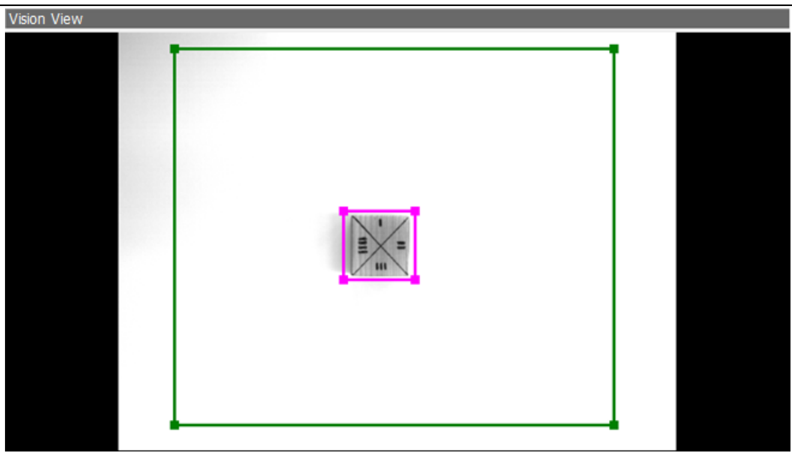
4. DEVICE COMMUNICATION AND CONNECTION SETTINGS
→ (7 CONFIGURING THE VISION SENSOR COMMUNICATION SETTINGS)
Making devices ready
- Finding vision sensors on the same network and configuring the connection settings
- Making the robot and vision sensor ready to communicate with each other



5. CALIBRATION → (8 CALIBRATION)
- Registering an imaging position (8.2 Registering the calibration start position and setting the vision)
- Displaying vision live images
- Adjusting the lens focus and aperture
- Adjusting the field of view
- Setting vision sensor parameters
- Registering a calibration mark (8.3 Registering a calibration mark)
- Setting conditions
- Performing calibration (8.5 Performing calibration)
- Checking results (8.6 Results display)

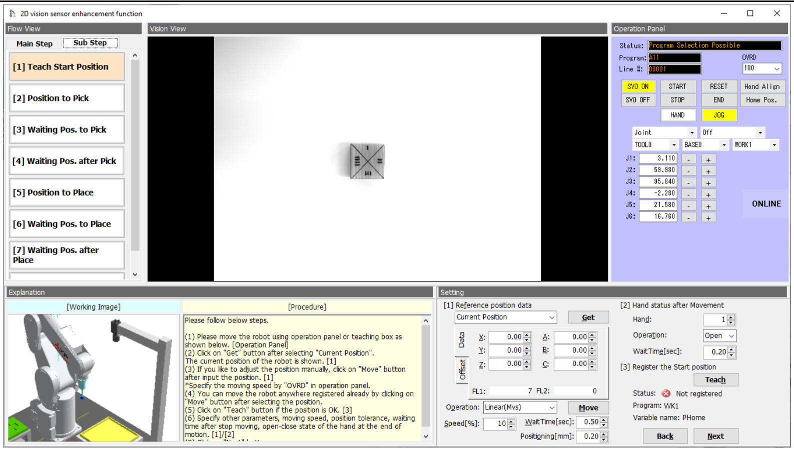


6. REGISTERING A WORKPIECE →
(9 REGISTERING A WORKPIECE)
- Registering an imaging (trigger) position
- Creating a vision program
(Registering the workpiece shape)
(Setting recognition conditions)



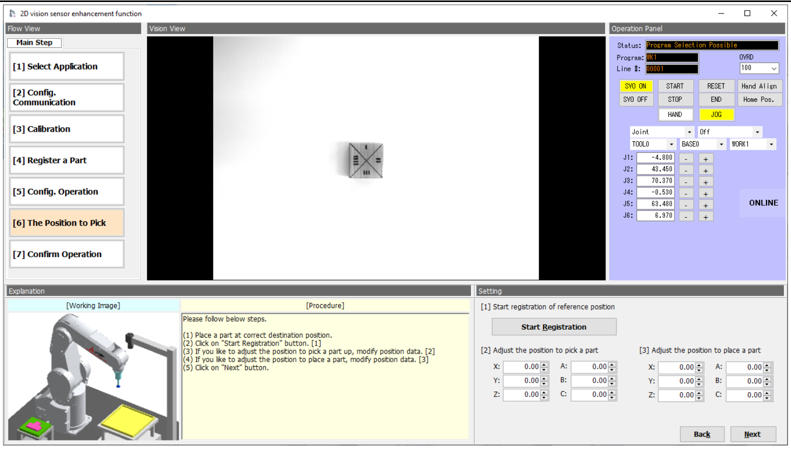
7. OPERATION SETTINGS → (10 OPERATION SETTINGS)

- Registering the starting point
- Registering a pick-up movement
- Registering a placing movement
- Setting a movement to the end point
- Defining a pallet as necessary



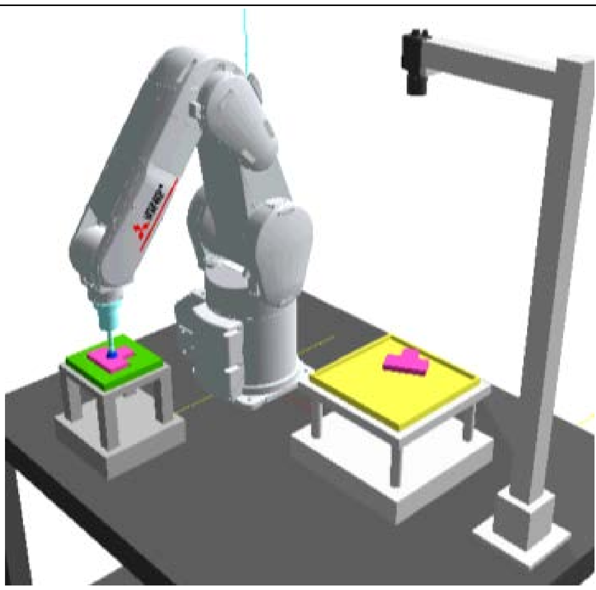
8. REGISTERING A GRASP POSITION → (11 REGISTERING A GRASP POSITION)

- Running the robot program to find the positional relationship between the taught position and the position of the workpiece recognized by the vision



9. CHECKING THE OPERATION → (12 CHECKING THE OPERATION)

- Checking a specified operation
- Checking a series of operations
- Going back to the desired setting screen and making adjustment as necessary



5. INSTALLATION AND START-UP OF THE PRODUCT

5.1. Installation and wiring of devices

This chapter provides information on the connection and wiring between the robot controller and the vision sensor.

5.1.1. CR800-Q

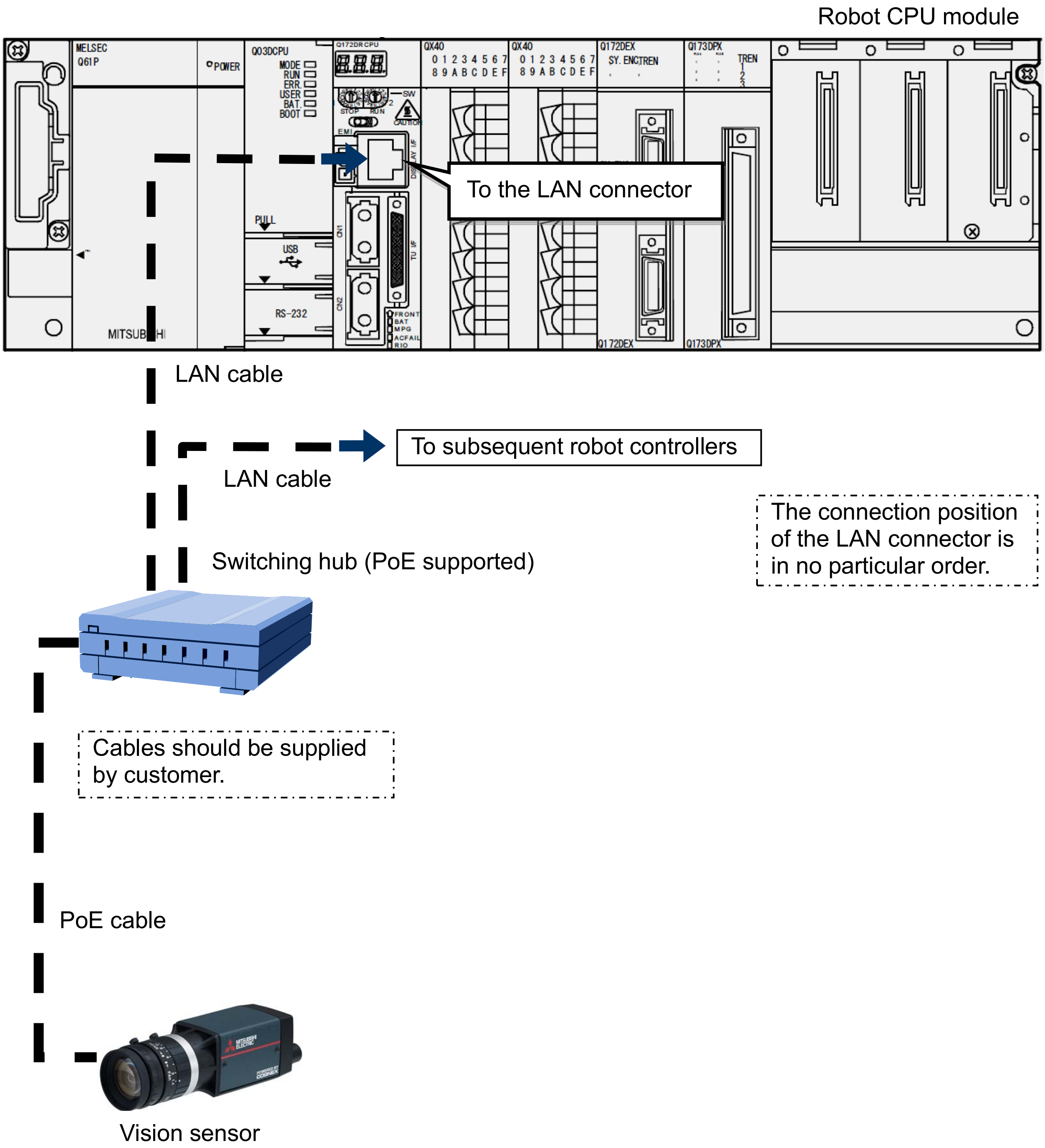


Fig. 5-1 Device wiring (CR800-Q)

5.1.2. CR800-D

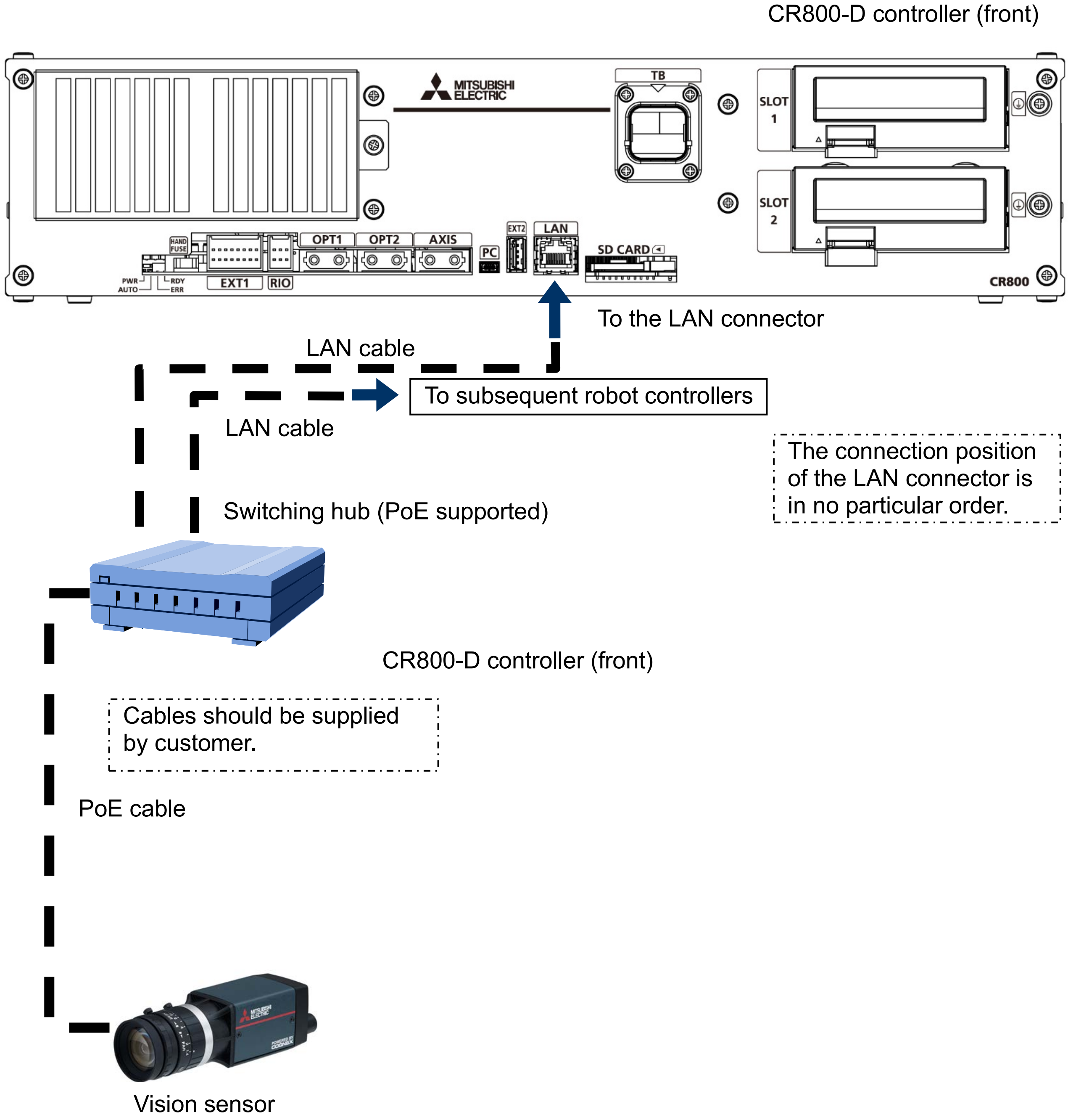


Fig. 5-2 Device wiring (CR800-D)

5.1.3. CR800-R

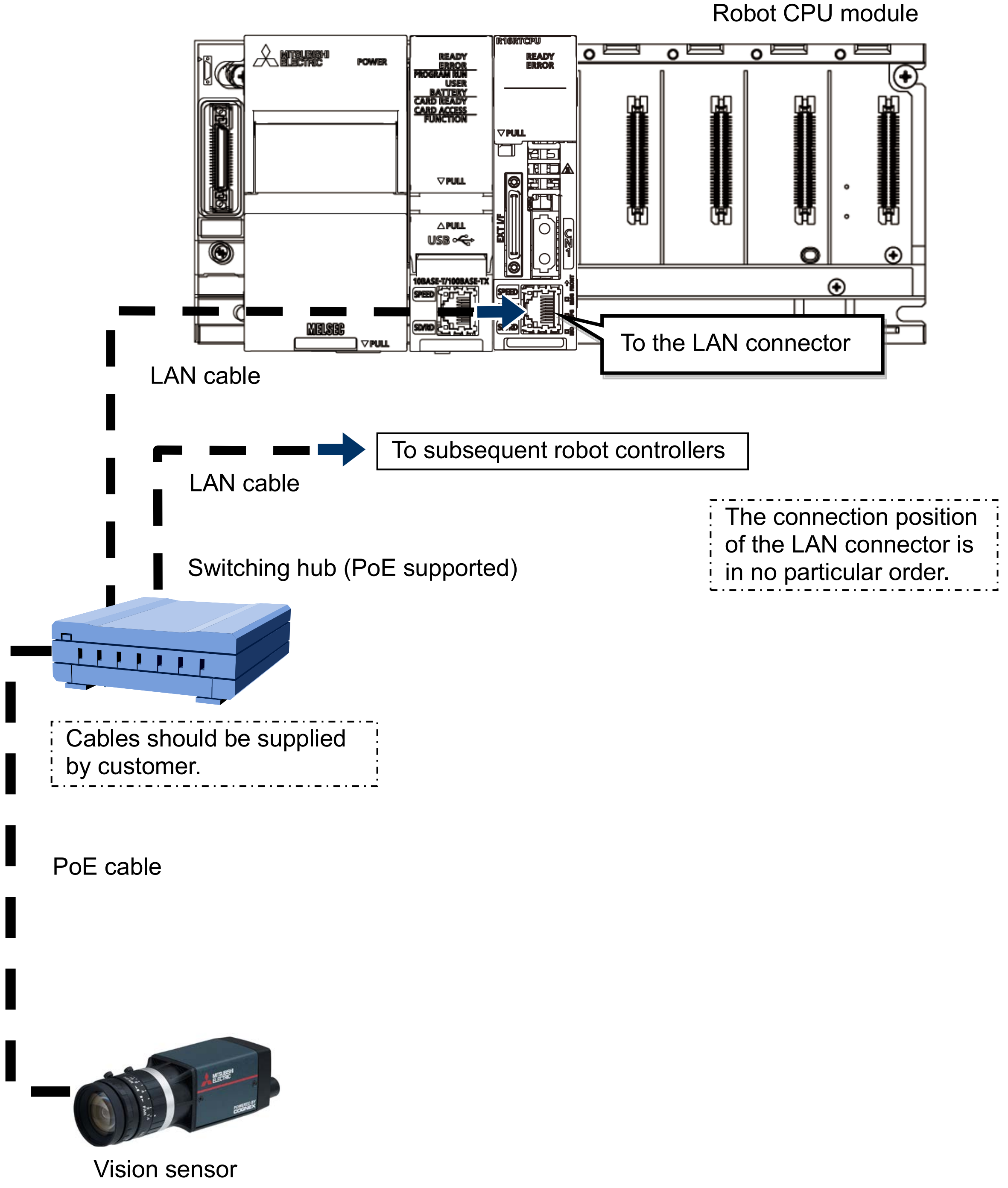


Fig. 5-3 Device wiring (CR800-R)

5.1.4. Vision sensor installation

Connect a vision sensor as shown below.

5.1.4.1. Hand cameras

Secure the vision sensor dedicated communication cable and power cable to the outer surface of the robot arm. Ensure that the cables are not stretched when the robot moves.

A jig to attach the vision sensor to the robot should be supplied by customer.

Fig. 5-4 Example of wiring when a hand camera is used

5.1.4.2. Fixed cameras

Secure the vision sensor dedicated communication cable and power cable to the stand. A vision sensor stand should be supplied by customer.

For fixed cameras, a jig to attach a calibration mark to the robot should be supplied by customer for calibration.

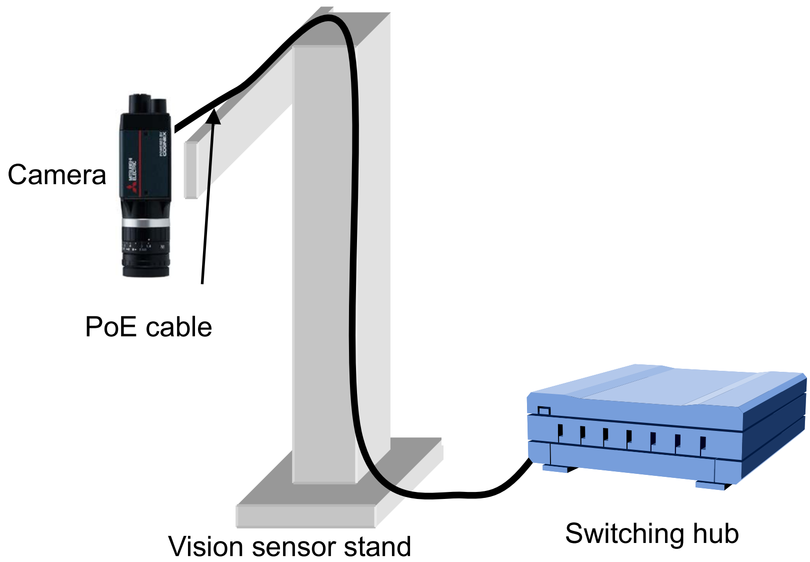


Fig. 5-5 Example of wiring when a fixed camera is used

5.2. Inserting the MELFA Smart Plus card

(1) Power off the robot controller.

- CR800-D type : 1) Turn off the switch of the earth-leakage circuit breaker.
- CR800-R/Q type : 1) Power off the robot CPU system.
  : 2) Turn off the switch of the earth-leakage circuit breaker.

(2) Insert the "MELFA Smart Plus card" or "MELFA Smart Plus card pack" into the robot controller. (For supported models, refer to "3.1Items in package".)

1) Lightly hold the lever to pull out the interface cover.

2) Hold the handle of the MELFA Smart Plus card, and insert it into SLOT1 or SLOT2.
Ensure that both ends of the card are inserted into the slits of the slot (SLOT1 or SLOT2 shown in Fig. 5-6).

3) Insert the connector as far as it will go until the lever is locked.

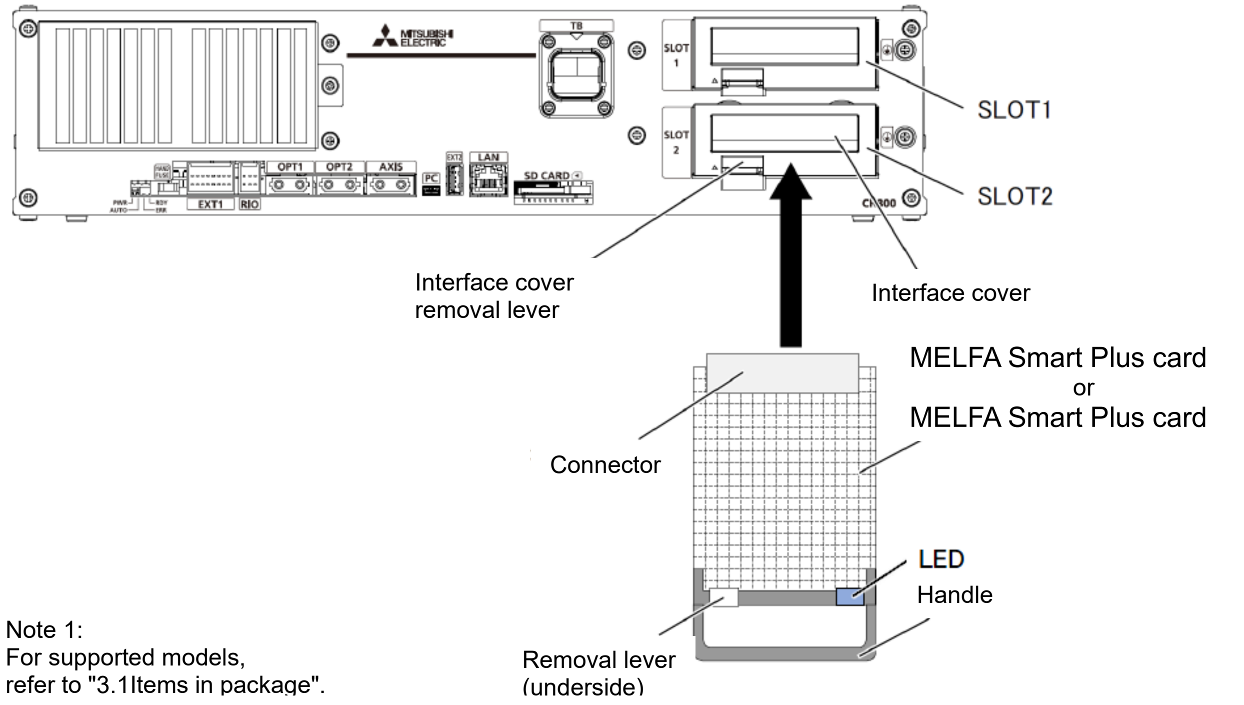


Fig. 5-6 Insertion of the MELFA Smart Plus card (MELFA Smart Plus card pack)

⚠ CAUTION

Insert one MELFA Smart Plus card only.
If multiple MELFA Smart Plus cards are inserted, the LED will not flash, but the error (L3782) will occur.

(3) Power on the robot controller.

- CR800-D type : 1) Turn on the switch of the earth-leakage circuit breaker.
  (The POWER lamp of the robot controller will flash. After startup, the lamp stays ON.)
- CR800-R/Q type : 1) Turn on the switch of the earth-leakage circuit breaker.
  (The POWER lamp of the robot controller will flash. After startup, the lamp stays ON.)
  2) Then, power on the robot CPU system.

(4) For the MELFA Smart Plus card, set the parameter to enable the vision sensor enhancement function.

[When using the MELFA Smart Plus card]

Any one of the functions supported by the card is available. Set the parameter to enable the MELFA Smart Plus and the vision sensor enhancement function, and restart the robot controller.

The parameter can be set with a teaching pendant or RT ToolBox3.
When using RT ToolBox3, connect it to the robot controller in which the vision sensor enhancement function is used.

1) Change the setting value of parameter "SMART+1" to "5".

2) Restart the robot controller.

- CR800-D type: Turn off and on the switch of the earth-leakage circuit breaker.
- CR800-R/Q type: Power off the robot CPU system.
  → Turn off the switch of the earth-leakage circuit breaker.
  → Turn on the switch of the earth-leakage circuit breaker.
  → After the POWER lamp on the robot controller flashes, power on the robot CPU system.

3) When the vision sensor enhancement function is enabled, the LED on the card will flash green (near the handle, refer to "Fig. 5-6 Insertion of the MELFA Smart Plus card (MELFA Smart Plus card pack)").

[When using the MELFA Smart Plus card pack]

・ It is not required to configure these settings and restart the robot controller. Continue to Step 5.
・ Check that the LED on the MELFA Smart Plus card pack flashes blue.

(5) Set the function code of the MELFA Smart Plus card.

[When setting the function code in RT ToolBox3 to which the robot controller is connected]

1) Start RT ToolBox3, then connect it to the robot controller.
2) Select [Option] from the Workspace tab.
3) Select [MELFA Smart Plus] from the tree structure on the left of the Option window.
4) Press the [Get function code] button. The function code of the MELFA Smart Plus card will be entered in the Function code field. Press the [Set] button.
5) Restart RT ToolBox3.

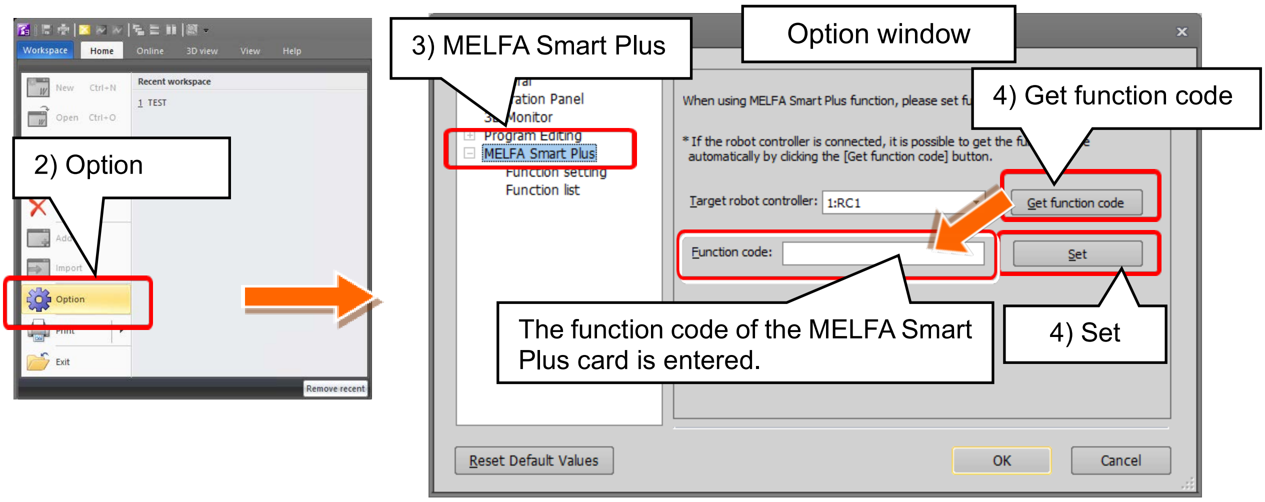


Fig. 5-7 Setting the function code of the MELFA Smart Plus card (when the robot controller is connected)

[When setting the function code in RT ToolBox3 to which the robot controller is not connected]

1) Read the value of parameter "MSPCODE" with a teaching pendant or in RT ToolBox3 to which the robot controller is connected. (The value of "MSPCODE" is the function code. This code does not appear unless the MELFA Smart Plus card is inserted.)
2) Write down the displayed value of parameter "MSPCODE" (24 alphanumeric characters).
3) Select [Option] from the Workspace tab in RT ToolBox3 to which the robot controller is not connected.
Select [MELFA Smart Plus] from the tree structure on the left of the Option window.
4) Enter the MELFA Smart Plus card function code (acquired in Step 1) in the Function code field, then press the [Set] button.
5) Restart RT ToolBox3.

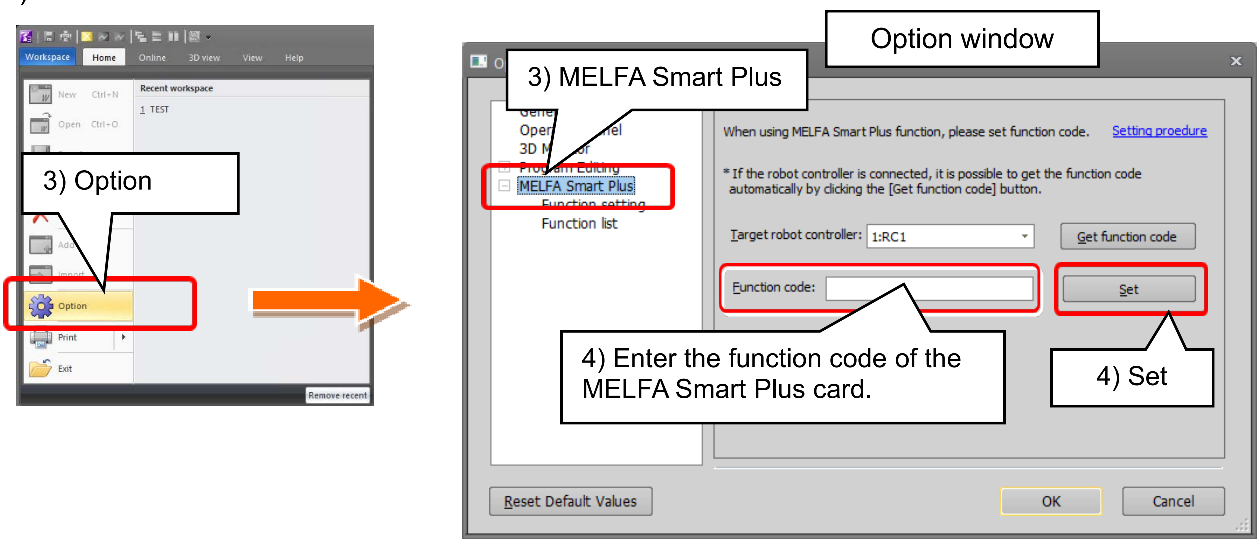


Fig. 5-8 Setting the function code of the MELFA Smart Plus card (when the robot controller is not connected)

(6) Check that the vision sensor enhancement function is enabled.

1) Select [MELFA Smart Plus] - [Function list] from the tree structure on the left of the Option window.

2) Whether MELFA Smart Plus functions are enabled will be listed on the right.
Check that "Vision sensor enhancement function" is "Enable".

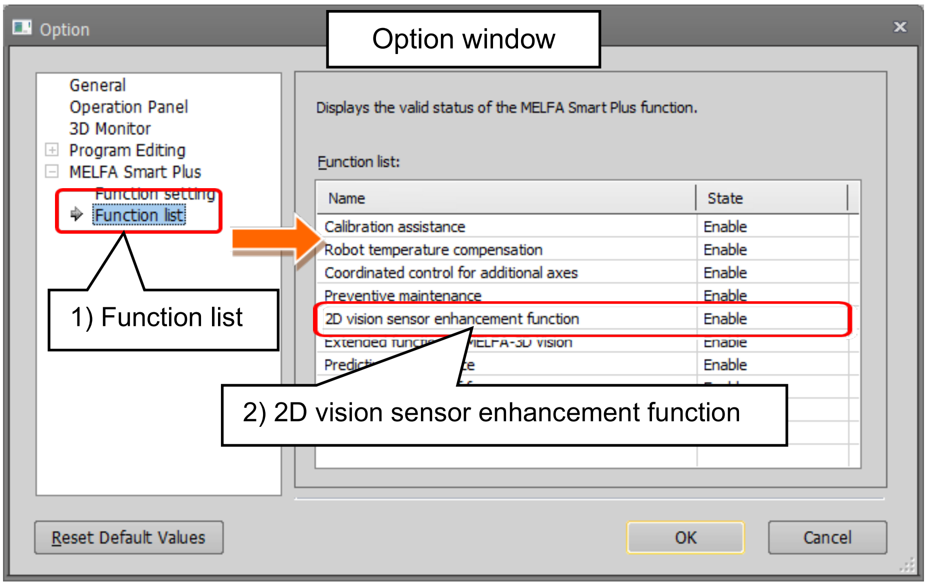


Fig. 5-9 Checking the status of the vision sensor enhancement function

5.3. Starting RT ToolBox3

To set the 2D vision sensor enhancement function, start RT ToolBox3.
To start RT ToolBox3, double-click the shortcut on the desktop or select as follows: [Start] button → [All Programs] → [MELSOFT] → [RT ToolBox3].

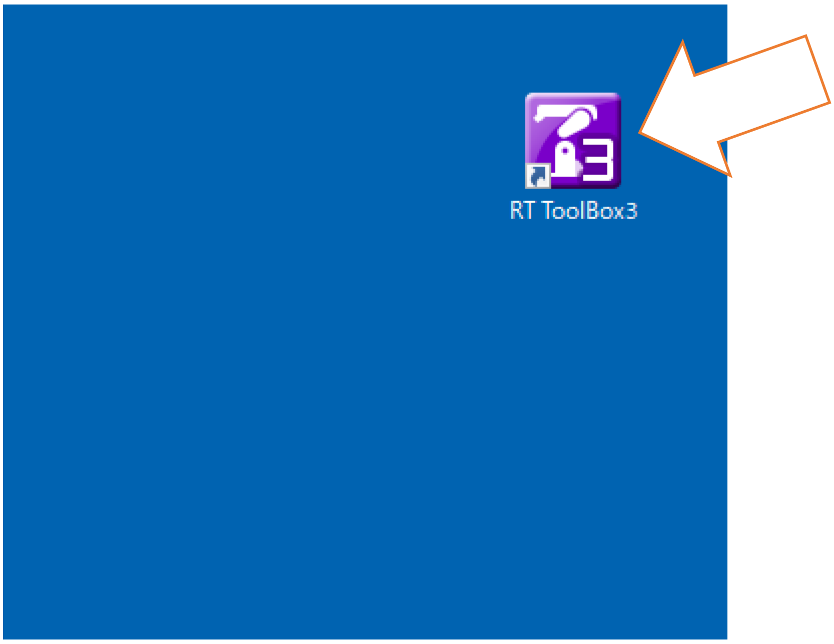


Fig. 5-10 RT ToolBox3 shortcut

5.4. Hand settings

Parameters must be set depending on which hand type (single solenoid type/double solenoid type) or solenoid valve type (sink type/source type) is selected. Refer to the following instruction manuals to set the related parameters (HIOTYPE, HANDTYPE, HANDINIT).

- "Tooling" in the "Instruction Manual/Standard Specifications Manual"
- "Hand type" and "default hand status" in the "Instruction Manual/Detailed explanations of functions and operations"

Once the related parameters have been set, check that the hand can open and close using the teaching pendant.
For information on opening and closing the hand, refer to the "Instruction Manual/Detailed explanations of functions and operations".

5.5. Starting the vision sensor enhancement function

This section explains how to display screens for the 2D vision sensor enhancement function.

After RT ToolBox3 starts up, the mode switches to [Online], and a connection with the robot controller is established, double-click "Vision sensor enhancement" found in the [2D Vision] group of the workspace.

<u>* If the MELFA Smart Plus card is not inserted, or the vision sensor enhancement function is disabled, the "Vision sensor enhancement" icon will not appear in the workspace.</u>

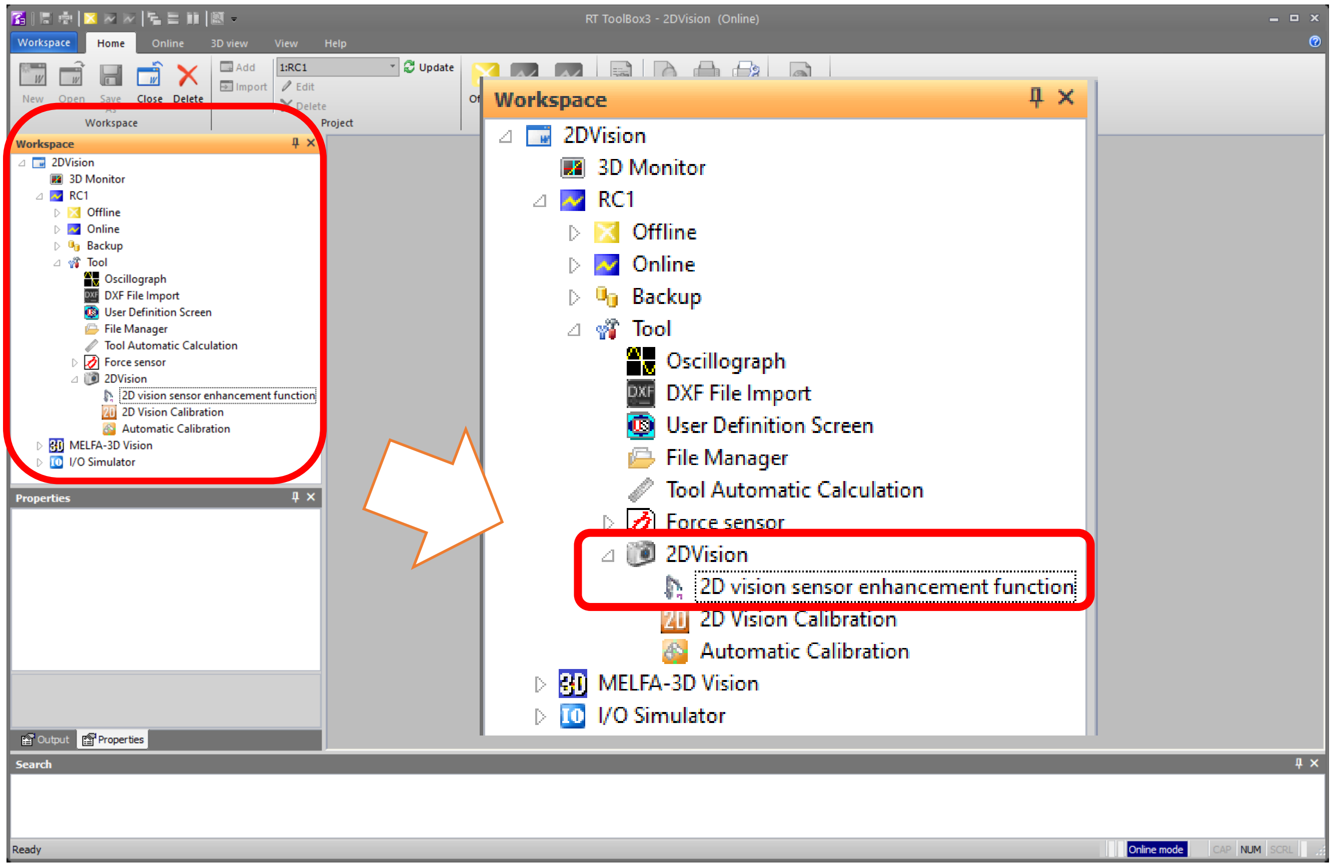


Fig. 5-11 RT ToolBox3 workspace

5.6. Start screen

Starting the 2D vision sensor enhancement function will display the start screen as shown below. Select [New] or [Import].

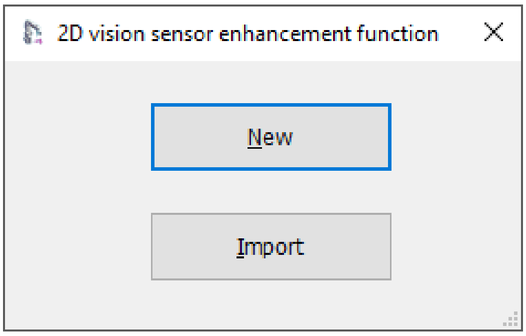


Fig. 5-12 Start screen

<New>

To use for the first time or create a different application, click the [New] button. Clicking the [New] button will display a message asking to delete relevant programs. To back up the programs, click the [Cancel] button for program backup. For information on how to back up the relevant programs, refer to "15.2 Backing up and restoring robot programs and parameters". To delete the relevant programs, click the [OK] button, then go to "6 SELECTING AN APPLICATION".

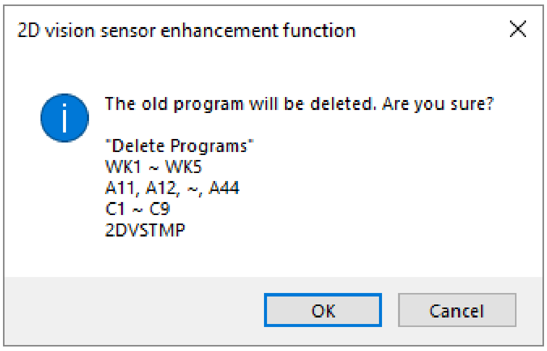


Fig. 5-13 Message asking to delete relevant programs

<Import>

To use exported data, or if created robot program data has been changed, click the [Import] button. Clicking the [Import] button will display the window to select an import file.

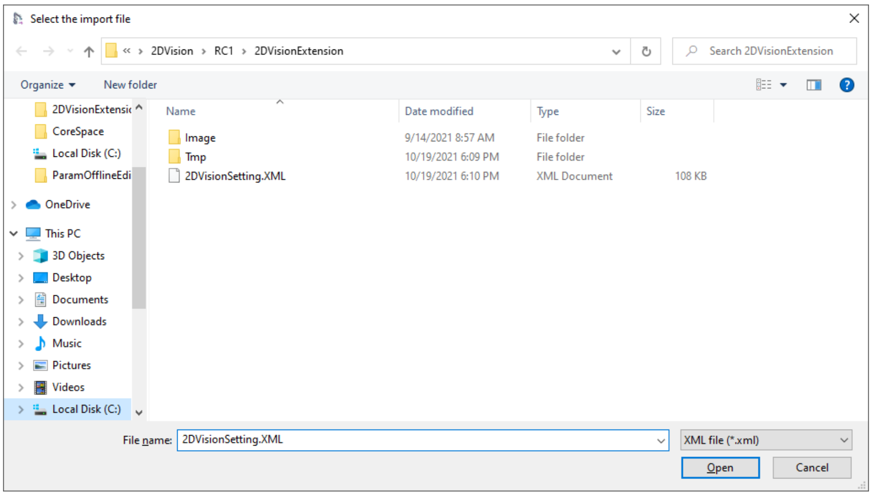


Fig. 5-14 Window to select an import file

After selecting a file to be imported, click the [Open] button. A message whether to use the robot program data will appear. Click [Yes] to use the data, or [No] to decline. (To apply the program contents changed without using this tool, select [Yes].)

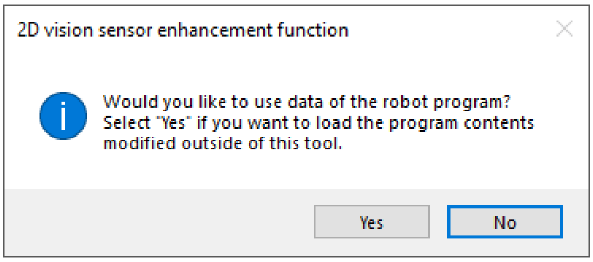


Fig. 5-15 Message asking to use the robot program data

Selecting [Yes] will apply the following data to the screen: Data stored in programs WK1 through WK5 containing workpiece type data. Some data of the calibration program and main program will also be applied to the screen.
If an imported file has a problem, the following error message will appear. Select an import file again.

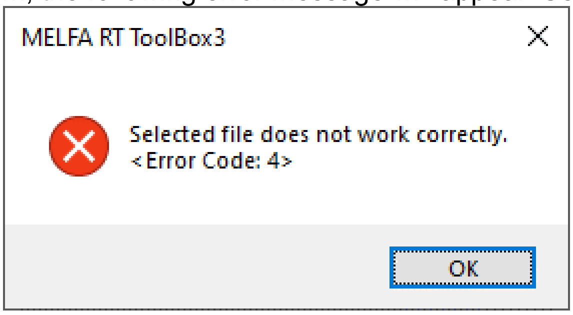


Fig. 5-16 Error message

[POINT]
If RT ToolBox3 is force quit during configuration, the data recorded until then can be restored for a resumption of the process. Performing the vision sensor enhancement function again will display the following message. Click the [Yes] or [No] button. Clicking the [Yes] button will resume the process. Clicking the [No] button will return to "5.6 Start screen".

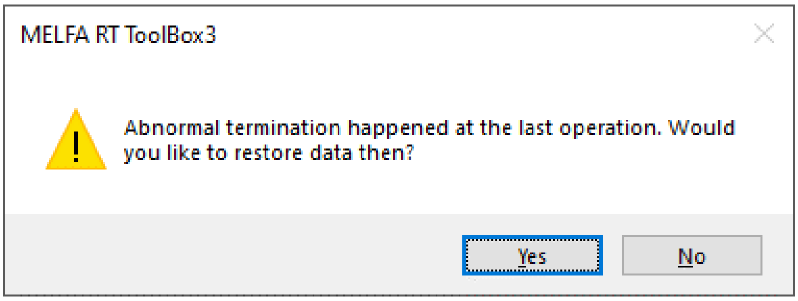


Fig. 5-17 Confirmation message whether to resume the process

6. SELECTING AN APPLICATION

First, select an application you want to set up (Fig. 6-1 Screen to select an application). Selectable applications can be switched with the [<] or [>] button. Clicking any of the application illustrations allows you to check the motion. Select the desired application while checking the motion.
For details on applications, refer to "6.1 Application list".
If clicking the illustration does not start an animation, refer to "18. TIPS".

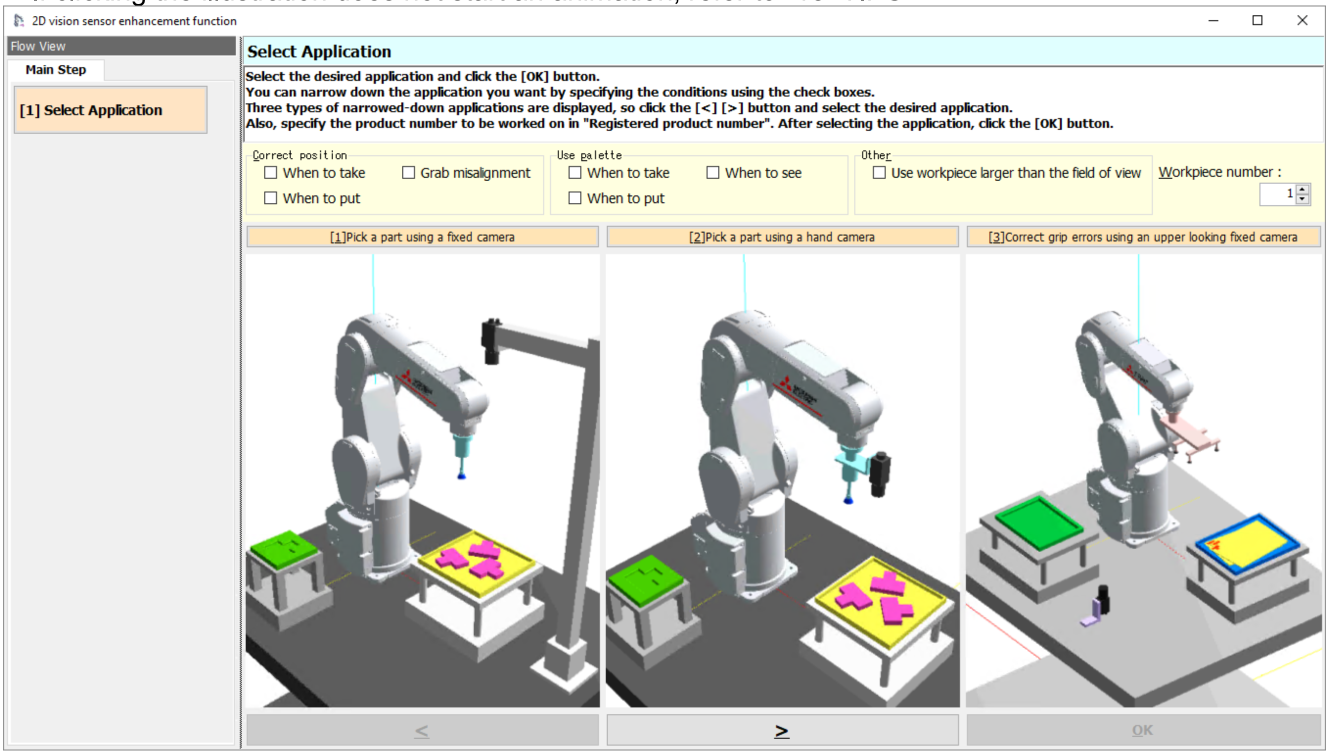


Fig. 6-1 Screen to select an application

By selecting check boxes, applications can be narrowed down according to the usage.
For details, refer to "6.2 Narrowing down applications".

After selecting an application you want to set up, select the number of the workpiece to be registered. Set any number from 1 to 5 for "Workpiece number", then click the [OK] button. For information on the specification of workpiece type numbers, refer to "6.3 Workpiece type number settings".

Selecting an application and clicking the [OK] button will display steps according to the application. Steps consist principally of "Config. Communication", "Calibration", "Register a Part", "Config. Operation", "The Position to Pick", and "Confirm Operation". However, if an application that incorporates multiple vision sensors is selected, "Calibration" and "Register a Part" must be performed for each vision sensor. In such a case, the number of steps is increased.

⚠ CAUTION

To change the selected application after the [OK] button has been pressed, go back to "5.6 Start screen".

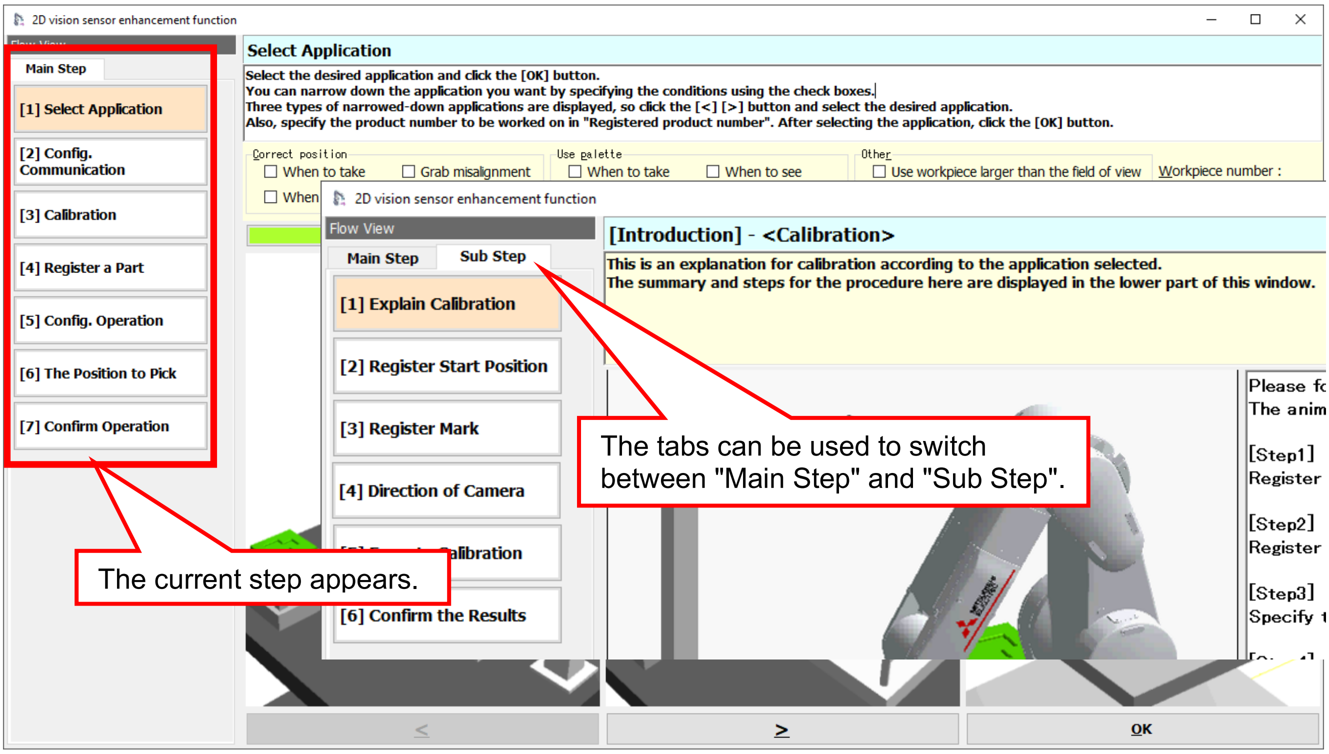


Fig. 6-2 Flow View (Main Step/Sub Step)

The step being performed can be checked from the Flow View panel. The main steps "Calibration", "Register a Part", and "Config. Operation" have sub steps. Configure the detailed settings in each step. The following table shows the main steps and their sub steps.

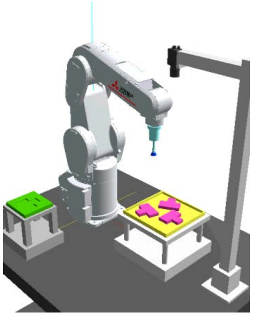
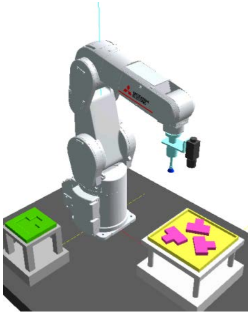
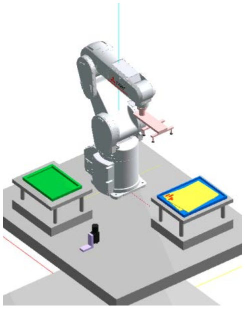
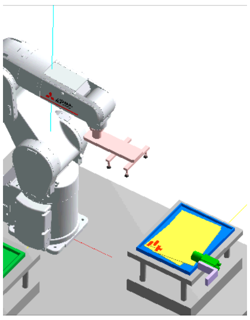
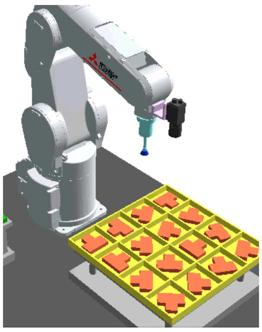
Table 6-1 List of basic sub steps

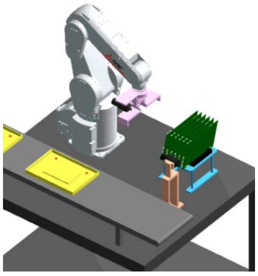
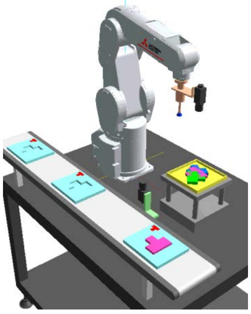
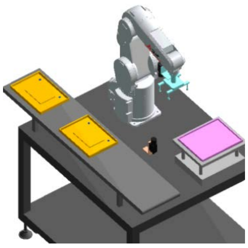
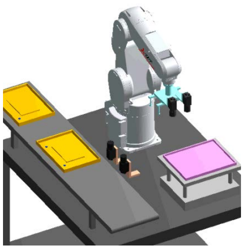
Main Step	Calibration	Register a Part	Config. Operation
Sub Step	[1] Explain Calibration [2] Register Start Position [3] Register Mark [4] Direction of Camera [5] Execute Calibration [6] Confirm the Results	[1] Exp. to Register a Part [2] Register Trigger Pos. [3] Register a Part	[1] Teach Start Position [2] Position to Pick [3] Waiting Pos. to Pick [4] Waiting Pos. after Pick [5] Position to Place [6] Waiting Pos. to Place [7] Waiting Pos. after Place [8] Move End Position

6.1. Application list

The following table lists applications.

Table 6-2 Application list

Category	Application		Robot type	Number of cameras	Camera attachment type	Target for compensation
Basic	Pick a part using a fixed camera		Vertical or horizontal	1	Fixed downward-facing	Pick-up point
	Pick a part using a hand camera		Vertical or horizontal	1	Hand	Pick-up point
	Correct grip errors using an upper looking fixed camera		Vertical or horizontal	1	Fixed upward-facing	Grip error
	Correct grip errors using a side looking fixed camera		Vertical	1	Fixed sideways	Grip error
Applied	Pick a part from palette using a hand camera		Vertical or horizontal	1	Hand	Pick-up point

Category	Application		Robot type	Number of cameras	Camera attachment type	Target for compensation
Applied	Using a hand camera, correcting grip errors by fixed camera		Vertical	2	Fixed sideways Hand	Pick-up point Grip error
	Correct grip errors (UF), place a part (HC)		Vertical or horizontal	2	Fixed upward-facing Hand	Grip error Placement point
	Correct grip errorsx2 (UF), place a partx2 (HC)		Vertical or horizontal	2	Fixed upward-facing Hand	Grip error Placement point
	Correct grip errors (UFx2), place a part (HCx2)		Vertical or horizontal	4	Fixed upward-facing x 2 Hand x 2	Grip error Placement point

6.2. Narrowing down applications

Applications can be narrowed down and displayed by selecting the timing of position compensation, whether to use a pallet, and the size of the workpiece being used.

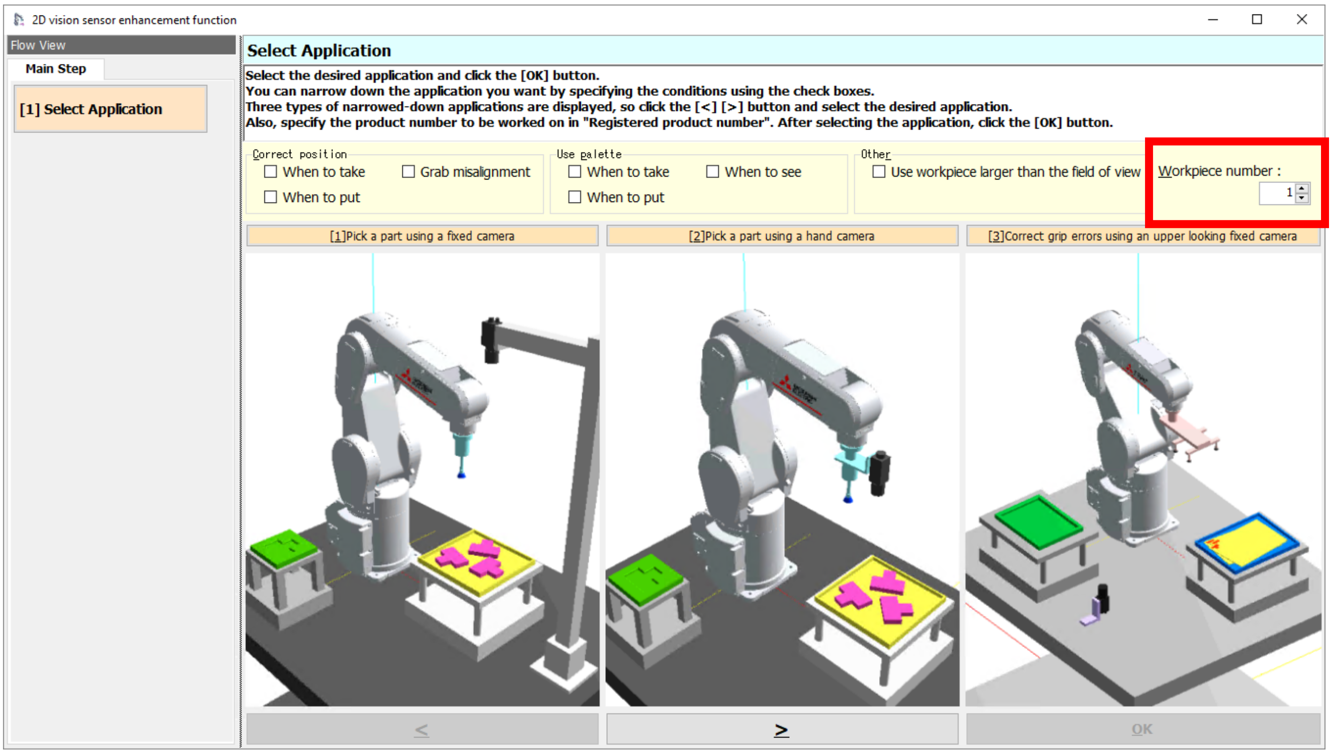
Table 6-3 Elements to narrow down applications

Element		Application characteristics
Correct position	When to take	Corrects the position to grasp the workpiece before grasping (sucking) it. (Available for downward-facing cameras and hand cameras)
	When to put	Corrects the position to place the workpiece. (Available for downward-facing cameras and hand cameras)
	Grab misalignment	Corrects grip errors when the robot grasps (sucks) the workpiece. (Available for upward-facing cameras and side-facing cameras)
Use palette	When to take	Picks up workpieces arranged in a grid pattern (or single column).
	When to put	Places workpieces in a grid pattern (or single column).
	When to see	Images workpieces arranged in a grid pattern (or single column). (Only available for hand cameras)
Use workpiece larger than the field of view		Corrects the position of the workpiece by taking two images focused on the workpiece characteristics (Only available for hand cameras).

6.3. Workpiece type number settings

Up to five types of workpieces can be registered per application (same program). When adding a different type of workpiece to a created application, read existing data by selecting [Import] on the start screen, and change the setting of "Workpiece number". Failure to do so will overwrite all created workpiece information.

Also, when an additional workpiece is registered, calibration is not required. "8 CALIBRATION" can be skipped.



<When switching the workpiece type number for actual operation>

The workpiece type number is stored in the "PWkNum" variable (X component) of the main program saved in "12.3 Saving a program". When switching the workpiece type for actual operation, set the X component of the "PWkNum" variable to the number corresponding to "Workpiece number".

For information on parameters, refer to "14.2.2 Variables related to operation settings".

7. CONFIGURING THE VISION SENSOR COMMUNICATION SETTINGS

Configure the communication settings of the robot controller and vision sensor. Select the manufacturer of the vision sensor to be used (MITSUBISHI ELECTRIC or Cognex) from the Select Vision Maker drop-down box.

Selecting MITSUBISHI ELECTRIC will display the "UsePatMax" check box next to the combo box.
PatMax is the name of the pattern search algorithm based on the outline information of the target object (a registered trademark of Cognex). The processing time sometimes takes longer than the normalized correlation, but it is generally highly accurate and operates stably against illumination light, focus variations, loss of information, and noise.
For details on the operation using PatMax, refer to "7.1. Using PatMax".

You can check the connection status (green light: connected, red light: not connected) with the status indicators displayed in the bottom right of vision sensors.

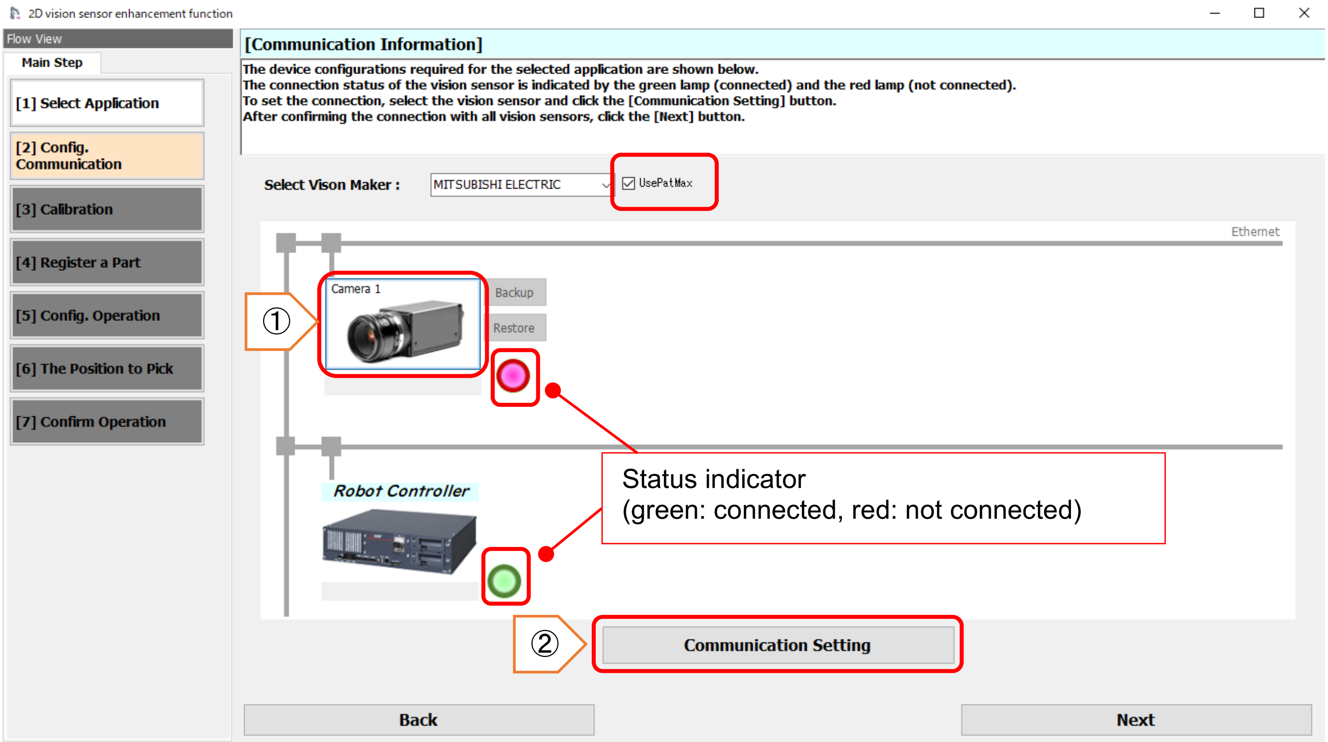


Fig. 7-1 Connection status of the vision sensors

① Select a vision sensor to be connected.

② Clicking the [Communication Setting] button will display the Config. Communication screen for the vision sensor.

If two or more network adapters exist on the computer, the following dialog (Fig. 7-2 Selecting a network adapter) will appear. Select the network adapter connected to the vision sensor.
Once selected, this window will not appear anymore. To select a different network adapter, close and restart the vision sensor enhancement function screen.

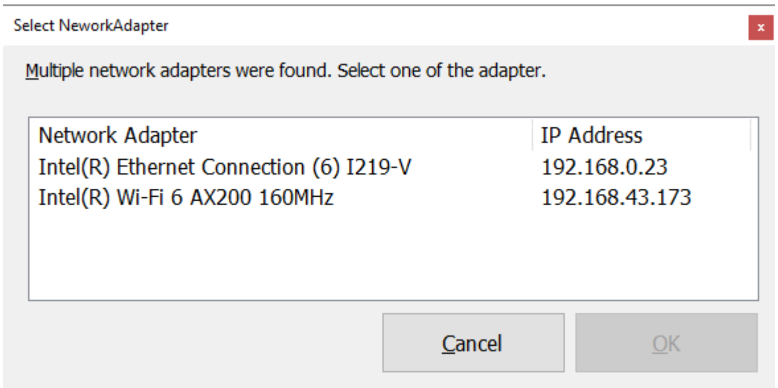


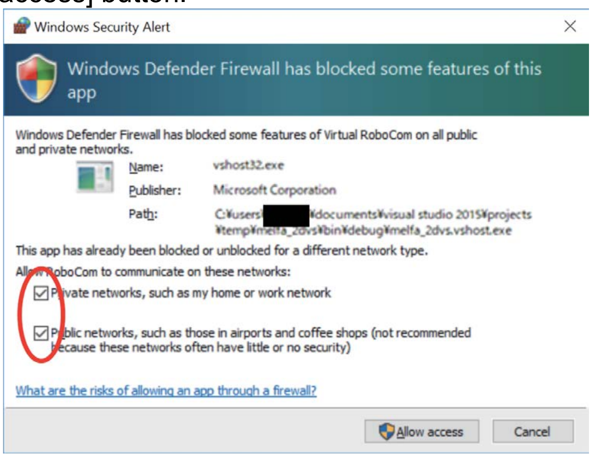
Fig. 7-2 Selecting a network adapter

After a network adapter is selected, the Config. Communication screen will appear.
The screen will display a list of vision sensors detected on the network to which the robot controller is connected.

⚠ CAUTION

If the following window appears in a connection attempt

If the following window appears while attempting to connect the vision sensor via Ethernet, check both boxes, then click the [Allow access] button.



③ Select the vision sensor to be connected, then click the [OK] button.

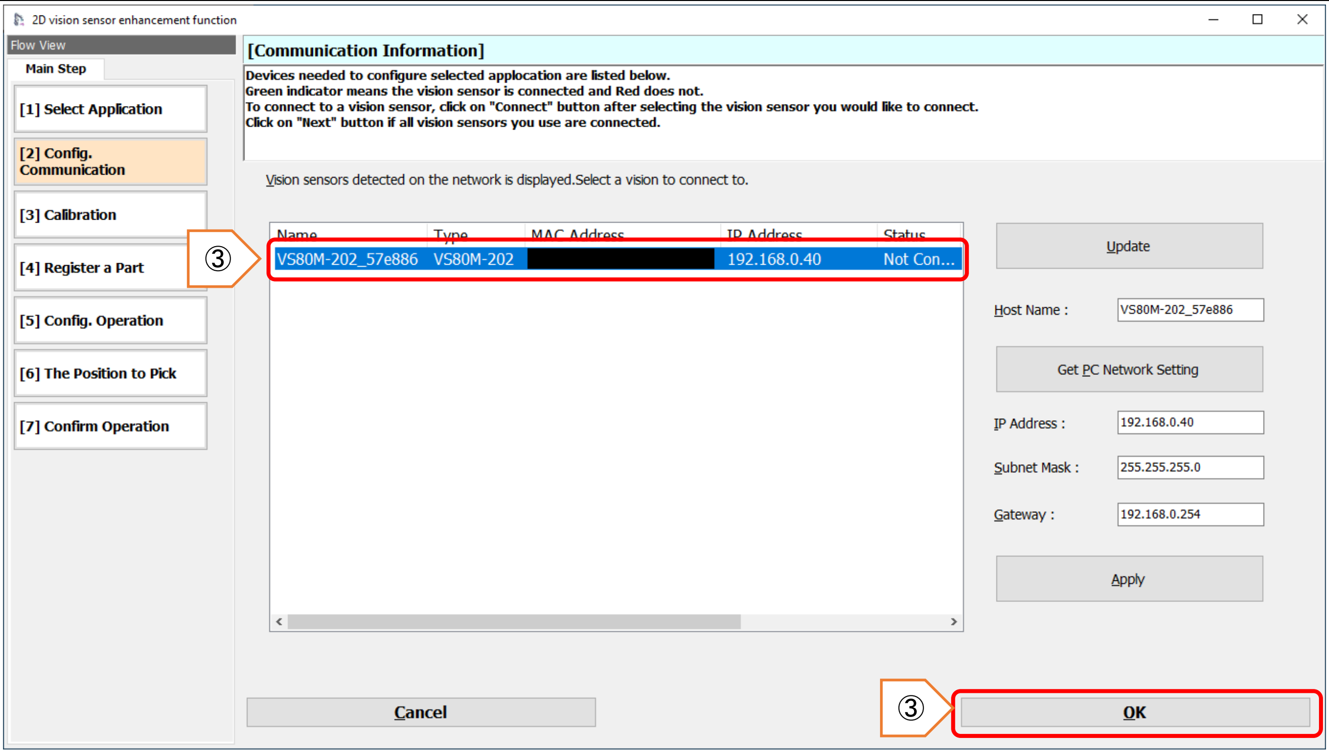
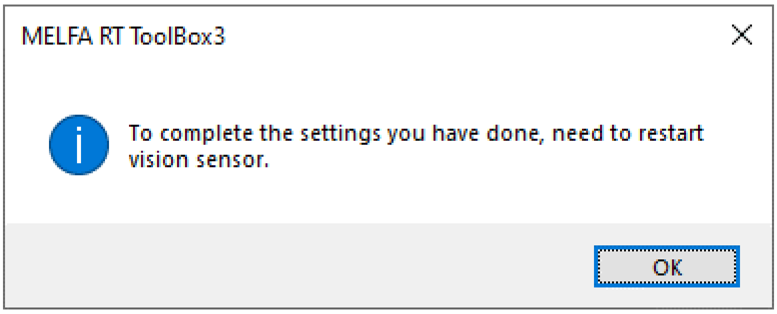


Fig. 7-3 Communication setting screen

[POINT]
If no vision sensor is detected, an error message will appear. Check the following settings. ・ The vision sensor is connected correctly. ・ The network adapter is selected correctly.

To refresh the Config. Communication screen, click the [Update] button.

<When changing the vision sensor communication settings>
Change the settings of the IP address, subnet mask, gateway according to the vision sensor to be connected, then click the [Apply] button. (Restarting takes about 40 seconds.)
The [Get PC Network Setting] button can be used to copy the computer settings.



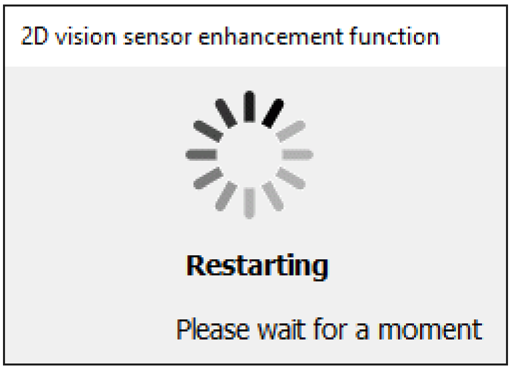


Fig. 7-4 Restarting screen

After a setting completion message appears, click the [OK] button.

[POINT]

The network settings will be applied after restart.
* For the CR800-R and CR800-Q, power on the robot controller manually.
If an error occurs after restart, change the network settings or solve network problems, then click the [Apply] button on the Config. Communication screen again.

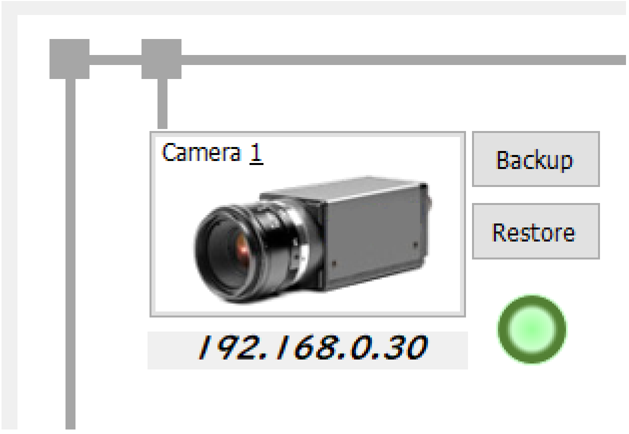


Fig. 7-5 Backup and restore

The vision sensor settings can be backed up upon connection with the vision sensor. To back up the settings, click the [Backup] button.
To restore data to the vision sensor, click the [Restore] button.
For details on backup and restore of vision sensor settings, refer to "15.3 Backing up and restoring vision programs".

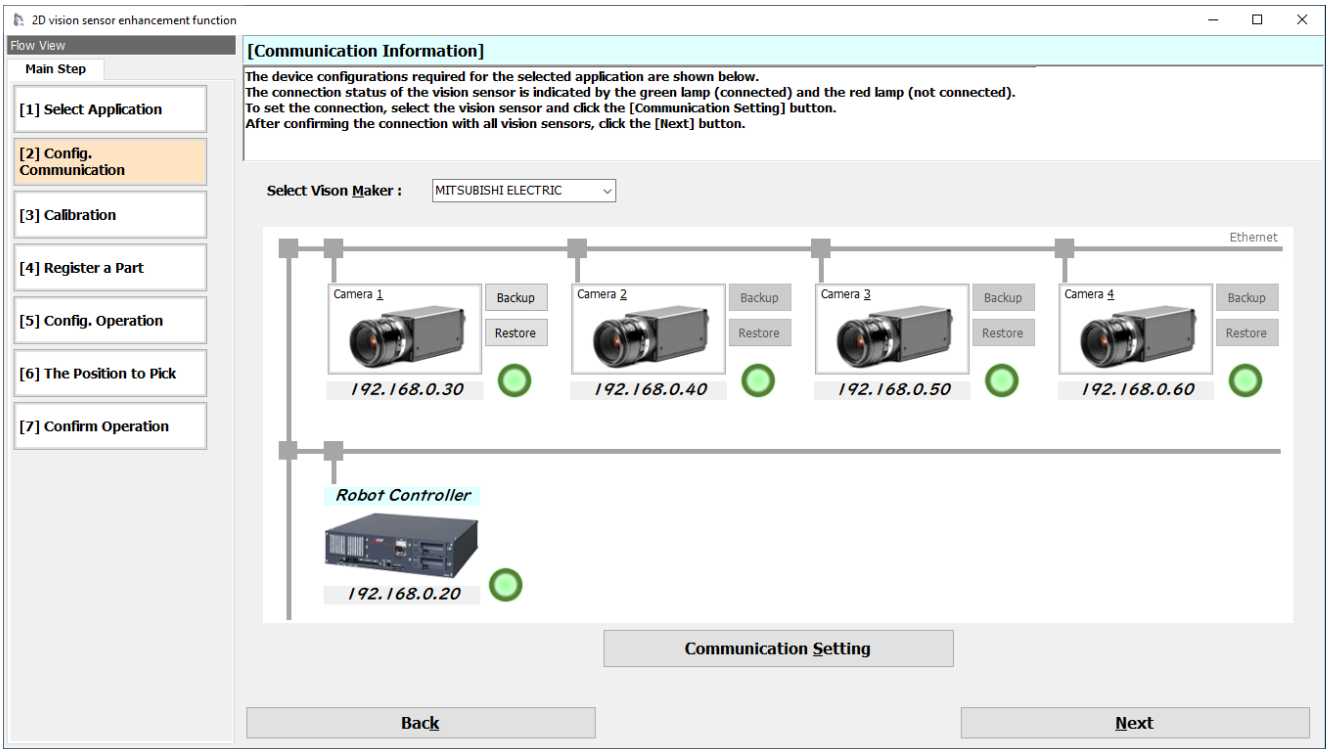


Fig. 7-6 Connection status of devices

After communication settings are configured in all the vision sensors to be connected, click the [Next] button.

7.1. Using PatMax

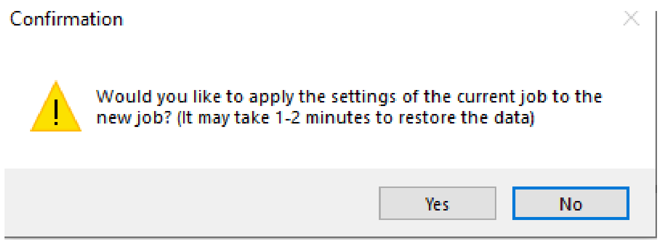
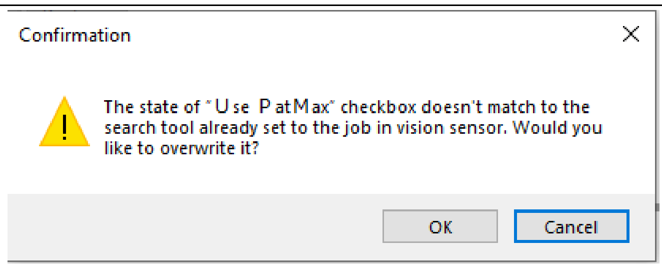
PatMax requires RT ToolBox3 Ver. 2.20W or later.
It is recommended to use the "Use PatMax" check box as follows:

・To focus on the reliability of the recognition results, select the "Use PatMax" check box.
・To focus on the processing time, deselect the check box.

In applications in which multiple vision sensors are used, the selection status of the check box is applied to all vision sensors.

Vision sensors are set according to the selection status of the "Use PatMax" check box.
If a job already exists in a vision sensor, for example, if the vision sensor is reused, the dialog box shown in Table 7-1 may appear.

Table 7-1 Dialog box that appears in the job preparation

Dialog box category	Dialog box appearance	Display condition
Overwrite confirmation		When the vision sensor job and the "Use PatMax" check box are not consistent
Setting reflection confirmation		When the above dialog appears and the [OK] button is clicked.

Clicking the [OK] button in the overwrite confirmation dialog box will overwrite the job that currently exists on the vision sensor with the default job.
A default job is a job whose basic configuration has completed, but recognition parameters have not been set in that job.
Clicking the [OK] button in the setting reflection confirmation dialog box will reflect the recognition parameter settings in the job that currently exists on the vision sensor to a default job.

Table 7-2 shows the ○ mark for the data that will be reflected when the [OK] button in the setting reflection confirmation dialog box is clicked.
The data that is reflected depends on the selection status of the "Use PatMax" check box.

Table 7-2 Recognition parameter settings that are reflected to a new job

Data		Selection status of the "Use PatMax" check box	
Item	Meaning	Selected	Deselected
Model	Coordinates and size information for the region surrounding the search model.	○	×
Threshold	A score from 0 to 100 to be specified to judge whether search results are considered as success. The higher the value, the stronger the similarity.	○	○
Angle tolerance	The limit angle that the search target is allowed to rotate in the ± directions.	○	○
Offset X (at workpiece registration)	A value added to intentionally move the X coordinate of the search results.	○	○
Offset Y (at workpiece registration)	A value added to intentionally move the Y coordinate of the search results.	○	○

The "Angle tolerance" applies only to the recognition conditions of the vision sensor.
It is different from the parameter "Allowable Rotation Angle" which limits the robot's rotation range at the time of calibration.

Note the following points when using a Cognex vision sensor.
In-Sight 7000/8000 series Cognex vision sensors with firmware Ver. 6.2.1 have been confirmed to be configurable as PatMax-compatible with this software. To use PatMax with a Cognex vision sensor, select "MITSUBISHI ELECTRIC" from the Select Vision Maker drop-down box.
If the communication specifications are changed in firmware Ver. 6.2.1 or later, Cognex vision sensors may not be available for PatMax.

8. CALIBRATION

Performing auto calibration will link the robot coordinate system (unit: mm) and vision coordinate system (unit: pixel). Steps differ depending on the application, but basically follow the steps shown below.

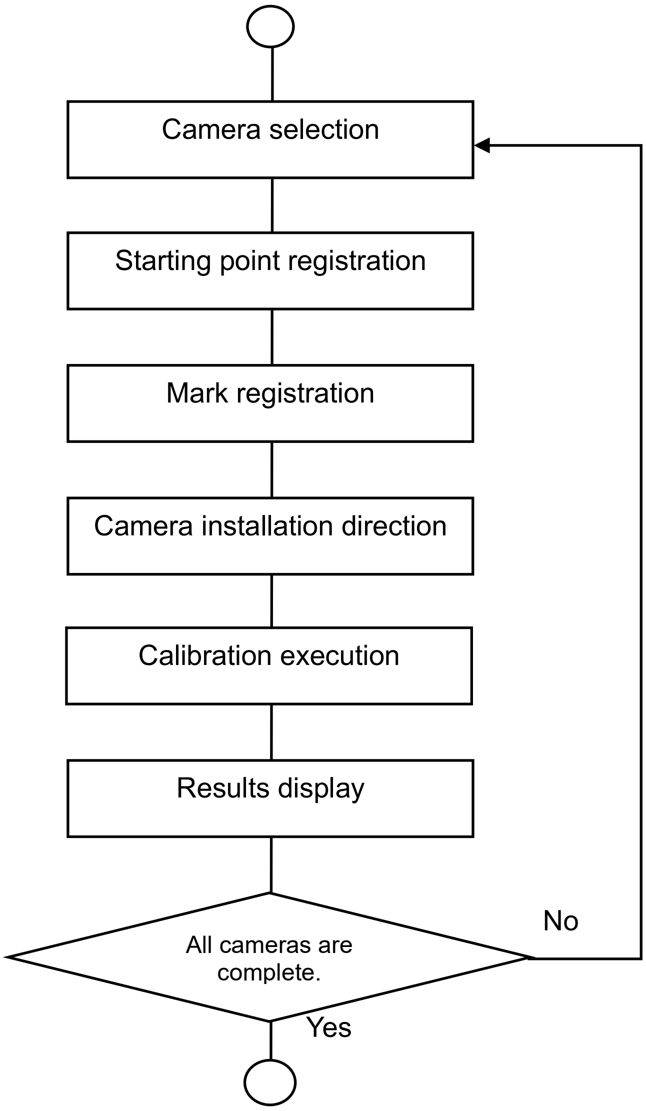


Fig. 8-1 Vision setting procedure

8.1. Explanation on calibration settings

You can check a series of operations for calibration on video.
(The calibration video does not include sound.)

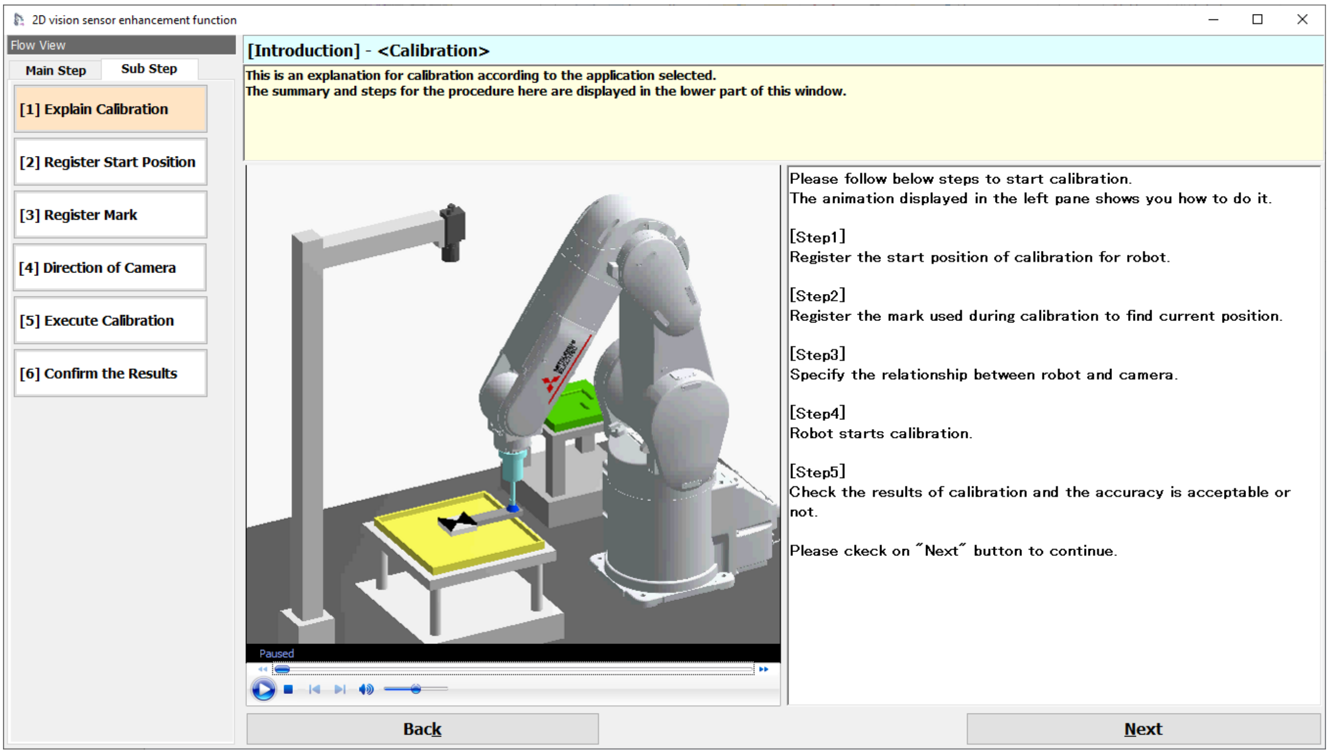


Fig. 8-2 Explanation on calibration

The panel on the left can be used to play back, pause, select where to play back, and for other purposes.

The panel on the right shows the steps for configuring the vision settings.

After checking the steps, click the [Next] button.

8.2. Registering the calibration start position and setting the vision

Teach the calibration start position and adjust the lens and focus in that position.
Follow the instructions shown in the Working Image panel and Procedure panel.

Calibration requires a calibration mark. The actual workpiece can be used as a calibration mark. Prepare a mark that can be recognized easily. (Use the calibration mark shown in "19.4 Calibration mark".)
<For fixed cameras>

Fig. 8-3 Registering the calibration start position and setting the vision

⚠ CAUTION

If using a calibration mark other than the one shown in "19.4 Calibration mark", do not use the marks shown below.

・ A mark with the same color as the background
・ A mark that does not have an orientation

Setting		Initial value	Function explanation
[3]	Specify the exposure	8.000	Set the exposure time of the vision sensor.

① Make the robot grasp (suck) the mark as shown below. Move the calibration mark so that the moving distance is the same as the working distance (between the camera and the workpiece to be recognized) in actual operation.
Open or close the hand using the Operation Panel or teaching pendant. For the Operation Panel, refer to "19.2 How to operate the Operation Panel".

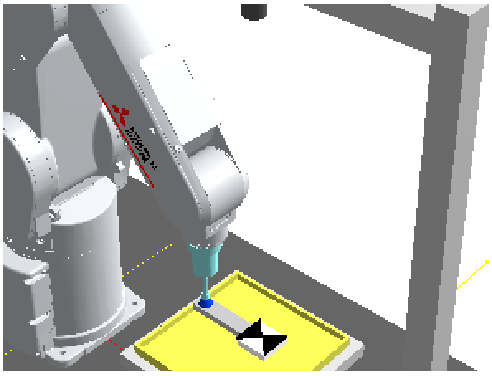
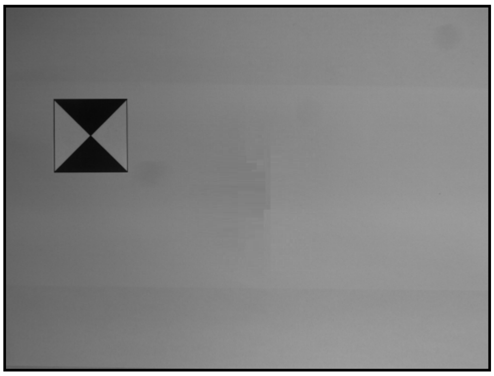


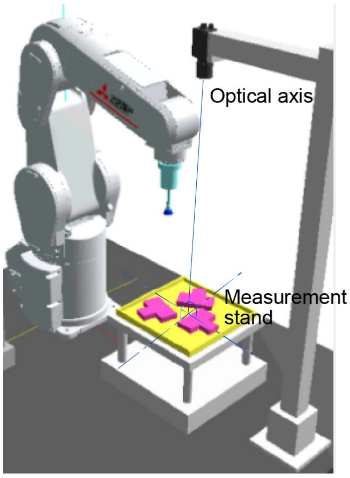
Fig. 8-4 Calibration start position (for downward-facing cameras)

② Select the camera number to be set from "Select a camera" of the Setting panel.
③ Set the resolution of the camera in "Specify the resolution". The resolution of the vision sensor will be automatically set. Check that the resolution of the vision sensor has been set.
④ Set the exposure time of the camera in "Specify the exposure".
⑤ A vision sensor image will appear in the Vision View.
After setting "Specify the exposure", click the [Stop] button, then the [Live] button.
(Live images can be switched ON and OFF with the [Live]/[Stop] button.)



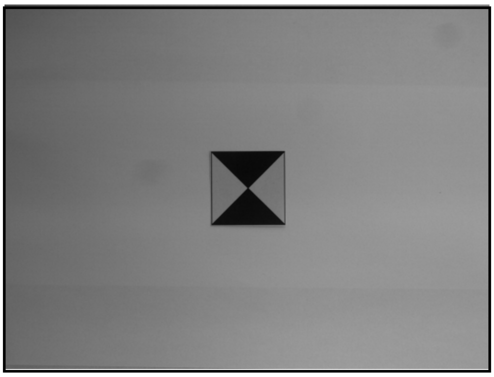
[POINT]
If no live image appears, follow the instructions in "3.4 Computer settings".

⑥ Move the robot to a position where the calibration mark can be seen. In this situation, the following conditions must be met at the same time.
<Conditions for the calibration start position (for fixed cameras)>
・ The calibration mark is positioned so that the moving distance is the same as the working distance (between the camera and the plane in which the workpiece is placed) in actual operation.
・ The calibration mark is displayed in the center of the image.
・ The optical axis and calibration mark are vertical to each other.

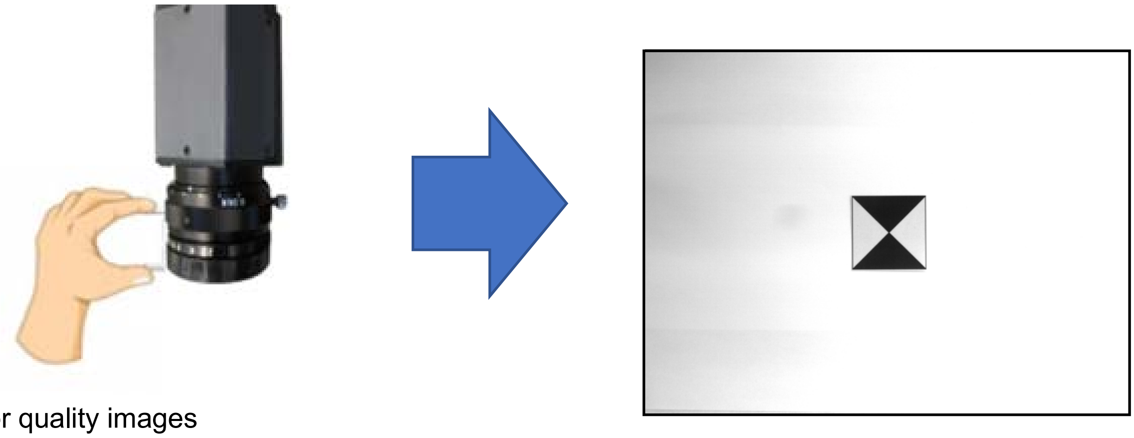


* For fixed upward-facing (side-facing) cameras (grip error correction), the optical axis of the camera is automatically aligned during calibration, and thereby visual alignment is possible. For fixed downward-facing cameras, however, the optical axis cannot be aligned. Make sure that the optical axis of the vision sensor and measurement stand are vertical to each other using a level or similar device for parallelism.

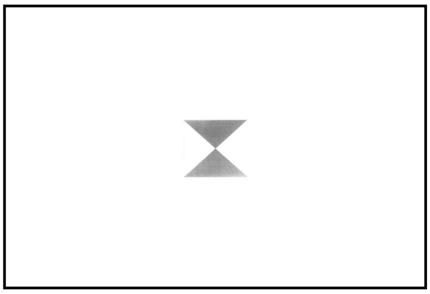
Move the mark to the center of the Vision View as shown in the figure below.
Move the robot using the Operation Panel, the position jump function, the teaching pendant, etc.
For the position jump function, refer to "19.3 Position jump".



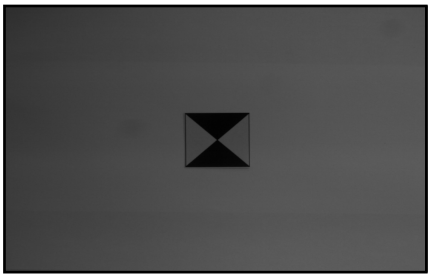
⑦ If the Vision View does not display images properly, adjust the focus and aperture of the vision sensor lens.



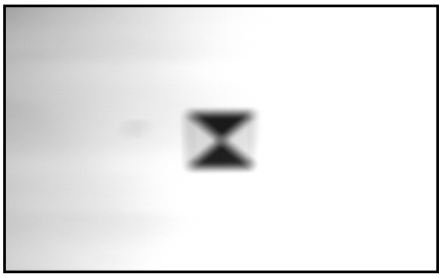
・ Poor quality images



Too bright



Too dark



Out of focus

[POINT]
If the focus cannot be adjusted properly, an extension tube may be required. If the focus has been changed, perform calibration again.

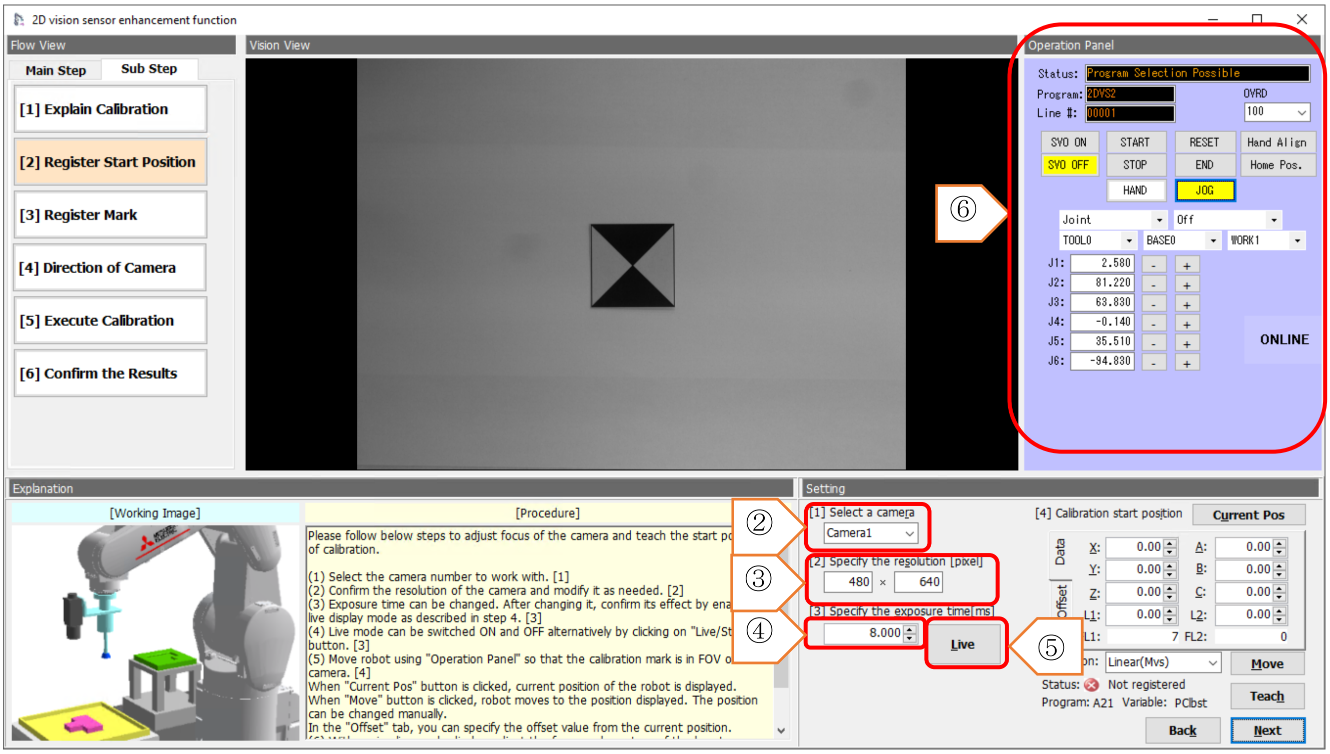
Click the [Current Pos] button to acquire and display the current position.
If required, adjust the displayed data (X, Y, Z, A, B, C), then click the [Move] button.

⑧ After determining the position, click the [Teach] button to register the <Calibration start position> as "PClbSt" of the WK* program. (The status indicator turns green upon completion of registration.)

[POINT]
Clicking the [Teach] button will only save position data on the screen, and will not cause an error in communication with the robot controller. The data is saved in an XML file upon click of the [Next] button. The data is applied to the position variable in the robot program upon click of the [Calibration Execution] button.

⑨ After completing the procedure, click the [Next] button.

<For hand cameras>



① Place the calibration mark as shown below.

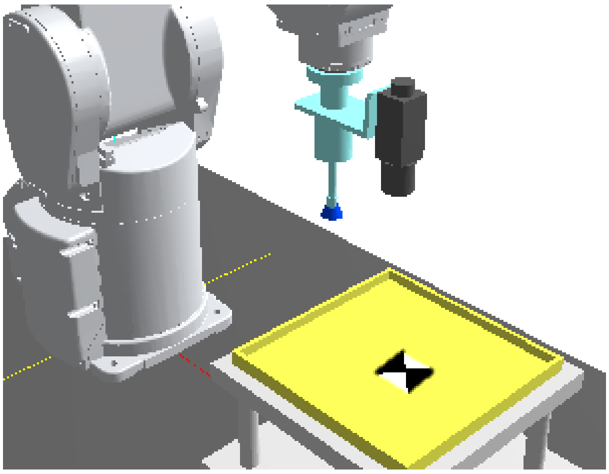


Fig. 8-5 Calibration start position (for hand cameras)

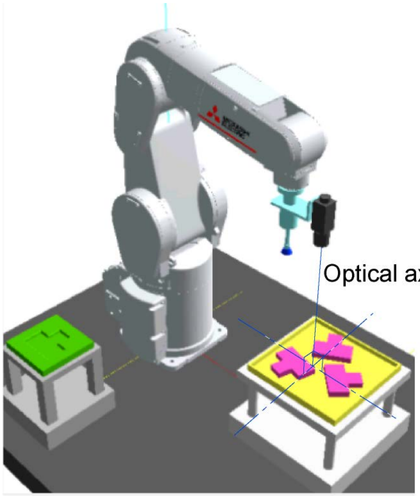
② Select the camera number to be set from "Select a camera" of the Setting panel.

③ Set the resolution of the camera in "Specify the resolution". The resolution of the vision sensor will be automatically set. Check that the resolution of the vision sensor has been set.

④ Set the exposure time of the camera in "Specify the exposure".

⑤ After setting "Specify the exposure", click the [Live] button.
A vision sensor image will appear in the Vision View.
(Live images can be switched ON and OFF with the [Live]/[Stop] button.)

⑥ Move the robot to a position where the calibration mark can be seen. In this situation, the following conditions must be met at the same time.
<Conditions for the calibration start position (for hand cameras)>
・ The optical axis of the camera and calibration mark (plane to be measured) are vertical to each other. (The measurement stand needs to be parallel to the robot stand.)
・ The calibration mark is positioned so that the moving distance is the same as the working distance (between the camera and the plane in which the workpiece is placed) in actual operation.
・ The calibration mark is displayed in the center of the image.



Move the mark to the center of the Vision View.
Move the robot using the Operation Panel, the position jump function, the teaching pendant, etc.
For the Operation Panel, refer to "19.2 How to operate the Operation Panel".
For the position jump function, refer to "19.3 Position jump".
Click the [Current Pos] button to acquire and display the current position.
If required, adjust the displayed data (X, Y, Z, A, B, C), then click the [Move] button.

⑦ If the Vision View does not display images properly, adjust the focus and aperture of the vision sensor lens.

⑧ After determining the position, click the [Teach] button to register the <Calibration start position> as "PClbSt" of the WK* program. (The status indicator turns green upon completion of registration.)

[POINT]
Clicking the [Teach] button will only save position data on the screen, and will not cause an error in communication with the robot controller. The data is saved in an XML file upon click of the [Next] button. The data is applied to the position variable in the robot program upon click of the [Calibration Execution] button.

⑨ After completing the procedure, click the [Next] button.

8.3. Registering a calibration mark

Register a mark to be used for auto calibration.
Follow the instructions shown in the Working Image panel and Procedure panel.
The operation method differs depending on whether PatMax is used for the vision sensor job.
The following figure shows when PatMax is not used.

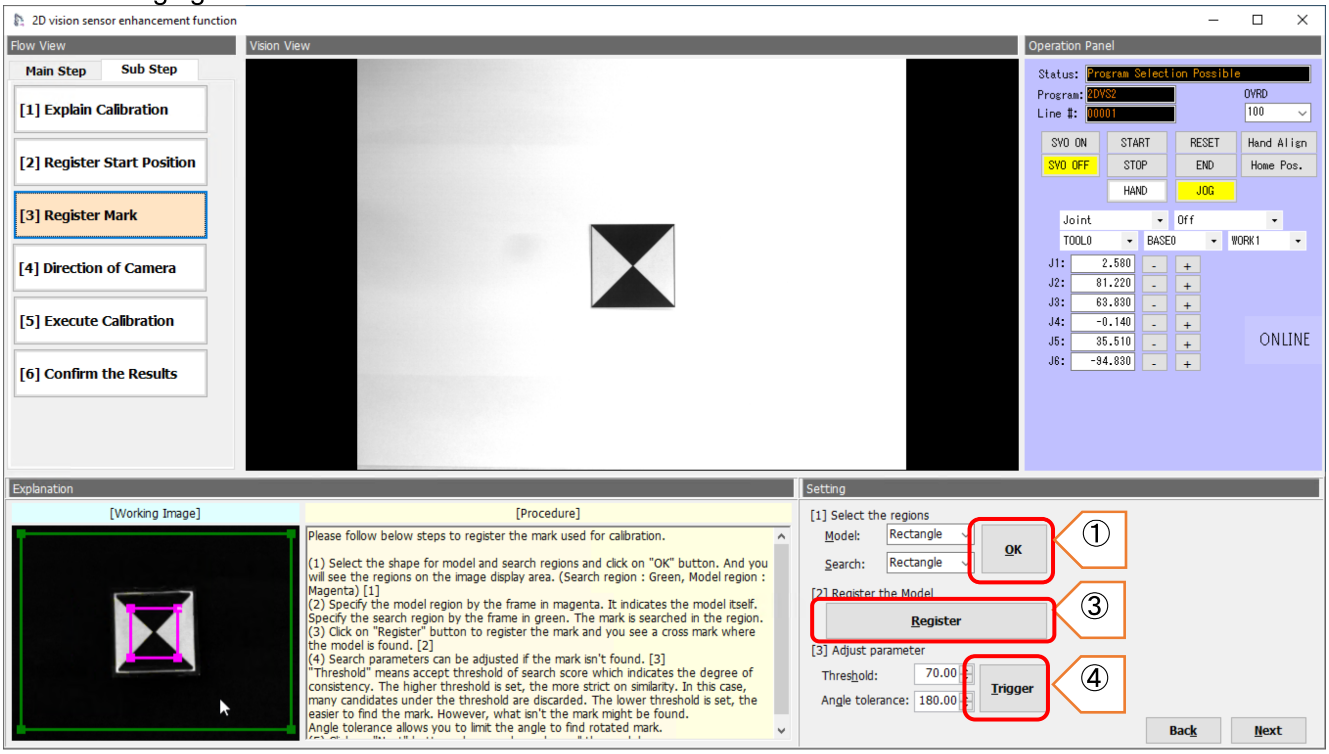
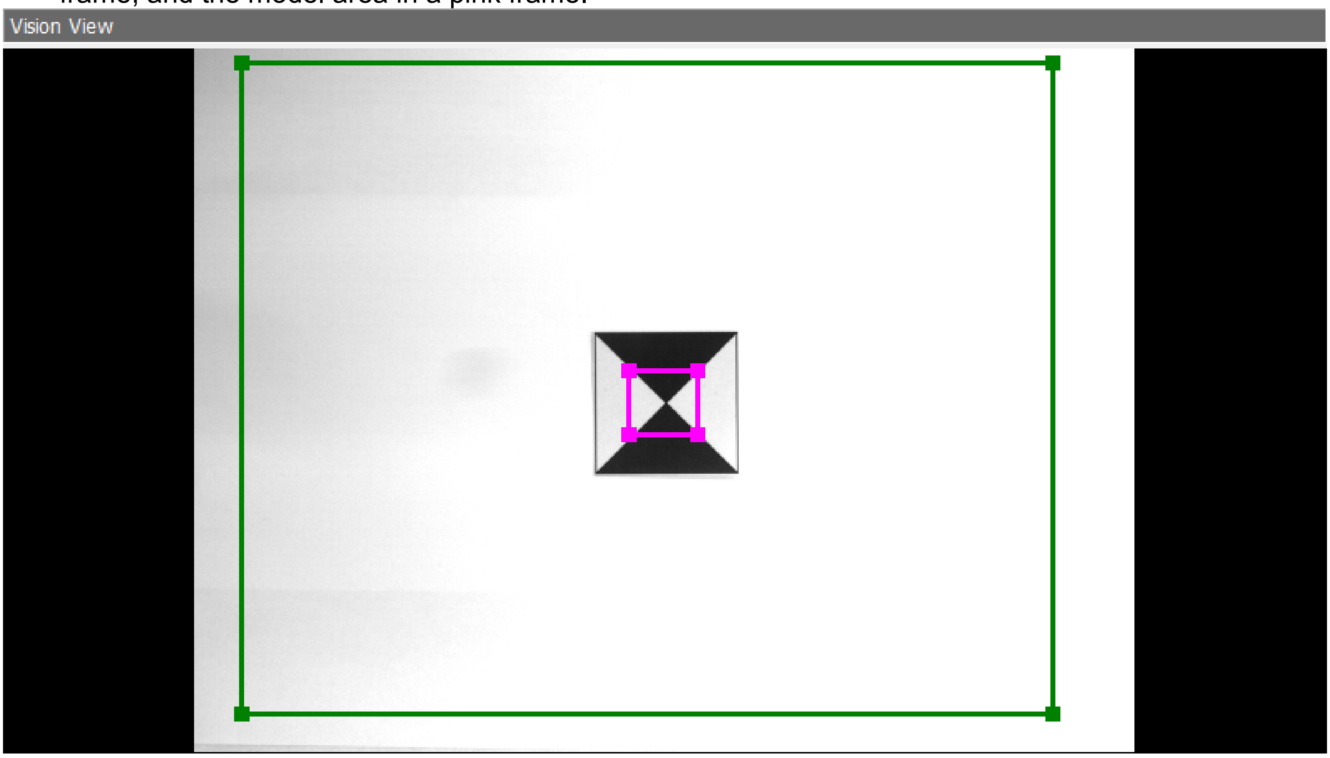


Fig. 8-6 Registering a calibration mark

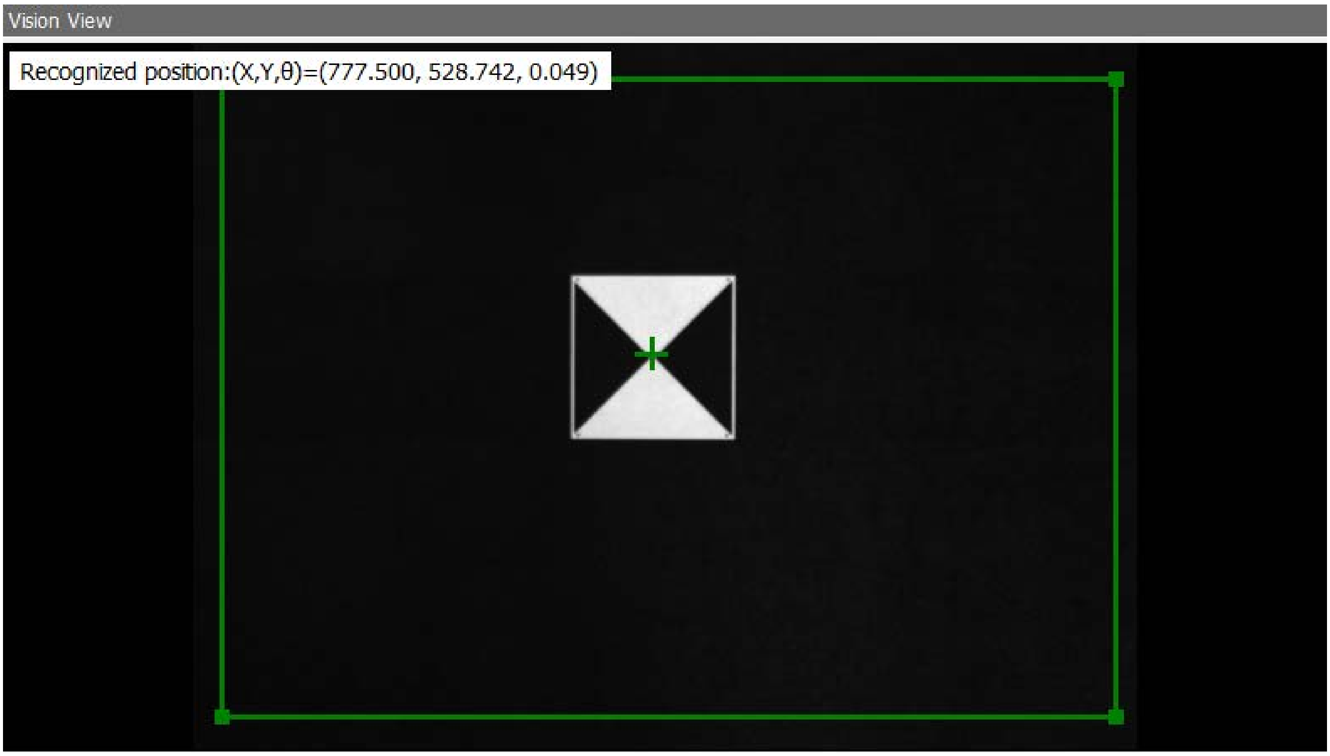
8.3.1. When not using PatMax

① Click the [OK] button of "Select the region" in the Setting panel. The search area will appear in a green frame, and the model area in a pink frame.



② Move the green frame to enclose an area used to detect the calibration mark. (During calibration, images are taken from various positions. Therefore, maximize the search area to the screen size.)
Move the pink frame to enclose a characteristic area of the calibration mark.

③ Click the [Registration] button.
If recognition is successful, a green crosshair will appear at the same position as the center of the pink frame that was set in step 1. The recognized vision coordinates will appear in the upper left.



If the position of the displayed crosshair is incorrect, adjust the following parameters.

Setting		Initial value	Function explanation
[3]	Threshold	70.00	Extracts images with the match percentage specified in this parameter or higher compared to the registered mark. (Example) Setting the value to "70" will show recognition information if there is a 70% or higher match between the mark displayed in the image and the registered mark.
[3]	Angle Range	180.00	Specify up to how many degrees the registered mark can rotate to be recognized.

④ Change the values and click the [Trigger] button. The image will be recognized using the latest parameters. Adjust the parameters for stable recognition.

⚠ CAUTION

After a parameter is changed, if the [Next] or [Back] button is clicked without clicking the [Trigger] button, the parameter on the screen returns to its initial value or to the value it was when you last clicked the [Trigger] button when you return to the same screen next time.

⑤ After recognition becomes possible, click the [Next] button.

8.3.2. When using PatMax

When PatMax is used, the contents in "Working Image", "Procedure", and "Setting" are different.
"[1] Select the regions" is disabled and cannot be used.
Clicking the [Register] button in "[2] Register the Model" will start the "Model Registration Editor". Register the calibration mark.
While the model registration editor is running, anything other than the operation panel of the 2D vision sensor enhancement function screen cannot be used.
For information on how to use the model registration editor, refer to "8.3.3. How to use the model registration editor".
The icon of the model registration editor is the same as that of the 2D vision sensor enhancement function.

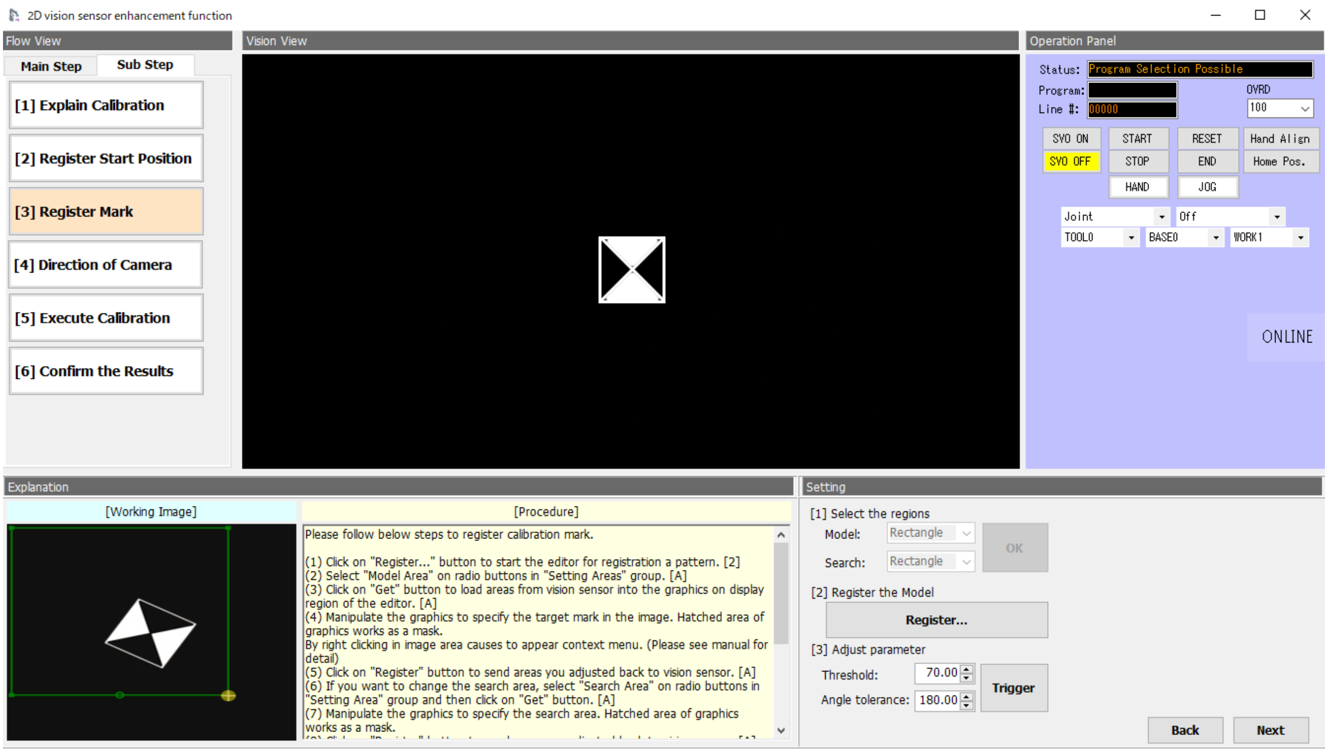


Fig. 8-7 Registering a calibration mark (when using PatMax)

After the registration is completed in the model registration editor, and the 2D vision sensor enhancement function screen appears again, the [Trigger] button will be enabled.
If the [Trigger] button is still disabled, restart the model registration editor and try the registration again.

8.3.3. How to use the model registration editor

When the model registration editor is started, the editor is automatically connected to the vision sensor and displays a captured image in standby mode.
Check that "Camera Number" and "IP Address" in [A] Connection are correct.
The name of an active job on the current vision sensor is added after the window title (Model Registration Editor).
"Offset X" and "Offset Y" in [C] Parameter Setting are displayed at workpiece registration, but not at the calibration mark registration.
For workpiece registration, refer to "9. REGISTERING A WORKPIECE".

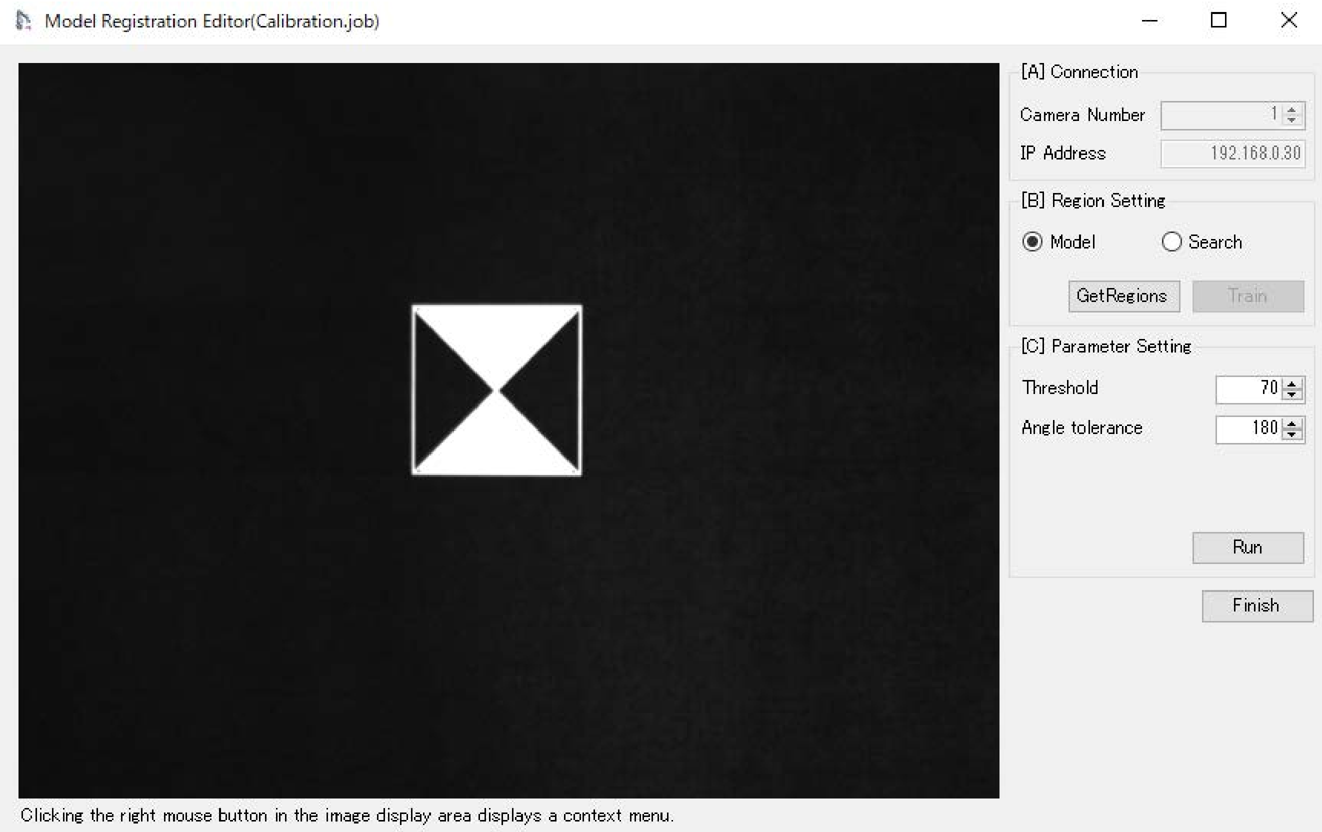


Fig. 8-8 Model registration editor startup screen

Select radio button [Model] or [Search] in [B] Region Setting, then click the [GetRegions] button.

Set the model region and the search region one by one.
While either one is being set, the other cannot be set.

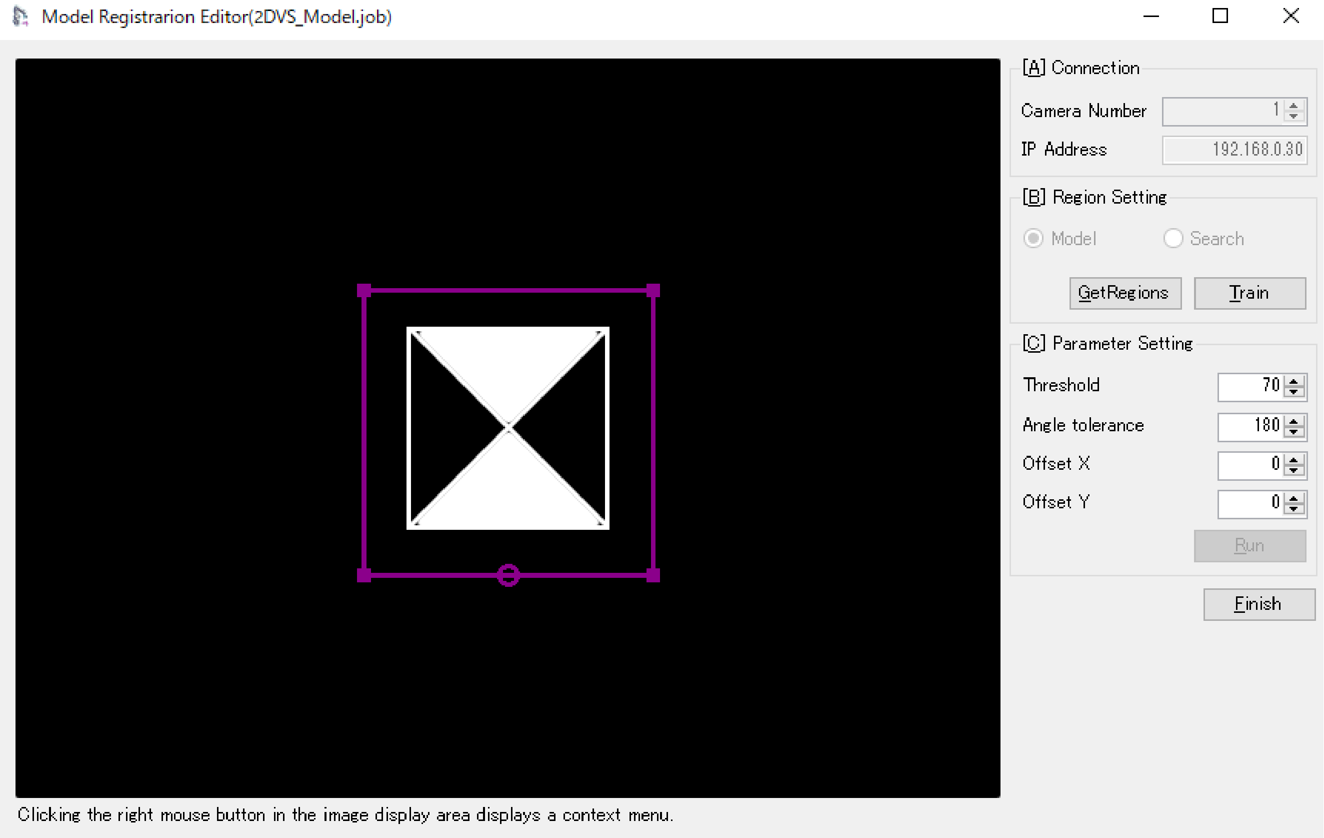


Fig. 8-9 Region acquisition

Manipulate the displayed graphics to adjust the region.
The following sections describe how to adjust the region.

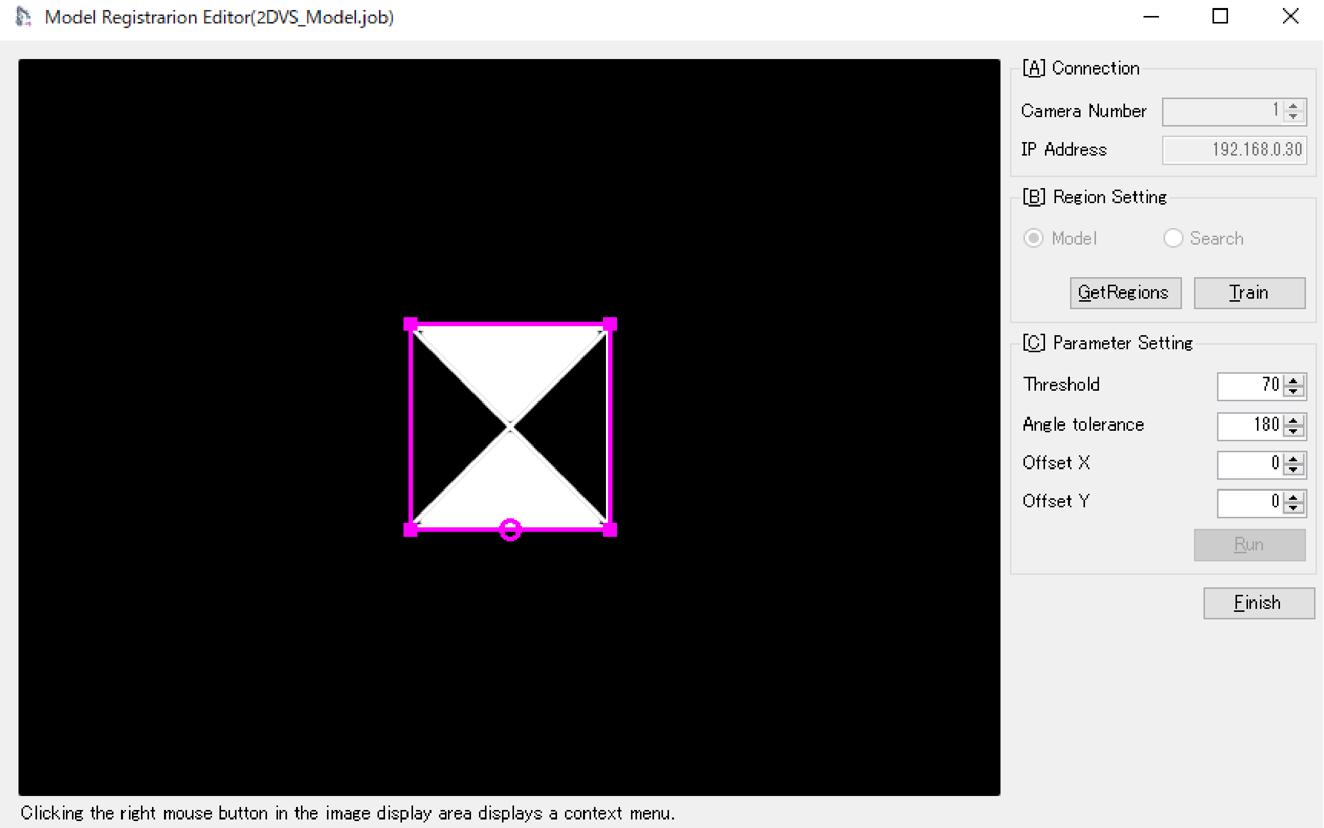


Fig. 8-10 Region display

Drag the mouse pointer at a position where the shape shown in the red circle in the following figure appears to resize the region.

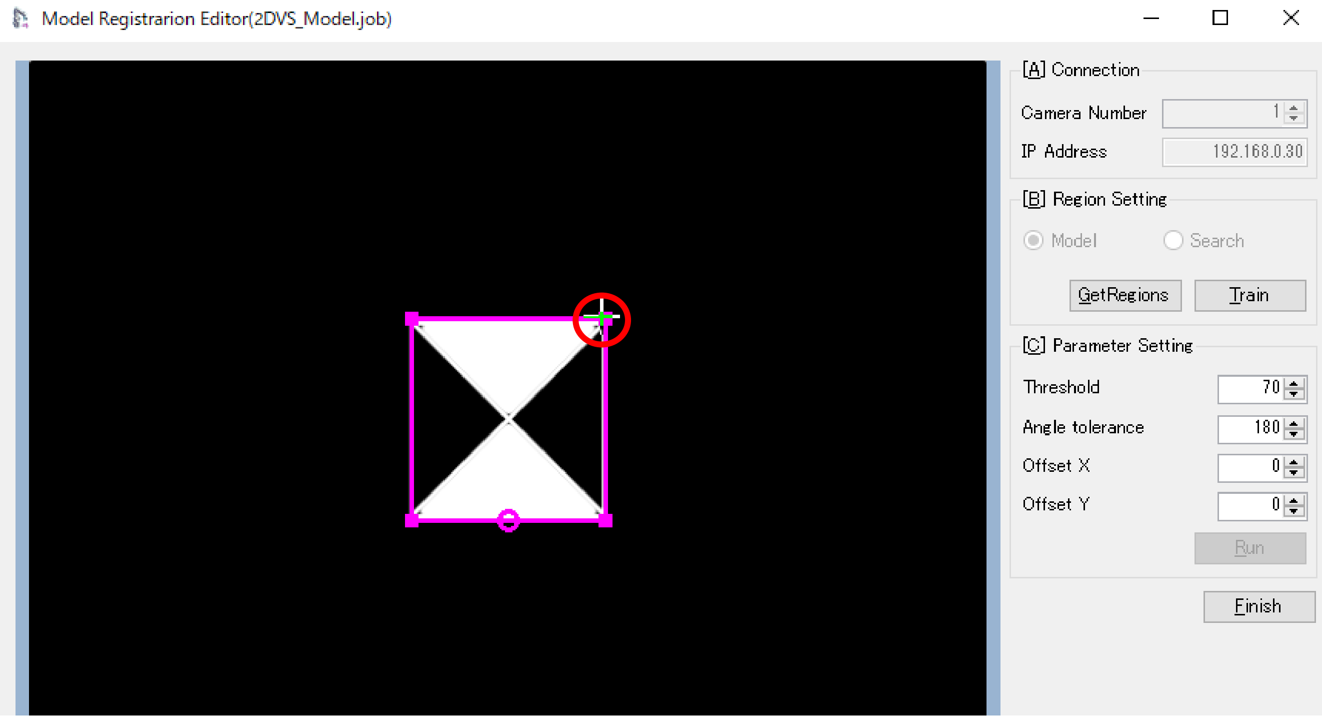


Fig. 8-11 Resizing

Drag the mouse pointer at a position where the shape shown in the red circle in the following figure appears to move a single region.

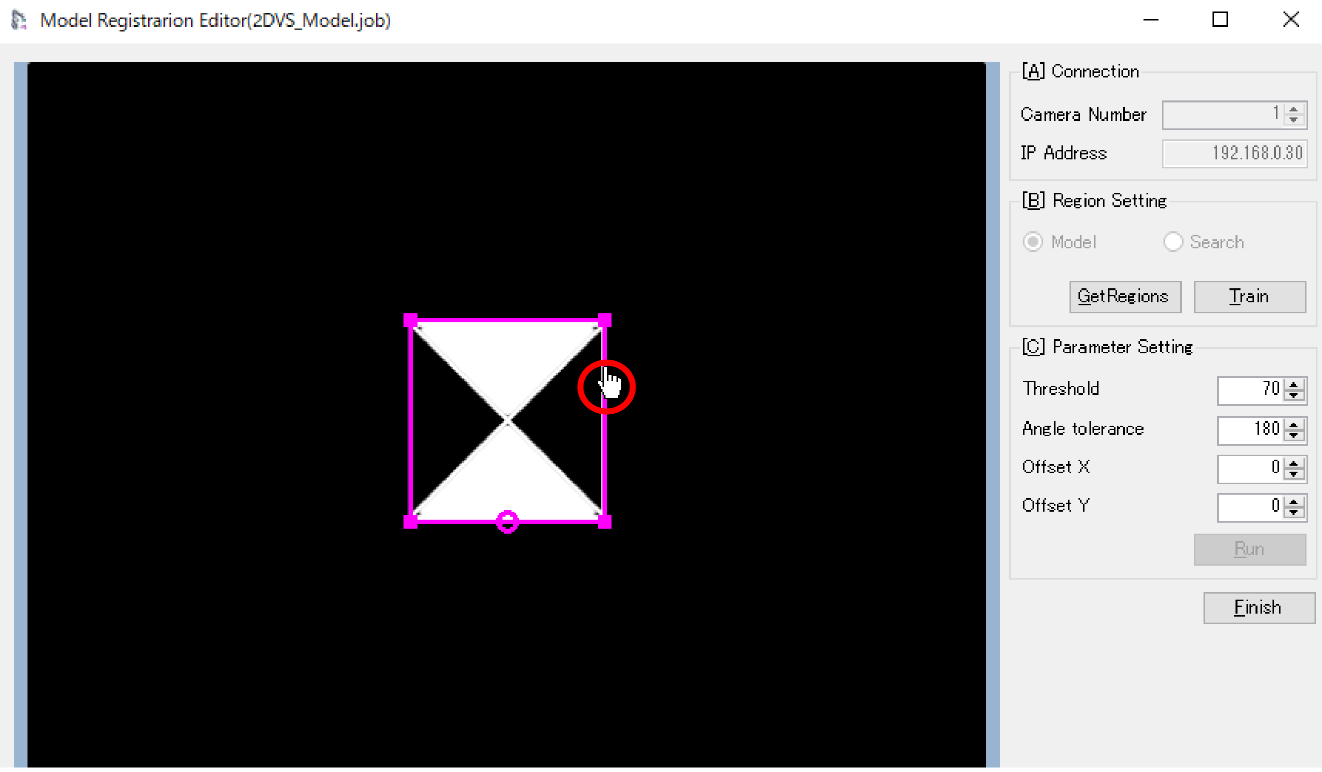


Fig. 8-12 Moving a single region

Drag the mouse pointer at a position where the shape shown in the red circle in the following figure appears to move all the regions with the relative positions maintained.
The pointer changes to that shape when the mouse is slightly outside the rectangle circumscribing all the regions.

Regions used in the model registration editor have "addition" and "subtraction" attributes.
In addition regions, the included characteristics are registered as models, and in subtraction regions, they are removed from the models.
The boundary of the addition region is represented by a solid line and the inside is represented by a transparent color. The boundary of the subtraction region is represented by a dashed line and the inside is represented by fine hatching.
When multiple regions overlap, the attributes of the upper layer region are prioritized.

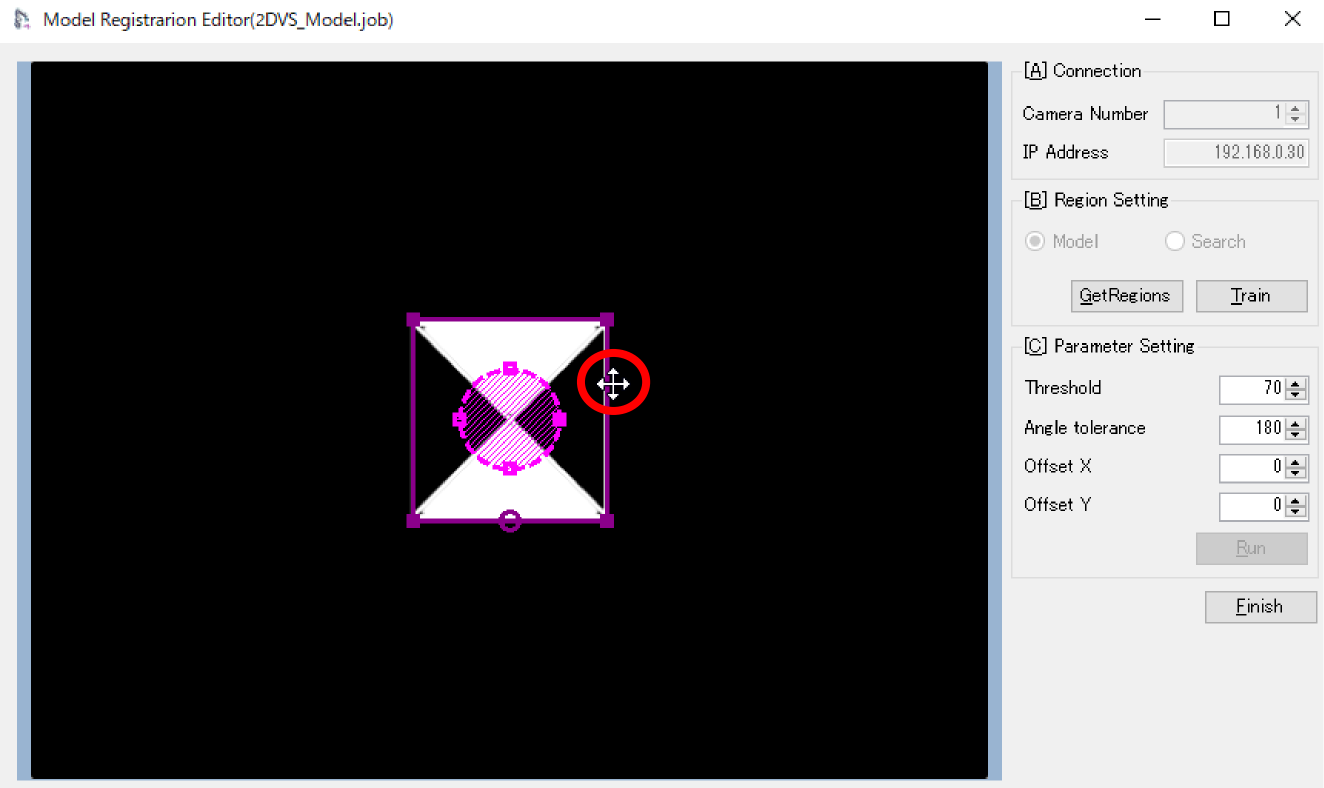


Fig. 8-13 Moving all graphics at once

Drag the mouse pointer at a position where the shape shown in the red circle in the following figure appears to change the rotational angle.
Currently, only rectangles have a rotation handle.

In the display region of the model registration editor, an active region is highlighted.
An active region is a newly added region or a region with the outline clicked.
Clicking where there are no regions will make all the regions inactive.

An invalid region appears when a rectangle region is rotated or multiple regions are placed away from one another.
Invalid regions are displayed with a thin dashed outline and fine hatching.
The characteristics contained in the invalid region are not registered as models.

Fig. 8-14 Rectangle rotation

Double-click on a side of the polygon to add a vertex.

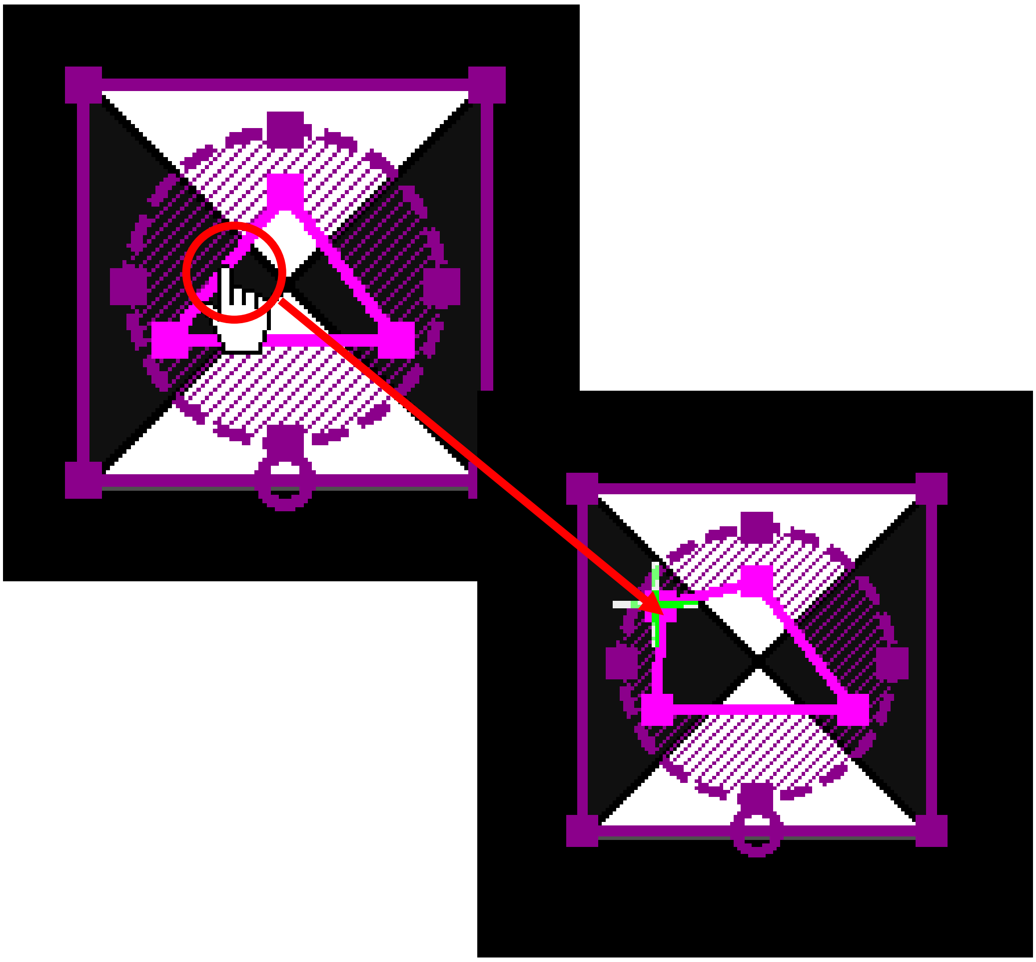


Fig. 8-15 Adding a vertex to the polygon

Double-click on the vertex of the polygon to delete the vertex.

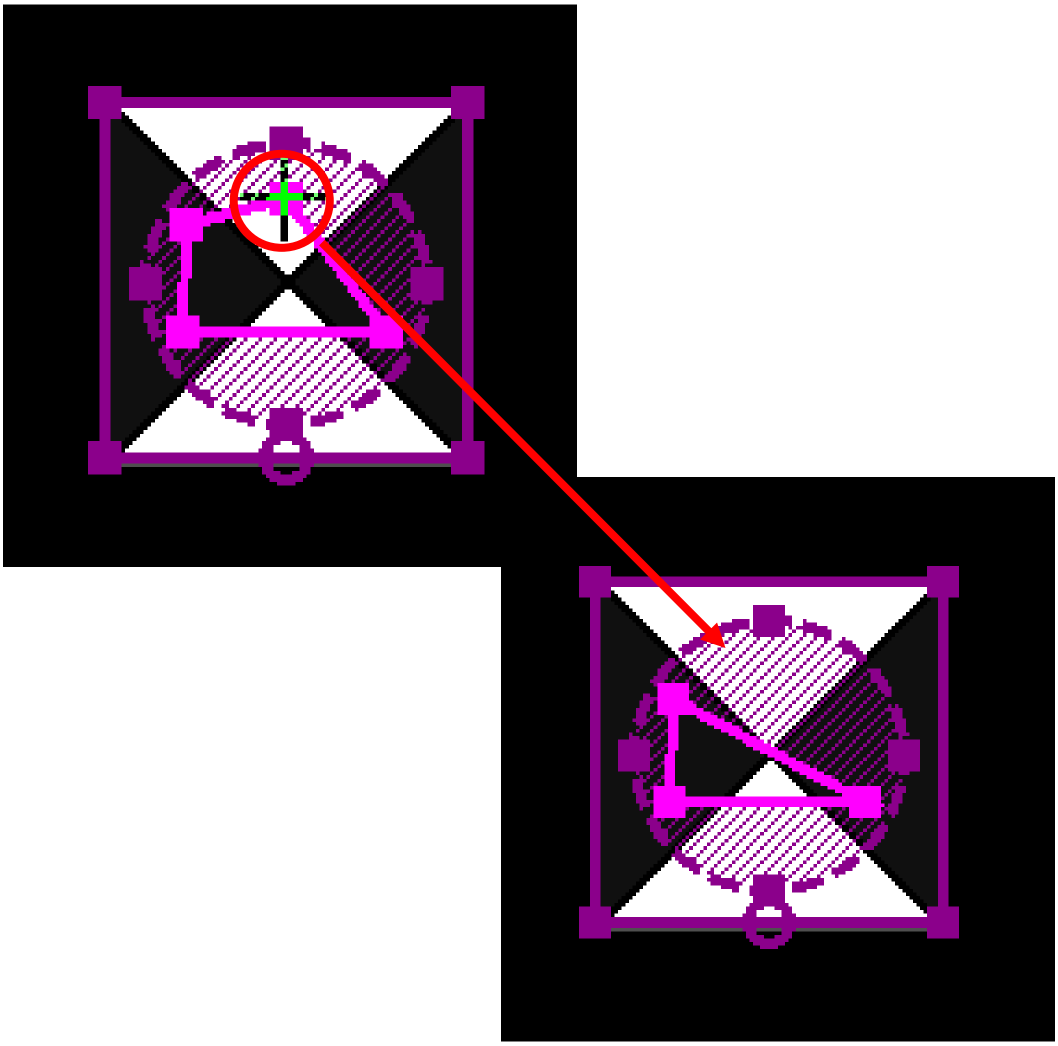


Fig. 8-16 Deleting the vertex of the polygon

Right-click on the display region to display the context menu.

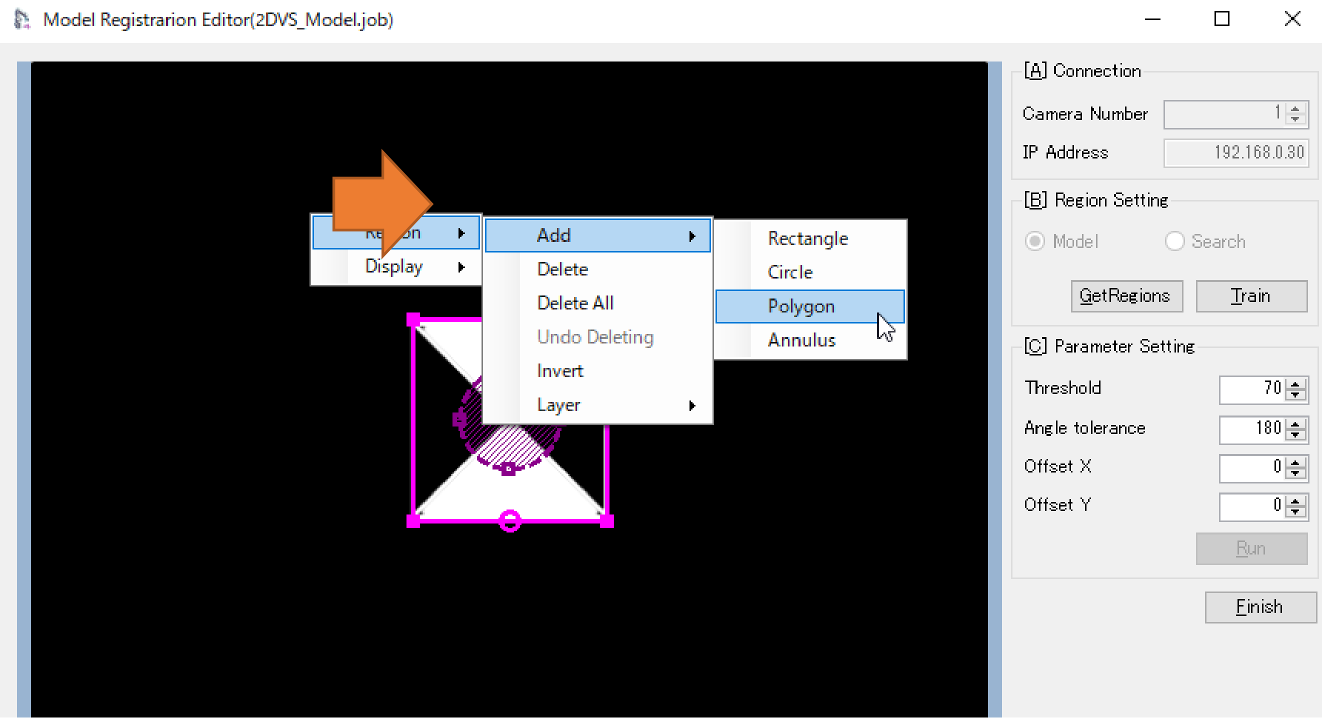


Fig. 8-17 Context menu

The following table lists the functions for each item.

Table 8-1 Context menu

Large category	Medium category	Small category	Function
Region	Add ※1)	Rectangle	Adds a rectangle region, which has an "addition" attribute, to the screen, and makes the region active.
		Circle	Adds a circle region, which has an "addition" attribute, to the screen, and makes the region active.
		Polygon	Adds a triangle region, which has an "addition" attribute, to the screen, and makes the region active.
		Annulus	Adds a region with two concentric circles, which has an "addition" attribute in the inner circle and a "subtraction" attribute in the area between the inner circle and outer circle, to the screen, and makes the region active.
	Delete		Deletes the active region.
	Delete all		Deletes all the regions.
	Undo deleting ※2)		Restores the last deleted region.
	Invert Attribute		Inverts the attributes of the active region. Converts the attribute from "addition" to "subtraction", and vice versa.
	Layer	To Top	Moves the active region to the top layer.
		Upward	Moves the active region to the next upper layer.
		Downward	Moves the active region to the next lower layer.
		To Bottom	Moves the active region to the bottom layer.
Display	Expand		Enlarges the image and graphics in the display region two times centering on the mouse click point. The maximum scaling factor is 8 times the default display.
	Shrink		Reduces the image and graphics in the display region to 0.5 times. The minimum scaling factor is that of the default display.
	Default magnification		Displays the image and graphics in the display region at the default scaling factor. With this scaling factor, all the regions of the image are displayed.

※1) The last added region is placed on the top layer.

※2) Even if the active region is deleted, it is inactive when restored.
The restored region is placed on the top layer regardless of in which layer the region was placed before the deletion.
Undeletion for "Delete all" restores the regions with them placed in the same layer as before.

After the region has been adjusted, click the [Train] button in [B] Region Setting.
This operation will overwrite the vision sensor's region data and cannot be canceled once the [Train] button is clicked.
Enter the "Threshold" and "Angle tolerance" in [C] Parameter Setting, then click the [Run] button.
The parameters will be registered, and the image will be captured and recognized.

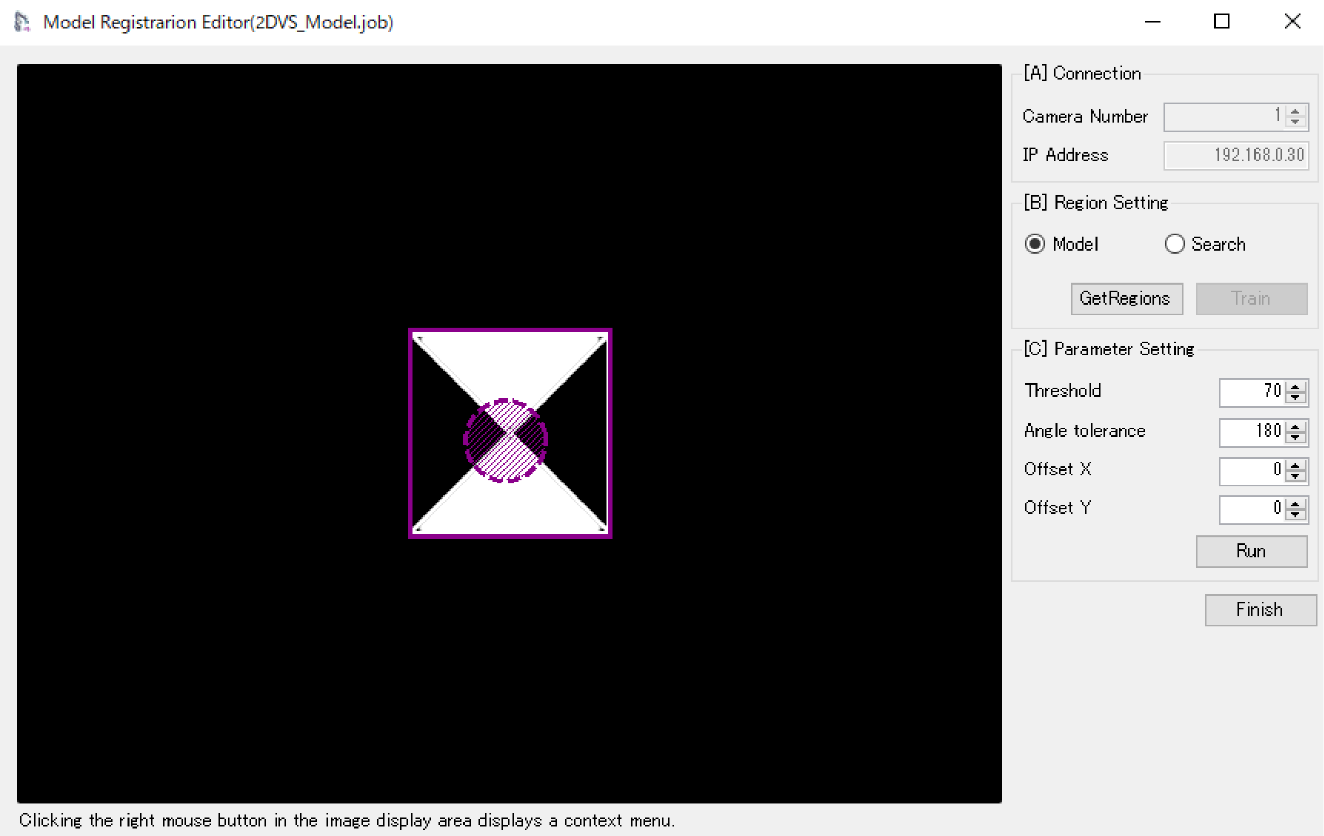


Fig. 8-18 Model registration

A cross mark appears at the position where the registration model is found.
After the checking process, click the [Finish] button to complete the procedure.

Fig. 8-19 Exiting the model registration editor

8.4. Camera installation direction

Specify which direction the camera faces at the calibration start position in the robot tool coordinates (hand coordinates).

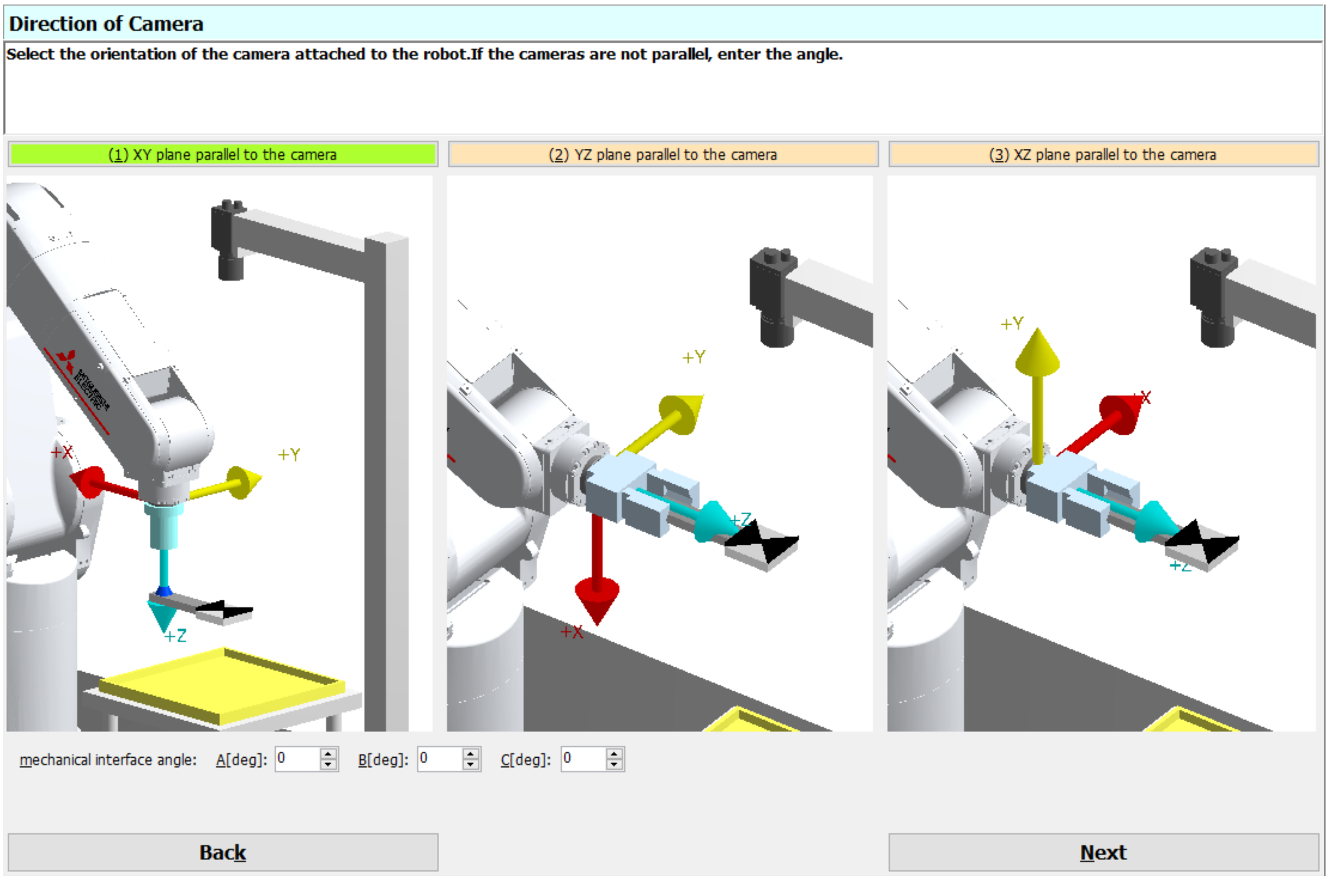


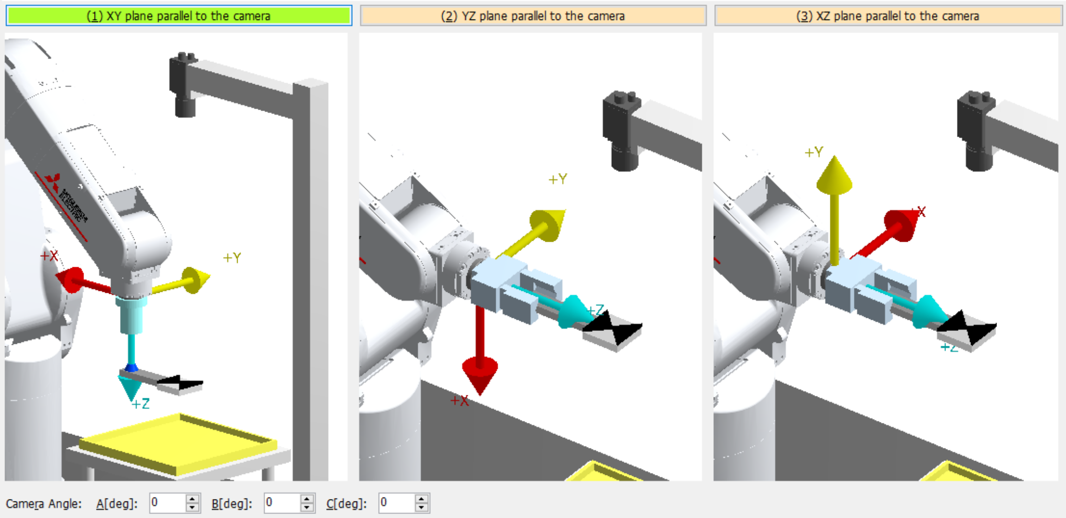
Fig. 8-7 Specifying which direction the camera faces

Select the directional relationship from the three illustrations.
Clicking any of the illustrations will change the color of the title. Values A, B, and C of "Camera Angle" will also be changed automatically.

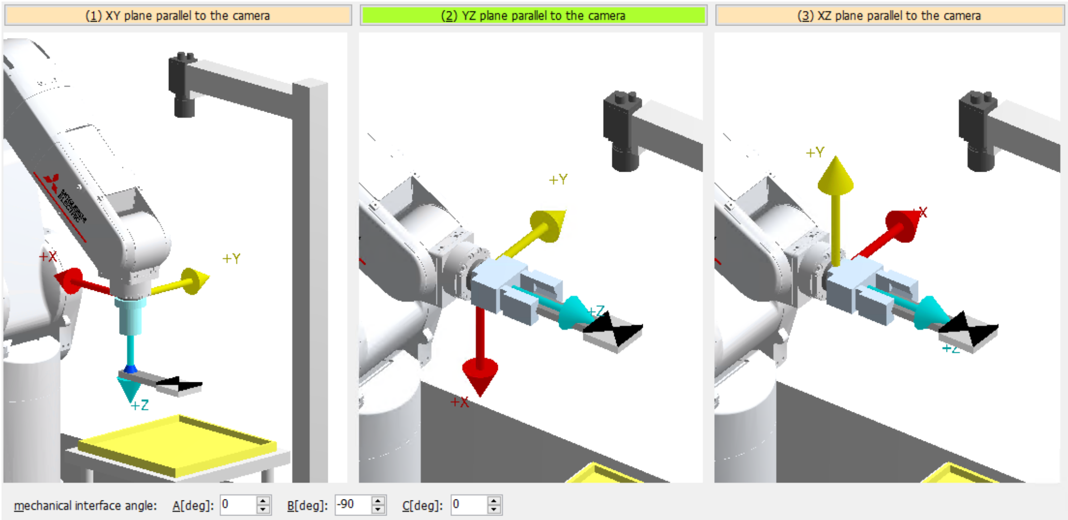
For special settings such as attaching the camera at an angle, enter values A, B, and C of "Camera Angle" manually.

Camera Angle (coordinate system definition)
Enter rotational angles used to change from the reference posture to the current posture. The definitions of the reference posture and the coordinate system during rotation are as follows: <Reference posture> Posture for "(1) XY plane parallel to the camera" <Coordinate system> Tool coordinate system For example, if a fixed downward-facing camera is used, clicking each of the three illustrations will change the values of Camera Angle as follows:

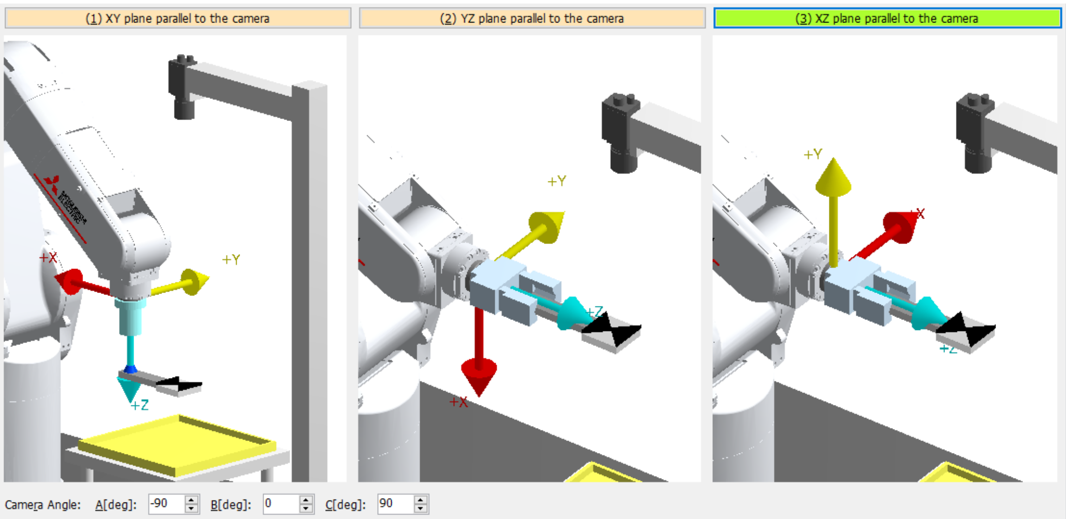
Camera Angle for "(1) XY plane parallel to the camera" (reference)



Camera Angle for "(2) YZ plane parallel to the camera"



Camera Angle for "(3) XZ plane parallel to the camera"



The following table shows the values of Camera Angle for each camera position.

Camera position	**Camera Angle**		
	(1) XY plane parallel to the camera	**(2) YZ plane parallel to the camera**	**(3) XZ plane parallel to the camera**
Fixed upward-facing camera	A: 0 B: 0 C: 0	A: 0 B: -90 C: 0	A: -90 B: 0 C: 90
Fixed downward-facing camera	A: 0 B: 0 C: 0	A: 0 B: -90 C: 0	A: -90 B: 0 C: 90
Fixed side-facing camera	A: 0 B: 0 C: 0	A: 0 B: 90 C: 0	A: 90 B: 0 C: 90
Hand camera	A: 0 B: 0 C: 0	A: 0 B: -90 C: 0	A: -90 B: 0 C: 90

The robot coordinate system can be checked using the following steps.

① In the RT ToolBox3 workspace, double-click [3D Monitor].

② The robot tool coordinates can be checked on the displayed 3D Monitor screen.

⚠ CAUTION

The operation of auto calibration is determined according to the angles set on this screen.
Incorrect settings will move the robot in an unexpected direction during auto calibration.

8.5. Performing calibration

Perform calibration according to the installation position of the vision sensor.
Follow the instructions shown in the Working Image panel and Procedure panel.

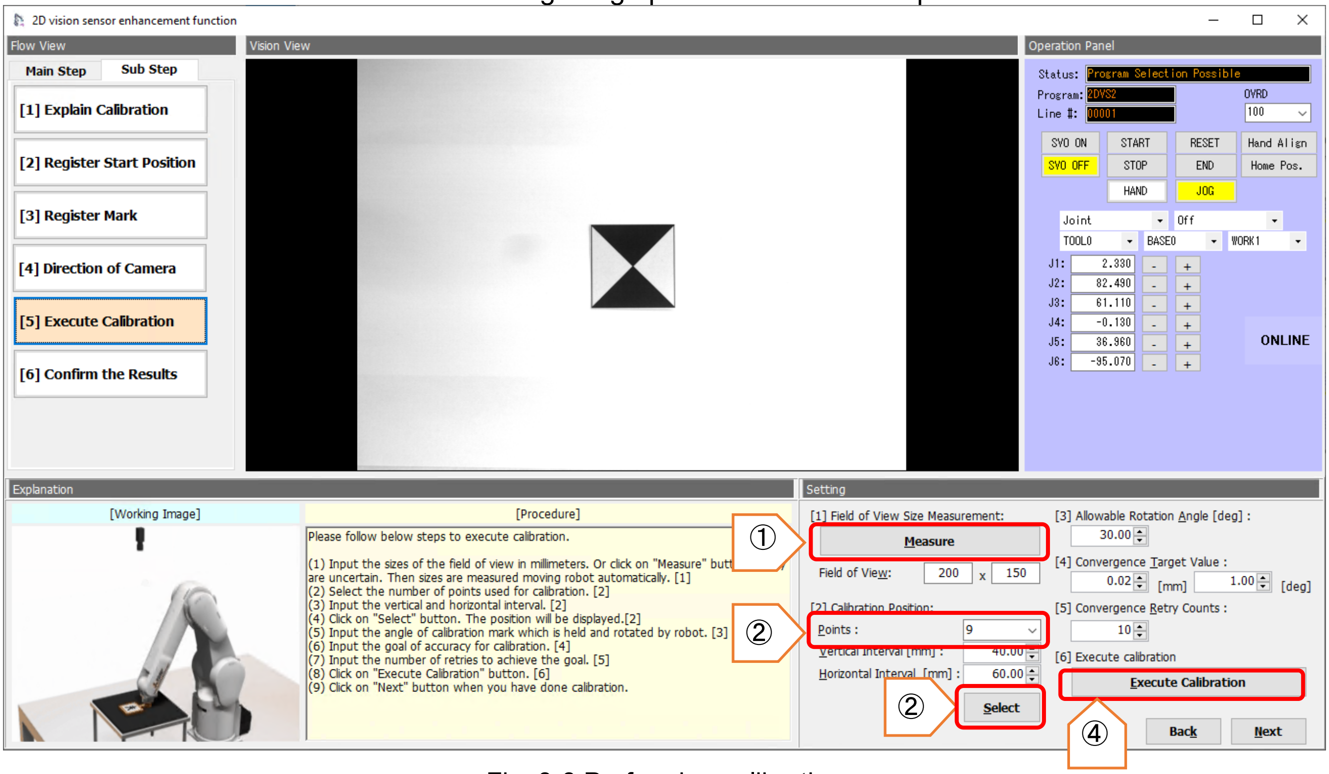


Fig. 8-8 Performing calibration

① Click the [Measure FOV] button in the Setting panel to move the robot and measure the field of view size. Note that the robot will move the distance specified in "Movement Interval (Vertical)" from the starting point.
If the vision sensor's field of view size is known, enter the size.

② Set the number of points to which the robot moves during calibration in "Calibration Points", then click the [Select] button.
The points to which the robot moves during calibration will appear in the Vision View. (As the number of points to which the robot moves is larger, calibration becomes more stable.)

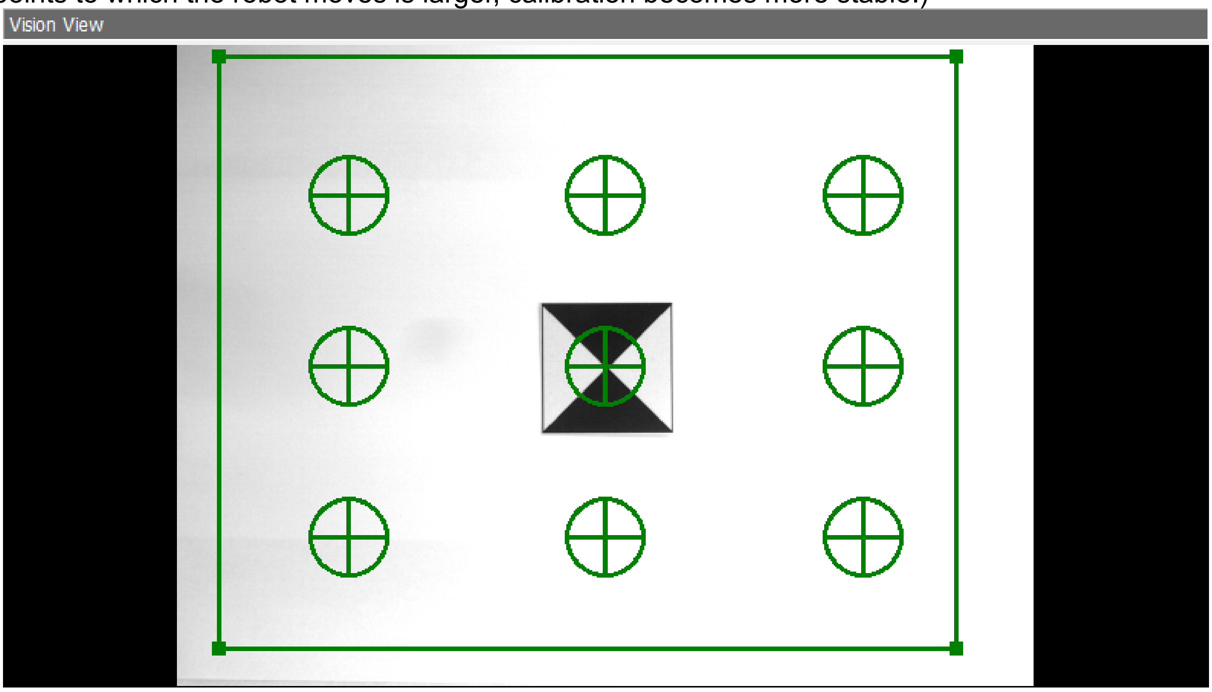
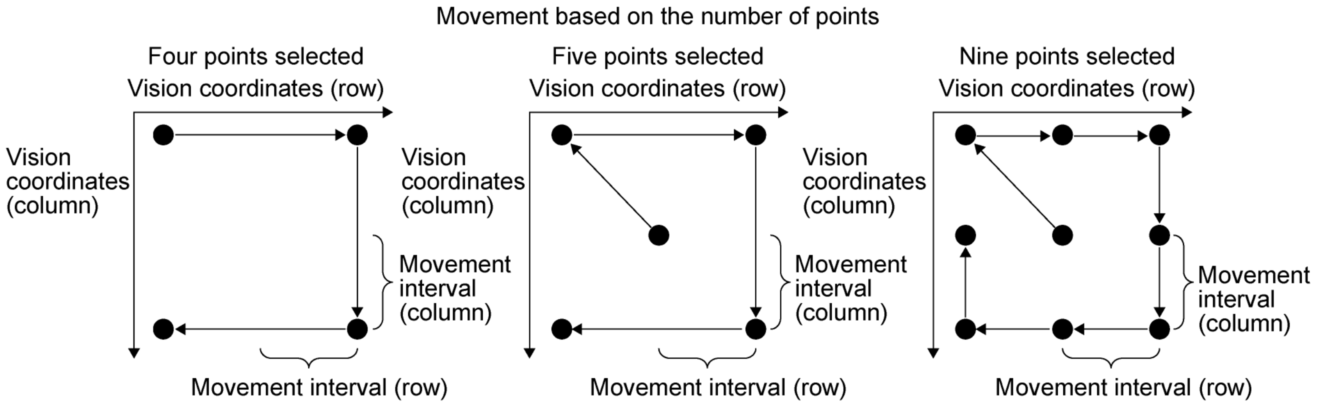


Fig. 8-9 Calibration points (9 points as an example)

③ To change the movement interval during calibration, correct the values of "Movement Interval (Vertical)" and "Movement Interval (Horizontal)", then click the [Select] button indicated by number 2. The movement interval is defined according to the number of calibration points as shown below.



If the value specified in "Movement Interval" is too large, the interval will be corrected so that it remains within the field of view. However, if an error occurs, decrease the value of "Movement Interval".

④ Change the values of "Allowable Rotation Angle", "Convergence Target Value", and "Convergence Retry Counts" as necessary.
After changing the values, click the [Calibration Execution] button.
If an error occurs, follow the solutions described in "17.1 Error details of the vision sensor enhancement function" and "18 TIPS".

<Robot movement during calibration>
During calibration, the robot moves within the following range while it maintains the position parallel to the camera imaging plane.

[Horizontal direction]: The robot moves within the area where the calibration mark can be recognized (within the search area).
[Vertical direction]: The robot moves ±10 mm vertically from the calibration start position.
[Rotational direction]: The robot moves so that the mark displayed in the center of the image rotates. The rotational angle used in this operation is the value specified in "Allowable Rotation Angle".

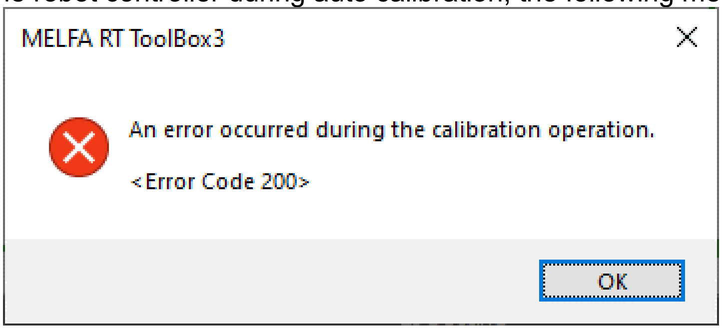
⚠ CAUTION

Clicking the [Calibration Execution] button will move the robot. Before clicking the button, check that this operation does not cause entanglement of cables around the hand and interference.

If calibration fails or higher accuracy is required, adjust the following parameters, then click the [Calibration Execution] button again. The calibration accuracy can be checked in the setting window shown in "8.6 Results display.

Setting	Initial value	Function explanation
Movement Interval (Vertical)	40.00	The robot moves to the points equal to the number of points specified in "Calibration Points". Specify the movement interval (amount of offset from the center). Specify how many millimeters the robot moves in the vertical direction in the field of view.
Movement Interval (Horizontal)	60.00	The robot moves to the points equal to the number of points specified in "Calibration Points". Specify the movement interval (amount of offset from the center). Specify how many millimeters the robot moves in the horizontal direction in the field of view.
Allowable Rotation Angle	30.00	When the robot rotates the vision, camera, or mark, the center of the camera or mark can be found. Specify the travel angle of this operation.
Convergence Target Value	0.02	The robot moves the mark to the center of the vision sensor's field of view. Adjustments are repeated until the deviation is in this setting range. Specify a permissible deviation.
	1.00	For applications using a hand camera, upward-facing camera, or side-facing camera, the robot makes automatic adjustments so that the vision sensor and workpiece are parallel to each other. Specify a permissible deviation.
Convergence Retry Counts	10	Specify the maximum number of times that the robot moves the mark to the center of the vision sensor's field of view. When the number reaches the set value, an error will be output, and the robot will stop.

If an error occurs in the robot controller during auto calibration, the following message will appear.



For the solution, refer to "17.2 List of errors related to the vision sensor enhancement function".

⑤ If calibration is successful, click the [Next] button.

8.6. Results display

If calibration is successful, the calibration results (residual deviations) will appear in the Calibration accuracy field.
If required, deselect check boxes for calibration points and click the [Recalculation] button. Calibration data can be calculated again. (To improve the accuracy in a specific area in the imaging area, it may be favorable if unrelated check boxes are deselected.)

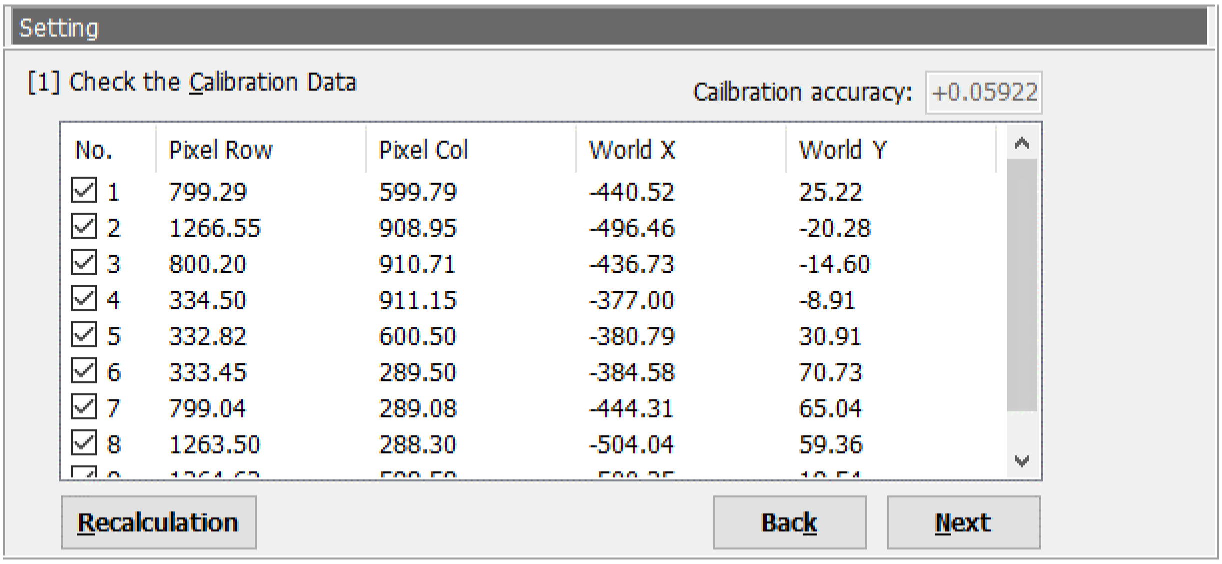


Fig. 8-10 Calibration results (when 9 points are selected for calibration)

After checking the calibration accuracy, click the [Next] button.

9. REGISTERING A WORKPIECE

Set the workpiece shape to be recognized by the vision sensor and the recognition conditions.

9.1. How to register a workpiece

You can check steps required to register a workpiece on video.
(The workpiece registration video does not include sound.)

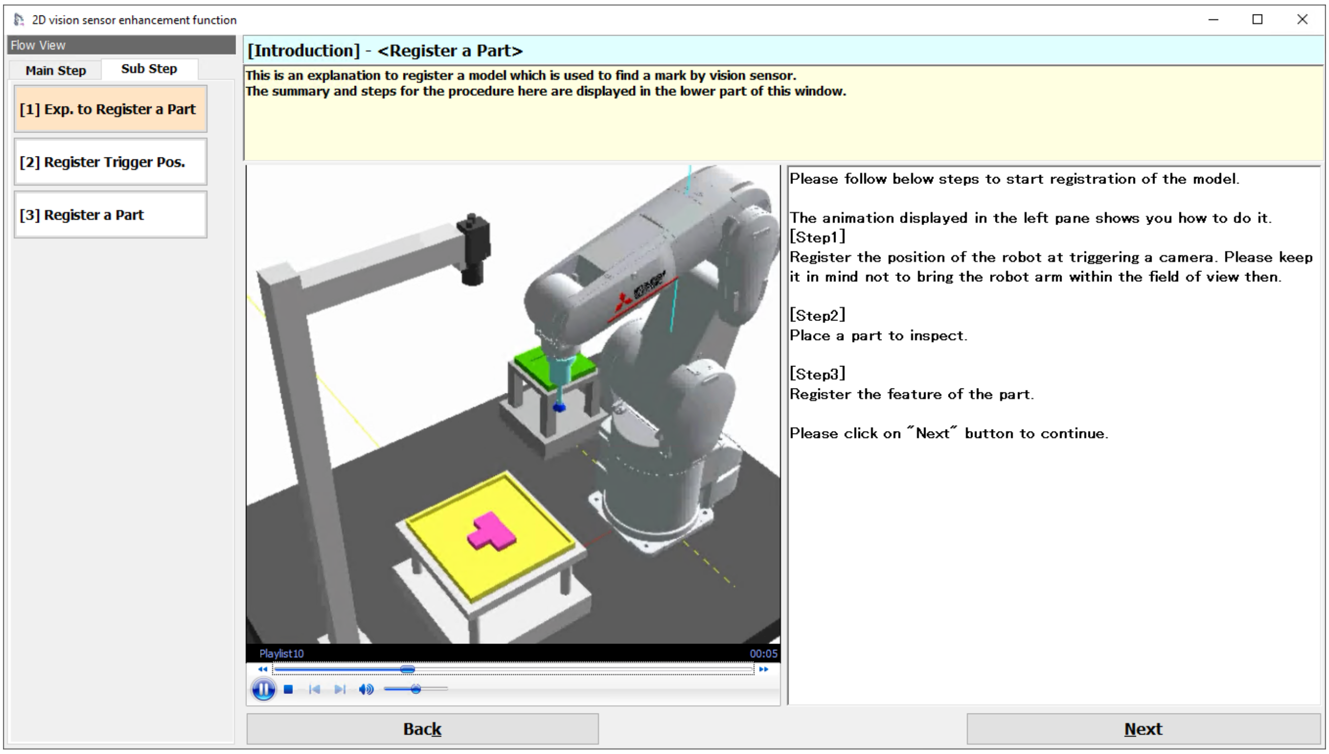


Fig. 9-1 Registering a workpiece

The panel on the left can be used to play back, pause, select where to play back, and for other purposes.

The panel on the right shows the steps for configuring the vision settings.

After checking the steps, click the [Next] button.

9.2. Registering a trigger position

Specify a position of the robot for workpiece recognition.
Follow the instructions shown in the Working Image panel and Procedure panel.

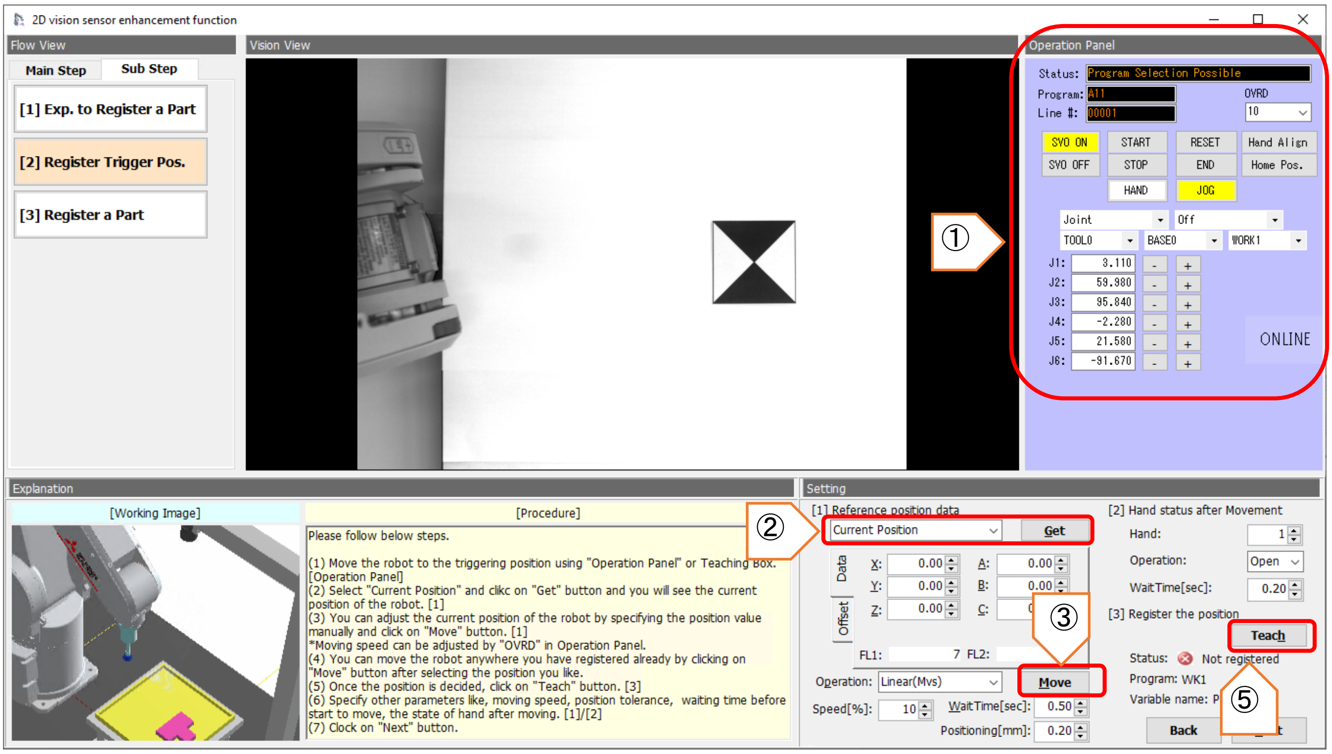


Fig. 9-2 Screen to register an imaging position

① Move the robot to a position for workpiece recognition using the Operation Panel or the teaching pendant.
In this situation, the following conditions must be met.
■ Conditions for the trigger position
1. The optical axis of the camera and the plane where the workpiece is recognized intersect at a right angle.
2. The working distance (between the workpiece and sensor) is the same as the distance taught during calibration.
3. The workpiece is displayed in the center of the image.

For hand cameras, fixed upward-facing cameras, and fixed side-facing cameras, the optical axis is automatically aligned during calibration. Moving the robot while it maintains the posture at the time calibration is complete facilitates teaching.

② Select "Current Position" and click the [Get] button in "Reference position data" of the Setting panel. The current position of the robot will be acquired.

③ If fine adjustments are required, edit the displayed position data (X, Y, Z, A, B, C), and click the [Move] button. The robot will move (using the position jump function) according to the settings (Joint movement or XYZ movement) selected in "Operation".
For details on the position jump function, refer to "19.3 Position jump".

The robot can move to the position displayed in the [Data] tab. Clicking the [Offset] tab can move the robot from the position displayed in the [Data] tab by a specified offset value. (The values of [Offset] are not applied unless the [Offset] tab is active.)

⚠ CAUTION

Clicking the [Move] button will move the robot. Before clicking the button, check that this operation does not cause entanglement of cables around the hand and interference.

④ To move the robot to a registered position, specify a position other than "Current Position", and click the [Move] button.
⑤ After determining the position, click the [Teach] button. The trigger position is registered as "PVsTrg*" of the WK* program.
⑥ To improve takt time, operation settings (speed, positioning accuracy, waiting time, etc.) must be adjusted for movement to the taught position. However, fine adjustments are not required in this process because you have an opportunity to make fine adjustments while checking a series of operations after all operation settings have been configured. Refer to "12.2 Modifying an operation".
⑦ After configuring the settings, click the [Next] button.

9.3. Registering a workpiece (model)

Register the workpiece to be recognized and transported.
Follow the instructions shown in the Working Image panel and Procedure panel.
The operation method differs depending on whether PatMax is used for the vision sensor job.
The following figure shows when PatMax is not used.

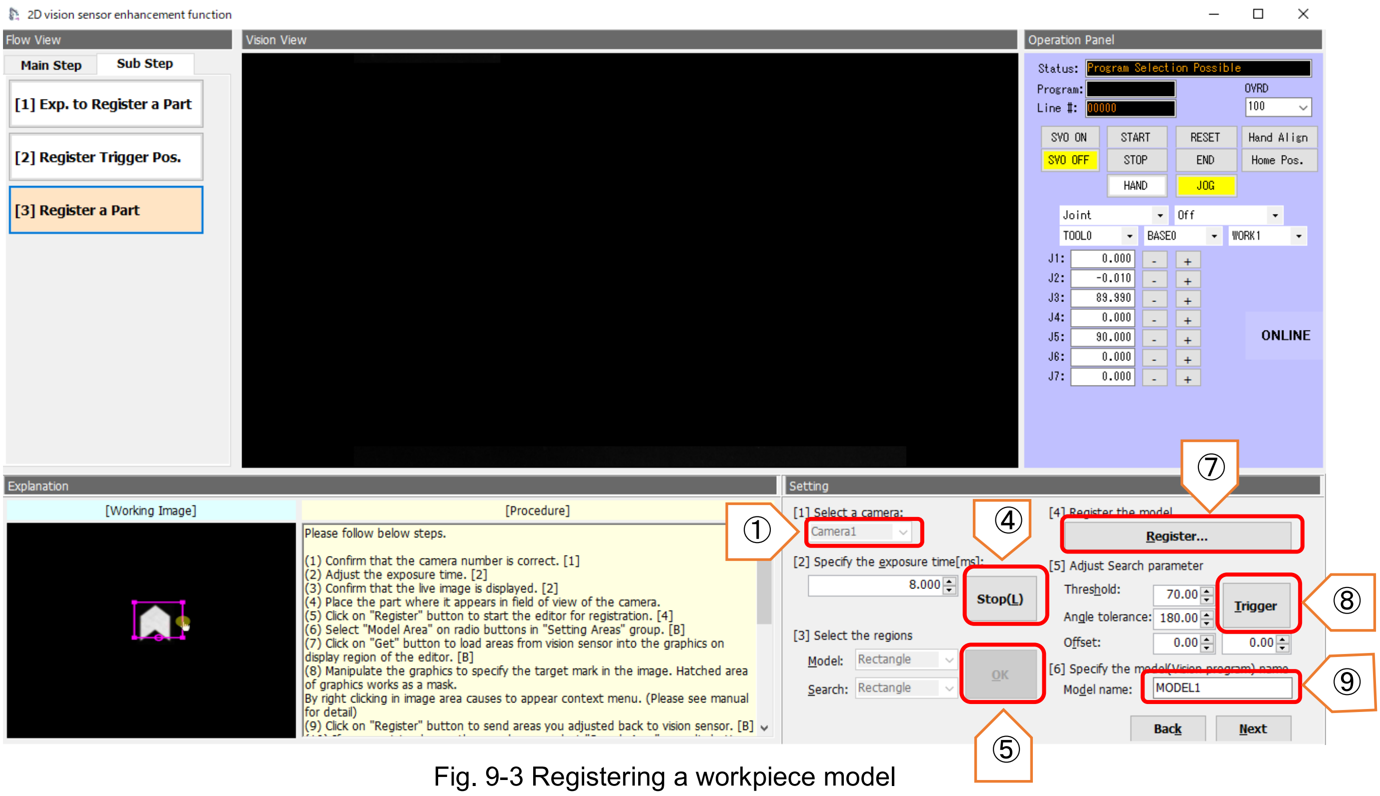


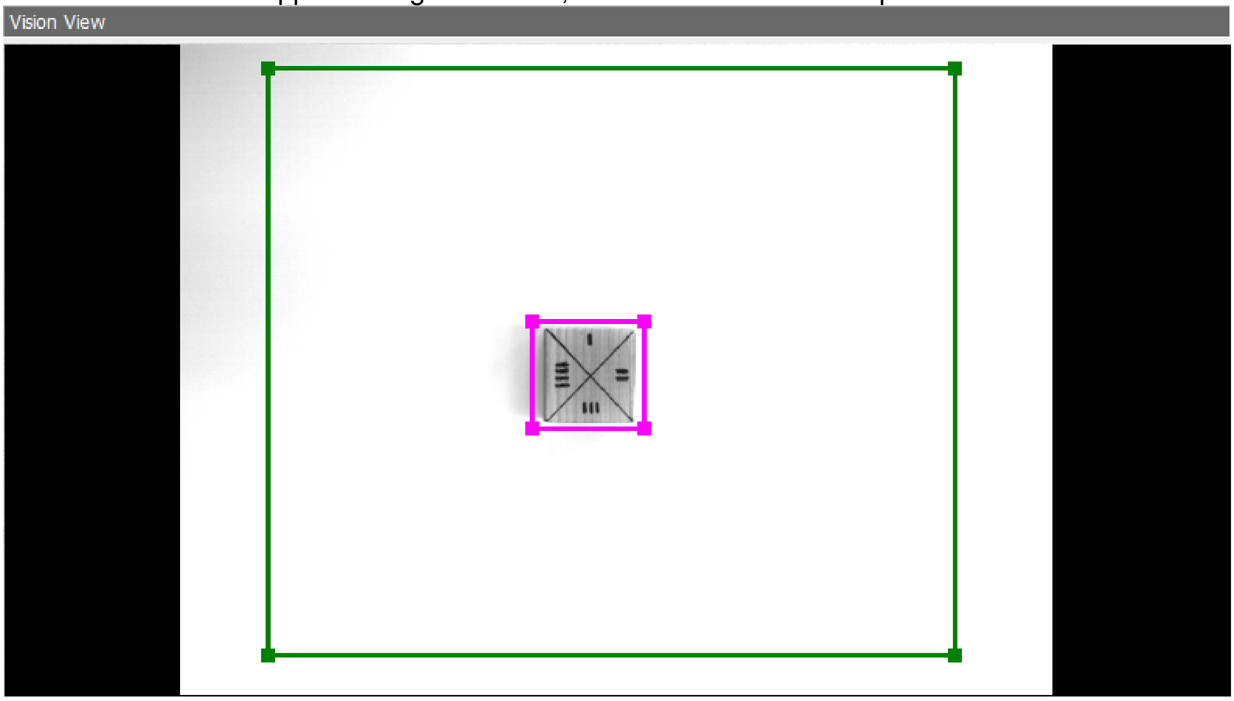
Fig. 9-3 Registering a workpiece model

9.3.1. When not using PatMax

① Select a camera to be used for workpiece registration in "Select a camera" of the Setting panel.
② Adjust the exposure time in "Specify the exposure".

Setting		Initial value	Function explanation
[2]	Specify the exposure	8.000	Set the exposure time of the vision sensor.

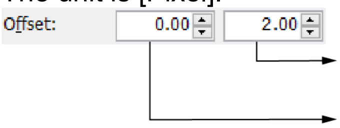
③ Place the workpiece within the field of view.
④ In the "Specify the exposure" section, click the [Stop] button, then the [Live] button.
Live images can be switched ON and OFF with the [Live]/[Stop] button.
⑤ Set shapes of the model area and search area in "Select the region", then click the [OK] button.
The search area will appear in a green frame, and the model area in a pink frame.



⑥ Move the green frame to enclose an area used to detect the model. Move the pink frame to enclose the model.

⑦ Click the [Registration] button.
If recognition is successful, a green crosshair will appear at the recognized point.
The recognized vision coordinates will appear in the upper left.

⑧ If recognition fails, the model may not have characteristics or the following parameters may not be set correctly.

Setting	Initial value	Function explanation
Threshold	70.00	Extracts images with the match percentage specified in this parameter or higher compared to the registered model. (Example) Setting the value to "70" will show recognition information if there is a 70% or higher match between the model displayed in the image and the registered model.
Angle Range	180.00	Specify up to how many degrees the registered model can rotate to be recognized.
Offset	0.00, 0.00	It is possible to offset the recognition point (position of the crosshair). The unit is [Pixel].  Offset amount in the vertical direction (The lower part of the screen is the positive direction.) Offset amount in the horizontal direction (The right side of the screen is the positive direction.)

After changing the values of "Adjust parameter", click the [Trigger] button to check the recognition results.

⑨ After registering the model, enter a vision program name in "Specify the model (vision program) name".

⑩ After configuring the settings, click the [Next] button.

9.3.2. When using PatMax

① Select a camera to be used for workpiece registration in "Select a camera" of the Setting panel.
② Adjust the exposure time in "Specify the exposure time".

Setting		Initial value	Function explanation
[2]	Specify the exposure	8.000	Set the exposure time of the vision sensor.

③ Place the workpiece within the field of view.
④ In "Specify the exposure time", click the [Stop] button, then the [Live] button.
⑤ Click the [Register] button.
The "Model Registration Editor" will start. Register the workpiece.
For information on how to use the model registration editor, refer to "8.3.3. How to use the model registration editor".
While the model registration editor is running, any control of the 2D vision sensor enhancement function (except the operation panel) cannot be used.

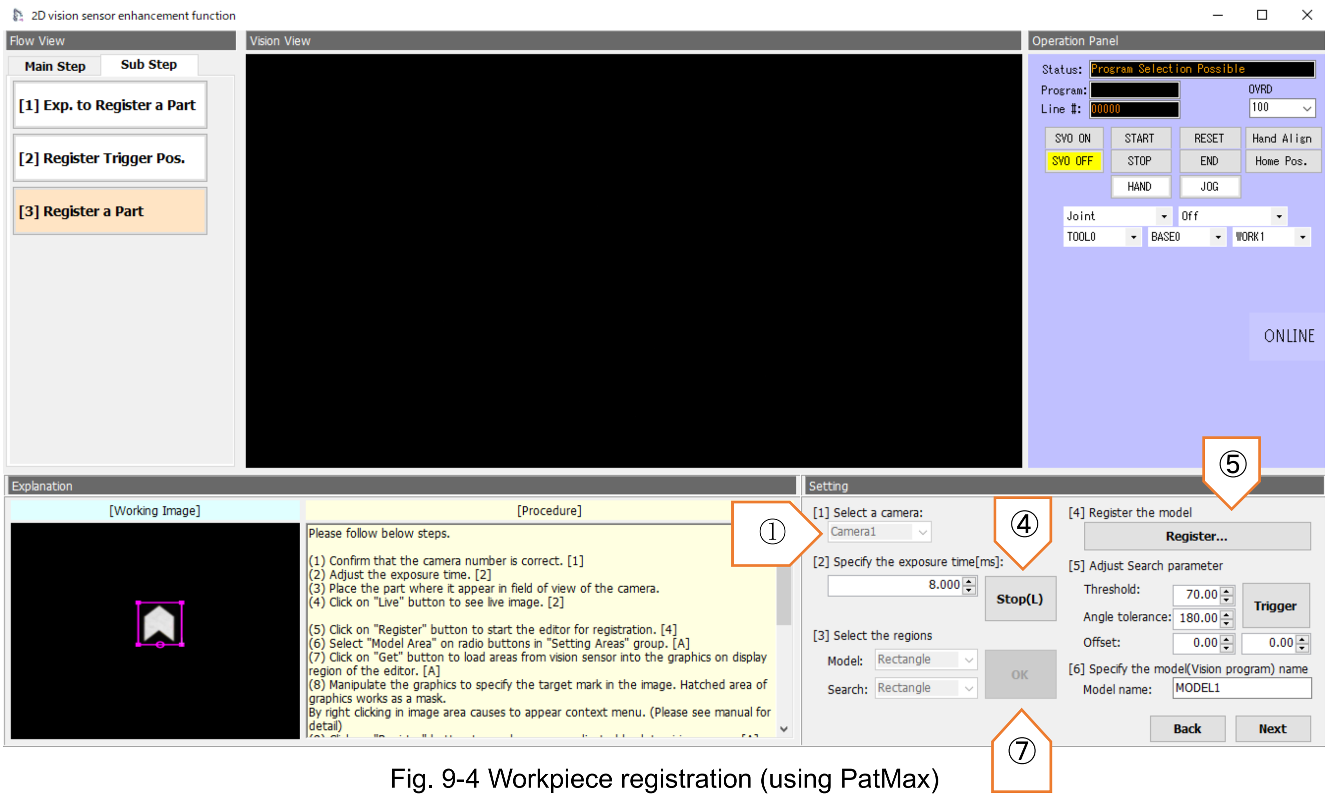
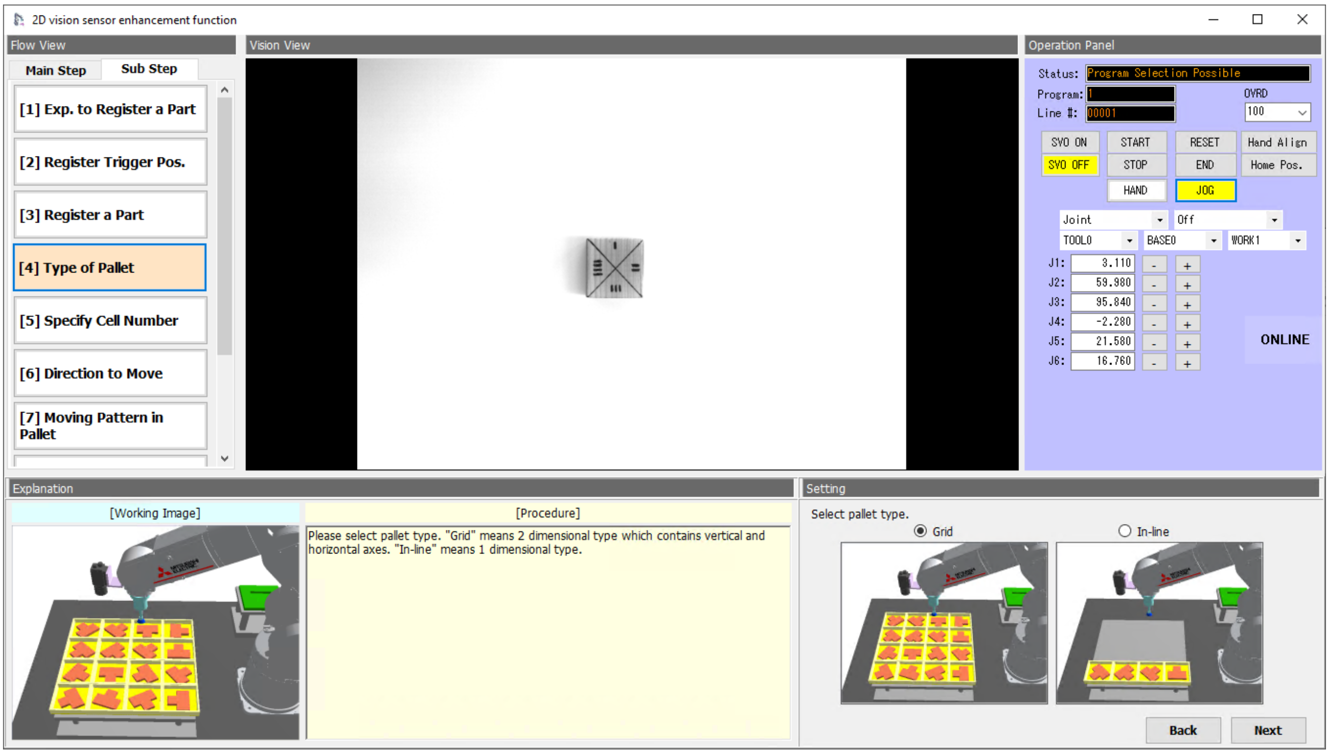


Fig. 9-4 Workpiece registration (using PatMax)

⑥ After the registration is completed in the model registration editor, and the "2D vision sensor enhancement function" screen appears again, the [Trigger] button will be enabled.
If the [Trigger] button is still disabled, restart the model registration editor and try the registration again.
⑦ After registering the model, enter a vision program name in "Specify the model(Vision program) name".
⑧ After configuring the settings, click the [Next] button.

9.4. Pallet settings

If an application that requires a pallet is selected, the pallet settings will appear as required steps. Set the pallet settings.

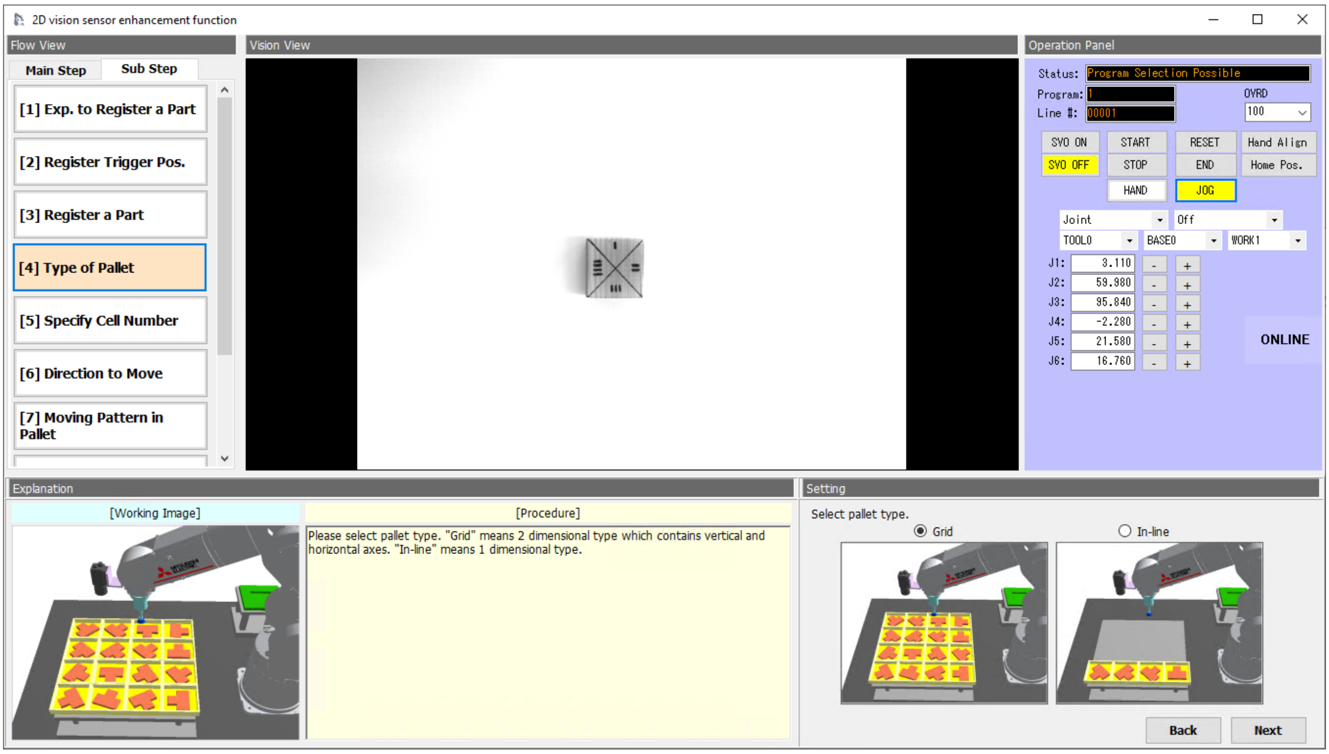


The sub steps of the pallet settings are composed of the following settings. Clicking a sub step will display the relevant window in the Setting panel.

- Type of Pallet
- Specify Cell Number
- Direction to Move
- Moving Pattern in Pallet
- Start position (S)
- End Position (A)
- End Position (B)

9.4.1. Pallet type

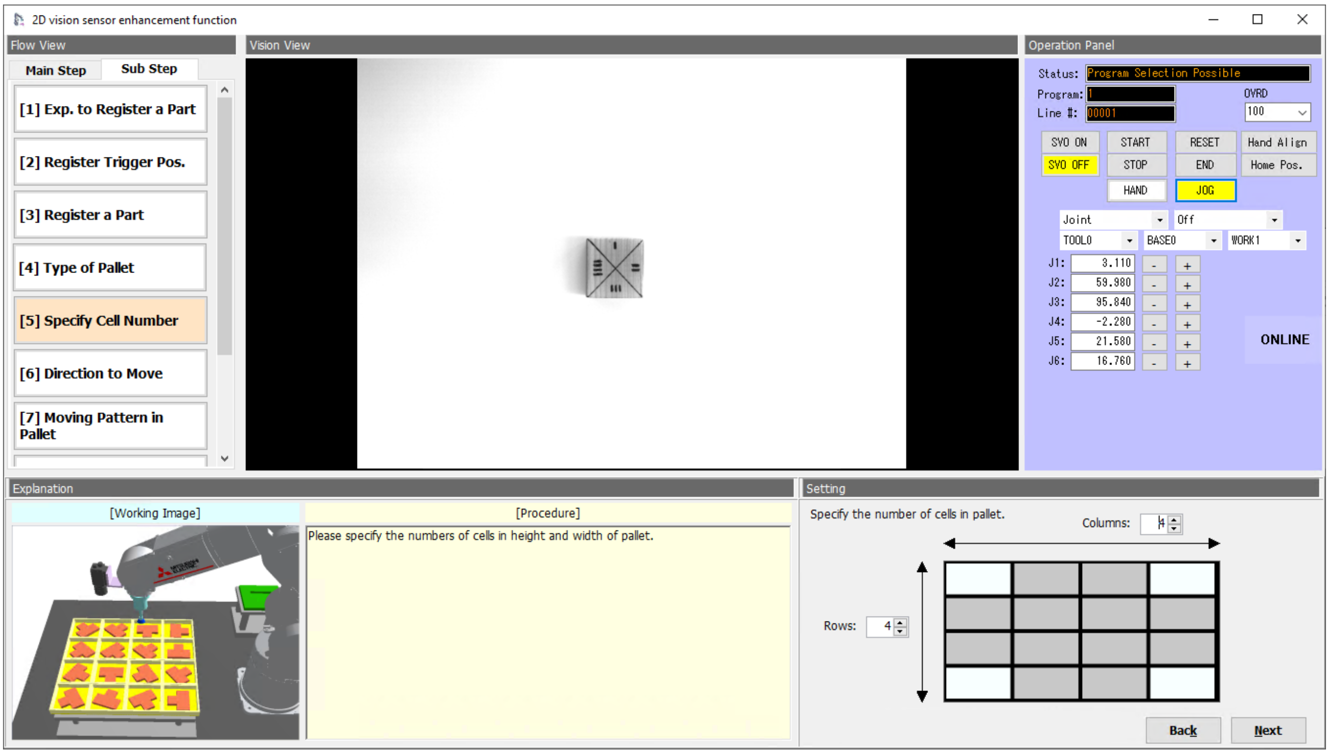
Specify whether to use a pallet in a grid pattern or single column.



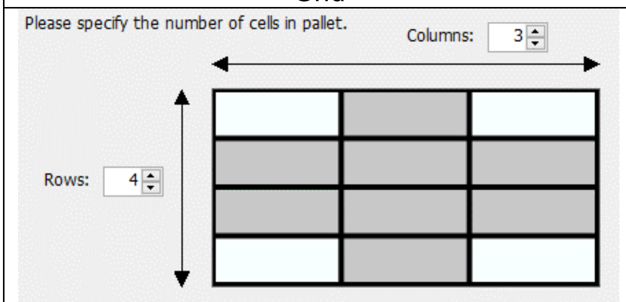
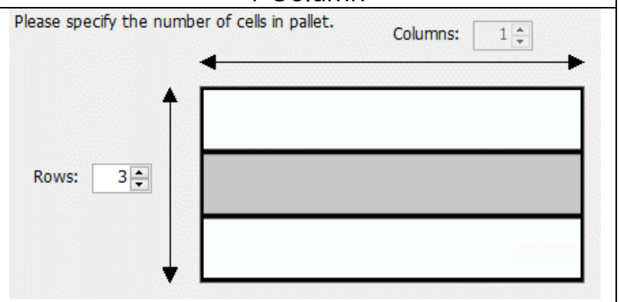
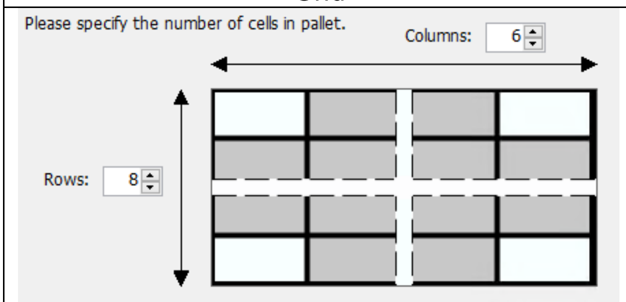
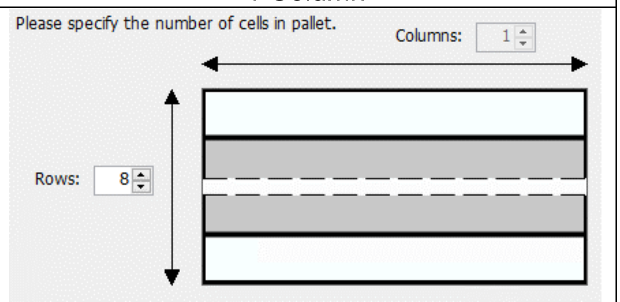
Specify "Grid" or "1 Column" in "Select pallet type" in the Setting panel, then click the [Next] button.

9.4.2. Number of pallet cells

Specify the numbers of cells to define the pallet.



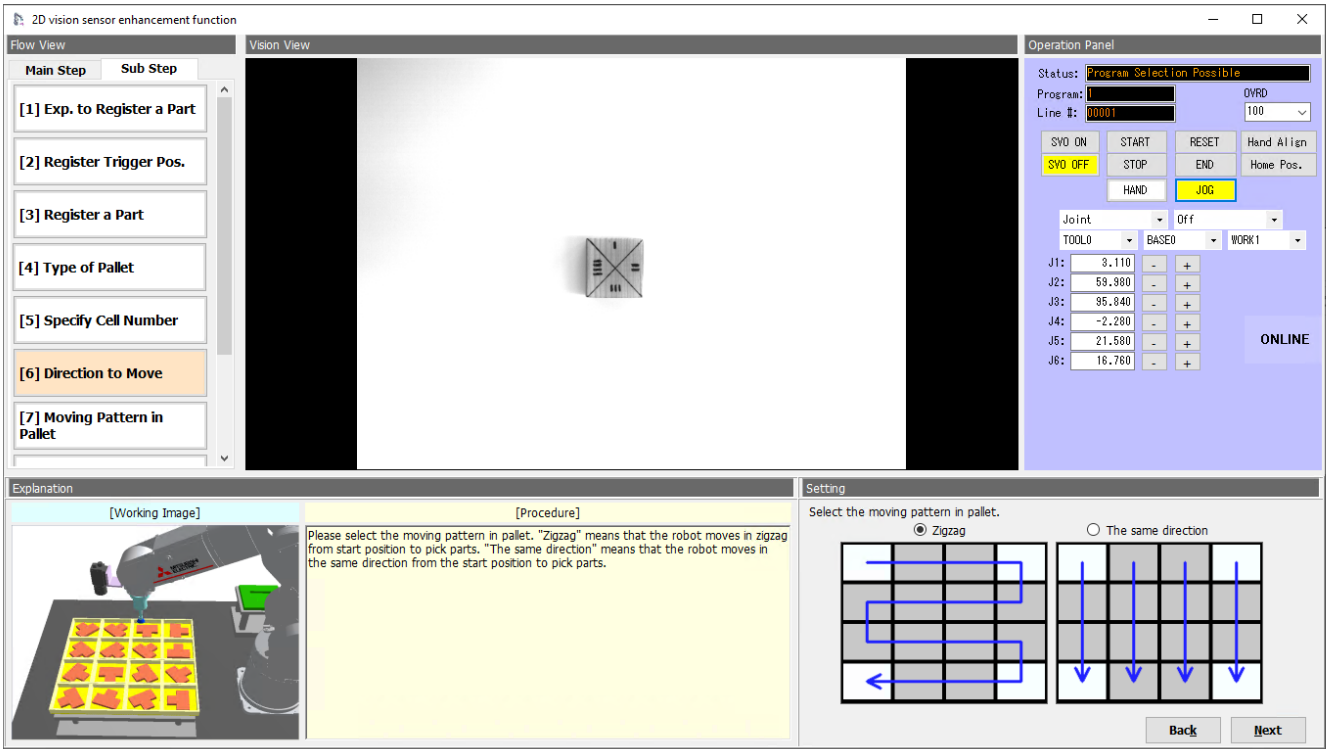
Use "Please specify the number of cells in pallet." in the Setting panel to specify the number of cells in rows and columns.
An image will be displayed in the Setting panel according to the specified number of cells as shown below.

<table>
<tr><td colspan="2">If the number of cells is 4 or less, an image will be displayed according to the number.</td></tr>
<tr><td>Grid</td><td>1 Column</td></tr>
<tr><td></td><td></td></tr>
<tr><td colspan="2">If the number of cells is 5 or more, an image will be displayed with cells in the middle area omitted.</td></tr>
<tr><td>Grid</td><td>1 Column</td></tr>
<tr><td></td><td></td></tr>
</table>

Specify the number of cells, then click the [Next] button.

9.4.3. Pallet operation pattern

If "Grid" is selected in the Setting panel (shown in 9.4.1 Pallet type), select an operation pattern for the pallet (Zigzag or the same direction).
If "1 Column" is selected in "9.4.1 Pallet type", this setting screen will not appear.



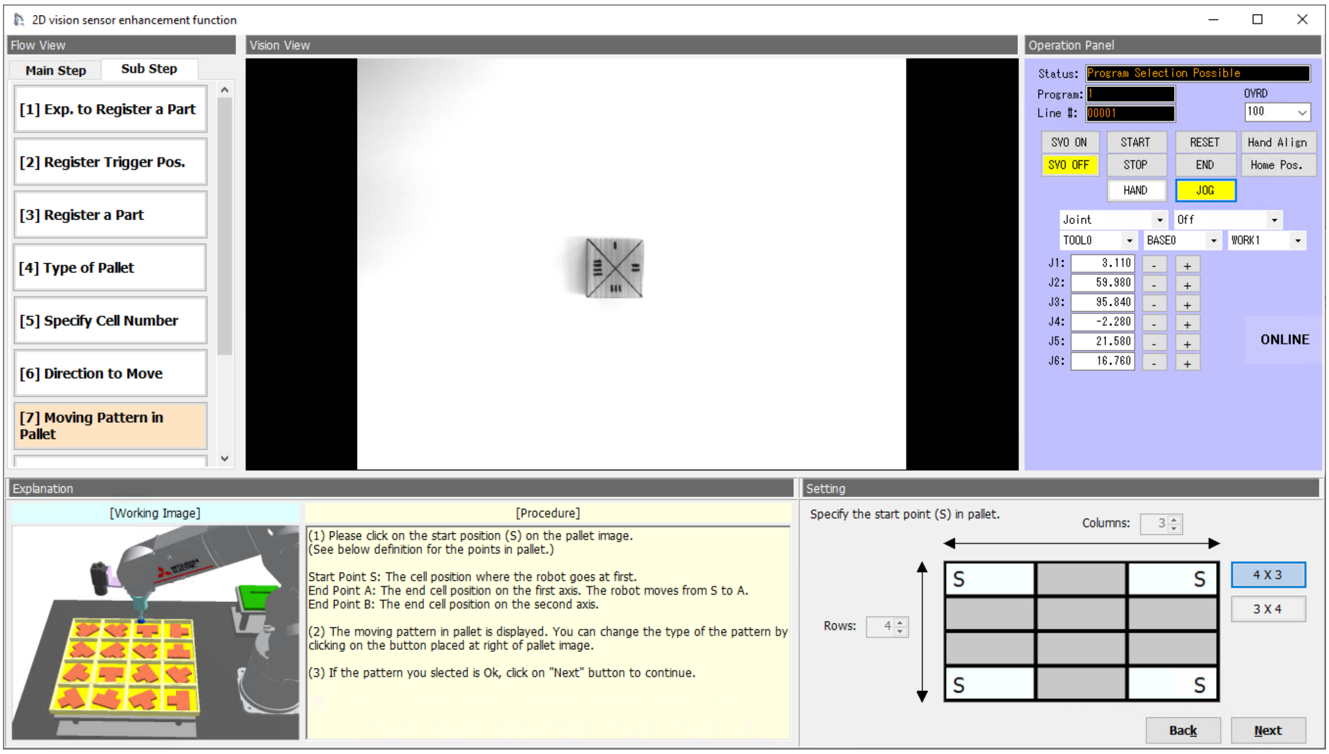
Selecting "Zigzag" will make the robot move in a zigzag manner.
Selecting "The same direction" will make the robot move in the same direction even when it moves to a different column.
Subsequent screens will show images according to the direction specified in this step.

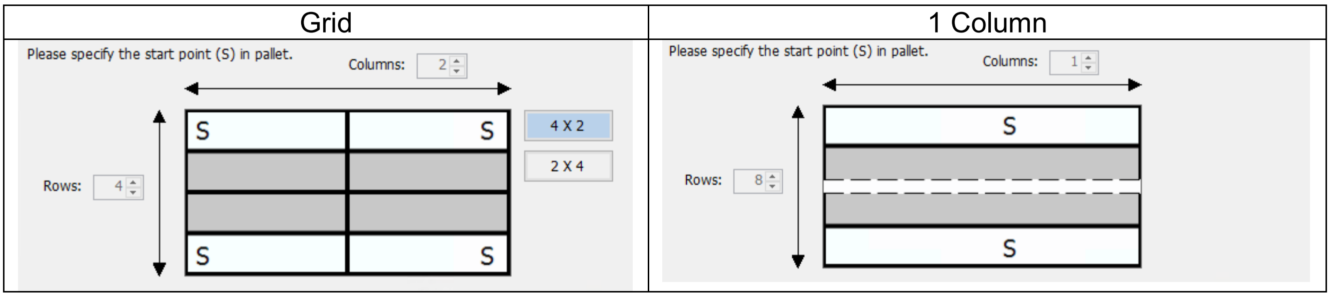
Specify a direction, then click the [Next] button.

9.4.4. Specifying a route for the pallet

Specify which route the robot takes for the pallet.

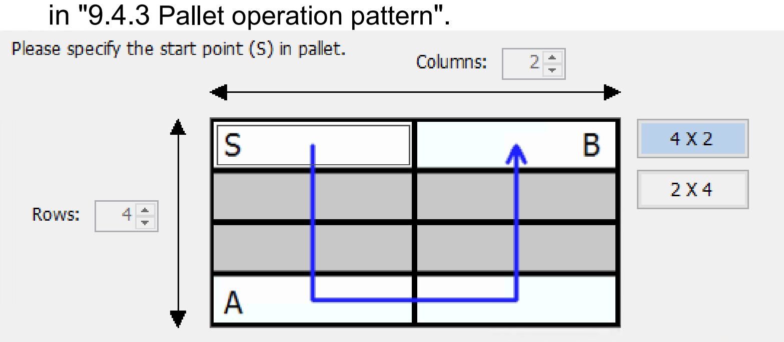
"S" representing a start point for pallet operations appears in the following positions of the Setting panel depending on the settings in "9.4.1 Pallet type": If "Grid" is selected, "S" will appear in the four corners. If "1 Column" is selected, "S" will appear on both sides.



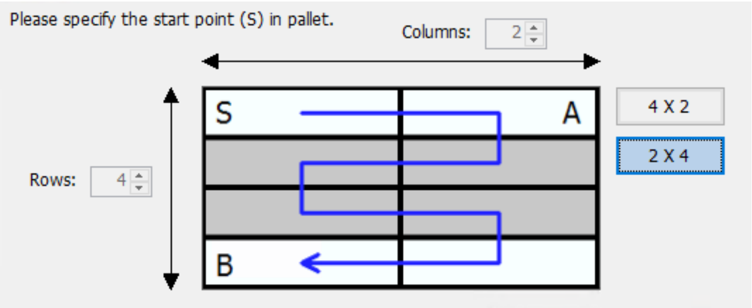


① Select a cell for the start point from the displayed start points "S", and click the cell.

② The direction of the pallet operation will be displayed.
For example, clicking the top left cell will change the image as shown below as "Zigzag" was specified in "9.4.3 Pallet operation pattern".



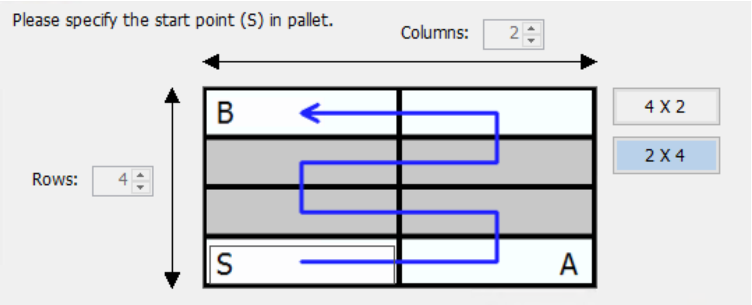
To change the robot moving direction from longitudinal to lateral, click the [4x2] or [2x4] button shown on the right of the image. The change will be applied to the image.



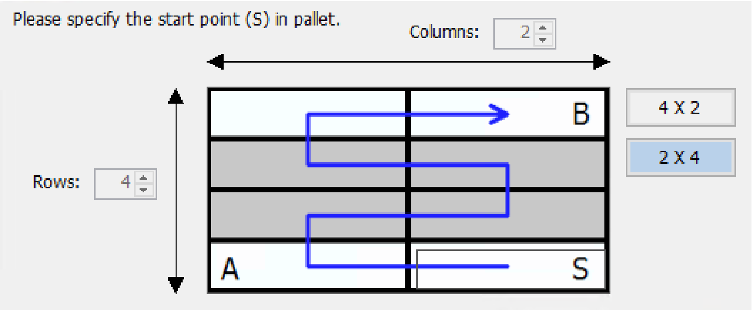
"A" and "B" shown in the image represent "End point A" and "End point B" of the robot moving direction. These end points are required to define the pallet. "End point A" and "End point B" refer to "End point A" and "End point B" of "Def Plt" in the MELFA-BASIC Ⅵ robot program language.

To change the position of the start point, click "S", "A", "B", or the remaining cell in the corners. The direction in which the robot moves can be changed.

For example, clicking "B" in the image above will change the start point as shown below.



Clicking "A" in the image above will change the start point as shown below.

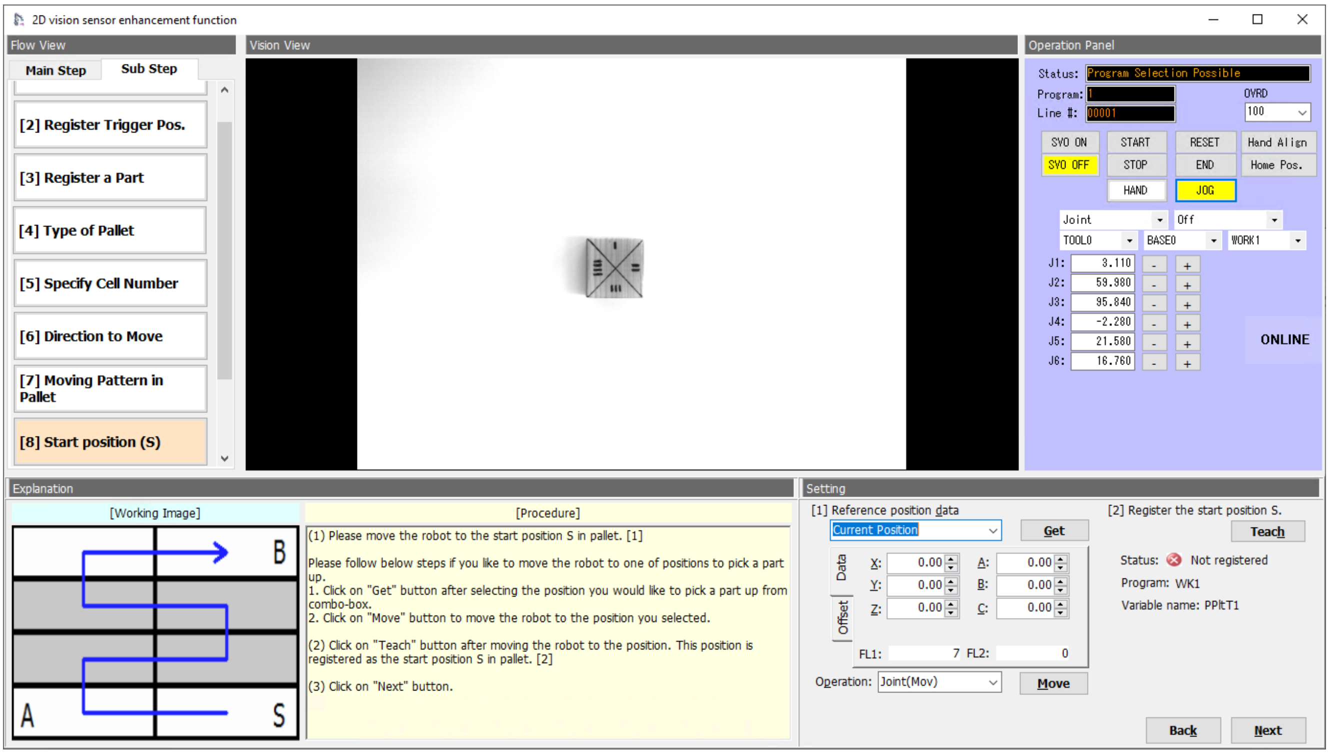


The image showing the moving direction specified on this screen will appear in the Working Image panel of subsequent screens.

③ After specifying the robot movement for the pallet, click the [Next] button.

9.4.5. Registering Start point S for the pallet

Teach Start point "S" on the route for the pallet.



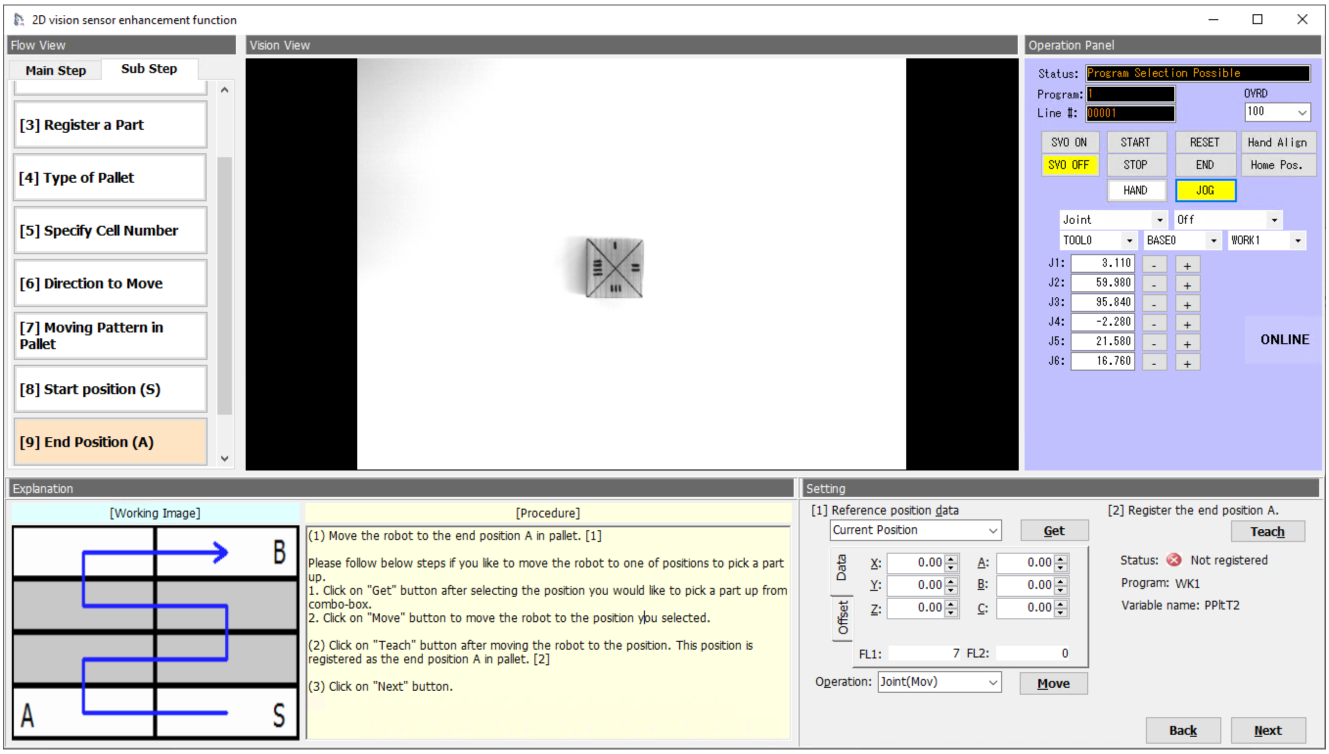
① Move the robot to the start point on the route for the pallet using the Operation Panel or teaching pendant.

② Click the [Teach] button in the Setting panel.
"Reference position data" can be used not only to select "Current Position" but also position data taught before this screen appears.
After the teaching process, click the [Next] button. The method of teaching the robot is the same as that described in "9.2 Registering a trigger position".

9.4.6. Registering End point A for the pallet

Teach End point "A" on the route for the pallet. Regarding End point "A", the robot moves differently depending on whether "Grid" or "1 Column" is selected in the Setting panel (shown in 9.4.1 Pallet type).

・ When selecting "Grid"
 A temporary end point to determine the moving direction from start point "S"
・ When selecting "1 Column"
 The final destination of the route



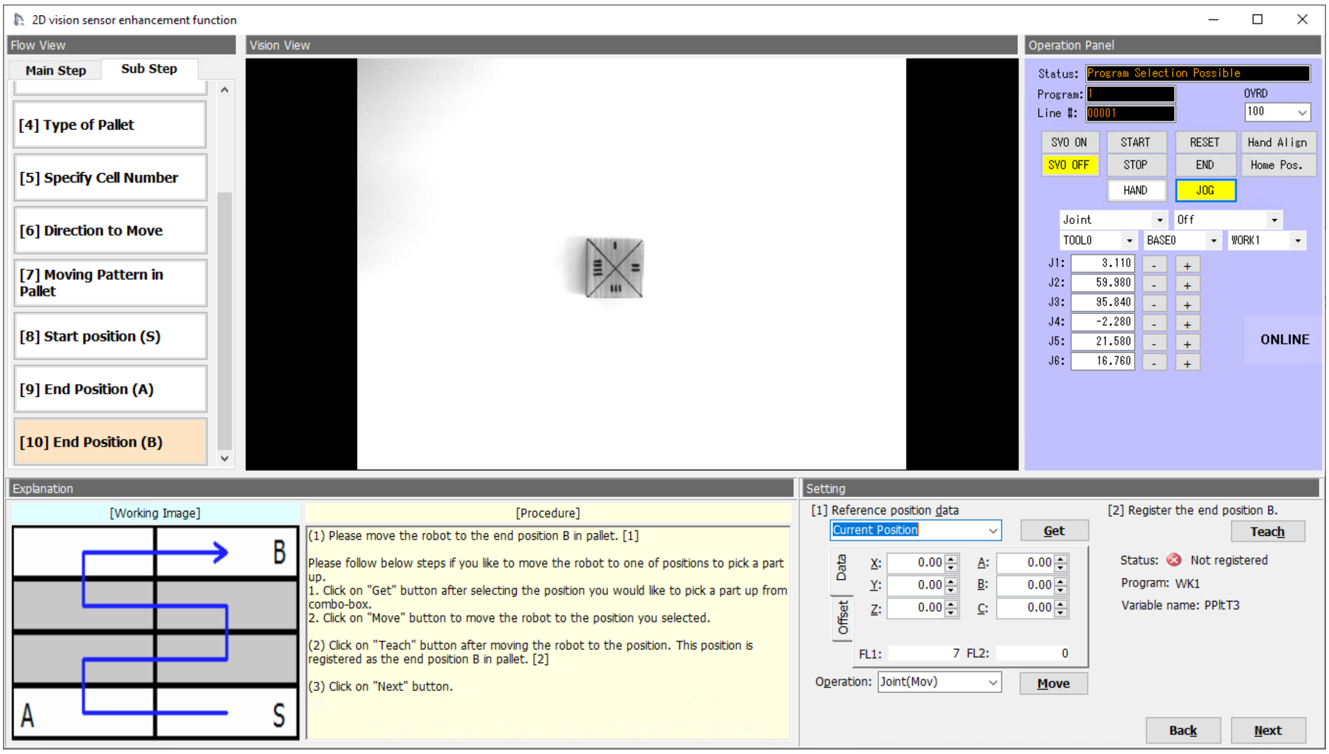
The method of configuring the settings displayed in the Setting panel is the same as that described in "9.2Registering a trigger position".

① Move the robot to End point A on the route for the pallet.
② Click the [Teach] button
 "Reference position data" can be used not only to select "Current Position" but also position data taught before this screen appears.
 After the teaching process, click the [Next] button. The method of teaching the robot is the same as that described in "9.2 Registering a trigger position".

9.4.7. Registering End point B for the pallet

If "Grid" is selected in the Setting panel (9.4.1 Pallet type), this screen will appear. Teach End point "B" on the route for the pallet.

If "1 Column" is selected, this screen will not appear. If "1 Column" is selected, End point B will be automatically set at the same point as End point A.



The method of configuring the settings displayed in the Setting panel is the same as that described in "9.2 Registering a trigger position".

① Move the robot to End point B on the route for the pallet.

② Click the [Teach] button.

"Reference position data" can be used not only to select "Current Position" but also position data taught before this screen appears.

After the teaching process, click the [Next] button. The method of teaching the robot is the same as that described in "9.2 Registering a trigger position".

10. OPERATION SETTINGS

The operation settings define the operation of the robot. Steps differ depending on the application, but basically follow the steps shown below.

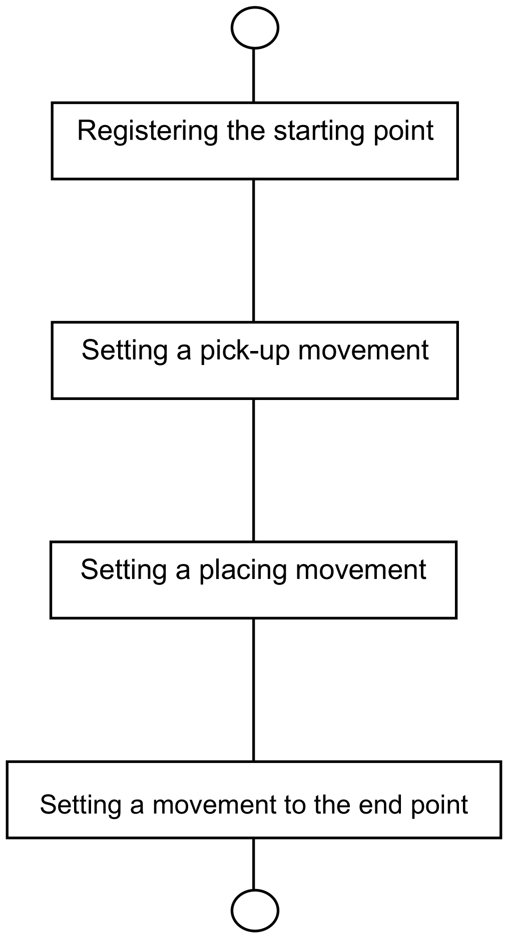


Fig. 10-1 Operation setting procedure

10.1. Teaching the starting point

Register the position to which the robot moves first when a specified application is executed.
Follow the instructions shown in the Working Image panel and Procedure panel.

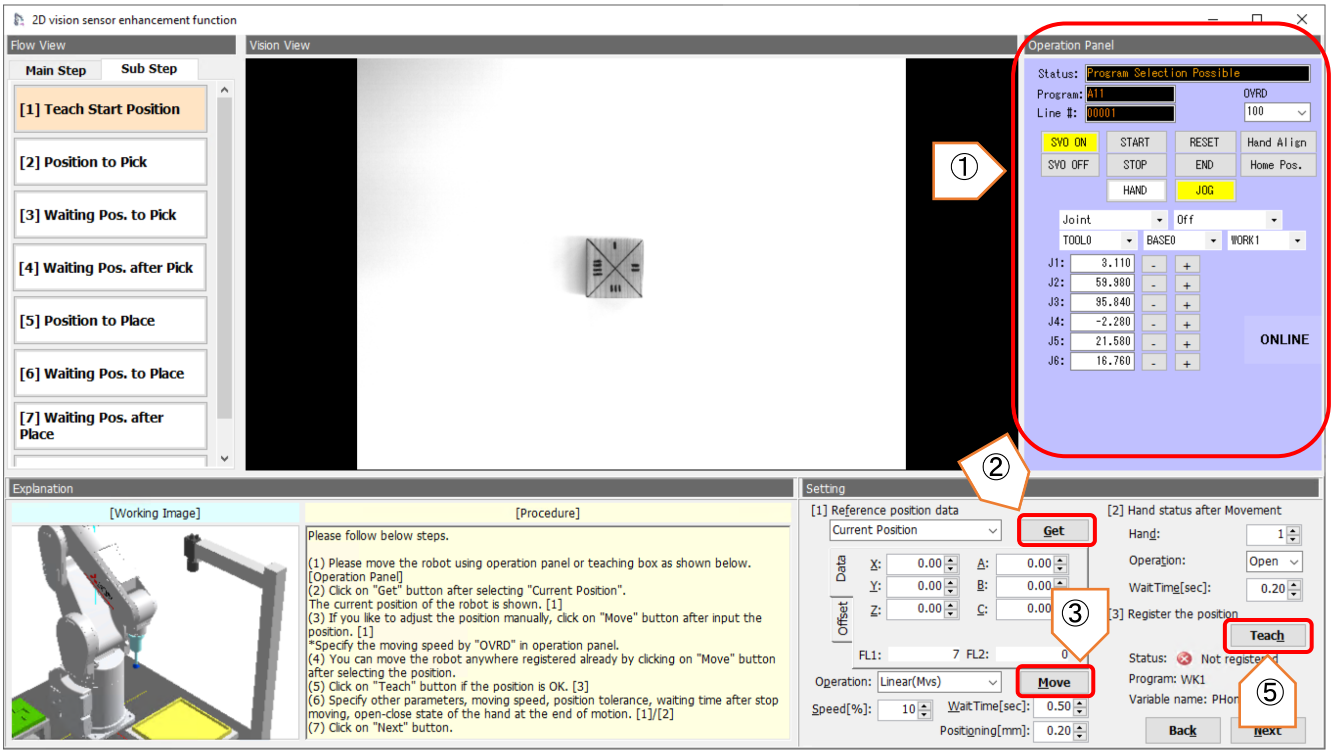


Fig. 10-2 Teaching the starting point

① Move the robot to your desired <starting point> using the Operation Panel or teaching pendant.
For further information on how to operate the Operation Panel, refer to "19.2 How to operate the Operation Panel".

② Select "Current Position" in "Reference position data" of the Setting panel, then click the [Get] button.

③ If required, adjust X, Y, Z, A, B, and C, then click the [Move] button.
The robot can move to the position displayed in the [Data] tab. Clicking the [Offset] tab can move the robot in the tool direction from the current position by a specified offset value.

④ To move the robot to a registered position, specify a position other than "Current Position", and click the [Move] button.
For details on the position jump function, refer to "19.3 Position jump".

⑤ After determining the position, click the [Teach] button to register the <starting point> as "PHome" of the WK* program.
Upon completion of registration, the status indicator turns green (indicating "registered"). You can proceed to the next step.

⑥ Settings related to the movement to the <starting point> such as speed, positioning accuracy, and waiting time can be set. Fine adjustments are not required in this process because you have an opportunity to make fine adjustments while checking a series of operations after all operation settings have been configured. Configure the settings as instructed in "12.2 Modifying an operation".

⑦ After configuring the settings, click the [Next] button.

10.2. Teaching a pick-up point

In the process of teaching a pick-up point, register the position to which the robot moves to grasp (suck) the workpiece.
Follow the instructions shown in the Working Image panel and Procedure panel.

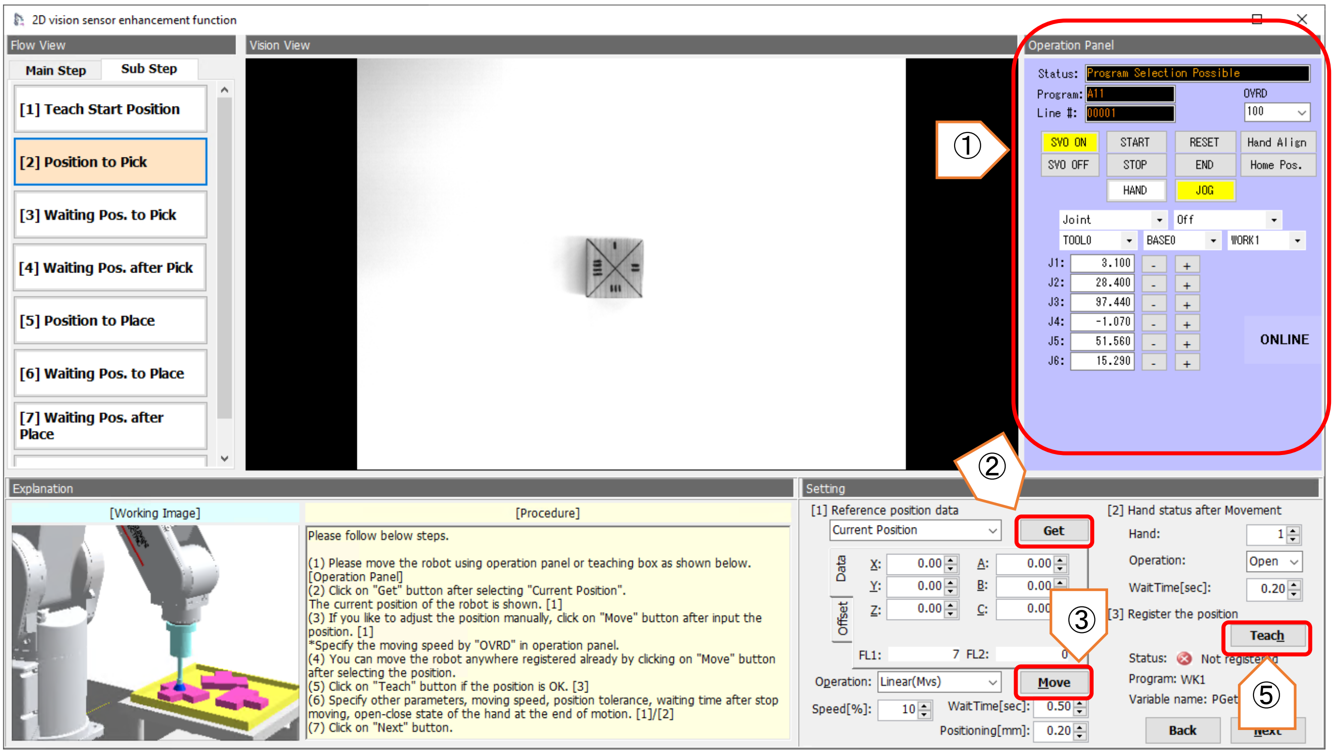


Fig. 10-3 Teaching a pick-up point

① Move the robot to the <pick-up point> using the Operation Panel or teaching pendant.
For further information on how to operate the Operation Panel, refer to "19.2 How to operate the Operation Panel".

② Select "Current Position" in "Reference position data" of the Setting panel, then click the [Get] button.

③ If required, adjust X, Y, Z, A, B, and C, then click the [Move] button.
The robot can move to the position displayed in the [Data] tab. Clicking the [Offset] tab can move the robot in the tool direction from the current position by a specified offset value.

④ To move the robot to a registered position, specify a position other than "Current Position", and click the [Move] button.
For details on the position jump function, refer to "19.3 Position jump".

⑤ After determining the position, click the [Teach] button to register the <pick-up point> as "PGet" of the WK* program.
(Upon completion of registration, the status indicator turns green (indicating "registered"). You can proceed to the next step.)

⑥ Settings related to the movement to the <pick-up point> can be configured. Configure the settings as instructed in "12.2 Modifying an operation".

⑦ After configuring the settings, click the [Next] button.

10.3. Teaching a position before pick-up operation

Teach a position before pick-up operation. Teach a via point before the robot picks up the workpiece. Follow the instructions shown in the Working Image panel and Procedure panel.

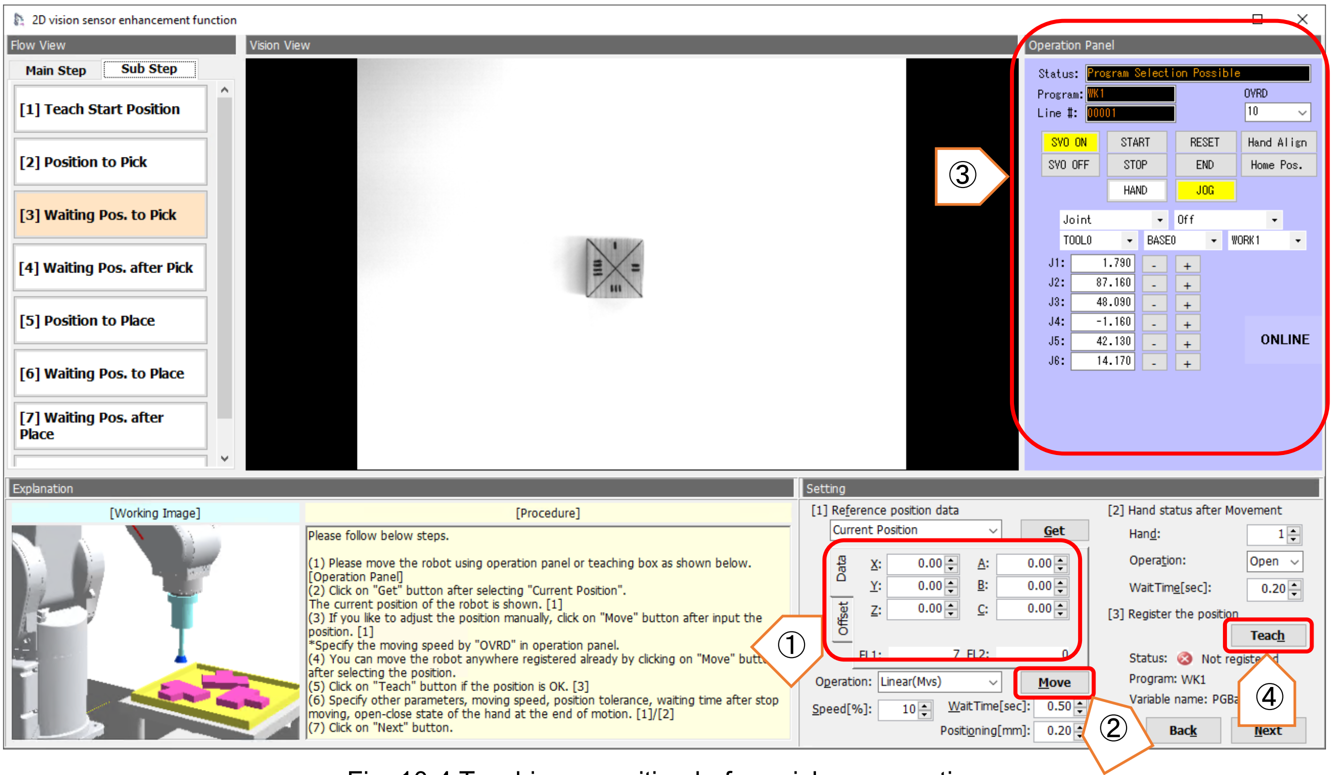


Fig. 10-4 Teaching a position before pick-up operation

① Click the [Offset] tab in "Reference position data" of the Setting panel, then specify the amount that the robot moves toward the front from the current position (where the robot grasps (sucks) the workpiece).

⚠ CAUTION

Note that the travel amount indicates "Tool direction". Entering a minus value in the Z axis of an RV robot will draw the arm.

② Specify the amount that the robot moves toward the front from the current position, then click the [Move] button.
③ If required, fine adjust the position using the Operation Panel.
④ After determining the position, click the [Teach] button to register the <position before pick-up operation> as "PGBack" of the WK* program. (Upon completion of registration, the status indicator turns green (indicating "registered"). You can proceed to the next step.)
⑤ Settings related to the movement to the <position before pick-up operation> can be configured. Configure the settings as instructed in "12.2 Modifying an operation".
⑥ After configuring the settings, click the [Next] button.

10.4. Teaching a position after pick-up operation

Teach a position after pick-up operation. Teach a via point after the robot grasps (sucks) the workpiece. Follow the instructions shown in the Working Image panel and Procedure panel.

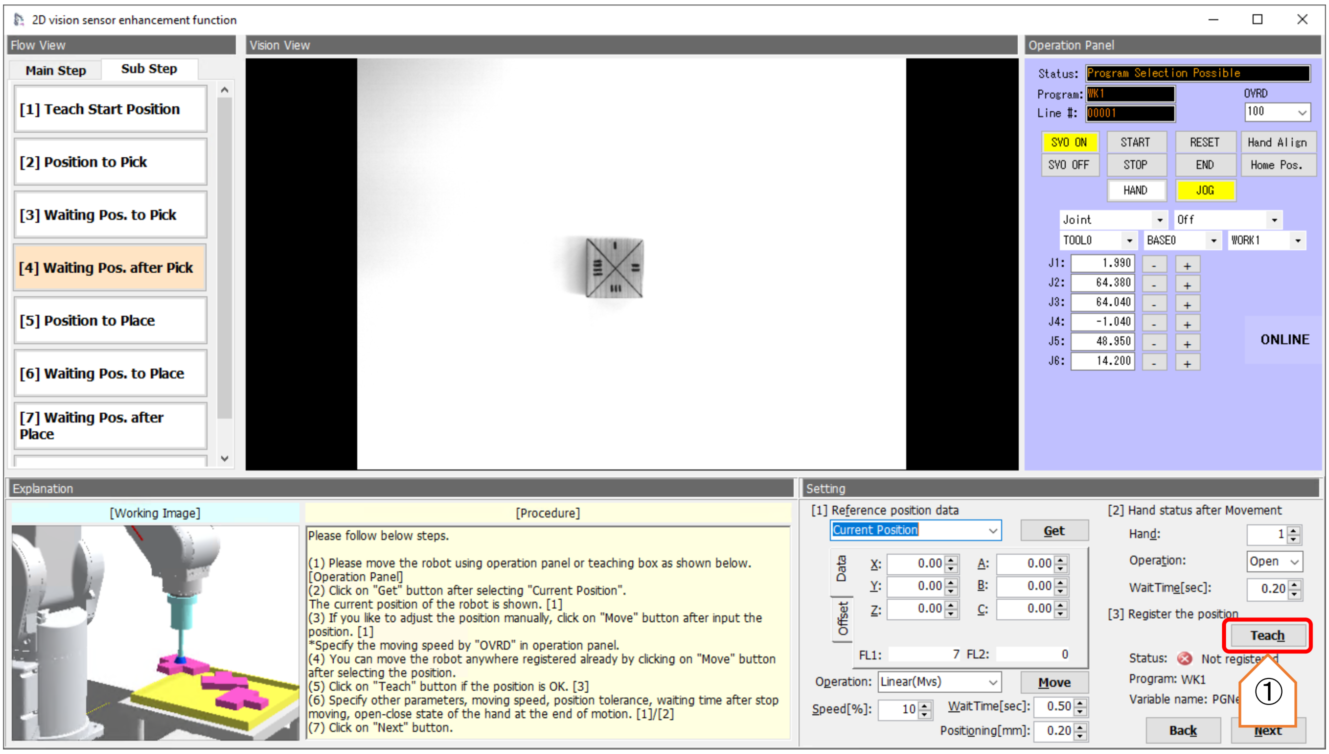


Fig. 10-5 Teaching a position after pick-up operation

Since this step is just after the <position before pick-up operation> is taught, it is recommended to teach the same position as the <position before pick-up operation>.

① Click the [Teach] button in the Setting panel to register the <position after pick-up operation> as "PGNext" of the WK* program.
(Upon completion of registration, the status indicator turns green (indicating "registered"). You can proceed to the next step.)

② To set a position different from the <position before pick-up operation> such as for reducing takt time, move the offset using the [Offset] tab and [Move] button in the Setting panel, or move the robot using the Operation Panel, then click the [Teach] button.

③ Settings related to the movement to the <position after pick-up operation> can also be configured. Configure the settings as instructed in "12.2 Modifying an operation".

④ After configuring the settings, click the [Next] button.

10.5. Teaching a placement point

In the process of teaching a placement point, register the position to which the robot moves to place the workpiece.
Follow the instructions shown in the Working Image panel and Procedure panel.

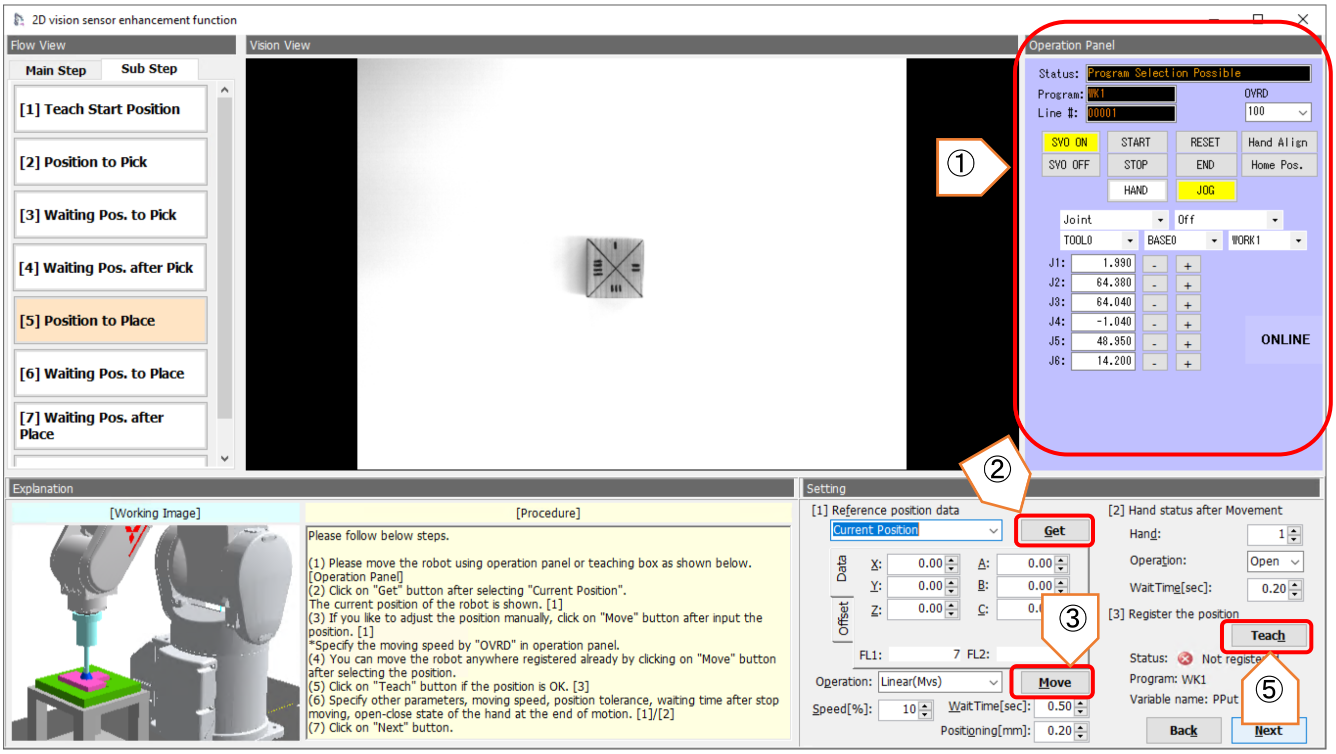


Fig. 10-6 Teaching a placement point

① Move the robot to the <placement point> using the Operation Panel or teaching pendant.
② Select "Current Position" in "Reference position data" of the Setting panel, then click the [Get] button.
③ If required, adjust X, Y, Z, A, B, and C, then click the [Move] button.
The robot can move to the position displayed in the [Data] tab. Clicking the [Offset] tab can move the robot in the tool direction from the current position by a specified offset value.
④ To move the robot to a registered position, specify a position other than "Current Position", and click the [Move] button (position jump function). For details on the position jump function, refer to "19.3 Position jump".
⑤ After determining the position, click the [Teach] button to register the <placement point> as "PPut" of the WK* program. (Upon completion of registration, the status indicator turns green (indicating "registered"). You can proceed to the next step.)
⑥ Settings related to the movement to the <placement point> can be configured. Configure the settings as instructed in "12.2 Modifying an operation".
⑦ After configuring the settings, click the [Next] button.

10.6. Teaching a position before placement operation

Teach a position before placement operation. Teach a via point before the robot places the workpiece. Follow the instructions shown in the Working Image panel and Procedure panel.

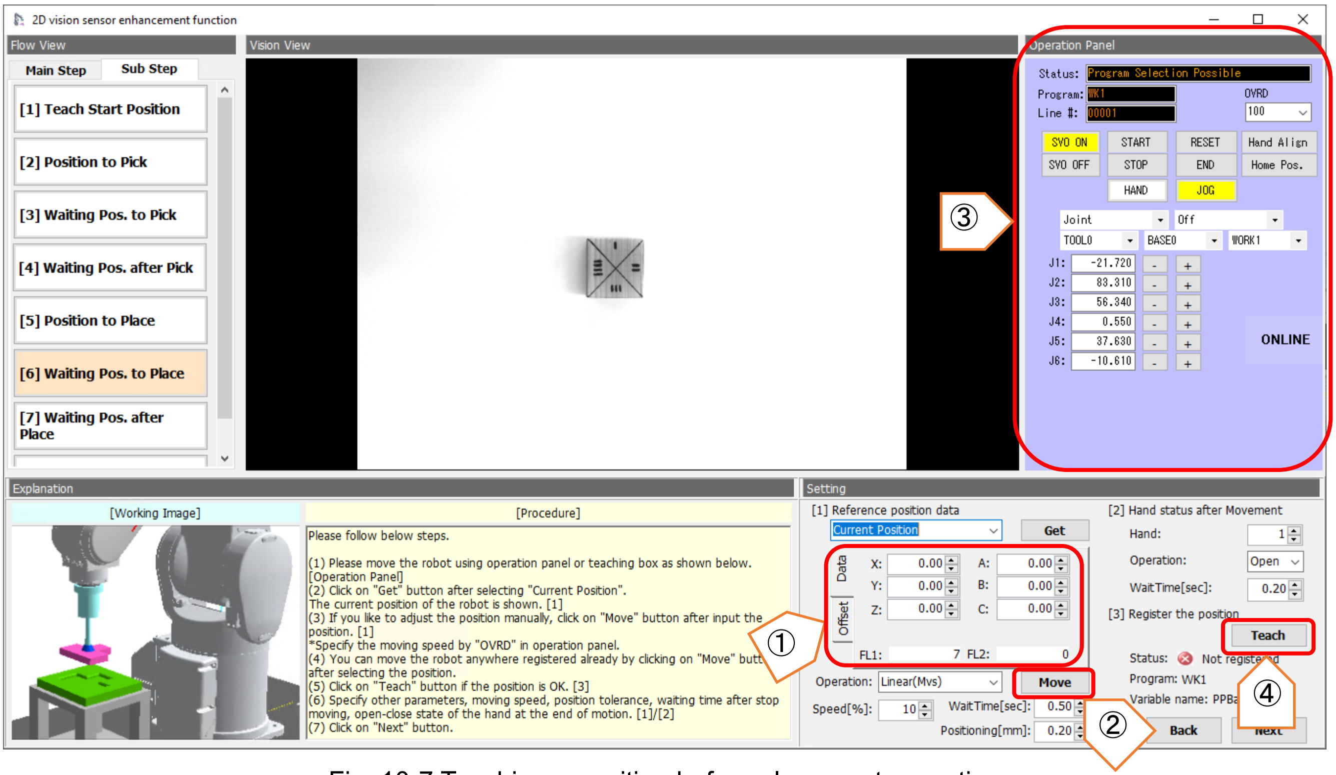


Fig. 10-7 Teaching a position before placement operation

① Click the [Offset] tab in "Reference position data" of the Setting panel, then specify the amount that the robot moves toward the front from the current position (where the robot places the workpiece).

> **⚠ CAUTION**
>
> Note that the travel amount indicates "Tool direction". Entering a minus value in the Z axis of an RV robot will draw the arm.

② Specify the amount that the robot moves toward the front from the current position, then click the [Move] button.
③ If required, fine adjust the position using the Operation Panel or teaching pendant.
④ After determining the position, click the [Teach] button to register the <position before placement operation> as "PPBack" of the WK* program. (Upon completion of registration, the status indicator turns green (indicating "registered"). You can proceed to the next step.)
⑤ Settings related to the movement to the <position before placement operation> can be configured. Configure the settings as instructed in "12.2 Modifying an operation".
⑥ After configuring the settings, click the [Next] button.

10.7. Teaching a position after placement operation

Teach a position after placement operation. Teach a via point after the robot places the workpiece. Follow the instructions shown in the Working Image panel and Procedure panel.

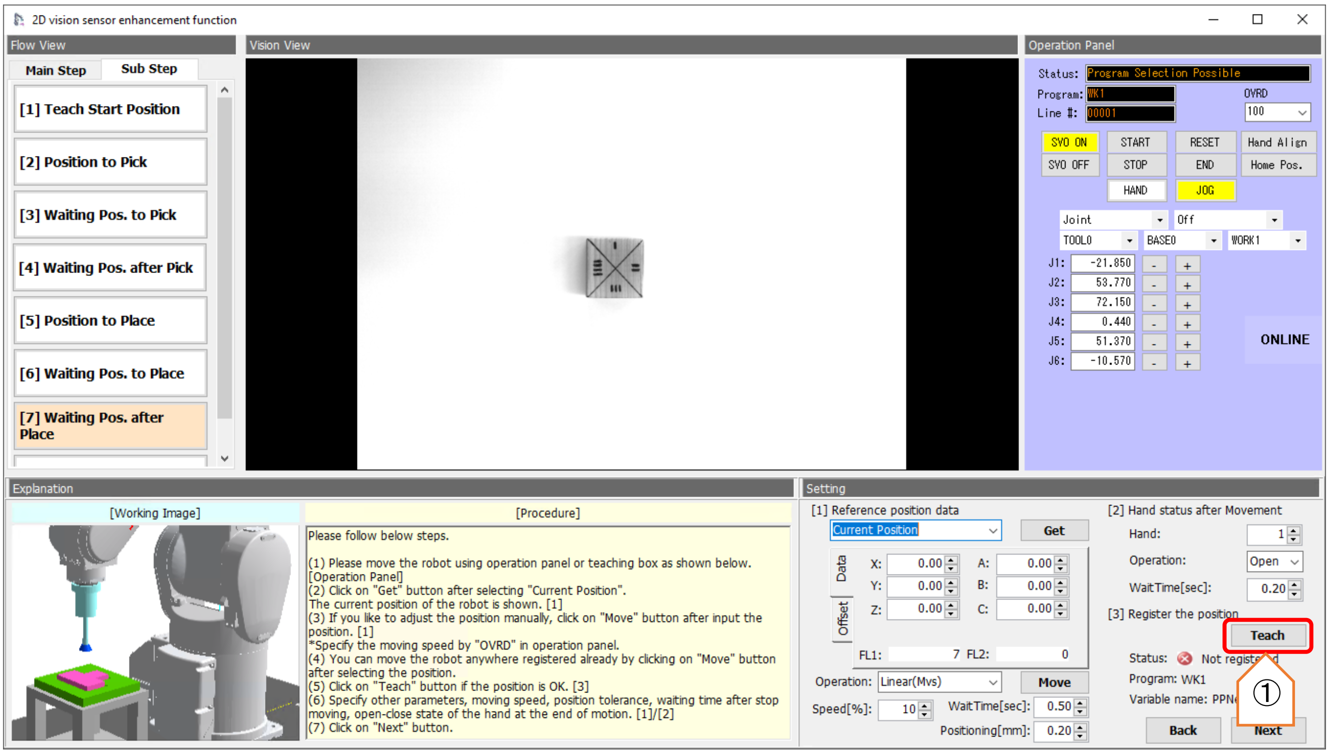


Fig. 10-8 Teaching a position after placement operation

Since this step is just after the <position before placement operation> is taught, it is recommended to teach the same position as the <position before placement operation>.

① Click the [Teach] button in the Setting panel to register the <position after placement operation> as "PPNext" of the WK* program. (Upon completion of registration, the status indicator turns green (indicating "registered"). You can proceed to the next step.)

② To set a position different from the <position before placement operation> such as for reducing takt time, move the offset using the [Offset] tab and [Move] button, or move the robot using the Operation Panel, then click the [Teach] button.

③ Settings related to the movement to the <position after placement operation> can be configured. Configure the settings as instructed in "12.2 Modifying an operation".

④ After configuring the settings, click the [Next] button.

10.8. Movement to the end point

This screen is used to specify the speed during movement to the final operation point after all operations are complete and whether the hand opens or closes after movement.
Set the final operation point in the same position as the <starting point> registered in "10.1 Teaching the starting point".
Follow the instructions shown in the Working Image panel and Procedure panel.

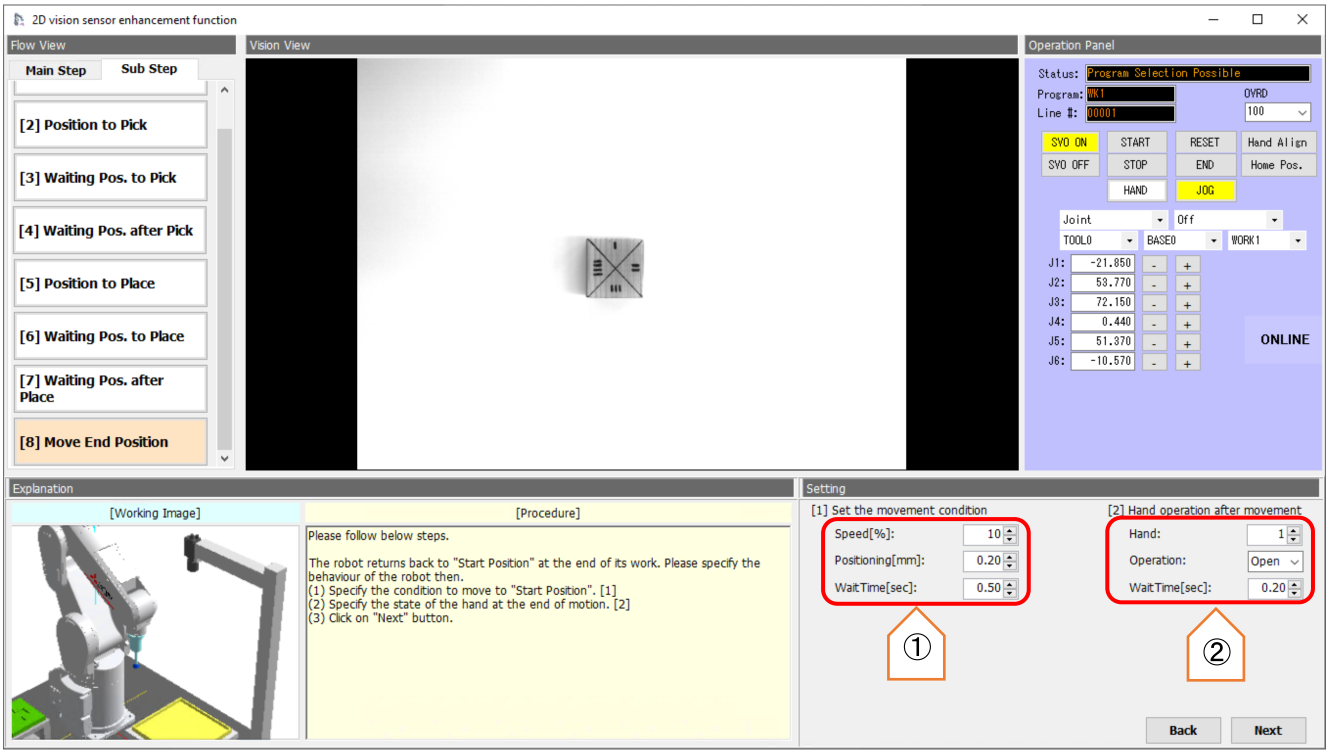


Fig. 10-9 Movement to the end point

① In "Set the movement condition" of the Setting panel, set "Speed", "Positioning", and "Wait Time" when the robot moves to the <final operation point> (<starting point>).

② In "Hand operation after movement", set "Operation" and "Wait Time" for the robot hand at the <final operation point> (<starting point>).
Settings related to the movement to the <end point> can be configured. Configure the settings as instructed in "12.2 Modifying an operation".

③ After configuring the settings, click the [Next] button.

11. REGISTERING A GRASP POSITION

Set a position to grasp a recognized workpiece.

11.1. Executing a registration program

Follow the instructions shown in the Working Image panel and Procedure panel.

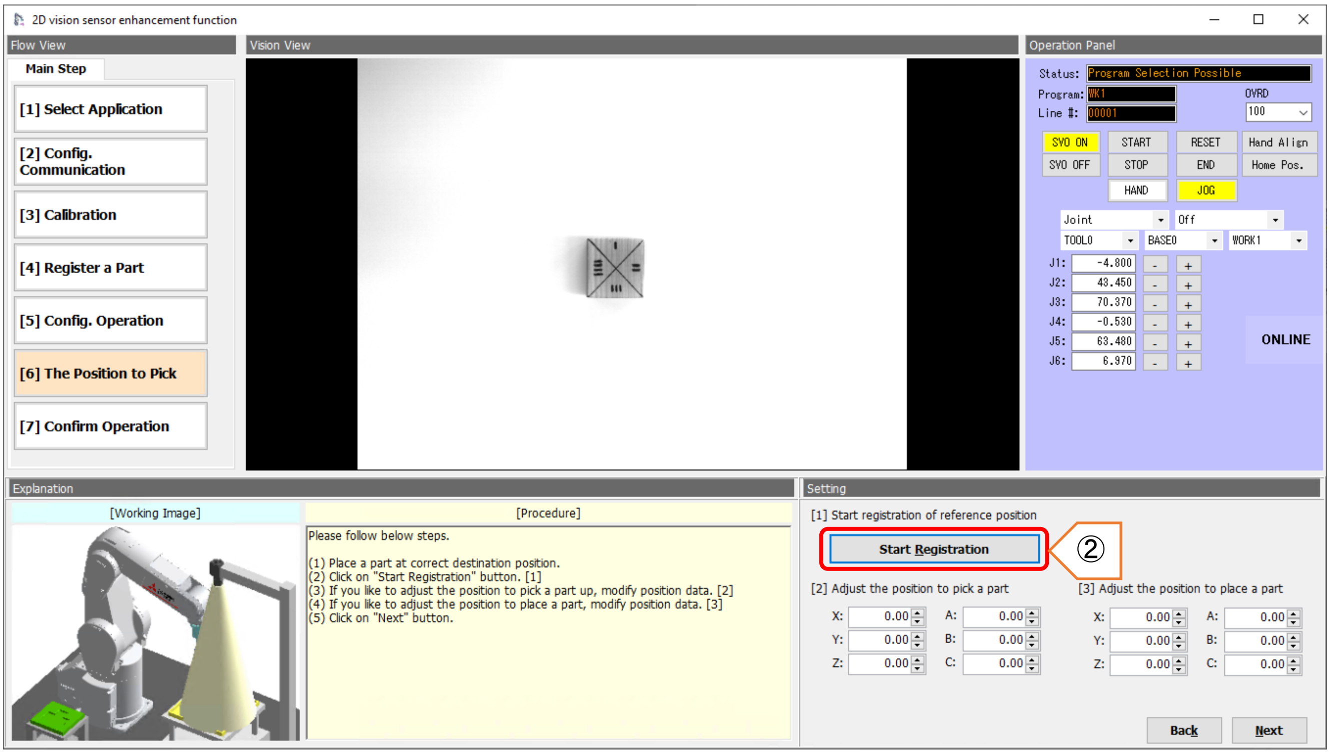


Fig. 11-1 Registration program execution screen

① Place the workpiece at the point intended for actual placement.

② Clicking the [Start Registration] button in the Setting panel will start tracing the transportation movement backward to automatically find the grasp position (the position to grasp the workpiece recognized by the vision sensor) automatically. Note that the robot will operate at low speed. Be careful of interference with the peripheral equipment.

<When making fine adjustments of the grasp position and the placement point>
To make fine adjustments of "The Position to Pick" and "The Position to Place" during the operation check described in "12 CHECKING THE OPERATION", set the offset for the grasp position in "[2] Adjust the position to pick a part" and "[3] Adjust the position to place a part". The offset value is set in the tool coordinate system.

③ After configuring the settings, click the [Next] button.

12. CHECKING THE OPERATION

In this step, you can check the settings made up to now by moving the robot.
Look at the illustration or video shown in the Working Image panel and follow the instructions in the Procedure panel.

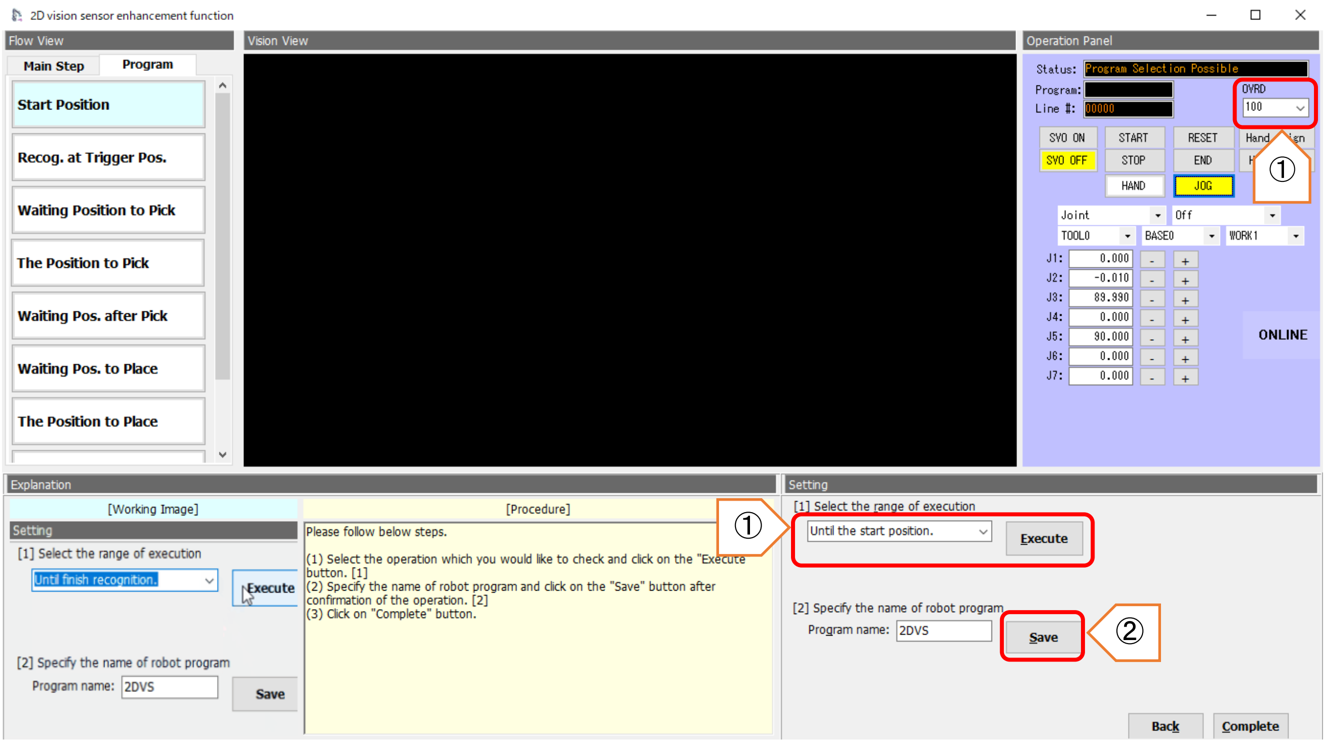


Fig. 12-1 Checking the operation

12.1. Checking the created operation program

You can check a series of settings in the robot program consecutively, and component elements of the operation flow partially. Use the following steps:

① Specify an execution part of the robot program in "Select the range of execution" of the Setting panel. Specify the operating speed using the Operation Panel, then click the [Execution] button. You can check the specified part of operation.
(If an error occurs while the robot is moving, refer to "17 TROUBLESHOOTING".)

② After checking the operation, specify a robot program name, then click the [Save] button.

③ Click the [Finish] button to complete the process.

⚠ CAUTION

The robot may collide with the peripheral equipment. Reduce the speed (Ovrd) sufficiently and ensure that you can press the emergency stop button anytime before checking operations that have never been checked.

12.2. Modifying an operation

If a checked operation needs to be fine-adjusted, go back to the relevant step and change the settings. Select a step you want to make adjustments from the Main Step tab or Program tab, then click the step. The relevant setting screen will appear.

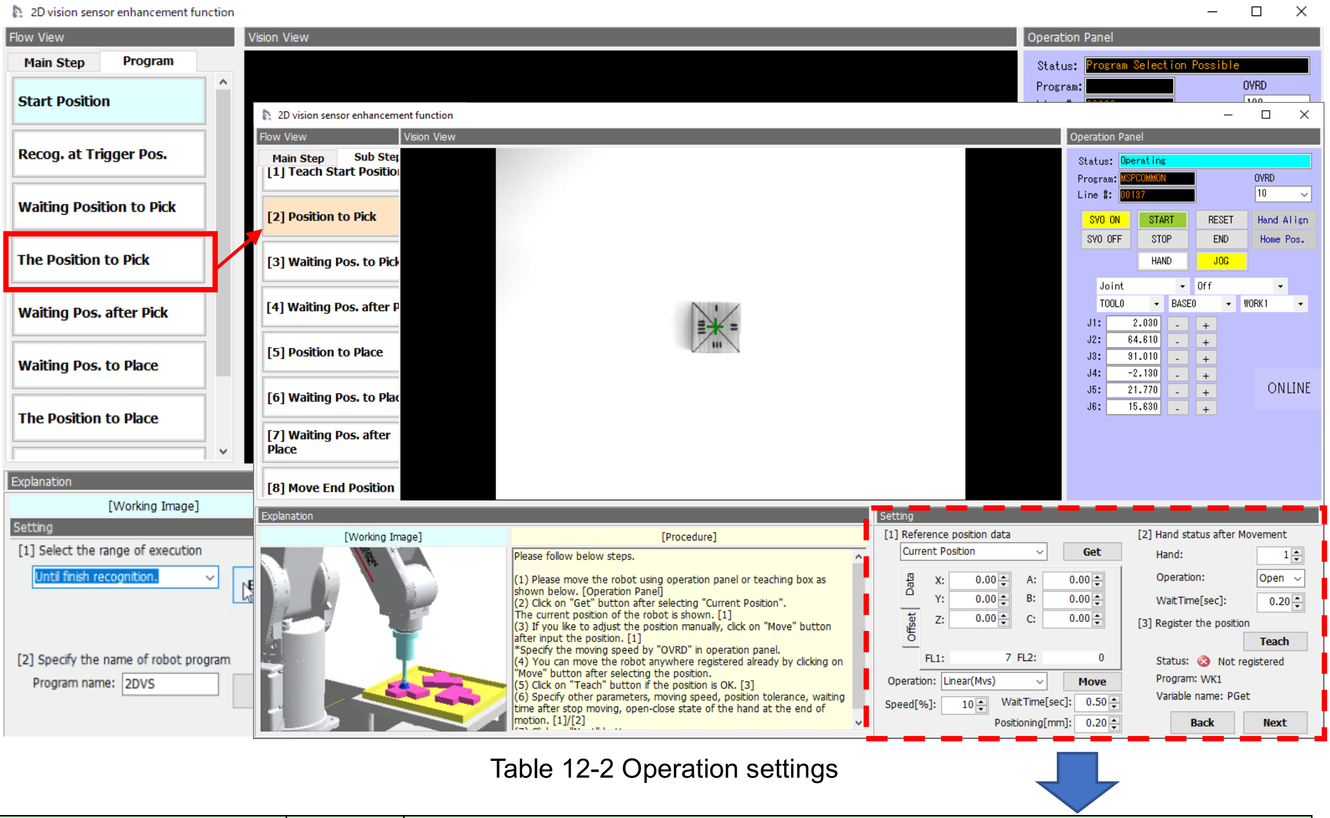


Table 12-2 Operation settings

Setting		Initial value	Function explanation
[1]	Speed [%]	10	Specify an Ovrd value for the movement to the registered position on this screen.
	Wait Time [sec]	0.50	Specify the waiting time after the robot moves.
	Positioning [mm]	0.20	If a value other than "0" is set, the Fine P function will be activated. If "0.20" is set, a travel completion state will be established after the robot moves to an area within 0.2 mm of a specified position. Then, the following process will be performed. If "0" is specified, the Cnt 1 function will be activated, and a specified position will be passed smoothly. For information on Fine commands and Cnt commands, refer to the "Instruction Manual/Detailed explanations of functions and operations". Stop the movement of the robot at the imaging position. For further information, refer to <Basic adjustments>.
[2]	Hand	1	Specify the number of the hand that opens or closes after movement to the registered position on this screen. The setting of the correspondence between the hand number and the hand connected to the robot can be changed by parameter settings. For further information, refer to the "Instruction Manual/Detailed explanations of functions and operations". (Related parameters: HIOTYPE, HANDTYPE, HANDINIT)
	Operation	Open	Specify the operation of the hand (open or close).
	Wait Time [sec]	0.20	Specify the waiting time after the hand opens or closes.

The operation settings shown above can be configured on the setting screen that includes robot operation. Configure the settings by referring to the adjustment method shown below.

<Basic adjustments>

- The operating speed is set to low speed from the factory. Increase the speed setting value gradually while checking the actual operation.
- If positioning is not required such as motion path points, set "0" for the positioning accuracy and waiting time. These settings allow the robot to proceed to the next operation with no stop, reducing takt time greatly.
- Operations that require positioning (to move to the workpiece grasp position or the vision sensor imaging position) become stable when a small value is set in "Positioning" and a large value is set in "Wait Time". However, takt time takes long in this situation. Adjust the settings while checking the robot operation and ensuring operational stability.

12.3. Saving a program

After checking and modifying the operation, save the robot program. Specify a robot program name as the main program in "Specify the robot program name", then click the [Save] button. The program will be saved in the robot controller.

Clicking the [Finish] button will display a message asking to export the setting file or back up the vision sensor settings. To use the settings configured in the wizard for a different controller, save the settings. For further information, refer to "15 MAINTENANCE".

13. ACTUAL OPERATION

Performing the robot program saved in "12.3 Saving a program" enables operation for the created application. However, the program must be customized for the system to allow for factors such as interlocking with the peripheral equipment and adding motion path points for interference avoidance.

13.1. Overview of robot program modification

The following table shows program modification examples.

Table 13-1 Program modification examples

	Classification	Information	Remarks
1	Vision imaging	Controlling when to take images Example) Interlocking with the peripheral equipment	Refer to [Example 1].
2		Modifying a motion path to the vision imaging position Example) Avoiding interference with the peripheral equipment	
3	Pick-and-place operations	Controlling when to start operation Example) Interlocking with the peripheral equipment	
4		Modifying a motion path Example) Avoiding interference with the peripheral equipment	Refer to [Example 2].
5	Error processing	Changing the processing at error occurrence Example) Notification of error conditions and recovery	Refer to [Example 3].

(The above examples are typical modification examples.)

If the basic structure or variable names of the program are changed, the application will not operate properly, and the setting screens will not be linked properly. Note the following points when modifying a program.

<Precautions>

① Do not change the basic structure of the existing programs.
② Do not change the existing variable names (external variable names) and Function names.
③ If adding a variable, define it with a name different from the variable names described in "14.2 List of variables".
④ If adding a function, define it with a name different from the Function names described in "14.3 Function procedures".

Modify the program by referring to "13.2(2) Program modification examples".

13.2. Program structure and modification examples

The following shows the main program structure and program modification examples when the application "Pick a part using a fixed camera" is selected.
The comments in the created program are displayed in English.

(1) Program structure

■ Program name: 2DVS

```
1 '##################################################
2 '# App 1
3 '# Robot Operation Program
4 '# This sample program is adjust a placement position with upper side Fixed vision.
5 '# Date of creation/version : 2021.06.21 A1
6 '# COPYRIGHT : MITSUBISHI ELECTRIC CORPORATION.
7 '##################################################
8 #Include "MFPLINK"
9 '
10 '#### Declare the global variables ####
11 Def Pos PWkNum      Define PWkNum as a position variable. (Set the workpiece type number in the X component.)
12 '
13 '#########################
14 '# Main process
15 '#########################
                                        Initialization
16 '+++ (1) Initialization +++
17 GoSub *GetData      Branch to the GetData label to acquire registered information (taught positions and settings).
18 MFPInitialization(PHome)      Perform robot's initial operation.
19 '[MFP_Hlt_1]
20 '+++ Vision Activation Check +++
21 MErrSts = MFPVsActivation(MVisionNo1)      Check the connection with the vision sensor.
22 If MErrSts < 0 Then GoTo *VsErr1      Branch to the VsErr1 label if a connection error occurs.
23 '
24 *LOOP
25 '+++ (2) Vision process +++      Vision processing
26 MFPGoTrigger1(PVsTrg1)      Move to the vision imaging position.
27 MFPVision1(MVisionNo1, PWkNum.X, MVsFound1, PVSDATA1, MErrSts)      Start vision imaging.
28 If MVsFound1 = 0 Then GoTo *VsErr2      Branch to the VsErr2 label if imaging fails.
29 '[MFP_Hlt_2]
                                        Pick-and-place operation processing
30 '+++ (3) Picking and Placement process +++
31 For MCount = 1 To MVsFound1      Repeat pick and place operations for the number of workpieces recognized during imaging.
32 PPICK = MFPVsGetNAlignUpFixCamT(MVisionNo1, MCalibNo1, MCount, PVsTrg1, MErrSts)
        Calculate the position for pick-up operation.
33 PPICKBk = PPICK * P_GBack      Calculate the position before pick-up operation.
34 PPICKUp = PPICK * P_GNext      Calculate the position after pick-up operation.
35 MFPBeforeGet(PPICKBk)      Pick-up operation (operation before pick-up operation)....[Corresponds to the operation in Section 10.3]
36 MFPGet(PPICK)      Pick-up operation....[Corresponds to the operation in Section 10.2]
37 MFPAfterGet(PPICKUp)      Pick-up operation (operation after pick-up operation)....[Corresponds to the operation in Section 10.4]
38 '[MFP_Hlt_3]
39 MFPBeforePut(PPutBk)      Placement operation (operation before placement operation) [Corresponds to the operation in Section 10.6]
40 MFPPut(PPut)      Placement operation....[Corresponds to the operation in Section 10.5]
41 MFPAfterPut(PPutUp)      Placement operation (operation after placement operation)....[Corresponds to the operation in Section 10.7]
42 '[MFP_Hlt_4]
43 Next MCount
```

```
44 '+++ (4) Move to Home position +++
45 MFPGoHome(PHome)          Move to the end point.
46 GoTo *LOOP
47 *LEnd
48 '[MFP_Hlt_5]
49 Hlt
50 End
51 '
52 '### Error Process ###
53 *VsErr1
54 Error 9100          Output the L9100 error.
55 Hlt
56 End
57 '
58 *VsErr2
59 Error 9101          Output the L9101 error.
60 Hlt
61 End
62 '
63 '#############################
64 '# Get the Teaching Data
65 '#############################
66 *GetData
67 MVisionNo1 = MFPGetVisionNo(1)     Acquire the vision sensor number.
68 MCalibNo1    = MFPGetCalibNo(1)    Acquire the calibration number.
69 MSP_GetWorkData(PWkNum.X)          Set the workpiece information set on each screen in the
                                    user-defined external variable.
70 PVsTrg1 = P_VsTrg1
71 PHome    = P_Home
72 PPutBk    = P_Put*P_PBack
73 PPut       = P_Put
74 PPutUp    = P_Put*P_PNext
75 PGetBk    = P_Get*P_GBack
76 PGet       = P_Get
77 PGetUp    = P_Get*P_GNext
78 Return
79 '
```

(2) Program modification examples

[Example 1] Modifying the program so that imaging is not performed until the signal (input signal 8) has been input

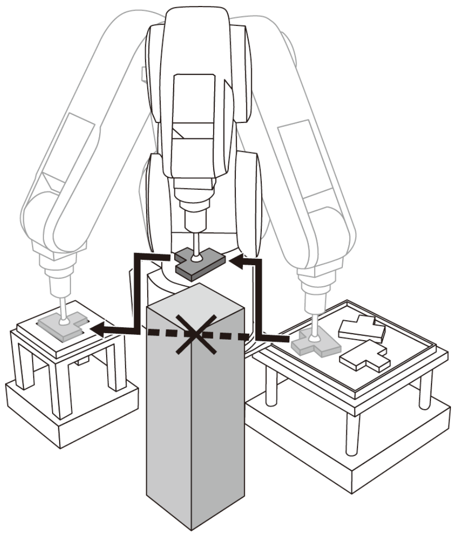
(Before modification)

```
25'+++ (2) Vision process +++                                          Vision processing
26 MFPGoTrigger1(PVsTrg1)          Move to the vision imaging position.
27 MFPVision1(MVisionNo1, PWkNum.X, MVsFound1, PVSDATA1, MErrSts)      Start vision imaging.
28 If MVsFound1 = 0 Then GoTo *VsErr2       Branch to the VsErr2 label if imaging fails.
```

(After modification)

```
25 '+++ (2) Vision process +++                                         Vision processing
26 MFPGoTrigger1(PVsTrg1)          Move to the vision imaging position.
27 'Wait M_In(8) = 1               Wait until input signal 8 has been turned ON.
28 MFPVision1(MVisionNo1, PWkNum.X, MVsFound1, PVSDATA1, MErrSts)      Start vision imaging.
29 If MVsFound1 = 0 Then GoTo *VsErr2       Branch to the VsErr2 label if imaging fails.
```

[Example 2] Adding a path point when the robot moves from the pick-up point to the placement point.



(Before modification)

```
                                                     Pick-and-place operation processing
30 '+++ (3) Picking and Placement process +++
31 For MCount = 1 To MVsFound1          Repeat pick and place operations for the number of workpieces
                                        recognized during imaging.
32 PPICK = MFPVsGetNAlignUpFixCamT(MVisionNo1, MCalibNo1, MCount, PVsTrg1, MErrSts)
                                        Calculate the position for pick-up operation.
33 PPICKBk = PPICK * P_GBack            Calculate the position before pick-up operation.
34 PPICKUp = PPICK * P_GNext            Calculate the position after pick-up operation.
35 MFPBeforeGet(PPICKBk)          Pick-up operation (operation before pick-up operation)
36 MFPGet(PPICK)                  Pick-up operation
37 MFPAfterGet(PPICKUp)           Pick-up operation (operation after pick-up operation)
38 '[MFP_Hlt_3]
39 MFPBeforePut(PPutBk)           Placement operation (operation before placement operation)
40 MFPPut(PPut)                   Placement operation
41 MFPAfterPut(PPutUp)            Placement operation (operation after placement operation)
42 '[MFP_Hlt_4]
43 Next MCount
```

(After modification)

	Pick-and-place operation processing
30 '+++ (3) Picking and Placement process +++	
31 For MCount = 1 To MVsFound1	Repeat pick and place operations for the number of workpieces recognized during imaging.
32 PPICK = MFPVsGetNAlignUpFixCamT(MVisionNo1, MCalibNo1, MCount, PVsTrg1, MErrSts)	Calculate the position for pick-up operation.
33 PPICKBk = PPICK * P_GBack	Calculate the position before pick-up operation.
34 PPICKUp = PPICK * P_GNext	Calculate the position after pick-up operation.
35 MFPBeforeGet(PPICKBk)	Pick-up operation (operation before pick-up operation)
36 MFPGet(PPICK)	Pick-up operation
37 MFPAfterGet(PPICKUp)	Pick-up operation (operation after pick-up operation)
38 Cnt 1	Add an operation.
39 MSP_Mov(80, 0, P2, 0, 1)	Move to position P2 using joint interpolation. (Ovrd=80% / Fine P=0mm / Dly=0s)
40 MSP_Mvs(30, 0.2, P3, 0.3)	Move to position P3 using linear interpolation. (Ovrd=30% / Fine P=0.2mm / Dly=0.3s)
41 MFPBeforePut(PPutBk)	Placement operation (operation before placement operation)
42 MFPPut(PPut)	Placement operation
43 MFPAfterPut(PPutUp)	Placement operation (operation after placement operation)
44 '[MFP_Hlt_4]	
45 Next MCount	

[Example 3] If imaging fails, the signal (output signal 8) is output for error notification. The imaging process continues after the signal (input signal 7), which indicates that the error has been removed, is input.

(Before modification)

52 '### Error Process ###	
53 *VsErr1	
54 Error 9100	Output the L9100 error.
55 Hlt	
56 End	
57 '	
58 *VsErr2	
59 Error 9101	Output the L9101 error.
60 Hlt	
61 End	

(After modification)

52 '### Error Process ###	
53 *VsErr1	
54 Error 9100	Output the L9100 error.
55 Hlt	
56 End	
57 '	
58 *VsErr2	
59 M_Out(8) =M_On	Output the signal that indicates an imaging error (output signal 8 in this example).
60 Wait M_In(7) = M_On	Wait until the cause of the imaging error has been removed (until input signal 7 has been input in this example).
61 M_Out(8) =M_OFF	
62 Goto *LOOP	Branch to the LOOP label and resume the process.
63 Error 9101	
64 Hlt	
65 End	

14. PROGRAM SPECIFICATIONS

14.1. List of robot programs

This function creates the following robot programs. (The program architecture differs depending on the application selected.)

Table 14-1 List of robot programs

Program name	Purpose	Remarks
VSCOGNEX	Vision sensor communication library	-
MFPACLBFIX	Fixed camera's calibration library	Note 1
MFPACLBHND	Hand camera's calibration library	Note 1
MFPCALCTL	Tool calculation library	Note 1
MFPVSALIGN	Alignment compensation library	Note 1
MFPLINK	I/F library for wizard	-
MEASURE	Program for measuring the field of view size	-
CALIBDATA	Recalculation program for calibration	-
MSPVISION	Vision control library	-
MSPCOMMON	Robot control library	-
BASE	User base program used in the "WK*" and "C*" programs	-
A1* (*: Camera No.)	Calibration program for downward-facing cameras	-
A2* (*: Camera No.)	Calibration program for hand cameras	-
A3* (*: Camera No.)	Calibration program for upward-facing cameras	-
A4* (*: Camera No.)	Calibration program for side-facing cameras	-
C1 to 9	Program for calculating a grasp position	-
WK* (* = Workpiece type number)	Program for defining information required to transport the workpiece (grasp position, placement point, etc.)	-
2DVSTMP (defined by customer)	Program to perform operations such as robot operation, vision recognition, and vision compensation	-

Note 1: The library cannot be viewed or edited.

14.2. List of variables

The data set on screens is stored as a robot program variable. The following table shows the list.

14.2.1. Variables for calibration

The position data and values set in "8 CALIBRATION" are saved in local variables that are used in the robot program "A*". The following table lists the variables.
(The settings are stored in P variables. Basically, the P variable handles position data. However, note that the variable is used to store data other than position data as well.)

Table 14-2 List of calibration variables

Screen title	Data	Variable (position variable)	Description
Register Start Position	Camera No.	PVsNum	X component: Camera No. (1 to 4)
	Starting point	PClbSt	Calibration start position [mm, deg]
Direction of Camera	Camera direction	PMechIfAng	The direction of the mechanical interface in relation to that of the camera (A, B, and C components only) [deg]
Execute Calibration	Field of view size	PViewSize	Field of view size [Pixel] X component: Field of view size (horizontal) Y component: Field of view size (vertical)
	Calibration No. and number of calibration points	PCalibData	X component: Calibration No. (0 to 8) Y component: Number of calibration points (4, 5, or 9)
	Movement interval	PSetOff (10)	Offset amount for movement to 9 positions max. during calibration [mm]
	Allowable rotation angle	PSetAngle (10)	Rotation angle during calibration [deg]
	Convergence target value	PThreshold	X component: Angle judged as convergence [deg] Y component: Distance judged as convergence [mm]
	Convergence retry counts	PRetry	X component: Maximum number of retries if convergence is not done [times]
	Travel distance	PMvDist	Travel amount in the robot tool Z direction to check whether the mark and camera are parallel to each other [mm]
Confirm the Results	9 points in the robot coordinates	PCalibRobo(9)	Positions to which the robot has moved during calibration (9 points max.) [mm]
	9 points in the vision coordinates	PCalibVision(9)	Pixel data for images taken at positions to which the robot moved during calibration (9 points max.) [Pixel]

In "5.6 Start screen", if you selected "Import", and the use of the robot program data, the following data saved in the robot program will be applied to the setting screen.

PVsNum
PClbSt
PMechIfAng
PThreshold
PRetry

14.2.2. Variables related to operation settings

The position data (enclosed in the red frames below) and operation parameters (enclosed in the blue frames below) taught in each step are saved in the "WK* (* = workpiece type number)" program.

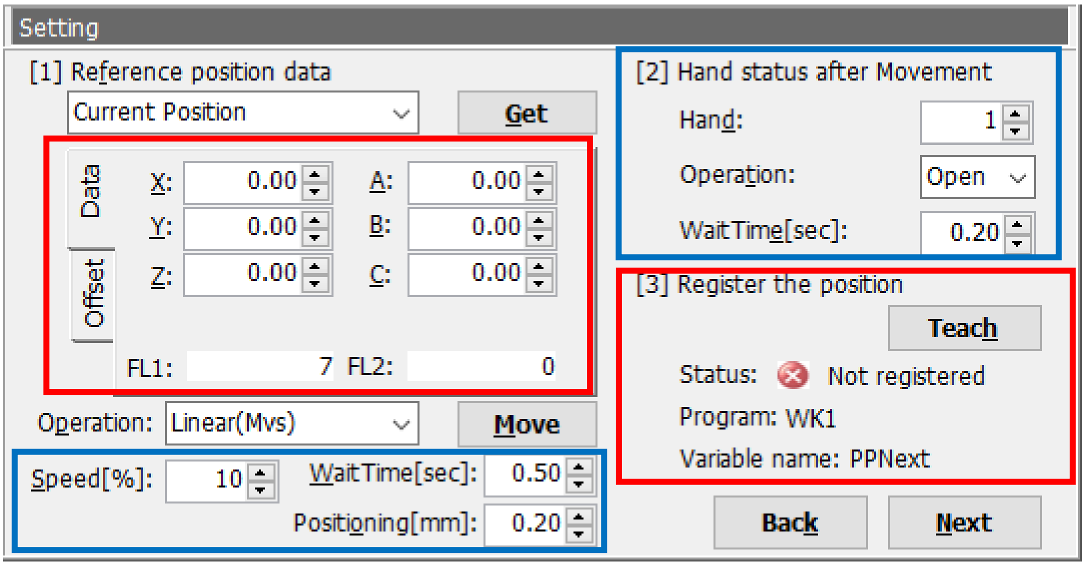


Fig. 14-1 Screen for teaching

The following table shows the data to be saved.
(The settings are stored in P variables. Basically, the P variable handles position data. However, note that the variable is used to store data other than position data as well.)

Table 14-3 List of variables related to operation settings

Program name	Variable name	Description
WK* (* = Workpiece type number)	PHome	Position data with the following two meanings Application starting point End point after all steps are complete
↑	PInitPrm	Operation parameter for movement to the application starting point X component = Speed (Ovrd) Y component = Positioning accuracy (Fine P) Z component = Waiting time after movement (Dly) A component = Hand No. B component = Hand operation (HOpen/HClose) C component = Waiting time after hand operation (Dly)
↑	PHomePrm	Operation parameter for movement after all steps are complete X component = Speed (Ovrd) Y component = Positioning accuracy (Fine P) Z component = Waiting time after movement (Dly) A component = Hand No. B component = Hand operation (HOpen/HClose) C component = Waiting time after hand operation (Dly)
↑	PVsTrg1 to 4	Robot position (trigger position) when taking images Variable names 1 to 4 are not camera numbers. They are trigger numbers, and trigger No.1 can be used for camera 4 imaging.
↑	PVsTrg1Prm to PVsTrg4Prm	Operation parameter for movement to the imaging position X component = Speed (Ovrd) Y component = Positioning accuracy (Fine P) Z component = Waiting time after movement (Dly) A component = Hand No. B component = Hand operation (HOpen/HClose) C component = Waiting time after hand operation (Dly)
↑	PGet	Position to grasp (suck) the workpiece [mm, deg]

Program name	Variable name	Description
↑	PGetPrm	Operation parameter for movement to the position to grasp (suck) the workpiece X component = Speed (Ovrd) Y component = Positioning accuracy (Fine P) Z component = Waiting time after movement (Dly) A component = Hand No. B component = Hand operation (HOpen/HClose) C component = Waiting time after hand operation (Dly)
↑	PPut	Position to release the workpiece
↑	PPutPrm	Operation parameter for movement to the position to release the workpiece X component = Speed (Ovrd) Y component = Positioning accuracy (Fine P) Z component = Waiting time after movement (Dly) A component = Hand No. B component = Hand operation (HOpen/HClose) C component = Waiting time after hand operation (Dly)
↑	PGBack	Position before the workpiece grasp (suck) point [mm, deg]
↑	PGBackPrm	Operation parameter for movement to the position before the workpiece grasp (suck) point X component = Speed (Ovrd) Y component = Positioning accuracy (Fine P) Z component = Waiting time after movement (Dly) A component = Hand No. B component = Hand operation (HOpen/HClose) C component = Waiting time after hand operation (Dly)
↑	PGNext	Position after the workpiece grasp (suck) point [mm, deg]
↑	PGNextPrm	Operation parameter for movement to the position after the workpiece grasp (suck) point X component = Speed (Ovrd) Y component = Positioning accuracy (Fine P) Z component = Waiting time after movement (Dly) A component = Hand No. B component = Hand operation (HOpen/HClose) C component = Waiting time after hand operation (Dly)
↑	PPBack	Position before the workpiece release point [mm, deg]
↑	PPBackPrm	Operation parameter for movement to the position before the workpiece release point X component = Speed (Ovrd) Y component = Positioning accuracy (Fine P) Z component = Waiting time after movement (Dly) A component = Hand No. B component = Hand operation (HOpen/HClose) C component = Waiting time after hand operation (Dly)
↑	PPNext	Position after the workpiece release point [mm, deg]
↑	PPNextPrm	Operation parameter for movement to the position after the workpiece release point X component = Speed (Ovrd) Y component = Positioning accuracy (Fine P) Z component = Waiting time after movement (Dly) A component = Hand No. B component = Hand operation (HOpen/HClose) C component = Waiting time after hand operation (Dly)
↑	PPltG1 to 3	Three points to define a pallet for picking (imaging) [mm, deg]
↑	PPltGPrm	Conditions to define a pallet for picking (imaging) X component: Number of divisions between the 1st point and the 2nd point Y component: Number of divisions between the 1st point and the 3rd point Z component: Grid direction (11: Zigzag, 12: The same direction)
↑	PPltP1 to 3	Three points to define a pallet for placement [mm, deg]

Program name	Variable name	Description
↑	PPltPPrm	Conditions to define a pallet for placement X component: Number of divisions between the 1st point and the 2nd point Y component: Number of divisions between the 1st point and the 3rd point Z component: Grid direction (11: Zigzag, 12: The same direction)
↑	PPltT1 to 3	Three points to define a pallet for imaging (triggering) [mm, deg]
↑	PPltTPrm	Conditions to define a pallet for imaging (triggering) X component: Number of divisions between the 1st point and the 2nd point Y component: Number of divisions between the 1st point and the 3rd point Z component: Grid direction (11: Zigzag, 12: The same direction)
↑	PGOffset	The offset set in "Adjust the position to pick a part" for grasp position registration (set in the tool coordinate system) X component: Offset amount in the X direction Y component: Offset amount in the Y direction Z component: Offset amount in the Z direction A component: Offset amount in the A direction B component: Offset amount in the B direction C component: Offset amount in the C direction
↑	PPOffset	The offset set in "Adjust the position to place a part" for grasp position registration (set in the tool coordinate system) X component: Offset amount in the X direction Y component: Offset amount in the Y direction Z component: Offset amount in the Z direction A component: Offset amount in the A direction B component: Offset amount in the B direction C component: Offset amount in the C direction
Main program	PWkNum	Workpiece type No. X component: Type number of the workpiece to be transported

If you selected the use of the robot program data in the import process, the above data saved in the robot program will be applied to the setting screen.

14.3. Function procedures

This section explains Function procedures to be used in the robot program created with this function.

14.3.1. List of Function procedures

The following table lists Function procedures made available to the customer.

Table 14-4 List of Function procedures

Classification	Function name	Function	Page
Vision sensor connection	MFPVsActivation	Connecting the vision sensor	14-113
	MFPVsActivation2P		14-114
	MFPVsActivation4P		14-115
Imaging	MFPVision1	Taking images with the vision sensor	14-116
	MFPVision1C2P		14-117
	MFPVision2C2P		14-118
	MFPVision12C2P		14-119
	MFPVision34C2P		14-120
Alignment adjustment	MFPVisionAlignHandCamT	Taking images with the vision sensor and calculating a position to which the robot moves	14-121
	MFPVisionAlignLowFixCamT		14-122
	MFPVsGetNAlignUpFixCamT		14-123
	MFPVsGetNAlignHandCamT		14-124
	MFPVsGetNAlignLowFixCamT		14-125
	MFPVsAlignHandCamT2P		14-126
	MFPVsAlignLowFixCamT2P		14-127
Preset operation	MFPInitialization	Moving the robot according to the settings configured in "Teach Start Position"	14-128
	MFPGoTrigger1	Moving the robot according to the settings configured in "Register Trigger Pos."	14-129
	MFPGoTrigger2		14-130
	MFPGoTrigger3		14-131
	MFPGoTrigger4		14-132
	MFPBeforeGet	Moving the robot according to the settings configured in "Waiting Pos. to Pick"	14-133
	MFPGet	Moving the robot according to the settings configured in "Position to Pick"	14-134
	MFPAfterGet	Moving the robot according to the settings configured in "Waiting Pos. after Pick"	14-135
	MFPBeforePut	Moving the robot according to the settings configured in "Waiting Pos. to Place"	14-136
	MFPPut	Moving the robot according to the settings configured in "Position to Place"	14-137
	MFPAfterPut	Moving the robot according to the settings configured in "Waiting Pos. after Place"	14-138
	MFPGoHome	Moving the robot according to the settings configured in "Move End Position"	14-139
	MSP_GetWorkData	Substituting the values set on each screen for the user-defined external variable.	14-140
Move	MSP_Mov_Hand	Moving to the destination point using joint interpolation and opening or closing a specified hand	14-141
	MSP_Mvs_Hand	Moving to the destination point using linear interpolation and opening or closing a specified hand	14-142
	MSP_Mov	Moving to the destination point using joint interpolation	14-143
	MSP_Mvs	Moving to the destination point using linear interpolation	14-144
Opening/closing hand	MSP_Hand	Opening or closing a specified hand	14-145

CAUTION

Do not edit or delete Function procedures. Doing so may cause the robot program not to run properly.
Do not use Function procedures not shown in the table. Doing so may cause the robot program not to run properly.

14.3.2. Detailed explanations of Function procedures

This section provides detailed information on Function procedures individually.

14.3.2.1. Interpreting descriptions

[Function]: Indicates the function of the Function procedure.
[Syntax]: Indicates how to enter arguments of the Function procedure.
[Reference program]: Shows an example of a program that uses the Function procedure.
[Terminology]: Indicates the meaning, range, and other information of an argument.
[Reference]: Indicates related Function procedures.
[Related variable]: Indicates related variables.
[Program]: Indicates the robot program in which the Function procedure is defined.

14.3.2.2. Explanation on each Function

Each Function is explained as follows.

MFPVsActivation

[Function]

In the following applications, a connection with one specified vision sensor is established.Make sure to perform this Function before performing the Functions of "Imaging" and "Alignment adjustment".

<Application>

[1] Pick a part using a fixed camera
[2] Pick a part using a hand camera
[3] Correct grip errors using an upper looking fixed camera
[4] Correct grip errors using a side looking fixed camera
[5] Pick a part from palette using a hand camera

[Syntax]

```
<Numerical variable>=MFPVsActivation(MVsNum)
```

[Terminology]

<MVsNum> Specify a vision number. Any value from 1 to 4 can be specified.
<Numerical variable> The error status is substituted.

0: Line connected * Same as the one returned by the state variable M_NvOpen
1: No error * Same as the one returned by the state variable M_NvOpen
-1: Not connected * Same as the one returned by the state variable M_NvOpen
-2: The vision number "MVsNum" is specified incorrectly.

[Reference program]

```
1 MErrSts = MFPVsActivation(MVisionNo1)            'Vision sensor activation
2 MFPVision1(MVisionNo1, PWkNum.X, MVsFound1, PVSDATA1, MErrSts)
3 PPICK = MFPVsGetNAlignUpFixCamT(MVisionNo1, MCalibNo1, MCount, PVsTrg1, MErrSts)
'Based on the imaging result (PVSDATA1), calculate the position to which the robot moves.
```

[Reference]

MFPVision1, MFPVisionAlignHandCamT, MFPVisionAlignLowFixCamT

[Program]

VSCOGNEX

MFPVsActivation2P

[Function]

In the following applications, a connection with two specified vision sensors is established.Make sure to perform this Function before performing the Functions of "Imaging" and "Alignment adjustment".

<Application>

[6] Using a hand camera, correcting grip errors by fixed camera
[7] Correct grip errors (UF), place a part (HC)
[8] Correct grip errorsx2 (UF), place a partx2 (HC)

[Syntax]

```
<Numerical variable>=MFPVsActivation2P(MVsNum1, MVsNum2)
```

[Terminology]

<MVsNum1> Specify a vision number. Any value from 1 to 4 can be specified.
<MVsNum2> Specify a vision number. Any value from 1 to 4 can be specified.
<Numerical variable> The error status is substituted.

0: Line connected * Same as the one returned by the state variable M_NvOpen
1: No error * Same as the one returned by the state variable M_NvOpen
-1: Not connected * Same as the one returned by the state variable M_NvOpen
-2: Either vision number MVsNum1 or MVsNum2 is specified incorrectly.

[Reference program]

```
1 MErrSts = MFPVsActivation2P(MVisionNo1, MVisionNo2)    'Vision sensor activation
2 MFPGoTrigger1(PVsTrg1)                                  'Move to the first imaging position.
3 MFPVision1C2P(MVisionNo1, 1, PWkNum.X, MVsFound1, PVSDATA1, MErrStt1)
4 MFPGoTrigger2(PVsTrg2)                                  'Move to the second imaging position.
5 MFPVision1C2P(MVisionNo1, 2, PWkNum.X, MVsFound2, PVSDATA2, MErrStt2)
6 PPLACE = MFPVsAlignLowFixCamT2P(MVisionNo1, MCalibNo1, PVSDATA1, MVisionNo1,
                 MCalibNo1, PVSDATA2, PPLACETMP, MErrStt)
                 'Based on the imaging results (PVSDATA1, PVSDATA2), calculate the position to which
                 the robot moves.
```

[Reference]

MFPVision1C2P, MFPVision2C2P, MFPVisionAlignHandCamT

[Program]

VSCOGNEX

MFPVsActivation4P

[Function]

In the following application, a connection with four specified vision sensors is established.Make sure to perform this Function before performing the Functions of "Imaging" and "Alignment adjustment".

<Application>
[9] Correct grip errors (UFx2), place a part (HCx2)

[Syntax]

```
<Numerical variable>= MFPVsActivation4P(MVsNum1, MVsNum2, MVsNum3, MVsNum4)
```

[Terminology]

<MVsNum1> Specify a vision number. Any value from 1 to 4 can be specified.
<MVsNum2> Specify a vision number. Any value from 1 to 4 can be specified.
<MVsNum3> Specify a vision number. Any value from 1 to 4 can be specified.
<MVsNum4> Specify a vision number. Any value from 1 to 4 can be specified.
<Numerical variable> The error status is substituted.

0: Line connected * Same as the one returned by the state variable M_NvOpen
1: No error * Same as the one returned by the state variable M_NvOpen
-1: Not connected * Same as the one returned by the state variable M_NvOpen
-2: Any of vision number MVsNum1, MVsNum2, MVsNum3, or MVsNum4 is specified incorrectly.

[Reference program]

```
1 MErrSts = MFPVsActivation4P(MVisionNo1, MVisionNo2, MVisionNo3, MVisionNo4)
                                                'Vision sensor activation
2 MFPGoTrigger1(PVsTrg1)                        'Move to the first imaging position.
3 MFPVision12C2P(MVisionNo1, 1, PWkNum.X, MVsFound1, PVSDATA1, MErrStt1)
4 MFPGoTrigger2(PVsTrg2)                        'Move to the second imaging position.
5 MFPVision12C2P(MVisionNo2, 2, PWkNum.X, MVsFound2, PVSDATA2, MErrStt2)
6 PPLACE = MFPVsAlignLowFixCamT2P(MVisionNo1, MCalibNo1, PVSDATA1, MVisionNo2,
    MCalibNo2, PVSDATA2, PPLACETMP, MErrStt)
    'Based on the imaging results (PVSDATA1, PVSDATA2), calculate the position to which the robot
    moves.
```

[Reference]

MFPVision12C2P, MFPVision34C2P

[Program]

VSCOGNEX

MFPVision1

[Function]

In the following applications, the program for a specified vision sensor is started, and the results of vision sensor recognition are received.Received results are saved in the argument.This Function is used to take one image of the workpiece with one vision sensor for alignment adjustments.

<Application>

[1] Pick a part using a fixed camera
[2] Pick a part using a hand camera
[3] Correct grip errors using an upper looking fixed camera
[4] Correct grip errors using a side looking fixed camera
[7] Correct grip errors (UF), place a part (HC)

[Syntax]

```
<No return value>=MFPVision1(MVsNum, MModelNo, ByRef MFound, ByRef PVSData,
                             ByRef MErr)
```

[Terminology]

<MVsNum> Specify a vision number. Any value from 1 to 4 can be specified.

<MModelNo> Specify a workpiece type number. Any value from 1 to 5 can be specified.

<MFound> The number of workpieces recognized by the vision sensor is substituted. [Call by reference]

<PVSData> The position of the workpiece recognized by the vision sensor is substituted. [Call by reference]

<MErr> The error status is substituted. [Call by reference]
1: No error
-2: The vision number "MVsNum" is specified incorrectly.

[Reference program]

```
1 MErrSts = MFPVsActivation(MVisionNo1)                'Vision sensor activation
2 MFPVision1(MVisionNo1, PWkNum.X, MVsFound1, PVSDATA1, MErrSts)
3 PPICK = MFPVsGetNAlignUpFixCamT(MVisionNo1, MCalibNo1, MCount, PVsTrg1, MErrSts)
                                        'Based on the imaging result (PVSDATA1), calculate the
                                         position to which the robot moves.
```

[Reference]

MFPVsGetNAlignUpFixCamT, MFPVsGetNAlignHandCamT, MFPVsGetNAlignLowFixCamT

[Program]

MFPLINK

MFPVision1C2P

[Function]

In the following applications, the job file for a specified vision sensor is started, and the results of vision sensor recognition are received.Received results are saved in the argument.This Function is used to take two images of the workpiece with one vision sensor for alignment adjustments.

<Application>
[6] Using a hand camera, correcting grip errors by fixed camera
[8] Correct grip errorsx2 (UF), place a partx2 (HC)

[Syntax]

```
<No return value>= MFPVision1C2P(MVsNum, MPosNum, MModelNo, ByRef MFound,
                                 ByRef PVSData, ByRef MErr)
```

[Terminology]

<MVsNum>	Specify a vision number. Any value from 1 to 4 can be specified.
<MPosNum>	Specify the imaging position. Either value 1 or 2 can be specified. Specify 1 to take an image at the first position, or specify 2 to take an image at the second position.
<MModelNo>	Specify a workpiece type number. Any value from 1 to 5 can be specified.
<MFound>	The number of workpieces recognized by the vision sensor is substituted. [Call by reference]
<PVSData>	The position of the workpiece recognized by the vision sensor is substituted. [Call by reference]
<MErr>	The error status is substituted. [Call by reference] 1: No error -2: The vision number "MVsNum" is specified incorrectly. -3: The imaging position "MPosNum" is specified incorrectly.

[Reference program]

```
1 MErrSts = MFPVsActivation2P(MVisionNo1, MVisionNo2)      'Vision sensor activation
2 MFPGoTrigger1(PVsTrg1)                         'Move to the first imaging position.
3 MFPVision1C2P(MVisionNo1, 1, PWkNum.X, MVsFound1, PVSDATA1, MErrStt1)
4 MFPGoTrigger2(PVsTrg2)                         'Move to the second imaging position.
5 MFPVision1C2P(MVisionNo1, 2, PWkNum.X, MVsFound2, PVSDATA2, MErrStt2)
6 PPLACE = MFPVsAlignLowFixCamT2P(MVisionNo1, MCalibNo1, PVSDATA1, MVisionNo1,
                                   MCalibNo1,PVSDATA2, PPLACETMP, MErrStt)
                                   'Based on the imaging results (PVSDATA1, PVSDATA2),
                                   calculate the position to which the robot moves.
```

[Reference]

MFPVsAlignLowFixCamT2P

[Program]

MFPLINK

MFPVision2C2P

[Function]

In the following application, the job file for a specified vision sensor is started, and the results of vision sensor recognition are received.Received results are saved in the argument.This Function is used to take two images of the workpiece with one vision sensor for alignment adjustments.

<Application>

[8] Correct grip errorsx2 (UF), place a partx2 (HC)

[Syntax]

```
<No return value>= MFPVision2C2P(MVsNum, MPosNum, MModelNo, ByRef MFound,
                                  ByRef PVSData, ByRef MErr)
```

[Terminology]

<MVsNum>	Specify a vision number. Any value from 1 to 4 can be specified.
<MPosNum>	Specify the imaging position. Either value 1 or 2 can be specified. Specify 1 to take an image at the first position, or specify 2 to take an image at the second position.
<MModelNo>	Specify a workpiece type number. Any value from 1 to 5 can be specified.
<MFound>	The number of workpieces recognized by the vision sensor is substituted. [Call by reference]
<PVSData>	The position of the workpiece recognized by the vision sensor is substituted. [Call by reference]
<MErr>	The error status is substituted. [Call by reference] 1: No error -2: The vision number "MVsNum" is specified incorrectly. -3: The imaging position "MPosNum" is specified incorrectly.

[Reference program]

```
1 MErrSts = MFPVsActivation2P(MVisionNo1, MVisionNo2)     'Vision sensor activation
2 MFPGoTrigger3(PVsTrg3)                          'Move to the first imaging position.
3 MFPVision2C2P(MVisionNo2, 1, PWkNum.X, MVsFound3, PVSDATA3, MErrStt3)
4 MFPGoTrigger4(PVsTrg4)                          'Move to the second imaging position.
5 MFPVision2C2P(MVisionNo2, 2, PWkNum.X, MVsFound4, PVSDATA4, MErrStt4)
6 PPLACETMP = MFPVsAlignHandCamT2P(MVisionNo2, MCalibNo2, PVSDATA3, PVsTrg3,
                                   MVisionNo2, MCalibNo2, PVSDATA4, PVsTrg4,
                                   MErrStt)
                                   'Based on the imaging results (PVSDATA3,
                                   PVSDATA4), calculate the position to which the robot
                                   moves.
```

[Reference]

MFPVsAlignHandCamT2P

[Program]

MFPLINK

MFPVision12C2P

[Function]

In the following application, the job file for a specified vision sensor is started, and the results of vision sensor recognition are received.
Received results are saved in the argument.
This Function is used to take two images of the workpiece with two vision sensors for alignment adjustments.

<Application>
[9] Correct grip errors (UFx2), place a part (HCx2)

[Syntax]

```
<No return value>= MFPVision12C2P(MVsNum, MPosNum, MModelNo, ByRef MFound,
                                  ByRef PVSData, ByRef MErr)
```

[Terminology]

<MVsNum> Specify a vision number. Any value from 1 to 4 can be specified.
<MPosNum> Specify the imaging position. Either value 1 or 2 can be specified.
Specify 1 to take an image at the first position, or specify 2 to take an image at the second position.
<MModelNo> Specify a workpiece type number. Any value from 1 to 5 can be specified.
<MFound> The number of workpieces recognized by the vision sensor is substituted. [Call by reference]
<PVSData> The position of the workpiece recognized by the vision sensor is substituted. [Call by reference]
<MErr> The error status is substituted. [Call by reference]
1: No error
-2: The vision number "MVsNum" is specified incorrectly.

[Reference program]

```
1 MErrSts = MFPVsActivation4P(MVisionNo1, MVisionNo2, MVisionNo3, MVisionNo4)
                                                'Vision sensor activation
2 MFPGoTrigger1(PVsTrg1)                        'Move to the first imaging position.
3 MFPVision12C2P(MVisionNo1, 1, PWkNum.X, MVsFound1, PVSDATA1, MErrStt1)
4 MFPGoTrigger2(PVsTrg2)                        'Move to the second imaging position.
5 MFPVision12C2P(MVisionNo2, 2, PWkNum.X, MVsFound2, PVSDATA2, MErrStt2)
6 PPLACE = MFPVsAlignLowFixCamT2P(MVisionNo1, MCalibNo1, PVSDATA1, MVisionNo2,
                                  MCalibNo2, PVSDATA2, PPLACETMP, MErrStt)
                                  'Based on the imaging results (PVSDATA1, PVSDATA2),
                                  calculate the position to which the robot moves.
```

[Reference]

MFPVsAlignLowFixCamT2P

[Program]

MFPLINK

MFPVision34C2P

[Function]

In the following application, the job file for a specified vision sensor is started, and the results of vision sensor recognition are received.
Received results are saved in the argument.
This Function is used to take two images of the workpiece with two vision sensors for alignment adjustments.

<Application>
[9] Correct grip errors (UFx2), place a part (HCx2)

[Syntax]

```
<No return value>= MFPVision34C2P(MVsNum, MPosNum, MModelNo, ByRef MFound,
                                  ByRef PVSData, ByRef MErr)
```

[Terminology]

<MVsNum>	Specify a vision number. Any value from 1 to 4 can be specified.
<MPosNum>	Specify the imaging position. Either value 1 or 2 can be specified. Specify 1 to take an image at the first position, or specify 2 to take an image at the second position.
<MModelNo>	Specify a workpiece type number. Any value from 1 to 5 can be specified.
<MFound>	The number of workpieces recognized by the vision sensor is substituted. [Call by reference]
<PVSData>	The position of the workpiece recognized by the vision sensor is substituted. [Call by reference]
<MErr>	The error status is substituted. [Call by reference] 1: No error -2: The vision number "MVsNum" is specified incorrectly.

[Reference program]

```
1 MErrSts = MFPVsActivation4P(MVisionNo1, MVisionNo2, MVisionNo3, MVisionNo4)
                                        'Vision sensor activation
2 MFPGoTrigger3(PVsTrg3)                'Move to the first imaging position.
3 MFPVision34C2P(MVisionNo3, 1, PWkNum.X, MVsFound3, PVSDATA3, MErrStt3)
4 MFPGoTrigger4(PVsTrg4)                'Move to the second imaging position.
5 MFPVision34C2P(MVisionNo4, 2, PWkNum.X, MVsFound4, PVSDATA4, MErrStt4)
6 PPLACETMP = MFPVsAlignHandCamT2P(MVisionNo3, MCalibNo3, PVSDATA3, PVsTrg3,
                                   MVisionNo4, MCalibNo4, PVSDATA4, PVsTrg4,
                                   MErrStt)
                                   'Based on the imaging results (PVSDATA3, PVSDATA4),
                                   calculate the position to which the robot moves.
```

[Reference]

MFPVsAlignHandCamT2P

[Program]

MFPLINK

MFPVisionAlignHandCamT

[Function]

Images are taken with a specified vision sensor (hand camera), and the position to which the robot moves is calculated based on the imaging results.

This Function is used to take one image of the workpiece with one vision sensor for alignment adjustments.

[Syntax]

```
<Position variable>= MFPVisionAlignHandCamT(MVsNum, MModelNo, MCalibNum,
                                          PVsTrgPos, ByRef MFound, ByRef MErr)
```

[Terminology]

<MVsNum> Specify a vision number. Any value from 1 to 4 can be specified.
<MModelNo> Specify a workpiece type number. Any value from 1 to 5 can be specified.
<MCalibNum> Specify a calibration number. Any value from 1 to 8 can be specified.
Ensure that the calibration number is the same value as the vision number.
<PVsTrgPos> Specify the imaging position.
<MFound> The number of workpieces recognized by the vision sensor is substituted. [Call by reference]
<MErr> The error status is substituted. [Call by reference]
1: No error
-1: The calibration number "MCalibNum" is specified incorrectly.
-2: The vision number "MVsNum" is specified incorrectly.

[Reference program]

```
1 MErrSts = MFPVsActivation(MVisionNo1)                    'Vision sensor activation
2 MFPGoTrigger1(PVsTrg1) 'Move to the imaging position.
3 PPLACETMP = MFPVisionAlignHandCamT(MVisionNo1, PWkNum.X, MCalibNo1, PVsTrg1,
                                     MVsFound1, MErrStt)
                                    'Take images and calculate the position to which the
                                     robot moves based on the imaging results.
```

[Reference]

MFPGoTrigger1, MFPGoTrigger2, MFPGoTrigger3, MFPGoTrigger4

[Program]

MFPLINK

MFPVisionAlignLowFixCamT

[Function]

Images are taken with a specified vision sensor (fixed upward-facing camera), and the position to which the robot moves is calculated based on the imaging results.

This Function is used to take one image of the workpiece with one vision sensor for alignment adjustments.

[Syntax]

```
<Position variable>= MFPVisionAlignLowFixCamT(MVsNum, MModelNo, MCalibNum,
                                    POperation, ByRef MFound, ByRef MErr)
```

[Terminology]

<MVsNum>	Specify a vision number. Any value from 1 to 4 can be specified.
<MModelNo>	Specify a workpiece type number. Any value from 1 to 5 can be specified.
<MCalibNum>	Specify a calibration number. Any value from 1 to 8 can be specified. Ensure that the calibration number is the same value as the vision number.
<POperation>	Specify a position to adjust alignment.
<MFound>	The number of workpieces recognized by the vision sensor is substituted. [Call by reference]
<MErr>	The error status is substituted. [Call by reference] 1: No error -1: The calibration number "MCalibNum" is specified incorrectly. -2: The vision number "MVsNum" is specified incorrectly.

[Reference program]

```
1 MErrSts = MFPVsActivation(MVisionNo1)             'Vision sensor activation
2 MFPGoTrigger1(PVsTrg1)                            'Move to the imaging position.
3 PPLACE = MFPVisionAlignLowFixCamT(MVisionNo1, PWkNum.X, MCalibNo1, PPut, MVsFound1,
                                    MErrSts)
                                    'Take images and calculate the position to which the robot
                                    moves based on the imaging results.
```

[Reference]

MFPGoTrigger1, MFPGoTrigger2, MFPGoTrigger3, MFPGoTrigger4

[Program]

MFPLINK

MFPVsGetNAlignUpFixCamT

[Function]

The position to which the robot moves is calculated based on the imaging results of a specified vision sensor (fixed downward-facing camera).

This Function is used to take one image of the workpiece with one vision sensor for alignment adjustments.

[Syntax]

```
<Position variable>= MFPVsGetNAlignUpFixCamT(MVsNum, MCalibNum, MVsRsultNum,
                                             PVsTrgPos, ByRef MErrStt)
```

[Terminology]

<MVsNum>	Specify a vision number. Any value from 1 to 4 can be specified.
<MCalibNum>	Specify a calibration number. Any value from 1 to 8 can be specified. Ensure that the calibration number is the same value as the vision number.
<MVsRsultNum>	Specify imaging results of the vision sensor to be used for alignment adjustments. Any value from 1 to MFound 1 (argument of MFPVision1) can be specified.
<PVsTrgPos>	Specify the imaging position.
<MErrStt>	The error status is substituted. [Call by reference] 1: No error -1: The calibration number "MCalibNum" is specified incorrectly. -3: The vision sensor imaging results "MVsRsultNum" is specified incorrectly.

[Reference program]

```
1 MErrSts = MFPVsActivation(MVisionNo1)              'Vision sensor activation
2 MFPGoTrigger1(PVsTrg1)                             'Move to the imaging position.
3 MFPVision1(MVisionNo1, PWkNum.X, MVsFound1, PVSDATA1, MErrSts)
4 PPICK = MFPVsGetNAlignUpFixCamT(MVisionNo1, MCalibNo1, MCount, PVsTrg1, MErrSts)
                                    'Based on the imaging result (PVSDATA1), calculate the
                                    position to which the robot moves.
```

[Reference]

MFPVision1

[Program]

MSPVISION

MFPVsGetNAlignHandCamT

[Function]

The position to which the robot moves is calculated based on the imaging results of a specified vision sensor (hand camera).

This Function is used to take one image of the workpiece with one vision sensor for alignment adjustments.

[Syntax]

```
<Position variable>= MFPVsGetNAlignHandCamT(MVsNum, MCalibNum, MVsRsultNum,
                                            PVsTrgPos, ByRef MErrStt)
```

[Terminology]

<MVsNum> Specify a vision number. Any value from 1 to 4 can be specified.

<MCalibNum> Specify a calibration number. Any value from 1 to 8 can be specified. Ensure that the calibration number is the same value as the vision number.

<MVsRsultNum> Specify imaging results of the vision sensor to be used for alignment adjustments. Any value from 1 to MFound 1 (argument of MFPVision1) can be specified.

<PVsTrgPos> Specify the imaging position.

<MErrStt> The error status is substituted. [Call by reference]
1: No error
-1: The calibration number "MCalibNum" is specified incorrectly.
-3: The vision sensor imaging results "MVsRsultNum" is specified incorrectly.

[Reference program]

```
1 MErrSts = MFPVsActivation(MVisionNo1)                'Vision sensor activation
2 MFPGoTrigger1(PVsTrg1)                               'Move to the imaging position.
3 MFPVision1(MVisionNo1, PWkNum.X, MVsFound1, PVSDATA1, MErrSts)
4 PPICK = MFPVsGetNAlignHandCamT(MVisionNo1, MCalibNo1, MCount, PVsTrg1, MErrSts)
                                    'Based on the imaging result (PVSDATA1), calculate the
                                    position to which the robot moves.
```

[Reference]

MFPVision1

[Program]

MSPVISION

MFPVsGetNAlignLowFixCamT

[Function]

The position to which the robot moves is calculated based on the imaging results of a specified vision sensor (fixed upward-facing camera).

This Function is used to take one image of the workpiece with one vision sensor for alignment adjustments.

[Syntax]

```
<Position variable>= MFPVsGetNAlignLowFixCamT(MVsNum, MCalibNum,
                                              MVsRsultNum, POperation, ByRef
                                              MErrStt)
```

[Terminology]

<MVsNum> Specify a vision number. Any value from 1 to 4 can be specified.
<MCalibNum> Specify a calibration number. Any value from 1 to 8 can be specified.
Ensure that the calibration number is the same value as the vision number.
<MVsRsultNum> Specify imaging results of the vision sensor to be used for alignment adjustments.
Any value from 1 to MFound 1 (argument of MFPVision1) can be specified.
<POperation> Specify a position to adjust alignment.
<MErrStt> The error status is substituted. [Call by reference]
1: No error
-1: The calibration number "MCalibNum" is specified incorrectly.
-3: The vision sensor imaging results "MVsRsultNum" is specified incorrectly.

[Reference program]

```
1 MErrSts = MFPVsActivation(MVisionNo1)                'Vision sensor activation
2 MFPGoTrigger1(PVsTrg1)                               'Move to the first imaging position.
3 MFPVision1(MVisionNo1, PWkNum.X, MVsFound1, PVSDATA1, MErrSts)
4 PPLACE = MFPVsGetNAlignLowFixCamT(MVisionNo1, MCalibNo1, 1, PPLACETMP, MErrStt)
                                     'Based on the imaging result (PVSDATA1), calculate the
                                     position to which the robot moves.
```

[Reference]

MFPVision1

[Program]

MSPVISION

MFPVsAlignHandCamT2P

[Function]

The position to which the robot moves is calculated based on the imaging results of a specified vision sensor (hand camera).

This Function is used to take two images of the workpiece with one or two vision sensors for alignment adjustments.

[Syntax]

```
<Position variable>= MFPVsAlignHandCamT2P(MVsNum, MCalibNum, PVSDATA,
                                          PVsTrgPos, MVsNum2, MCalibNum2,
                                          PVSDATA2, PVsTrgPos2, ByRef MErrStt)
```

[Terminology]

<MVsNum> Specify the vision number of the vision sensor that takes the first image.
Any value from 1 to 4 can be specified.

<MCalibNum> Specify the calibration number of the vision sensor that takes the first image.
Any value from 1 to 8 can be specified.
Ensure that the calibration number is the same value as the vision number.

<PVSDATA> Specify the position of the workpiece recognized by the vision sensor at the first position.

<PVsTrgPos> Specify the first imaging position.

<MVsNum2> Specify the vision number of the vision sensor that takes the second image.
Any value from 1 to 4 can be specified.

<MCalibNum2> Specify the calibration number of the vision sensor that takes the second image.
Any value from 1 to 8 can be specified.
Ensure that the calibration number is the same value as the vision number.

<PVSDATA2> Specify the position of the workpiece recognized by the vision sensor at the second position.

<PVsTrgPos2> Specify the second imaging position.

<MErrStt> The error status is substituted. [Call by reference]
0: No error
-1: The calibration number is specified incorrectly.

[Reference program]

```
1 MErrSts = MFPVsActivation2P(MVisionNo1, MVisionNo2)    'Vision sensor activation
2 MFPGoTrigger3(PVsTrg3)                                  'Move to the first imaging position.
3 MFPVision2C2P(MVisionNo2, 1, PWkNum.X, MVsFound3, PVSDATA3, MErrStt3)
4 MFPGoTrigger4(PVsTrg4)                                  'Move to the second imaging position.
5 MFPVision2C2P(MVisionNo2, 2, PWkNum.X, MVsFound4, PVSDATA4, MErrStt4)
6 PPLACETMP = MFPVsAlignHandCamT2P(MVisionNo2, MCalibNo2, PVSDATA3, PVsTrg3,
                                   MVisionNo2, MCalibNo2, PVSDATA4, PVsTrg4,
                                   MErrStt)
                                   'Based on the imaging results (PVSDATA3, PVSDATA4),
                                   calculate the position to which the robot moves.
```

[Reference]

MFPVision1C2P, MFPVision2C2P, MFPVision12C2P, MFPVision34C2P

[Program]

MFPVSALIGN

MFPVsAlignLowFixCamT2P

[Function]

The position to which the robot moves is calculated based on the imaging results of a specified vision sensor (fixed upward-facing camera).

This Function is used to take two images of the workpiece with one or two vision sensors for alignment adjustments.

[Syntax]

```
<<Position variable>= MFPVsAlignLowFixCamT2P(MVsNum, MCalibNum, PVSDATA,
                                             MVsNum2, MCalibNum2, PVSDATA2,
                                             POperation, ByRef MErrStt)
```

[Terminology]

<MVsNum>	Specify the vision number of the vision sensor that takes the first image. Any value from 1 to 4 can be specified.
<MCalibNum>	Specify the calibration number of the vision sensor that takes the first image. Any value from 1 to 8 can be specified. Ensure that the calibration number is the same value as the vision number.
<PVSDATA>	Specify the position of the workpiece recognized by the vision sensor at the first position.
<MVsNum2>	Specify the vision number of the vision sensor that takes the second image. Any value from 1 to 4 can be specified.
<MCalibNum2>	Specify the calibration number of the vision sensor that takes the second image. Any value from 1 to 8 can be specified. Ensure that the calibration number is the same value as the vision number.
<PVSDATA2>	Specify the position of the workpiece recognized by the vision sensor at the second position.
<POperation>	Specify a position to adjust alignment.
<MErrStt>	The error status is substituted. [Call by reference] 0: No error -1: The calibration number is specified incorrectly.

[Reference program]

```
1 MErrSts = MFPVsActivation2P(MVisionNo1, MVisionNo2)    'Vision sensor activation
2 MFPGoTrigger1(PVsTrg1)                                  'Move to the first imaging position.
3 MFPVision1C2P(MVisionNo1, 1, PWkNum.X, MVsFound1, PVSDATA1, MErrStt1)
4 MFPGoTrigger2(PVsTrg2)                                  'Move to the second imaging position.
5 MFPVision1C2P(MVisionNo1, 2, PWkNum.X, MVsFound2, PVSDATA2, MErrStt2)
6 PPLACE = MFPVsAlignLowFixCamT2P(MVisionNo1, MCalibNo1, PVSDATA1, MVisionNo1,
                                  MCalibNo1, PVSDATA2, PPLACETMP, MErrStt)
                                  'Based on the imaging results (PVSDATA1, PVSDATA2),
                                  calculate the position to which the robot moves.
```

[Reference]

MFPVision1C2P, MFPVision2C2P, MFPVision12C2P, MFPVision34C2P

[Program]

MFPVSALIGN

MFPInitialization

[Function]

The robot moves according to the settings configured in "Teach Start Position".
It moves to the destination point PHome according to the operation parameter PInitPrm, and opens or closes the hand.

[Syntax]

```
<No return value>= MFPInitialization(PDest)
```

[Terminology]

<PDest> Specify the position taught in "Teach Start Position".
A position other than the taught position can also be specified.

[Reference program]

1 MSP_GetWorkData(PWkNum.X)	'Substitute the position variable value of the robot program WK* (* represents the value of PWkNum.X) for the user-defined external variable.
2 PHome = P_Home	'Substitute the user-defined external variable value for the local variable.
3 MFPInitialization(PHome)	'Move the robot according to the settings.

[Reference]

MSP_GetWorkData

[Related variable]

PHome, PInitPrm

[Program]

MFPLINK

MFPGoTrigger1

[Function]

The robot moves according to the settings configured in "Register Trigger Pos.".
It moves to the destination point PVsTrg1 according to the operation parameter PVsTrg1Prm, and opens or closes the hand.

[Syntax]

```
<No return value>= MFPGoTrigger1(PDest)
```

[Terminology]

<PDest>	Specify the position taught in "Register Trigger Pos.". A position other than the taught position can also be specified.

[Reference program]

1 MSP_GetWorkData(PWkNum.X)	'Substitute the position variable value of the robot program WK* (* represents the value of PWkNum.X) for the user-defined external variable.
2 PVsTrg1 = P_VsTrg1	'Substitute the user-defined external variable value for the local variable.
3 MFPGoTrigger1(PVsTrg1)	'Move the robot according to the settings.

[Reference]

MSP_GetWorkData

[Related variable]

PVsTrg1, PVsTrg1Prm

[Program]

MFPLINK

MFPGoTrigger2

[Function]

The robot moves according to the settings configured in "Register Trigger Pos.".
It moves to the destination point PVsTrg2 according to the operation parameter PVsTrg2Prm, and opens or closes the hand.

[Syntax]

```
<No return value>= MFPGoTrigger2(PDest)
```

[Terminology]

<PDest> Specify the position taught in "Register Trigger Pos.".
A position other than the taught position can also be specified.

[Reference program]

1 MSP_GetWorkData(PWkNum.X)	'Substitute the position variable value of the robot program WK* (* represents the value of PWkNum.X) for the user-defined external variable.
2 PVsTrg2 = P_VsTrg2	'Substitute the user-defined external variable value for the local variable.
3 MFPGoTrigger2(PVsTrg2)	'Move the robot according to the settings.

[Reference]

MSP_GetWorkData

[Related variable]

PVsTrg2, PVsTrg2Prm

[Program]

MFPLINK

MFPGoTrigger3

[Function]

The robot moves according to the settings configured in "Register Trigger Pos.".
It moves to the destination point PVsTrg3 according to the operation parameter PVsTrg3Prm, and opens or closes the hand.

[Syntax]

```
<No return value>= MFPGoTrigger3(PDest)
```

[Terminology]

<PDest> Specify the position taught in "Register Trigger Pos.".
A position other than the taught position can also be specified.

[Reference program]

1 MSP_GetWorkData(PWkNum.X)	'Substitute the position variable value of the robot program WK* (* represents the value of PWkNum.X) for the user-defined external variable.
2 PVsTrg3 = P_VsTrg3	'Substitute the user-defined external variable value for the local variable.
3 MFPGoTrigger3(PVsTrg3)	'Move the robot according to the settings.

[Reference]

MSP_GetWorkData

[Related variable]

PVsTrg3, PVsTrg3Prm

[Program]

MFPLINK

MFPGoTrigger4

[Function]

The robot moves according to the settings configured in "Register Trigger Pos.".
It moves to the destination point PVsTrg4 according to the operation parameter PVsTrg4Prm, and opens or closes the hand.

[Syntax]

```
<No return value>= MFPGoTrigger4(PDest)
```

[Terminology]

<PDest> Specify the position taught in "Register Trigger Pos.".
A position other than the taught position can also be specified.

[Reference program]

1 MSP_GetWorkData(PWkNum.X)	Substitute the position variable value of the robot program WK* (* represents the value of PWkNum.X) for the user-defined external variable.
2 PVsTrg4 = P_VsTrg4	Substitute the user-defined external variable value for the local variable.
3 MFPGoTrigger4(PVsTrg4)	Move the robot according to the settings.

[Reference]

MSP_GetWorkData

[Related variable]

PVsTrg4, PVsTrg4Prm

[Program]

MFPLINK

MFPBeforeGet

[Function]
The robot moves according to the settings configured in "Waiting Pos. to Pick".
It moves to the destination point PGBack according to the operation parameter PGBackPrm, and opens or closes the hand.

[Syntax]

```
<No return value>= MFPBeforeGet(PDest)
```

[Terminology]

<PDest> Specify the position taught in "Waiting Pos. to Pick".
A position other than the taught position can also be specified.

[Reference program]

1 MSP_GetWorkData(PWkNum.X)	Substitute the position variable value of the robot program WK* (* represents the value of PWkNum.X) for the user-defined external variable.
2 PGetBk = P_Get*P_GBack	Substitute the user-defined external variable value for the local variable.
3 MFPBeforeGet(PGetBk)	Move the robot according to the settings.

[Reference]
MSP_GetWorkData

[Related variable]
PGBack, PGBackPrm

[Program]
MFPLINK

MFPGet

[Function]

The robot moves according to the settings configured in "Position to Pick".
It moves to the destination point PGet according to the operation parameter PGetPrm, and opens or closes the hand.

[Syntax]

```
<No return value>= MFPGet(PDest)
```

[Terminology]

<PDest> Specify the variable of the position taught in "Position to Pick".
A position other than the taught position can also be specified.

[Reference program]

1 MSP_GetWorkData(PWkNum.X)	Substitute the position variable value of the robot program WK* (* represents the value of PWkNum.X) for the user-defined external variable.
2 PGet = P_Get	Substitute the user-defined external variable value for the local variable.
3 MFPGet(PGet)	Move the robot according to the settings.

[Reference]

MSP_GetWorkData

[Related variable]

PGet, PGetPrm

[Program]

MFPLINK

MFPAfterGet

[Function]

The robot moves according to the settings configured in "Waiting Pos. after Pick".
It moves to the destination point PGNext according to the operation parameter PGNextPrm, and opens or closes the hand.

[Syntax]

<No return value>= MFPAfterGet(PDest)

[Terminology]

<PDest>	Specify the position taught in "Waiting Pos. after Pick". A position other than the taught position can also be specified.

[Reference program]

1 MSP_GetWorkData(PWkNum.X)	Substitute the position variable value of the robot program WK* (* represents the value of PWkNum.X) for the user-defined external variable.
2 PGetUp = P_Get*P_GNext	Substitute the user-defined external variable value for the local variable.
3 MFPAfterGet(PGetUp)	Move the robot according to the settings.

[Reference]

MSP_GetWorkData

[Related variable]

PGNext, PGNextPrm

[Program]

MFPLINK

MFPBeforePut

[Function]

The robot moves according to the settings configured in "Waiting Pos. to Place".
It moves to the destination point PPBack according to the operation parameter PPBackPrm, and opens or closes the hand.

[Syntax]

```
<No return value>= MFPBeforePut(PDest)
```

[Terminology]

<PDest>	Specify the position taught in "Waiting Pos. to Place". A position other than the taught position can also be specified.

[Reference program]

1 MSP_GetWorkData(PWkNum.X)	Substitute the position variable value of the robot program WK* (* represents the value of PWkNum.X) for the user-defined external variable.
2 PPutBk = P_Put*P_PBack	Substitute the user-defined external variable value for the local variable.
3 MFPBeforePut(PPutBk)	Move the robot according to the settings.

[Reference]

MSP_GetWorkData

[Related variable]

PPBack, PPBackPrm

[Program]

MFPLINK

MFPPut

[Function]

The robot moves according to the settings configured in "Position to Place".
It moves to the destination point PPut according to the operation parameter PPutPrm, and opens or closes the hand.

[Syntax]

```
<No return value>= MFPPut(PDest)
```

[Terminology]

<PDest> Specify the variable of the position taught in "Position to Place".
A position other than the taught position can also be specified.

[Reference program]

1 MSP_GetWorkData(PWkNum.X)	Substitute the position variable value of the robot program WK* (* represents the value of PWkNum.X) for the user-defined external variable.
2 PPut = P_Put	Substitute the user-defined external variable value for the local variable.
3 MFPPut(PPut)	Move the robot according to the settings.

[Reference]

MSP_GetWorkData

[Related variable]

PPut, PPutPrm

[Program]

MFPLINK

MFPAfterPut

[Function]

The robot moves according to the settings configured in "Waiting Pos. after Place".
It moves to the destination point PPNext according to the operation parameter PPNextPrm, and opens or closes the hand.

[Syntax]

```
<No return value>= MFPAfterPut(PDest)
```

[Terminology]

<PDest> Specify the position taught in "Waiting Pos. after Place".
A position other than the taught position can also be specified.

[Reference program]

1 MSP_GetWorkData(PWkNum.X)	Substitute the position variable value of the robot program WK* (* represents the value of PWkNum.X) for the user-defined external variable.
2 PPutUp = P_Put*P_PNext	Substitute the user-defined external variable value for the local variable.
3 MFPAfterPut(PPutUp)	Move the robot according to the settings.

[Reference]

MSP_GetWorkData

[Related variable]

PPNext, PPNextPrm

[Program]

MFPLINK

MFPGoHome

[Function]

The robot moves according to the settings configured in "Move End Position".
It moves to the destination point PHome according to the operation parameter PHomePrm, and opens or closes the hand.

[Syntax]

```
<No return value>= MFPGoHome(PDest)
```

[Terminology]

<PDest>	Specify the position taught in "Teach Start Position". A position other than the taught position can also be specified.

[Reference program]

1 MSP_GetWorkData(PWkNum.X)	Substitute the position variable value of the robot program WK* (* represents the value of PWkNum.X) for the user-defined external variable.
2 PHome = P_Home	Substitute the user-defined external variable value for the local variable.
3 MFPGoHome(PHome)	Move the robot according to the settings.

[Reference]

MSP_GetWorkData

[Related variable]

PHome, PHomePrm

[Program]

MFPLINK

MSP_GetWorkData

[Function]

The values set on each screen (saved in the robot program WK*) are substituted for the user-defined external variable.

[Syntax]

<No return value>= MSP_GetWorkData(MModelNo)

[Terminology]

<MModelNo> Specify a workpiece type number. Any value from 1 to 5 can be specified.

[Reference program]

1 MSP_GetWorkData(PWkNum.X)	Substitute the position variable value of the robot program WK* (* represents the value of PWkNum.X) for the user-defined external variable.
2 PHome = P_Home	Substitute the user-defined external variable value for the local variable.
3 MFPGoHome(PHome)	Move the robot according to the settings.

[Reference]

MSP_GetWorkData

[Related variable]

PWkNum

[Program]

MSPCOMMON

MSP_Mov_Hand

[Function]

The robot moves to the destination point using joint interpolation and opens or closes a specified pneumatic hand.

[Syntax]

```
<No return value>= MSP_Mov_Hand(ByVal MOvrd, ByVal MFine, ByVal PGoPos, ByVal
                                MDlyTime, ByVal MHandNo, ByVal MSwitch, ByVal
                                MHandDlyTime, ByVal MType1)
```

[Terminology]

<MOvrd>	pecify an override value. Any value from 1 to 100 can be specified.
<MFine>	pecify the linear distance for the FineP command. The value that can be specified for the FineP command can be used.
<PGoPos>	pecify the destination point.
<MDlyTime>	pecify the time from when the robot starts moving to the destination point until when the hand starts opening or closing. The value that can be specified for the Dly command can be used.
<MHandNo>	pecify a hand number. The value that can be specified for the HOpen/HClose command can be used.
<MSwitch>	pecify whether to open or close the hand. Specifying 0 will execute the HClose command. Specifying 1 will execute the HOpen command.
<MHandDlyTime>	pecify the time from when the hand starts opening or closing until when this Function is complete. The value that can be specified for the Dly command can be used.
<MType1>	pecify roundabout operation or shortcut operation. The value that can be specified for TYPE numeric constant 1 of the Mov command can be used.

[Reference program]

```
1 MSP_Mov_Hand(MOvrd, MFine, PGoPos, MDlyTime, MHandNo, MSwitch, MHandDlyTime,
MType1)
```

[Program]

MSPCOMMON

MSP_Mvs_Hand

[Function]

The robot moves to the destination point using linear interpolation and opens or closes a specified pneumatic hand.

[Syntax]

```
<No return value>= MSP_Mvs_Hand(ByVal MOvrd, ByVal MFine, ByVal PGoPos, ByVal
                               MDlyTime, ByVal MHandNo, ByVal MSwitch, ByVal
                               MHandDlyTime)
```

[Terminology]

<MOvrd> Specify an override value. Any value from 1 to 100 can be specified.

<MFine> Specify the linear distance for the FineP command. The value that can be specified for the FineP command can be used.

<PGoPos> Specify the destination point.

<MDlyTime> Specify the time from when the robot starts moving to the destination point until when the hand starts opening or closing.
The value that can be specified for the Dly command can be used.

<MHandNo> Specify a hand number. The value that can be specified for the HOpen/HClose command can be used.

<MSwitch> Specify whether to open or close the hand.
Specifying 0 will execute the HClose command.
Specifying 1 will execute the HOpen command.

<MHandDlyTime> Specify the time from when the hand starts opening or closing until when this Function is complete.
The value that can be specified for the Dly command can be used.

[Reference program]

```
1 MSP_Mvs_Hand(MOvrd, MFine, PGoPos, MDlyTime, MHandNo, MSwitch, MHandDlyTime)
```

[Program]

MSPCOMMON

MSP_Mov

[Function]

The robot moves to the destination point using joint interpolation.

[Syntax]

```
<No return value>= MSP_Mov (ByVal MOvrd, ByVal MFine, ByVal PGoPos, ByVal
                                        MDlyTime, ByVal MType1)
```

[Terminology]

<MOvrd>	Specify an override value. Any value from 1 to 100 can be specified.
<MFine>	Specify the linear distance for the FineP command. The value that can be specified for the FineP command can be used.
<PGoPos>	Specify the destination point.
<MDlyTime>	Specify the time from when the robot starts moving to the destination point until when this Function is complete. The value that can be specified for the Dly command can be used.
<MType1>	Specify roundabout operation or shortcut operation. The value that can be specified for TYPE numeric constant 1 of the Mov command can be used.

[Reference program]

```
1 MSP_Mov (MOvrd, MFine, PGoPos, MDlyTime, MType1)
```

[Program]

MSPCOMMON

MSP_Mvs

[Function]

The robot moves to the destination point using linear interpolation.

[Syntax]

```
<No return value>= MSP_Mvs (ByVal MOvrd, ByVal MFine, ByVal PGoPos, ByVal
                                        MDlyTime)
```

[Terminology]

<MOvrd> Specify an override value. Any value from 1 to 100 can be specified.

<MFine> Specify the linear distance for the FineP command. The value that can be specified for the FineP command can be used.

<PGoPos> Specify the destination point.

<MDlyTime> Specify the time from when the robot starts moving to the destination point until when this Function is complete.
The value that can be specified for the Dly command can be used.

[Reference program]

1 MSP_Mvs (MOvrd, MFine, PGoPos, MDlyTime)

[Program]

MSPCOMMON

MSP_Hand

[Function]

A specified hand opens or closes.

[Syntax]

<No return value>= MSP_Hand(ByVal MHandNo, ByVal MSwitch, ByVal MDlyTime)

[Terminology]

<MHandNo> Specify a hand number. The value that can be specified for the HOpen/HClose command can be used.

<MSwitch> Specify whether to open or close the hand.
Specifying 0 will execute the HClose command.
Specifying 1 will execute the HOpen command.

<MDlyTime> Specify the time from when the hand starts opening or closing until when this Function is complete.
The value that can be specified for the Dly command can be used.

[Reference program]

```
1 MSP_Hand(MHandNo, MSwitch, MDlyTime)
```

[Program]

MSPCOMMON

15. MAINTENANCE

When the wizard of the vision sensor enhancement function finishes, the following three types of data are created.

① Settings selected or entered in the wizard (Refer to "15.1 Exporting and importing settings".)

The settings set in the wizard are temporarily saved in the "2DVisionSetting.XML" file.
You can name the file as you like.

② Robot program and parameters (Refer to "15.2 Backing up and restoring robot programs and parameters".)

The robot program shown in "14.1 List of robot programs" is created in the robot controller according to the application selected.
The parameters shown in "16 PARAMETER SPECIFICATIONS" are automatically set. You can save "robot programs" and "parameters" in the folder with your desired name.

③ Vision program (Refer to "15.3 Backing up and restoring vision programs".)

A vision program is created with a name specified in the Model name field of the Register a Part screen.
If a name is specified as shown below, a vision program named with "Model1.job" is created in the vision sensor.

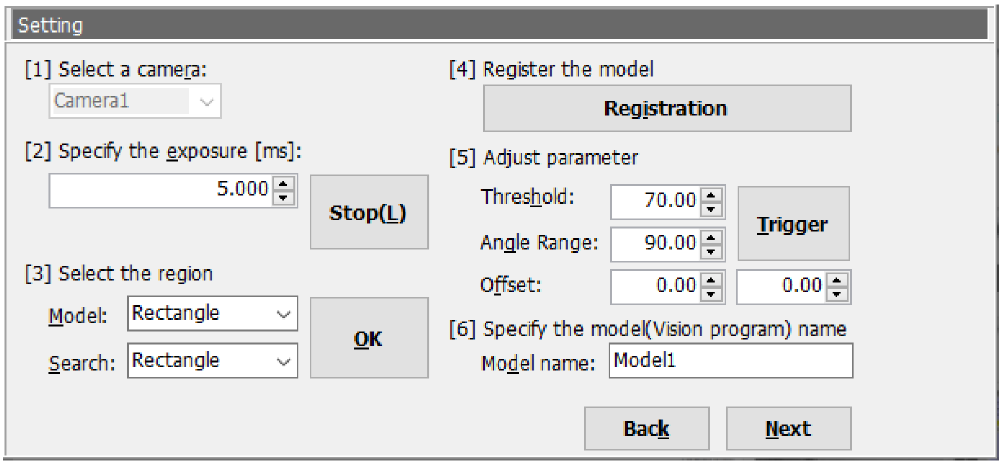


Fig. 15-1 Register a Part screen

All information saved in the vision sensor including this vision program can be saved.
A folder is automatically named only for vision sensor backup.

To avoid accidental data corruption and deletion, the above three types of data are recommended to be backed up.

15.1. Exporting and importing settings

All the data set in the wizard of this function are saved in the "2DVisionSetting.XML" file located in the folder for RT ToolBox3 workspace.
When the wizard or RT ToolBox3 finishes, a confirmation window asking to export the setting file will appear. To export the file, click the [Yes] button and specify a destination folder and file name.

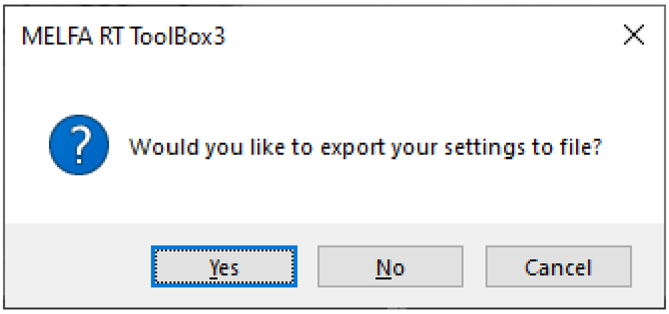
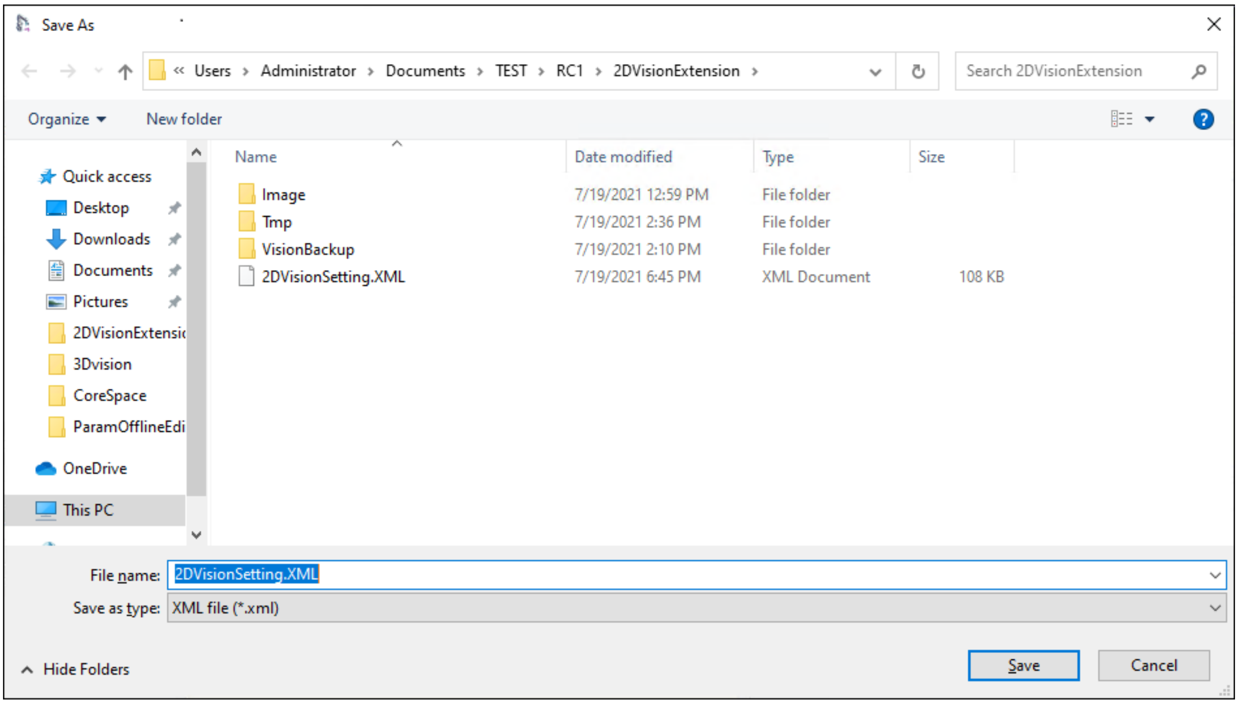


Fig. 15-2 Exporting the enhancement function setting file

Clicking the [Yes] button will display a file dialog. The "2DVisionExtension" folder of the project folder is selected as the default setting.



Clicking the [Save] button will display a confirmation message whether to back up the vision sensor settings "15.3 Backing up and restoring vision programs".
Clicking the [Cancel] button will return to the previous screen.

Clicking the [No] button in "Fig. 15-2 Exporting the enhancement function setting file" will close the wizard. (A message asking to back up the vision sensor settings will not appear.)

Clicking the [Cancel] button will return to the previous screen.

To restore exported data, click the [Import] button on the start screen.
For further information, refer to "5.6 Start screen".

15.2. Backing up and restoring robot programs and parameters

To back up all data including created robot programs and changed parameters with the help of this function, use the backup function of RT ToolBox3.
For information on how to use the backup function, refer to "Backup (robot → computer)" in the "RT ToolBox3 / RT ToolBox3 mini User's Manual (BFP-A3495)".

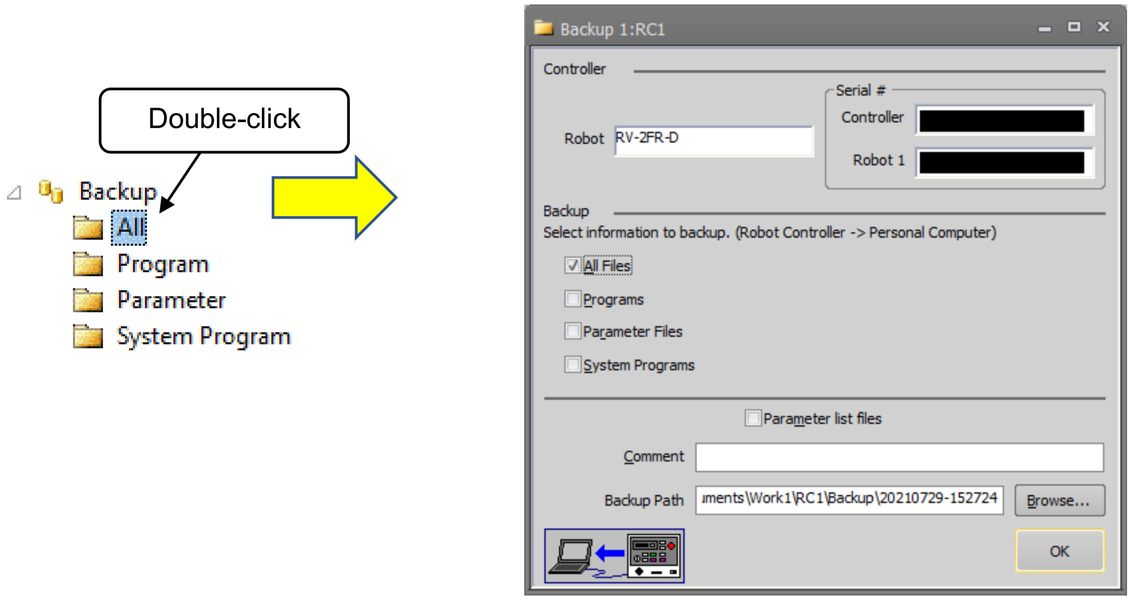


Fig. 15-3 RT ToolBox3 "Backup"

To restore the settings and all data, refer to "Restore (PC -> Robot)" in the "RT ToolBox3 / RT ToolBox3 mini User's Manual (BFP-A3495)".

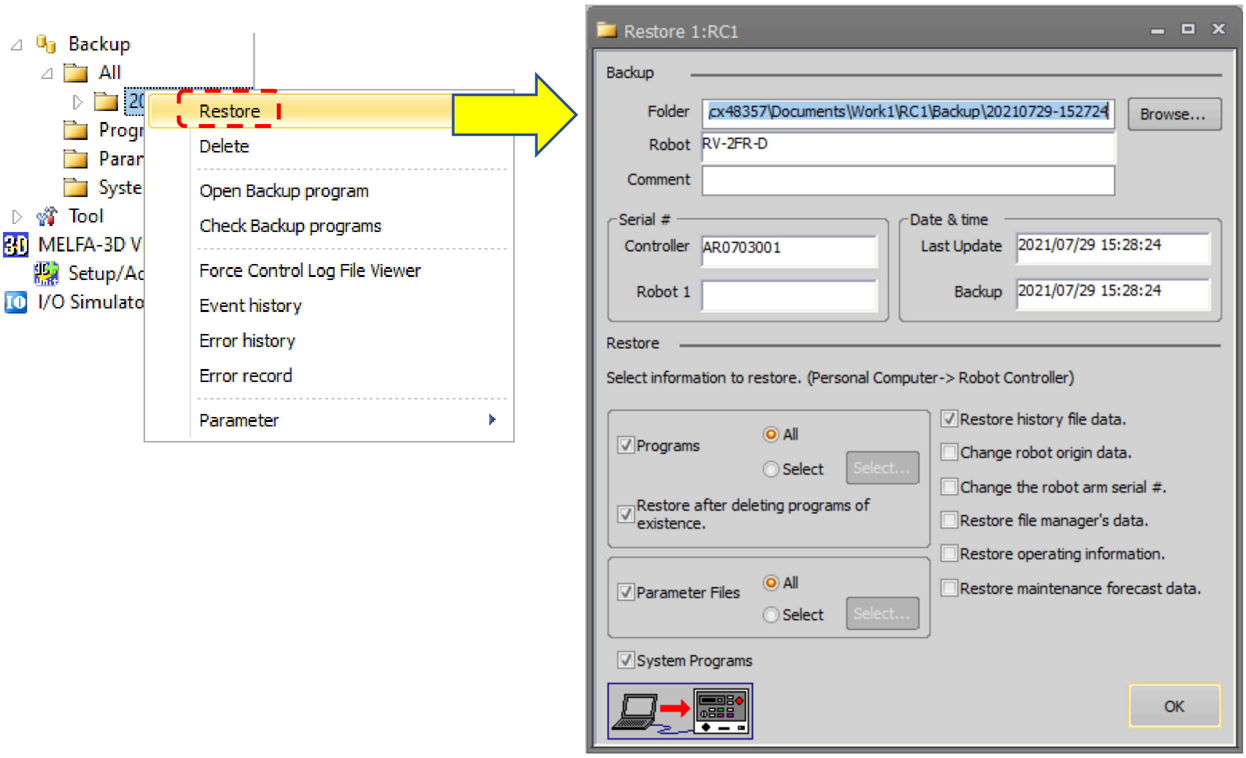


Fig. 15-4 RT ToolBox3 "Restore"

15.3. Backing up and restoring vision programs

There are two methods of backing up vision programs created using this function.

1) Click the [Backup] button displayed on the right of the vision sensor image on the Config. Communication screen shown in "7 CONFIGURING THE VISION SENSOR COMMUNICATION SETTINGS".

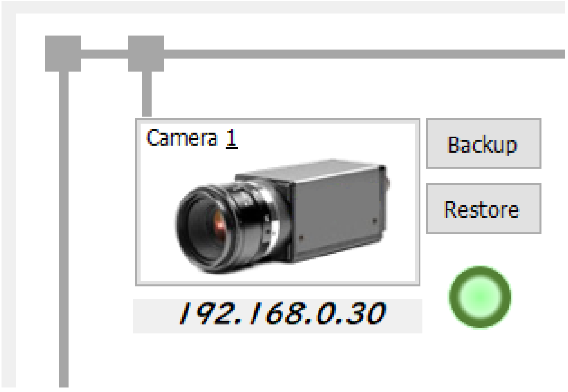


Fig. 15-5 Backup and restore

Clicking the [Backup] button will display a confirmation message.

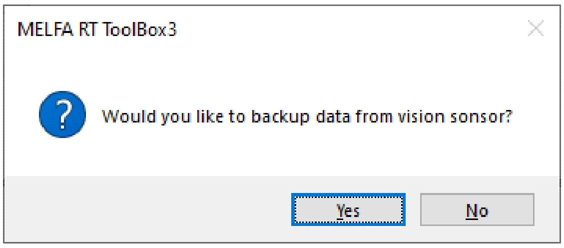


Fig. 15-6 Backup confirmation message

Clicking the [Yes] button will start backup. When the backup is complete, a completion message will appear. Clicking the [No] button will return to the previous screen. For a message asking to finish the wizard, the wizard closes.

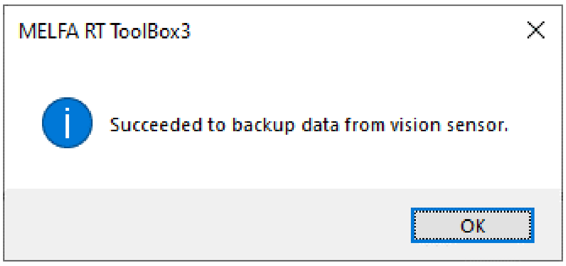


Fig. 15-7 Backup completion message

② When RT ToolBox3 finishes, an export confirmation window shown in "15.1 Exporting and importing settings" appears, and the backup confirmation message shown above appears. The program can be backed up at the end of the application.

Backup data is saved in the "VisionBackup" folder that is created in the "2DVisionExtension" folder of the project folder.
Structure of the save folder:
[00 Note 1] - [Vision sensor model_6-digit MAC address.Number of backup times starting from 000]
Note 1: [01] for camera number 2, [02] for camera number 3, and [03] for camera number 4

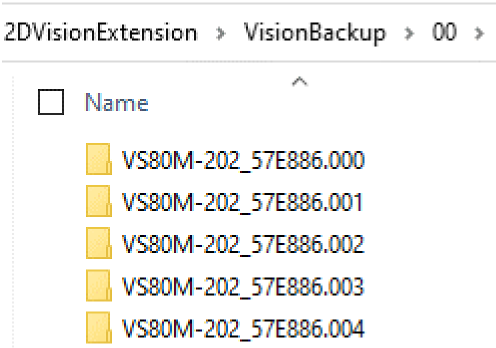


Fig. 15-8 Structure of the backup folder

To restore data to the vision sensor, click the [Restore] button displayed on the right of the vision sensor image on the Config. Communication screen.
Select a folder for restore from the displayed Browse for Folder window, then click the [OK] button.

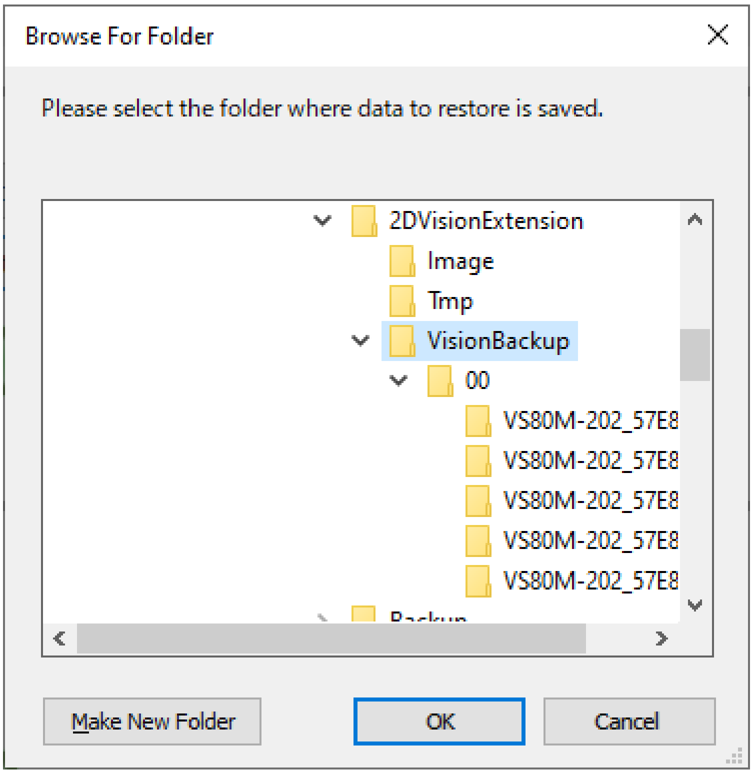


Fig. 15-9 Browse for Folder window for restore

When restore is complete, the following message will appear.

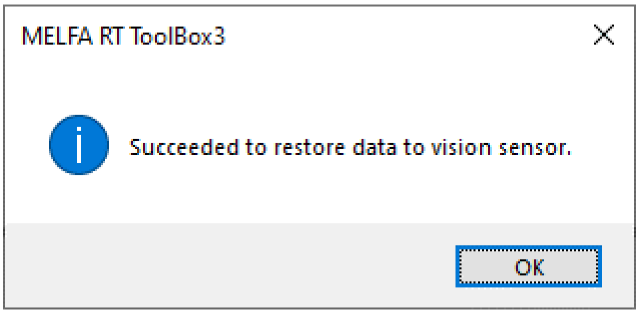


Fig. 15-10 Restore completion message

16. PARAMETER SPECIFICATIONS

16.1. List of related parameters

The following shows the parameters that are changed in each step of the vision sensor enhancement function.

Table 16-1 List of related parameters

Parameter	Parameter name	Number of arrays Number of characters	Description	Before setting	After change
User base program	PRGUSR	String 1	Specify the name of the user base program.	""	"BASE"
Device assignment	COMDEV	Element 2 Element 3 Element 4 Element 5	Specify COM2 to COM5 devices in any of OPT11 to OPT19 according to the number of vision sensors.	"" "" "" ""	"OPT12" "OPT13" "OPT14" "OPT15"
Server specification	NETMODE	Element 2 Element 3 Element 4 Element 5	Specify a server. 0: Client, 1: Server	1	0
Communication destination IP address	NETHSTIP	Element 2 Element 3 Element 4 Element 5	Specify the IP address of the communication destination device.	""	On-screen setting value
Port No.	NETPORT	Element 3 Element 4 Element 5 Element 6	Specify a communication port number.	10002 10003 10004 10005	23 23 23 23
Protocol	CPRCE12 CPRCE13 CPRCE14 CPRCE15	Integer 1	Specify a communication protocol. 0: Without procedure 1: With procedure 2: Data link	0	2
End code	CTERME12 CTERME13 CTERME14 CTERME15	Integer 1	Specify the end code of the communication message. 0:CR / 1:CR+LF * If an end code does not exist, specify it with parameter "NETTERM".	0	1
Trigger timing	NVTRGTMG	Integer 1	Specify the trigger timing when the NvRun or NvTrg command is executed.	2	1
Job load time-out time	NVJBTOUT	Integer 1	Set the job load time-out time of the network vision. (Unit: seconds, Setting range: 1 to 32767)	90	90
Vision sensor calibration data	VSCALB1 to 8	Integer 12	Clicking the [Calibration Execution] button will make the robot perform calibration automatically, and set the calibration data automatically. Clicking the [Recalculation] button in the results window will also set data automatically.	0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0	Changes depending on calibration and whether the [Recalculation] button is clicked.

Parameter	Parameter name	Number of arrays Number of characters	Description	Before setting	After change
Using both other program reading commands and Function procedure	FUNCSPEC	Integer 1	Set whether to use other program reading commands (CallP, XLoad, XRun, #Include) together. Changing the value to "1" may slightly increase takt time when other program reading commands are used. 0: Not used 1: Used	0	1
Unit setting of position data rotational components	PRGMDEG	Integer 1	Specify the unit of position data rotational components in the robot program. 0: RAD 1: DEG [CAUTION] This function transfers robot programs to the robot controller. The programs are created assuming that the parameter is the initial value (RAD). Do not change the parameter.	0	0

17. TROUBLESHOOTING

17.1. Error details of the vision sensor enhancement function

The errors of the vision sensor enhancement function are displayed as error messages in RT ToolBox3. If an error occurs, refer to the error list shown below, and follow the solutions.

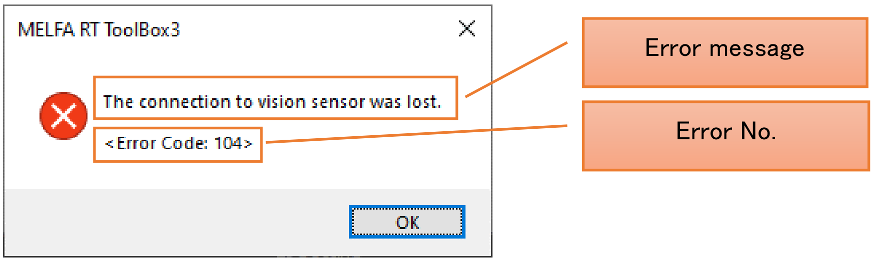


Fig. 17-1 Error message and error number

Table 17-1 [General errors (from 1)] list

Error No.	Causes and solutions	
2	Error message	The network adapter was not found.
	Cause	A network adapter does not exist in the device being used.
	Solution	Use items such as a USB-LAN converter to add a network adapter.
4	Error message	Selected file does not work correctly.
	Cause	The selected import file has been edited.
	Solution	Select a proper file.
5	Error message	Failed to export the settings to XML file.
	Cause	1) Rights of access to the export destination are not given. 2) The drive does not have enough storage space. 3) The export destination does not exist.
	Solution	1) Give rights of access to the export destination. 2) Provide enough storage space. 3) Check whether the export destination exists.

Table 17-2 [Vision errors (from 100)] list

<table>
<tr><th>Error No.</th><th colspan="2">Causes and solutions</th></tr>
<tr><td rowspan="3">100</td><td>Error message</td><td>The vision sensor was not found.</td></tr>
<tr><td>Cause</td><td>A vision sensor of the specified manufacturer is not found on the same network.</td></tr>
<tr><td>Solution</td><td>1) Check the wiring.
2) Check the settings of the Windows firewall.</td></tr>
<tr><td rowspan="3">101</td><td>Error message</td><td>Failed to write to the vision sensor.</td></tr>
<tr><td>Cause</td><td>1) The vision program is not loaded.
2) The vision sensor is in the processing state (Busy state).
3) In In-Sight Explorer, the vision program was switched to offline mode.
4) The setting values attempted to be written to the vision sensor are incorrect.</td></tr>
<tr><td>Solution</td><td>1), 2) Perform steps again.
(Example) To display a live image again: Click the [Back] button, then the [Next] button.
3) Power off and on the vision sensor.
4) Check if the setting values attempted to be written to the vision sensor are correct (such as IP addresses, subnet masks, and gateways).</td></tr>
<tr><td rowspan="3">102</td><td>Error message</td><td>Vision sensor backup failed.</td></tr>
<tr><td>Cause</td><td>1) A cable is disconnected.
2) The vision sensor is powered off.</td></tr>
<tr><td>Solution</td><td>1) Check the wiring.
2) Check if the vision sensor is powered on.</td></tr>
<tr><td rowspan="3">103</td><td>Error message</td><td>Vision sensor restore failed.</td></tr>
<tr><td>Cause</td><td>1) A cable is disconnected.
2) The vision sensor is powered off.</td></tr>
<tr><td>Solution</td><td>1) Check the wiring.
2) Check if the vision sensor is powered on.</td></tr>
<tr><td rowspan="3">104</td><td>Error message</td><td>The connection to vision sensor was lost.</td></tr>
<tr><td>Cause</td><td>1) A cable is disconnected.
2) The vision sensor is powered off.</td></tr>
<tr><td>Solution</td><td>1) Check the wiring.
2) Check if the vision sensor is powered on.</td></tr>
<tr><td rowspan="3">106</td><td>Error message</td><td>Communication with the vision sensor has been lost.</td></tr>
<tr><td>Cause</td><td>1) A cable is disconnected.
2) The vision sensor is powered off.
3) A time-out occurred.</td></tr>
<tr><td>Solution</td><td>1) Check the wiring.
2) Check if the vision sensor is powered on.
3) Restart the vision sensor.</td></tr>
<tr><td rowspan="3">107</td><td>Error message</td><td>Some destination positions for robot are out of FOV.</td></tr>
<tr><td>Cause</td><td>The value specified in "Movement Interval (Vertical)" or "Movement Interval (Horizontal)" is too large, and the point is displayed outside the field of view.</td></tr>
<tr><td>Solution</td><td>Review the values of "Movement Interval (Vertical)" and "Movement Interval (Horizontal)".</td></tr>
</table>

Table 17-3 [Robot controller errors (from 200)] list

Error No.	Causes and solutions	
200	Error message	An error occurred during the calibration operation.
	Cause	An error occurred during calibration.
	Solution	Table 17-5 List of errors output from the robot controllerCheck .
201	Error message	An error occurred at the position to pick a part up.
	Cause	An error occurred in the process of registering the grasp position.
	Solution	Table 17-5 List of errors output from the robot controllerCheck .
202	Error message	Failed to turn on the servo.
	Cause	The servos have failed to turn on within the specified time.
	Solution	Check that the primary power voltage is within the specification value. Contact the manufacturer if the servos still fail to turn on.
203	Error message	The specified program name is invalid.
	Cause	The specified program name includes unusable characters or reserved (*1) characters. (*1) Program name starting from "MSP" Program name starting from "MFP" Program name starting from "VS" "BASE" "2DVSTMP" "CALIBDATA" "MEASURE"
	Solution	Change the program name.
204	Error message	Transmission Error.
	Cause	1) A cable is disconnected. 2) The robot controller is powered off.
	Solution	1) Check the wiring. 2) Check if the robot controller is powered on.
206	Error message	Communication with the robot controller has been lost.
	Cause	1) A cable is disconnected. 2) The robot controller is powered off. 3) A time-out occurred.
	Solution	1) Check the wiring. 2) Check if the robot controller is powered on. 3) Restart the vision sensor.

Table 17-4 [System errors (from 900)] list

Error No.	Causes and solutions	
900	Error message	Failed to read the program variable.
	Cause	1) If this error occurs in the process of calibration or result checking, a robot program with a name starting from "A" does not include all or any of the variables "PCalibData", "PCalibRobo", and "PCalibVision". 2) If this error occurs in the process of setting a grasp position, a robot program with a name starting from "C" does not include the variable "PAuto".
	Solution	1) Check whether the program with a name starting from "A" includes the variables "PCalibData", "PCalibRobo", and "PCalibVision". If the variables do not exist, delete the program with a name starting from "A", go back to the communication setting screen, and reset the communication settings. The program with a name starting from "A" will be transferred again. 2) Check whether the program with a name starting from "C" includes the variable "PAuto". If the variable does not exist, delete the program with a name starting from "C", go back to the communication setting screen, and reset the communication settings. The program with a name starting from "C" will be transferred again.

17.2. List of errors related to the vision sensor enhancement function

The following shows the structure of the error number.

◇◆◇The meaning of the error number in the following table are shown below.◇◆◇

□ 0000 *

- ･An error marked with a * reset by turning the power OFF and ON. Take the measures given.
- ･The error type is indicated with a 4-digit number.
- ･Three types of error classes are indicated.

H: High level error	The servo turns OFF.
L: Low level error	The operation will stop.
C: Warning	The operation will continue.

Fig. 17-2 Error details

Table 17-5 List of errors output from the robot controller

(1) MELFA Smart Plus

Error No.	Causes and solutions	
L3780	Error description	Cannot use the MELFA Smart Plus.
	Cause	The MELFA Smart Plus card or MELFA Smart Plus card pack has not been inserted. If the MELFA Smart Plus card has been inserted, the value of the parameter SMART+1 is incorrect.
	Solution	Insert the MELFA Smart Plus card or MELFA Smart Plus card pack. Set the parameter SMART+1 correctly.
L3781	Error description	Cannot use the MELFA Smart Plus.
	Cause	The MELFA Smart Plus card or MELFA Smart Plus card pack has not been inserted. If the MELFA Smart Plus card has been inserted, the value of the parameter SMART+1 is incorrect.
	Solution	Insert the MELFA Smart Plus card or MELFA Smart Plus card pack. Set the parameter SMART+1 correctly.
L3782	Error description	There're MELFA Smart Plus Cards.
	Cause	Multiple MELFA Smart Plus cards are inserted.
	Solution	Power off the robot controller and remove an unnecessary MELFA Smart Plus card.

(2) Communication between the vision sensor and robot controller

Error No.	Causes and solutions	
L3142	Error description	The communication line can not be opened.
	Cause	The communication line between the vision sensor and robot controller cannot be opened.
	Solution	1) Check the communication cable or communication parameters. 2) Check that the communication settings have been configured correctly. Refer to "7 CONFIGURING THE VISION SENSOR COMMUNICATION SETTINGS".
L8610	Error description	The communication is abnormal.
	Cause	Communication between the vision sensor and robot controller is disconnected.
	Solution	1) Check the communication cable.
L8632	Error description	The vision is a time-out
	Cause	The vision sensor does not respond within the specified time.
	Solution	1) Check the value of the vision sensor communication time-out time (NVJBTOUT). (Recommended value: 90 seconds) 2) Restart the vision sensor.
L8650	Error description	Please make the vision online.
	Cause	The vision sensor is in offline mode. (Switching the mode to offline in In-Sight Explorer may trigger this error.)
	Solution	Restart the vision sensor. (If the mode has been switched to offline in In-Sight Explorer, switch the mode to online in In-Sight Explorer again.)

(3) During robot program running

Error No.	Causes and solutions	
L9100	Error description	Vision sensor activation has failed.
	Cause	A connection with the vision sensor has failed.
	Solution	1) Check the communication cable or communication parameters. 2) Check that the communication settings have been configured correctly. Refer to "7 CONFIGURING THE VISION SENSOR COMMUNICATION SETTINGS".
L9101	Error description	The registered workpiece (or mark) cannot be recognized.
	Cause	The registered workpiece (or mark) cannot be recognized.
	Solution	Review the settings of the vision sensor recognition parameters ("Threshold" and "Angle Range"). Refer to "9.3 Registering a workpiece (model)".
L9110	Error description	Calibration has failed. (mark recognition)
	Cause	Data required for calibration calculation has not been acquired.
	Solution	a) The recognition target may be outside the camera's field of view. Review the calibration conditions. Refer to "8.5 Performing calibration". b) The recognition target may not be recognized correctly. Review the settings of the vision sensor recognition parameters ("Threshold" and "Angle Range"). Refer to "9.3 Registering a workpiece (model)".
L9111	Error description	Calibration has failed. (calculation of the center of the field of view)
	Cause	The registered mark has failed to be moved to the center of image (within the convergence target value [mm]) within the specified number of retries.
	Solution	Review the conditions of calibration completion (Convergence Target Value and Convergence Retry Counts). Refer to "8.5 Performing calibration".
L9112	Error description	Calibration has failed. (calculation of parallelism)
	Cause	The robot has failed to move so that the registered mark and vision sensor are parallel to each other (within the convergence target value [deg]) within the specified number of retries.
	Solution	Review the conditions of calibration completion (Convergence Target Value and Convergence Retry Counts). Refer to "8.5 Performing calibration".

18. TIPS

The following table provides information on cases that may occur during operation and explains how to take countermeasures.

Table 18-1 Solutions

	Case	Cause	Solution
1	"Vision sensor enhancement" does not appear in the RT ToolBox3 workspace.	The function code of the MELFA Smart Plus card has not been set.	Set the function code as instructed in "5.2 Inserting the MELFA Smart Plus card".
		The version of RT ToolBox3 is old.	Refer to "3.3 Software version". The latest version can be downloaded from the FA website.
2	Communication with the vision sensor cannot be established.	The vision sensor is not connected correctly.	Check the wiring of devices by referring to "5.1 Installation and wiring of devices".
		The network adapter is not set correctly.	Select a network adapter to be used correctly according to "7 CONFIGURING THE VISION SENSOR COMMUNICATION SETTINGS".
3		Communication is blocked by the Windows firewall.	Allow the firewall to unblock communication. Refer to "7 CONFIGURING THE VISION SENSOR COMMUNICATION SETTINGS".
4	No live images appear.	It is affected by the internet proxy settings.	Change the proxy settings by referring to "3.4 Computer settings".
5		The firmware version of the vision sensor is old.	Check that the firmware of the vision sensor being used is version 5.07.** or later. Refer to "3.3.3 Vision sensor".
6	The region graphics are not displayed when the [Get] button is clicked in the model registration editor.	The vision sensor image format and the job do not match.	Refer to "3.3.3 Vision sensor" and delete the vision sensor job, and go back to "7 CONFIGURING THE VISION SENSOR COMMUNICATION SETTINGS".
7	The animation is not displayed properly. (For example, a black image appears.)	The AVI file is not associated with Windows Media Player.	Go to [Start] → [Settings] → [Apps] → [Default apps] → [Choose default apps by file type] to associate ".avi video clips" with Windows Media Player.
8	If the application aborts in the process of creating a program	-	The state during process is automatically saved. The process can be resumed from the automatically saved state. Refer to "15.1 Exporting and importing settings".
9	The position where the robot grasps or places the workpiece is not stable.	The installed position of the sensor has shifted.	Check that the vision sensor is fixed securely. If the position of the sensor has shifted due to reasons such as interference, perform calibration again. Refer to "8 CALIBRATION".
		The robot moves while taking images.	Images need to be taken with the vision while the robot is at a complete stop. Check the settings of "Positioning" and "Wait Time" at the imaging position. Refer to "12.2 Modifying an operation".
		The focus and exposure time of the vision sensor are incorrect.	Adjust the focus and exposure time. If the focus has been changed, perform calibration again. Refer to "8 CALIBRATION".
10	The workpiece grasp position or placement point is offset.	-	Register a grasp position again. Refer to "11 REGISTERING A GRASP POSITION".

11	Takt time takes long.	The set speed is too low.	Review the operating speed, positioning accuracy, and waiting time settings. Refer to "12.2 Modifying an operation".
12	Switching workpiece type	-	Up to five types of workpieces can be registered per application. For information on how to register and switch workpiece type, refer to "6.3 Workpiece type number settings".
13	Modifying a robot program	-	When modifying a program for reasons such as interlocking with the peripheral equipment and changing movement for interference avoidance, refer to "13 ACTUAL OPERATION".
14	An error occurred during calibration, causing the robot to stop.	An error indicating that the robot is outside the operating range has occurred. (L2601, L2602, L2603, etc.)	Perform calibration within the operating range of the robot.
15		The robot was stopped and restarted from a position different from the stop position.	Move the robot back to its initial position and perform calibration from the beginning.
16		The robot is operated in a position where the multi-rotation flag or structure flag is switched.	Operate the robot in a position where the multi-rotation flag or structure flag is not switched.
17		The error "L9110 calibration failure (mark recognition)" has occurred.	Data required for calibration calculation cannot be acquired. 1) The recognition target may be outside the camera's field of view. Review the calibration conditions. Refer to "8.5 Performing calibration". 2) The recognition target may not be recognized correctly. Review the settings of the vision sensor recognition parameters ("Threshold" and "Angle Range"). Refer to "9.3 Registering a workpiece (model) ".
18		The error "L9111 calibration failure (calculation of the center of the field of view)" has occurred.	The registered mark has failed to be moved to the center of image (within the convergence target value [mm]) within the specified number of retries. Review the conditions of calibration completion (Convergence Target Value and Convergence Retry Counts). Refer to "8.5 Performing calibration".
19		The error "L9112 calibration failure (calculation of parallelism)" has occurred.	The robot has failed to move so that the registered mark and vision sensor are parallel to each other (within the convergence target value [deg]) within the specified number of retries. Review the conditions of calibration completion (Convergence Target Value and Convergence Retry Counts). Refer to "8.5 Performing calibration".

19. APPENDIX

19.1. Viewing RT ToolBox3 option card information

Option card information can be viewed in RT ToolBox3.
Clicking "Slotn (n = 1, 2): MELFA Smart Plus" (accessed from [Online] - [Board] in the tree structure of the workspace) in online mode will display information on the MELFA Smart Plus card in the Properties window.

* The option card information in the Properties window is not updated automatically. To update the information, change the mode to offline, then online, and perform the above step again.

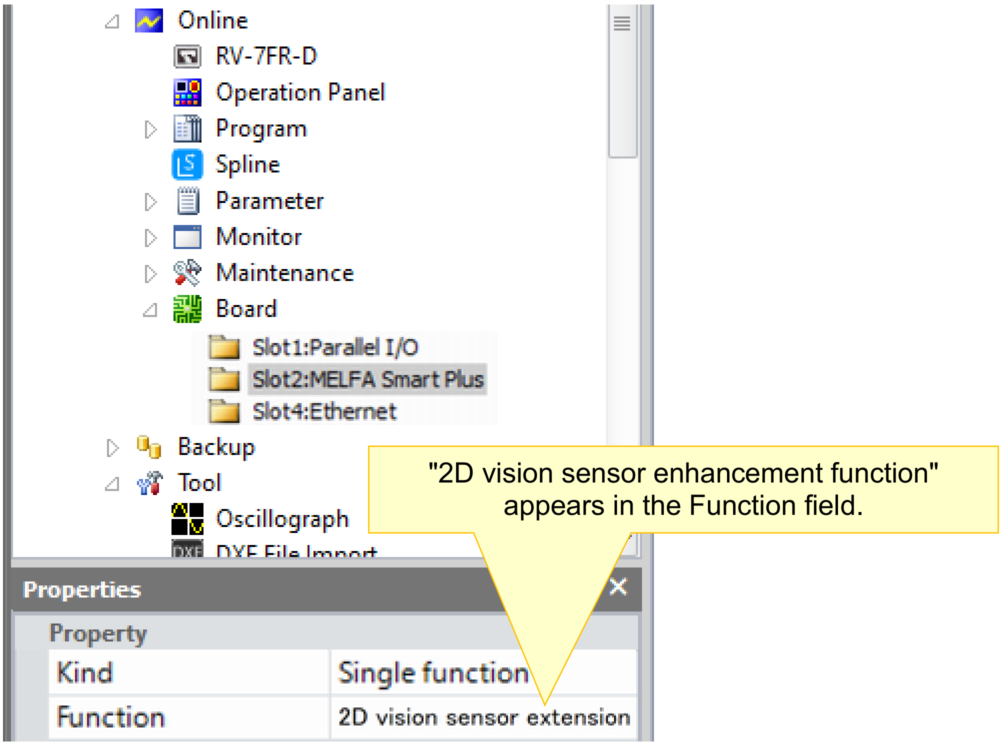
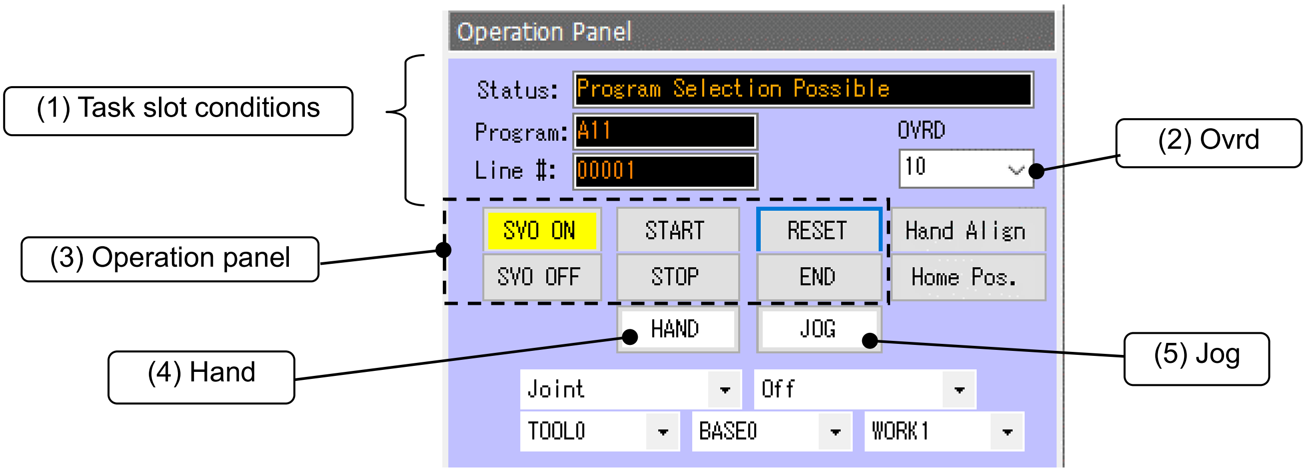


Fig. 19-1 RT ToolBox3 option card information

19.2. How to operate the Operation Panel

The Operation Panel can be used to run a robot program, operate the hand, and perform Jog operation. Hand operation and Jog operation are also possible for the robot controller, but pay great attention to the robot and its surroundings.



(1) Task slot conditions
Indicates the task slot conditions, program being selected, and line number being executed.

(2) Ovrd
Used to display and set the robot's override speed.
It can also be selected from the drop-down box.

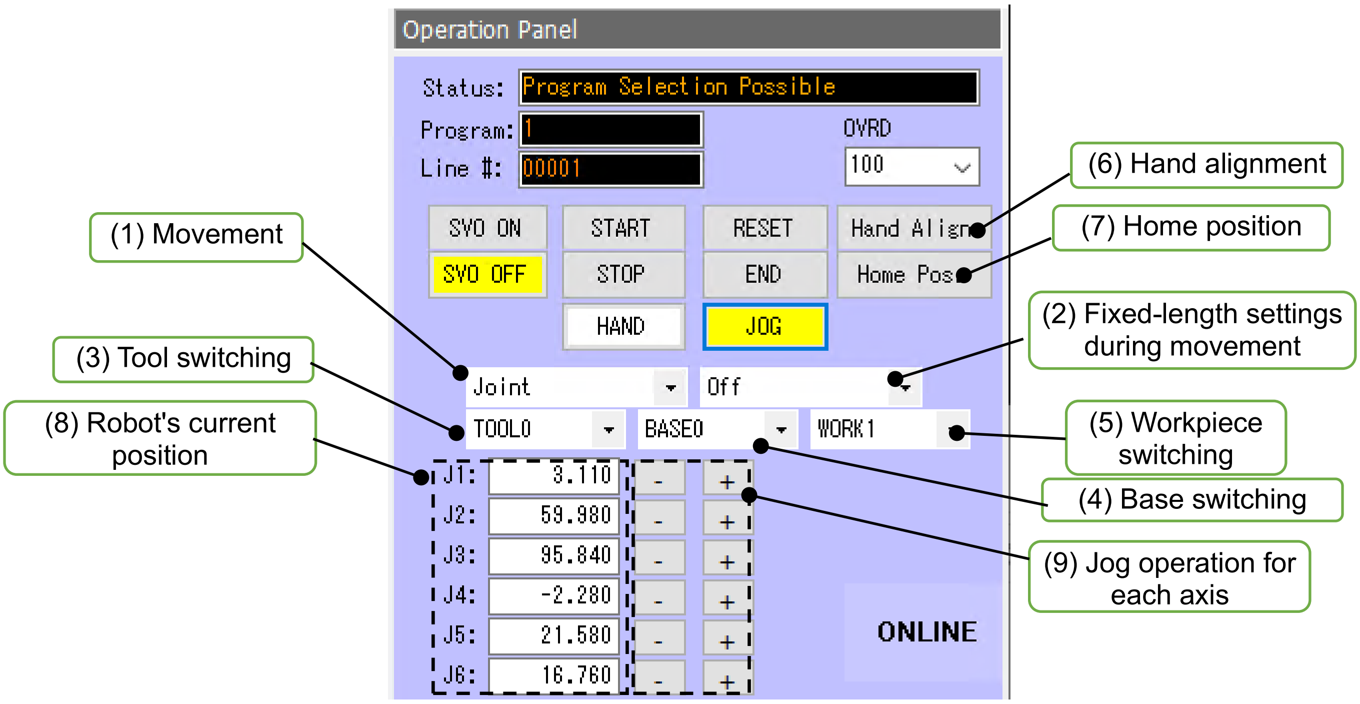
(3) Operation panel
Used to control the robot. A program can be started, stopped, reset, ended, and the servos can be turned on and off.

(4) Hand
Used to operate the robot hand. Clicking the button will display a hand operation screen. Each hand can be opened or closed.

(5) Jog
Used to move the robot using Jog operation. Clicking the button will display a jog operation screen.
Jog operation is also possible for the robot controller.

19.2.1. Jog operation

To perform Jog operation in online mode, allow the computer to acquire operation rights. Set the MODE key of the robot controller to "Automatic". If external I/O signals are used, turn OFF the operation right input from the external I/O.



(1) Movement
Select how the robot moves. The movement can be selected from "Joint", "XYZ", "TOOL", "3-axis XYZ", "Cylinder", and "WORK". For information on each type of Jog operation, refer to the controller Instruction Manual: Detailed explanations of functions and operations.

(2) Fixed-length settings during movement
Select the amount of robot movement. The amount of movement can be selected from "Off", "High", and "Low". For further information on the amount of movement, refer to the controller Instruction Manual: Detailed explanations of functions and operations.

(3) Tool switching
Select a robot tool. The tool that can be selected differs depending on the specifications of the robot controller being connected.

(4) Base switching
Select the base coordinate number of the robot. The base coordinate number can be selected from "BASE0" to "BASE8".
If the base has been switched without using the base coordinate number, "BASE*" will appear.
This is available only for controllers that have the base coordinate number parameter (MEXBSNO).

(5) Workpiece switching
Select a workpiece for workpiece jog operation. The workpiece can be selected from "WORK1" to "WORK8".

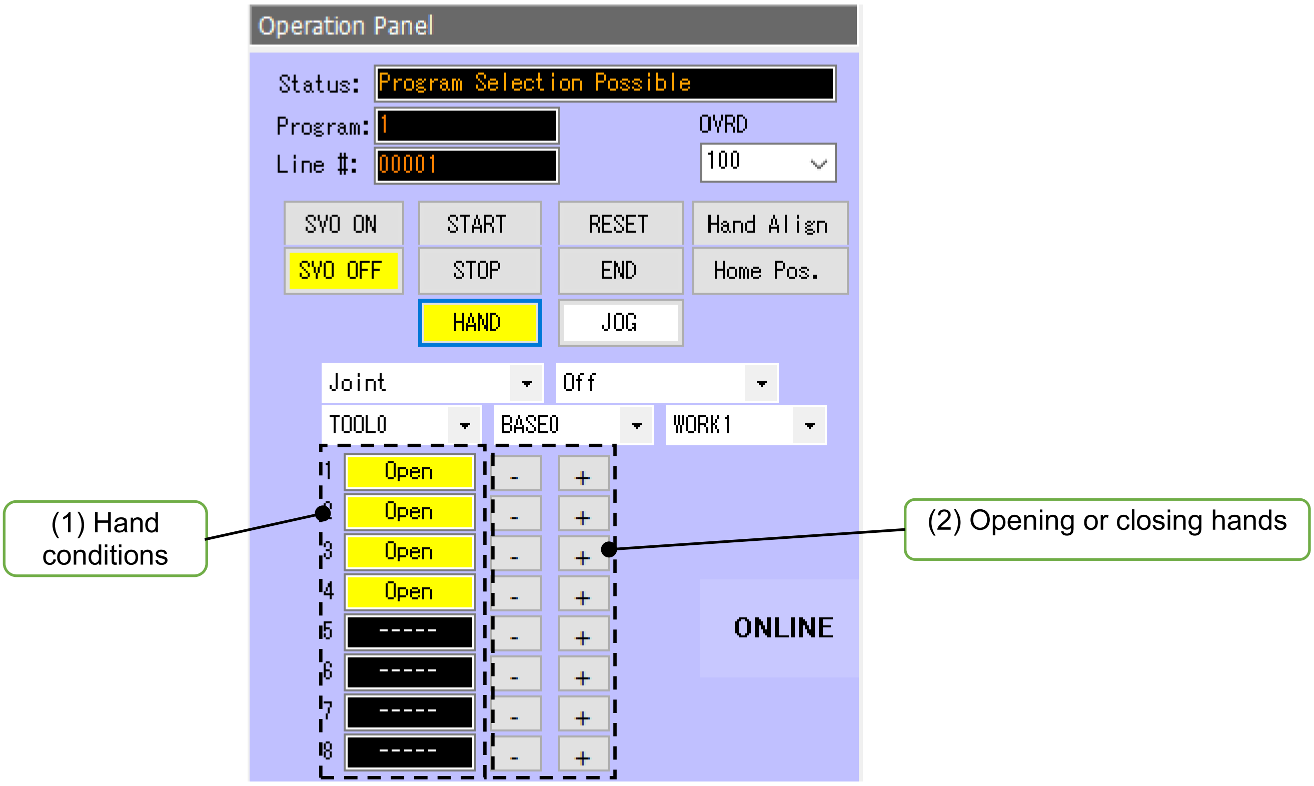
(6) Hand alignment
The posture of the robot hand can be aligned in 90° increments. This function moves the hand in 90° increments to the values closest to the current positions of components A, B, and C.

(7) Home position
The robot can be moved to the position set in the home position parameter (JSAFE).

(8) Robot's current position
Displays the robot's current position.

(9) Jog operation for each axis
Move axes of the robot using Jog operation.

19.2.2. Hand operation



(1) Hand conditions
Displays whether the hand opens or closes.

(2) Opening or closing hands
Open or close the numbered hands. (+: Opens, -: Closes)
The setting of the correspondence between the hand number and the hand connected to the robot can be changed by parameter settings. For details, refer to "5.4 Hand settings".

19.3. Position jump

In the phase of registering positions of the robot (registration of calibration start position, registration of trigger position, and operation settings), the robot can be moved (using the position jump function) to a specified position with the [Move] button in the Setting panel.

(1) Operation when registering a calibration start position

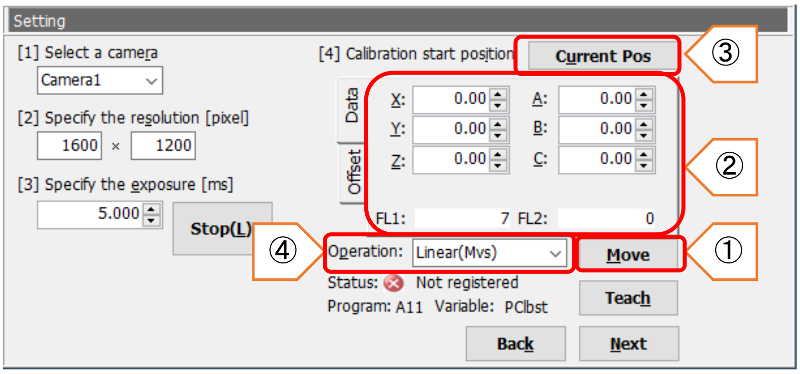


Fig. 19-2 Setting panel when registering a calibration start position

- Clicking the [Move] button (indicated by number 1) will move the robot to the position displayed in the [Data] tab (indicated by number 2).
- Clicking the [Current Pos] button (indicated by number 3) will display the robot's current position in the [Data] tab. Use this step when finely adjusting the position based on the current position.
- Select the movement from "Operation" (indicated by number 4). (XYZ movement or joint movement)
- When the [Offset] tab is selected, the position data (indicated by number 2) shows the relative position from the current position. Use it for relative movement from the current position. (The relative position is set in the Tool coordinate system.)

(2) Operation when registering a trigger position or configuring operation settings

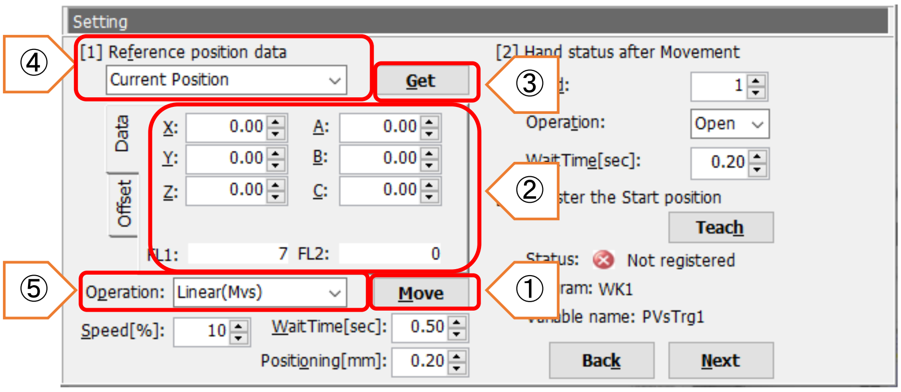
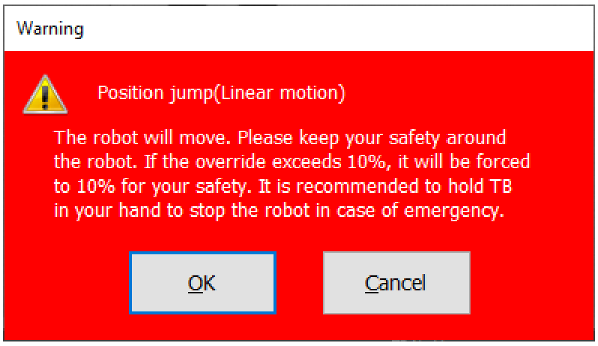


Fig. 19-3 Setting panel when configuring trigger or operation settings

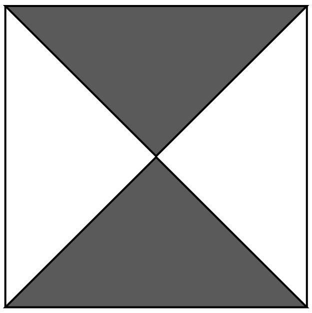
- Clicking the [Move] button (indicated by number 1) will move the robot to the position displayed in the [Data] tab (indicated by number 2).
- Clicking the [Get] button (indicated by number 3) will display the position under "Reference position data" (indicated by number 4) in the [Data] tab. Use this step when finely adjusting the position based on the current position.
- Select the movement from "Operation" (indicated by number 5). (XYZ movement or joint movement)
- When the [Offset] tab is selected, the position data (indicated by number 2) shows the relative position from the current position. Use it for relative movement from the current position. (The relative position is set in the Tool coordinate system.)

An alert window will appear before operation starts. Read the message, and click the [OK] button.



19.4. Calibration mark

Use the mark as a marker when performing non-contact type calibration for robots other than vertical 6-axis robots.
Increase or decrease the size of the mark depending on the usage environment (camera's field of view size).



MITSUBISHI ELECTRIC CORPORATION

HEAD OFFICE: TOKYO BLDG., 2-7-3, MARUNOUCHI, CHIYODA-KU, TOKYO 100-8310, JAPAN
NAGOYA WORKS: 1-14, YADA-MINAMI 5-CHOME, HIGASHI-KU, NAGOYA 461-8670, JAPAN

Sep. 2023 MEE Printed in Japan on recycld paper.