MITSUBISHI
ELECTRIC

Mitsubishi Electric Industrial Robot

CR800-D/R/Q series controller

MELFA Smart Plus User's Manual

2F-DQ510
2F-DQ511
2F-DQ520
2F-DQ521

MELFA

BFP-A3559-B

⚠ Safety Precautions

Before using the robot, always carefully read the precautions below as well as the separate "Safety Manual" and take all necessary safety measures.

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

⚠CAUTION For the sake of safety, teaching work should only be performed by workers who have undergone special training.
(The same is true for any maintenance work done with the power source not cut off.)
→Implementation of safety education

⚠CAUTION For teaching work, prepare work regulations concerning robot operation methods and procedures, measures for when there is an abnormality and when restarting, etc.
Perform teaching work according to these regulations.
(The same is true for any maintenance work done with the power source not cut off.)
→Prepare work regulations.

⚠WARNING For teaching work, set up a device that can stop operation immediately.
(The same is true for any maintenance work done with the power source not cut off.)
→Emergency stop switch setting

⚠CAUTION During teaching work, label the start switch etc. to indicate that teaching work is underway.
(The same is true for any maintenance work done with the power source not cut off.)
→Display that teaching work is underway

⚠DANGER During operation, prevent contact between workers and robots by preparing a fence and a barrier.
→ Setting up a safety fence

⚠CAUTION Determine a uniform signal to relevant staff for the start of operation and use that signal.
→ Signal for the start of operation

⚠CAUTION For maintenance work, in principle, cut off the power and label the start switch etc. to indicate that maintenance work is underway.
→Display that maintenance work is underway

⚠CAUTION Before starting work, check the robot, emergency stop switches, related devices, etc. and make sure there are no abnormalities.
→ Check before the start of work

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

⚠ **DANGER**	By using multiple control devices (GOT, PLC, push button switch) When automatic operation is performed, the interlock such as the operation right of each device is designed by the customer please.
⚠ **CAUTION**	Use the robot in an environment that is within the range of its specifications. Failure to do this can cause a drop in reliability and breakdown. (Temperature, humidity, atmosphere, noise, etc.)
⚠ **CAUTION**	When transporting the robot, put it into its specified transport posture. Failure to do this can cause a drop in reliability and breakdown.
⚠ **CAUTION**	Install the robot on a solid platform. If the robot is in an unstable posture, this can cause positional deviation and vibration.
⚠ **CAUTION**	Wire cables away from noise sources as possible. If cables are brought too close to noise sources, this can cause positional deviation and malfunction.
⚠ **CAUTION**	Do not apply excess force to connectors or bend cables excessively. Doing so can cause a contact defect or cut line.
⚠ **CAUTION**	Set work masses, including hands, so as not to exceed the rated load or permitted torque. Exceeding either of these can cause an alarm or breakdown.
⚠ **WARNING**	Install hands and tools and hold work securely. Failure to do this can cause objects to fly loose during operation and cause personal injury or damage.
⚠ **WARNING**	Ground the robot and controller reliably. Failure to do this can cause malfunction due to noise or in an extreme case, electrical shock.
⚠ **CAUTION**	Display the operating state while the robot is operating. Lack of such a display can result in someone coming too close to the robot by mistake or mistaken operation.
⚠ **WARNING**	Always secure the priority rights for control of the robot before doing any teaching work within the robot's operating range. Failure to do this can allow the robot to start due to an external command, causing personal injury or damage.
⚠ **CAUTION**	Make the jog speed as slow as possible and do not take your eyes off the robot. Failure to do this may cause a collision between the workpiece and peripheral devices.
⚠ **CAUTION**	After completing program editing, but before starting automatic operation, always check operations in a step run. Failure to do this may cause a collision with a peripheral device due to a programming error or the like.
⚠ **CAUTION**	Set up the safety fence in such a way that, while the equipment is running on automatic, either the safety fence door is locked or if anyone tries to open the door, the robot is stopped. Failure to take these protective measures can cause an accident resulting in injury.

CAUTION Never make alterations on your own judgment or use maintenance parts other than those designated. Doing so can cause breakdown and malfunctions.

WARNING When moving the robot arm from the outside, never stick a hand or finger into an opening. Depending on the posture, the hand or finger could get caught in the equipment.

CAUTION Do not stop the robot or perform an emergency stop by switching the robot controller's main power supply OFF.
If the robot controller's main power supply is switched OFF during automatic operation, this can reduce the robot's precision. It could result in collisions with peripheral device or the like due to arm drop or inertia.

CAUTION When rewriting a program, parameters, or other internal information within the robot's controller, do not switch the robot controller's main power supply OFF.
If the robot controller's main power supply is switched OFF during automatic operation, or while a program or parameter is being written, there is a danger of the internal information in the robot controller being corrupted.

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 emergency stop, and door switch) are working properly after the wiring setup is completed.

⚠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 safety of the robot system against unauthorized access from external devices via the network, take appropriate measures. To maintain the safety against unauthorized access via the Internet, take measures such as installing a firewall.

■ User's Manual Revision History

Printing Date	Manual No.	Revision Contents
2017-06-30	BFP-A3559	・First edition
2018-02-01	BFP-A3559-A	・The CR800-Q controller was added.
2018-09-01	BFP-A3559-B	・2F-DQ520 and 2F-DQ521 were added.

*Introduction

Thank you for purchasing this MELFA Mitsubishi Electric industrial robot.
This instruction manual describes the explanation of the MELFA Smart Plus card/card pack option.

Be sure to read this manual before using it and ask MELFA Smart Plus card/card pack to use it after fully understanding the contents.
Although we strive to describe special handling as much as possible in this book, please interpret the items not described in this document as "can not".

This manual is described on the premise that basic operations and functions of Mitsubishi Electric Industrial Robots are understood. For basic operation, please refer to the separate manual "Detailed explanation of functions and operation".

Notation method in this document

⚠Danger	This indicates an item for which incorrect handling could present imminent danger of death or serious injury.
⚠Warning	This indicates an item for which incorrect handling could present a danger of death or serious injury.
⚠Caution	This indicates an item for which incorrect handling could present a danger of injury. It could also present a danger of just physical damage.

Contents

1. Usage

This chapter explains the items to be checked and precautions before using the MELFA Smart Plus card/card pack.

1.1 How to Use this Document

This document explains the functions of the MELFA Smart Plus card as shown in Table 1−1. For the functions provided by the standard robot controller and how to operate them, refer to the "Instruction Manual" attached to the robot controller.
The functions not described in Table 1−1 are explained in the separate instruction manuals. Refer to the following instruction manuals.

- Extended function of MELFA-3D Vision (BFP-A3626)
- Preventive Maintenance (BFP-A3625)

Table 1-1 Contents of this instruction manual

Chapter	Title	Description
1	How to Use this Document	It explains how to use this manual (MELFA Smart Plus card Instruction manual). Please read this before actually using MELFA Smart Plus card/card pack.
2	Confirmation before use	When purchasing the MELFA Smart Plus card/card pack, please confirm the necessary products and check the version of the robot controller.
3	MELFA Smart Plus card	This product explains the function list and installation method to the robot controller.
4	Calibration assistance function	We will describe the calibration support function using 2D vision sensor.
5	Robot mechanism temperature compensation function	Explain the robot mechanism temperature compensation function.
6	Coordinated control for additional axes	Explain the coordinated control for additional axes.
7	Appendix	This section explains how to display card information of MELFA Smart Plus card in RT ToolBox3.

2. Confirmation before use

2.1 Product confirmation

The standard configuration of the product you purchased is as follows. Please confirm.

Table 2-1 Standard configuration of product

No.	Component name		Model name	Quantity
(1)	Instruction manual (this CD-ROM)		BFP-A3563	1
(2)	MELFA Smart Plus card pack	A-type	2F-DQ510	One of the four models included
		AB-type	2F-DQ520	
	MELFA Smart Plus card	A-type	2F-DQ511	
		B-type	2F-DQ521	

Note) The numbers in the table correspond to the numbers in the figure below.

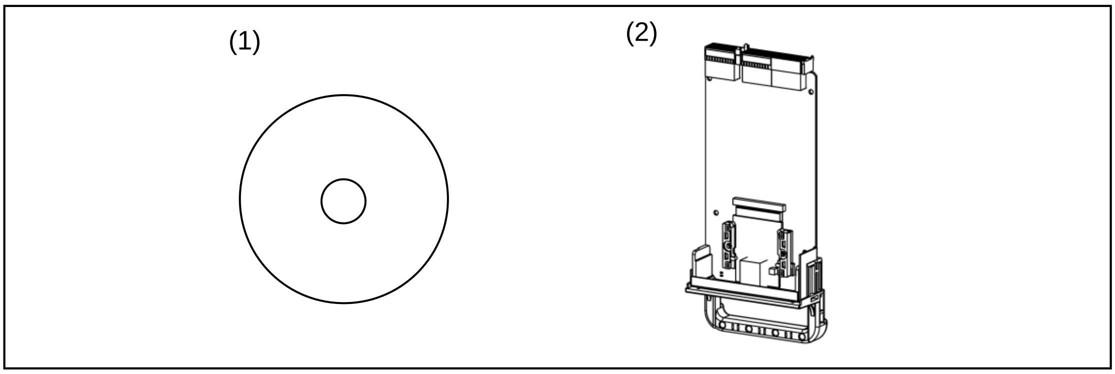


Figure 2-1 Items contained in the delivered product

3. MELFA Smart Plus card

MELFA Smart Plus card .... You can select and use one of the functions according to the MELFA Smart Plus card type. Refer to “3.1 List of function” for the details of functions.

MELFA Smart Plus card pack .... You can use all the functions according to the MELFA Smart Plus card pack type. Refer to “3.1 List of function” for the details of functions.

Table 3-1 Function supported by each type

Functions	A-type	B-type
Calibration assistance function	○	×
Robot mechanism temperature compensation function	○	×
Coordinated control for additional axes	○	×
Preventive Maintenance	○	×
Extended function of MELFA-3D Vision	×	○

You can use all the functions of A-type and B-type according to the MELFA Smart Plus card pack type (AB-type).

3.1 List of function

3.1.1 Functions of A-type

Table 3-2 List of functions of A-type

	Function name	Description	Parameter “SMART+1” setting(*1)
1	Calibration assistance function (Chapter 4)	Using the 2D vision sensor, the following calibration functions can be used. (1) Automatic Calibration Calibration of robot and vision sensor can be automated. (2) Workpiece coordinate calibration Correct between the robot coordinates and workpiece coordinates from the vision sensor. (3) Inter-robot relational calibration You can calculate the relationship between multiple robots.	1
2	Robot mechanism temperature compensation function (Chapter 5)	Measure the temperature of the robot arm and automatically correct errors due to thermal expansion of the arm.	2
3	Coordinated control for additional axes (Chapter 6)	(1) Base coordinate cooperative control Allows synchronized operation where a robot is installed on an additional axis (linear axis) and its speed relative to the workpiece in specified. Supports machining of large workpieces using linear, circular or spline interpolation that exceeds the robot’s range of movement. (2) Additional axis tracking Allows synchronized operation where tracking of the robot and workpieces on an additional axis (linear axis) is specified. Linear or circular interpolation while the workpiece is being transported allows operations such as precision sealing workand surface inspections.	3

	Function name	Description	Parameter "SMART+1" setting(*1)
4	Preventive Maintenance (Refer to the separate instruction manual: BFP-A3625.)	(1) Maintenance simulation In the simulation of an actual device or RT ToolBox3, the timing of parts replacement or maintenance in a repeated specific pattern of movement can be estimated. You can consider maintenance cycle in advance, and test robots to verify they are parts friendly. (2) Consumption degree calculation function Based on the actual robot operation conditions (such as the number of motor revolutions and load conditions), the consumption degree of robot components [%] is calculated, and the timing of maintenance, inspection, and overhaul is displayed and reported. The function brings maintenance efficiency by reporting maintenance timing and setting maintenance priorities, etc.	4

(*1) For MELFA-Smart Plus card pack, this setting is not necessary.

In order to use the above function, the MELFA Smart Plus card/card pack must be attached to the robot controller.

3.1.2 Functions of B-type

Table 3-3 List of functions of B-type

	Function name	Description	Parameter "SMART+1" setting(*1)
1	Extended function of MELFA-3D Vision (Refer to the separate instruction manual: BFP-A3626.)	During the adjustment for the model-less recognition of MELFA-3DVision, our AI and simulation technologies allow to automatically adjust sensor parameters without needing expert knowledge typically required.	101

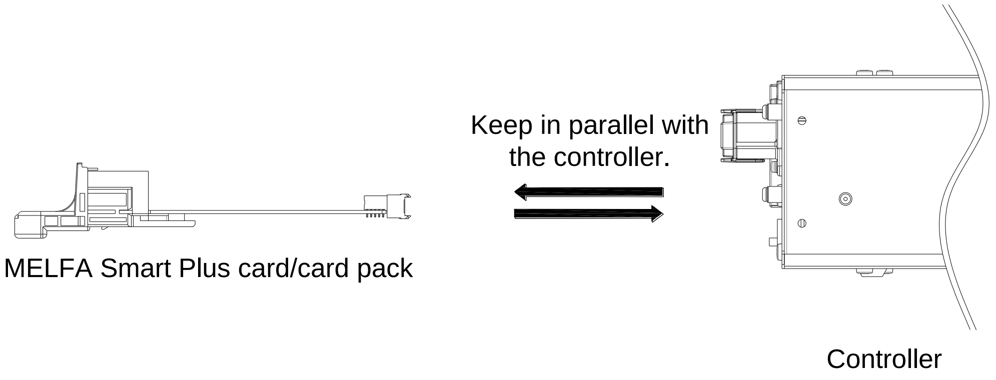
(*1) For MELFA-Smart Plus card pack, this setting is not necessary.

In order to use the above function, the MELFA Smart Plus card/card pack must be attached to the robot controller.

3.2 Installing/removing and setting of the MELFA Smart Plus card

3.2.1 Installing/removing of the MELFA Smart Plus card

⚠Caution When installing or removing the MELFA Smart Plus card/card pack, it is required to keep the card/card pack in parallel with the controller.



3.2.1.1. Installing of the MELFA Smart Plus card

Here is the procedure for installing the MELFA Smart Plus card/card pack.

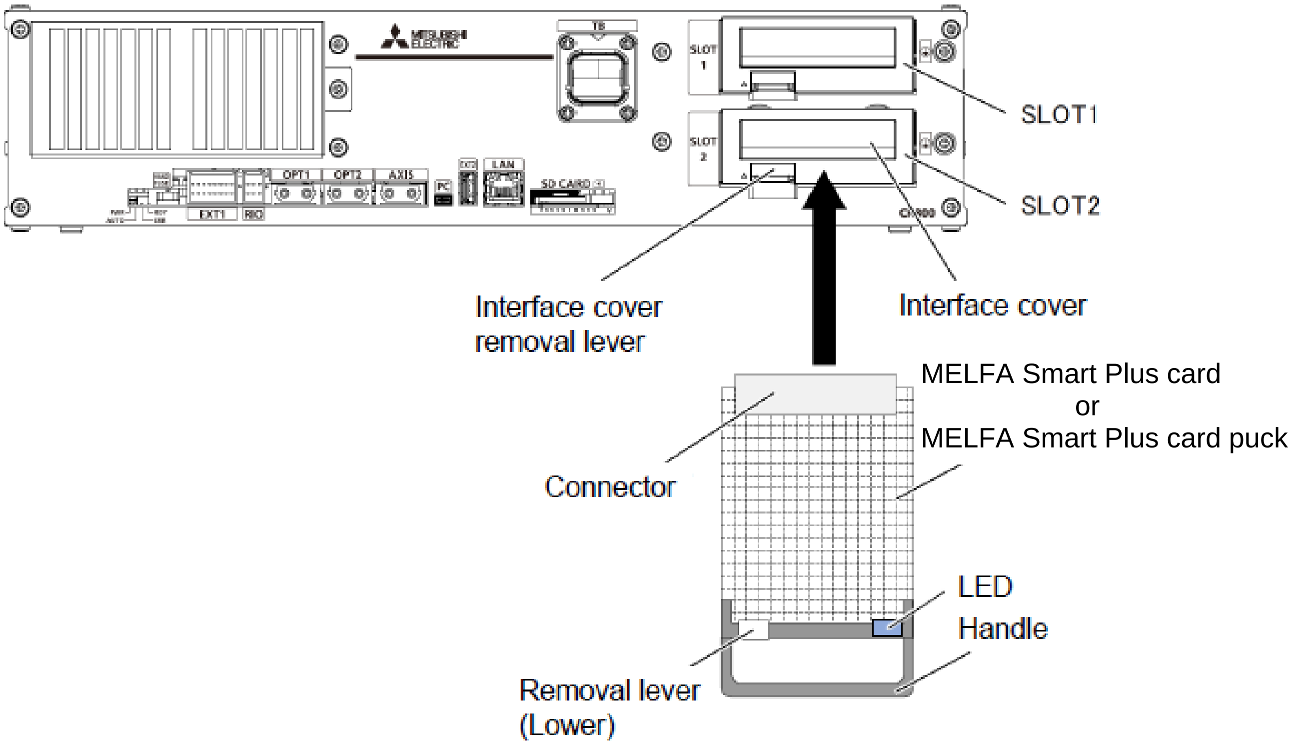


Figure 3-1 Installation of MELFA Smart Plus card/card pack

1) Turn off the controller.
2) Grasp the interface cover removal lever lightly and pull out the interface cover.
3) Hold the handle of the MELFA Smart Plus card and insert it into SLOT 1 or SLOT 2.
   At this time, please insert so that both ends of the card fit into the slots of the slot (SLOT 1 and SLOT 2 in **Figure 3-1**).
4) Insert the connector firmly until it is firmly locked until the release lever clicks.

Installation of the MELFA Smart Plus card is now complete.

3.2.1.2. Removing of the MELFA Smart Plus card

Here is the procedure for removing the MELFA Smart Plus card/card pack.

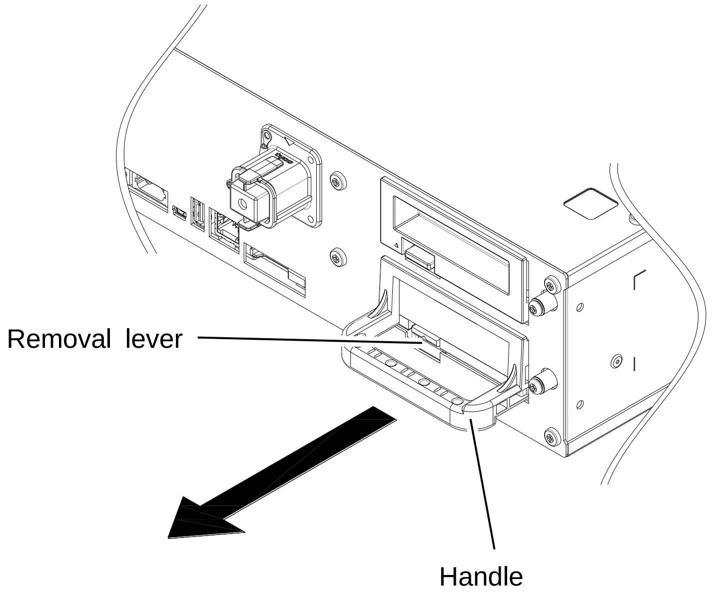


Figure 3-2 Removing of MELFA Smart Plus card/card pack

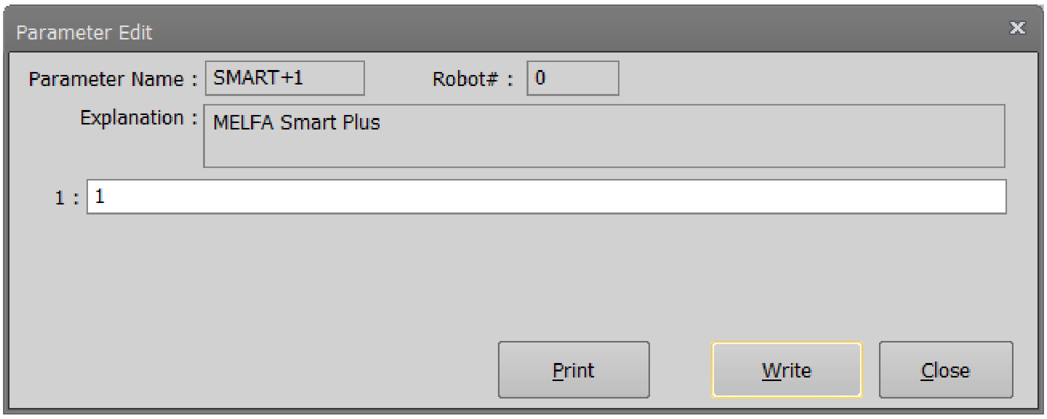
1) Turn off the controller.
2) Remove the MELFA Smart Plus card with the removal lever pushed up. Hold the handle of the MELFA Smart Plus card and pull out the MELFA Smart Plus card in parallel with the controller.

Removing of the MELFA Smart Plus card is now complete.

3.2.2 Setting of the MELFA Smart Plus card (Only MELFA Smart Plus card)

When using the MELFA Smart Plus card, you can use only one supported function.
The setting method of the function to be used is shown below.
If you use the MELFA Smart Plus card pack, this setting is not necessary.

1) Select [Online] -> [Parameter] -> [Parameter List] from the project tree of RT Toolbox3 and display the parameter list screen.
2) Enter the parameter name "SMART+1" on the parameter list screen and display the parameter editing screen.



3) Change the setting value of parameter "SMART+1". Refer to the following table.
The error occurs if the robot programming language (commands, system functions, and state variables) other than the set function is used. Refer to Table 3-6 for the details of errors.

Table 3-4 Relationship between the setting value and function

<table>
<tr><th>Setting value</th><th>Available functions</th><th>Flashing color of the LED (*1)</th></tr>
<tr><td>0</td><td>Cannot use the MELFA Smart Plus function
(Factory setting)</td><td>Red</td></tr>
<tr><td>1</td><td>Calibration assistance function</td><td rowspan="5">Green</td></tr>
<tr><td>2</td><td>Robot mechanism temperature compensation function</td></tr>
<tr><td>3</td><td>Coordinated control for additional axes</td></tr>
<tr><td>4</td><td>Preventive Maintenance</td></tr>
<tr><td>101</td><td>Extended function of MELFA-3D Vision</td></tr>
</table>

(*1) The location of the LED is shown in the **Table 3-2**.
If using the MELFA Smart Plus card pack, the LED flashes blue.

⚠CAUTION Install only one MELFA Smart Plus card.
If multiple MELFA Smart Plus cards are installed, error (L3782) will occur without flashing the LED. Refer to Table 3−6 for the details of errors.

3.3 Robot language specification

This chapter explains the MELFA-BASIC VI robot program language relating to MELFA Smart Plus.

3.3.1 Robot status variable list

Below is a list of state variables related to the MELFA Smart Plus.

Table 3-5 Robot status variable list

Variable name	Array designation	Function	Attribute (*1)	Data type
M_SmartPlus	1	MELFA Smart Plus Function Enabled Available	R	Integer type
C_SmartPlus	1	MELFA Smart Plus function name	R	Character string type

(*1) R : Only reading is possible. R/W : Both reading and writing are possible.

3.3.2 Detailed explanation of robot (system) state variable

The details of the state variables related to MELFA Smart Plus are shown below.

The way of viewing the contents described in the explanation of the status variable is as follows.

[Function] : This indicates a function of a variable.
[Format] : This indicates how to enter arguments of an instruction.
[ ] means that arguments may be omitted. System status variables can be used in conditional expressions, as well as in reference and assignment statements. In the format example, only reference and assignment statements are given to make the description simple.
[Terminology] : This indicates the meaning and range of an argument.
[Reference Program] : An example program using variables is shown.
[Explanation] : This indicates detailed functions and precautions.

M_SmartPlus

[Function]
Refer to the available status of each function of MELFA Smart Plus.

[Format]

```
<Numeric variable> = M_SmartPlus(<ID>)
```

[Terminology]

<Numeric variable> Specify numerical variable to assign.
0 : Disabled
1 : Enabled

<ID> Specify the function ID.
1 : Calibration assistance function
2 : Robot mechanism temperature compensation function
3 : Coordinated control for additional axes
4 : Preventive Maintenance
101 : Extended function of MELFA-3D Vision

[Reference Program]

```
1    M1 = M_SmartPlus(1)      ' Refer to the available state of the calibration assistance function.
```

[Explanation]
(1) Refers to the available status of the MELFA Smart Plus function specified by <ID>.
(2) It is read-only.

C_SmartPlus

[Function]
Returns the function name of MELFA Smart Plus.

[Format]

<Character string variable> = C_SmartPlus(<ID>)

[Terminology]

<Character string variable>	Specify the character string variable to assign. Calibration assistance / Robot temperature compensation / Coordinated control for additional axes / Preventive Maintenance / Extended function of MELFA-3D Vision
<ID>	Specify the function ID. 1 : Calibration assistance function 2 : Robot mechanism temperature compensation function 3 : Coordinated control for additional axes 4 : Preventive Maintenance 101 : Extended function of MELFA-3D Vision

[Reference Program]

```
1    CMSP$ = C_SmartPus(1)  ' Get the name of the calibration assistance function.
```

[Explanation]
(1) Returns the name of the MELFA Smart Plus function specified by <ID>.
(2) It is read-only.

3.4 Error list

<The meaning of the error>

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

Table 3-6 MELFA Smart Plus card related error list

Error number	Error cause and measures	
L.3780	Error message	Cannot use the MELFA Smart Plus.
	Cause	Invalid MELFA Smart Plus.
	Measures	Check the MELFA Smart Plus card or parameter.
L.3781	Error message	Cannot use the MELFA Smart Plus.
	Cause	Invalid MELFA Smart Plus.
	Measures	Check the MELFA Smart Plus card or parameter.
L.3782	Error message	There're MELFA Smart Plus Cards.
	Cause	Multiple MELFA Smart Plus cards are installed.
	Measures	Turn off controller and pull unnecessary MELFA Smart Plus card.

4. Calibration assistance function

This section explains the calibration assistance function using 2D vision sensor.

In this section we use Cognex's vision for vision sensor, Describe the operation method when setting vision in the EasyBuilder view of Cognex In-Sight Explore.
If you are using another vision sensor, please use the communication part of the sample program output by this function according to the specification of each manufacturer.

4.1 Outline

Example for the calibration assistance function is as follows.

Table 4-1 calibration assistance function list

Name	Contents
Automatic calibration	Teach the robot coordinate and image coordinate to calculate the calibration data between the robot and the camera.
Workpiece coordinate calibration	Use the vision sensor to do the calibration of robot coordinate and arbitrary workpiece coordinate.
Inter-robot relational calibration	According to the same workpiece coordinate defined by multiple robots, the positional relationship between the robots can be calculated.

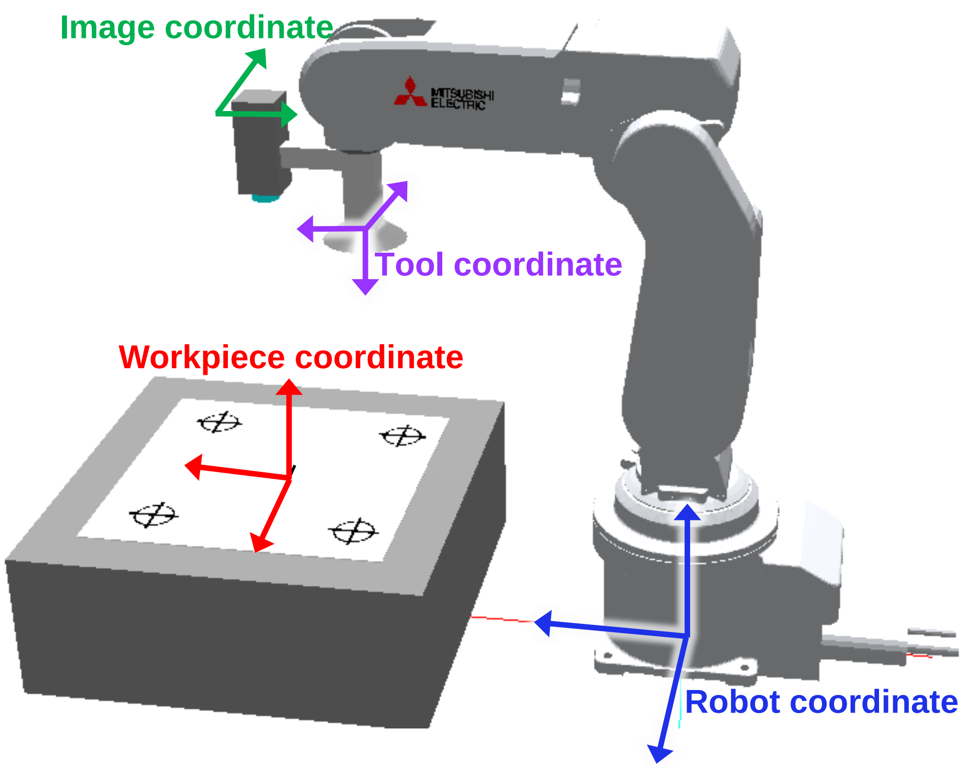


Figure4-1 Calibration assistance function

(1) System components

System components example is as follows

a) D type robot

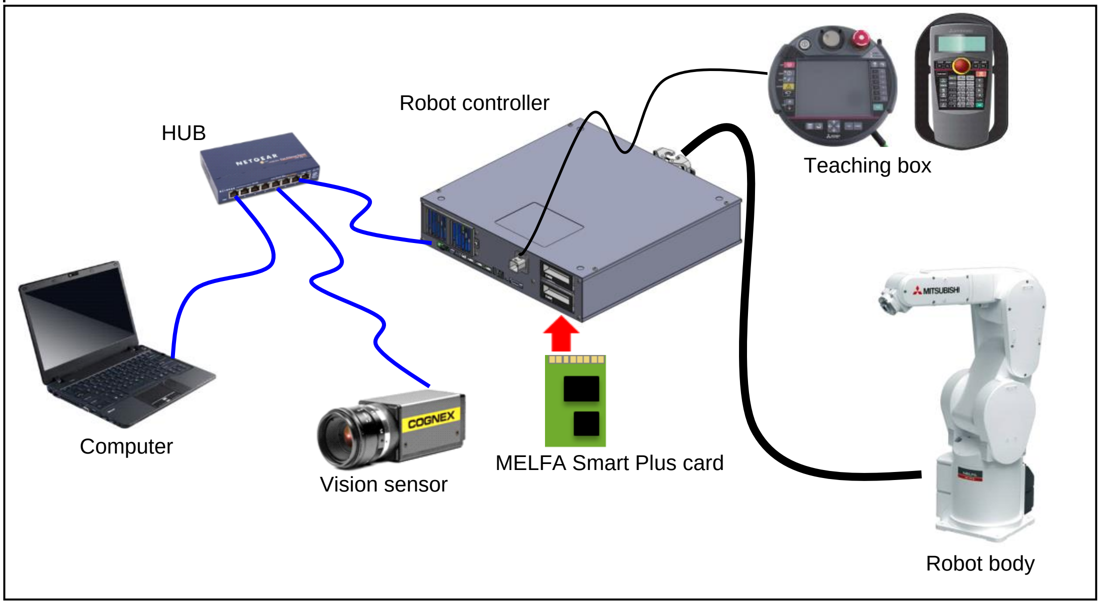


Figure 4-2 D type System components example

b) R type/Q type robot

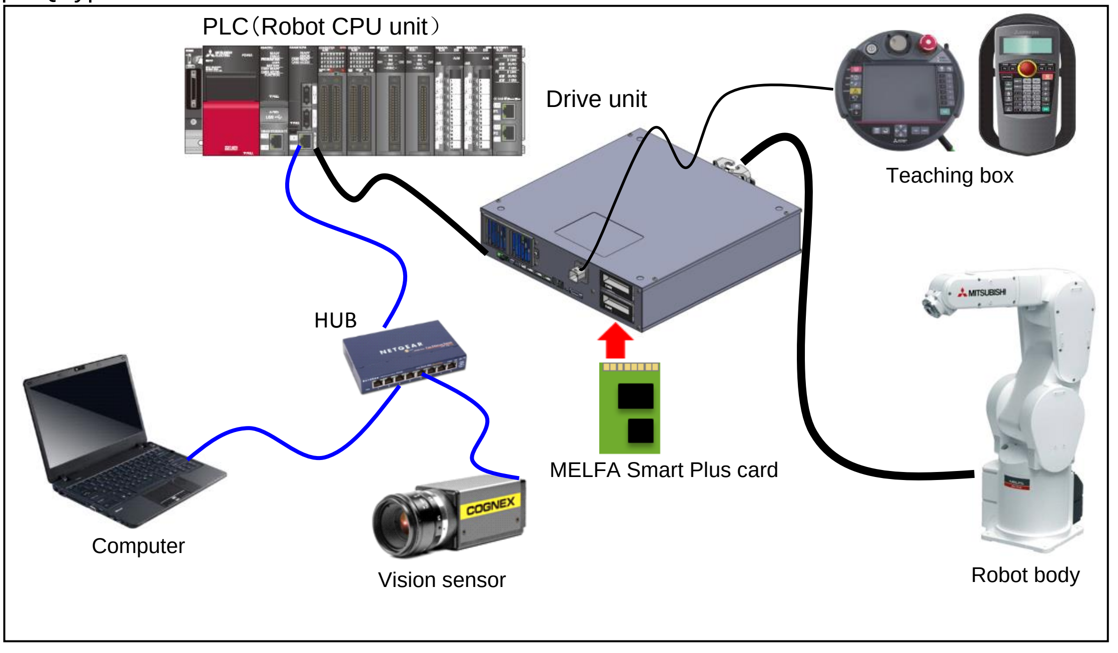


Figure 4-3 System components example of the R type/Q type robot

4.2 Automatic Calibration

(1)Function Outline

"Automatic calibration" is a function for matching the coordinate in the vision sensor with the coordinate in the robot.

The function is possible to convert the image coordinate (work position, release position, etc.) measured by the vision sensor into robot coordinate.

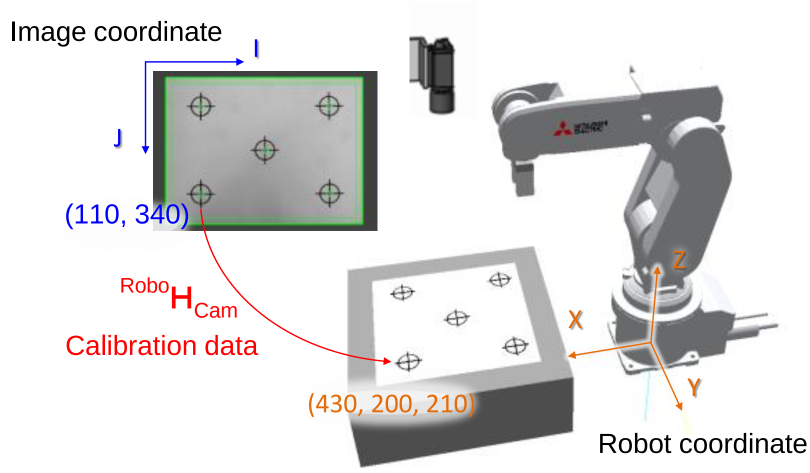


Figure 4-4 Automatic calibration image

(2) Standard Specifications

Table 4-2 Standard specifications of the Automatic calibration

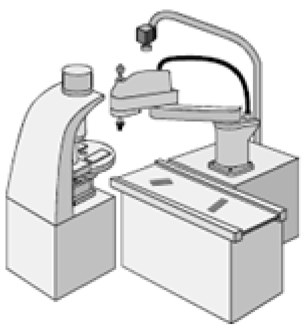
Items	Specifications
Robot	Vertical 6-axis robot Horizontal 4-axis robot * It can't be used with the vertical 5-axis robot and the user robot.
Language	Only MELFA-BASIC VI
Number of the calibration data that can be registered	8（VSCALB1 to 8）
Number of correspondence points that can be set for one calibration data	20 pairs
Remarks	Do not pass through the singular point of the robot, Please teach the points of the calibration. * Only the robot's XY plane can be used.

(3) Camera setting method

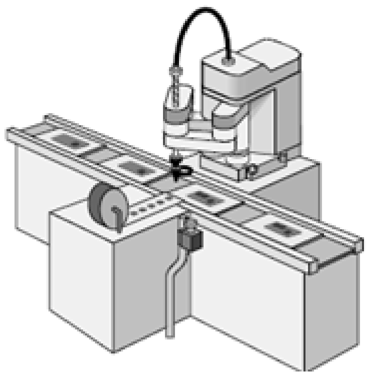
Corresponding camera setting method by automatic calibration is as follows.

Table 4-3 Corresponding camera setting method by automatic calibration

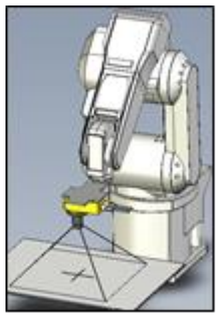
Name	Camera	Use the calculated calibration data to change the coordinate
Fixed method (Top fixed)	Applicable to the workpiece at the top, the camera faces down.	World coordinate
Fixed method (Bottom fixed)	Applicable object workpiece is below, camera facing up.	World coordinate
Hand eye method	Applicable to the camera attached to the robot hand (under the flange).	Tool coordinate



Fixed method (Top fixed)



Fixed method (Bottom fixed)



Hand eye method

Figure 4-5 Camera setting method

(4) Workflow

Start work

↓

1.Preparation and installation of equipment Refer to 4.2.(4-1)
Prepare and install marks to make your vision perceive.

↓

2.Set communication parameters Refer to 4.2.(4-2)
Execute the visual sensor and robot communication settings.

↓

3. Set robot data Refer to 4.2.(4-3)
Set the calibration data and teach the initial point.

↓

4. Create vision sensor job Refer to 4.2.(4-4)
Set up the vision sensor job (program).

↓

5. Operation execution Refer to 4.2.(4-5)
Automatically activates the robot program and executes the calibration operation.

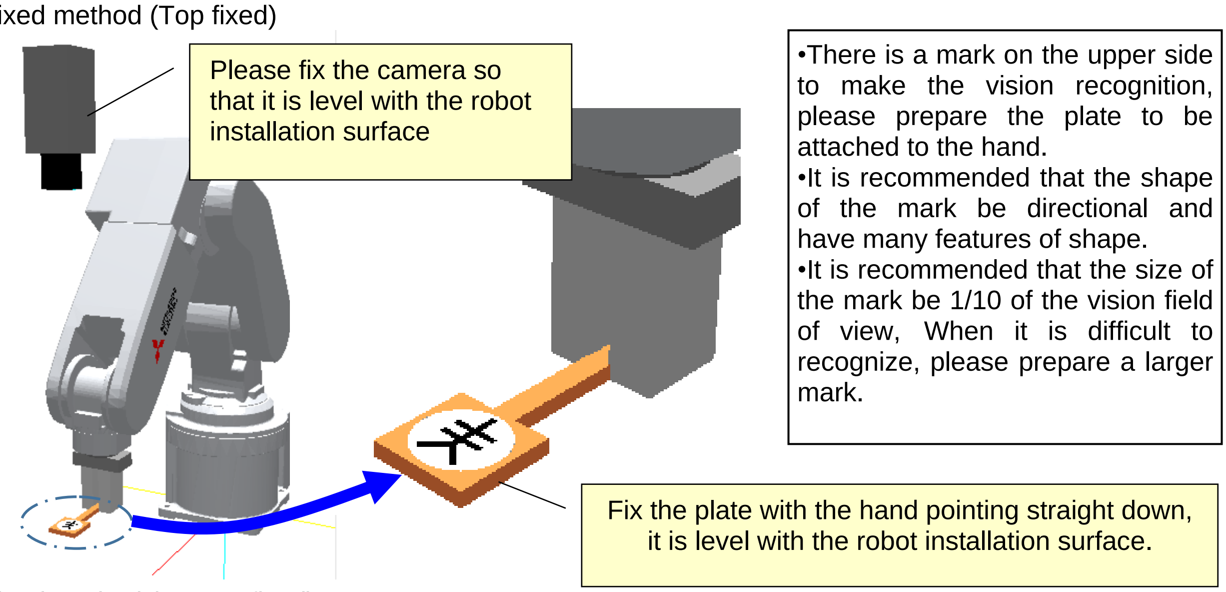
↓

End work

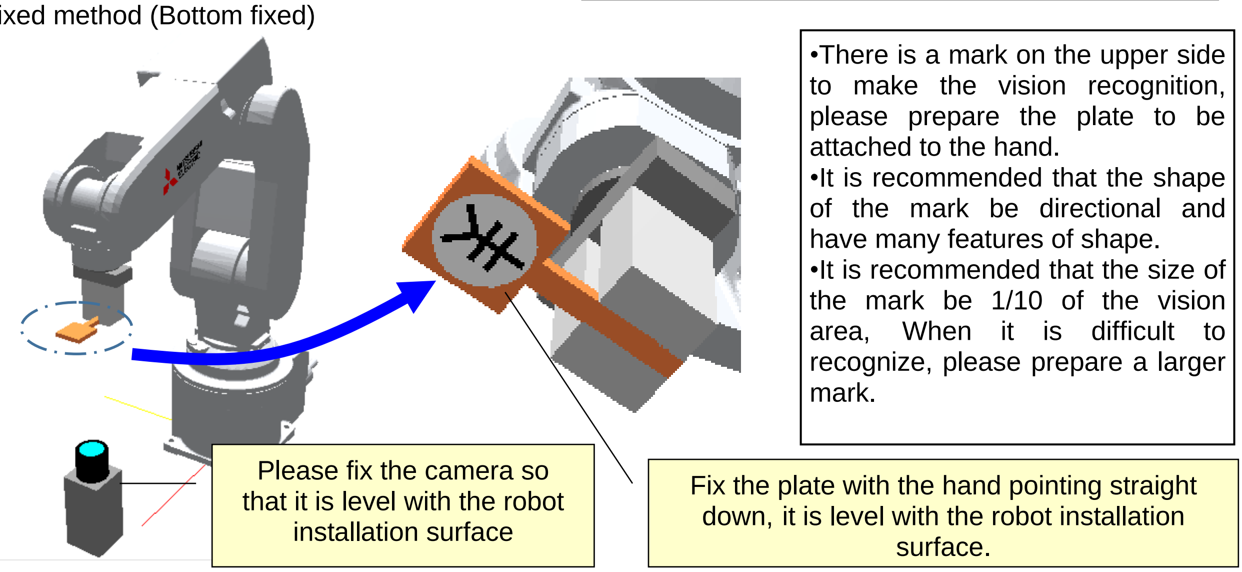
(4-1)Preparation and installation of equipment

In the automatic calibration, the robot recognizes the calibration plate while changing the posture and proceed with work.The calibration plate is a product for customers.Please prepare the plate according to the following contents.

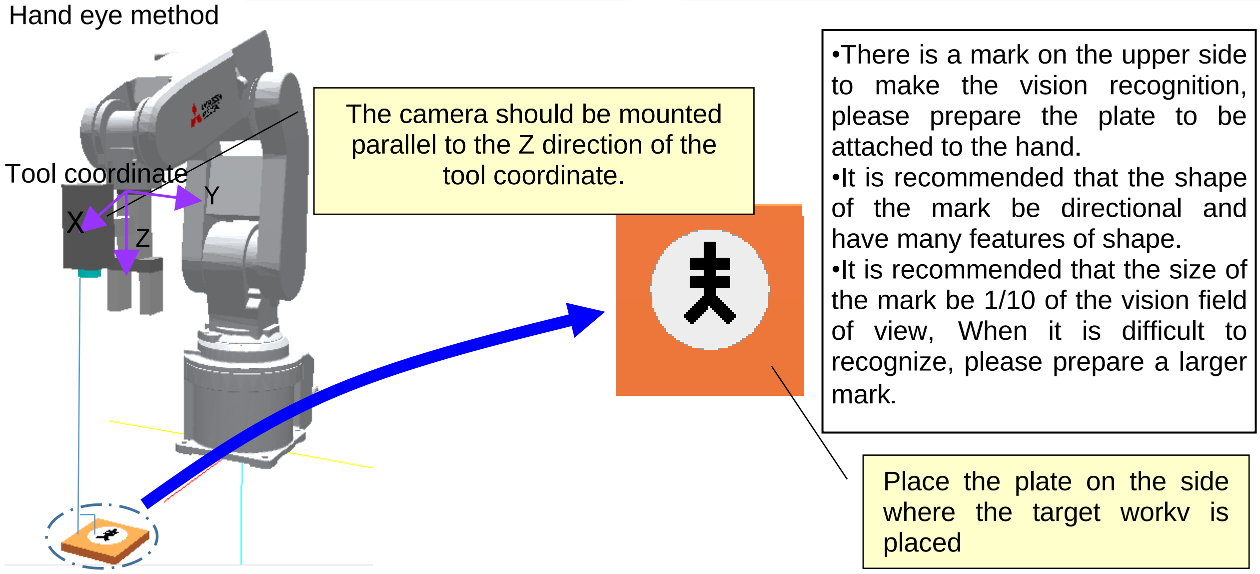
a) Fixed method (Top fixed)



b) Fixed method (Bottom fixed)



c) Hand eye method

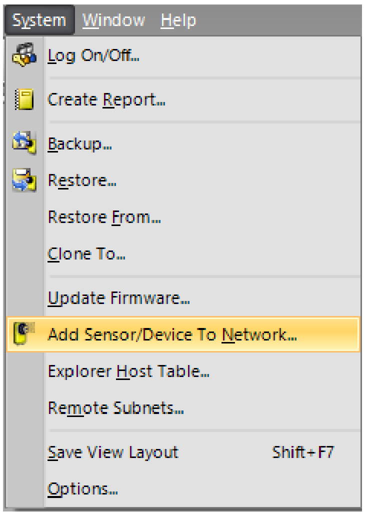


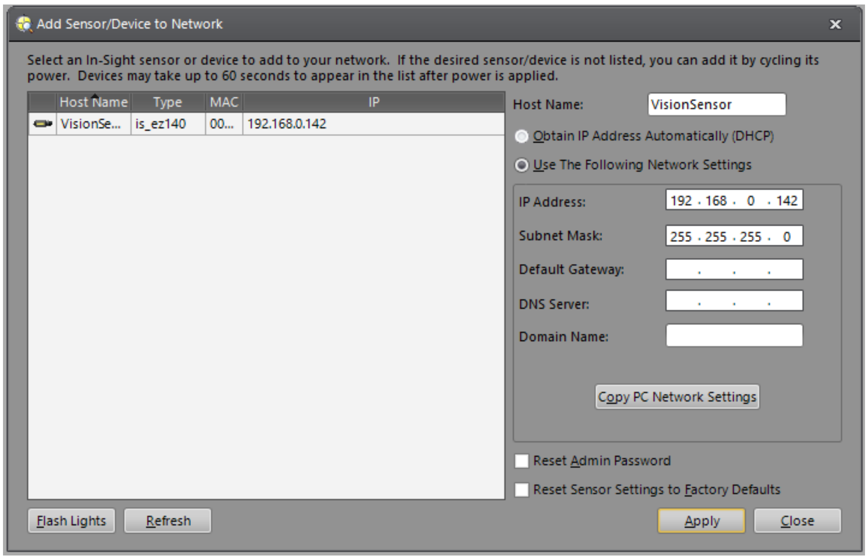
(4-2)Set communication parameters

a) Vision side setting

Start up the In-Sight Explorer, Set the IP address and subnet mask of the vision sensor.

Select [System] - [Add Sensor/Device to Network] from the In-Sight Explorer menu. On the displayed "Add Sensor / Device to Network" screen, Sensors / devices that can be added to the network are displayed. Select the device from the list and set IP address · subnet mask.





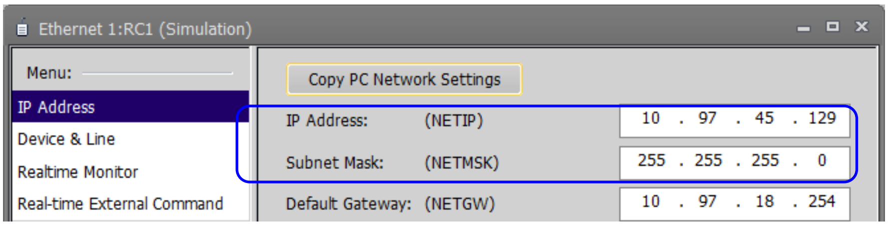
b) Robot side setting

Start RT ToolBox 3 and set the parameters.

①Set IP address · Subnet mask.

1) Select <online> - <parameter> - <communication parameter> - <Ethernet> and display the Ethernet parameter screen.

2) Set the IP address and subnet mask of the robot from the IP address menu.



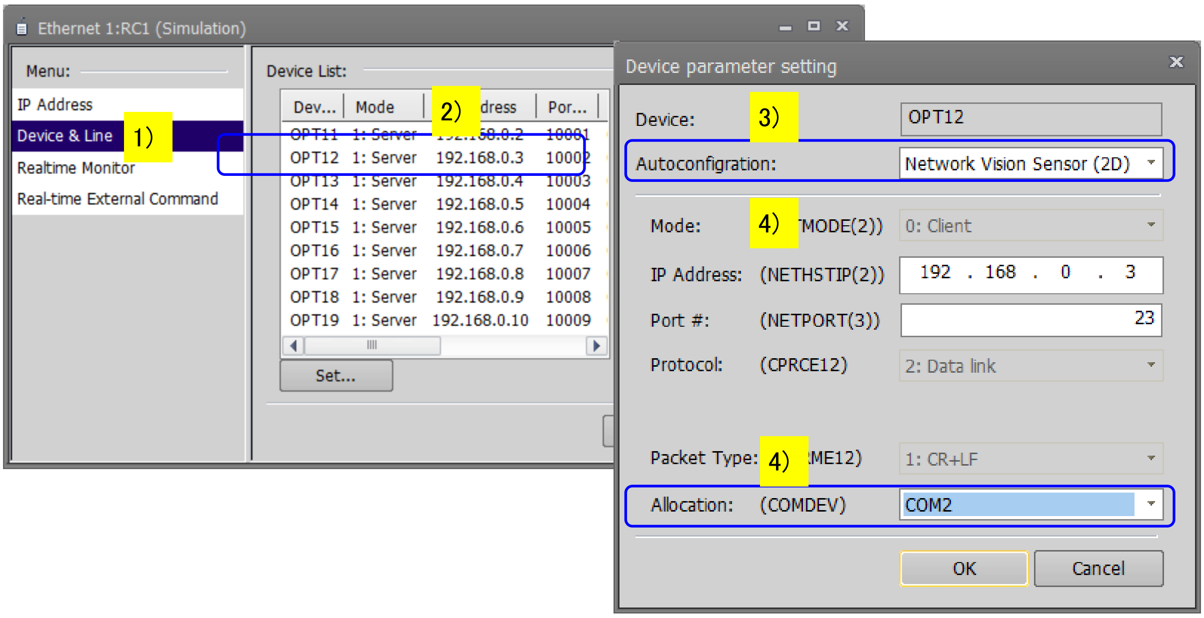
②Set IP address, port number, COM assignment of vision sensor.

1) Select device line menu.

2) Double-click the target device, Display the device parameter setting screen.

3) Select "Network vision sensor (2D)" by automatic setting.

4) Set IP address · port number · COM assignment.

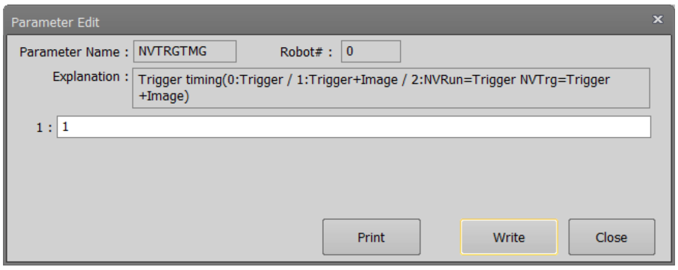


③Set the vision trigger timing parameter (parameter name: NVTRGTMG).

1) Select < Online> - <Parameter> - <Parameter List> and display the parameter list screen.

2) Enter the parameter name "NVTRGTMG" and display the parameter editing screen.

3) Set the value of NVTRGTMG to "1".

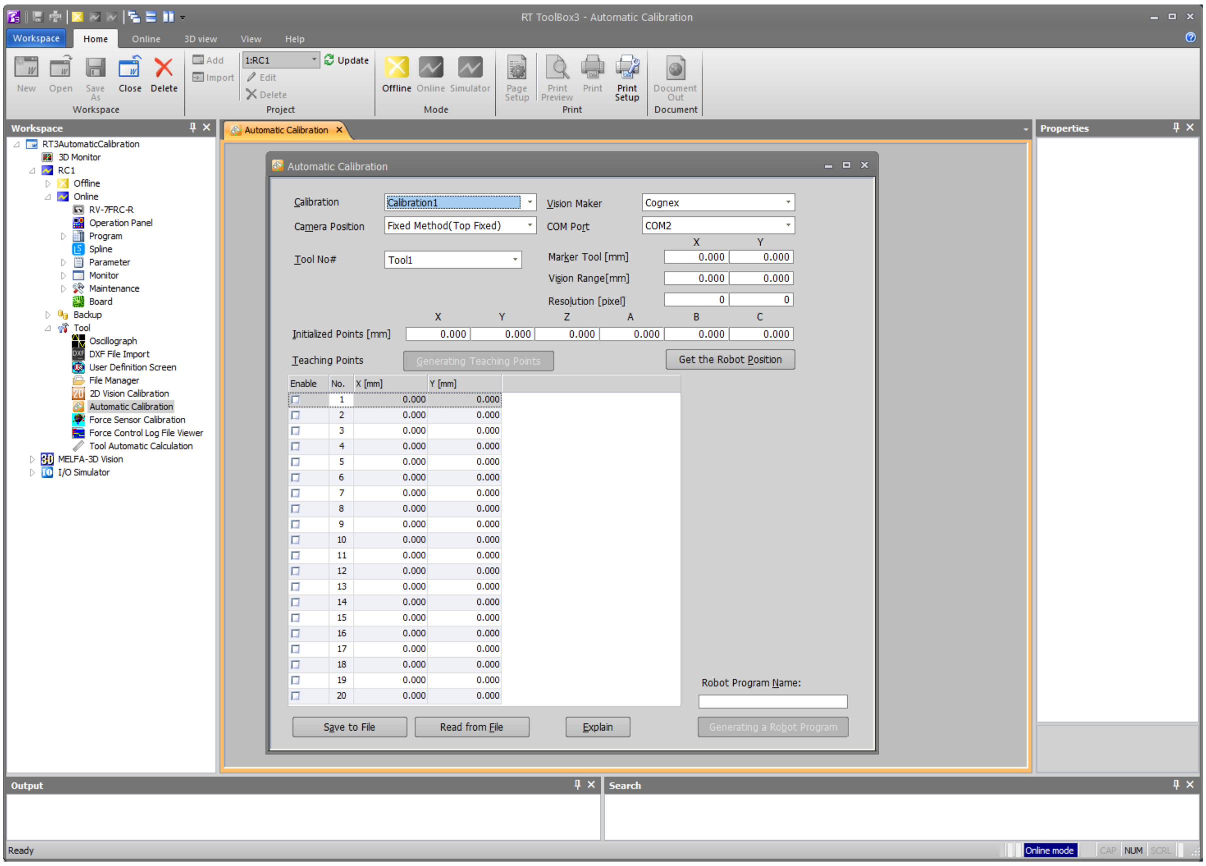


(4-3) Set robot data

Set calibration data from the automatic calibration screen of RT ToolBox 3.

① Launch the automatic calibration screen

Start RT ToolBox 3, select <Tool> - <Automatic Calibration>, Display the automatic calibration screen.
<Automatic Calibration> is not displayed when offline.



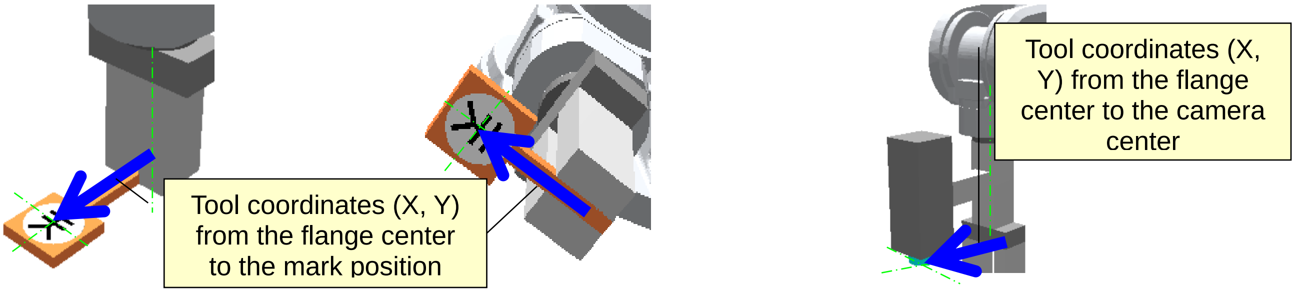
② Set the initial data

The Auto Calibration screen is set as follows.

Calibration [Calibration1 ▾]	Select the the calibration number. ･Selection range：Calibration1 to Calibration8
Camera Position [Fixed Method(Top Fixed) ▾]	Select camera setting method.Click the [Camera Position] button to display the schematic diagram of the setting method. ･Selection:Fixed Method（Top Fixed）/ Fixed Method（Bottom Fixed）/ Hand-eye Method
Vision Maker [Cognex ▾]	Select maker of vision sensor. ･Selection:Only Cognex
COM Port [COM2 ▾]	Select COM port number. ･Selection range：COM2 to COM8
Tool No# [Tool1 ▾]	Select the tool parameter number. (*1) ･Selection range：Tool1 to Tool16
Marker Tool [mm] X 0.000 Y 0.000	Set approximate value of marker tool length. (*2) ･Fixed Method: Tool coordinates from the flange center to the mark position. ･Hand-eye Method: Tool coordinates from the center of the flange to the camera center.

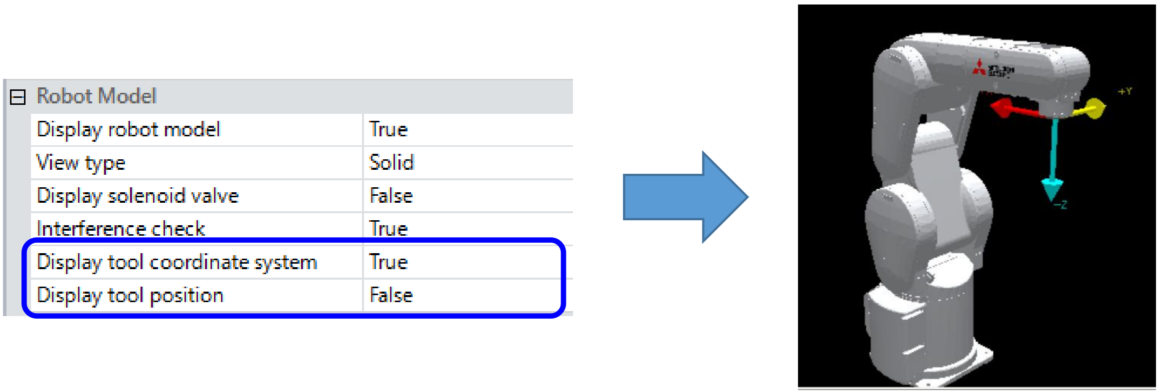
Vision Range [mm] X 0.000 Y 0.000	Set the vision range of the vision sensor. (*3) Please set the viewing range (X, Y) based on the base coordinates.
Resolution [pixel] X 0.000 Y 0.000	Set the resolution of the vision sensor. Please check the specification of the vision sensor to be used.

(*1) In the process of calibration operation, calculate the following tool coordinates. Specify the storage location of the tool coordinates at that time.

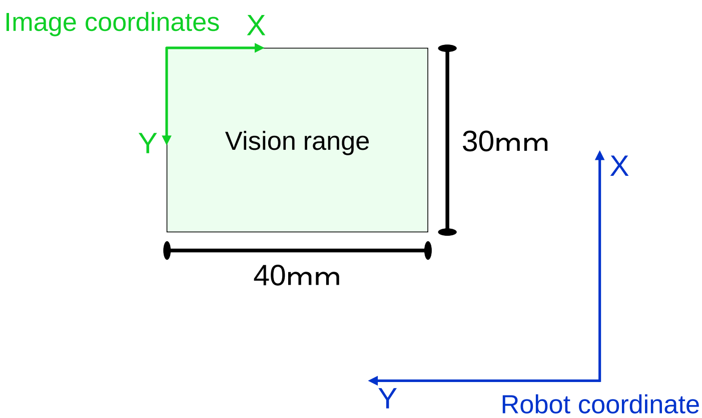


fixed method（Top fixed） fixed method（Bottom fixed） Hand eye method

(*2) Please enter the length of the marker tool so that the calibration plate will not come out of the vision sensor's field of view.
For the marker tool, please input the value in the mechanical interface coordinate.
The mechanical interface coordinate can be checked by 3D monitor in the RT ToolBox 3.
To display the mechanical interface coordinate, please select [Properties] -> [Robot Model] in 3D monitor and set "Display tool coordinate system" and "Display tool position" as shown below.



(*3) An example of setting the Vision Range is shown.
In the case of the following figure, the Vision Range (x, y) is set to (30, 40).



③ The initial point of teaching

1) Start In-Sight Explorer, set the application step as [Set Up Image], select [Live Video], The camera image is displayed in the monitor.

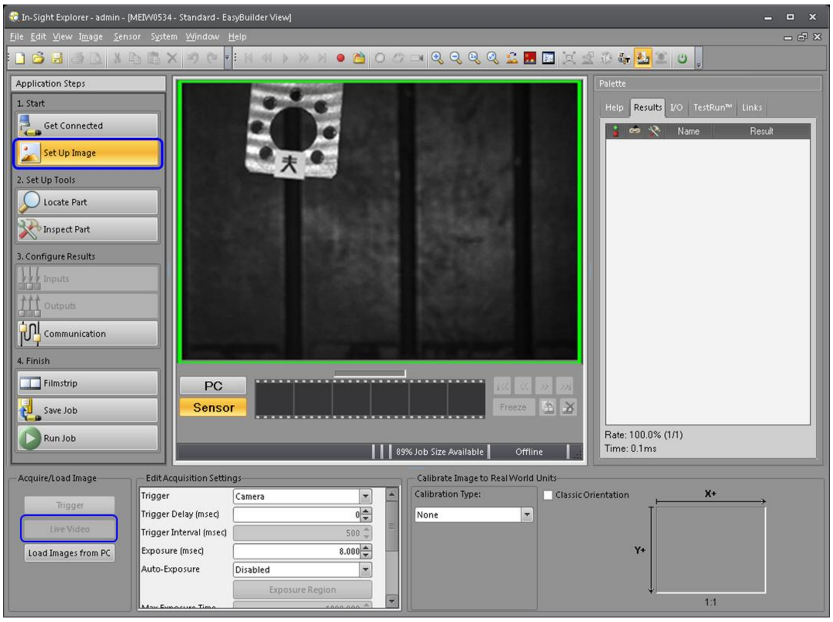


Figure 4-6 In-Sight Explorer Live video

2) Move the recognition mark by the robot's jog operation so that it is near the center of the camera field of view.
At this time, the distance from the recognition mark to the camera should match the distance between the work surface and the camera when recognizing the actual work.
(At this time, please adjust the focus and aperture of the camera. When the camera's focus / aperture is not adjusted, calibration may not be performed accurately.)

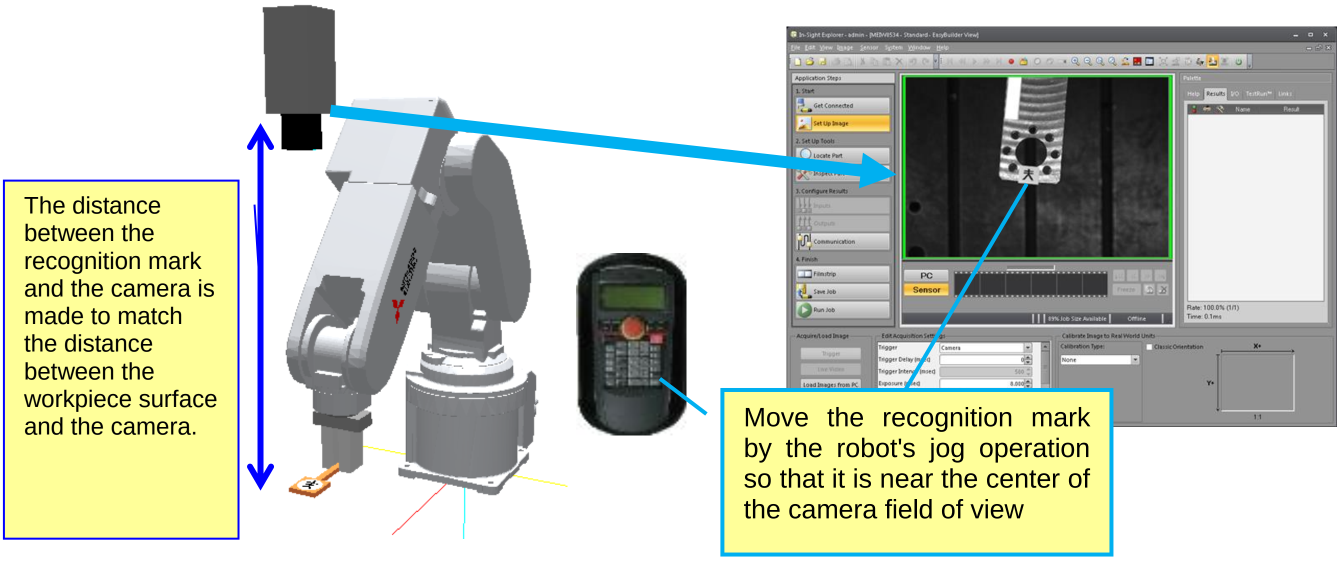


Figure 4-7 Movement of recognition mark to center position

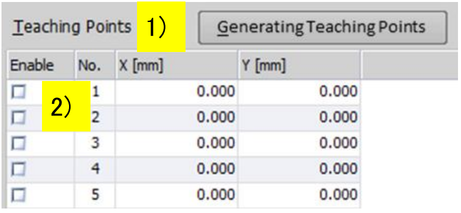
(It shows the fixed type (Top fixed) case, Please operate fixed method (Bottom fixed) / hand-eye method equally)

3) Teach the initial point on the automatic calibration screen of RT Toolbox 3.

Teaching the initialized point  	Click the [Get the Robot Position] button, the initialized point is taught and the taught coordinate values are displayed.

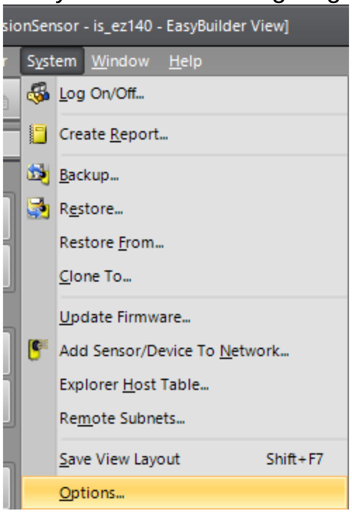
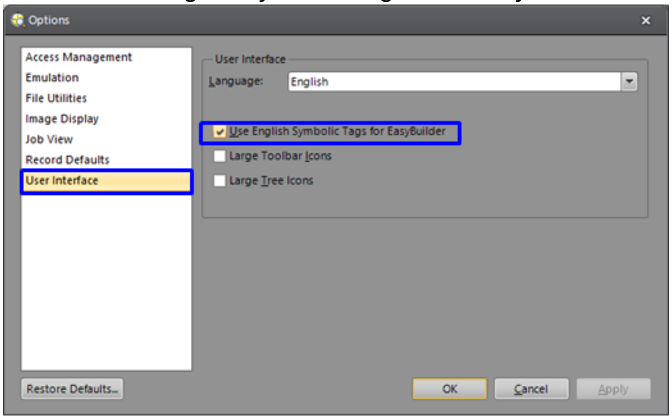
④ Teaching points setting

Set the automatic calibration.

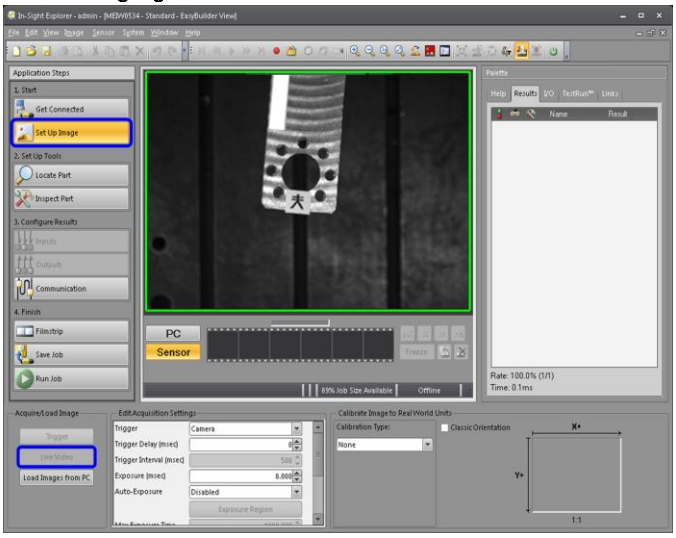
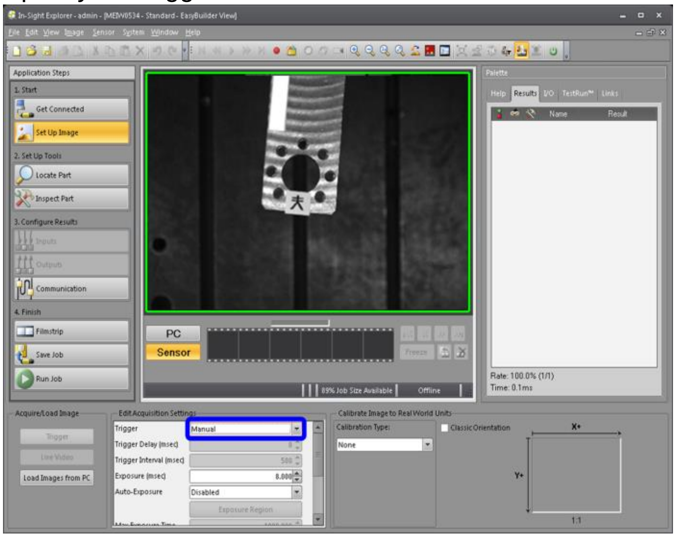
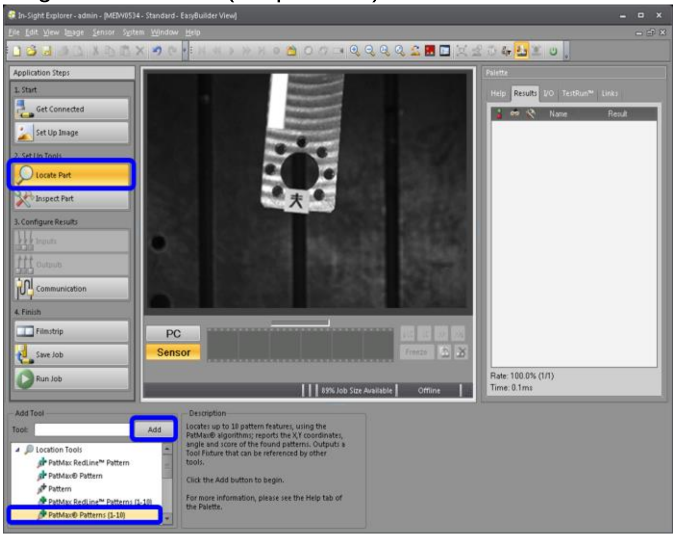
Teaching points  	Set the teaching points of the calibration. Enter the offset from the initial point and specify each teaching points. 1) By clicking the [Generating Teaching Points] button, offset values of teaching points No. 1 to No. 5 are automatically generated based on the input field of view range. 2) If you want to change the teaching point and increase / decrease the score, manually enter the values of X and Y in the list. Please set the number of teaching points of 4 points or more. Teach points with valid checks will be used for calibration.

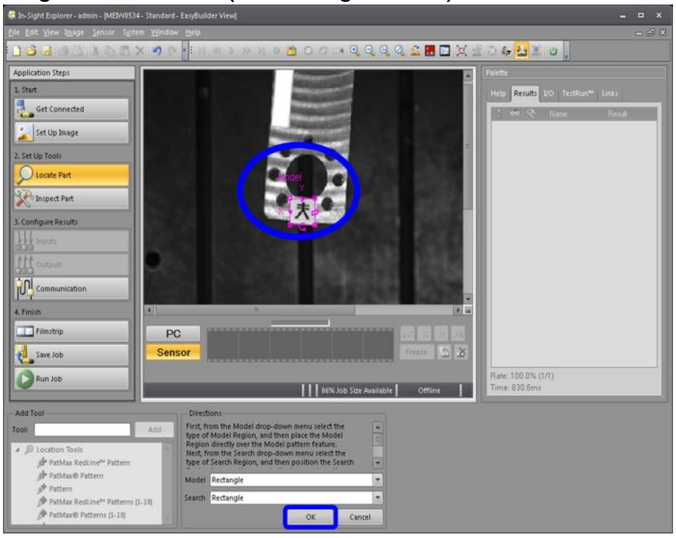
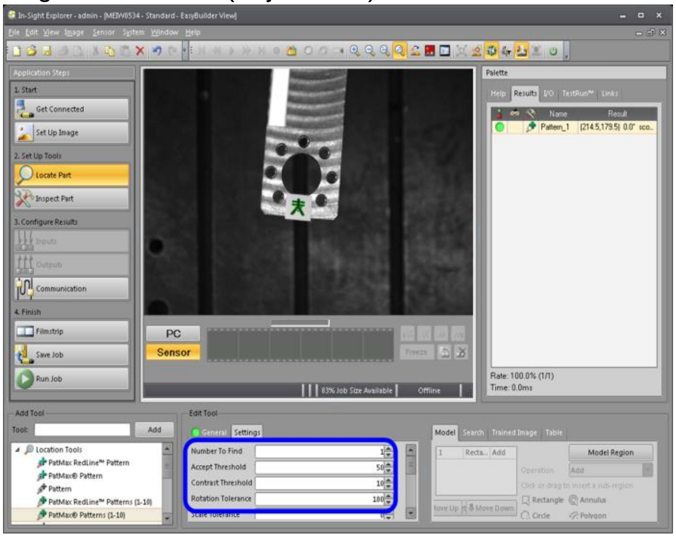
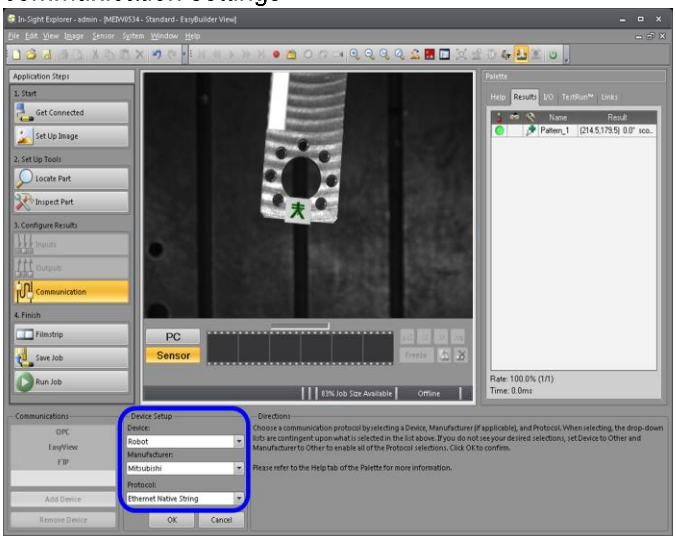
(4-4) create a vision job
Set vision job from In-Sight Explorer.

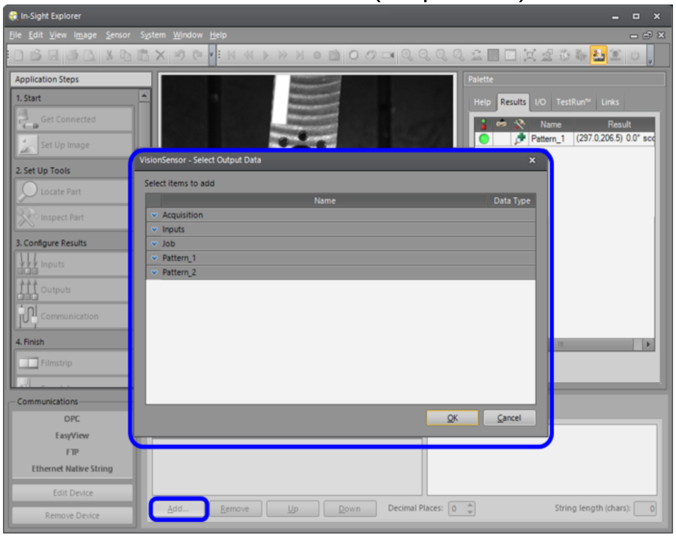
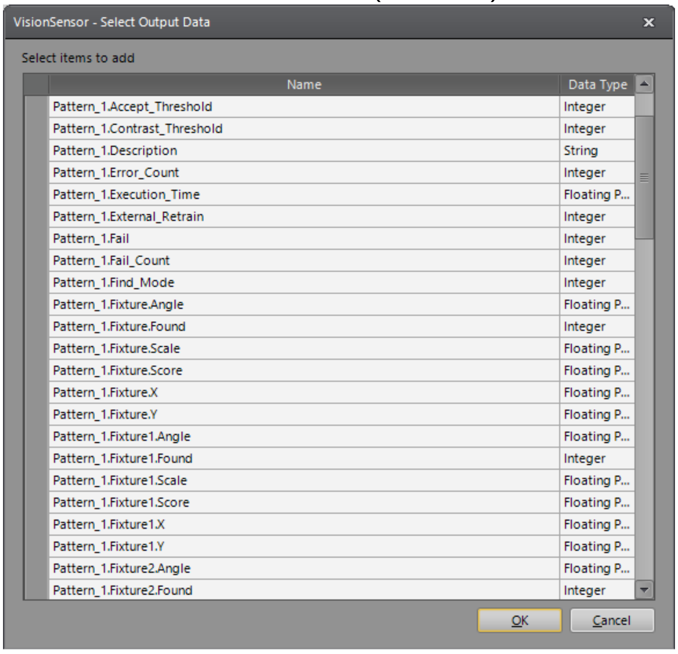
①English symbolic tag setting

Carry out work for using English symbolic tags.  	Select [System] - [Options] from the In-Sight Explorer menu.
Check "Use English symbolic tags with EasyBuilder".  	Select [User Interface] from [Options], Check the check box "Use English Symbolic Tags for EasyBuilder" and click the [OK] button.

②Create job

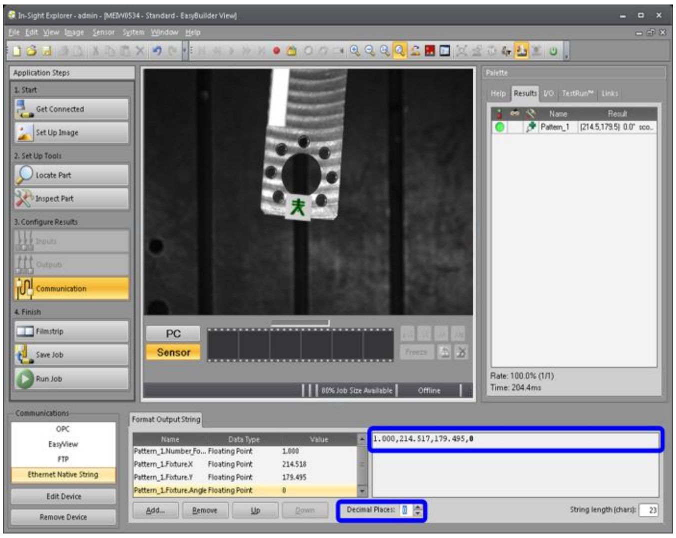
Work imaging. 	From the application step, click [Set Up Image]. Click [Live Video] and take a picture of the target tool. Click [Live Video] again to stop live video.
Specify the trigger. 	**Change "Trigger" from"Camera"to "Manual".** If you do not change, 8640 (image trigger specification abnormality) error occurs when outputting imaging request from the robot controller.
Register the work. (Preparation) 	From the application step, click [Locate Part]. Select "PatMax Pattern (1-10)" from "Add tool" and click the [Add] button.

Register the work. (Model registration) 	Move the displayed "Model" frame and enclose the work. Click the [OK] button in "Directions".
Register the work. (Adjustment) 	From "Edit Tool", click the "Settings" tab and change [Rotation Tolerance] to "180". （The work can be recognized up to ± 180 deg.） Change Accept Threshold to adjust work recognition rate. At this stage, it is fine to keep the initial value "50".
communication settings 	From the application step, click [Communication]. From "Communications", click "Add Device". From "Device Setup", select the following. Device： “Robot” Manufacturer： “Mitsubishi” Protocol： “Ethernet Native String” Click the [OK] button.

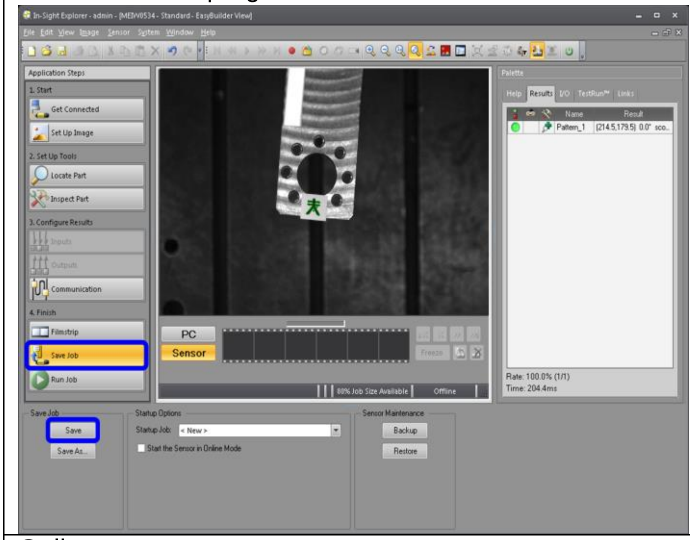
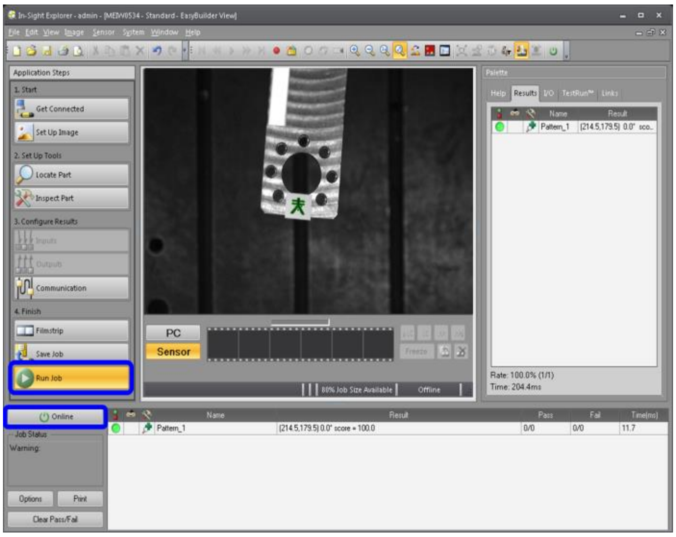
<table>
<tr>
<td>Set the communication format. (Preparation)
</td>
<td>From "Format Output String", click the [Add] button.
➔ Open the "Select Output Data".</td>
</tr>
<tr>
<td>Set the communication format. (Selection)

</td>
<td>Click the [+] mark in [Pattern_1], hold down the [Ctrl] key, and select in the following order.

(1) Pattern_1.Number_Found
(2) Pattern_1.Fixture.X
(3) Pattern_1.Fixture.Y
(4) Pattern_1.Fixture.Angle

Click the [OK] button.

Note) If the selection order is different from the above, the calibration cannot be performed correctly.</td>
</tr>
<tr>
<td>Confirm communication format
</td>
<td>Confirm the value set in the format output string. The data to be sent to the robot controller is shown on the square frame on the right.

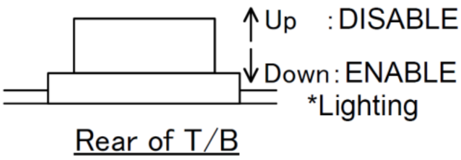
It is also possible to change the number of digits in the decimal part of the data to be transmitted.</td>
</tr>
</table>

Save the vision program.  	From the application step, click Save Job. From "Save Job", click [Save]. The name of the job to be saved is "Calibration". If the file name is different from “Calbration”, change the line of "CPRG $ =" in the robot program.
Online  	From the application step, click [Run Job]. Click "Online" above "Job Status".

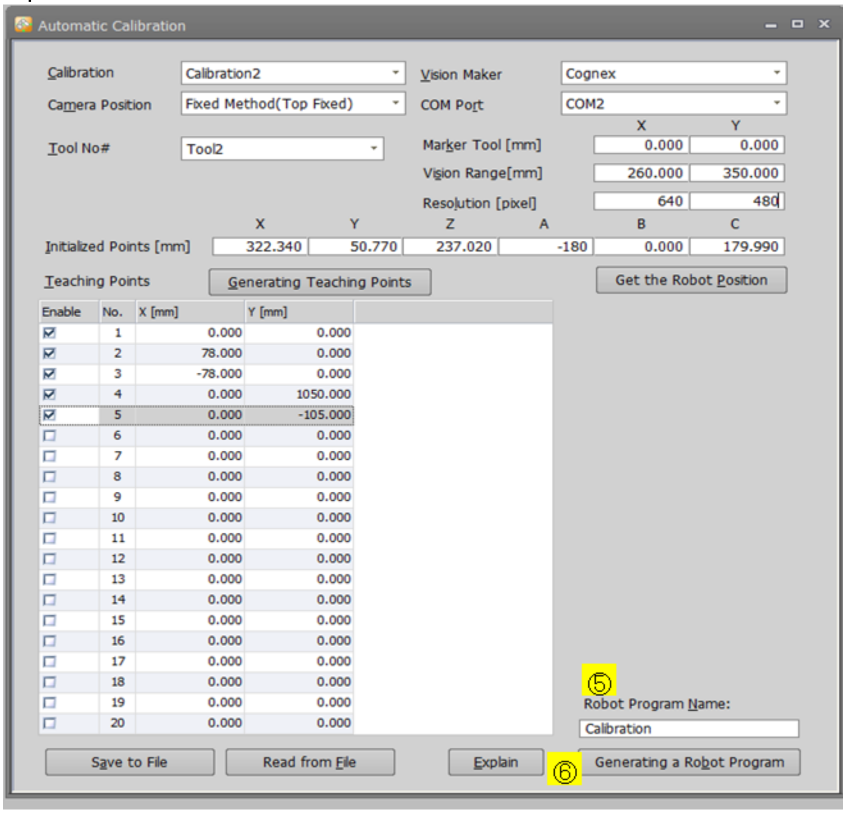
(4-5) Operation execution

Generate the robot calibration program and run it.

① Make sure there are no interfering objects within the range of motion of the robot.
② Bring the vision sensor online.
③ Push the [ENABLE] switch of T/B and disable T/B. Set the controller mode to "AUTOMATIC".



④ Open the automatic calibration screen of RT ToolBox 3.



⑤ Please enter the name in the [Robot Program Name].
⑥ Click the [Generate a Robot Program] button, the program is automatically saved in the robot controller. When saving of the program is completed, O / P starts up.

⑦ Select override.The default setting is 5%.
※Please drop the robot speed as much as possible in order to avoid the influence of vibration etc. （Override 10% or less）
⑧ Press [Start] button to start operation.
⑨ When the operation ends normally, the stop button lights up. The calculation result of the calibration is stored in the parameter VSCALB*n*. The *n* is the specified calibration number.

The operation of the robot in automatic calibration is shown below.
1) The robot's hand moves to the initialized point.
2) To calculate the camera angle and scale, the robot's hand moves from the initialized point by the value set at the second teaching point and the third teaching point.
3) In order to calculate the marker tool, the robot's hand rotates around the marker in +C axis drection on the tool coordinate in order of 0, 5, 20, 45, 90 degree.
Note) If there is a possibility of interference with the surroundings, change the angle of the C component of the position variables PAngle (1) to PAngle (5) in the robot program.
4) To calculate the calibration data, the robot's hand moves to teaching points.

⚠Caution

- When changing the generated robot program and re-executing the robot program, close the O/P and open the O/P again before executing.
- When deleting the generated robot program, please confirm that the Automatic Calibration screen is closed before deleting the robot program.
  If the Automatic Calibration screen is not closed, you may get an error saying "The program is being edited", "Cannot delete designated files".

Table 4-4 Automatic calibration’s trouble shooting

No.	Cause of and countermeasure	
9100	Error	Can not get the recognition result of the vision sensor.
	Cause	There is a possibility that the vision sensor has failed to recognize.
	Countermeasure	a) When the object to be recognized is outside the field of view of the camera ① Please change the XY component of the position variables POFS (2), POFS (3) in the program to a value located within the field of view of the camera. POFS (2) and POFS (3) mean the teaching points No. 2 and No. 3, respectively. ② Change the angle of the C component of the position variables PAngle (1) to PAngle (5) in the program to a value located within the field of view of the camera. PAngle (1) to PAngle (5) are used to calculate tool length data. b) When the object to be recognized is located within the field of view of the camera To be able to recognize the target correctly, Adjust the recognition parameters etc. of the vision sensor.
9101 to 9120	Error	*At the Nth teaching point, the recognition result of the vision sensor could not be acquired*. Please judge the teach point number N with the error number lower two digits.
	Cause	There is a possibility that the vision sensor has failed to recognize.
	Countermeasure	a) When the object to be recognized is outside the field of view of the camera Change the XY component of the position variable POFS (N) in the program to a value located within the field of view of the camera. b) When the object to be recognized is located within the field ofview of the camera

		To be able to recognize the target correctly, Adjust the recognition parameters etc. of the vision sensor.
9152	Error	Within the default retry count, It was not possible to move the recognition target to the image center (within the default value).
	Cause	Less number of retries, Or the termination condition may be strict.

	Countermeasure	a) Changing the number of retries
		Please change the description line 126 of the program. The initial value retry count is set to 10.
		If MRTRY>10 Then
		b) Change termination condition
		Please change the description line 123 of the program. The initial value is set to 2. MSCale stores the distance value [mm] per pixel.
		If MDH<=(MSCale*2) Then
		
		If the default value is small due to the resolution etc. of the vision sensor, please change it to a large value.

9153	Error	Can not calculate the tool length.
	Cause	The auxiliary point used to calculate the tool length may be incorrectly set. Please check the value of numerical variable MErr in the program. a) When MErr = -1 The auxiliary point is less than 3. b) When MErr = -2 It is not possible to calculate the tool length from the specified auxiliary point. c) When MErr = -3 Estimated error of calculated tool data is 100 mm or more.
	Countermeasure	a) When MErr = -1 To calculate the tool length, at least three points of auxiliary points are required. Please set at least 3 auxiliary points. b)When MErr = -2 There may be multiple same auxiliary points set up. Please set the auxiliary point again. c) When MErr = -3 Please set the auxiliary point again.
9154	Error	Can not calculate calibration data.
	Cause	Corresponding points may be set incorrectly. Please check the value of MErr in the program. a) When MErr = -1 The auxiliary point is less than 4. b) When MErr = -2 It is not possible to calculate calibration data from the set corresponding points.
	Countermeasure	a) When MErr = -1 To calculate the calibration data, at least 4 pairs of corresponding points are required. Please set up more than 4 pairs of corresponding points. b) When MErr = -2 There is a possibility that the corresponding point being set exists on the same straight line. Please set corresponding points so that they do not exist on the same straight line.
9155	Error	The estimation error of the calibration data exceeds the default value.
	Cause	The estimation error of the calibration data exceeds the default value.
	Countermeasure	Estimated error [mm] of calibration data is stored in numerical variable MScore in the program. If the estimation error is large, check Table 4-5 on the confirmation items at the time of execution, Please re-execute the calibration.

Table 4-5 Items to check during automatic calibration

No.	Check item	**Solution**
1	Whether the corresponding point is set correctly. For example, Whether the order of the argument of the instruction that sets the corresponding point is wrong. The calibration number is incorrect.	Please correctly set corresponding points.
2	Whether OVRD speed is too high.	Please set the OVRD speed to 10% or less.
3	Whether there is interference light.	Please block it if there is interference light.
4	Whether lens distortion does not occur in recognition image.	Perform lens distortion correction. Please use lens that lens distortion is unlikely to occur.
5	During calibration execution, Whether the heights of the vision sensor and the robot calibration marks changed.	Do not change the height between the vision sensor and the mark for robot calibration.
6	Whether the mark for robot calibration is specularly reflected and does not shine.	Please change the position to a non-specular reflection.
7	Whether the vision sensor (optical axis of the lens) is perpendicular to the operation range (calibration range).	Set the vision sensor (optical axis of the lens) is perpendicular to the operating range (calibration range).
8	Whether the origin setting of the robot is correctly set.	Please reset the origin of the robot.

(4-6) saving / reading the setting data of the automatic calibration

①Save method

Click the [Save to file] button, the automatic calibration data is saved in the form of a file.
The extension is ".acin".

②Reading method

Click the [Read from file] button, select an automatic calibration data file saved on the computer, you can read the data.
Please note that the calibration setting being edited will be cleared.

4. 3 Workpiece coordinate calibration

(1) Function Outline

Automatically obtain the positional relationship between robot coordinate and workpiece coordinate by program execution and store coordinate values in work coordinate parameters (WK1CORD to WK8CORD).
For this function, it is a prerequisite to use the vision sensor as a hand eye.

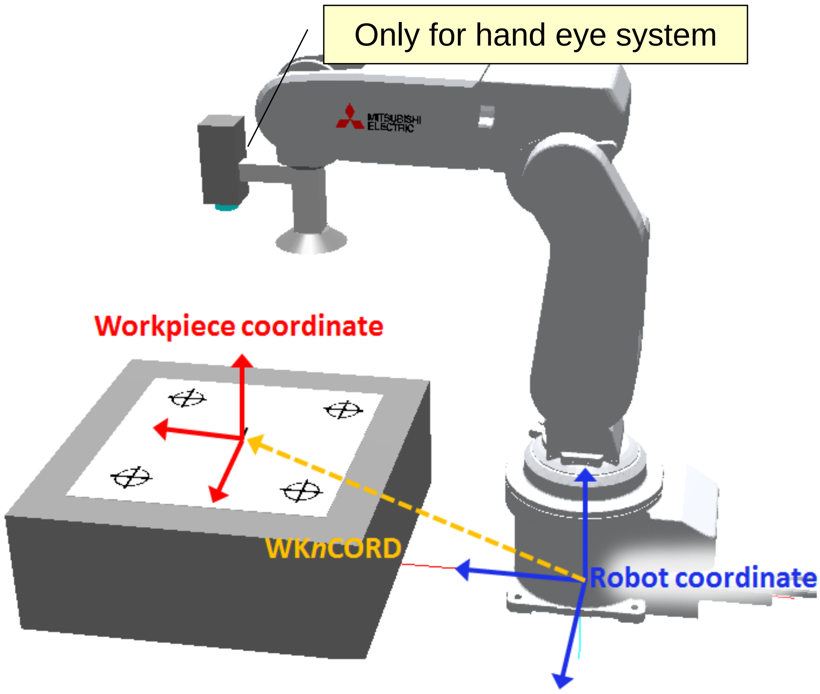


Figure 4-8 Image of work coordinate calibration

(2) Standard Specifications

Table 4-6 Standard specifications of the workpiece coordinate calibration

Items	Specifications
Robot	Vertical 6-axis robot * It can't be used with the vertical 5-axis robot, the horizontal 4-axis robot and the user robot.
Language	Only MELFA-BASIC VI
Setting method of vision sensor	Hand eye method
Output information	Position and posture of the workpiece coordinate
Number of the workpiece coordinate that can be registered	8（WK1CORD to WK8CORD）

(3) Equipment preparation

Prepare the following calibration sheet.

Set the position of each marker in work coordinates in advance.

Determine the X axis and Y axis of work coordinates with the center origin mark, the coordinate value of the cross mark is judged from the relative position.

* Please make sure that the arrangement (dimensions) of the marks is reflected in the vision field of view.

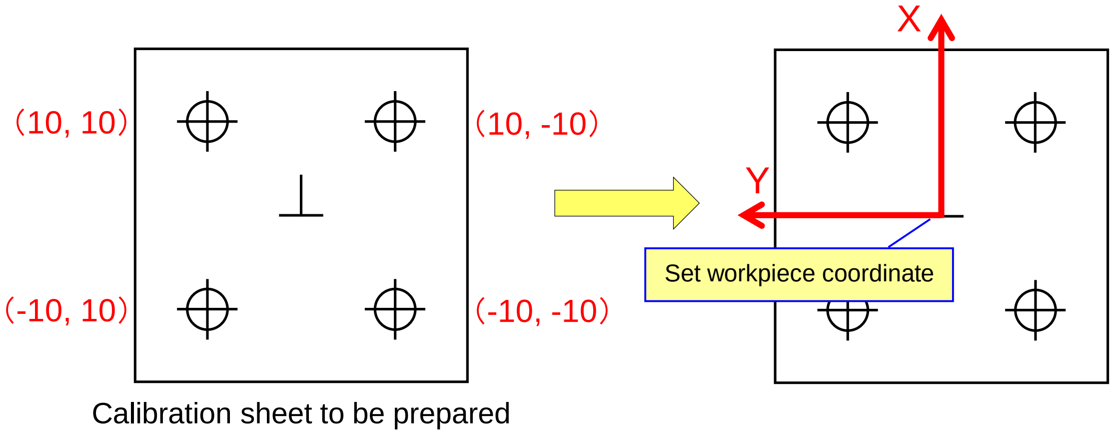


Figure 4-9 Example of the calibration sheet

(4) Workflow

Start work

↓

1. Camera setting — Refer to 4.3.(4-1)
Attach the camera to the robot.

↓

2. Implementation of hand eye calibration — Refer to 4.2
Calibrate robot and vision sensor.

↓

3. Setting of calibration sheet — Refer to 4.3.(4-2)
Place the calibration sheet on the side to set the work coordinates.

↓

4. Teach base position of robot — Refer to 4.3.(4-3)
Teach the center of the camera to the center of the calibration sheet by jog operation.

↓

5. Set condition data of robot — Refer to 4.3.(4-4)
Set robot operation condition.

↓

6. Create a vision job — Refer to 4.3.(4-5)
Set the job (program) of the vision sensor.

↓

7.Operation execution — Refer to 4.3.(4-6)
Automatically start the robot program and calculate work coordinates.

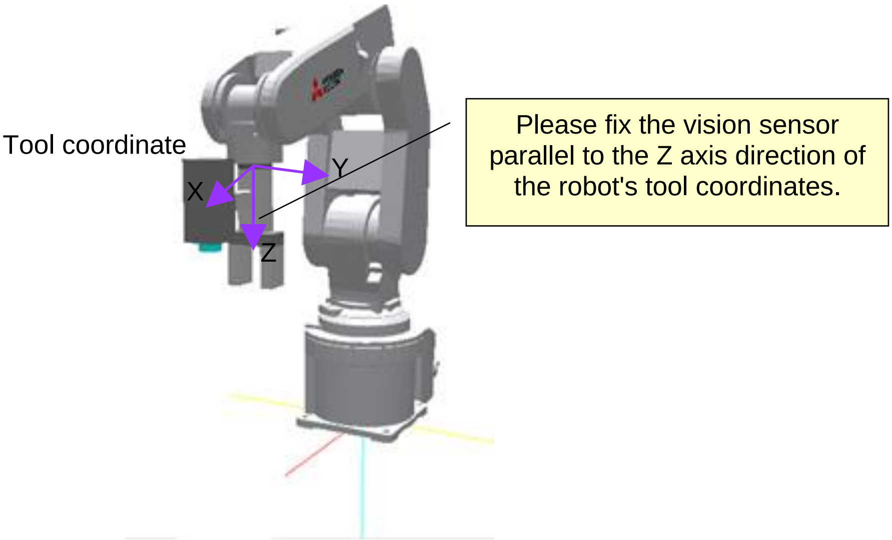
↓

8. Result — Refer to 4.3.(4-7)
Confirm whether work coordinate data was set correctly.

↓

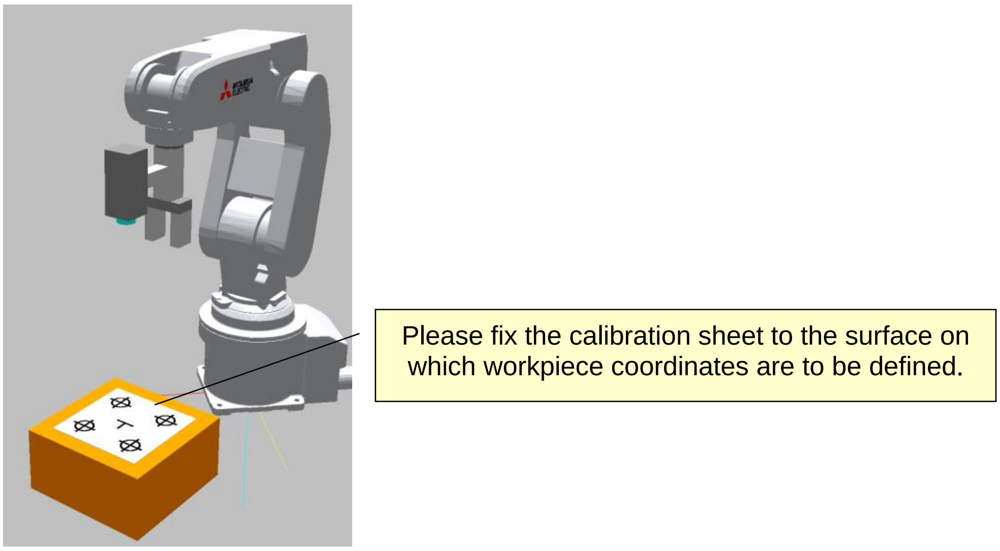
End work

(4-1) Setting of the vision sensor



(4-2) Setting of calibration sheet

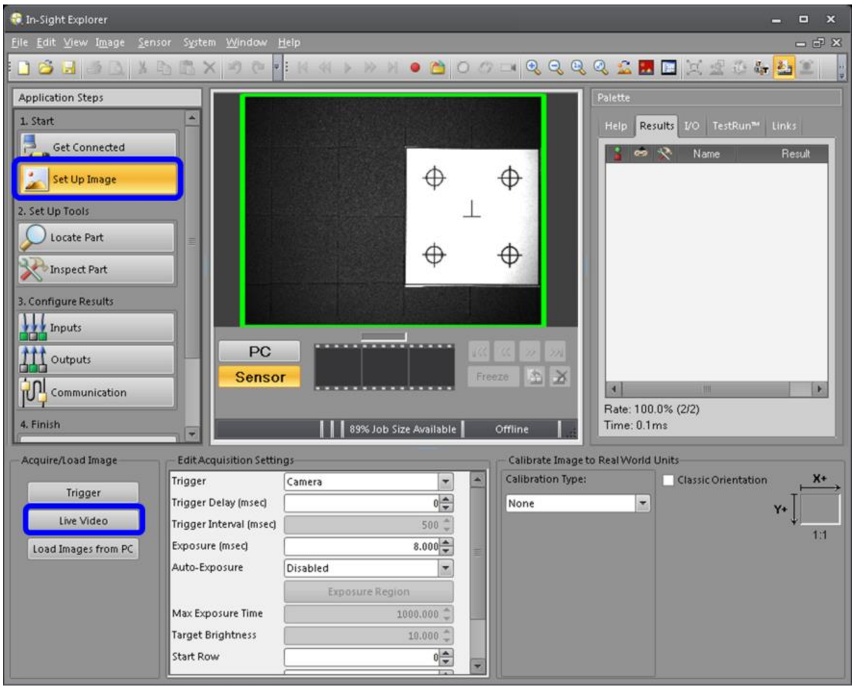
The calibration sheet is fixed to the surface on which workpiece coordinates are to be defined.



(4-3) Teach base position of robot

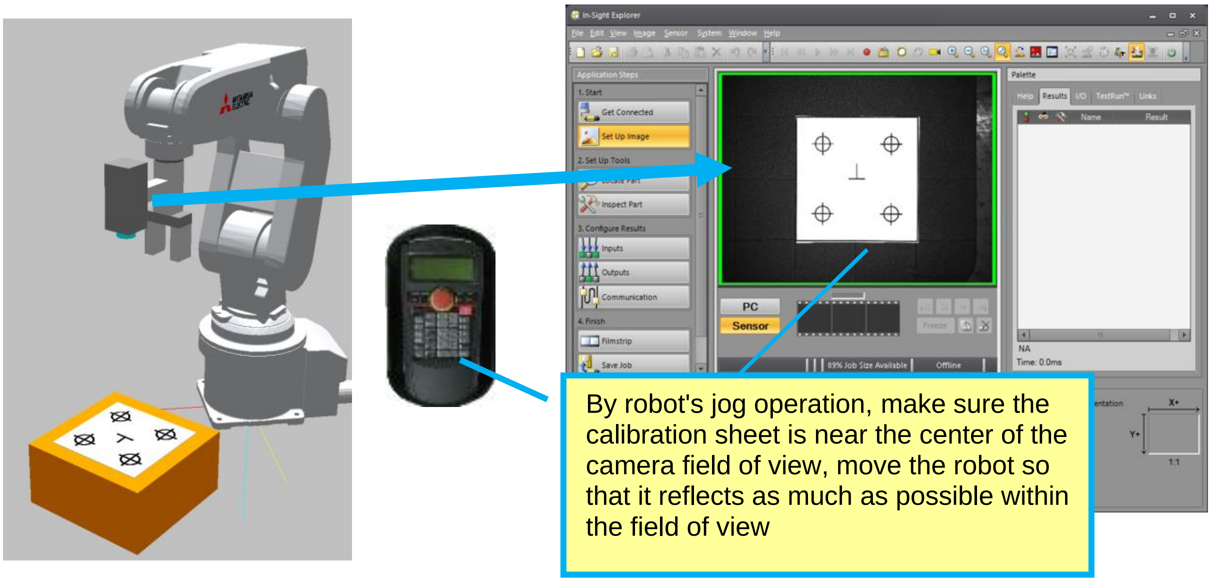
Move the robot so that the target tool is near the center of the camera field of view, teach the position of the robot.

①In-Sight Explorer application step as [Set Up Image] and select [Live Video], the camera image is displayed in the monitor.



②By robot's jog operation, make sure the calibration sheet is near the center of the camera field of view, and move the robot so that it reflects as much as possible within the field of view.

(At this time, please adjust the focus and aperture of the camera. When the camera's focus / aperture is not adjusted, Calibration may not be performed accurately.)



③Teach present position

Open the robot program "WRKCALB.prg" with T / B or RT Toolbox 3 and teach the current position of robot to the position variable "P0". The robot program "WRKCALB.prg" is in the CD-ROM.

```
<POS.> XYZ   10% P0
 X: +392.11        A: +180.00
 Y:  -83.24        B:   +0.00
 Z: +160.60        C:    0.00
L1:   +0.00       L2:   +0.00
FL1:00000007     FL2:00000000
MOVE  TEACH  123  Prev  Next
```

(4-4) Set condition data of robot

Open sample program "WRKCALB.prg" with T/B or RT ToolBox 3, please change condition data as necessary.
The condition data can be changed by setting the position variable / program change.
The position variable and program description are as follows.

① Position variable

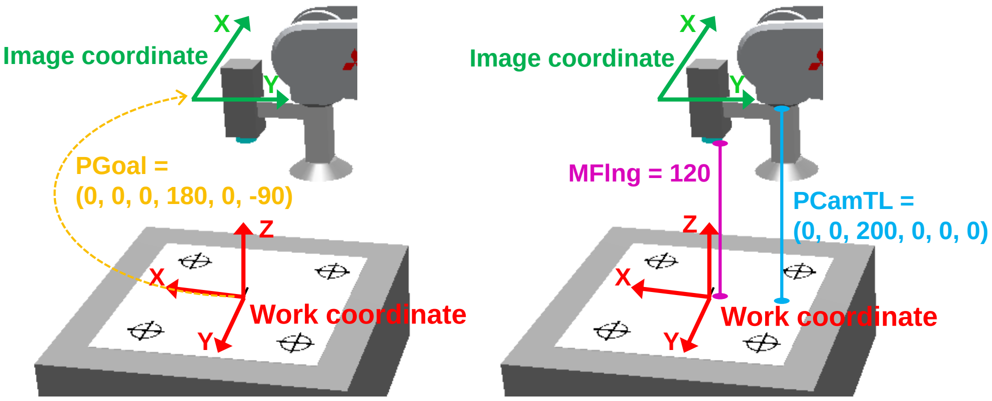
Variable name	Elements	Contents	Initial value
PVSize	X	Vision resolution (Pixel value in X direction)[pixel]	(+640, +480, 0, 0, 0, 0)
	Y	Vision resolution (Pixel value in X direction)[pixel]	
PGoal	A	Final vision sensor posture（A,B,C）[deg] (Note 1)（Please set the posture of the vision sensor in workpiece coordinate.）	(0, 0, 0, +180, +0, -90)
	B		
	C		
PCamTL	Z	Approximate distance from flange center to calibration sheet [mm]	(0, 0, +200, 0, 0, 0)
PW(1)	X	Elements in work coordinate No.1 (Note 2)	(+10, +10, 0, 0, 0, 0)
	Y		
PW(2)	X	Elements in work coordinates (Note 2)	(+10, -10, 0, 0, 0, 0)
	Y		
PW(3)	X	Elements in work coordinates (Note 2)	(-10, +10, 0, 0, 0, 0)
	Y		
PW(4)	X	Elements in work coordinates (Note 2)	(-10, -10, 0, 0, 0, 0)
	Y		
PAng	A	The movement amount at the time of calculating the work coordinate origin (X, Y, Z) (Note 3)	(0, 0, 0, +5, 0, 0)
	B		
POFS(1)	X	Camera angle, amount of movement during scale calculation (Note 2)	(+10, +0, 0, 0, 0, 0)
	Y		
POFS(2)	X	Camera angle, amount of movement during scale calculation (Note 2)	(-10, +0, 0, 0, 0, 0)
	Y		

② Program

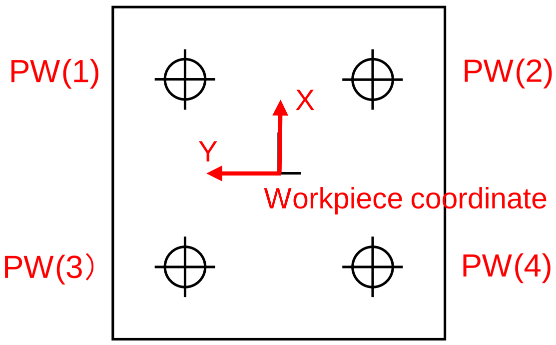
Contents	Sample program
Ethernet COM No. (Note 4)	21 CCOM$="COM2:" ' Line number setting to open line
Vision job name	22 CPRG$="WRKCALB.job" 'Set vision job name
HandEye Calibration No.	23 MHndENo = 1 'Hand Eye Calibration number
With handeye calibration Set tool No.	24 MTLCalNo = 1 'Tool No.(Distance from flange to camera center)
CalibrationNo. of image coordinates and work coordinates	25 MCalNo = 2 'Calibration Nol.(Image coordinates⇒work coordinates)
Work coordinate No.	26 MWrkNo = 1 ' Work coordinate No. of registration destination
Distance from the lens to the calibration sheet [mm] (Note 1)	27 MFlng = 120 ' Focal length [mm]
Gain value	28 MGain = 1.0 'Gain value of feedback control
Tolerance of calibration data [mm]	67 If MScore > 0.05 Then
Vision sensor No. (Note 4)	99 If M_NvOpen(1) = 1 Then
	101 Wait M_NvOpen(1)<>1
	104 NVOpen CCOM$ As #1 ' Line open + log on
	105 Wait M_NvOpen(1)=1 'Waiting logon to vision sensor
	106 NVLoad #1,CPRG$ 'Load vision program
	111 NVRun #1,CPRG$ 'Vision start
	112 EBRead #1,,MRes,PVS,MNUM,PV(1),PV(2),PV(3),PV(4) ' Acquisition of recognition result
Move the work coordinate origin to the image center	196 If MDH<=(MSCale*2) Then ' within the specified value (± 1 pixel)

Tolerance [mm] (Note 5)	
Parallel the vision sensor with the XY plane of work coordinates Tolerance [deg] (Note 4)	256 If MDH<=1.0 Then ' within the specified value (1 degree)

(Note 1) Please set PGoal, PCamTL, MFlng referring to the figure below.



(Note 2) Elements of work coordinates on the calibration sheet, please refer to as follows.



(Note 3) Mark at the center of camera view, do not leave the field of view after moving. However, to improve accuracy, please set the value to move as far as possible to the edge of the camera field of view.

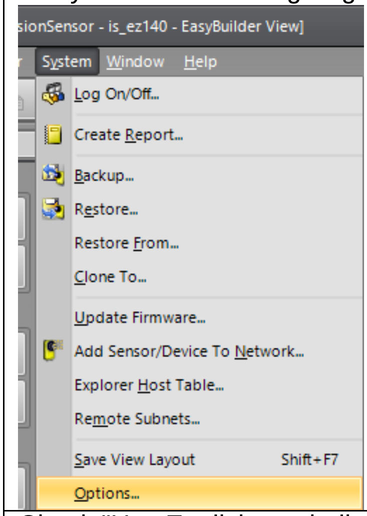
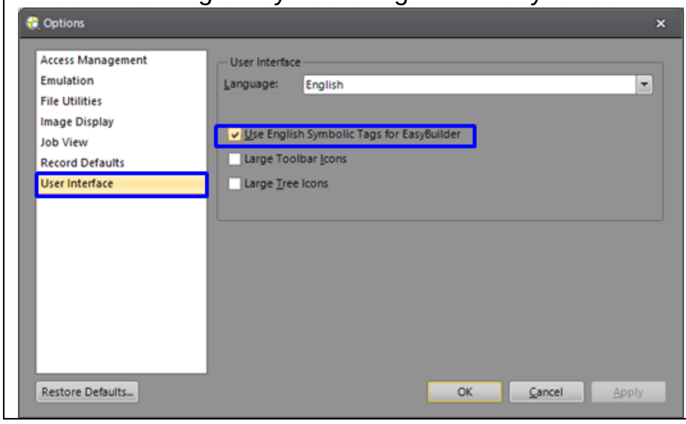
(Note 4) Another vision sensor (Including 3D vision sensor) is connected or using the Open command, Please do not duplicate numbers.

(Note 5) Tolerance after movement(initial value:MSCale*2mm).
Vision recognition mark after moving, the correction operation is repeated until it falls within this error.
If convergence does not occur even after 10 retries, error 9152 will be generated.

(4-5) create a vision job

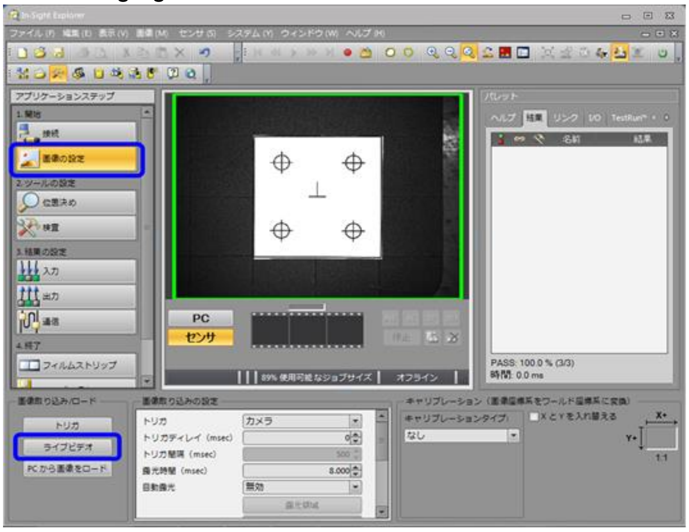
Set vision job from In-Sight Explore.

①English symbolic tag setting

Carry out work for using English symbolic tags.  	Select [System] - [Options] from the In-Sight Explorer menu.
Check "Use English symbolic tags with EasyBuilder".  	Select [User Interface] from [Options], Check "Use English Symbolic Tags for EasyBuilder", Click the [OK] button.

②Create job

Work imaging.

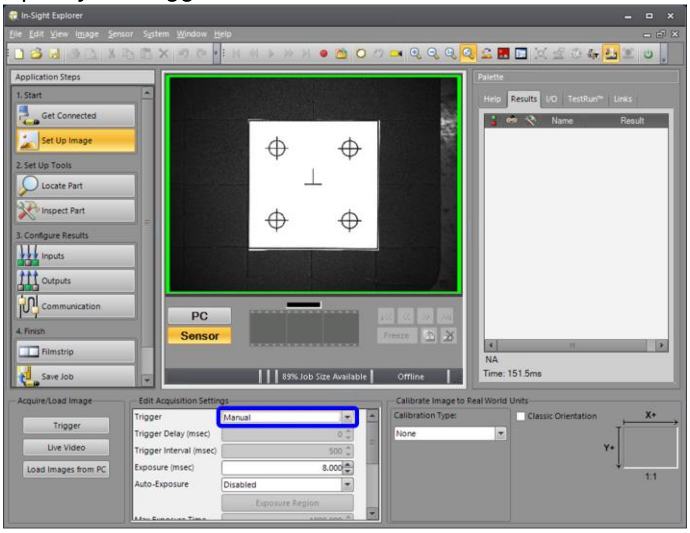


From the application step, click [Set Up Image].

Click [Live Video] and take a picture of the target tool.

Click [Live Video] again to stop live video.

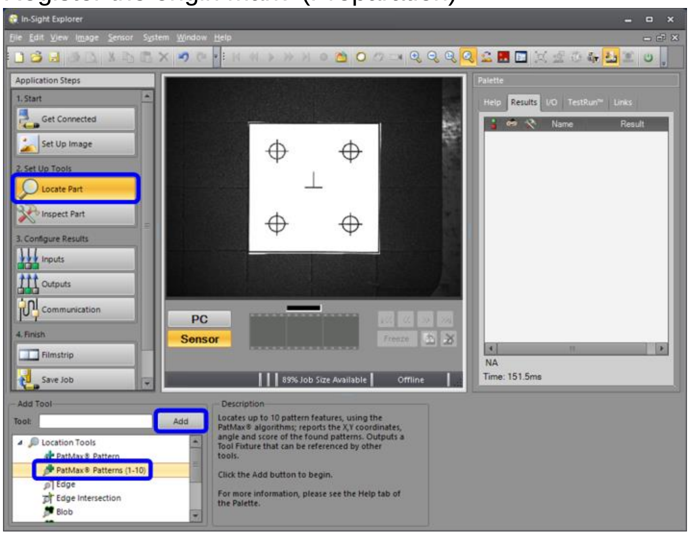
Specify the trigger.



Change "Trigger" from"Camera"to "Manual".

If you do not change, 8640 (image trigger specification abnormality) error occurs when outputting imaging request from the robot controller.

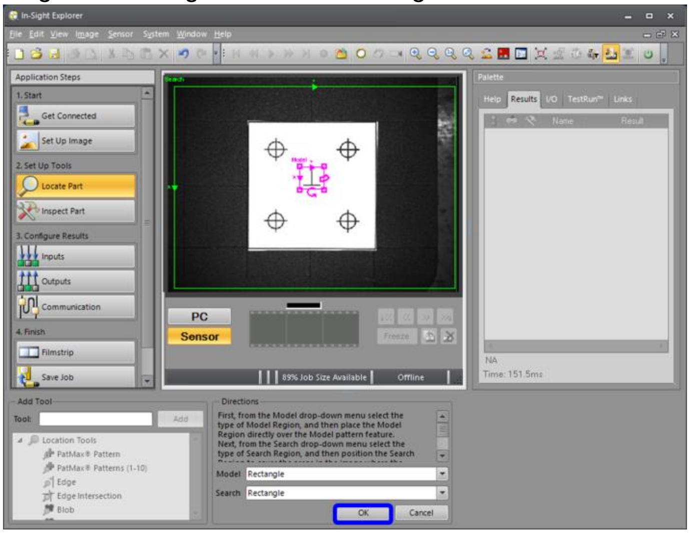
Register the origin mark. (Preparation)



From the application step, click [Locate Part].

Select "PatMax pattern (1-10)" from "Add Tool" and click the [Add] button.

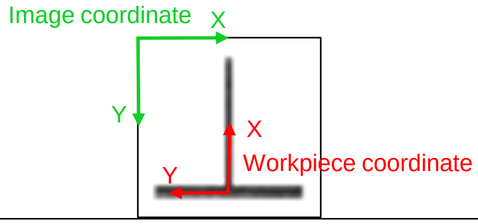
Register the origin mark.（Model register）



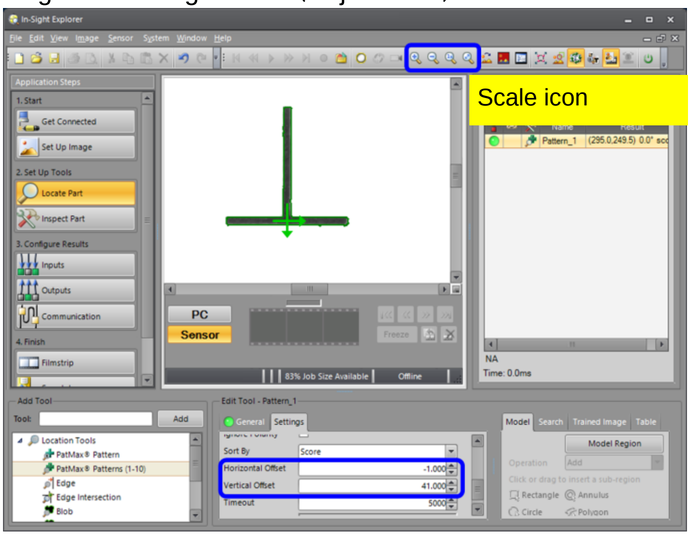
Move the displayed "model" frame and enclose the work.

Click the [OK] in "Directions".

Note) Make sure the image coordinates of the model and work coordinates have the following relationship.
When it is not the relationship below, calibration may not be performed accurately.



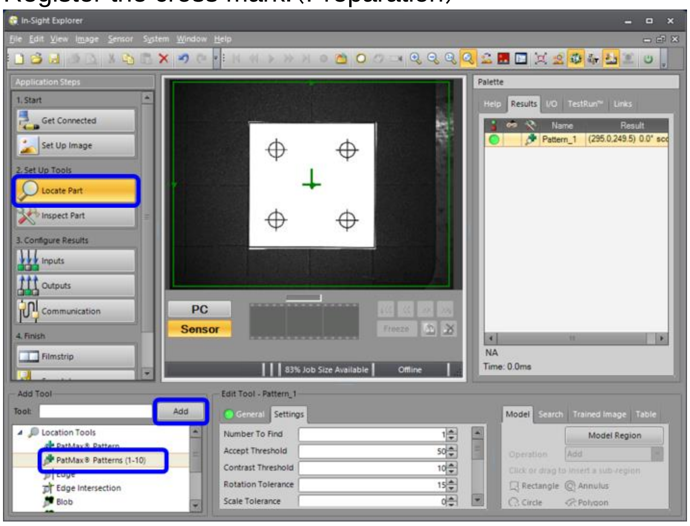
Register the origin mark.（Adjustment）



•From “Edit Tool”, Click the Settings tab, Adjust [Horizontal Offset] and [Vertical Offset], Align the output position with the corner of the origin mark.
（Enlarge or reduce the image to fit exactly as necessary）

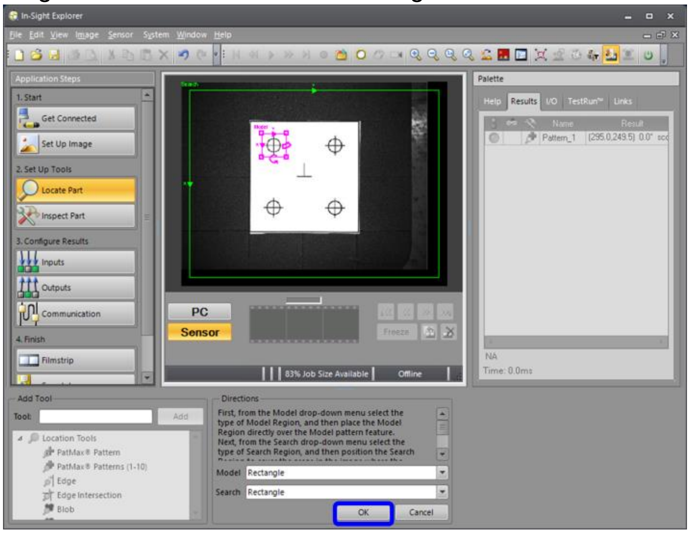
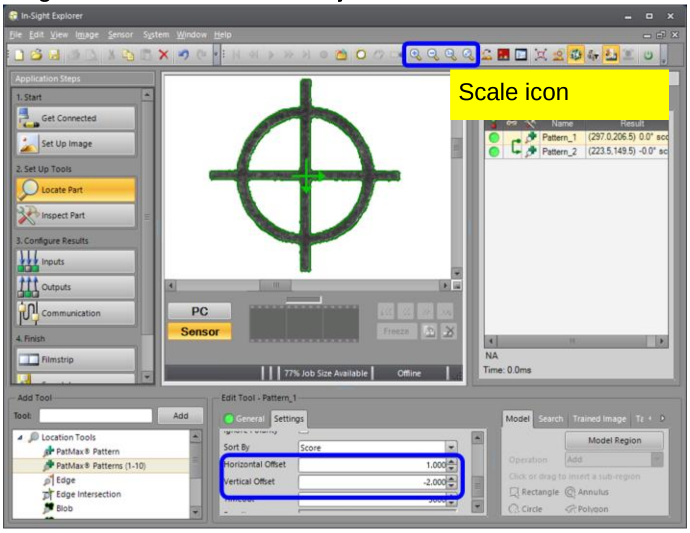
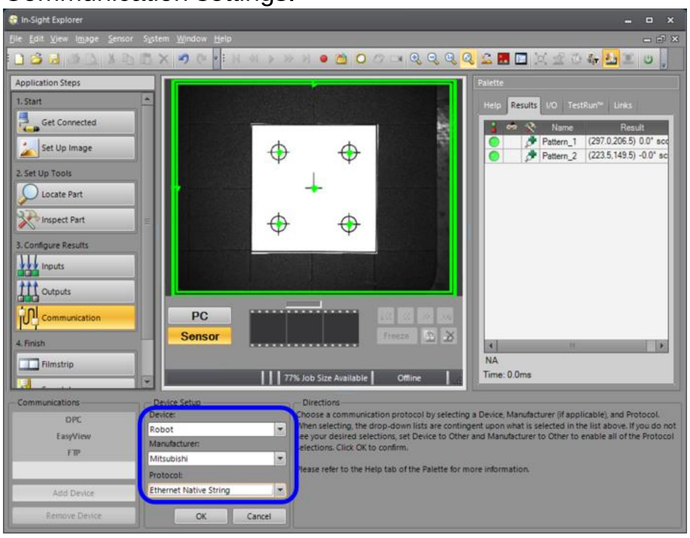
•Change Accept Threshold to adjust work recognition rate.
It is fine to keep the initial value "50".

Register the cross mark.（Preparation）

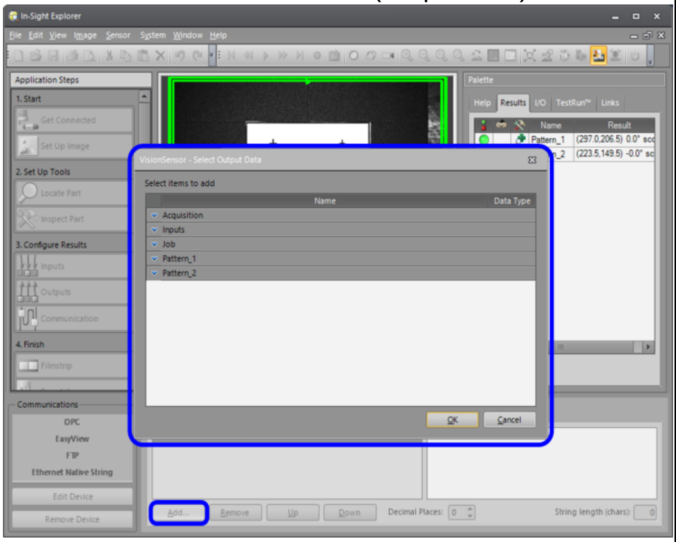


From the application step, click [Locate Part].

Select "PatMax Pattern (1-10)" from "Add Tool" and click the [Add] button.

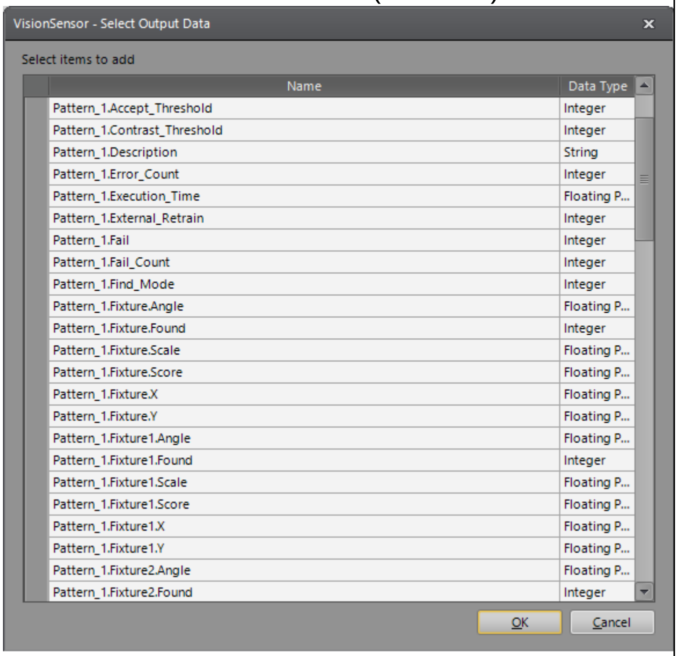
Register the cross mark.（Model register） 	Move the displayed "Model" frame and enclose the work. Click the [OK] in "Directions".
Register the cross mark.（Adjustment） 	•From “Edit Tool”, Click the Settings tab, Adjust [Horizontal Offset] and [Vertical Offset], Align the output position with the cross mark. （Enlarge or reduce the image to fit exactly as necessary） •Change Accept Threshold to adjust work recognition rate. It is fine to keep the initial value "50". •Set [Number to Find] to "4".
Communication settings. 	From the application step, click [Communication]. From "Communications", click "Add Device". From "Device Settings", select the following. ・Device：“Robot” ・Manufacturer：“Mitsubishi” ・Protocol：“Ethernet Native String” Click the [OK] button.

<table>
<tr>
<td>
Set the communication format. (Preparation)



</td>
<td>
From "Format Output String", click the [Add] button.

➔ Open the "Select Output data".
</td>
</tr>
<tr>
<td>
Set the communication format. (Selection)

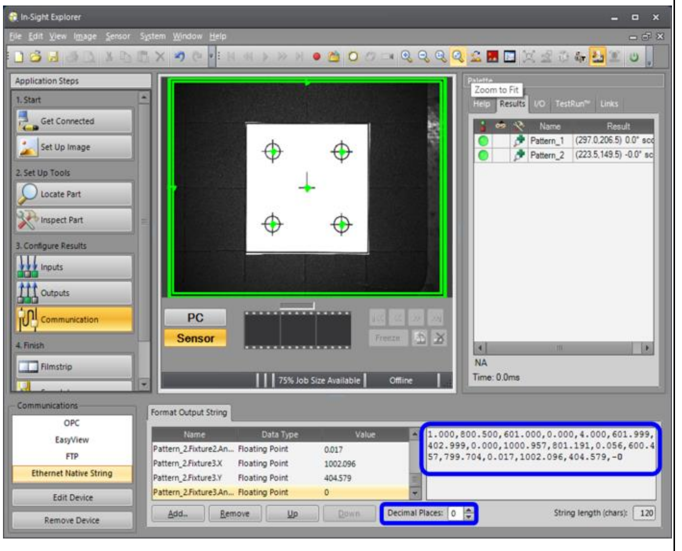


</td>
<td>
Click the [+] mark in [Pattern_1] and [Pattern_2], hold down the [Ctrl] key, and select in the following order.

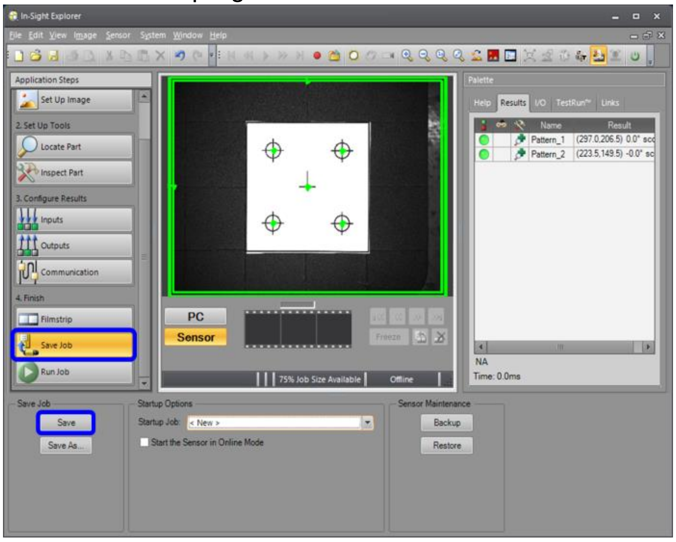
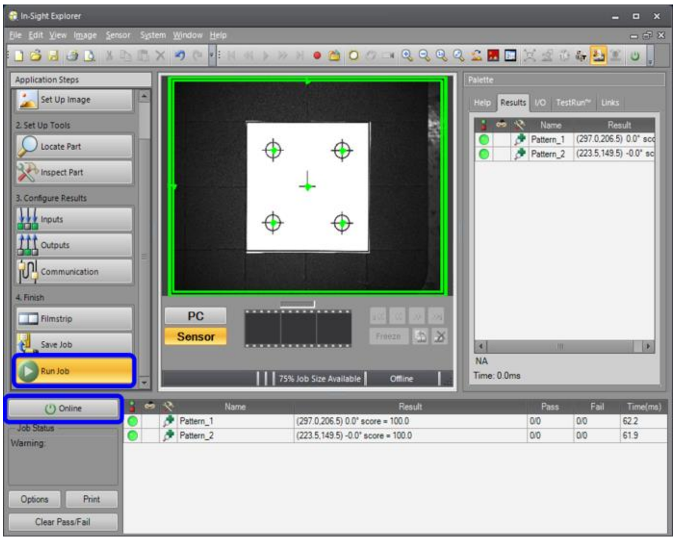
(1) Pattern_1.Number_Found
(2) Pattern_1.Fixture.X
(3) Pattern_1.Fixture.Y
(4) Pattern_1.Fixture.Angle
(5) Pattern_2.Number_Found
(6) Pattern_2.Fixture.X
(7) Pattern_2.Fixture.Y
(8) Pattern_2.Fixture.Angle
(9) Pattern_2.Fixture1.X
(10) Pattern_2.Fixture1.Y
(11) Pattern_2.Fixture1.Angle
(12) Pattern_2.Fixture2.X
(13) Pattern_2.Fixture2.Y
(14) Pattern_2.Fixture2.Angle
(15) Pattern_2.Fixture3.X
(16) Pattern_2.Fixture3.Y
(17) Pattern_2.Fixture3.Angle

Click the [OK] button.

Note) If the selection order is different from the above, the calibration cannot be performed correctly.
</td>
</tr>
</table>

<table>
<tr>
<td>Confirm communication format

</td>
<td>Confirm the value set in the format output string. The data to be sent to the robot controller is shown on the square frame on the right.

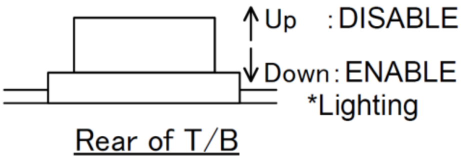
It is also possible to change the number of digits in the decimal part of the data to be transmitted.</td>
</tr>
</table>

Save the vision program.  	From the application step, click [Save Job]. From "Save Job", click [Save]. The name of the job to be saved is "WRKCALB". If the file name is different from “WRKCALB”, change the line of "CPRG $ =" in the robot program.
Online  	From the application step, click [Run Job]. Click "Online" above "Job Status".

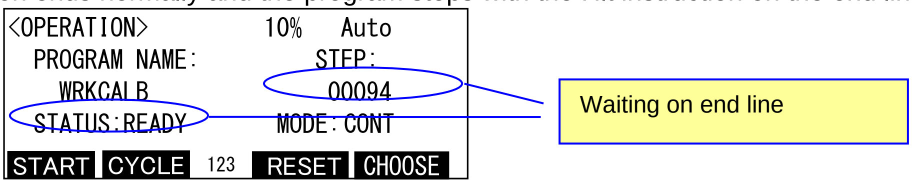
(4-6) **Operation execution**

Select the robot's calibration program and run it.

①Make sure there are no interfering objects.
②Online the vision sensor.
③Push the [ENABLE] switch of T/B and disable T/B. Set the controller mode to "AUTOMATIC".



④Select Override in T/B.
* In order to avoid the influence of vibrations etc and to make operation even a little accurate, Please drop the robot speed. (Override 10% or less)
⑤Select the calibration program (program: WRKCALB.prg) at T/B.
⑥Start the program at T/B.
⑦Operation ends normally and the program stops with the Hlt instruction on the end line.



The operation of the robot in the workpiece coordinate calibration is shown below.

1) The robot's hand moves to the initialized point.
2) To calculate the camera angle and scale, the robot's hand moves from the initialized point by the value set at the position variables POFS(1) and POFS(2).
3) The robot's hand moves so that the calibration sheet and the vision sensor are parallel.
The parallel pose of the vision sensor is the position variable PGoal.
The PGoal means the pose of the vision sensor in workpiece coordinate.
4) In order to calculate the position of the workpiece coordinate origin, the robot's hand moves by the set position variables PAng.

Table 4-7 Troubleshooting workpiece coordinate calibration

No.	Cause and countermeasure	
9101	Error	Can not get the recognition result of the vision sensor. （Recognition target：Origin mark of workpiece coordinates）
	Cause	There is a possibility that the vision sensor has failed to recognize.
	Countermeasure	a) When the object to be recognized is outside the field of view of the camera Please change the XY component of the position variables POFS (1), POFS (2) in the program to a value located within the field of view of the camera. b) When the object to be recognized is located within the field ofview of the camera To be able to recognize the target correctly, Adjust the recognition parameters etc. of the vision sensor.
9102	Error	Can not get the recognition result of the vision sensor. (Recognition target：Teaching mark of workpiece coordinates)
	Cause	The vision sensor may not recognize the four teach points.
	Countermeasure	a) When the object to be recognized is outside the field of view of the camera ① Make sure that the Z component of the position variable PCamTL in the program is correctly entered. ② Please check whether the XY component of the position variable PCamTL in the program is correctly entered. When the value is incorrect, Please perform hand eye

		calibration again. Now, The XY component means the tool length from the flange to the center of the camera. b) When the object to be recognized is located within the field of view of the camera To be able to recognize the target correctly, Adjust the recognition parameters etc. of the vision sensor.
9130	Error	Within the default retry count, The vision sensor could not be moved in parallel (within the default value) to the XY plane of workpiece coordinates.
	Cause	Less number of retries, Or the termination condition may be strict.

	Countermeasure	a) Changing the number of retries
		Please change the description line 259 of the program. The initial value retry count is set to 10.
		If MRTRY>10 Then
		b) Change termination condition
		Please change the description line 123 of the program. The initial value is set to 1.0[deg].
		If MDH<=1.0 Then
		
		If the default value is small due to the resolution etc. of the vision sensor, please change it to a large value.

9131	Error	Can not calculate the coordinates of the workpiece coordinate origin.
	Cause	Can not calculate the point (intersection) of the two position data.
	Countermeasure	Please set up two position data that can calculate the point (intersection).
9152	Error	Within the default retry count, It was not possible to move the recognition target to the image center (within the default value).
	Cause	Less number of retries, Or the termination condition may be strict.

	Countermeasure	a) Changing the number of retries
		Please change the description line 199 of the program. The initial value retry count is set to 10.
		If MRTRY>10 Then
		b) Change termination condition
		Please change the description line 123 of the program. The initial value is set to 2. Addisionally, MSCale stores the distance value [mm] per pixel.
		If MDH<=(MSCale*2) Then
		
		If the default value is small due to the resolution etc. of the vision sensor, please change it to a large value.

9154	Error	Can not calculate the calibration data.
	Cause	Corresponding points may be set incorrectly. Please check the value of MErr in the program. a) When MErr = -1 The auxiliary point is less than 4. b) When MErr = -2 It is not possible to calculate calibration data from the set corresponding points.
	Countermeasure	a) When MErr = -1 To calculate the calibration data, at least 4 pairs of corresponding points are required. Please set up more than 4 pairs of corresponding points. b) When MErr = -2 There is a possibility that the corresponding point being set exists on the same straight line. Please set corresponding points so that they do not exist on the same straight line.
9155	Error	The estimation error of the calibration data exceeds the default value.
	Cause	The estimation error of the calibration data exceeds the default value.
	Countermeasure	Estimated error [mm] of calibration data is stored in numerical

		variable MScore in the program. If the estimation error is large, check Table 4-5 on the confirmation items at the time of execution, Please re-execute the calibration.

(4-7) Result

When the calibration is completed normally, please make sure of the following as a precaution.

1) Make sure the workpiece coordinate data is set in the robot.

a) Select <Online> - <Parameter> - <Operation Parameter> - <Work Coordinate> to display the work coordinate parameter.

b) Make sure the work coordinate data has been updated.

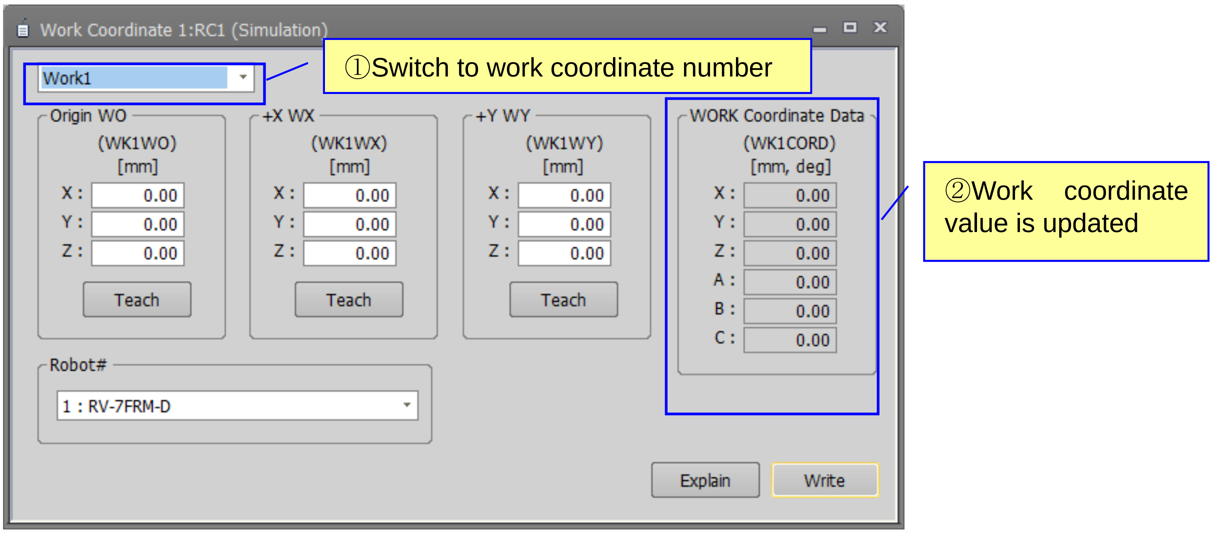


Figure 4-10 update of workpiece coordinates

(5) Sample program (WRKCALB.prg)

```
'#################################################################
'Workpiece coordinate calibration program
'#################################################################
'
Dim POFS(2)
Dim PW(4), PV(4), PVtmp(4)
'
'### Initialization processing ###
*Init
  P0 = P0 ' Teaching point
  PVSize = PVSize   ' resolution (X, Y)
  PGoal = PGoal     ' End condition (Posture of vision sensor in workpiece coordinates)
  PW(1) = PW(1)
  PW(2) = PW(2)
  PW(3) = PW(3)
  PW(4) = PW(4)
  PAng = PAng        'The posture of the second point used when estimating the work coordinate origin
  POFS(1) = POFS(1)
  POFS(2) = POFS(2)
'
  CCOM$ = "COM2:"   'Open the line and set the line number
  CPRG1$ = "WRKCALB.job" 'Set vision job
  MHndENo = 1   'Hand eye calibration number
  MTLCalNo = 1 'Tool number (Distance from flange to camera center)
  MCalNo = 2     'Calibration number (Image coordinates -> Workpiece coordinates)
  MWrkNo = 1      ' Workpiece coordinate number of registration destination
  MFlng = 120   ' Focal length [mm]
  MGain = 1.0   'Gain value of feedback control
'
  PVCenter = P_Zero   ' Set image center
  PVCenter.X = PVSize.X/2         'X
  PVCenter.Y = PVSize.Y/2         'Y
  PT2V = P_Zero
  Ptmp = PVSCal( MHndENo, 0, 0, 0 )
  PT2V.C = Ptmp.C   'Vision sensor mounting error
'#####     Main Program     #####
*main
  PBTL = P_Tool
  Mov P0 'Move to teaching point
  Tool P_NTool 'Tool initialization
  PC0 = P_Fbc   ' Get current position
  '
  MFLG = 0
  GoSub *SVSOPEN 'Open line of vision sensor
  GoSub *SCalcAngleScale         'Calculate camera angle / scale
' Tool settings
  M_Tool = MTLCalNo
  PTL = P_Tool
  Tool P_NTool
  PCamTL.X = PTL.X
  PCamTL.Y = PTL.Y
'
  MPRMA = MFlng/MScale ' Internal parameter A
  MPRMB = MFlng/MScale ' Internal parameter B
  MPRMC = PVCenter.X     ' Internal parameter C
  MPRMD = PVCenter.Y     ' Internal parameter D
'
  PCS = PC0                    ' Set initial position
  GoSub *SMovCenter      'Move mark to camera center
  Tool PCamTL 'Set camera tool
  PC0 = P_Fbc ' Get current position
```

```
'
   MFLG = 1
   GoSub *PrlVSWRK 'Parallelizing the vision sensor to the XY plane of work coordinates
   MScore = M_VSCErr(MCalNo) 'Accuracy evaluation of calibration data
   If MScore > 0.05 Then
      Error 9155
   EndIf
'
' Calculate posture of workpiece coordinates
   PRob = P_Fbc
   PRob2Wrk = PRob * PT2V * Inv(PVSPose)
'
   Tool PTL    'Set tool as camera center
'Calculate the origin position of workpiece coordinates
   MFLG = 0
   PCS = P_Fbc          ' Set initial position
   GoSub *SMovCenter          'Move mark to camera center
   P1 = P_Fbc ' Get current position
'
   PCS = P1 * PAng
   GoSub *SMovCenter          'Move mark to camera center
   P2 = P_Fbc ' Get current position
'
   MErr = PCalObtP(P1,P2,Pres) ' intersection exists
   If MErr <> 0 Then
      Error 9131
   EndIf
   PRob2Wrk.X = Pres.X
   PRob2Wrk.Y = Pres.Y
   PRob2Wrk.Z = Pres.Z
   '
   P_WkCord(MWrkNo) = PRob2Wrk   'workpiece coordinate registration
   '
   Tool PBTL
   Hlt
End
'
'###   Vision line open   ###
*SVSOPEN
   If M_NvOpen(1) = 1 Then
      NVClose     'line close
      Wait M_NvOpen(1) <> 1
   EndIf
   Dly 1
   NVOpen CCOM$ As #1       ' Line open + log on
   Wait M_NvOpen(1) = 1   'Waiting logon to vision sensor
   NVLoad #1,CPRG1$         'Load vision program
Return
'
'### Vision recognition ###
*VSTRG
   NVRun #1,CPRG1$   'Vision start
   EBRead #1,,MRes,PVS,MNUM,PV(1),PV(2),PV(3),PV(4)   ' Acquisition of recognition result
   If MRes <> 1 Then
      Error 9101
      GoTo *VSTRG
   EndIf
   If MFLG = 1 Then
      If MNUM <> 4 Then
         Error 9102
         GoTo *VSTRG
      EndIf
      GoSub *VSResSort
   EndIf
```

```
Return
'
'ecognition result of vision and Correspondence of workpiece coordinate
*VSResSort
  ' X axis linear equation (ax+by+c=0)
  Ma = Cos(PVS.C)
  Mb = Sin(PVS.C)
  Mc = -Ma*PVS.X - Mb*PVS.Y
  ' Y axis linear equation (-bx+ay+d=0)
  Md = Mb*PVS.X - Ma*PVS.Y
  '
  For M1=1 To 4 Step 1
    Ptmp = PV(M1)
    If (Ma*Ptmp.X + Mb* Ptmp.Y + Mc) < 0 Then
        If (-Mb*Ptmp.X + Ma* Ptmp.Y + Md) < 0 Then
            PVtmp(1) = Ptmp              ' ax+by+c < 0 && -bx+ay+d < 0
        Else
            PVtmp(3) = Ptmp              ' ax+by+c < 0 && -bx+ay+d >= 0
        EndIf
    Else
        If (-Mb*Ptmp.X + Ma* Ptmp.Y + Md) < 0 Then
            PVtmp(2) = Ptmp              ' ax+by+c >= 0 && -bx+ay+d < 0
        Else
            PVtmp(4) = Ptmp              ' ax+by+c >= 0 && -bx+ay+d >= 0
        EndIf
    EndIf
  Next M1
    '
  For MI=1 To 4 Step 1
      PV(MI) = PVtmp(MI)
  Next MI
Return
'
'### Camera angle / scale calculation ###
*SCalcAngleScale
'//First point shooting
  Mvs PC0*POFS(1)
  Dly 0.5
  GoSub *VSTRG     ' Image acquisition, check the mark.Assign mark position to PV
  PVS1=PVS
'//Second point shooting
  Mvs PC0*POFS(2)
  Dly 0.5
  GoSub *VSTRG     ' Image acquisition, check the mark.Assign mark position to PV
  PVS2=PVS
'//Camera angle calculation (optical axis rotation)
  MRDX=POFS(2).X-POFS(1).X
  MRDY=POFS(2).Y-POFS(1).Y
  MVDX=PVS1.X-PVS2.X
  MVDY=PVS1.Y-PVS2.Y
  MVSC=Atn2(MRDY,MRDX)-Atn2(MVDY,MVDX)
  PVSC=P_Zero
  PVSC.C=MVSC
'//Scale value calculation（Distance per pixel）
  MPL=Sqr(MVDX*MVDX+MVDY*MVDY)
  MRL=Sqr(MRDX*MRDX+MRDY*MRDY)
  MScale=MRL/MPL
Return
'
'### Move mark to camera center ###
*SMovCenter
'// Center position imaging
  MRTRY = 0                        'Retry counter initialization
  MFIN = 0                        ' Initialize end flag
```

```
    PC = PCS
    While MFIN<>1
      Mov PC Type 0,0
      Dly 0.5
      GoSub *VSTRG
      PVSH = PVS
      GoSub *SCalcCenter           ' Calculate the correction amount to the center position
      PC = PC*PH
      If MDH <= (MScale*2) Then     ' Within the specified value (Plus or minus 1 pixel)
        MFIN = 1                    'end flag
      Else
        If MRTRY > 10 Then           'Retry more than 10 times
          Error 9152
          MRTRY = 0
        Else
          MRTRY = MRTRY+1
        EndIf
      EndIf
    WEnd
    PCE = PC
Return
'
'### From vision recognition result (PVSH) to center position (PVCenter) of Corrected coordinates (PH),
Calculate the correction distance (MDH) ###
*SCalcCenter
    PHPXY = P_Zero                  'Correction amount in the vision coordinate (pixel)
    PHPXY.X = PVSH.X-PVCenter.X
    PHPXY.Y = PVSH.Y-PVCenter.Y
    PHP = PVSC*PHPXY                 ' Correction amount in the tool coordinate (pixel)
    PH = P_Zero                     ' Correction amount in the tool coordinate (mm)
    PH.X = PHP.X*MScale
    PH.Y = PHP.Y*MScale
    MDH = Sqr(PH.X*PH.X+PH.Y*PH.Y)
Return
'
'### Calculate calibration data to convert to workpiece coordinate from image coordinate ###
*CalVision2Work
    GoSub *VSTRG
    VSCalClr MCalNo     'Reset point for calibration
    For M1 = 1 To 4
      VSSetCP MCalNo, M1, PV(M1), PW(M1)    'Set vision coordinates and workpiece coordinate
    Next M1
    MRes = VSRegCD MCalNo    ' Register calibration data to convert to workpiece coordinate from image
coordinate
    If MRes <> 0 Then
      Error 9154
    EndIf
Return
'
'###Parallelizing the plane of the vision sensor and the workpiece ###
*PrlVSWRK
    '// Loop until it becomes parallel
    MRTRY = 0                       'Retry counter initialization
    MFIN = 0                         ' Initialize end flag
    PTrg = PC0
    While MFIN<>1
      Mov PTrg Type 0,0
      GoSub *CalVision2Work
      PVSPose = PVSDrct( MCalNo, MPRMA, MPRMB, MPRMC, MPRMD ) 'Posture of vision sensor in
workpiece coordinate
      PVSMov = Inv(PVSPose) * PGoal
      MDH = Abs(Deg(PVSMov.A)) + Abs(Deg(PVSMov.B)) + Abs(Deg(PVSMov.C))
      'Convert movement amount in workpiece coordinate to robot coordinate
      PRMov = P_Zero
```

```
        PRMov.A = PVSMov.B * MGain
        PRMov.B = PVSMov.A * MGain
        PRMov.C = PVSMov.C * MGain
        PRMov = PT2V * PRMov * Inv( PT2V )
        PTrg = PTrg * PRMov
        If MDH <= 1.0 Then      ' Within the specified value (1.0 degree)
          MFIN = 1                      'end flag
        Else
          If MRTRY > 10 Then            'Retry more than 10 times
            Error 9130
            MRTRY = 0
          Else
            MRTRY = MRTRY+1
          EndIf
        EndIf
    WEnd
Return
```

4.4 Inter-robot relational calibration

(1) Function Outline

"Inter-robot relational calibration" can determine the positional relationship between robots by defining the same workpiece coordinate with multiple robots. “Collision avoidance function”and“Cooperative operation function” is easy to set the required position relationship between robots. (CR800-R/Q only)

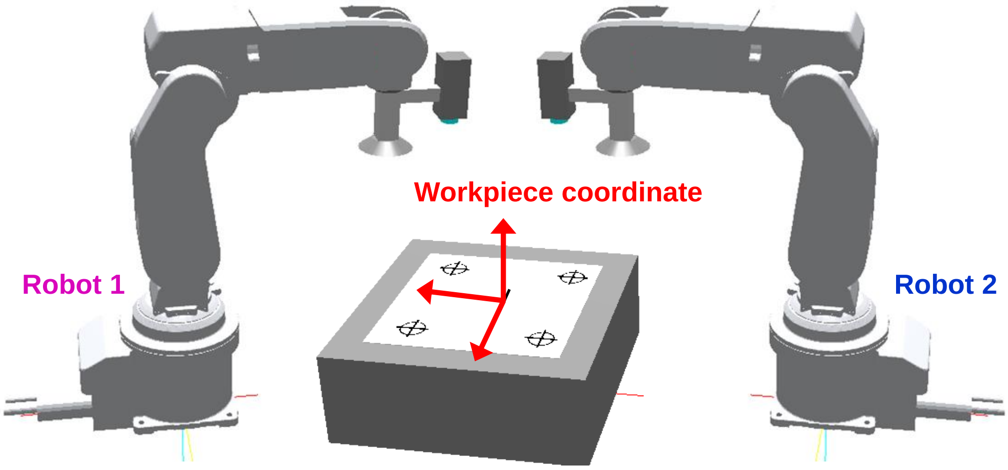


Figure 4-11 Inter-robot relational calibration

(2) Standard Specifications

Table 4-8 Standard specifications of the inter-robot relational calibration

Items	Specifications
Robot	Vertical 6-axis robot * It can’t be used with the vertical 5-axis robot, the horizontal 4-axis robot and the user robot.
Language	Only MELFA-BASIC VI
Setting method of vision sensor	Hand eye method
Output information	Position and posture of the workpiece coordinate of each robot
Remarks	

(3) Workflow

Start work

↓

1. Set the calibration sheet Refer to 4.4.(3-1)
Within the motion range of each robot, Place the calibration sheet.

↓

2. Perform work coordinate calibration（Robot1） Refer to 4.3
Perform calibration of RobotA and work coordinates.

↓

3. Perform work coordinate calibration（Robot2） Refer to 4.3
Perform calibration of RobotB and work coordinates.

↓

4. Perform inter-robot relational calibration Refer to 4.4.(3-2)
Perform inter-robot relational calibration with each work coordinate data.

↓

End work

(3-1) set the calibration sheet

Within the motion range of each robot, Place the calibration sheet.

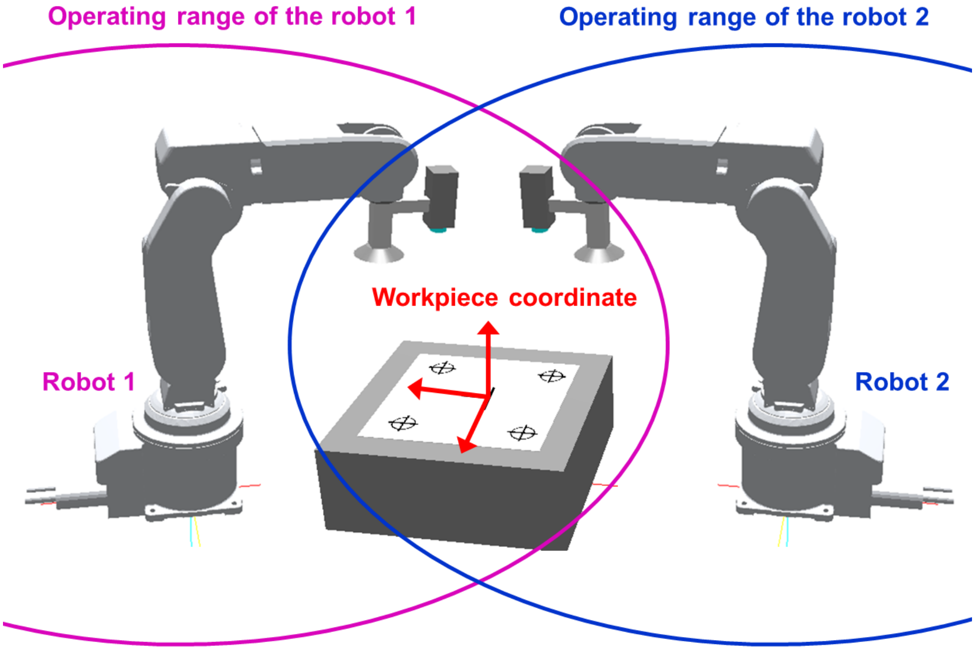


Figure 4-12 Set the calibration sheet

(3-2) Perform work coordinate calibration

Using work coordinate data obtained in the previous object, Perform work coordinate calibration.

The following describes how to set the relative calibration between robots in “Collision avoidance function" and " Cooperative operation function".

① Collision avoidance function

In order to use the collision avoidance function (collision check between robots), it is necessary to perform Inter-robot relational calibration. By using this function, the object can be simplified.

For collision avoidance function, refer to the instruction manual “Detailed explanations of functions and operations”.

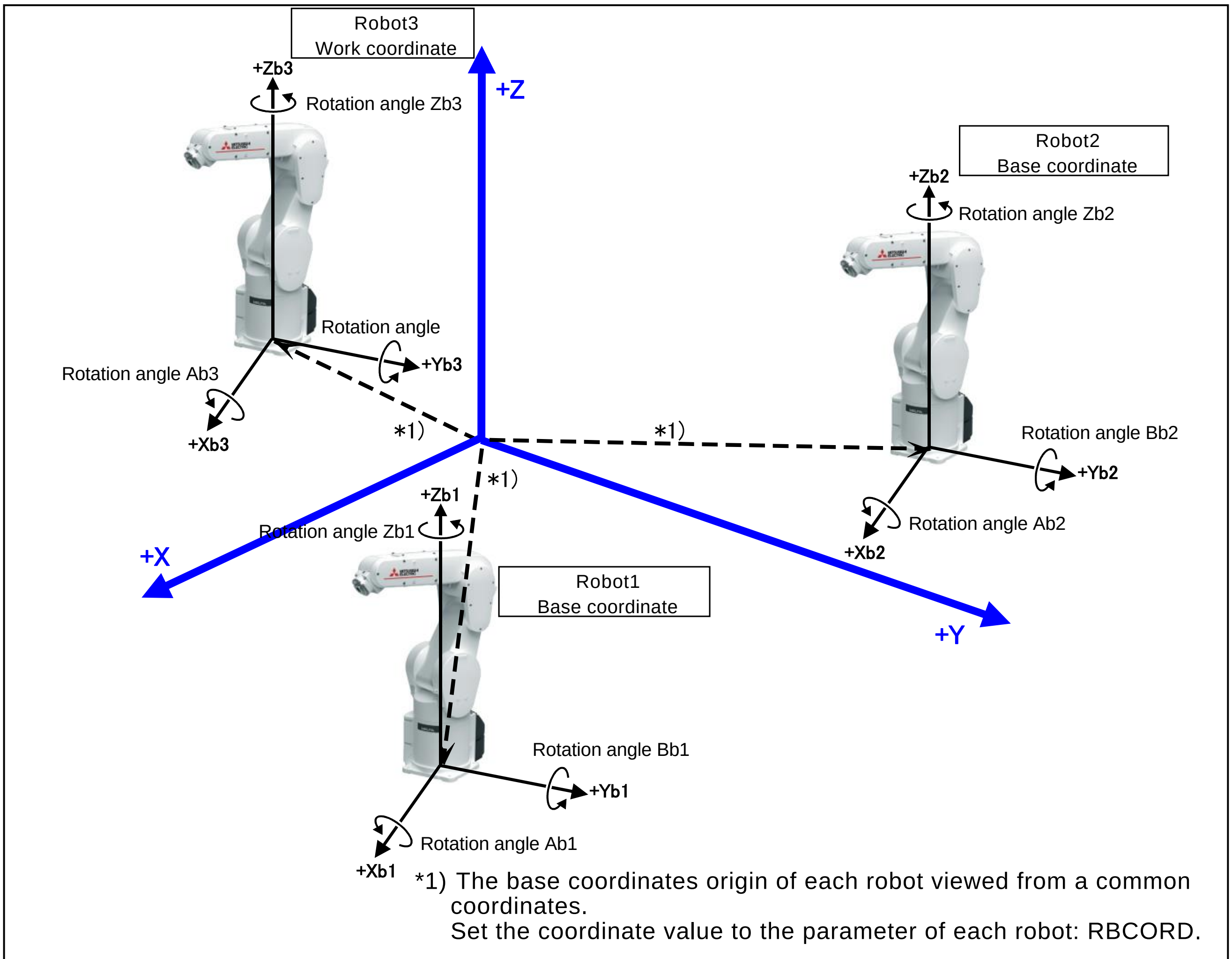


Figure 4-13 Inter-robot relational calibration

1) Setting method of parameter RBCORD

For each robot, after performing workpiece coordinate calibration, Please execute the following program.

For the variable MWrkNo, enter the registered workpiece coordinate number.

<Program>

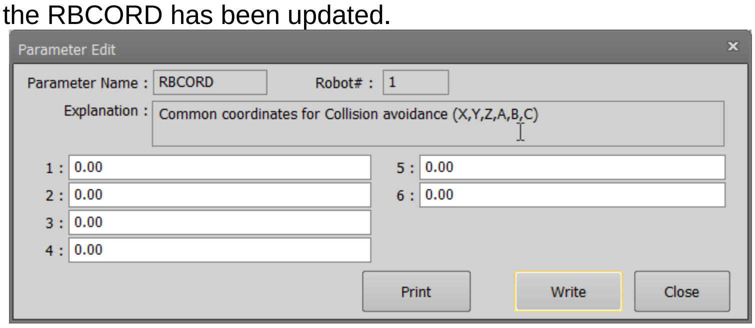
```
MWrkNo = 1                          ' Workpiece coordinate number of registration destination
PRob2Wrk = P_WkCord(MWrkNo)         'Get workpiece coordinate data
PRBCORD = Inv(PRob2Wrk)             ' From the common coordinate, the base coordinate origin of each robot
PrmWrite 1, "RBCORD", PRBCORD       'Write RBCORD
```

2) Check the parameter RBCORD

a) Select <online> - <parameter> - <parameter list> and display the parameter list.

b) Enter the "RBCORD" on the parameter list and display the parameter editing.

c) Check the RBCORD has been updated.



② Cooperative operation function

In order to use the cooperative operation functionit is necessary to perform Inter-robot relational calibration.By using this function, the object can be simplified.

For cooperative operation function, refer to the instruction manual “Detailed explanations of functions and operations”.

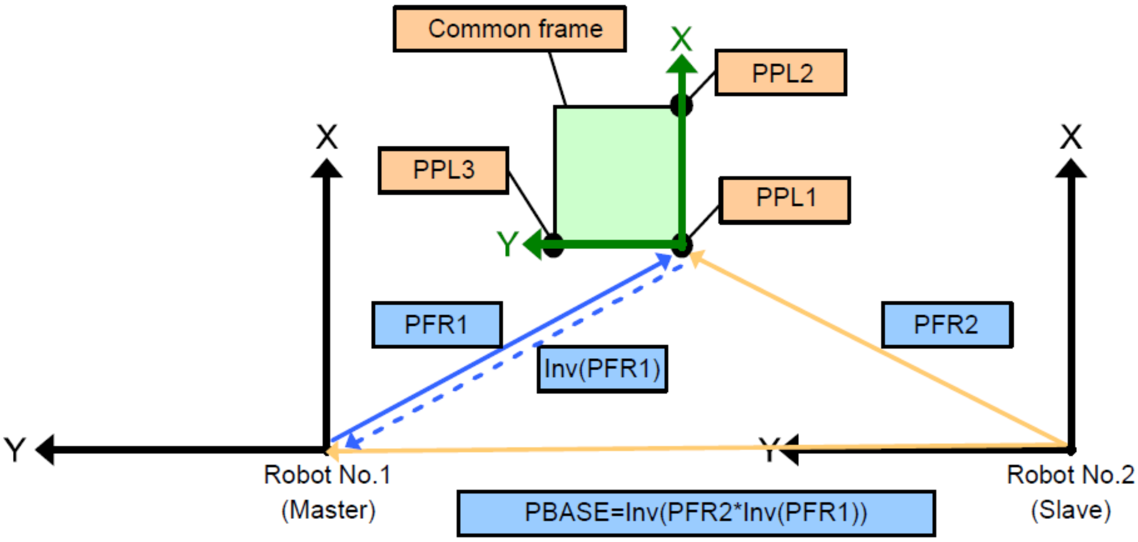


Figure 4-14 Adjust common base coordinates

1) Setting common frame coordinates of robot 1 and robot 2

The cooperative operation function defines common frame coordinates. The position variables PFR1 and PFR2 can be obtained as follows.

For each robot, execute the following program after workpiece coordinate calibration.

For the variable MWrkNo, enter the registered workpiece coordinate number.

```
<Program of robot 1>
MWrkNo = 1                      ' Workpiece coordinate number of registration destination
PRob2Wrk = P_WkCord(MWrkNo)     'Get workpiece coordinate data
PFR1 = PRob2Wrk                 'Common frame coordinate origin as viewed from the robot 1

<Program of robot 2>
MWrkNo = 1                      ' Workpiece coordinate number of registration destination
PRob2Wrk = P_WkCord(MWrkNo)     ' Get workpiece coordinate data
PFR2 = PRob2Wrk                 ' Common frame coordinate origin as viewed from the robot 2
```

2) Setting the base coordinates of robot 2（Only robot 2）

Execute the following program and set base coordinates of robot 2 to base coordinates of robot 1.

```
< Program >
'The base coordinates of the robot 2 are made common to the robot 1.
PBTMP = PFR2*Inv(PFR1)
PBASE = Inv(PBTMP)
Base PBASE  'Setting the base coordinates
```

4.5 Robot Programming Language

It describes commands, system functions, and state variables used in the automatic calibration function.

4.5.1 Language list

(1) Command list

Table 4-9 Commands used in the calibration support function

No.	Target function	Command	Explanation	Page
1	Tool length calibration	TlClrPt	Initialize calculation auxiliary points. (Set all points to zero)	4-63
2		TlSetPt	Set one calculation auxiliary point.	4-64
3		TlCal2D	Calculate the tool coordinate (XY) and write it in the parameter MEXTLn (n: 1 to 16).	4-65
4		TlCal3D	Calculate the tool coordinate (XYZ) and write it in the parameter MEXTLn (n: 1 to 16).	4-66
5	Vision calibration	VSCalClr	Clear the image coordinates of the target calibration number and the robot coordinates.	4-67
6		VSSetCP	Set corresponding points between image coordinates and robot coordinates.	4-68
7		VSRegCD	Calibration data is obtained from a plurality of corresponding points and registered in the robot controller.	4-69

(2) System function list

Table 4-10 System function used in the calibration support function

No.	Target function	System function	Explanation	Page
1	-	PVSCal	Convert image coordinates to robot coordinates using vision calibration data (parameters VSCALB 1-8).	4-70
2	Workpiece coordinate calibration	PVSDrct	Calculate the attitude of the camera from the calibration data.	4-71
3		PCalObtP	The point of interest is calculated from the two position data of the robot.	4-73

(3) State variable list

Table 4-11 State variable used in the calibration support function

No.	Target function	State variable	Explanation	Page
1	Tool length calibration	M_TlErr	Returns the estimation error of the tool coordinates of the specified tool number.	4-75
2	Vision caribration	M_VSDErr	Returns the estimation error of the specified vision calibration data.	4-76

4.5.2 Language detailed description

<How to read the described items>

[Function]	: Indicates the command word functions.
[Format]	: Indicates how to input the command word argument. The argument is shown in <>. [ ] indicates that the argument can be omitted. [] indicates that a space is required.
[Terminology]	: Indicates the meaning and range, etc. of the argument.
[Reference Program]	: Indicates a program example.
[Explanation]	: Indicates detailed functions and cautions, etc.
[The available robot type]	: Indicates the available robot type.
[Related parameter]	: Indicates the related parameter.
[Related system variables]	: Indicates the related system variables.
[Related instructions]	: Indicates the related instructions.

TlClrPt (Tool Clear Point)

[Function]

In the tool length calibration, initialize the calculation auxiliary point set to the target tool number and the tool data (parameter MEXTL <Tool number>).

[Format]

```
TlClrPt <Tool number>
```

<Tool number> Specify the tool number to be registered.
Setting range: 1 to 16

[Reference Program]

```
Dim PTL(3)
TlClrPt 1                        'Clear tool length and auxiliary point of tool number 1.
For M1=1 To 3
    TlSetPt 1, M1, PTL(M1)       'Set auxiliary point.
Next
MErr=TlCal3D 1                   'Calculate tool length (X, Y, Z) and register in tool number 1.
If MErr<>0 Then Error 9100       'If registration of the tool length fails, an error is output.
MCalErr=M_TlErr(1)               'Estimate tool length error.
If MCalErr>0.25 Then Error 9100
```

[Explanation]

(1) All set auxiliary points and tool data are deleted. Please use with care.

[Related instructions]

TlSetPt, TlCal2D, TlCal3D

[Related system variables]

M_TlErr

[Related parameter]

MEXTL1 to 16

TlSetPt (Tool Set Point)

[Function]

In the tool length calibration, set the calculation auxiliary point.
In order to calculate the tool length, at least three auxiliary points are required.

[Format]

```
TlSetPt   <Tool number>, <Auxiliary point number>, <Robot position>
```

<Tool number>	Specify the tool number to be registered. Setting range: 1 to 16
<Auxiliary point number>	Specify the auxiliary point number of tool length calibration. Setting range: 1 to 8
<Robot position>	Set the robot position (XYZ coordinate) to be registered in the auxiliary point.

[Reference Program]

```
Dim PTL(3)
TlClrPt 1                          'Clear tool length and auxiliary point of tool number 1.
For M1=1 To 3
    TlSetPt 1, M1, PTL(M1)         'Set auxiliary point.
Next
MErr=TlCal3D 1                     'Calculate tool length (X, Y, Z) and register in tool number 1.
If MErr<>0 Then Error 9100         'If registration of the tool length fails, an error is output.
MCalErr=M_TlErr(1)                 'Estimate tool length error.
If MCalErr>0.25 Then Error 9100
```

[Explanation]

(1) Use the X, Y, Z, A, B, C elements of the position type to set the auxiliary point. The position data to be set is XYZ coordinate.

(2) Before calculating the tool length, it is necessary to set three or more auxiliary points.

[Related instructions]

TlClrPt, TlCal2D, TlCal3D

[Related system variables]

M_TlErr

[Related parameter]

MEXTL1 to 16

TlCal2D (Tool Calibration 2D)

[Function]

In the tool length calibration, calculate the tool length using the registered auxiliary point (3 points or more). The calculated tool data is set to the robot controller (parameter MEXTL <tool number>), and its accuracy is estimated.

In this command, two-dimensional tool data is calculated and the set tool coordinates are (X, Y, 0, 0, 0, 0).

[Format]

```
<Result> = TlCal2D <Tool number>
```

<Tool number> Specify the tool number to be registered.
Setting range: 1 to 16

<Result> Returns the execution result.

- 0 : Success
- -1 : Failure
  Not enough auxiliary points. To calculate the tool length using at least three auxiliary points, please set at least two auxiliary points.
- -2 : Failure
  It is not possible to calculate the tool length from the set auxiliary point. Please set the auxiliary point again.
- -3 : Failure
  Estimated error of calculated tool data is 100 mm or more. Please set the auxiliary point again.

[Reference Program]

```
Dim PTL(3)
T I ClrPt 1                     'Clear tool length and auxiliary point of tool number 1.
For M1=1 To 3
    TlSetPt 1, M1, PTL(M1)      'Set auxiliary point.
Next
MErr=TlCal2D 1                  'Calculate tool length (X, Y, 0) and register in tool number 1.
If MErr<>0 Then Error 9100      'If registration of the tool length fails, an error is output.
MCalErr=M_TlErr(1)              'Estimate tool length error.
If MCalErr>0.25 Then Error 9100
```

[Explanation]

(1) In order to use TlCal2D, it is necessary to set at least three auxiliary points to be used for tool length calibration beforehand. (Use the TlSetPt command to set the auxiliary point.)

(2) If auxiliary point setting is registered incorrectly, tool length may not be calculated.

[Related instructions]

TlClrPt, TlSetPt

[Related system variables]

M_TlErr

[Related parameter]

MEXTL1 to 16

TlCal3D (Tool Calibration 3D)

[Function]

In the tool length calibration, calculate the tool length using the registered auxiliary point (3 points or more). The calculated tool data is set to the robot controller (parameter MEXTL <tool number>), and its accuracy is estimated.

In this command, three-dimensional tool data is calculated and the set tool coordinates are (X, Y, Z, 0, 0, 0).

[Format]

```
<Result> = TlCal3D <Tool number>
```

<Tool number> Specify the tool number to be registered.
Setting range: 1 to 16

<Result> Returns the execution result.
- 0 : Success
- -1 : Failure
  Not enough auxiliary points. To calculate the tool length using at least three auxiliary points, please set at least two auxiliary points.
- -2 : Failure
  It is not possible to calculate the tool length from the set auxiliary point. Please set the auxiliary point again.
- -3 : Failure
  Estimated error of calculated tool data is 100 mm or more. Please set the auxiliary point again.

[Reference Program]

```
Dim PTL(3)
T l ClrPt 1                   'Clear tool length and auxiliary point of tool number 1.
For M1=1 To 3
    TlSetPt 1, M1, PTL(M1)    'Set auxiliary point.
Next
MErr=TlCal3D 1                'Calculate tool length (X, Y, Z) and register in tool number 1.
If MErr<>0 Then Error 9100    'If registration of the tool length fails, an error is output.
MCalErr=M_TlErr(1)            'Estimate tool length error.
If MCalErr>0.25 Then Error 9100
```

[Explanation]

(1) In order to use TlCal3D, it is necessary to set at least three auxiliary points to be used for tool length calibration beforehand. (Use the TlSetPt command to set the auxiliary point.)

(2) If auxiliary point setting is registered incorrectly, tool length may not be calculated.

[Related instructions]

TlClrPt, TlSetPt

[Related system variables]

M_TlErr

[Related parameter]

MEXTL1 to 16

VSCalClr (Vision Calibration Clear)

[Function]

In the vision calibration, erase all corresponding points and calibration data set for the target calibration number.

[Format]

```
VSCalClr <Calibration number>
```

< Calibration number > Specify the target calibration number.
Setting range: 1 to 8

[Reference Program]

```
Dim P(4)
CPRG$="PatternMatching.job"      'Set vision job name.
VSCalClr 1                        'Clear corresponding point.
For M1=1 To 4
  Mov P(M1)                       'Move to capturing position.
  Dly 0.5
  NVRun #1, CPRG$                 'Run vision.
  EBRead #1, , MNUM, PV           'Capture recognition result.
  PCam=PV                         'Assign image coordinates.
  PRob=P_Fbc                      'Assign robot coordinates.
  VSSetCP 1, M1, PCam, PRob       'Set corresponding points to be used for calibration.
Next
MErr=VSRegCD 1                    'Registering calibration data.
If MErr<>0 Then Error 9100        'Failed to register the calibration data.
MCalErr=M_VSCErr(1)               'Estimate error of calibration data.
If MCalErr>0.25 Then Error 9100
```

[Explanation]

(1) All correspondence points and calibration data set by the VSSetCP instruction are deleted. Please use with care.

[Related instructions]

VSSetCP, VSRegCD

[Related state variables]

M_VSCErr

[Related parameter]

VSCALB1 to 8

VSSetCP (Vision Set Calibration Point)

[Function]

In vision calibration, specify the combination (corresponding point) of the three-dimensional position (robot coordinates) of the calibration mark and the mark position (image coordinates) of the image.
In order to perform the calibration, it is necessary to specify at least four corresponding points.

[Format]

```
VSSetCP <Calibration number>, <Setting number of corresponding point>,
        <Position of image coordinates>, <Position of robot coordinates>
```

<Calibration number>	Specify the calibration number to be registered. Setting range: 1 to 8
<Setting number of corresponding point>	Specify the correspondence point number of the calibration. Setting range: 1 to 20
<Position of image coordinates>	Specify the position of image coordinates.
<Position of robot coordinates>	Specify the position of robot coordinates.

[Reference Program]

```
Dim P(4)
CPRG$="PatternMatching.job"      'Set vision job name.
VSCalClr 1                       'Clear corresponding point.
For M1=1 To 4
  Mov P(M1)                      'Move to capturing position.
  Dly 0.5
  NVRun #1, CPRG$                'Run vision.
  EBRead #1, , MNUM, PV          'Capture recognition result.
  PCam=PV                        'Assign image coordinates.
  PRob=P_Fbc                     'Assign robot coordinates.
  VSSetCP 1, M1, PCam, PRob      'Set corresponding points to be used for calibration.
Next
MErr=VSRegCD 1                   'Registering calibration data.
If MErr<>0 Then Error 9100       'Failed to register the calibration data.
MCalErr=M_VSCErr(1)              'Estimate error of calibration data.
If MCalErr>0.25 Then Error 9100
```

[Explanation]

(1) Both image coordinates and robot coordinates are registered using X, Y elements of position type data.
(2) Before registering the calibration data, it is necessary to set four or more corresponding points to be used for the calibration with this command.

[Related instructions]

VSCalClr, VSRegCD

[Related state variables]

M_VSCErr

[Related parameter]

VSCALB1 to 8

VSRegCD (Vision Register Calibration Data)

[Function]

In the vision calibration, calibration data is calculated using the set corresponding point (4 points or more), the data is set to the robot controller (parameter VSCALB <calibration number>), and the accuracy is estimated.

[Format]

```
<Result> = VSRegCD <Calibration number>
```

<Calibration number> Specify the calibration number to be registered.
Setting range: 1 to 8

<Result> Returns the execution result.
- 0 : Success
- -1 : Failure
  Not enough auxiliary points. To calculate the tool length using at least four auxiliary points, please set at least four auxiliary points.
- -2 : Failure
  Calibration data can not be calculated from the set corresponding point. There is a possibility that the corresponding point being set exists on the same straight line. Please set corresponding points that do not exist on the same straight line.

[Reference Program]

```
Dim P(4)
CPRG$="PatternMatching.job"     'Set vision job name.
VSCalClr 1                      'Clear corresponding point.
For M1=1 To 4
  Mov P(M1)                     'Move to capturing position.
  Dly 0.5
  NVRun #1, CPRG$               'Run vision.
  EBRead #1, , MNUM, PV         'Capture recognition result.
  PCam=PV                       'Assign image coordinates.
  PRob=P_Fbc                    'Assign robot coordinates.
  VSSetCP 1, M1, PCam, PRob     'Set corresponding points to be used for calibration.
Next
MErr=VSRegCD 1                  'Registering calibration data.
If MErr<>0 Then Error 9100      'Failed to register the calibration data.
MCalErr=M_VSCErr(1)             'Estimate error of calibration data.
If MCalErr>0.25 Then Error 9100
```

[Explanation]

(1) To use this command, it is necessary to set four or more corresponding points to be used for calibration beforehand. (Use VSSetCP to set corresponding points)

(2) Calibration data may not be registered if the corresponding point setting is registered incorrectly. If the calibration data can not be registered (when the accuracy of the calibration data is low), check the **Table 4-5** and re-execute the calibration.

[Related instructions]

VSCalClr, VSSetCP

[Related state variables]

M_VSCErr

[Related parameter]

VSCALB1 to 8

PVSCal (PVS calibration)

[Function]

Using the calibration data (parameters VSCALB 1 to 8) set by the vision calibration function, convert the image coordinates of the vision sensor to the robot world coordinates.

[Format]

<Position variables> = PVSCal (<Calibration number>,<Vision X>,
<Vision Y>,<Vision θ> [,<Reference Position Variables>])

<Position variables>	Specifies the position variable to assign. Returns the robot world coordinate for the calculation result of the coordinate conversions.
<Calibration number>	Specify the target calibration number. Setting range: 1 to 8
<Vision X>	X pixel coordinate of the vision sensor. [pixel]
<Vision Y>	Y pixel coordinate of the vision sensor. [pixel]
< Vision θ>	θ pixel coordinate of the vision sensor. [deg.]
<Reference Position Variables>	Specify the reference position as s position constant or position variable. When attaching a camera to the hand, specify the robot position (at the time of the image recognition) where the image is recognized with the vision sensor as a reference position. The relative calculation is performed as follows. <Robot position at the time of the image recognition>*<Calculation result of the coordinate conversion> When omitted, the absolute coordinate is set.

[Reference Program]

```
'Start the target vision sensor with Open/Print/Input command to substitute the acquired image
coordinate [pixel] for numeric variable.
'MX=X [pixcel] of the vision sensor
'MY=Y [pixcel] of the vision sensor
'MT=θ [deg.] of the vision sensor
PVS=PVSCal(1,MX,MY,MT)        'Changes the image coordinate to the robot (world)
coordinate with Calibration 1.
PVS.Z=PDST.Z                  'Specifies Z height.
Mov PVS, -50                  'Moves to a position 50 mm above the calculated position.
Mvs PVS                       'Moves to the calculated position.
```

[Explanation]

(1) Using the vision calibration data (parameters VSCALB 1 to 8), convert the image coordinates of the vision sensor to the world coordinates of the robot.
Please use the automatic calibration function or the 2D vision calibration function of RT Toolbox 3 beforehand to set the parameters VSCALB 1 ~ 8 used for coordinate conversion.

(2) If the calibration numbers other than 1 to 8 are set, error L3110 (Arg. value range over) occurs.

(3) If the numbers of argument are not either four or five, error L3120 (No. of arg. is over) occurs.

(4) If the type of arguments is different, error L3810 (Different argument type) occurs.

(5) If calibration data is calculated with VSRegCD, coordinate conversion taking into consideration the relationship between the robot coordinates and the hand system of the image coordinates is automatically carried out.

[Related instructions]

VSRegCD

[Related parameter]

VSCALB1 to 8

PVSDrct (PVS Direction)

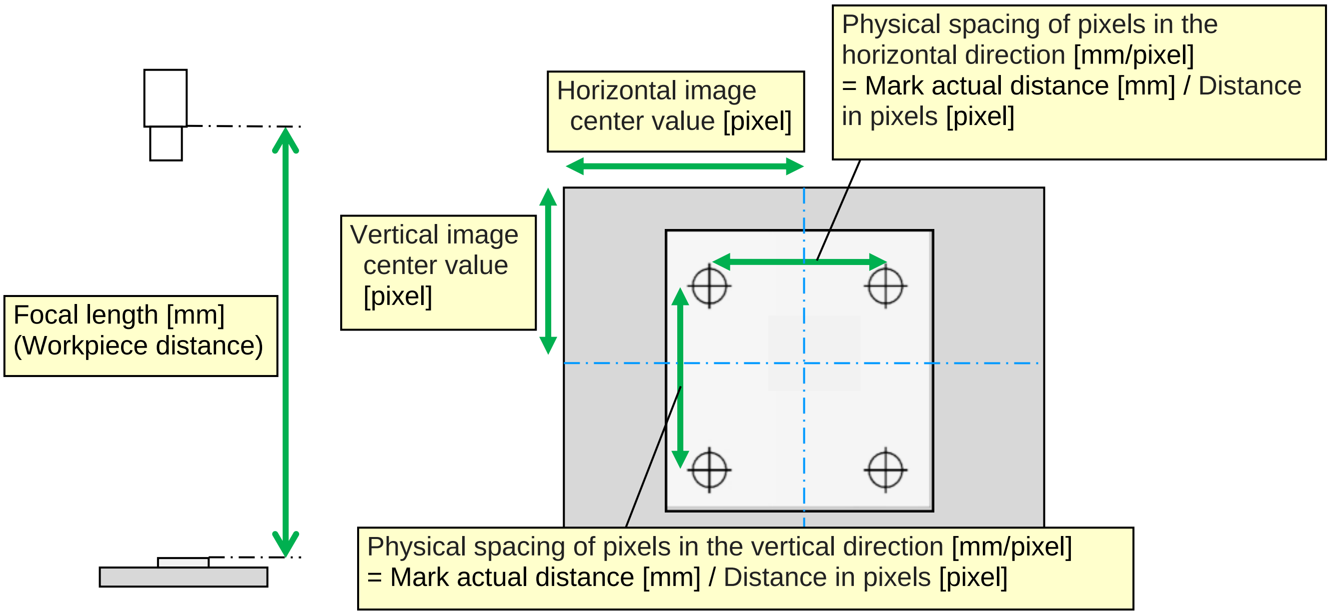
[Function]

Estimate the camera posture from the vision calibration data.
However, it is necessary to give the internal parameters of the vision sensor as an argument.

[Format]

```
<Position variables> = PVSDrct(<Calibration number>, <Internal parameter A>,
                        <Internal parameter B>, <Internal parameter C>,<Internal parameter D>)
```

<Position variables>	Specifies the position variable to assign.
<Calibration number>	Specify the target calibration number. Setting range: 1 to 8
<Internal parameter A>	Focal length [mm] / Physical spacing of pixels in the horizontal direction [mm/pixel]
<Internal parameter B>	Focal length [mm] / Physical spacing of pixels in the vertical direction [mm/pixel]
<Internal parameter C>	Horizontal image center value [pixel]
<Internal parameter D>	Vertical image center value [pixel]



[Reference Program]

```
Dim PW(4), PV(4)
VSCalClr 1     ' Clear corresponding point.
For M1 = 1 To 4
  VSSetCP 1, M1, PV(M1), PW(M1)       'Set corresponding points between vision
coordinates and work coordinates
Next M1
MRes = VSRegCD 1                       'Register calibration data
If MRes <> 0 Then
  Error 9100
EndIf
'MPRMA : Internal parameter A (Focal length [mm] / Physical spacing of pixels in the horizontal
direction [mm/pixel])
'MPRMB : Internal parameter B (Focal length [mm] / Physical spacing of pixels in the vertical
direction [mm/pixel])
'MPRMC : Internal parameter C (Horizontal image center value [pixel])
'MPRMD : Internal parameter D (Vertical image center value [pixel])
PVSPose = PVSDrct( 1, MPRMA, MPRMB, MPRMC, MPRMD ) 'Posture of vision sensor in
work coordinates
```

[Explanation]

(1) Estimate the camera posture (relative to the plane of vision calibration) from the vision calibration data and

the internal parameters of the vision sensor.

(2) For the <positional variable of the assignment destination>, only the value of the ABC component is substituted. 0 is assigned to the XYZ component.

[Related instructions]
VSRegCD

[Related parameter]
VSCALB1 to 8

PCalObtP (P Calcurate Object Position)

[Function]

Calculate the position data of the point of interest (intersection) of the two position data.

[Format]

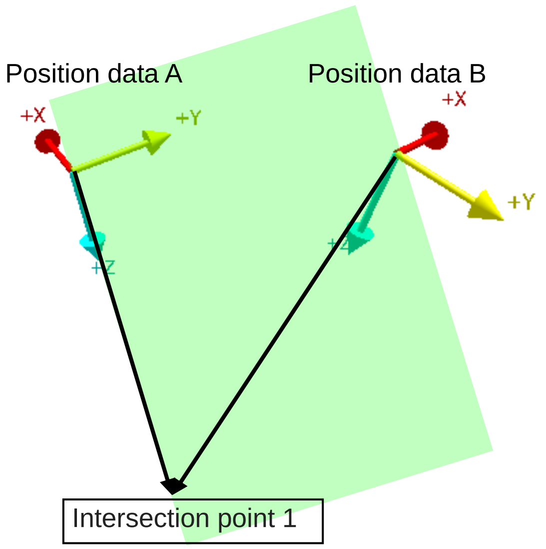
<Numeric Variable> = PCalObtP(<Position data A>,< Position data B>,<Calculated position>)

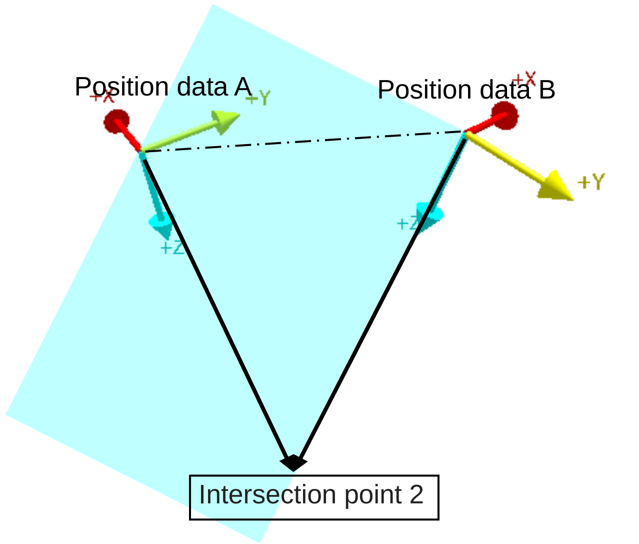
<Position data A> Specify the first position data.
<Position data B> Specify the second position data.
<Calculated position> Specify the storage location variable of the calculation result.
<Numeric variable> Returns the execution result.
0 : Success
-1 : Failure
It is not possible to calculate the point of interest.

The point of interest (intersection) is calculated as follows.

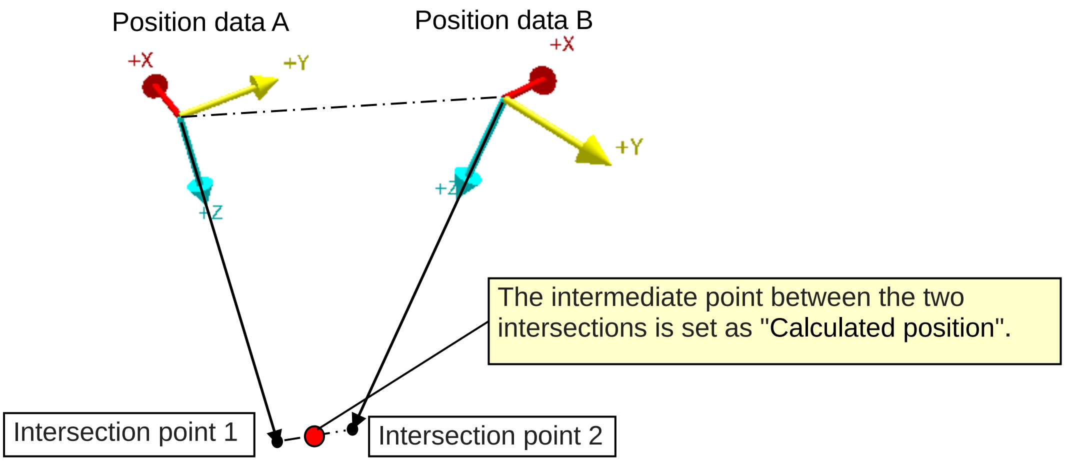
(1) Calculate the intersection point 1 of the two position data and a straight line projected position data B and extended in the + Z direction on a plane defined from an arbitrary point on the Z coordinate of position data A.

(2) Calculate the intersection 2 of the straight line projected position data A and extended in the + Z direction on the two position data and the plane defined from the arbitrary point on the Z coordinate of position data B.





(3) Returns the midpoint of the two intersection points as the calculated position.



[Reference Program]

```
'P1 : Position data A,   P2 : Position data B
MErr = PCalObtP(P1,P2,Pres)        'Calculate the intersection point of P1 and P2
If MErr <> 0 Then                  'When the intersection point does not exist
  Error 9100
EndIf
```

[Explanation]

(1) Calculate the intersection point of the straight line extended in + Z direction of the two position data and output it.

(2) Only the XYZ component is assigned to the position variable specified in <Calculated position>. 0 is assigned to the ABC component.
If two position and orientation data are the same, or if no intersection is obtained, all components 0 will be substituted.

M_TlErr

[Function]

Returns error of tool data calculated by tool length calibration (execution of TlCal2D / TlCal3D command).

[Format]

<Numeric variable>=M_TlErr（<Tool number>）

<Numeric variable>	Returns the execution result.
<Tool number>	Specify the target tool number. Setting range: 1 to 16

[Reference Program]

```
Dim PTL(3)
TlClrPt 1                      'Clear tool length and auxiliary point of tool number 1
For M1=1 To 3
    TlSetPt 1, M1, PTL(M1)     'Set auxiliary point
Next
MErr=TlCal3D 1                 'Calculate tool length (X, Y, Z) and register in tool number 1
If MErr<>0 Then Error 9100     'Failed to register the calibration data.
MCalErr=M_TlErr(1)             'Estimate error of calibration data.
If MCalErr>0.25 Then Error 9100
```

[Explanation]

(1) This evaluation value means the accuracy of the tool data. It does not mean an error when the robot operates.
(2) It turns to 0 when the power of the robot controller is turned off.
(3) Returns the error of tool data last calculated by tool length calibration (execution of TlCal2D / TlCal3D command).

[Related instructions]

TlCal2D, TlCal3D

[Related parameter]

MEXTL1 to 16

M_VSCErr

[Function]

In the vision calibration function, it returns the estimation error [mm] of the calibration data estimated by the VSRegCD command.

[Format]

```
<Numeric variable> = M_VSCErr(<Calibration number>)
```

<Numeric variable>	Specify numerical variable to assign. When estimation error can not be calculated, it becomes "-1".
<Calibration number>	Specify the target calibration number. Setting range: 1 to 8

[Reference Program]

```
Dim P(4)
CPRG$="PatternMatching.job"    'Set vision job name.
VSCalClr 1                     'Clear corresponding point.
For M1=1 To 4
  Mov P(M1)                    'Move to capturing position.
  Dly 0.5
  NVRun #1, CPRG$              'Run vision.
  EBRead #1, , MNUM, PV        'Capture recognition result.
  PCam=PV                      'Assign camera coordinates.
  PRob=P_Fbc                   'Assign robot coordinates.
  VSSetCP 1, M1, PCam, PRob    'Set corresponding points to be used for calibration.
Next
MErr=VSRegCD 1                 'Registering calibration data.
If MErr<>0 Then Error 9100     'Failed to register the calibration data.
MCalErr=M_VSCErr(1)            'Estimate error of calibration data.
If MCalErr>0.25 Then Error 9100
```

[Explanation]

(1) The reliability (accuracy) of the registered calibration data can be calculated by using the corresponding point group when calculating the calibration data and also setting the error [mm] between the robot coordinate value and the robot coordinate value calculated from the image coordinate value and it will be evaluated.

(2) This evaluation value means the accuracy of the calibration data and does not mean error when the robot operates.

(3) If the calibration data is not registered, -1 is returned.

(4) If the accuracy of the calibration data is low, check the **Table 4-5** and re-execute the calibration.

[Related instructions]

VSRegCD

[Related parameter]

VSCALB1 to 8

5. Robot mechanism temperature compensation function

This chapter explains the robot mechanism temperature compensation function.

5.1 Specification

Robot mechanism temperature compensation function is a function to measure the temperature of the robot arm and automatically correct errors due to thermal expansion of the arm.
With this function, even if the temperature changes depending on the season and time zone, positional shift due to thermal expansion of the robot arm can be suppressed.

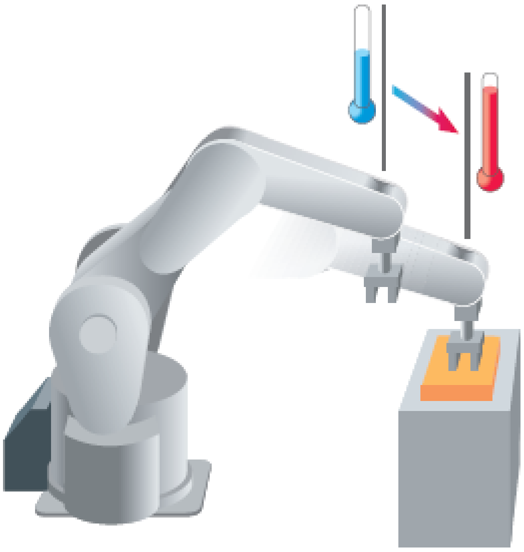


Table 5-1 Basic specification of robot mechanism temperature compensation function

Contents	Specifications
Compatible robot	Vertical articulated (6-axis) robot Horizontal articulated (4-axis) robot *Not supported with vertical articulated (5-axis) robot or user mechanism. *Additional axis is not subject to temperature correction.
Remarks	Temperature correction is always enabled. However, when specifying the position command by pulse coordinates like the Mxt command (real-time external control command) specified with motor pulse coordinate data, temperature correction is not performed.

This function is effective only by installing the MELFA Smart Plus card pack.
*When using the MELFA Smart Plus card, "2" must be set in the parameter "SMART+1".
（When the origin setting is not complete, this function is invalid.）

If you wish to disable this function with the MELFA Smart Plus card/card pack installed, set the parameter "MTCENA" to "0" and turn the controller power ON again.

Table 5-2. State of Robot mechanism temperature compensation function

Item	parameter “SMART+1”	parameter “MTCENA”	Function state
MELFA Smart Plus card	2	0	Invalid
		1	Enable
	Others	0	Invalid
		1	Invalid
MELFA Smart Plus card pack	-	0	Invalid
		1	Enable

5.2 Precautions

Describe points to be aware of when using robot mechanism temperature compensation function.

5.2.1 Please enable this function from the beginning

During automatic operation by the robot program, compensation is performed with the position data registered in the program as the reference position.

When using this function, it is necessary to register the position data that considered temperature compensation, so please enable this function before teaching position.

⚠Caution

When this function is enabled after position teaching is performed with this function invalid state, or when this function is invalidated after position teaching is performed with this function enable state, there is a possibility that the position will be shifted.
It is possible to confirm the state of this function by a variable M_SmartPlus(2).
(For details on variable, refer to "3.3.Robot language specification".)

5.2.2 Accuracy is not obtained near the singular point and the vicinity of the motion range

In the vicinity of the singular point, since the robot may operate greatly with slight position correction, the robot mechanism temperature compensation function is automatically invalidated. (For details on singularities, refer to "separate volume: instruction manual / detailed explanation of functions and operations".)

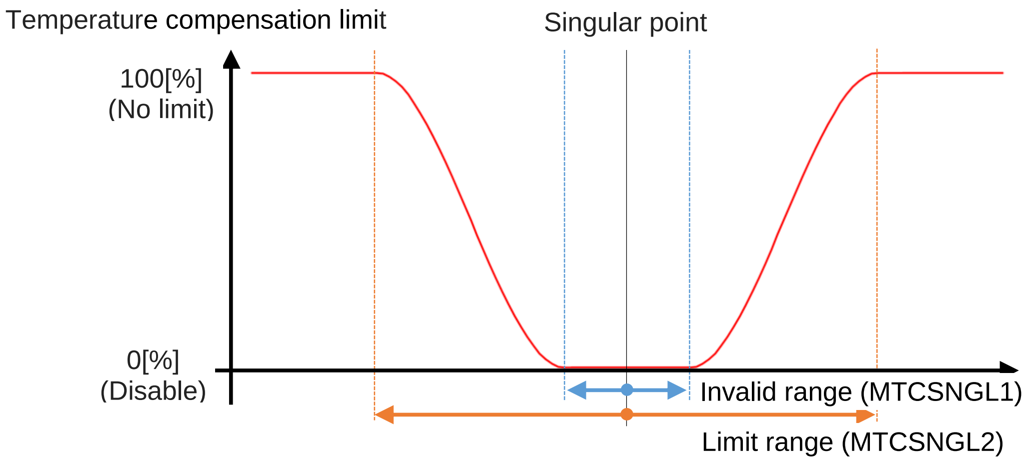
Also, even when the position after temperature compensation is out of the range of motion of the robot, the robot mechanism temperature compensation function is automatically invalidated.

In work requiring precision, we recommend avoiding singular points and the vicinity of the operating range.

In order to suppress the sudden change of the command position due to automatically invalidation of the robot mechanism temperature compensation function, the range where the temperature compensation is restricted is set near the singular point and the outside of the operation range.

This limit range can be adjusted with the parameter "MTCSNGL1" (invalid range), "MTCSNGL2" (limit range).

If the robot vibrates due to a sudden change in the command position when passing through the singular point, it can be improved by widening the limit range.



⚠Caution

When the limit range is narrow or the range between the invalid range and the restricted range is narrow, the command position abruptly changes near the singular point or the operation range, an error H115n (servo amplifier communication data command value error) or an error H091n (servo amplifier overspeed) occurs, and an error stop may occur.
(n is the axis Number)

5.3 Parameter setting

Parameters used in robot mechanism temperature compensation function are explained.
(For the setting method of parameters, refer to "separate volume: instruction manual / detailed explanation of functions and operations".)

Parameter	Parameter name	No. of arrays	etails explanation	Factory setting
Robot mechanism temperature compensation Function selection	MTCENA	Integer 1	Specify enable / disable of robot mechanism temperature compensation function.	1 (Enable) Applicable robot only 0 (invalid) for other robot

Parameter	Parameter name	No. of arrays	etails explanation	Factory setting
Robot mechanism temperature compensation Invalid range	MTCSNGL1	Real value 4	In the vicinity of the singular point and in the vicinity of the movement range, specify the range in which the robot mechanism temperature compensation function is to be invalidated. <Vertical articulated (6-axis) robot> 1st element : Near the structure flag (RIGHT / LEFT) It is specified by the distance on the XY plane from the singular point. Unit : [mm] * 0 is on a singular point 2nd element : Near the structure flag (ABOVE / BELOW) It is specified by the J3 axis angle from the singular point. Unit : [deg] * 0 is on a singular point 3rd element : Near the structure flag (NONFLIP/FLIP) It is specified by the J5 axis angle from the singular point. Unit : [deg] * 0 is on a singular point 4th element : Near the operating range It is specified by the distance from the movement range boundary when the position farthest from the movement range is 0 [%] and the movement range boundary is 100 [%]. Unit : [%] * 0 is the operating range boundary <Horizontal articulated (4-axis) robot> 1st element : Near the structure flag (RIGHT / LEFT) It is specified by the J2 axis angle from the singular point. Unit : [mm] * 0 is on a singular point 2nd element : Insignificant 3rd element : Insignificant 4th element : Near the movement range It is specified by the distance from the movement range boundary when the position farthest from the movement range is 0 [%] and the movement range boundary is 100 [%]. Unit : [%] * 0 is the operating range boundary	Vertical articulated (6-axis) robot = (0, 5, 5, 0) Horizontal articulated (4-axis) robot = (5, 0, 0, 0)
Robot mechanism temperature compensation Limit range	MTCSNGL2	Real value 4	In the vicinity of the singular point and the vicinity of the motion range, specify the range to limit the compensation amount of the robot mechanism temperature compensation function. The position of each element is the same as the parameter "MTCSNGL1".	Vertical articulated (6-axis) robot = (100, 15, 15, 1) Horizontal articulated (4-axis) robot = (15, 0, 0, 1)

6. Coordinated control for additional axes

In this chapter, we explain coordinated control for additional axes.

Table 6-1 Function list of coordinated control for additional axes

Functions	Contents
Base coordinate cooperative control (Refer to 6.2)	Allows synchronized operation where a robot is installed on an additional axis (linear drive axis) and its speed relative to the workpiece is specified. Supports machining of large workpieces using linear, circular or spline interpolation that exceeds the robot's range of movement.
Additional axis tracking (Refer to 6.3)	Allows synchronized operation where tracking of the robot and workpieces on an additional axis (linear axis) is specified. Linear or circular interpolation while the workpiece is being transported allows operations such as precision sealing work and surface inspections.

6.1 Calibration of base coordinates

6.1.1 Overview

In this chapter, we explain how to match the base coordinates of robot and additional axis / user mechanisms. This setting is necessary when you use base coordinate cooperative control or additional axis tracking.

6.1.2 Specification

You can match the base coordinates of robot and additional axis / user mechanisms by setting the offset from the robot's world coordinate system origin to the base coordinate origin of the additional axis / user mechanisms.

- The base coordinate system of the additional axis / user mechanisms is the origin where the position data is 0mm, and the +X axisi is the movement direction.
- Set it individually for each joint axis.

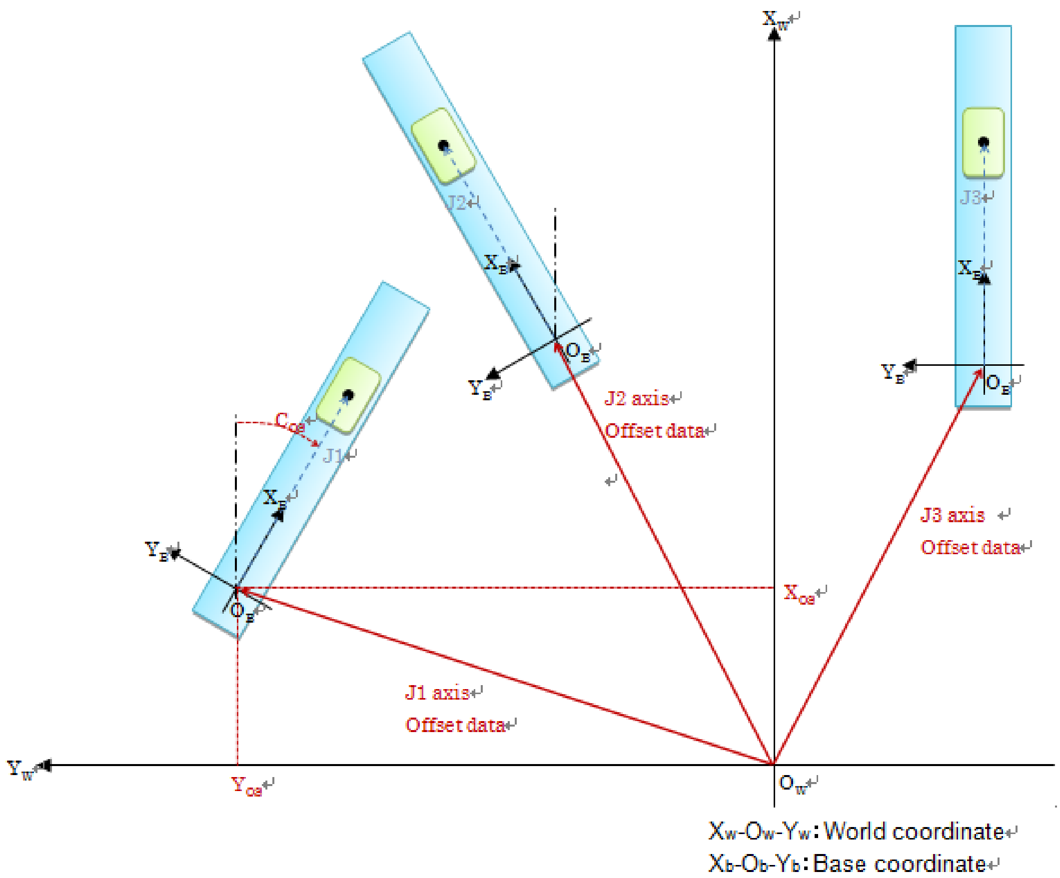


Figure 6-1 Offset data of additional axis/user mechanisms base coordinate（Example of user mechanisms）

6.1.3 Operation procedure

Operation start

↓

1. Setting of additional axis / user mechanisms（Advance preparation）
Set additional axis / user mechanisms that you use.

↓

2. Setting of base coordinate offset of additional axis / user mechanisms Refer to 6.1.4
Set the parameter of base coordinate offset of additional axis/user mechanisms.

↓

Operation end

6.1.4 Parameter setting

The parameter list which is used is below.

Table 6-2 The parameter list of base coordinate calibration.

Parameter	Parameter name	Element number	Explain of contents		Factory setting
Base coordinate offset data of additional axis / user mechanisms.	BSWOFST1	Real number 6	L1(J7)axis/ J1 single axis	Set the offset data from the robot's world coordinate system origin to the base coordinate origin of the additional axis / user mechanisms. (X, Y, Z, A, B, C). (In case of robot (mechanism 1), it is equal to additional axis.) Element 1： Translation amount of X axis direction[mm] Element 2： Translation amount of Y axis direction[mm] Element 3： Translation amount of Z axis direction[mm] Element 4： Rotation amount about X axis [deg] Element 5： Rotation amount about Y axis [deg] Element 6： Rotation amount about Z axis [deg]	0.00, 0.00, 0.00, 0.00, 0.00, 0.00
	BSWOFST2		L2(J8)axis/ J2 single axis		
	BSWOFST3		J3 single axis		

6.2 Base coordinate cooperative control

6.2.1 Overview

You can move robot base coordinate by using additional axis(traveling axis).

If you move traveling axis on which robot rides, robot moves. In that case, if the origin of the base coordinate system of robot is on the traveling axis, the base coordinates will also draw a similar trajectry with the movement of the traveling axis. In this way, it becomes possible to move the base coordinates of robot attached to the additional axis.
The example of moving robot base coordinate is shown by Figure 6-2.

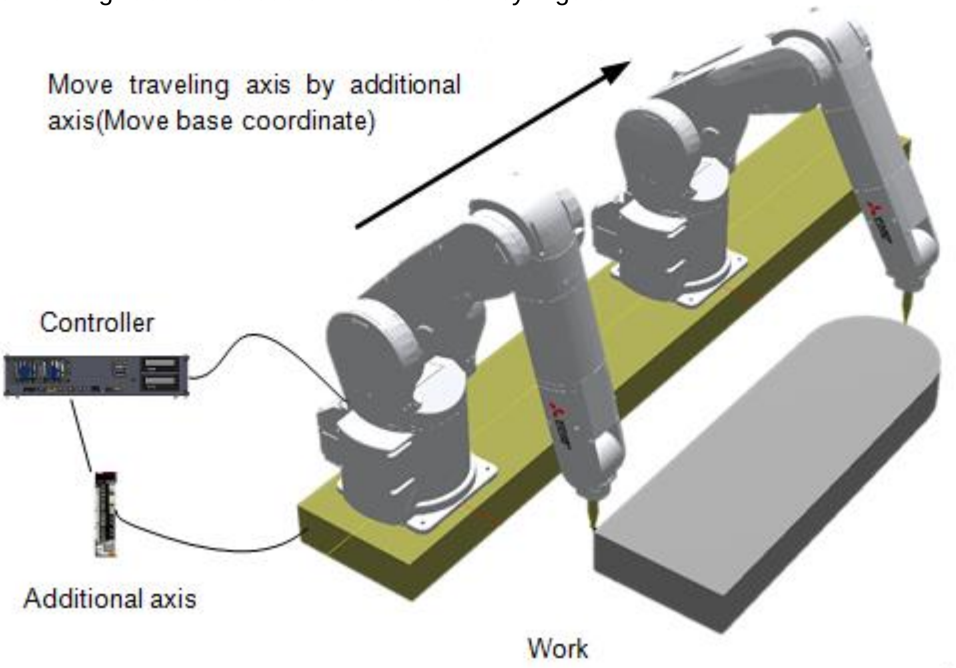


Figure 6-2 Move image of base coordinate

Next, consider the case that the target position is outside the movement range for the robot alone as shown in Figure 6-3.

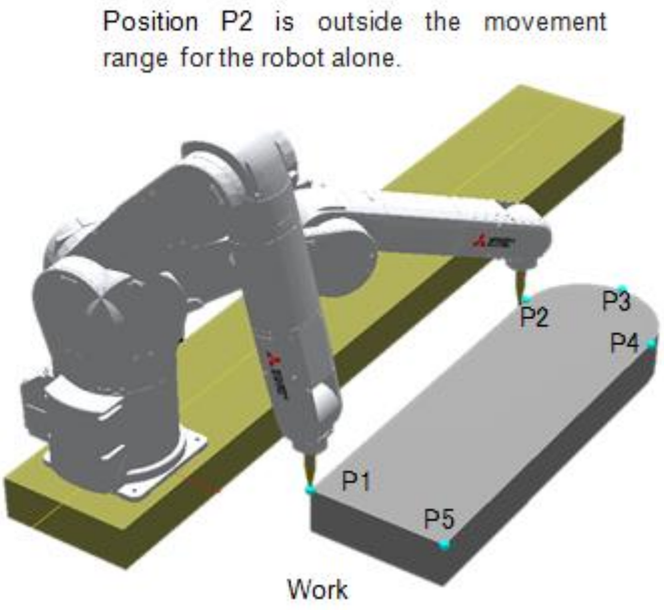


Figure 6-3 Move image of robot alone

When the target position is set outside the movement range of the robot, robot can move to a position where the robot alone can not move by enlarging movement range to move the robot and the additional axis simultaneously.
Coordination move with additional axis is shown by Figure 6-4.

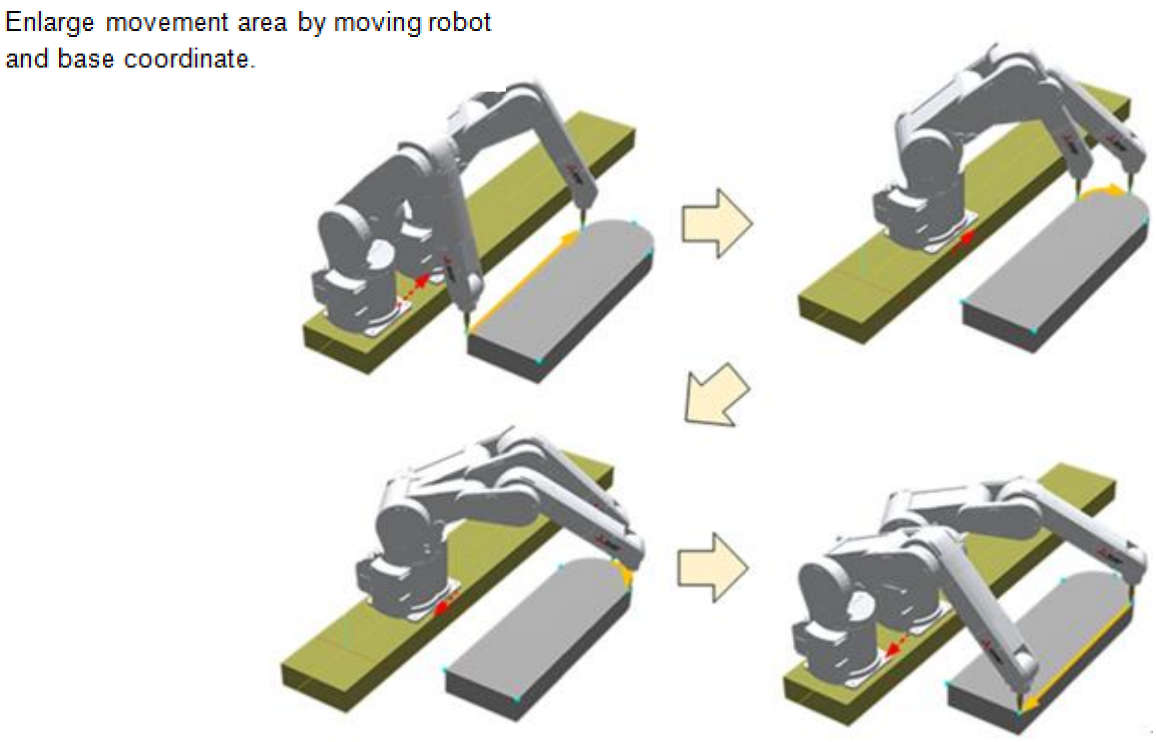


Figure 6-4 Move image of coordination with additional axis

In additon, for a work of complicated shape with large size which was difficult with conventional interpolation operation, it is possible to operate by executing a spline interpolation command in coordination control of base coordinate.

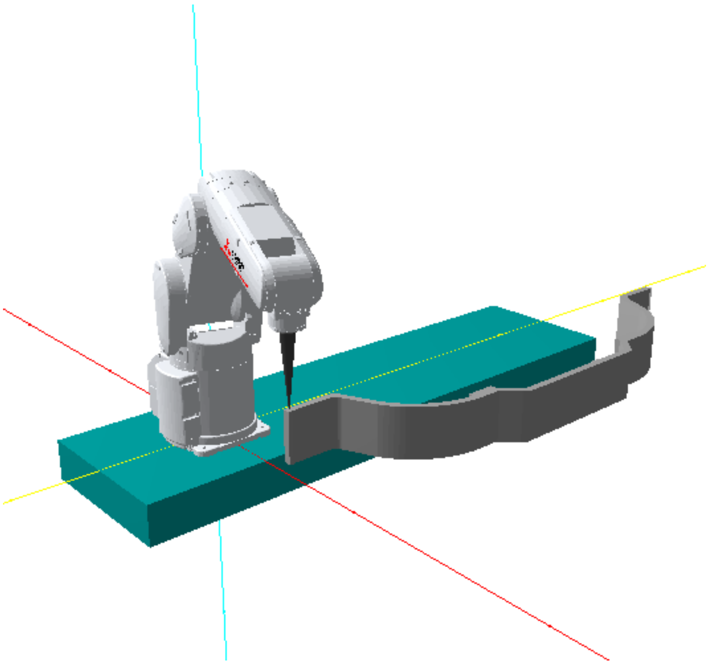


Figure 6-5 Motion image of spline interpolation by cooperation with additional axis

6.2.2 System configration

6.2.2.1. Customer preparation equipment

The following table shows the equipment that customers need to prepare, which is necessary for the base coordinate coordinated control system.

Table 6-3 The list of customer preparation equipment.

Equipment name	Format	Quantity	Remarks
Servo amplifier, servomotor, options, peripheral device	Refer to remarks	-	Refer to "Instruction Manual for Servo Amplifier and Servomotor".
Battery (For absolute position detection system)	MR-BAT6V1SET	Amplifier number	The battery case (MR-BT6VCASE) and the battery (MR-BAT6V1) are necessary when using MR-J4W☐-B.
Servo support software (MR configurator2) (For setup the parameter of servo amplifier and the graph indication, etc.)	SW1DNC-MRC2-E	1	
Communication cable (Communication cable between personal computer and servo amplifier for setup software)	MR-J3USBCBL3M	1	
SSCNET III cable	MR-J3BUS☐M etc (☐ inside cable length)	Amplifier number	
Travel axis unit	—	1	
RT ToolBox3	3F-14C-WINE/ 3F-15C-WINE/ 3F-16D-WINE	1	For details on the specifications of the personal computer, please refer to "RT ToolBox 3 instruction manual".

6.2.2.2. Example of system configration

Example of base coordinate cooperative controlsystem is below.

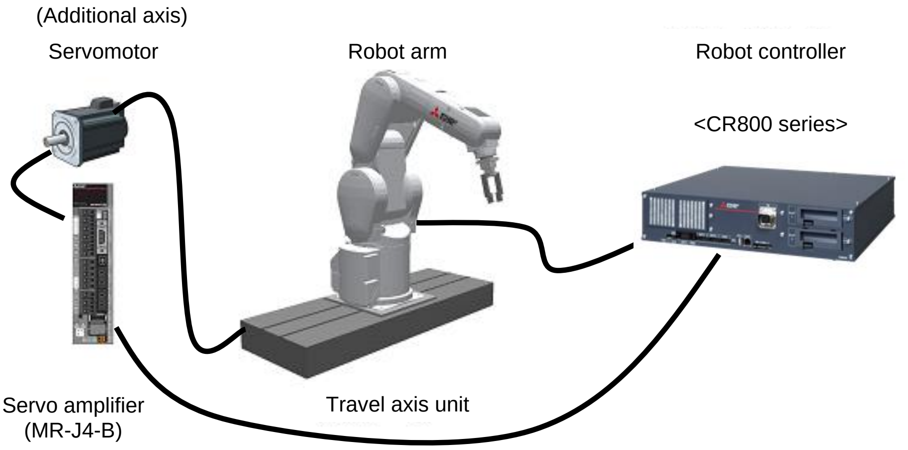


Figure 6-6 Example of system configuration of base coordinate cooperative control.

* For details of connection method, refer to Chapter 5 of "Additional axis function instruction manual (BFP-A3504)".

6.2.3 Specification

(1) Basic specification

Basic function is below.

Table 6-4 Basic specification of base coordinate corporative control.

<table>
<tr><th colspan="2">Item</th><th>Specification</th></tr>
<tr><td colspan="2">Usable robot</td><td>Vertical Multiple-joint roobts, Horizontal multiple-joint robots</td></tr>
<tr><td colspan="2">Usable robot language</td><td>MELFA-BASIC VI

・ State variable
<table>
<tr><th>Variable Name</th><th>Explanation</th><th>Page</th></tr>
<tr><td>M_BsCoopMd</td><td>Setting of base coordinate cooperative controlof spline interporation.</td><td>6-93</td></tr>
<tr><td>P_BsCdCurr</td><td>Current base coordinate</td><td>6-93</td></tr>
</table></td></tr>
<tr><td colspan="2">Definition of base coordinate</td><td>If the function is valid, set base coordinate by the parameter described in "Attachment of base coordinate".
If the function is invalid, set base coordinate by the conventional base data.</td></tr>
<tr><td rowspan="3">Attachment of base coordinate</td><td>Usable mechanisms</td><td>It is possible to attach base coordinate to robot additional axis
(7 axis or 8 axis)
* Only direct-acting axis is usable. (Rotation axis is not allowed)</td></tr>
<tr><td>Attachment method</td><td>Set the attach axis by using parameter of BSSYNC.</td></tr>
<tr><td>Attachment position</td><td>Set the attach position and posture by using parameter of BSCALIB.</td></tr>
<tr><td colspan="2">Move base coordinate</td><td>You can move only base coordinate of robot (mechanism 1).
You can move base coordinate by jog operation, joint interpolation, linear interpolation, circular interpolation, spline interpolation.</td></tr>
<tr><td colspan="2">Operation method of additional axis</td><td>・Synchronous movement mode（Initial state at power on）
The additional axis moves from the start point to the end point of interpolation.
* Additional axis don't move in the case of circular interpolation or spline interpolation.
・Relative movement mode
Assign one of the X / Y / Z axis motion in spline interpolation to the additional axis. The relative movement amount from the start point in the specified axis direction is operated by the additional axis and after that, the robot moves to compensate for the remaining amount of movement.
* If two additional axes exist, additional axis which is not used for cooperation will not operate.</td></tr>
<tr><td colspan="2">Movement area check</td><td>Execute moving area check of teaching position.</td></tr>
</table>

(2) Restrictions
Restriction of this function is below.

Table 6-5 Restrictions of base coordinate cooperative control

Restrictions		Contents
Interpolation processing		It is possible to use Mov, Mvs, Mvr, Mvr2, Mvr3, Mva command, Spline interpolation.
Peculiar point transit interpolation. Interporation of Ex-Tcontrol		If the additional axis moves, this function cannot be executed. (Error L2661 (Interpolation cannot be executed) is occurred) * If the additional axis don't move, it is possible to execute.
Circular interpolation (Mvc, EMvc) Ex-T spline interpolation		It does not move in the additional axis.
Acceleration/deceleration processing		Optimum acceleration / deceleration is unavailable. The robot moves with a fixed acceleration / deceleration. (Oadl On command is ignored.)
Combined use with other function	Tracking function	Cannot be used together. Traking function takes priority. (Error L2661 (Cannot be used (base coop)) is occurred.)
	Synchronous control of additional axis.	Cannot be used together. Synchronous control of additional axis takes priority. (Error L2661 (Cannot be used (base coop)) is occurred.)
	Jrc command	In the case of additional axis which is the target of base coordinatecooperative control, this command cannot be used. (Error L2661 (Jrc cannot be executed) is occurred.)
	Interference avoidance function	Cannot be used together. Interference avoidance function takes priority. (Error L2661 (Cannot be used (base coop)) is occurred.)
Change setting of base coordinate		The value that is calculated by base coordinate cooperative control is used as base data of the robot. * Traditional base setting such as base command and parameter MEXTL, MEXBSNO etc cannot be used.
3D display of RT ToolBox3		Operation by base coordinate cooperative control is not displayed at 3D monitor of RT ToolBox 3.
The setting of Horizontal type (4 axis), Ceiling type, Vertical type (5 axis) robot		Base coordinate cooperative control can be used only when the installation plane of the robot is parallel to the world coordinate system XY plane.
PtoJ()/JtoP()		Execute coordinate change by reflecting the settings of the base coordinate cooperative control. If there is no additional axis data in the argument, calculate the additional axis value as 0.

Points of attention

- Before enabling the base coordinate coordinated control with the parameter BSSYNC, it is necessary to complete the origin setting of the robot and additional axis. If base coordinate coordination control is enabled with the origin setting not completed, error L2660 (Origin unsetting (base coop)) will occur at power on, base coordinate cooperative control will not function.
- When the base coordinate cooperative control is functioning, if you reset the additional axis to which the base coordinate is attached, the error H2663 (Origin data was changed) will be generated. In that case, turn the power of the robot controller again and reset the error.
- It can not be used while switching base coordinate cooperative control on / off.
- The base data of the robot is calculated using the position data of the additional axis. If the position of the additional axis is not registered in the teaching data, base data is calculated using the position of the additional axis when using the teaching data.
- Behavior of jog movement (R56TB, RT ToolBox3)
  Joint jog:
  When additional axis is operated, each joint angle of the robot is held and operated. The current position of Cartesian coordinates changes according to the motion of the additional axis.
  Other jogs:
  When the additional axis is operated, the current position of rectangular coordinates is held and operated. Although the position of the control point of the robot does not change, the joint angle of the robot changes according to the motion of the additional axis.
  The additional axis jog of R32TB is the operation of the above joint jog.

6.2.4 Operation procedure

Operation start

↓

1.Setting of additional axis
Set additional axis (traveling axis) that you use.
(1) Set additional axis
(2) Execute origin setting of additional axis.

↓

2.Setting of base coordinate offset of additional axis Refer to 6.1
Set base coordinate offset data of additional axis.

↓

3.Attachment of robot base coordinate to additional axis Refer to 6.2.5
Set the parameter of attachment of base coordinate.
(1) Set the axis number to be attached.
(2) Set the position and posture to be attached.

↓

4.Creation of robot program
Create robot program and write to robot controller.

↓

5.Exexute operation
Execute saved robot program.

↓

Operation end

6.2.5 Parameter setting

The parameter list which is used is below.

Table 6-6 The list of parameter of base coordinate cooperative control

Parameter	Parameter name	Element number	Explanation of contents	Factory setting
Base coordinate synchronization setting	BSSYNC	Integer 2	Enable or disable base coordinate cooperative control, and specify the additional axis to which base coordinate is attached. Element1 : Enable or disable base coordinate cooperative control. 0: Disable 1: Enable Element 2 : specify the additional axis to which base coordinate is attached. 1: L1(J7)axis 2: L2(J8)axis	0, 1
The position and posture of attachment of base coordinate	BSCALIB	Integer 6	Position relationship setting of base coordinate and additional axis coordinate. Element 1 : Base coordinate mounting position offset in the X axis direction from the origin of the axis mechanical interface coordinate system [mm] Element 2 : Base coordinate mounting position offset in the Y axis direction from the origin of the axis mechanical interface coordinate system [mm] Element 3 : Base coordinate mounting position offset in the Z axis direction from the origin of the axis mechanical interface coordinate system [mm] Element 4 : Base coordinate mounting posture offset around X axis of mechanical interface coordinate system [deg] Element 5 : Base coordinate mounting posture offset around Y axis of mechanical interface coordinate system [deg] Element 6 : Base coordinate mounting posture offset around Z axis of mechanical interface coordinate system [deg]	0.0, 0.0, 0.0, 0.0, 0.0, 0.0

6.2.6 Creatinn of robot program

In this chapter, robot program language MELFA-BASIC VI which is used in this function is explained.

6.2.6.1. The list of robot status variable.

The list of robot status variable related to this function is below.

Table 6-7 The list of robot status variable

Variable name	Number of array	Function	Attribute (*1)	Data type
M_BsCoopMd	-	Enable the base coordinate cooperative control by spline interpolation.	R/W	Integer
P_BsCdCurr	-	Get the current base coordinate.	R	Position

(*1) R: Only read is possible. R/W: Read and write is possible.

6.2.6.2. Detail explanation of robot(system) status variable

Detail of status variable of this function is below.

The meanings of the items used to explain the status variables are as follows.

【Function】 : This indicates a function of a variable.
【Format】 : This indicates how to enter arguments of valiable.
[ ] means that arguments can be omitted. System status variables can be used in conditional expressions, as well as reference and assignment statements. In the format example, only reference and assignment statements are given to make the description simple.
【Terminology】 : This indicates the meaning and range of an argument.
【Reference program】 : An example program using variables is shown.
【Explanation】 : This indicates detailed functions and precautions.

M_BsCoopMd

【Function】
When the coordinated control for base coordinate is effective, specify and refer to the direction to which additional axis moves at spilne interpolation operation.

【Format】

```
M_BsCoopMd = <Coordinate axis number>
<Numeric variable> = M_BsCoopMd
```

【Terminology】

<Coordinate axis number> Specify the coordinate axis direction thet decides the additional axis operation.

Setting range：0 to 3
0（Not assigned）,1=（X axis direction）,2=（Y axis direction）
3=（Z axis direction）

<Numeric variable> Specify the numerical variable of the assignment target.

【Reference program】

```
1 M_BsCoopMd = 2 ' Assign Y axis direction to additional axis at spline interpolation operation.
2 MvSpl 1,50,100 ' Execute the spline interpolation.
```

【Explanation】
(1) When the coordinated control for base coordinate is effective, specify the additional axis direction at slpine interpolation.
(2) The initial state immediately after turning on the power supply is 0.
(3) This value returns to 0 at the end of main program and by program reset operation.
(4) When specify the direction in which the additional axis can not move (for example, specify the X axis with respect to the additional axis that can move in the Y axis direction), the additional axis will not move.
(5) When you refer to or set this status variable, if the coordinated control for base coordinate is disable (MELFA Smart Plus card is not installed,parameter “SMART+1” is not set correctly, parameter ”BSSYNC” is not set correctly), error L3781 (MELFA Smart Plus command can not be used.) occurs.
(6) It is a status variable dedicated to mechanism 1 (Robot), and control privilege of mechanism 1 is necessary when setting (reference is unnecessary).

P_BsCdCurr

【Function】
Refer to current robot base coordinate data.

【Format】

```
<Position variable> = P_BsCdCurr
```

【Terminology】

<Position variable> Specify the position variable of the assignment target.

【Reference program】

```
1 P1 = P_BsCdCurr 'Assign the current robot base coordinate data to P1.
```

【Explanation】
(1) Refer to the current position at base coordinate origin of robot (mechanism 1).
(2) When you refer to or set this status variable, if the coordinated control for base coordinate is disable (MELFA Smart Plus card is not installed,parameter “SMART+1” is not set correctly, parameter ”BSSYNC” is not set correctly), error L3781 (MELFA Smart Plus command can not be used.) occurs.

6.3 Additional axis tracking

6.3.1 Specification

With this function, the robot can execute another interpolation operation following the work attached to the user mechanisms.

Figure 6 6 shows an example of how the robot follows the workpiece and traces the outline of the workpiece.

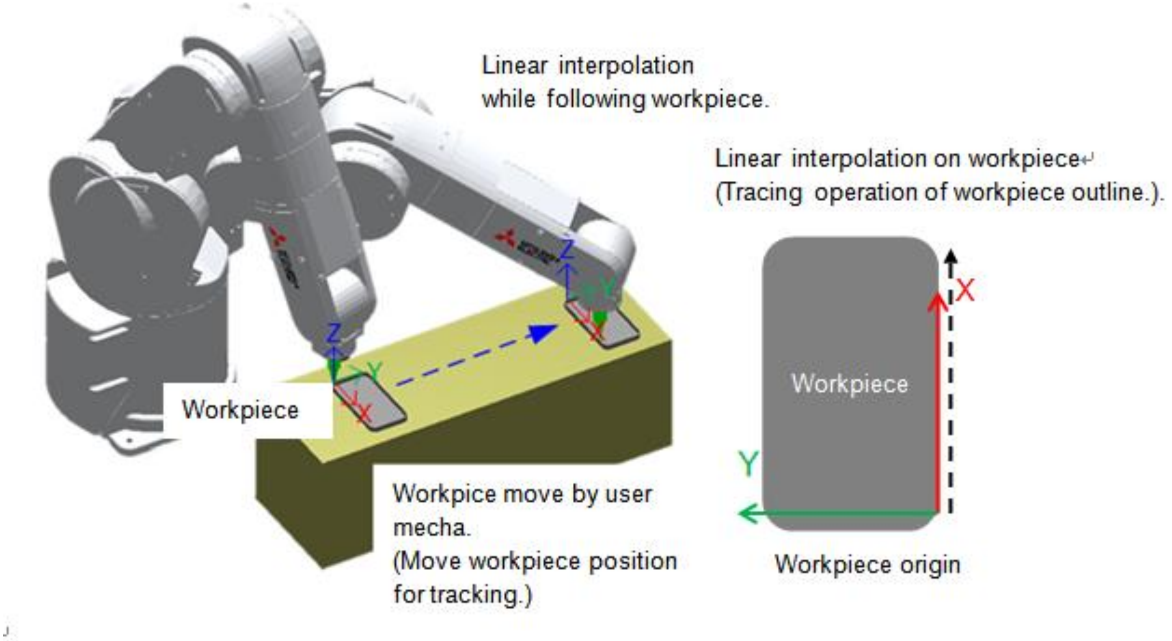


Figure 6-7 Move image of additional axis tracking.

6.3.2 System configuration

6.3.2.1. Customer preparation equipment

The following table shows the equipment that customers need to prepare, which is necessary for the additional axis tracking system.

Table 6-8 The list of customer preparation equipment

Equipment name	Format	Quantity	Remarks
Servo amplifier, servomotor, options, peripheral device	Refer to remarks	-	Refer to "Instruction Manual for Servo Amplifier and Servomotor".
Battery (For absolute position detection system)	MR-BAT6V1SET	Amplifier number	The battery case (MR-BT6VCASE) and the battery (MR-BAT6V1) are necessary when using MR-J4W☐-B.
Servo support software (MR configurator2) (For setup the parameter of servo amplifier and the graph indication, etc.)	SW1DNC-MRC2-E	1	
Communication cable (Communication cable between personal computer and servo amplifier for setup software)	MR-J3USBCBL3M	1	
SSCNET III cable	MR-J3BUS☐M etc (☐ inside cable length)	Amplifier number	
Travel axis unit	-	1	
Hand	-	(1)	
Hand sensor	-	(1)	Used to confirm that workpieces are gripped correctly. Provide as necessary.
Solenoid valve set Hand input cable	Refer to remarks	(1)	Different models are used depending on the robot used. Check the robot version and provide as necessary.
Calibration jig	-	(1)	This is a jig with a sharp tip that is attached to the mechanical interface of the robot arm and used for calibration tasks. It is recommended to use the jig if high precision is required.
RT ToolBox3	3F-14C-WINE/ 3F-15C-WINE/ 3F-16D-WINE	1	For details on the specifications of the personal computer, please refer to "RT ToolBox 3 instruction manual".

6.3.2.2. Example of system configuration

Example of additional axis tracking system is below.

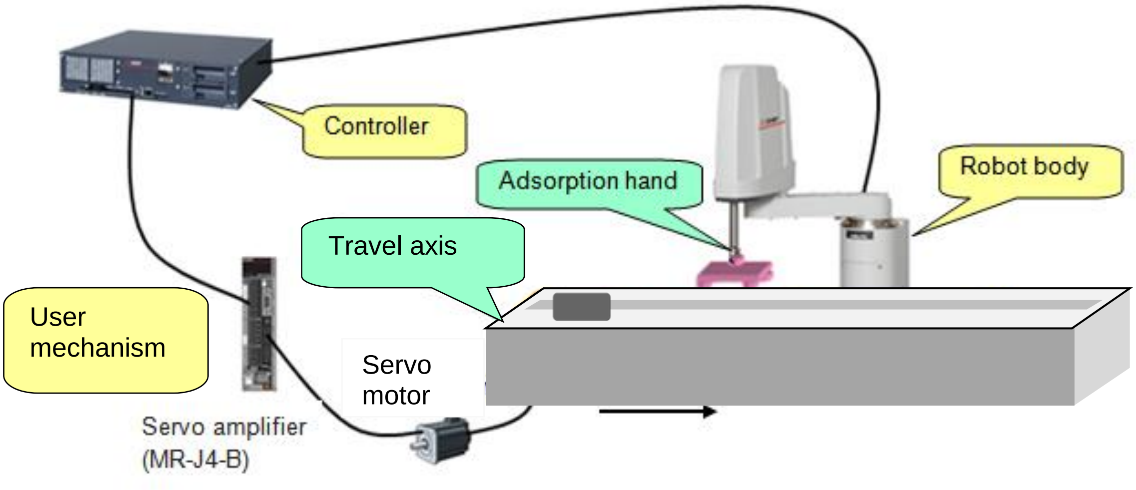


Figure 6-8 Example of additional axis tracking system configuration

* For details of connection method, refer to Chapter 5 of "Additional axis function instruction manual (BFP-A3504)".

6.3.3 Specification

(1) Basic specification

Basic function is below.

Table 6-9 Basic specification of additional axis tracking

<table>
<tr><th colspan="2">Item</th><th>Specification</th></tr>
<tr><td colspan="2">Usable robot</td><td>Vertical Multiple-joint roobts, Horizontal multiple-joint robots</td></tr>
<tr><td colspan="2">Usable robot language</td><td>MELFA-BASIC VI

・ Status variable
<table>
<tr><th>Variable name</th><th>Explanation</th><th>Page</th></tr>
<tr><td>M_AxTrkWkNo</td><td>Workpiece number used by additional axis tracking.</td><td>6-105</td></tr>
<tr><td>P_WkCalib</td><td>Offset amount of axis to which workpiece coordinate attached.</td><td>6-106</td></tr>
<tr><td>P_TrkPAcl</td><td>Additional axis tracking acceleration acceadjustment coefficient.</td><td>6-107</td></tr>
<tr><td>P_TrkPDcl</td><td>Additional axis tracking deceleration acceadjustment coefficient.</td><td>6-107</td></tr>
<tr><td>P_TrkBase</td><td>Workpiece reference position used by additional axis tracking.</td><td>6-108</td></tr>
<tr><td>P_TrkTarget</td><td>Workpiece current position used by additional axis tracking.</td><td>6-109</td></tr>
</table>
・ Command
<table>
<tr><th>Command name</th><th>Explanation</th><th>Page</th></tr>
<tr><td>Trk</td><td>Enable additional axis tracking.</td><td>6-111</td></tr>
</table></td></tr>
<tr><td rowspan="3">Attachment of workpiece coordinate</td><td>Usable mechanisms</td><td>J1/J2/J3 axis of user mechanisms.</td></tr>
<tr><td>Attach method</td><td>For each workpiece number, set the axis to which workpiece coordinate attached.
・ Parameter “WK1SYNC” to “WK8SYNC”</td></tr>
<tr><td>Attach position</td><td>Setting of mounting position and posture is below.
・ Status variable "P_WkCalib"
・ Parameter "WK1CALIB"to "WK8CALIB"</td></tr>
<tr><td colspan="2">Workpiece move</td><td>In the case of user mechanisms, it is possible to move workpiece.</td></tr>
<tr><td colspan="2">Follow the workpiece</td><td>* User mechanisms cannot be used.</td></tr>
<tr><td colspan="2">Movement area check
at the following of the workpiece.</td><td>In the case of workpiece coordinate following control, the movement area at command execution is not checked.</td></tr>
</table>

(2) Restrictions
Restriction of this function is below.

Table 6-10 Restrictions of additional axis tracking

Restrictions		Contents
Interporation processing		Mov, Mvs, Mvr, Mvr2, Mvr3, Mvc command is available. Mva command, Ex-Tcontrol interpolation, Spline/Ex-T spline interpolation is unavailable.
Singular point passage		Unavailable.
Optimum acceleration / deceleration		Optimum acceleration / decelerationis unavailable. * If the Optimum acceleration / deceleration is effective, robot moves by a fixed acceleration / deceleration.
Combined usewith other functions	Tracking function (Tracking using external encoder)	Cannot be used together. Switch with the parameter "TRMODE". TRMODE = 1: Conveyor tracking TRMODE = 2: Additional axis tracking
	Additional axis synchronization control	Cannot be used together. Additional axis synchronization control takes precedence. When power is ON, error L2661 (Cannot be used (axis trk)) is occurred.
Change setting of work		If you try to change the mounting position of workpiece that is being tracked with the parameter "WKnCALIB" or the state variable P_WkCalib while additional axis tracking is in effect (after execution of Trk On command), error L2662 (Cannot be used (axis trk)) is occurred.
Usable user mechanisms		It can be used only for linear movement 1 axis.

6.3.4 Operation procedure

Operation start

↓

1.Setting of user mechanisms
Set the user mechanisms that you use.
(1) Set the mecha number and axis number that you use.
(2) Perform advanced setting of user mechanisms that you use.

↓

2.Setting parameter Refer to 6.3.5
Parameter setting related to additional axis tracking is explained.

↓

3. Installing the sample programs Refer to 6.3.7
The functions related to the sample program is explained.

↓

4.Calibration between robot and base (“A1” program) Refer to 6.3.8
The work procedure for calculating the position of user mechanisms in the robot coordinate is explaned.

↓

5.Registering work location of the user mechanisms (“B1” program) Refer to 6.3.9
The work procedure for registering work position of start and end of user mechanisms is explained.

↓

6.Registering workpiece reference position (“C1” program) Refer to 6.3.10
The work procedure for registering the reference position of the target workpiece is explained.

↓

7. Setting operation condition and checking operation (“1” program) Refer to 6.3.11
Adjusting position in automatic operation is explained.

↓

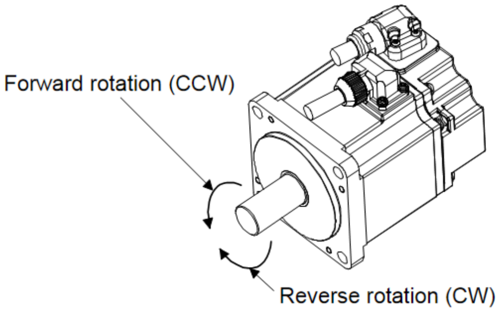
Operation end

6.3.5 Parameter setting

The parameter which is used is below.

Table 6-11 The list of parameter of additional axis tracking

Parameter	Parameter name	Element number	Content explanation	Factory setting
Workpiece synchronization settings	WKnSYNC (n = 1～8)	Integer 2	Set the mechanism number and axis number attached to workpiece. Element1：mechanism number 0: Disable 2: mechanism 2 3: mechanism 3 Element2：Axis number 1/2/3: 1/2/3 axis of corresponding mechanism.	0, 1
Workpiece mounting position、posture data	WKnCALIB (n = 1～8)	Real number 6	Set the position relationship of workpiece and coordinate ofattached axis. Element 1：Mounting position offset in the X axis direction from the origin of the mounting axis. [mm] Element 2：Mounting position offset in the Y axis direction from the origin of the mounting axis. [mm] Element 3：Mounting position offset in the Z axis direction from the origin of the mounting axis. [mm] Element 4：Mounting posture offset around X axis of the munting axis. [deg]. Element 5：Mounting posture offset around Y axis of thr mounting axis. [deg]. Element 6：Mounting posture offset around Z axis of the mounting axis. [deg]. (The direction of rotation is the same as the coordinate system of the robot.)	0.0, 0.0, 0.0, 0.0, 0.0, 0.0
Tracking mode	TRMODE	Integer 1	Specify tracking disable/tracking/additional axis tracking. 0: Tracking disable 1: External encoder tracking 2: Additional axis tracking	0
Tracking Acceleration(*1)	TRPACL	Real number 8	Tracking acceleration. Acceleration during execution of tracking movement.	1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0
Tracking Deceleration(*1)	TRPDCL	Real number 8	Tracking deceleration. Deceleration during execution of tracking movement.	1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0
Number of multi mechanisms used(*2)	AXUNUM	Integer 1	The number of multi-mechanism to use. (The robot and the mechanism of mechanism number 1 are exclude.)	0
Mechanism No. designation(*2)	AXMENO	Integer 16	Input the mechanism No. to the element which corresponds to the servo control axis No. used and be sure to set "0" for the axis not used.	0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ,0, 0

Parameter	Parameter name	Element number	Content explanation	Factory setting
Setting axis No. (*2)	AXJNO	Integer 16	Designate what number of the axis of the robot arm is used for the additional axis.	0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ,0, 0
Unit sstem(*2)	AXUNT	Integer 16	Unit system of additional axis 0: Angle[deg] 1: Length[mm] 2: Length[mm] Using linear servo	0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ,0, 0
Rotate direction(*2)	AXSPOL	Integer 16	Set the rotation direction of the motor. 0: Forward(CCW) 1: Reverse(CW) The direction of rotation of the servomotor is as follows. 	0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ,0, 0
Acceleration time(*2)	AXACC	Real numbe 16	Acceleration time for additional axis [sec].	0.20, 0.20, 0.20, 0.20, 0.20, 0.20, 0.20, 0.20, 0.20, 0.20, 0.20, 0.20, 0.20, 0.20, 0.20, 0.20
Deceleration time(*2)	AXDEC	Real number 16	Deceleration time for additional axis [sec].	0.20, 0.20, 0.20, 0.20, 0.20, 0.20, 0.20, 0.20, 0.20, 0.20, 0.20, 0.20, 0.20, 0.20, 0.20, 0.20
Total speed ratio numerator(*2)	AXGRTN	Integer 16	Total speed ratio numerator of additional axis	1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1
Total speed ratio denominator(*2)	AXGRTD	Integer 16	Total speed ratio denominator of additional axis	10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10

Parameter	Parameter name	Element number	Content explanation	Factory setting
Rated speed(*2)	AXMREV	Integer 16	Rated speed (Unit: r/min.) of motor or Rated speed (Unit: mm/s.) of linear motor	2000, 2000, 2000, 2000, 2000, 2000, 2000, 2000, 2000, 2000, 2000, 2000, 2000, 2000, 2000, 2000
Maximum speed(*2)	AXJMX	Integer 16	Maximum speed (Unit: r/min.) of motor or Maximum speed (Unit: mm/s.) of linear motor	3000, 3000, 3000, 3000, 3000, 3000, 3000, 3000, 3000, 3000, 3000, 3000, 3000, 3000, 3000, 3000
Encoder resolution(*2)	AXENCR	Integer 16	Encoder resolution of motor (Unit: pulse/rev)	4194304, 4194304, 4194304, 4194304, 4194304, 4194304, 4194304, 4194304, 4194304, 4194304, 4194304, 4194304, 4194304, 4194304, 4194304, 4194304
JOG Smoothing time constant(*2)	AXJOGTS	Real numbe 16	If it vibrates at JOG, set a larger value. (Unit: ms)	150.00, 150.00, 150.00, 150.00, 150.00, 150.00, 150.00, 150.00, 150.00, 150.00, 150.00, 150.00, 150.00, 150.00, 150.00, 150.00

Parameter	Parameter name	Element number	Content explanation	Factory setting
Joint movement range(*2)	MEJAR	Real numbe 16	Set the overrun limit value for the joint coordinate system. Sets the movement range for each axis. Expanding of the movement range is not recommended, since there is possibility that the robot may strike the mechanical stopper. Note) Please note that the joint movement range of J1 axis cannot be changed after the J1 axis offset angle (J1OFFSET) is specified in vertical 5-axis type robot. Set the minus and plus directions. (-J1,+J1,-J2,+J2,......-J8,+J8) Unit:deg	Setting value for each mechanism
User-designated origin(*2)	USERORG	Real numbe 8	Designate the user-designated origin position. This normally does not need to be set. (J1,J2,J3,J4,J5,J6,J7,J8) Unit：deg	0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00

*1: For details of parameters, refer to chapter 18.1 “List of Parameters Related to Tracking” of “Tracking Function Instruction Manual (BFP-A3520)”.

*2: For details of parameters, refer to chapter 7.1 “Description of parameters” of “ADDITIONAL AXIS FUNCTION INSTRUCTION MANUAL (BFP-A3504)”.

6.3.6 Creation of robot program

In this chapter, robot program language MELFA-BASIC VI which is used in this function is explained.

6.3.6.1. The list of robot status variable

The list of status variable related to this function is below.

Table 6-12 The list of robot status variable

Variable name	Number of array	Function	Attribute (*1)	Data type
M_AxTrkWkNo	-	Specify and refer to the workpiece which is the target of the additional axis tracking.	R/W	Interger
P_WkCalib	1	Specify and refer to offset amount of axis to which work coordinate is attached.	R/W	Position
P_TrkPAcl	1	Specify the additional axis tracking acceleration adjustment coefficient.	R/W	Position
P_TrkPDcl	1	Specify the additional axis tracking deceleration adjustment coefficient.	R/W	Position
P_TrkBase	1	Refer to the workpiece reference position which is the target Of the additional axis tracking.	R/W	Position
P_TrkTarget	-	Refer to the workpiece current position which is the target of the additional axis tracking.	R	Position

(*1) R・・・Only read is possible. R/W・・・Read and write is possible.

6.3.6.2. Detail explanation of robot status variable

Detail of status variable of this function is below.

The meanings of the items used to explain the status variables are as follows.

【Function】 : This indicates a function of a variable.
【Format】 : This indicates how to enter arguments of valiable.
[ ] means that arguments can be omitted. System status variables can be used in conditional expressions, as well as reference and assignment statements. In the format example, only reference and assignment statements are given to make the description simple.
【Terminology】 : This indicates the meaning and range of an argument.
【Reference program】 : An example program using variables is shown.
【Explanation】 : This indicates detailed functions and precautions.

M_AxTrkWkNo

【Function】

Specify and refer to the Workpiece number used by additional axs tracking.

【Format】

```
M_AxTrkWkNo= < Workpiece number >
<Position variable> = M_AxTrkWkNo
```

【Terminology】

<Workpiece number>	Specify the target work number. Setting range：0～8
<Position variable>	Specify the position variable to assign.

【Reference program】

```
1    M_AxTrkWkNo = 1          ' Set the workpiece used by additional axis tracking 1.
```

【Explanation】

(1) Execute additional axis tracking by using workpiece specified at <workpiece number>.
(2) Initial value is 0(Not set).
(3) The value assigned to the status variable is retained until the power is turned off.
(4) Attempting to change will result in error L2662 (Work setting cannot be changed).
(5) It is impossible to change workpiece number while additional axis tracking is in effect.If you try tochange, error L2662_00000 occurs.
(6) When you refer to or set this status variable, if the additional axis tracking is disable (MELFA Smart Plus card is not installed, parameter “SMART+1” is not set correctly, parameter “TRMODE” is not set correctly), error L3781 (MELFA Smart Plus command can not be used.) occurs.

P_WkCalib

【Function】

Specify and refer to the position relationship of workpiece reference point and mounting target axis coordinate.

【Format】

```
P_WkCalib [(<Workpiece number>)] = < Position variable 1>
<Position variable 2> = P_WkCalib [(<Workpiece number>)]
```

【Terminology】

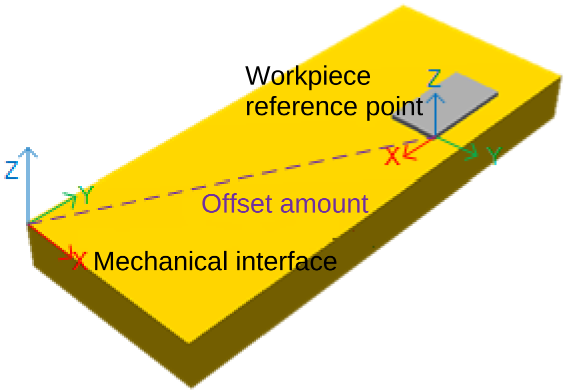
<Workpiece number>	Specify the target workpiece number. Setting range：1 to 8, When it is ommited, it will be treated as 1 (0 is also treated as omission.) Setting value 1 to 8 correspond to parameter WK1CALIB to WK8CALIB.
<Position variable1>	Specify the offset amount from mechanical interface of axis to which workpiece attached.
< Position variable2>	Specify the position variable to assign.

【Reference program】

```
1    P_WkCalib(1) = P1          ' The mounting offset of workpiece No. 1 is set as P1.
```

【Explanation】

(1) Specify and refer to the position relationship of workpiece reference point which is specified by <Workpiece number> and mechanical interface coordinate origin of mounting target axis. When placing a workpiece reference point on the direct drive axis as below, specify the amount of change in position and posture from the mechanical interface coordinate system origin to the workpiece reference point as the offset amount.



(2) Initial value is set value of parameter "WK1CALIB" ~ "WK8CALIB"
(3) When you reset the program, it becomes the setting value of the corresponding parameter "WKnCALIB".
(4) When the variable value is changed, it is not reflected to the parameter "WK1CALIB" to "WK8CALIB". The value assigned to the status variable is retained until the power is turned off.
(5) The value assigned to the status variable is held until the power is turned off.
(6) When the additional axis tracking is effective, you can not change workpiece mounting position.
If you ttry to change, error L2662 (Work setting cannot be changed) occurs.
(7) When you refer to or set this status variable, if the additional axis tracking is disable (MELFA Smart Plus card is not installed, parameter “SMART+1” is not set correctly, parameter “TRMODE” is not set correctly), error L3781 (MELFA Smart Plus command can not be used.) occurs.

P_TrkPAcl / P_TrkPDcl

【Function】

Change the tracking acceleration/deceleration coefficient of the parameter “TRPACL/TRPDCL”.

【Format】

```
P_TrkPAcl(<Condition number >) = <Position data>
<Position variables> = P_TrkPAcl(<Condition number>)
P_TrkPDcl(<Condition number>) = <Position data>
<Position variables> = P_TrkPDcl(<Condition number>)
```

【Terminology】

<Condition number>	Specify the condition number corresponding to the tracking. Setting range：1 to 8
<Position data>	Specify the tracking acceleration/deceleration coefficient. Setting range: For each component, 0.10 to 10.0

【Reference program】

```
1   P_TrkPAcl(1) = (0.2,0.2,0.2,1.0,1.0,1.0,1.0,1.0)
                          ' Specify the tracking acceleration coefficient.
2   P_TrkPDcl(1) = (0.2,0.2,0.2,1.0,1.0,1.0,1.0,1.0)
                          ' Specify the tracking deceleration coefficient.
```

【Explanation】

(1) Specify the acceleration/deceleration coefficient of the additional axis tracking.
(2) You can confirm the tracking acceleration/ deceleration coefficient by referencing “P_TrkPAcl/P_TrkPDcl”.
(3) You can omit the step to specify <Condition number>.When it is omitted, condition number will be treated as "1."
(4) Number which you can enter to specify <Condition number> is an integer in the range of "1" to "8." Entering anything else causes L3110 (Argument value range over) error to occur.

P_TrkBase

【Function】
Specify and refer to the origin of the workpiece to be followed by additional axis tracking.
The robot moves to the relative position correspond to this reference point by the movement instruction during the additional axis tracking

【Format】

```
P_TrkBase(<Condition number>) = <Position data>
<Position variable> = P_TrkBase(<Condition number>)
```

【Terminology】

< Condition number >	Specify the condition number corresponding to the tracking. Setting range：1 to 8
< Position data >	Specify the base position of the tracking.
< Position variables >	Return the base coordinates of the specified tracking.

【Reference program】

```
1    P_TrkBase = PTBASE     ' Specify the origin of the workpiece used by additional axis tracking.
```

【Explanation】
(1) Specify the workpiece coordinate system origin used by additional axis tracking.
(2) You can confirm the workpiece coordinate system origin by referencing “P_TrkBase”.
(3) You can omit the step to specify <Condition number>.When it is omitted, condition number will be treated as "1."
(4) Number which you can enter to specify <Condition number> is an integer in the range of "1" to "8." Entering anything else causes L3110 (Argument value range over) error to occur.

P_TrkTarget

【Function】

Refer to the workpiece current position to be followed by additional axis tracking.

【Format】

```
<Position variables> = P_TrkTarget
```

【Terminology】

<Position variables> Specifies the position variable to assign.

【Reference program】

```
1    PWrkNow = P_TrkTarget ' Specify the workpiece current position to be followed by
                             additional axis tracking.
```

【Explanation】

(1) Refer to the workpiece current position to be followed by additional axis tracking.
(2) If you execute the writing to “M_TrkTarget”, L3210 (This variable is write protected) error occurs.

6.3.6.3. The list of command

The list of command related to this function is shown below.

Table 6-13 The list of command

Command name	Function
Trk	Enable or disable additional axis tracking function.

6.3.6.4. Detail explanation of robot(system) status variable

Detail of status variable of this function is below.

The meanings of the items used to explain the state variables are as follows.

【Function】 : This indicates a function of a command word.
【Format】 : This indicates how to enter arguments of the command word.
The aurgument is shown in <>.
[ ] indicates that the argument can be omitted.
【Terminology】 : This indicates the meaning and range of an argument.
【Reference program】 : An example program using variables is shown.
【Explanation】 : This indicates detailed functions and precautions.

Trk

【Function】

After Trk ON is executed, the robot goes into the tracking mode and operates while following the conveyer operation until Trk OFF is executed.

【Format】

```
Trk On [, <Measurement position data> [, [<Encorder data>][,[<Reference position data>][,[<Encorder
 logic number>]]]]]
Trk Off
```

【Terminology】

【Reference program】

```
1    P_TrBase P0     ‘ Specify the workpiece coordinate origin at the teaching position.
2    Trk On          ' Enable additional tracking.
3    Mvs P2          ' Execute the interpolation operation following workpiece attached to user
                       mechanisms.
4    Trk Off         ' End the tracking operation.
```

【Explanation】

(1) If additional axis tracking is used (parameter TRMODE is "2"), arguments after Trk On can not be specified.

6.3.7 Installation of a sample program

This chapter explains the structure of the sample robot programs.

Poogram structures are shown in “Table 6-14 The list of sample programs”.

Table 6-14 The list of sample programs

Prgram name	Description	Explanation
A1	Program for calibration between robot and base	This program matches the coordinate systems of robot and user mechanisms.
B1	Program for workpiece position registration of user mechanisms	This program registers the start end positions of work.
C1	Program for work base position registration	This program registers the reference position of the workpieces where the robot works.
1	Robot operation program	This program operates the robot while following workpieces mounted on the user mechanisms.
CM1	User mechanisms monitoring program	This program moves user mechanisms.

6.3.8 Calibration of Robot and Base Coordinate Systems (“A1” program)

This chapter explains the tasks carried out by using “A1” program.
Calibration of robot and base coordinate systems refers to determining the origin position of user mechanisms in the robot coordinate system.
“A1” program performs specified tasks and automatically calculates the origin position of user mechanisms in the robot coordinate system.
The procedures of operations specified by “A1” program and items to be confirmed after the operations are explained below.
Please refer to “Detailed Explanations of Functions and Operations” for the steps involved in each operation.

6.3.8.1. Operation procedure

(1) Set the controller mode to "MANUAL". Set the T/B to "ENABLE".

(2) Press one of the keys (example, [EXE] key) while the <TITLE> screen is displayed. The <MENU> screen will appear.

```
MELFA CR800-D                Ver. A1
RH-3FRH5515-D

COPYRIGHT (C) 2017  MITSUBISHI
ELECTRIC CORPORATION ALL RIGHTS
RESERVED
```

→

```
<MENU>

1.FILE/EDIT        2.RUN
3.PARAM            4.ORIGIN/BRK
5.SET/INIT
                123           CLOSE
```

(3) Select "1. FILE /EDIT" screen on the <MENU> screen.

```
<MENU>

1.FILE/EDIT        2.RUN
3.PARAM            4.ORIGIN/BRK
5.SET/INIT
                123           CLOSE
```

→

```
<FILE/EDIT>

1          17-03-31 17:06:39 12345
A1         17-03-31 17:06:39 12345
CM1        17-03-31 17:06:39 12345
EDIT   POSI   123   NEW   COPY
```

(4) Press the arrow key, combine the cursor with the program name "A1" and press the [EXE] key. Display the <program edit> screen.

```
<FILE/EDIT>

1          17-03-31 17:06:39 12345
A1         17-03-31 17:06:39 12345
CM1        17-03-31 17:06:39 12345
EDIT   POSI   123   NEW   COPY
```

→

```
<PROGRAM> A1
1 '## Ver.A1 #####################
2 '# Coordinated control for addit
3 '# Program Type              : A1.
4 '# Date of creation/version : 201
EDIT   POSI   123   NEW   COPY
```

(5) Press the [FUNCTION] key, and change the function display.

```
<PROGRAM> A1
1 '## Ver.A1 #####################
2 '# Coordinated control for addit
3 '# Program Type              : A1.
4 '# Date of creation/version : 201
EDIT   DELETE   123   INSERT   TEACH
```

→

```
<PROGRAM> A1
1 '## Ver.A1 #####################
2 '# Coordinated control for addit
3 '# Program Type              : A1.
4 '# Date of creation/version : 201
FWD   JUMP   123         BWD
```

(6) Press the [F1] (FWD) key and execute step feed. “(1) Input a mechanism number ... “is displayed. Execute work according to the comment in the robot program.

```
<PROGRAM> A1
4 '# Date of creation/version : 201
5 '# COPYRIGHT                 : MIT
6 '##############################
7 '(1)Input a mechanism number to
FWD   JUMP   123         BWD
```

Specify the mechanism number.
If you want to change the mechanism number, please edit the program as follows.

(a) Display the following command.

```
<PROGRAM> A1
5 '# COPYRIGHT                 : MIT
6 '##############################
7 '(1)Input a mechanism number to
8 MMechNo = 2
EDIT  DELETE  123  INSERT  TEACH
```

(b) Press the [F1] (EDIT) key and specify the mechanism number in the variable "MMechNo".
 Example) When "3" is specified as the mechanism number.

```
<PROGRAM> A1              Edit
8 MEncNo = 2

EDIT  DELETE  123  INSERT  TEACH
```

→

```
<PROGRAM> A1              Edit
8 MEncNo = 3

EDIT  DELETE  123  INSERT  TEACH
```

(c) Press the [EXE] key and the change is determined.

```
<PROGRAM> A1
5 '# COPYRIGHT                 : MIT
6 '##############################
7 '(1)Input a mechanism number to
8 MMechNo = 3
EDIT  DELETE  123  INSERT  TEACH
```

(7) Press the [F1] (FWD) key and execute step feed. "(2) Input a parameter name ... "is displayed.

```
<PROGRAM> A1
6 '##############################
7 '(1)Input a mechanism number to
8 MMechNo = 3
9 '(2)Input a parameter name "BSWO
EDIT  DELETE  123  INSERT  TEACH
```

Specify the parameter name.
If you want to change the parameter name, please edit the program as follows.
(a) Display the following command.

```
<PROGRAM> A1
7 '(1)Input a mechanism number to
8 MMechNo = 3
9 '(2)Input a parameter name "BSWO
10 CPrmName$ = "BSWOFST1"
EDIT  DELETE  123  INSERT  TEACH
```

(b) Press the [F1] (EDIT) key and specify the mechanism number in the variable "CPrmName$".
 Example) When "BSWOFTS3" is specified as the parameter name.

```
<PROGRAM> A1              Edit
10 CPrmName$ = "BSWOFST1"

EDIT  DELETE  123  INSERT  TEACH
```

→

```
<PROGRAM> A1              Edit
10 CPrmName$ = "BSWOFST3"

EDIT  DELETE  123  INSERT  TEACH
```

(c) Press the [EXE] key and the change is determined.

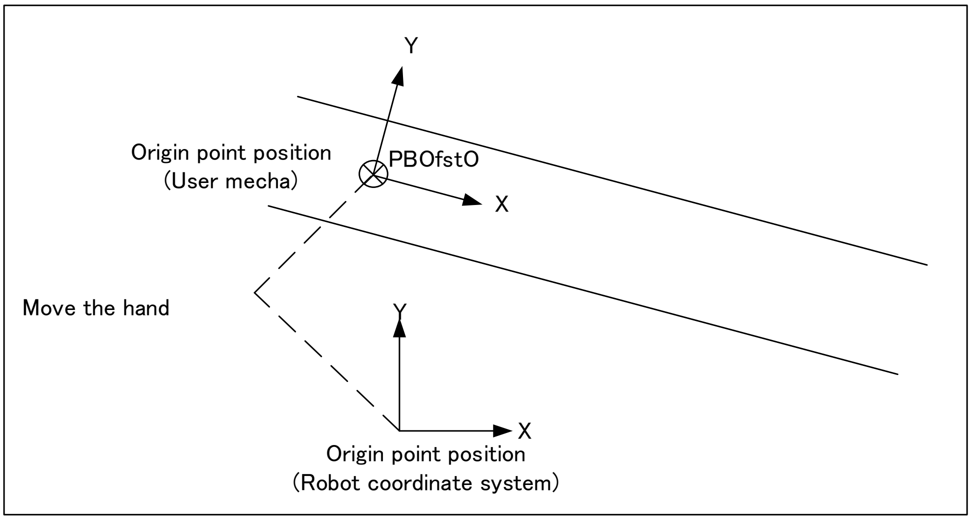
```
<PROGRAM> A1
7 '(1)Input a mechanism number to
8 MMechNo = 3
9 '(2)Input a parameter name "BSWO
10 CPrmName$ = "BSWOFST3"
EDIT  DELETE  123  INSERT  TEACH
```

(8) Press the [F1] (FWD) key and execute step feed. “(3) Move the robot to ... “is displayed.

```
<PROGRAM> A1
8 MMechNo = 3
9 '(2)Input a parameter name "BSWO
10 CPrmName$ = "BSWOFST1"
11 '(3)Move the robot to the origin
EDIT  DELETE  123  INSERT  TEACH
```

```
<PROGRAM> A1
9 '(2)Input a parameter name "BSWO
10 CPrmName$ = "BSWOFST1"
11 '(3)Move the robot to the origin
12 PBOfstO = P_Fbc(1)
EDIT  DELETE  123  INSERT  TEACH
```

Set the the origin of the user mechanism in the robot coordinate system.

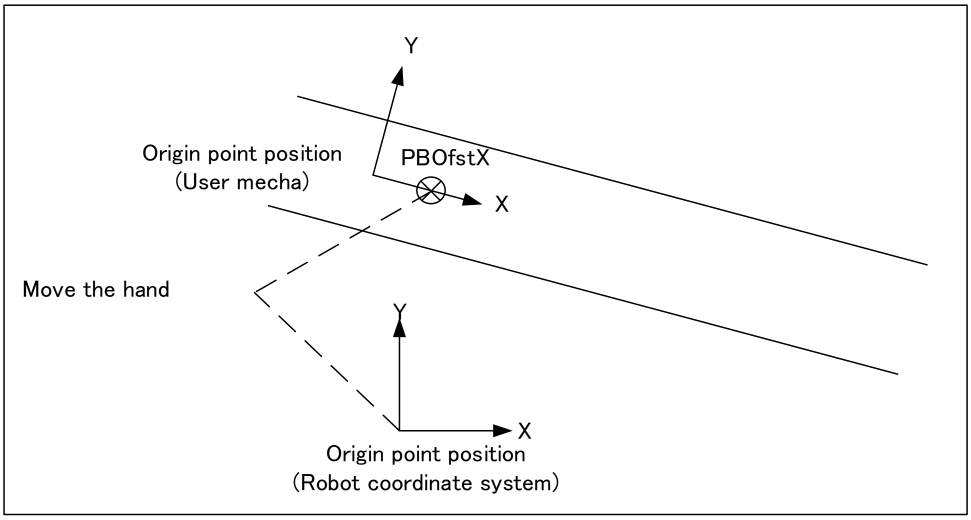


(9) Press the [F1] (FWD) key and execute step feed. “(4) Move the robot to ... “is displayed.

```
<PROGRAM> A1
10 CPrmName$ = "BSWOFST1"
11 '(3)Move the robot to the origin
12 PBOfstO = P_Fbc(1)
13 '(4)Move the robot to the X axis
EDIT  DELETE  123  INSERT  TEACH
```

```
<PROGRAM> A1
11 '(3)Move the robot to the origin
12 PBOfstO = P_Fbc(1)
13 '(4)Move the robot to the X axis
14 PBOfstX = P_Fbc(1)
EDIT  DELETE  123  INSERT  TEACH
```

Set a point on the X axis of the user mechanism origin in the robot coordinate system.

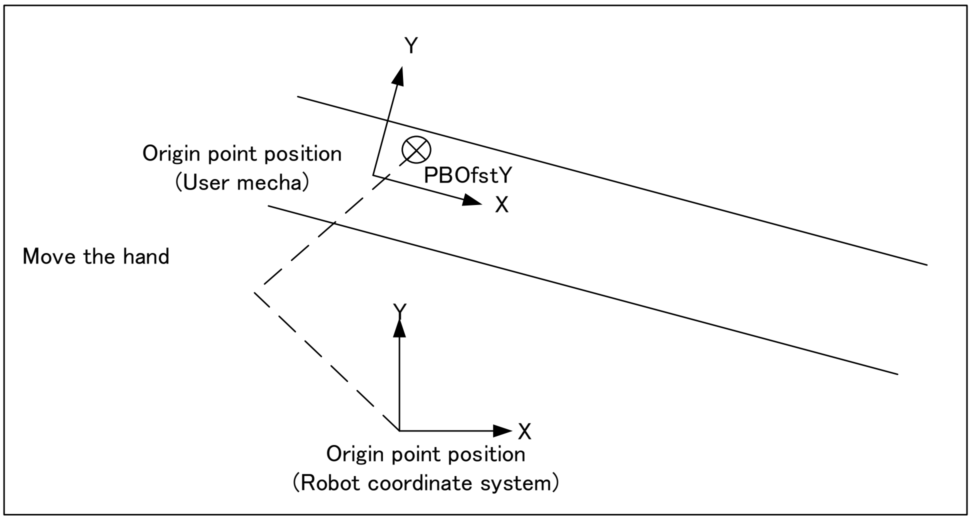


(10) Press the [F1] (FWD) key and execute step feed. “(5) Move the robot to ... “is displayed.

```
<PROGRAM> A1
12 PBOfstO = P_Fbc(1)
13 '(4)Move the robot to the X axis
14 PBOfstX = P_Fbc(1)
15 '(5)Move the robot to the +Y dire
EDIT  DELETE  123  INSERT  TEACH
```

```
<PROGRAM> A1
13 '(4)Move the robot to the X axis
14 PBOfstX = P_Fbc(1)
15 '(5)Move the robot to the +Y dire
16 PBOfstY = P_Fbc(1)
EDIT  DELETE  123  INSERT  TEACH
```

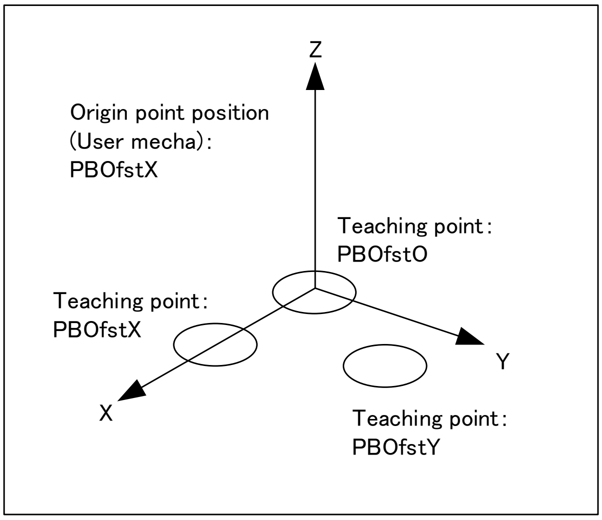
Set a point in the positive Y Direction of the X-Y plane of the user mechanism origin in the robot coordinate system.



(11) Press the [F1] (FWD) key and execute step feed. “(6) Perform step operation ... “is displayed.

```
<PROGRAM> A1
14 PBOfstX = P_Fbc(1)
15 '(5)Move the robot to the +Y dire
16 PBOfstY = P_Fbc(1)
17 '(6)Perform step operation until
EDIT  DELETE  123  INSERT  TEACH
```

```
<PROGRAM> A1
19 PrmRead MMechNo, CPrmName$, PBO
20 PBOfst = SetPos(PBOfst2.X,PBOfs
21 PrmWrite MMechNo, CPrmName$, PB
22 End
EDIT  DELETE  123  INSERT  TEACH
```



Perform step operation until “End”.
* The origin position of the user mechanism in the robot coordinate system is calculated based on this operation.

6.3.8.2. Confirmation after operation

Check the parameter “BSWOFSTn”.
* This value indicates the offset from base coordinate origin of the robot to base coordinate origin of the user mechanism.

6.3.8.3. When multiple user mechanisms are used

Carry out the same operations as above when multiple user mechanisms are used as well, but pay attention to be following points.

Example) When using mechanism 3, J2 axis
(a) Copy the “A1” program, please create an “A2” program.
(b) Please change the mechanism number for variable “MMechaNo” in the “A2” program to “3”.
(c) Please change the parameter name for variable “CPrmName$” in the “A2” program to “BSWOFST2”.

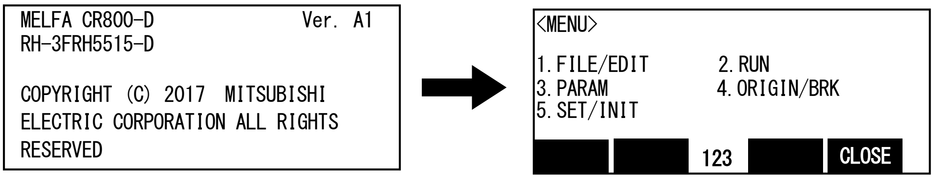
6.3.9 Resistration of User Mechanism Work Position ("B1" program)

This chapter explains the tasks carried out by using "B1" program.
"B1" program performs specified tasks and registers the work position to be used by the additional axis tracking.
The procedures of operations specified by "B1" program and items to be confirmed after the operations are explained below.
Please refer to "Detailed Explanations of Functions and Operations" for the steps involved in each operation.

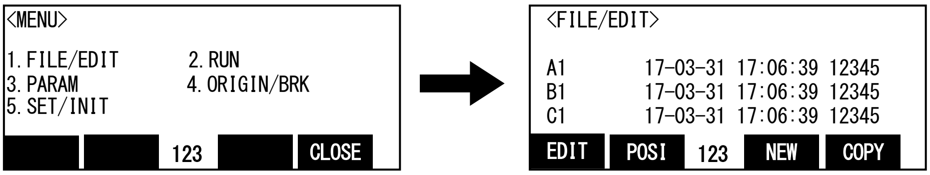
6.3.9.1. Operation procedure

(1) Set the controller mode to "MANUAL". Set the T/B to "ENABLE".

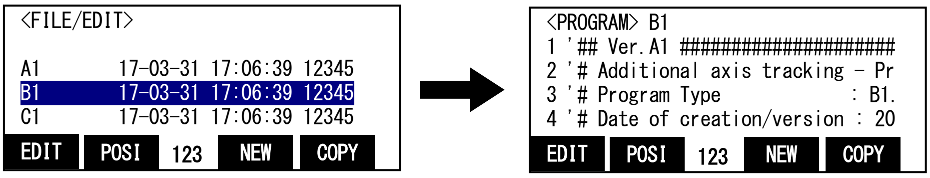
(2) Press one of the keys (example, [EXE] key) while the <TITLE> screen is displayed. The <MENU> screen will appear.



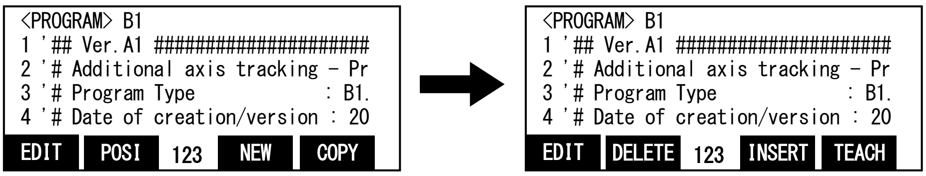
(3) Select "1. FILE /EDIT" screen on the <MENU> screen.



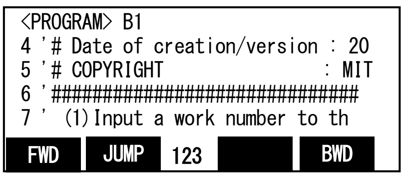
(4) Press the arrow key, combine the cursor with the program name "B1" and press the [EXE] key. Display the <program edit> screen.



(5) Press the [FUNCTION] key, and change the function display.



(6) Press the [F1] (FWD) key and execute step feed. "(1) Input a work number ... "is displayed. Execute work according to the comment in the robot program.



Specify the work number.
If you want to change the work number, please edit the program as follows.

(a) Display the following command.

```
<PROGRAM> B1
5 '# COPYRIGHT                 : MIT
6 '##############################
7 '  (1)Input a work number to th
8 MWrkNo = 1
EDIT  DELETE  123  INSERT  TEACH
```

(b) Press the [F1] (EDIT) key and specify the workpiece number in the variable “MWrkNo”.
Example) When “2” is specified as the work number.

```
<PROGRAM> B1              Edit
8 MWrkNo = 1

EDIT  DELETE  123  INSERT  TEACH
```

→

```
<PROGRAM> B1              Edit
8 MWrkNo = 2

EDIT  DELETE  123  INSERT  TEACH
```

(c) Press the [EXE] key and the change is determined.

```
<PROGRAM> B1
5 '# COPYRIGHT                 : MIT
6 '##############################
7 '  (1)Input a work number to th
8 MWrkNo = 2
EDIT  DELETE  123  INSERT  TEACH
```

(7) Press the [F1] (FWD) key and execute step feed. “(2) Input a mechanism number ... “is displayed.

```
<PROGRAM> B1
6 '##############################
7 '  (1)Input a work number to th
8 MWrkNo = 2
9 '  (2)Input a mechanism number to th
EDIT  DELETE  123  INSERT  TEACH
```

```
<PROGRAM> B1
7 '  (1)Input a work number to th
8 MWrkNo = 2
9 '  (2)Input a mechanism number to th
10 MMechNo = 2
EDIT  DELETE  123  INSERT  TEACH
```

(8) Press the [F1] (FWD) key and execute step feed. “(3) Input an operating speed ... “is displayed.

```
<PROGRAM> B1
8 MWrkNo = 2
9 '  (2)Input a mechanism number to th
10 MMechNo = 2
11 '  (3)Input an operating speed of
EDIT  DELETE  123  INSERT  TEACH
```

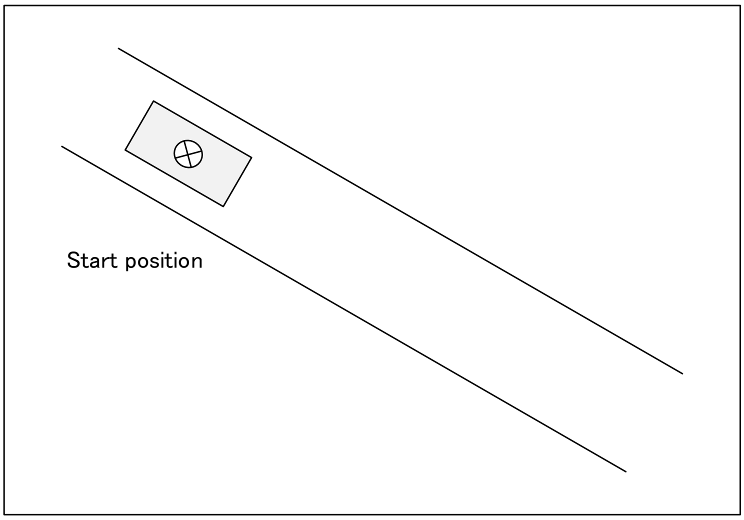
```
<PROGRAM> B1
9 '  (2)Input a mechanism number to th
10 MMechNo = 2
11 '  (3)Input an operating speed of
12 MSpd = 300
EDIT  DELETE  123  INSERT  TEACH
```

(9) Press the [F1] (FWD) key and execute step feed. “(4) Move user mechanisms to ... “is displayed.

```
<PROGRAM> B1
10 MMechNo = 2
11 '  (3)Input an operating speed of
12 MSpd = 300
13 '  (4)Move user mechanisms to the
EDIT  DELETE  123  INSERT  TEACH
```

```
<PROGRAM> B1
11 '  (3)Input an operating speed of
12 MSpd = 300
13 '  (4)Move user mechanisms to the
14 P_102(MWrkNo) = P_Fbc(MMechNo)
EDIT  DELETE  123  INSERT  TEACH
```

Move the user mechanism to the work start position.

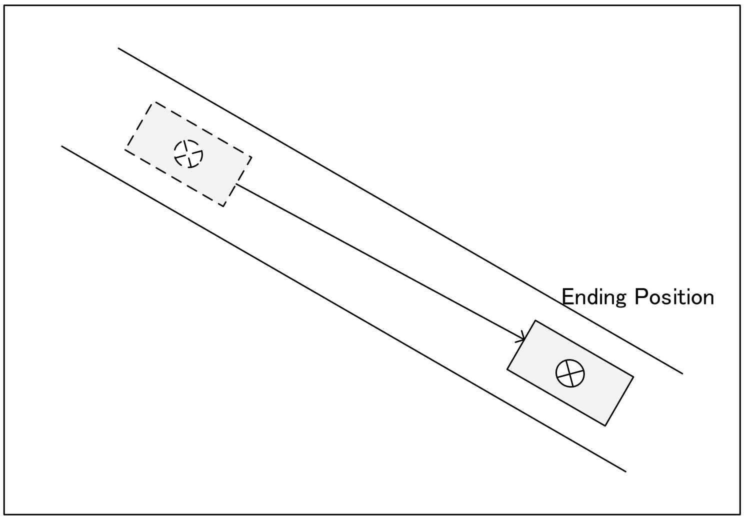


(10) Press the [F1] (FWD) key and execute step feed. “(5) Move user mechanisms to ... “is displayed.

```
<PROGRAM> B1
12 MSpd = 300
13 '  (4)Move user mechanisms to the
14 P_102(MWrkNo) = P_Fbc(MMechNo)
15 '  (5)Move user mechanisms to the
EDIT  DELETE  123  INSERT  TEACH
```

```
<PROGRAM> B1
13 '  (4)Move user mechanisms to the
14 P_102(MWrkNo) = P_Fbc(MMechNo)
15 '  (5)Move user mechanisms to the
16 P_103(MWrkNo) = P_Fbc(MMechNo)
EDIT  DELETE  123  INSERT  TEACH
```

Move the user mechanism to the work end position.



(11) Press the [F1] (FWD) key and execute step feed. “(6) Perform step operation ... “is displayed.

```
<PROGRAM> B1
14 P_102(MWrkNo) = P_Fbc(MMechNo)
15 '  (5)Move user mechanisms to the
16 P_103(MWrkNo) = P_Fbc(MMechNo)
17 ' (6) Perform step operation until
EDIT  DELETE  123  INSERT  TEACH
```

```
<PROGRAM> B1
17 ' (6) Perform step operation until
18 P_109(MWrkNo).X = MMechNo
19 P_109(MWrkNo).Y = MSpd
20 End
EDIT  DELETE  123  INSERT  TEACH
```

6.3.9.2. Confirmation after operation

Check the value of “P_102()”, “P_103()”, “P_109()” using T/B.
Enter the work number in the array element.

- “P_102()”: Start position of the user mechanism
- “P_103()”: End position of the user mechanism
- “P_109()”: Values of variables “MMechaNo” (mechanism number), “MSpd” (speed of the user mechanism)

Confirm that the above values are entered.

6.3.9.3. When multiple user mechanisms are used

Carry out the same operations as above when multiple user mechanisms are used as well, but pay attention to be following points.

Example) When using the mechanism 3, work number 5
(a) Copy the “B1” program, please create an “B2” program.
(b) Please change the work number for variable “MWrkNo” in the “B2” program to “5”.
(c) Please change the mechanism number for variable “MMechNo” in the “B2” program to “3”.

6. 3. 10 Work Base Position Registration (“C1” program)

This chapter explains the tasks carried out by using “C1” program.
“C1” Program performs specified tasks and register the work base coordinate.
The procedures of operations specified by “C1” program and items to be confirmed after the operations are explained below.
Please refer to “Detailed Explanations of Functions and Operations” for the steps involved in each operation.

6.3.10.1. Operation procedure

Using "C1" program, operate in the following procedures.

(1) Set the controller mode to "MANUAL". Set the T/B to "ENABLE".

(2) Press one of the keys (example, [EXE] key) while the <TITLE> screen is displayed. The <MENU> screen will appear.

```
MELFA CR800-D                Ver. A1
RH-3FRH5515-D

COPYRIGHT (C) 2017  MITSUBISHI
ELECTRIC CORPORATION ALL RIGHTS
RESERVED
```

→

```
<MENU>

1.FILE/EDIT          2.RUN
3.PARAM            4.ORIGIN/BRK
5.SET/INIT
                   123        CLOSE
```

(3) Select "1. FILE /EDIT" screen on the <MENU > screen.

```
<MENU>

1.FILE/EDIT          2.RUN
3.PARAM            4.ORIGIN/BRK
5.SET/INIT
                   123        CLOSE
```

→

```
<FILE/EDIT>

A1        17-03-31 17:06:39 12345
B1        17-03-31 17:06:39 12345
C1        17-03-31 17:06:39 12345
EDIT   POSI   123   NEW   COPY
```

(4) Press the arrow key, combine the cursor with the program name "C1" and press the [EXE] key. Display the <program edit> screen.

```
<FILE/EDIT>

A1        17-03-31 17:06:39 12345
B1        17-03-31 17:06:39 12345
C1        17-03-31 17:06:39 12345
EDIT   POSI   123   NEW   COPY
```

→

```
<PROGRAM> C1
1 '## Ver.A1 #####################
2 '# Additional axis tracking - Pr
3 '# Program Type            : C1.
4 '# Date of creation/version : 20
EDIT   POSI   123   NEW   COPY
```

(5) Press the [FUNCTION] key, and change the function display

```
<PROGRAM> C1
1 '## Ver.A1 #####################
2 '# Additional axis tracking - Pr
3 '# Program Type            : C1.
4 '# Date of creation/version : 20
EDIT   POSI   123   NEW   COPY
```

→

```
<PROGRAM> C1
1 '## Ver.A1 #####################
2 '# Additional axis tracking - Pr
3 '# Program Type            : C1.
4 '# Date of creation/version : 20
EDIT   DELETE   123   INSERT   TEACH
```

(6) Press the [F1] (FWD) key and execute step feed. “(1) Input a work number to the … “is displayed. Execute work according to the comment in the robot program.

```
<PROGRAM> C1
4 '# Date of creation/version : 20
5 '# COPYRIGHT                : MIT
6 '###############################
7 '  (1)Input a work number to the
FWD   JUMP   123         BWD
```

Specify the workpiece number.
If you want to change the workpiece number, please edit the program as follows.

(a) Display the following command.

```
<PROGRAM> C1
5 '# COPYRIGHT                    : MIT
6 '################################
7 '  (1)Input a work number to the
8 MWrkNo = 1
EDIT  DELETE  123  INSERT  TEACH
```

(b) Press the [F1](FWD) key and specify the workpiece number in the variable “MWrkNo”
Example) When “2” is specified as the workpiece number.

```
<PROGRAM> C1              Edit
8 MWrkNo = 1

EDIT  DELETE  123  INSERT  TEACH
```

→

```
<PROGRAM> C1              Edit
8 MWrkNo = 2

EDIT  DELETE  123  INSERT  TEACH
```

(c) Press the [F1] (FWD) key and the change is determined.

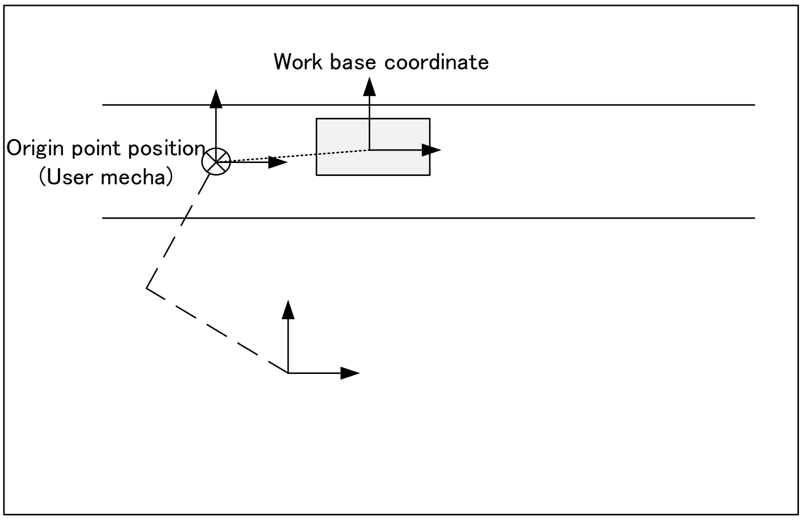
```
<PROGRAM> C1
5 '# COPYRIGHT : MITSUBISHI ELECTRIC
6'################################
7 '(1)Input a workpiece number to the
8 MWrkNo = 2               'Set a wo
EDIT  DELETE  123  INSERT  TEACH
```

(7) Press the [F1] (FWD) key and execute step feed. “(2) Mov the robot to the origin … “is displayed. Execute work according to the comment in the robot program.

```
<PROGRAM> C1
9 For M1=1 To 8
10   P_100(M1) = P_Zero
11 Next M1
12 '  (2) Mov the robot to the origin
FWD  JUMP  123        BWD
```

```
<PROGRAM> C1
10   P_100(M1) = P_Zero
11 Next M1
12 '  (2) Mov the robot to the origin
13 PUMOrg = P_Fbc(1)
FWD  JUMP  123        BWD
```

Move the robot to the origin point position of user mechanism.

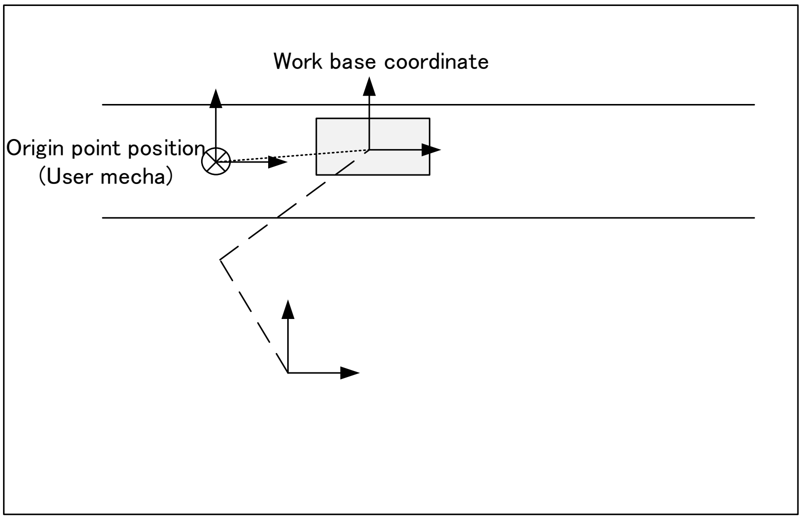


(8) Press the [F1] (FWD) key and execute step feed. “(3)Move the robot to the work … “is displayed. Execute work according to the comment in the robot program.

```
<PROGRAM> C1
11 Next M1
12 ' (2) Mov the robot to the origin
13 PUMOrg = P_Fbc(1)
14 ' (3)Move the robot to the work
FWD  JUMP  123        BWD
```

```
<PROGRAM> C1
12 ' (2) Mov the robot to the origin
13 PUMOrg = P_Fbc(1)
14 ' (3)Move the robot to the work
15 P_100(MWrkNo)=P_Fbc(1)
FWD  JUMP  123        BWD
```

Move the robot to the work base coordinate.



(9) Press the [F1] (FWD) key and execute step feed. “(4)Perform step operation until … “is displayed. Execute work according to the comment in the robot program.

```
<PROGRAM> C1
13 PUMOrg = P_Fbc(1)
14 ' (3)Move the robot to the work
15 P_100(MWrkNo)=P_Fbc(1)
16 ' (4)Perform step operation until
FWD  JUMP  123        BWD
```

```
<PROGRAM> C1
16 ' (4)Perform step operation until
17 P_101(MWrkNo) = Inv(PUMOrg)*P_10
18 M_00# = MwrkNo
19 End
FWD  JUMP  123        BWD
```

6.3.10.2. Confirmation after operation

Confirm the values of “P_100()” and “P_101()” using T/B.
Enter **work numbers** in array elements.

- “P_100()”: Work base coordinate
- “P_101()”: Offset amount of axis to which work coordinate is attached

Check that each of the values above has been entered correctly.

6.3.10.3. When multiple workpieces are used

Carry out the same operations as above when multiple workpieces are used as well, but pay attention to the following points.

Example）When using work number “5”,

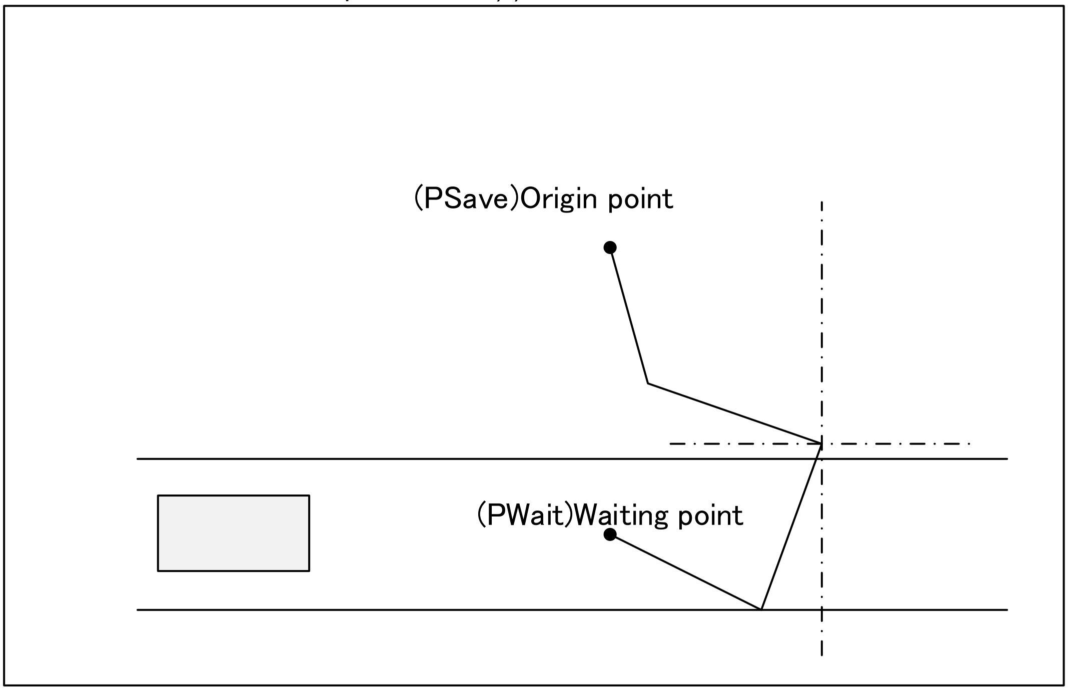
(a) Copy the “C1” program, please create a “C2” program.
(b) Please change the kind number for variable “MWrkNo” in the “C2” program to “5”.

6.3.11 Teaching and Setting of Adjustment Variables (“1” program)

This chapter explains operations required to run “1” program.
In addition, this chapter explains a method to check the operation in the condition that it was designated, and to coordinate again.

6.3.11.1. Teaching

The teaching of “Origin point position (position in which system is started)” and “Waiting point position (position in which it is waited that workpiece arrives)”) is executed.



Teach the origin position, waiting position and transportation point. The following explains how to perform these operations.

1) Open “1” program using T/B.
2) Open the [Position data Edit] screen.
3) Display “PSave” in order to set the robot origin position when the system is started.
4) Move the robot to the origin position and teach it the position.
5) Display “PWait” in order to set the waiting position in which it is waited that workpiece arrives.
6) Move the robot to the waiting position and teach it the position.
7) Display “PSave" at the starting point position on the [Position data Edit] screen. Turn on the servo by gripping the deadman switch.
8) Push [F1] (MOVE) and move the robot to the position of “PSave".

```
<POS> JNT 100% Psave
   X: -100.00  A:+0000.00
   Y: -300.00  B:   +90.00
   Z: +400.00  C:  +180.00
  L1:+0000.00  L2:+0000.00
 FL1:00000007  FL2:00000000
 MOVE  TEACH  123  Prev  Next
```

9) Move the robot to the position of “PWait" pushing F1 (MOVE).

6.3.11.2. Setting of adjustment variables in the program

The following section explains how to set adjustment variables, which are required at transportation, and details about their setting.
Please refer to separate manual "Detailed Explanations of Functions and Operations" for how to set adjustment variables.

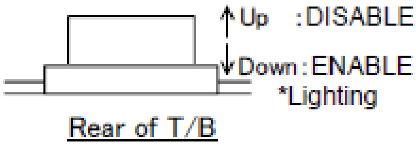
Table 6-1 List of adjustment variables in the program

Variable name	Explanation	Setting Example
PUp1	When the adsorption operation of workpiece, set the offset in the z-axis that the robot works. Offset is the amount of elevation (mm) from the position where workpiece is adsorbed. **[*]Since this variable shows the distance in a tool coordinate system, the sign changes depending on a robot model.**	When you raise the workpiece 50mm from the adsorption position: （Example）RV series: (X,Y,Z,A,B,C)=(+0,+0,-50,+0,+0,+0) （Example）Other than RV series: (X,Y,Z,A,B,C)=(+0,+0,+50,+0,+0,+0)
PDly1	Set the time to keep following work reference position	When tracking time is set to 1 second (X,Y,Z,A,B,C)= (+1,+0,+0,+0,+0,+0)
PPri	"1" program and "CM1" program are run simultaneously (multitasking). "1" program moves the robot, and "CM1" program observes the sensor. It is possible to specify which program is processed with a higher priority, rather than performing the same amount of processing at the same time. X = Set the line numbers of "1" program to be performed (1 to 31). Y = Set the line numbers of "CM1" program to be performed (1 to 31).	When you set to run "1" program by one line and run "CM1" program by 10 lines: (X,Y,Z,A,B,C)=(+1,+10,+0,+0,+0,+0)
POffset	When the adsorption position shifts, the gap can be corrected. Set the correction value. **[*]The direction of the correction is a direction of the hand coordinate system. Please decide the correction value after changing the job mode to "Tool", pushing the [+X] key and the [+Y] key, and confirming the operation of the robot.**	When the deviation to +X direction in hand-coordinate system is 2mm, and deviation to -Y direction in hand-coordinate system is 1mm: (X,Y,Z,A,B,C)=(+2,-1,+0,+0,+0,+0)

6.3.11.3. Automatic Operation

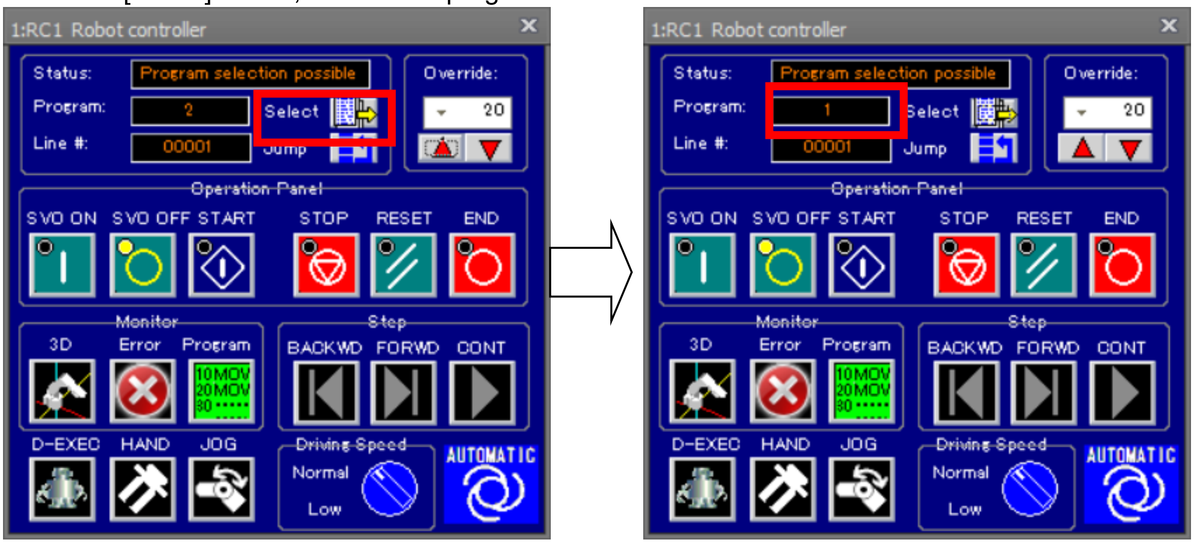
This chapter explains how to prepare the robot before starting the system.

1) Confirm that there isn't an intervention thing in the robot movement area.
2) Set the T/B [ENABLE] switch to "DISABLE"



3) Set the controller mode to "AUTOMATIC".
4) Use operation panel in RT ToolBox 3, and specify the override to 20% - 30%.

5) Press the [Select] button, and select "program 1"



6） Automatic operation will start when the [START] button is pressed.

*Prepare for the unexpected operation of the robot, please can press anytime emergency stop switch of T/B.

7) Confirm to be a work that is moved to waiting point position after following the workpiece.

8) If you check the operation, press the [STOP] button and stop the robot.

POINT

T / B software in a specific version or later, you can be the automatic operation from T / B.

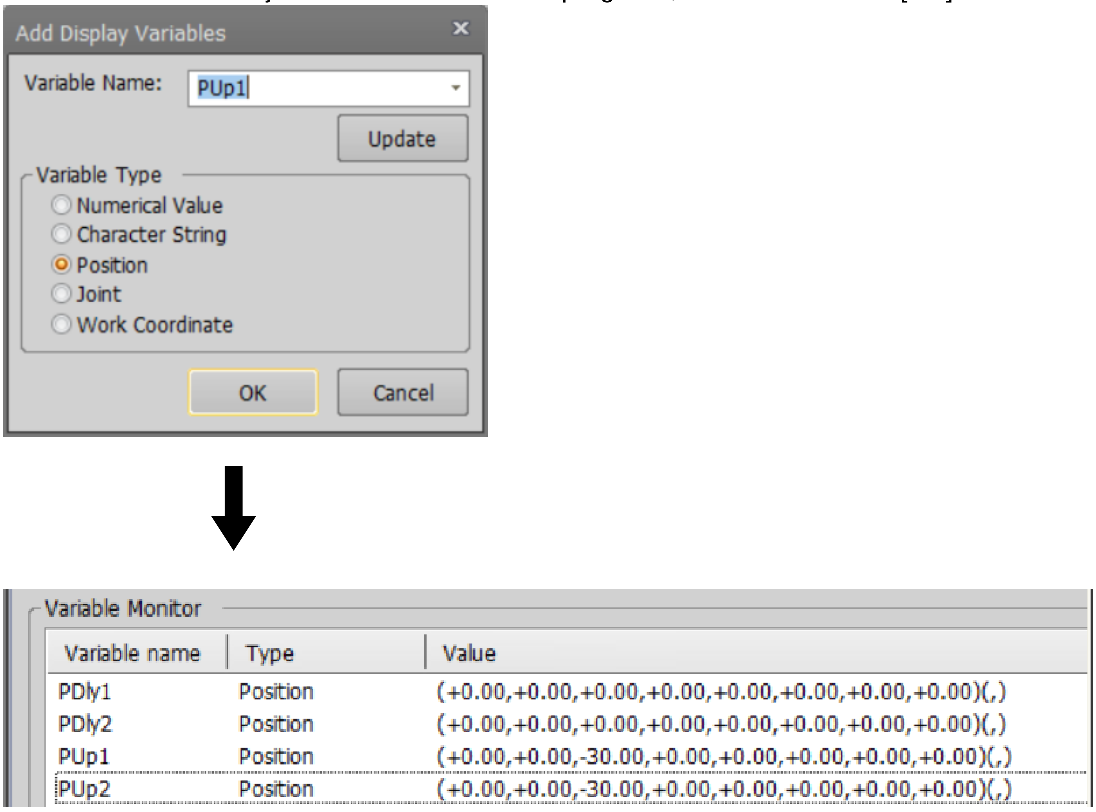
With R32/33T/B software version 1.7 or later, the program’s automatic operation can be started from the T/B (With R56/57TB, version 3.0 or later). Please refer to “Detailed Explanations of Functions and Operations” for operation procedures and details.

6.3.11.4. Adjustment of operating conditions

In automatic operation, if you want to adjust the vertical movement and adsorption time of the robot arm that was described in "6.3.11.2 Setting of adjustment variables in the program" should be changed in the following procedure.

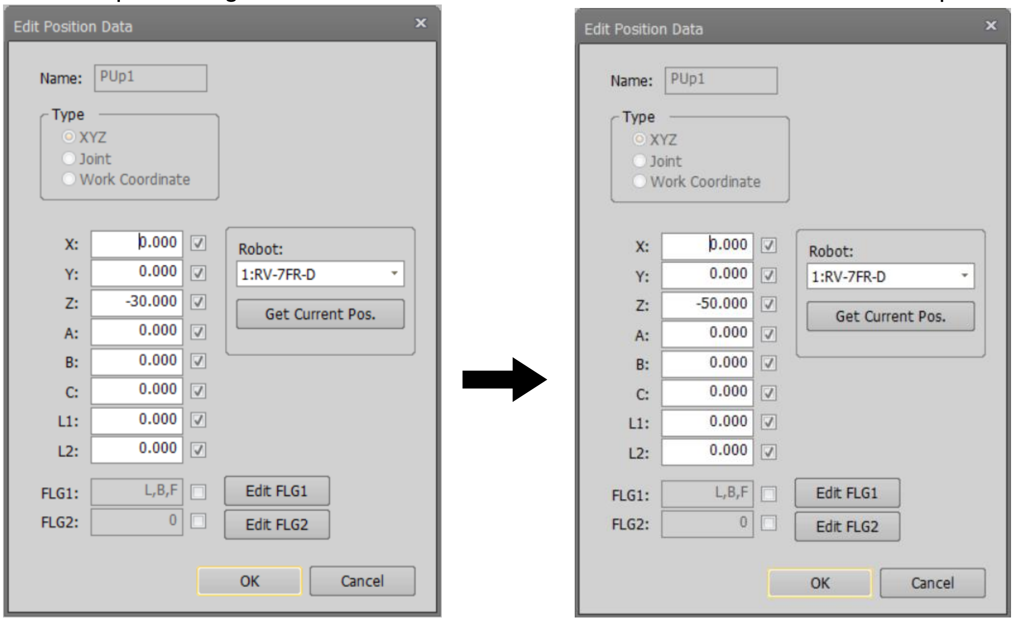
(1) Start the "Program monitor" of RT ToolBox3.

(2) Click the [Add] button and open the "Add display variables" screen. Enter the variables listed in the "Table 6-1 List of adjustment variables in the program", and then click the [OK] button.

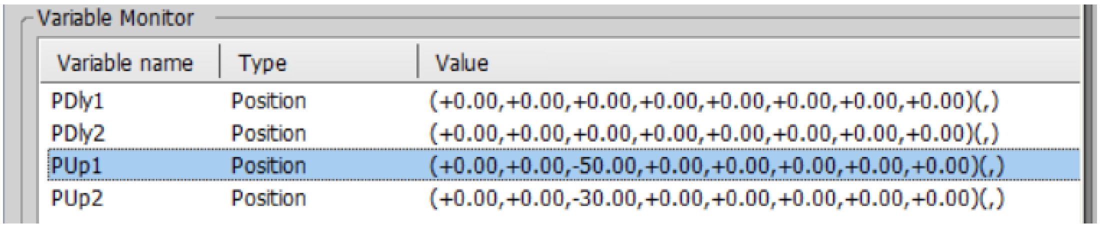


(3) Double-click the variable you want to change, and change the appropriate value for displayed in the "Edit Position data".

For example, change to "-50" from "-30" the value of the Z-coordinate of the PUp1 :



(4) Click [OK] button, and confirm that was able to change the value of the variable that is specified in the "Variable Monitor".



(5) Return to the "6.3.11.3 Automatic Operation", and then check to see whether the can be corrected by implementing the automatic operation.

6.4 Troubleshooting

In this chapter, we explain cause and measure when an error occurred.

Error number related to coordinated control for additional axes is below.

■■■■ □□ ◆ ◇◇

■■■■ : Function number
0□ : Additional axis tracking
1□ : Base coordinate corporate control
◆ : Mechanism number
◇◇ : Servo axis number

Table 6-15 The list of error related to coordinated control for additional axes.

Error number	Error cause and measures	
L2660	One of the following errors was detected. Please take appropriate measures corresponding to the error message.	
	Error message	Error concerning axis coop.
	Cause	The error concerning the axis coop occurred.
	Masures	Please Confirm the content by a detail number of the error.
	Error message	Illegal robot No(axis tracking).
	Cause	The designated robot No. is illegal.
	Masures	Set the correct robot No.
	Error message	Illegal robot No(base coop).
	Cause	The designated robot No. is illegal.
	Masures	Set the correct robot No.
	Error message	Illegal axis No(axis tracking).
	Cause	The designated axis No. is illegal.
	Masures	Set the correct axis No.
	Error message	Illegal axis No(base coop).
	Cause	The designated axis No. is illegal.
	Masures	Set the correct axis No.
	Error message	Illegal unit system (axis tracking).
	Cause	A rotary axis is designated
	Masures	Please designate a linear drive axis.
	Error message	Illegal unit system (base coop).
	Cause	A rotary axis is designated
	Masures	Please designate a linear drive axis.
	Error message	Origin unsetting (axis tracking).
	Cause	Axis tracking cannot be executed because of origin unsetting.
	Masures	Please set the origin and turn the power OFF and ON.
	Error message	Origin unsetting (base coop).
	Cause	Axis tracking cannot be executed because of origin unsetting.
	Masures	Please set the origin and turn the power OFF and ON.
L2661	One of the following errors was detected. Please take appropriate measures corresponding to the error message.	
	Error message	Error concerning axis coop (combi.).
	Cause	The function cannot be used at the same time with the axis coop.
	Masures	Please confirm the content by a detailed number of the error.
	Error message	Cannot be used (base coop).
	Cause	Tracking function is effective.
	Masures	Invalidate the tracking function.
	Error message	Cannot be used (axis tracking).
	Cause	A synchronous addition axis control is effective.
	Masures	Invalidate a synchronous addition axis control.

Error number	Error cause and measures	
	Error message	Cannot be used (base coop).
	Cause	A synchronous addition axis control is effective.
	Masures	Invalidate a synchronous addition axis control.
	Error message	Jrc cannot be executed.
	Cause	The base coop is executing.
	Masures	Please do not use Jrc command.
	Error message	Interpolation cannot be executed.
	Cause	An addition axis is going to move.
	Masures	Please do not move an addition axis.
	Error message	Cannot be used (base coop).
	Cause	Interference avoidance function is effective.
	Masures	Invalidate the interference avoidance function.
L2662	Error message	Work setting cannot be changed.
	Cause	The work coop is executing.
	Masures	Please change after the axis tracking is invalid.
H2663 *	Error message	Origin data was changed
	Cause	Origin data was changed about the base cooperation target axis
	Masures	Turn the power OFF and ON once.

(*1) An error with * is an error requiring a power reset.

7. Appendix

7.1 Display of option card information

You can display the option card information in RT ToolBox3 option. When you click “Online”->”Board”->"Slotn (n = 1 to 2): MELFA Smart Plus" of the tree on the workspace in online state, you can read MELFA Smart Plus card information in the property window.

*** Option card information in property window is not updated automatically. If you want to update, change it to offline once and change it to online, then perform the above operation again.**

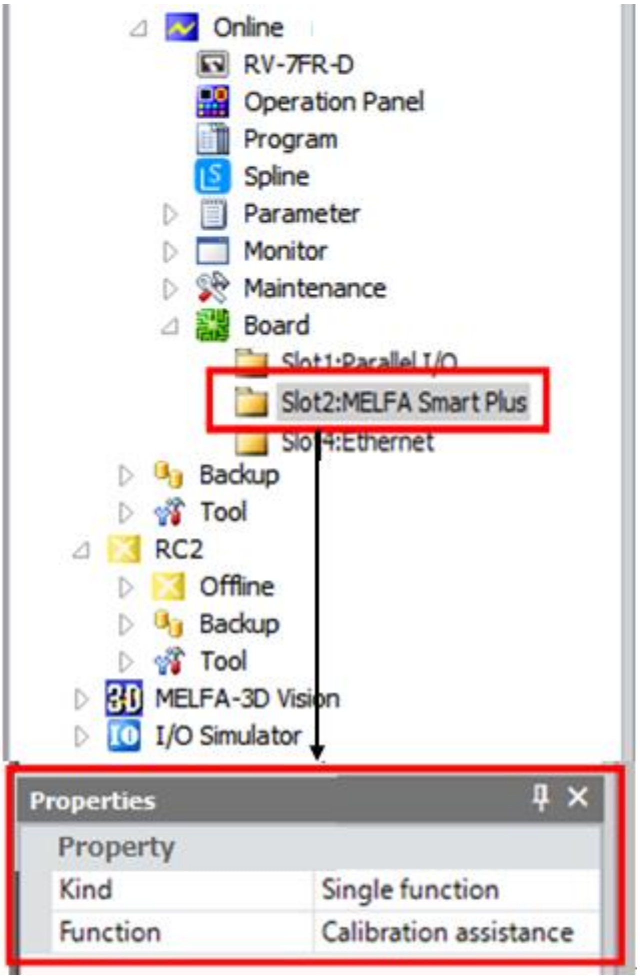


Figure 7-1 Example of display of option card information in RT ToolBox3

Table 7-1 Card information of MELFA Smart Plus card/card pack

Display item		Display example	Meaning	Software version of controller	Remarks
Card name		MELFA Smart Plus	Card name	Ver. A1 or later	
		MELFA Smart Plus A-type		Ver. A3 or later	
		MELFA Smart Plus B-type		Ver. A3 or later	
		MELFA Smart Plus AB-type		Ver. A3 or later	
Card information	[Kind]	Single function	Card type of MELFA Smart Plus.	Ver. A1 or later	
		Multi function		Ver. A1 or later	
	[Function]	Calibration assistance	The function name set by parameter of “SMART+1”.	Ver. A1 or later	In the case of MELFA Smart Plus card pack, it is blank.
		Robot temperature compensation		Ver. A1 or later	
		Coordinated control for additional axes		Ver. A1 or later	
		Preventive Maintenance		Ver. A3 or later	
		Extended function of MELFA-3D Vision		Ver. A3 or later	

MITSUBISHI ELECTRIC CORPORATION

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

Authorised representative:
Mitsubishi Electric Europe B.V. FA - European Business Group
Mitsubishi-Electric-Platz 1, D-40882 Ratingen, Germany
Tel: +49(0)2102-4860

Sep., 2018 MEE Printed in Japan on recycled paper. Specifications are subject to change without notice.