MITSUBISHI

Mitsubishi Industrial Robot

FQ/FD/SQ/SD Series

Mitsubishi Robot Tool for EasyBuilder Instruction Manual

MELFA

BFP-A8820-C

⚠ Safety Precautions

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

⚠ CAUTION

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

⚠ CAUTION

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

⚠ WARNING

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

⚠ CAUTION

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

⚠ WARNING

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

⚠ CAUTION

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

⚠ CAUTION

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

⚠ CAUTION

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

The points of the precautions given in the separate "Safety Manual" are given below.
Refer to the actual "Safety Manual" for details.

⚠WARNING When automatically operating the robot with multiple control devices (GOT, PLC, pushbutton switch), the interlocks, such as each device's operation rights must be designed by the user.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

⚠CAUTION

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

⚠CAUTION

When using this product’s GOT direct connection function, do not connect a handy GOT. The handy GOT can automatically operate the robot regardless of whether the operation rights are enabled or not, so use could result in property damage or personal injuries.

⚠CAUTION

When using an iQ Platform compatible product with CRnQ, do not connect the handy GOT. The handy GOT can automatically operate the robot regardless of whether the operation rights are enabled or not, so use could result in property damage or personal injuries.

⚠CAUTION

When the SSCNETIII cable is removed, install the cap in the connector.
If the cap is not installed, there is a possibility of malfunctioning by adhesion of the dust etc.

⚠CAUTION

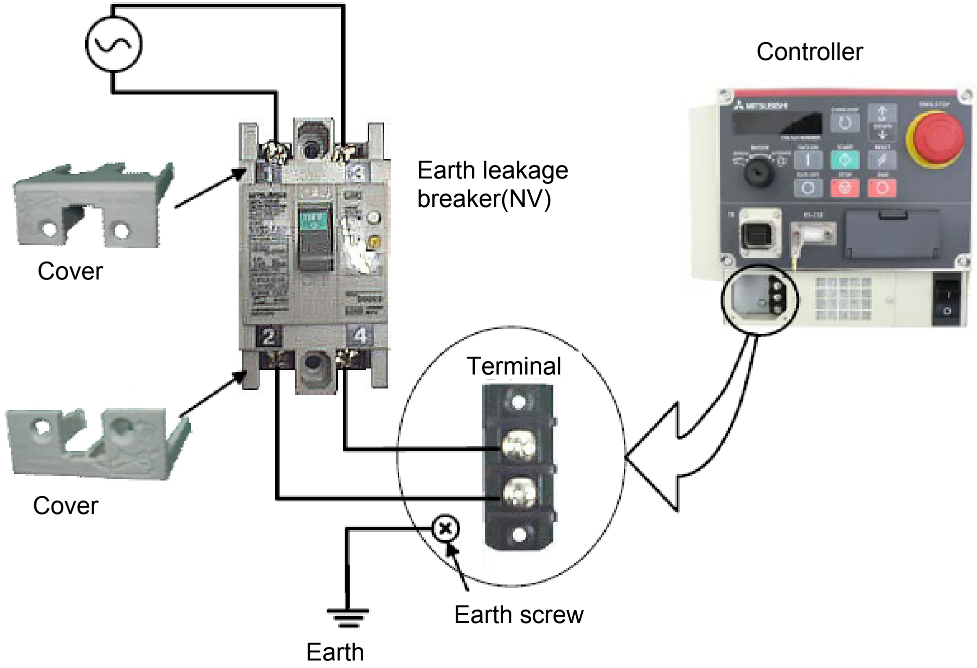
Don't remove the SSCNETIII cable, when the power supply of the robot controller is turned on. Don't face squarely the light emitted from the tip of the SSCNETIII connector or the cable. If light strikes the eyes, there is a possibility of feeling the sense of incongruity for the eyes. (The light source of SSCNETIII is equivalent to the class 1 specified to JISC6802 and IEC60825-1.)

⚠CAUTION

Take care not to wire the units incorrectly. Connections which do not satisfy the specifications could result in malfunction such as emergency stop not being released.
When completed with the wiring, confirm that each function properly operates without malfunction, including emergency stop with the robot controller operation panel, emergency stop with the teaching box, user’s emergency stop, and each door switch, etc.

Precautions for the basic configuration are shown below.(When CR1D-7xx/CR1Q-7xx is used for the controller.)

⚠CAUTION Please install the earth leakage breaker in the primary side supply power supply of the controller because of leakage protection



■ Revision History

Printing Date	Instruction Manual No.	Revision Contents
2010-08-06	BFP-A8820-*	First Edition
2011-11-17	BFP-A8820-A	Corresponded to In-sight Explorer 4.5/4.6
2013-03-14	BFP-A8820-B	Corresponded to In-sight Explorer 4.8
2014-04-01	BFP-A8820-C	Corresponded to In-sight Explorer 4.9

■ Preface

Thank you for downloading the Mitsubishi robot tool for EasyBuilder.
This software supports the calibration function needed when the vision sencer made by Cognex Corporation is connected with the Mitsubishi robot. This software can be used by adding to (add-on) the EasyBuilder.

Before using this sensor, please read this manual well so that you utilize the contents of this manual when using this network vision sensor.
This manual attempts to cover special handling as well. Please interpret the absence of an operation from this manual as meaning that it can not be done.

The contents of this manual target the following software versions.

SQ series :	Version R1 or later
SD series :	Version S1 or later

* It is necessary to upgrade (for a fee) if you used an earlier version. Please ask to the bought shop.

■ Symbols & Notation Method in This Manual

⚠ ***DANGER***	This indicates a situation in which a mistake in handling will expose the user to the danger of death or severe injury.
⚠ ***WARNING***	This indicates a situation in which a mistake in handling has the possibility of resulting in death or severe injury of the user.
⚠ ***CAUTION***	This indicates a situation in which a mistake in handling has the danger of causing injury to the user. Equipment damage is also possible.

- No part of this manual may be reproduced by any means or in any form,without prior consent from Mitsubishi.
- The details of this manual are subject to change without notice.
- An effort has been made to make full descriptions in this manual.However,if any discrepancies or unclear points are found,please contact your dealer.
- The information contained in this document has been written to be accurate as much as possible.Please interpret that items not described in this document “cannot be perfomed.”.
  Please contact your nearest dealer if you find any doubtful,wrong or skipped point.
- Microsoft, Windows, and .NET Framework are registered trademarks of the Microsoft Corporation of the United States in the United States and/or other countries.
- In-Sight is registered trademark of Cognex Corporation.
- Adobe,the Adobe logo,Acrobat,and the Acrobat logo are trademarks of Adobe Systems incorporated.
- Reference to registered trademarks and trademarks are omitted in this manual.

Contents

1. Summary

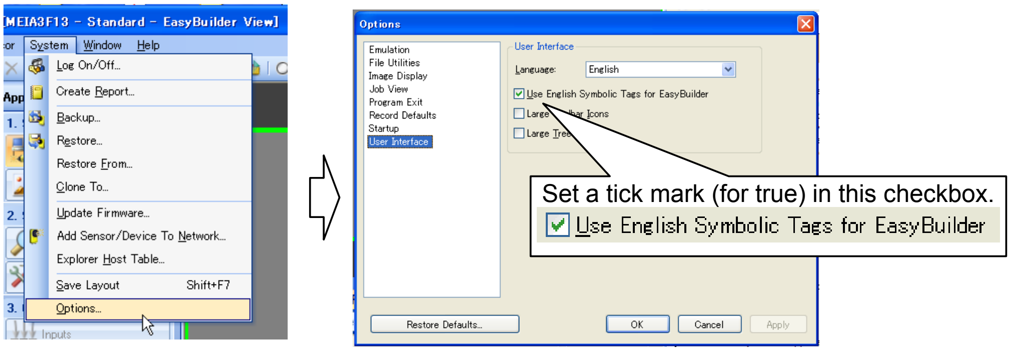
This manual is an Instruction Manual to use EasyBuiler made by Cognex Corporation with the MELFA robot.

This manual explains about a calibration function for the Mitsubishi robot that adds to the EasyBuilder. Therefore, this does not explain a basic operation and the screen about the EasyBuilder or the products other than our company. Please refer to the manual and help of the product if necessary.

⚠ Caution

Please use "English symbolic tag" in the EasyBuilder.

Please set optional information on EasyBuilder according to the following procedures to improve the compatibility between the robot controller and the vision sensor. As a result, it is possible to acquire the information from the vision sensor by using the value of robot parameter "EBRDTAG".



Set a tick mark (for true) in this checkbox.

☑ Use English Symbolic Tags for EasyBuilder

"EasyBuilder menu -> system -> Option -> User interface "

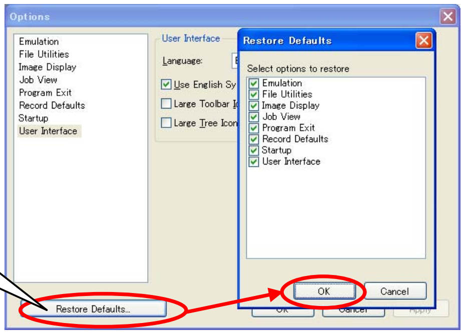
⚠ Caution

If the error occurs when the value is changed, please execute "Restore Defaults".

When this error occurred, please set a tick mark in the checkbox "Use English Symbolic Tags for EasyBuilder" after executing "Restore Defaults".

Click "Restore Defaults" button.

Restore Defaults...



1.1. Difference from MELFA-Vision

The difference between MELFA-Vision and this software is as follows.

Model	MELFA-Vision	Mitsubishi robot tool	Notes
In-Sight EZ model	×	○	The name and the state of the cell cannot be confirmed because the spread sheet cannot be displayed in the EZ model. It is not possible to use it with MELFA-Vision because there is the one for which the specification of the cell name is necessary in a special instruction.
In-Sight micro series	○	○	
In-Sight 5000 series	○	○	

Table 1-1 Feature of MELFA-Vision and correspondence method in the EasyBuilder.

No.	function	MELFA-Vision	EasyBuilder
1	Logon / Logoff	It corresponds by an exclusive screen.	It corresponds by In-Sight Explorer
2	Image operation		
	Capture request	It corresponds by an exclusive screen.	It corresponds by In-Sight Explorer
	Capture request	It corresponds by an exclusive screen.	It corresponds by In-Sight Explorer
3	Online / Offline	It corresponds by an exclusive screen.	It corresponds by In-Sight Explorer
4	Vision program writing	This registers frequently used image processing (pattern matching, blob and color) as templates.	Simple image recognition processing can easily add various positioning tools and the inspection tools with GUI.
5	Getting the recognition result	Information on the cell is acquired by using a special command of MELFA BASIC.	The recognition result is stored to one cell by the custom format character string. And receives it by the MELFA BASIC command which specifies the symbolic tag.
6	Recognition result display	It corresponds by an exclusive screen.	It corresponds by In-Sight Explorer
7	Setting the communication method of robot controller and the vision sensor	It is easy to make settings for communicating between the robot controller and vision sensor.	It is possible to set the communication method of the vision sensor by In-Sight Explorer, and to set up the communication method of the robot controller by RT ToolBox2.
8	Robot and vision sensor calibration	It corresponds by an exclusive screen.	It corresponds by Mitsubishi calibrationtool.
9	Image Log	It corresponds by an exclusive screen.	It corresponds by In-Sight Explorer
10	File transfer		
	Backup	It corresponds by an exclusive screen.	It corresponds by In-Sight Explorer
	Restore	It corresponds by an exclusive screen.	It corresponds by In-Sight Explorer
	Cloning	It corresponds by an exclusive screen.	It corresponds by In-Sight Explorer

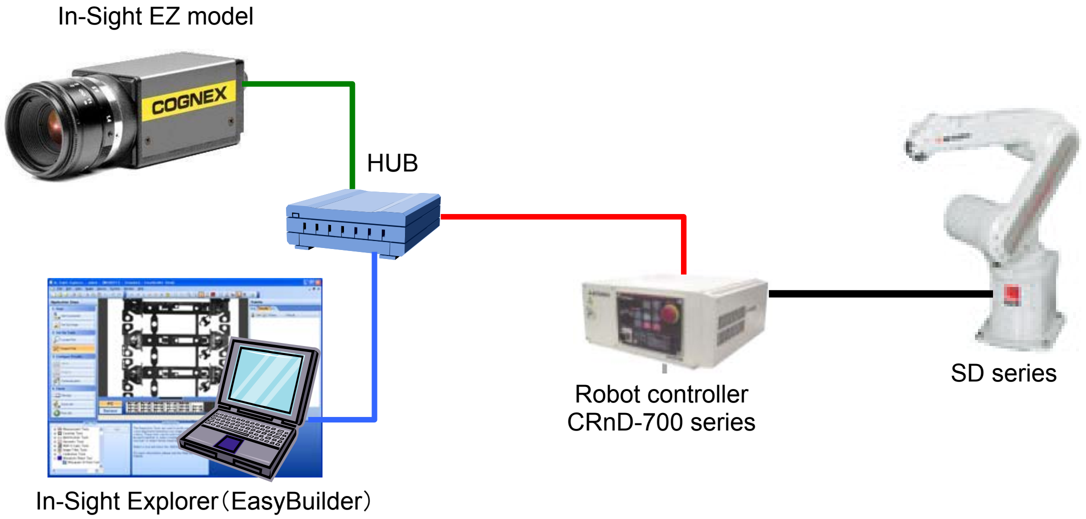
2. Component Devices

This chapter explains the equipment composition of using the In-Sight EZ model and EasyBuilder.

2.1. Example of connecting each controller

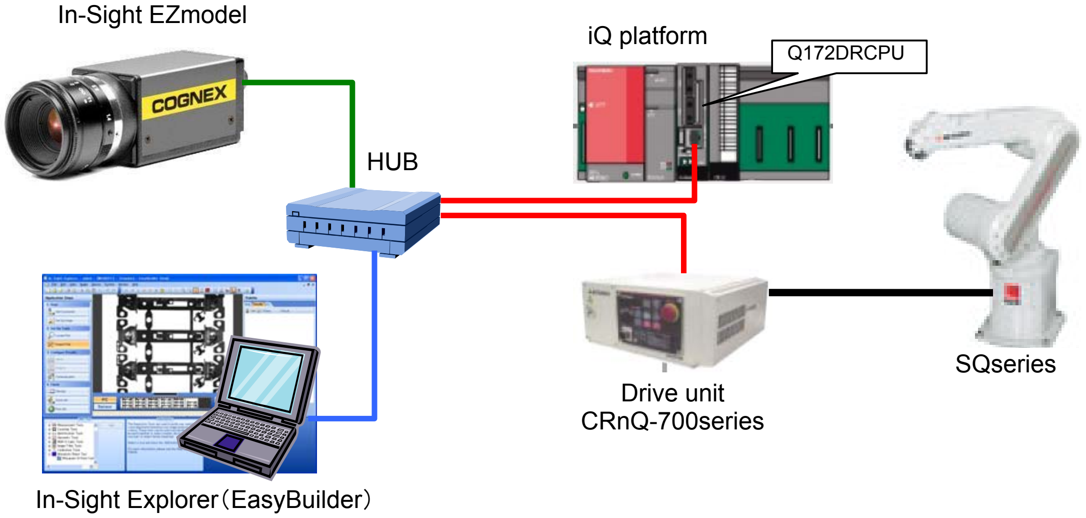
2.1.1. SD series

The robot controller and the EZ model are connected with the Ethernet hub. Please connect the personal computer with the Ethernet hub when you edit the jobs (Image processing programs) by using In-Sight Explorer (EasyBuilder).



2.1.2. SQ series

The robot CPU unit(Q172DRCPU) and the EZ model are connected with the Ethernet hub. Please connect the personal computer with the Ethernet hub when you edit the jobs (Image processing programs) by using In-Sight Explorer (EasyBuilder).



2.2. Robot controller

Item	Specification
Robot controller	CRnQ-7xx / CRnD-7xx series. < Software version > CRnQ-7xx series : version R1 or later CRnD-7xx series : Version S1 or later * It is necessary to upgrade (for a fee) if you used an earlier version. Please ask to the bought shop.
Connectable robots	All robots
Robot program language	The special command (EBRead) of MELFA BASIC V is used. (This command can be used in version R1/S1 or later.)
Parameter	The special parameter (EBRDTAG) of MELFA BASIC V is used. (This parameter can be used in version R1/S1 or later.)

* - Please refer to **"7. Detailed Explanation of Functions"** for the special command and the special parameter.

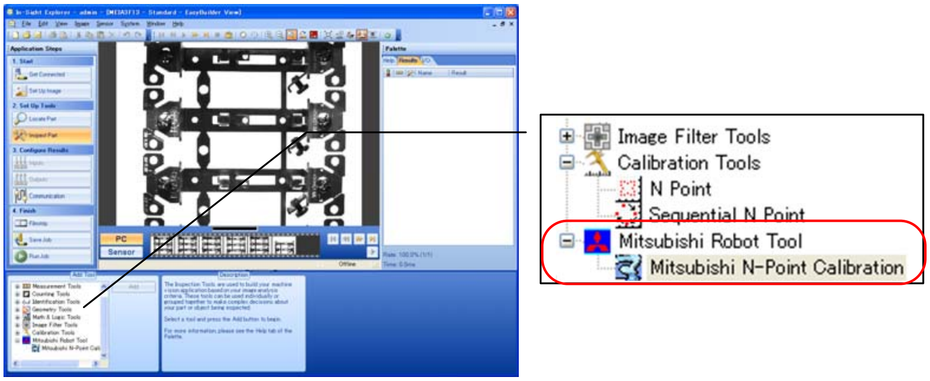
2.3. Vision sensors

Item	Specification
Software	In-Sight Explorer : version 4.4.1 (7526) or later * If you were using the In-Sight Explorer Version 4.4.0 or earlier, please download Ver4.4.1 from Cognex Corporation HP. * The tool that can be used by the version of In-Sight Explorer is different. Please use an appropriate tool after confirming the version of your In-Sight Explorer.
Vision sensors	In-Sight series all models : Firmware Ver4.04.00(155) or later In-Sight EZ series In-Sight 5000 series In-Sight Micro series

2.4. Mitsubishi robot tool (add-on for EasyBuilder)

EasyBuilder is one of the mode of expressions in In-Sight Explorer, and it is showing of the spread sheet by the GUI form. In the Ez model, it is possible to use only EasyBuiler. The Mitsubishi robot tool is a tool only for a Mitsubishi robot that is able to add on the calibration function to EasyBuilder.

An add-on function is displayed on EasyBuilder as shown in next figure.



3. List of add-on functions

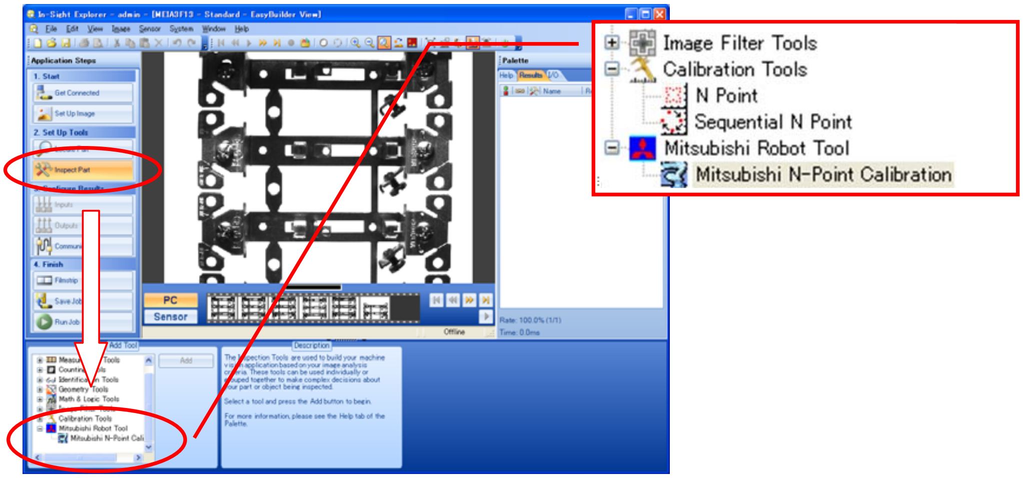
The add-on function is as follows.

No.	Function	Explanation
1	Mitsubishi N-Point Calibration	The calibration data is made by using some points (from 2 to 16) that become a pair on on the screen and the robot coordinates.

3.1. Mitsubishi N-Point Calibration

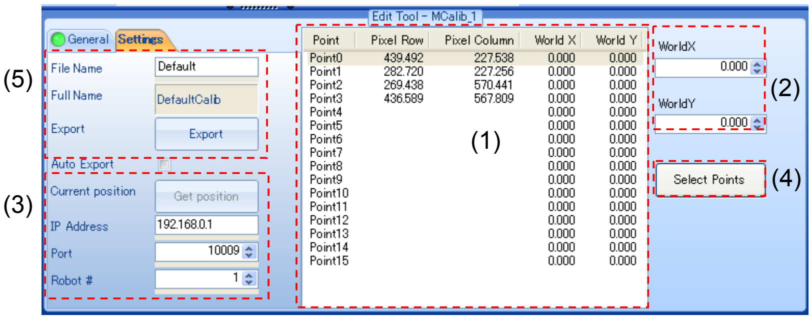
Mitsubishi N-Point Calibration make the calibration data by using some points (from 2 to 16) that become a pair on on the screen and the robot coordinates. By using this tool, it is possible to get directly the current position (X,Y) by way of Ethernet.

When "Mitsubishi N point calibration" is installed, it is registered partially of the tool in EasyBuilder as shown in the figure below.



3.1.1. Explanation of screen

This chapter explains the operation methods of Mitsubishi N-point calibration screen.



File Name: Default
Full Name: DefaultCalib
Export
Auto Export □
Current position
IP Address: 192.168.0.1
Port: 10009
Robot #: 1

Point	Pixel Row	Pixel Column	World X	World Y
Point0	439.492	227.538	0.000	0.000
Point1	282.720	227.256	0.000	0.000
Point2	269.438	570.441	0.000	0.000
Point3	436.589	567.809	0.000	0.000
Point4			0.000	0.000
Point5			0.000	0.000
Point6			0.000	0.000
Point7			0.000	0.000
Point8			0.000	0.000
Point9			0.000	0.000
Point10			0.000	0.000
Point11			0.000	0.000
Point12			0.000	0.000
Point13			0.000	0.000
Point14			0.000	0.000
Point15			0.000	0.000

WorldX: 0.000
WorldY: 0.000

(1) Selects the target points.

Selects two or more feature at the user definition point and the recognition position, etc, when Mitsubishi N-point calibration is added. The camera coordinates (pixel line/row) and the robot coordinates (world X/Y) corresponding to the point that the feature indicates are set to this list. In the initial state, the current location of the feature is set to the camera coordinates, and 0 is set to the robot coordinates. In the calibration work, the robot coordinates corresponding to the camera coordinates are set step by step.

(2) The Robot coordinates set to a target point.

The robot coordinates (world coordinates X/Y) corresponding to the point that has been selected in the list (1) are displayed. This value can be edited directly. Moreover, it can be increased and decreased with spin button. And, the input value is reflected in the list immediately. It is also possible to take the location data from the robot directly.

(3) Getting the current position of the robot.

The value sent from the robot is set to (2) when clicking on "Get position" button after inputting Internet Protocol address, the communication port and the robot number. The value is reflected in the point selected by the list (1).

(4) Re-selecting a target point.

The feature registered in the list (1) can be selected by clicking again.

(5) Exporting the calibration result.

The information set to the list (1) is preserved in the file when clicking "Export" button after setting the file name. As a result, it is possible to use the calibration result by reading it from other jobs.

4. Installation

It is necessary to install the add-on software to use the Mitsubishi robot tool in EasyBuilder.

4.1. List of the file that relates to Mitsubishi robot tool

The file that relates to the Mitsubishi robot tool is as follows.

No.	File name	Explanation	Installation folder
1	MelfaClassLibrary.dll	The Mitsubishi robot tool is added to the item tree of the EasyBuilder.	After ISE 4.9 C:/Program Files/Cognex/In-Sight/ In-Sight Explorer *.*.*¥ *AddIns*¥ After ISE 4.8 C:/Program Files/Cognex/In-Sight/ In-Sight Explorer *.*.*¥*bin*¥ Before ISE 4.7.3 C:/Program Files/Cognex/In-Sight/ In-Sight Explorer *.*.*¥ * Change this folder, when the installation destination of In-Sight Explorer has been changed.
2	Cognex.ezaddin	Enable this tool.	After ISE 4.9 C:/Program Files/Cognex/In-Sight/ In-Sight Explorer *.*.*¥ *bin*¥
3	MELFACALIB_jp.cxd	The snippet that achieves Mitsubishi N-point calibration function	C:/Program Files/Cognex/In-Sight/ In-Sight Explorer 4.5.0/Snippets/EasyBuilder * “Specified folder”+”/Snippets/EasyBuilder”
4	MELFACALIB_en.cxd		
5	BFP-A8820.pdf	Instruction manual	This manual. This is not installed.

4.2. How to install

There are the following two kinds of installation methods.

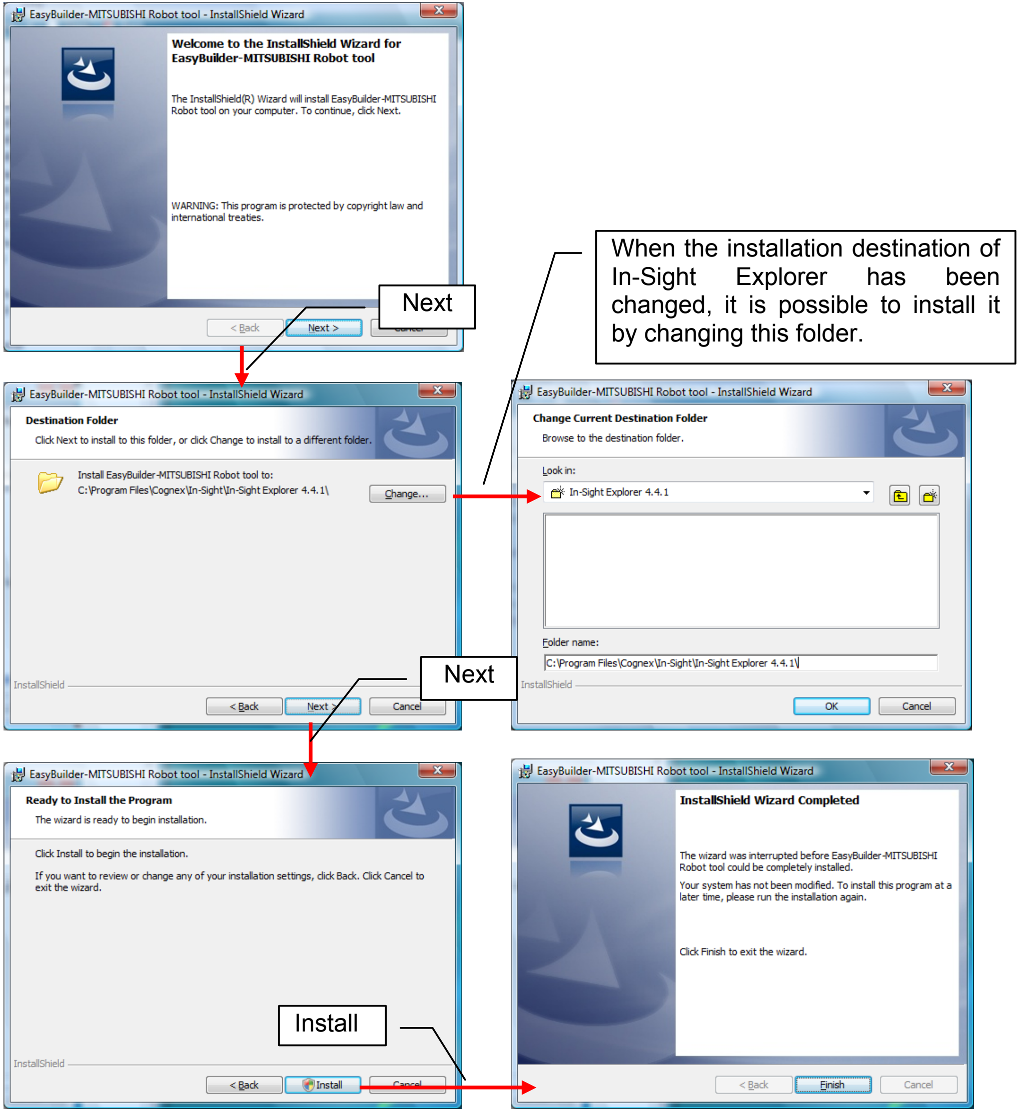
Automated install	Manual installation
The installer copies the file onto a prescribed folder. **<Merit>** It is possible to install it easily. **<Demerit>** It is not possible to install it in In-sight Explorer of two or more versions at the same time. For instance, it is necessary to uninstall it to install it in 4.5.1 when installing it in In-sight Explorer4.5.0 once.	The relation file in Chapter 4.1 is copied onto a prescribed folder. **<Merit>** It is possible to install it in In-sight Explorer of two or more versions at the same time. **<Demerit>** It is a little difficult.

Please install with the following procedure.

(1) Obtain the installer.

(Please inquire the obtaining method of the purchase shop.)

(2) Execute the installer. (Automated install)



EasyBuilder-MITSUBISHI Robot tool - InstallShield Wizard

Welcome to the InstallShield Wizard for EasyBuilder-MITSUBISHI Robot tool

The InstallShield(R) Wizard will install EasyBuilder-MITSUBISHI Robot tool on your computer. To continue, click Next.

WARNING: This program is protected by copyright law and international treaties.

< Back | Next > | Cancel

EasyBuilder-MITSUBISHI Robot tool - InstallShield Wizard

Destination Folder

Click Next to install to this folder, or click Change to install to a different folder.

Install EasyBuilder-MITSUBISHI Robot tool to:
C:\Program Files\Cognex\In-Sight\In-Sight Explorer 4.4.1\

Change...

InstallShield

< Back | Next > | Cancel

When the installation destination of In-Sight Explorer has been changed, it is possible to install it by changing this folder.

EasyBuilder-MITSUBISHI Robot tool - InstallShield Wizard

Change Current Destination Folder

Browse to the destination folder.

Look in: In-Sight Explorer 4.4.1

Folder name: C:\Program Files\Cognex\In-Sight\In-Sight Explorer 4.4.1\

InstallShield

OK | Cancel

EasyBuilder-MITSUBISHI Robot tool - InstallShield Wizard

Ready to Install the Program

The wizard is ready to begin installation.

Click Install to begin the installation.

If you want to review or change any of your installation settings, click Back. Click Cancel to exit the wizard.

InstallShield

< Back | Install | Cancel

EasyBuilder-MITSUBISHI Robot tool - InstallShield Wizard

InstallShield Wizard Completed

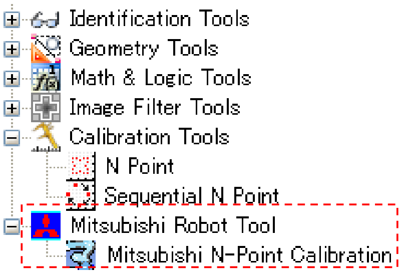
The wizard was interrupted before EasyBuilder-MITSUBISHI Robot tool could be completely installed.

Your system has not been modified. To install this program at a later time, please run the installation again.

Click Finish to exit the wizard.

< Back | Finish | Cancel

(3) Please start In-Sight Explorer, and confirm the result of the installation.
When succeeding in the installation, "Mitsubishi Robot Tool" and "Mitsubishi N-point calibration" are added to "Add tool" tree in EasyBuilder.



- Identification Tools
- Geometry Tools
- Math & Logic Tools
- Image Filter Tools
- Calibration Tools
  - N Point
  - Sequential N Point
- Mitsubishi Robot Tool
  - Mitsubishi N-Point Calibration

4.3. Common settings

The connection of the robot controller and the vision sensor is necessary to use the Mitsubishi robot tool. Please set up the information necessary for the connection according to the following procedures after installing it.

4.3.1. Communications Settings of the robot controller

It is necessary to set up the communication method of Ethernet for communicating between the robot and the vision sensor. Set up the following parameter by using “RT ToolBox2”.

(1) Set up the communication method for the calibration.

In the Mitsubishi robot tool, port 10009 is used to communicate with the robot controller. When port 10009 has already been used,it is possible to use other ports by changing the corresponding parameter.

No.	Parameter name	Value	Details explanation
1	NETIP	xxx.xxx.xxx.xxx	IP address of robot controller
2	NETTERM(9)	1 (Default value : 0)	The end code is added with communicating by Ethernet.
3	CTERME19	1 (Default value : 0)	The End code of port 10009 is changed to "CR+LF".
4	NETPORT(10)	10009 (Default value : 10009)	Port number allocated to device OPT19
5	CPRCE19	0 (Default value : 0)	The protocol used is "Non-procedure"
6	NETMODE(9)	1 (Default value : 1)	Opens as "Server".
7	NETHSTP(9)	unused	

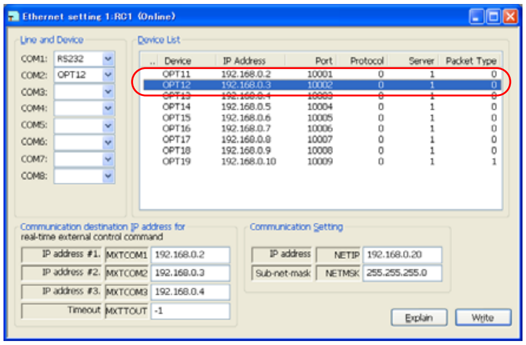
* No.1, 2, and 3 in above list should be changed from an initial value. No.4, 5, and 6 should be same as the above mentioned.

(2) Set up the communication method for receiving the recognition result.

The setting to receive the recognition result is the same as MELFA-Vision. Please set up the following parameters in "Ethernet setting" screen of the parameter. In the following examples, device "OPT12" is set to the communication line "COM2".

If the tick mark is set in the "Change the Parameter to connect_Vision" checkbox in the "Device Setting" screen, the communication method except Internet Protocol address of the vision sensor are set automatically. Please input Internet Protocol address of the vision sensor manually.

* Internet Protocol address of the vision sensor is set with In-Sight Explorer.



Ethernet setting 1:RC1 (Online)

Line and Device

COM1: RS232
COM2: OPT12
COM3:
COM4:
COM5:
COM6:
COM7:
COM8:

Device List

..	Device	IP Address	Port	Protocol	Server	Packet Type
	OPT11	192.168.0.2	10001	0	1	0
	OPT12	192.168.0.3	10002	0	1	0
	OPT13	192.168.0.4	10003	0	1	0
	OPT14	192.168.0.5	10004	0	1	0
	OPT15	192.168.0.6	10005	0	1	0
	OPT16	192.168.0.7	10006	0	1	0
	OPT17	192.168.0.8	10007	0	1	0
	OPT18	192.168.0.9	10008	0	1	0
	OPT19	192.168.0.10	10009	0	1	1

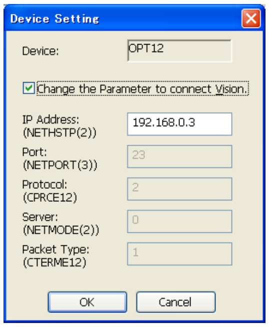
Communication destination IP address for real-time external control command

IP address #1.	MXTCOM1	192.168.0.2
IP address #2.	MXTCOM2	192.168.0.3
IP address #3.	MXTCOM3	192.168.0.4
Timeout	MXTTOUT	-1

Communication Setting

IP address	NETIP	192.168.0.20
Sub-net-mask	NETMSK	255.255.255.0

Double click



Device Setting

Device: OPT12

☑ Change the Parameter to connect Vision.

IP Address: (NETHSTP(2)) 192.168.0.3
Port: (NETPORT(3)) 23
Protocol: (CPRCE12) 2
Server: (NETMODE(2)) 0
Packet Type: (CTERME12) 1

No.	Parameter name	Value	Details explanation
1	NETIP	xxx.xxx.xxx.xxx	IP address of robot controller
2	NETTERM(9)	1 (Default value : 0)	The end code is added with communicating by Ethernet.
3	CTERME12	1 (Default value : 0)	Changing the end code of Port 10009 to "CR+LF".
4	NETPORT(3)	23 (Default value :10003)	Port number allocated to device OPT12.
5	CPRCE12	2(Default value :0)	The protocol used is "data link".
6	NETMODE(2)	0(Default value :1)	Opens as "Client".
7	NETHSTP(2)	xxx.xxx.xxx.xxx	Internet Protocol address of vision sensor to receive result

* No.3, 4, 5, 6 and 7 in above list should be changed from an initial value.

4.3.2. Communications Settings for the vision sencer

Please set up the Internet Protocol address of the vision sensor according to the following procedures.

(1) Starts “In-Sight Explorer **(Ver 4.4.1(7526) or later)”**.
(2) Selects menu “System” -> “Add Sensor /Device to Network”.
(3) Clicks "Copy PC Network Settings" button after selecting the icon of vision sensor. Afterwards, set up the **Internet Protocol address** of the vision sensor.
Clicks "Connect" button after selecting the target camera in "Get Connected" tab

The firmware needs **4.04.00 (155)** or later .

5. The usage

This chapter explains the operational procedure using the EasyBuilder and the Mitsubishi robot tool.　　A general procedure when workpiece is positioned by the vision sensor is as follows.

Adds the teaching position in image
↓
Adds "Mitsubisi N-Point Calibration"
↓
Gets the current position of the robot
↓
Exports the calibration data
↓
Imports the calibration data
↓
Sets the output format of robot
↓
Executes the recognition and gets the recognition results

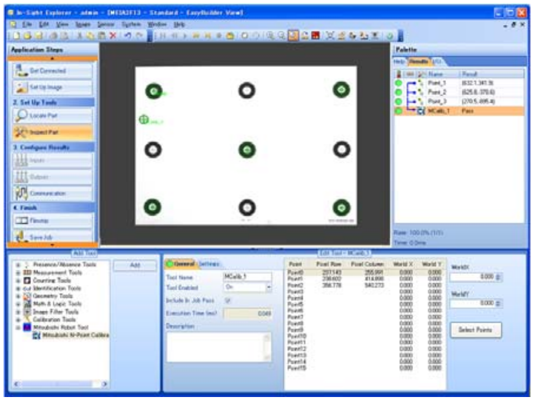
5.1. Adds the teaching position in image

It is necessary to define the target points in the image to do the calibration.

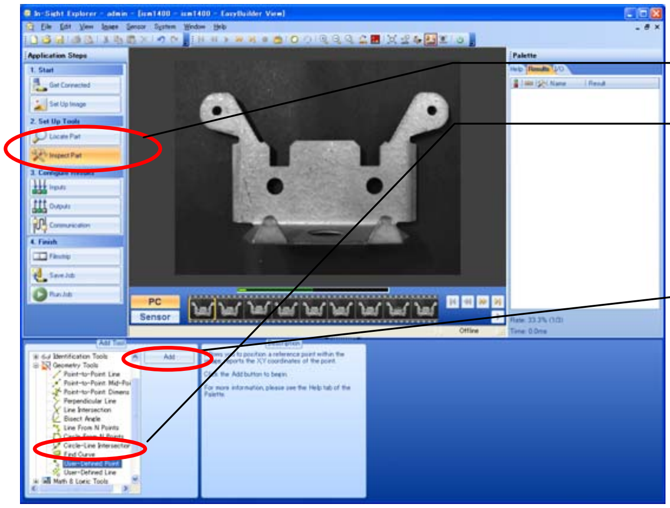
It is possible to use the abundant positioning and the inspection tools prepared in EasyBuilder for the method of defining the point. The preparation method is as follows.

- "User definition point" : Specifies an arbitrary point on the screen.
- "Pattern matching" : Positions the registered template.
- Center of a circle and straight line intersection,
- etc.

There is a calibration seat in the appendix. Please use it. The figure below is an example of taking the calibration seat. When the calibration is added, a round object on the screen is automatically recognized because the EasyBuilder has the smart feature function. Therefore, the calibration work only of the selection of the point on the screen can begin.



Moreover, this manual explains the case of using the user definition position that can be generally used.



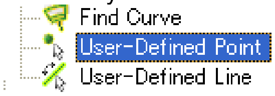
Allows you to position a reference point within the image. Reports the X,Y coordinates of the point.

Click the Add button to begin.

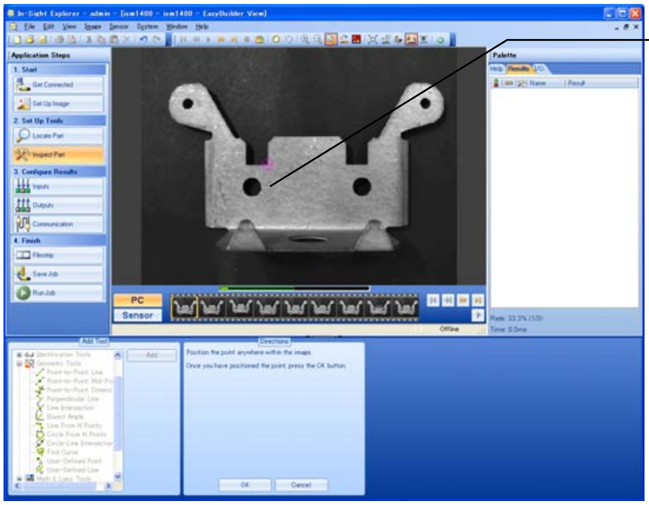
For more information, please see the Help tab of the Palette.

(1) Selects “Inspect Part”.

(2) Selects “User-Defined Point”in “Geometry Tools”.



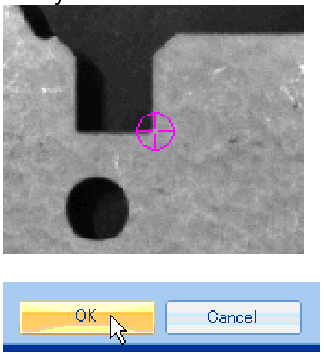
(3) Clicks “Add” button.

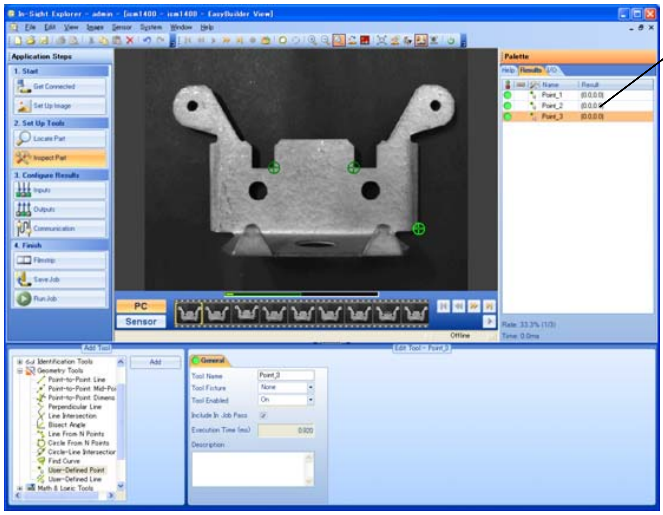


Position the point anywhere within the image.

Once you have positioned the point, press the OK button.

(4) click "OK" button, after setting the cursor to the position to be taught easily.





General

Tool Name: Point_3

Tool Fixture: None

Tool Enabled: On

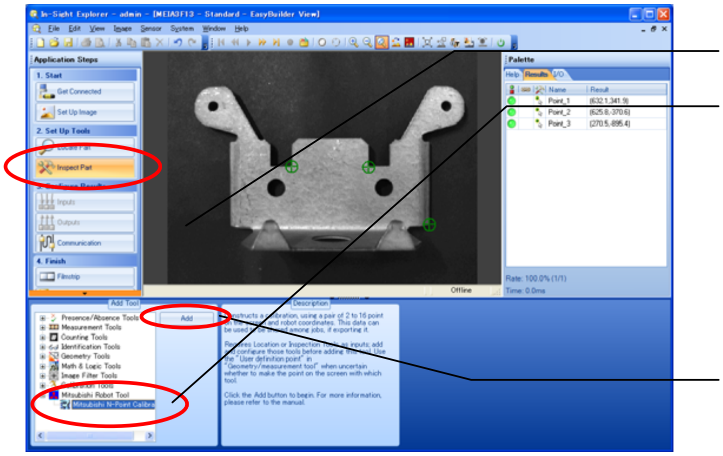
Include In Job Pass: ☑

Execution Time (ms): 0.920

Description

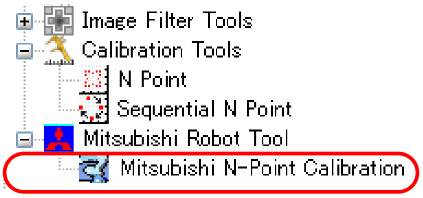
(5) Please teach in **two positions or more** according to a similar procedure.
(It is possible to calculate of the calibration by two positions. However, to improve accuracy, two or more positions are recommended to be taught on the entire screen.)

5.2. Adds “Mitsubisi N-Point Calibration”

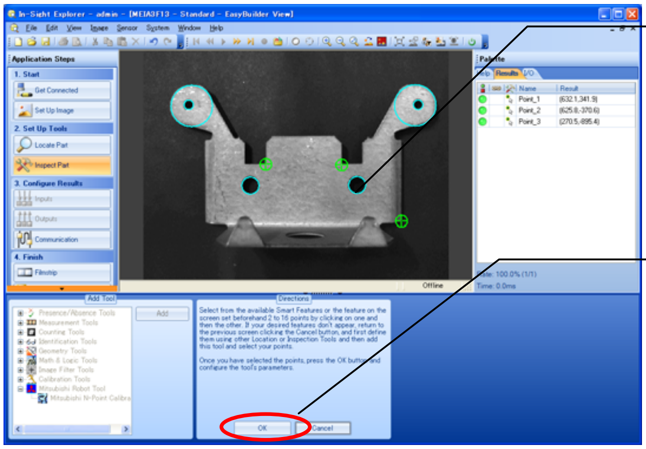


(1) Selects “Inspect Part”.

(2) Selects “Mitsubishi N-Point Calibration”.

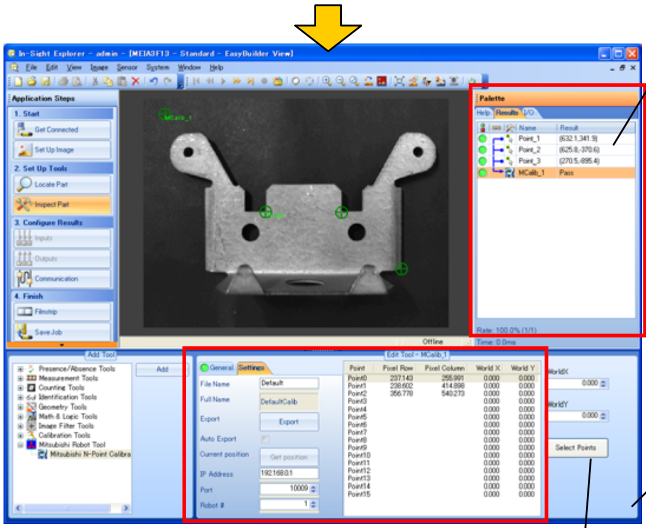


(3) Clicks “Add” button.



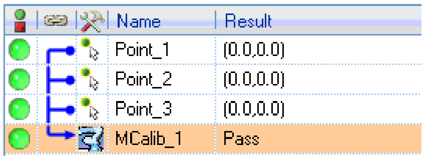
(4) Please select **two positions or more** in the vision sensor coordinates to use for calibration. The color of selected item is changed. It is possible that the point selected here will be

(5) The calibration is added to the palette when "OK" button is clicked, and the calibration screen is displayed.



Palette

The relation between the tool of the calibration etc. and the recognition points is shown.



			Name	Result
●			Point_1	(0.0,0.0)
●			Point_2	(0.0,0.0)
●			Point_3	(0.0,0.0)
●			MCalib_1	Pass

* Note that the calibration is deleted when the point is deleted after adding the calibration.

Edit Tool

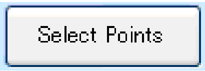
Edits the details of the tool of the calibration etc.

For "Mitsubishi N point calibration", the following information can be edited.

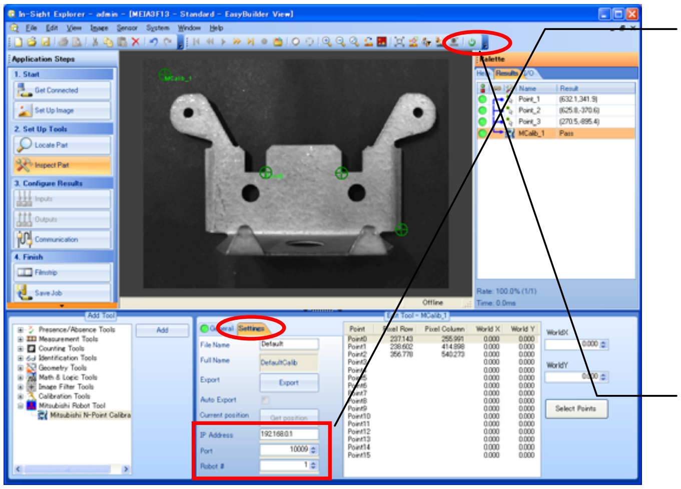
- The communication setting of the robot controller.
- The coordinates of the camera.
- The coordinates of the robot.

Re-select the points

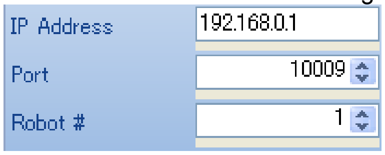
Clicks "Select Points" button in the calibration screen to re-select the points used.



5.3. Gets the current position of the robot



(1) Sets the following communication method of the robot in a "Settings" tab.

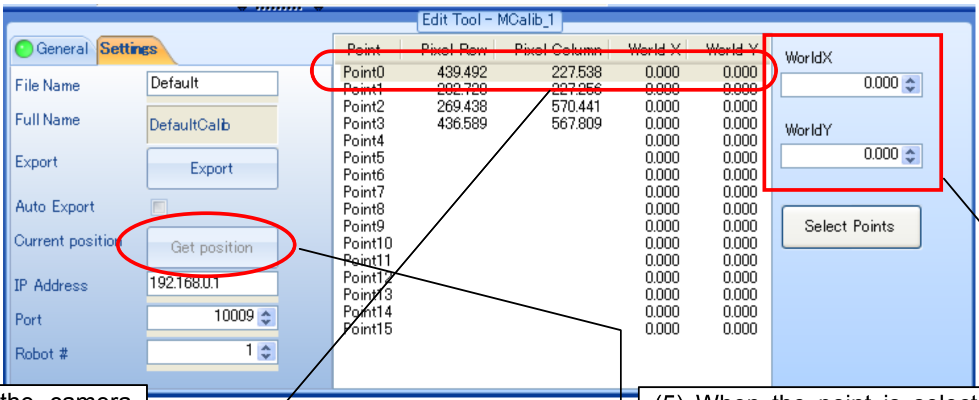


IP Address	192.168.0.1
Port	10009
Robot #	1

* The address of the robot set in Chapter 2.1.2 is set to Internet Protocol address. It is possible to use the initial value to the communication port and the robot number.

(2) Switches the In-Sight Explorer to "Online".

(Switching of online/offline)



Edit Tool - MCalib_1

General Settings

File Name	Default
Full Name	DefaultCalib
Export	Export
Auto Export	□
Current position	Get position
IP Address	192.168.0.1
Port	10009
Robot #	1

Point	Pixel Row	Pixel Column	World X	World Y
Point0	439.492	227.538	0.000	0.000
Point1	282.720	227.256	0.000	0.000
Point2	269.438	570.441	0.000	0.000
Point3	436.589	567.809	0.000	0.000
Point4			0.000	0.000
Point5			0.000	0.000
Point6			0.000	0.000
Point7			0.000	0.000
Point8			0.000	0.000
Point9			0.000	0.000
Point10			0.000	0.000
Point11			0.000	0.000
Point12			0.000	0.000
Point13			0.000	0.000
Point14			0.000	0.000
Point15			0.000	0.000

WorldX	0.000
WorldY	0.000

Select Points

(3) Selects the camera coordinates related to the robot

(4) When "Get position" button is clicked after selecting the point in the list and moving the robot to the position, the current position of the robot is set into "World X/World Y". Getting the current position of the robot is possible only <u>at online mode</u>.

(5) When the point is selected (is clicked) in the list, the current position of the robot is displayed here. It is possible to change the value by clicking the spin button, by getting the current position of the robot or by inputting directly.

------ POINT ------

The calibration is possible by two points. However, It is possible to improve the recognition accuracy by selecting many points in the distance.

⚠ Caution

Please switch to online mode, when getting the current position of the robot.

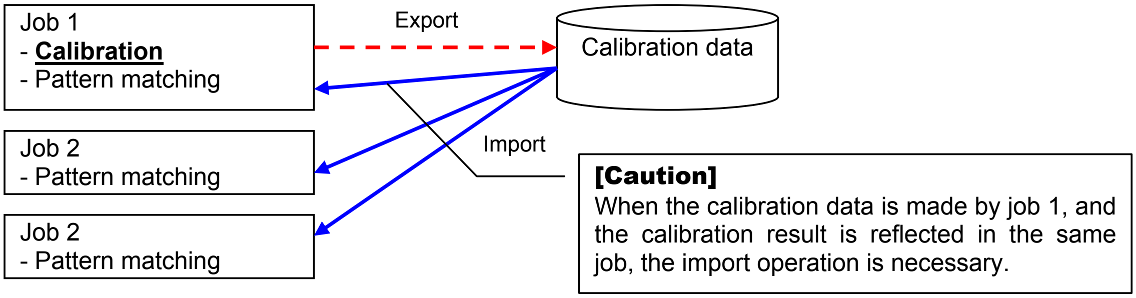
Please switch In-Sight Explorer to online mode to get the current position of the robot.

<u>If this were operated in an off-line mode, the robot current location acquisition operation is repeated when switching to an online mode. In that case, please get the current position of the robot again.</u>

5.4. Exports / Imports the calibration data

The result of the calibration is valid by importing it after exporting to a file. Please note, it is not valid only by adding the calibration.

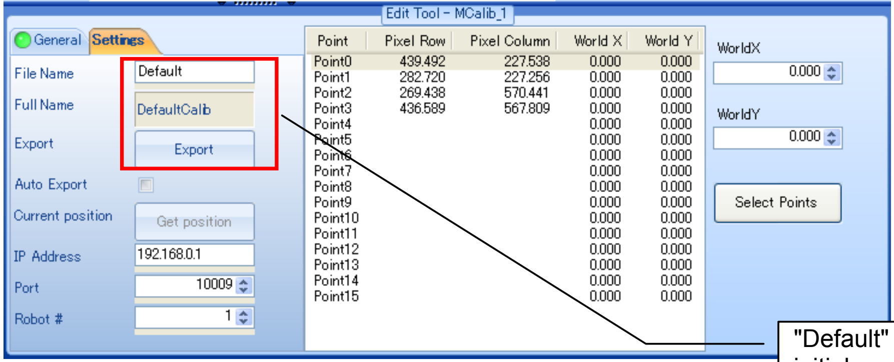
When the position of the robot coordinate system and the vision coordinate system is not different, the calibration data can be recycled.



[Caution]
When the calibration data is made by job 1, and the calibration result is reflected in the same job, the import operation is necessary.

5.4.1. Exports the calibration data

To export the calibration data, click "Export" button after specifying the file name in the calibration screen.



Edit Tool - MCalib_1

General | Settings

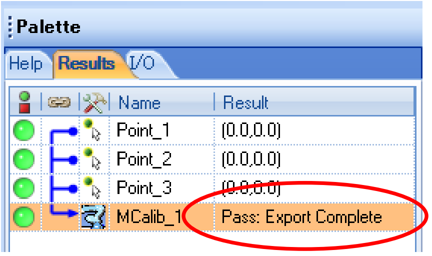
File Name: Default
Full Name: DefaultCalib
Export: Export
Auto Export: □
Current position: Get position
IP Address: 192.168.0.1
Port: 10009
Robot #: 1

Point	Pixel Row	Pixel Column	World X	World Y
Point0	439.492	227.538	0.000	0.000
Point1	282.720	227.256	0.000	0.000
Point2	269.438	570.441	0.000	0.000
Point3	436.589	567.809	0.000	0.000
Point4			0.000	0.000
Point5			0.000	0.000
Point6			0.000	0.000
Point7			0.000	0.000
Point8			0.000	0.000
Point9			0.000	0.000
Point10			0.000	0.000
Point11			0.000	0.000
Point12			0.000	0.000
Point13			0.000	0.000
Point14			0.000	0.000
Point15			0.000	0.000

WorldX: 0.000
WorldY: 0.000
Select Points

"Default" is set to the file name as the initial value. Please change to the name that you can understand the calibration situation.
(It is possible to use the initial name.)
The exported file is preserved in the vision sensor.

The information is displayed in the palette after the export is completed normally.



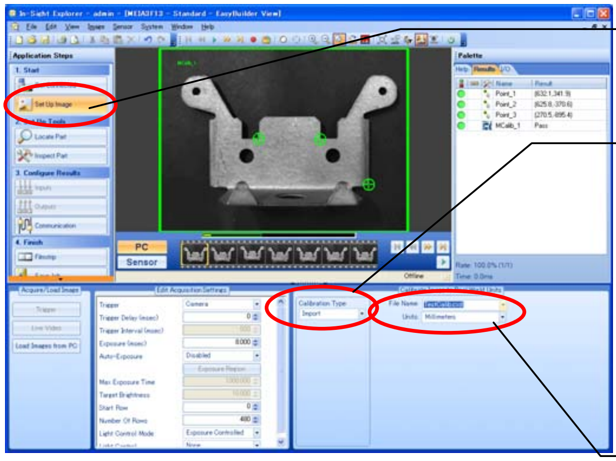
Palette

Help | Results | I/O

Name	Result
Point_1	(0.0,0.0)
Point_2	(0.0,0.0)
Point_3	(0.0,0.0)
MCalib_1	Pass: Export Complete

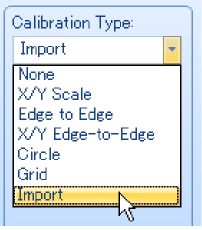
5.4.2. Imports the calibration data

Data on the vision sensor is used by pixel units. Therefore, it is necessary to convert these to data on the robot coordinate system. Please get the calibration data by the following procedure when you make a new job.



(1) Click “Set Up Image” button.

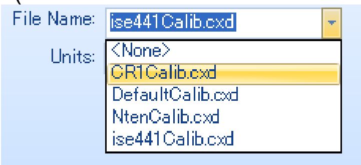
(2) Select “Import.



Calibration Type: Import

- None
- X/Y Scale
- Edge to Edge
- X/Y Edge-to-Edge
- Circle
- Grid
- Import

(3) Select calibration file.
(Selects the millimeter as a unit.)



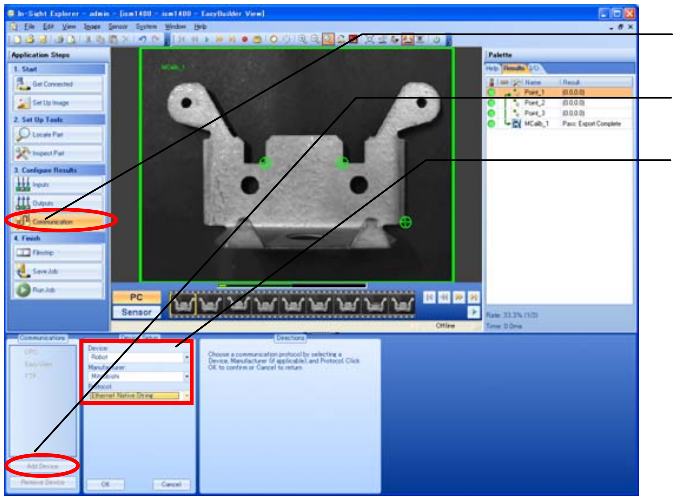
File Name: ise441Calib.cxd
Units:

- <None>
- CR1Calib.cxd
- DefaultCalib.cxd
- NtenCalib.cxd
- ise441Calib.cxd

The file name is as follows.
Export name + “Calib.cxd”

5.5. Sets the output format of robot

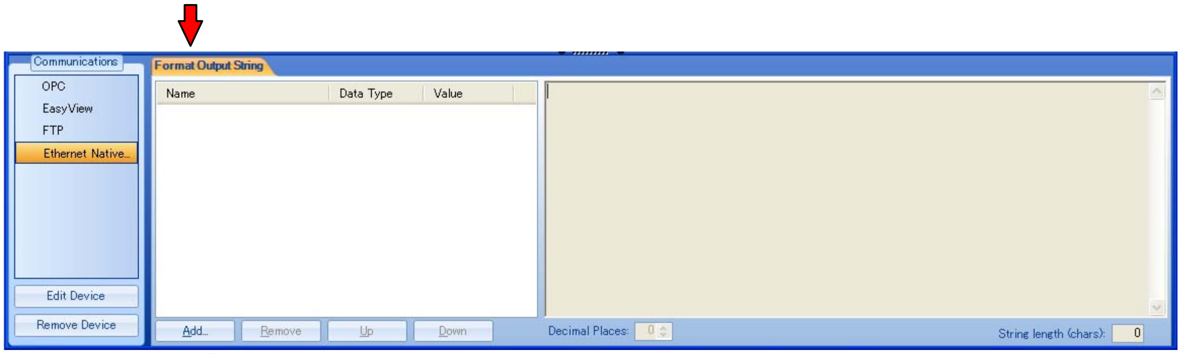
It is necessary to set the output format in order to output the recognition result to the robot controller. When clicking the add button of the output format character strings in the following figure, the output data list that can be added is displayed. Please select output data from this list and add it. The data that can be selected by the status of the job is different.

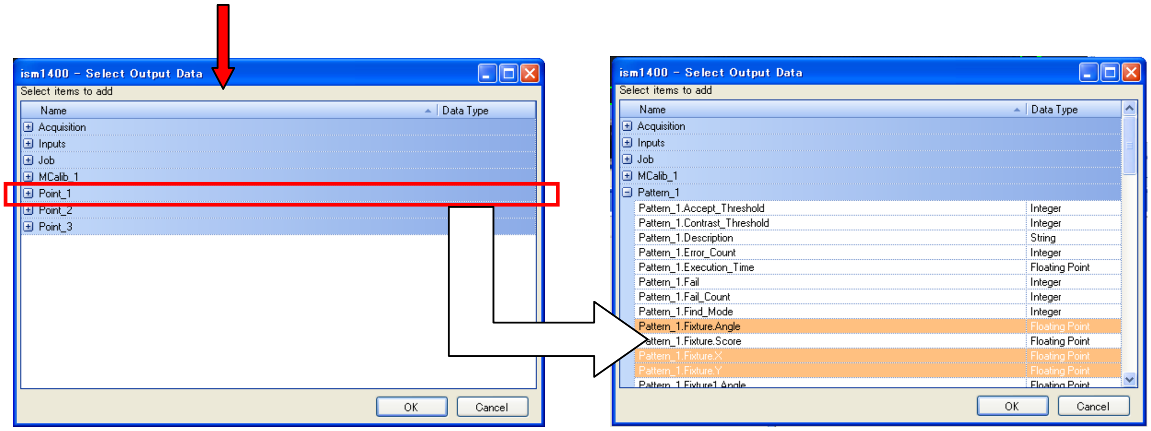


(1) Select “Communication”.

(2) Select “ Add device”.

(3) Click “OK” button after setting as follows.
Device : Robot
Manufacturer : Mitsubishi
Protocol : Ethernet Native String





ism1400 - Select Output Data

Select items to add

Name	Data Type
⊞ Acquisition	
⊞ Inputs	
⊞ Job	
⊞ MCalib_1	
⊞ Point_1	
⊞ Point_2	
⊞ Point_3	

ism1400 - Select Output Data

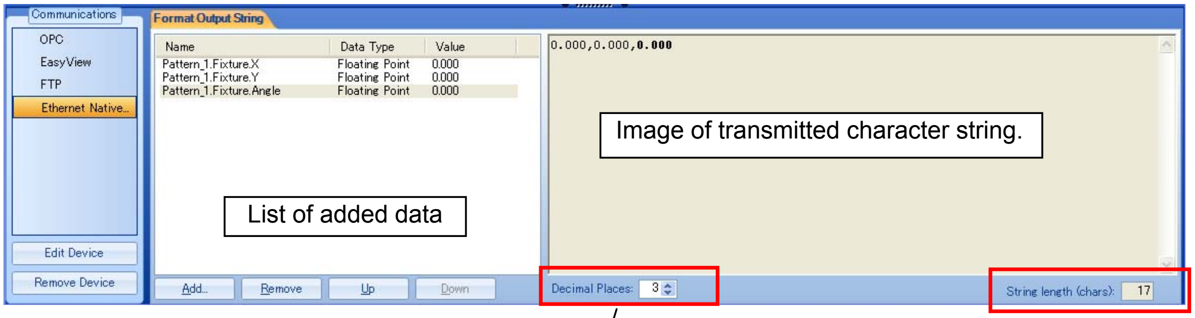
Select items to add

Name	Data Type
⊞ Acquisition	
⊞ Inputs	
⊞ Job	
⊞ MCalib_1	
⊟ Pattern_1	
Pattern_1.Accept_Threshold	Integer
Pattern_1.Contrast_Threshold	Integer
Pattern_1.Description	String
Pattern_1.Error_Count	Integer
Pattern_1.Execution_Time	Floating Point
Pattern_1.Fail	Integer
Pattern_1.Fail_Count	Integer
Pattern_1.Find_Mode	Integer
Pattern_1.Fixture.Angle	Floating Point
Pattern_1.Fixture.Score	Floating Point
Pattern_1.Fixture.X	Floating Point
Pattern_1.Fixture.Y	Floating Point
Pattern_1.Fixture1.Angle	Floating Point

The example above is to register one pattern of the pattern matching, and to recognize two or more pieces. In this case, the recognition result is stored in “Pattern_1", and selects number, X, Y, and θ from it. The table below shows the display name and the meaning of a typical item.

No.	Display	Explanation
1	Pattern_1.Number_Found	Recognized number
2	Pattern_1.Fixture.X	1st recognition results X
3	Pattern_1.Fixture.Y	1st recognition results Y
4	Pattern_1.Fixture.Angle	1st recognition results *θ*
5	Pattern_1.Fixture1.X	2nd recognition results X
6	Pattern_1.Fixture1.Y	2nd recognition results Y
7	Pattern_1.Fixture1.Angle	2nd recognition results *θ*
8	<Same as above>	<Same as above>

If these are added, it becomes as shown in the figure below.



Name	Data Type	Value
Pattern_1.Fixture.X	Floating Point	0.000
Pattern_1.Fixture.Y	Floating Point	0.000
Pattern_1.Fixture.Angle	Floating Point	0.000

```
0.000,0.000,0.000
```

Decimal Places: 3

String length (chars): 17

Image of transmitted character string.

List of added data

There is a limitation in the number of characters that can be transmitted at once. It is possible to save the number of transmission characters by changing the position of decimal point.

Decimal Places: 3 (Initial value : Three digits)

A current number of characters is displayed. Please refer as standards of the number of transmission characters. The number of characters that can be transmitted to the robot controller at once is 240 characters.

String length (chars): 17

This process might change depending on the specification of EasyBuilder. Please ask to Cognex Corporation whether to refer to help of EasyBuilder about latest information.

5.6. Executes the recognition and gets the recognition results

MELFA BASIC V instruction "EBRead" is used to acquire data.
In the following program, three recognition results or less are received.

< Preparation >

(1) PatMax is added with the measurement tool of EasyBuilder, and the pattern and the recognition number (example is 3) are set.

(2) In the communication setting, the Format Output String is set as follows.
number, X1, Y1, angle1, X2, Y2, angle2, X3, Y3, angle3, X4, Y4, angle4, X5, Y5, angle5,

(3) The job is preserved by the name "TEST", and puts the vision sensor "Online".

(4) The robot program is made with RT ToolBox2 as follows.
(It is necessary to set the parameters beforehand.)

```
1 If M_NvOpen(1)<>1 Then         ' When logon has not been completed for vision sensor
2     NVOpen "COM2:" As #1        ' Connects with the vision sensor connected to COM2
3     Wait M_NvOpen(1)=1         ‘ Connects with vision sensor and waits for logon to be completed.
4 EndIf
5 '
6 NVRun #1,"TEST"                ' Starts "TEST" program.
7 EBRead #1,,MNUM,PVS1,PVS2,PVS3   ‘The data of "Job.Robot.FormatString" tag is read,
                                   and they are preserved in the variable MNUM, PVS1, and PVS2.
8 NVClose #1
9 HLT
```

(5) When the robot program is executed, the recognition number and the recognition position are stored in each variable.

6. Robot Program Writing

In order to start (execute) image processing with the vision sensor from the robot, it is necessary to execute commands controlling the vision sensor in a robot program written in MELFA-Basic V.

6.1. Flow for starting of image processing by robot

The method for starting image processing from a robot program is as follows.

1) Check the line connection with the vision sensor (Status variable : M_NVOpen)
2) Line connection with vision sensor (MELFA-BASIC V : NVOpen)
3) Vision program start (MELFA-BASIC V : NVRun)
4) Vision recognition results acquisition (MELFA-BASIC V : EBRead)
5) After this, the robot is moved with the recognition results (the position data) detected with the vision sensor.

For details on the vision program dedicated MELFA-BASIC V commands and status variables, see **"7.1 Vision Sensor Dedicated Commands and Status Variables".**

This manual explains by using MELFA-BASIC V command.

6.2. Samle Robot Program Writing

The robot program below is written and stored in the robot controller. For details on the storage method, see the "RT ToolBox2 User's Manual".

Example for acquiring data in the absolute coordinates using pattern matching.

```
 1 ' Before this program is run, the evacuation position P0, the work grasping position P1, and the work placement position P2 must
have already been taught.
 2 ' Example) P0=(+250.000,+350.000,+300.000,-180.000,+0.000,+0.000)(7,0)
 3 '             P1=(+500.000, +0.000, +100.000, -180.000, +0.000, +10.000)(7,0)
 4 '             P2=(+300.000, +400.00, +100.000, -180.000, +0.000, +90.000)(7,0)
 5   Dim PVS(10)
 6   If M_NvOpen(1)<>1 Then                  ' When logon has not been completed for vision sensor number 1
 7      NVOpen "COM2:" As #1                 ' Connects with the vision sensor connected to COM2
 8   EndIf
 9   Wait M_NvOpen(1)=1                       ' Connects with vision sensor number 1 and waits for logon to be completed.
10   NVRun #1,"Job1"                         ' Start vision program [Job1]
11   EBRead #1, ,MNUM, PVS(1), PVS(2), PVS(3), PVS(4), PVS(5), PVS(6), PVS(7), PVS(8), PVS(9), PVS(10)
12     'The data of "Job.Robot.FormatString" tag is read, and they are preserved in the variable MNUM, and PVS(1) - PVS(10).
13   Mov P0                                          ' Moves to the safe point
14   If (MNUM<1 OR MNUM>10) Then *NG                 ' It jumps to the error when the detection number is 0 or less / 11 or more.
15   For M1=1 To MNUM                                 ' Loops once for each detection by vision sensor number 1.
16      P10=P1                                        ' Creates the target position P10 using the vision sensor 1 results data
17      P10.X=PVS(M1).X
18      P10.Y=PVS(M1).Y
19      P10.C=PVS(M1).C
20      Mov P10,10                               ' Moves to 10 mm above the work grasping position P10.
21      Mvs P10                                  ' Moves to the work grasping position P10.
22      Dly 0.1                                  ' Wait time of 0.1 second
23      HClose 1                                 ' Closes hand 1
24      Dly 0.2                                  ' Wait time of 0.2 second
25      Mvs P10,10                               ' Moves to 10 mm above the work grasping position P10
26      Mov P2,10                                ' Moves to 10 mm above the work placement position P2
27      Mvs P2                                   ' Moves the work placement position P2
28      Dly 0.1                                  ' Wait time of 0.1 second
29      HOpen 1                                  ' Opens hand 1
30      Dly 0.2                                  ' Wait time of 0.2 second
31      Mvs P2,10                                ' Moves to 10 mm above the work placement position P2
32   Next M1                                     ' Repeats.
33   Hlt                                         ' Program pause (Create the appropriate processing.)
34 End                                           ' Exit
35 '
36 *NG                                           ' Error processing
37   Error 9000                                  ' Error 9000 output.
38   Hlt                                         ' Program pause (Create the appropriate processing.)
39 End                                           ' Exit
```

⚠ Caution

Please set parameter NVTRGTMG.

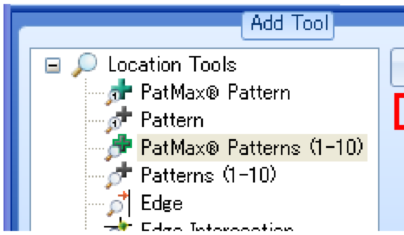
Please set 1 to parameter NVTRGTMG when EBRead command is executed immediately after NVRun command.

When parameter NVTRGTMG is factory shipment setting (NVTRGTMG=2), the next command is executed without waiting the completion of the vision recognition processing. Therefore, there is a possibility being gotten the last recognition result when EBRead command is continuously executed.

6.3. Vision Sensor Settings

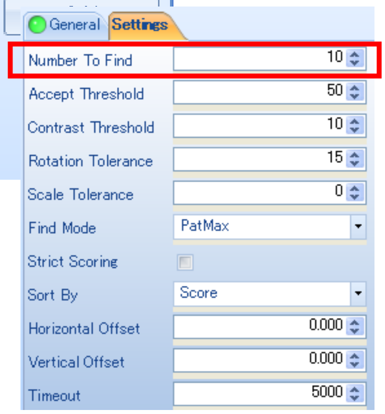
In the example of programming this robot, the following vision setting is assumed.

- The subject of the job : Job1.
- The tool of positioning.
  Tool : PatMax pattern(1-10)
  number of recognition : 10



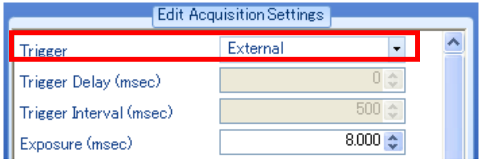
Add Tool

- Location Tools
  - PatMax® Pattern
  - Pattern
  - PatMax® Patterns (1-10)
  - Patterns (1-10)
  - Edge
  - Edge Intersection



Number To Find	10
Accept Threshold	50
Contrast Threshold	10
Rotation Tolerance	15
Scale Tolerance	0
Find Mode	PatMax
Strict Scoring	□
Sort By	Score
Horizontal Offset	0.000
Vertical Offset	0.000
Timeout	5000

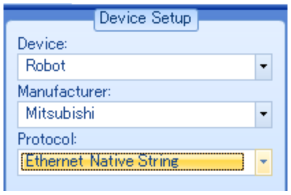
- Edit Acquisition Settings
  Trigger : External, Manual or Network



Edit Acquisition Settings

Trigger	External
Trigger Delay (msec)	0
Trigger Interval (msec)	500
Exposure (msec)	8.000

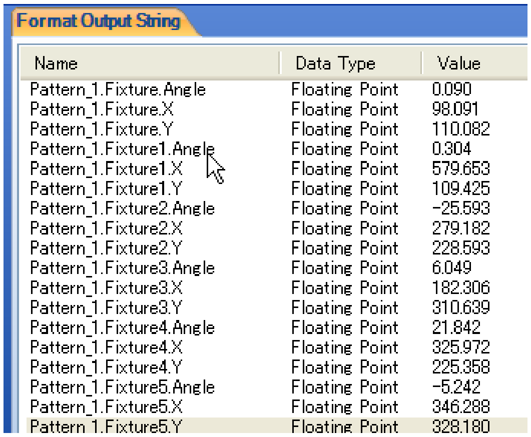
- Communications Settings
  Device : Robot
  Maker : MITSUBISHI ELECTRIC CORPORATION
  Protocol : EtherNet Native String



Device Setup

Device: Robot
Manufacturer: Mitsubishi
Protocol: Ethernet Native String

- Format Output String (item of setting)
  number of recognition (Pattern_1.Number_Found)
  Recognition coordinates 1-X（Pattern_1.Fixture.X）
  Recognition coordinates 1-Y（Pattern_1.Fixture.Y）
  Recognition coordinates 1- Angle（Pattern_1.Fixture.Angle）
  :
  Recognition coordinates 10-X（Pattern_1.Fixture9.X）
  Recognition coordinates 10-Y（Pattern_1.Fixture9.Y）
  Recognition coordinates 10- Angle（Pattern_1.Fixture9.Angle）



Format Output String

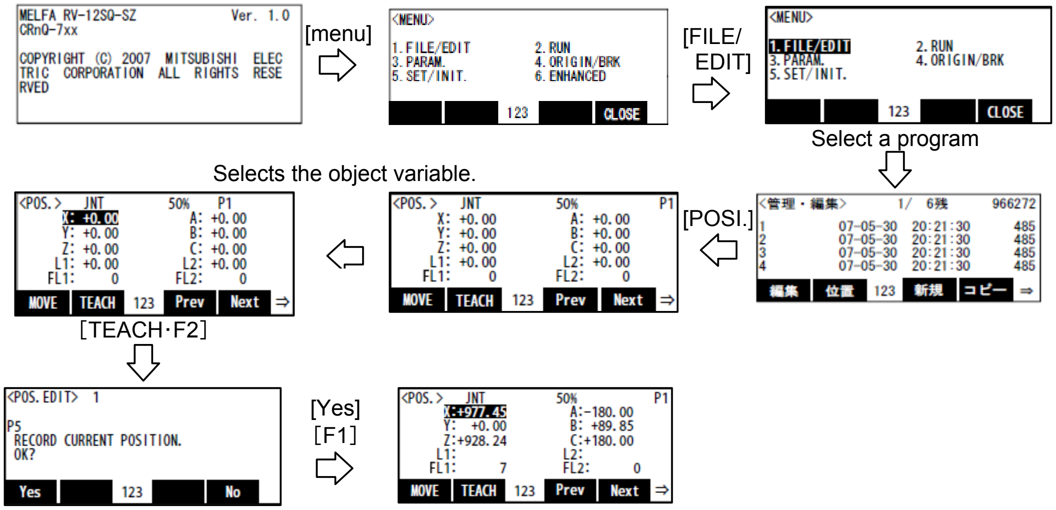
Name	Data Type	Value
Pattern_1.Fixture.Angle	Floating Point	0.090
Pattern_1.Fixture.X	Floating Point	98.091
Pattern_1.Fixture.Y	Floating Point	110.082
Pattern_1.Fixture1.Angle	Floating Point	0.304
Pattern_1.Fixture1.X	Floating Point	579.653
Pattern_1.Fixture1.Y	Floating Point	109.425
Pattern_1.Fixture2.Angle	Floating Point	-25.593
Pattern_1.Fixture2.X	Floating Point	279.182
Pattern_1.Fixture2.Y	Floating Point	228.593
Pattern_1.Fixture3.Angle	Floating Point	6.049
Pattern_1.Fixture3.X	Floating Point	182.306
Pattern_1.Fixture3.Y	Floating Point	310.639
Pattern_1.Fixture4.Angle	Floating Point	21.842
Pattern_1.Fixture4.X	Floating Point	325.972
Pattern_1.Fixture4.Y	Floating Point	225.358
Pattern_1.Fixture5.Angle	Floating Point	-5.242
Pattern_1.Fixture5.X	Floating Point	346.288
Pattern_1.Fixture5.Y	Floating Point	328.180

6.4. Robot teaching

Please teach of the robot positions before executing the automatic operation.

(1) The evaluation position(P0), work grasping position(P1), and work placement position(P2) are taught in order to operate the robot.
Use the teaching box to open the stored robot program and open the position edit screen.

a) Turn the controller's MODE select switch to [MANUAL].
b) Turn ON the teaching pendant (R32TB) TB ENABLE switch.
c) The position edit screen is displayed by pressing [MENU]-[1.FILE/EDIT]-［Select a program］-[POSI.] key.
d) The position variable to be taught is displayed by pressing [↓][↑] key or inputting the position name.
e) The confirmation screen is displayed by pressing ［TEACH･F2］ key. The teaching is executed by pressing [YES･F1] key.
f) Teach all the positions by operating the above processing.



```
MELFA RV-12SQ-SZ          Ver. 1.0
CRnQ-7xx

COPYRIGHT (C) 2007 MITSUBISHI ELEC
TRIC CORPORATION ALL RIGHTS RESE
RVED
```

[menu]

```
<MENU>

1.FILE/EDIT        2.RUN
3.PARAM.           4.ORIGIN/BRK
5.SET/INIT.        6.ENHANCED
                  123        CLOSE
```

[FILE/ EDIT]

```
<MENU>

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

Select a program

```
<管理・編集>         1/   6残      966272
1          07-05-30  20:21:30       485
2          07-05-30  20:21:30       485
3          07-05-30  20:21:30       485
4          07-05-30  20:21:30       485
 編集   位置   123   新規   コピー  ⇒
```

[POSI.]

```
<POS.>   JNT        50%           P1
      X: +0.00       A: +0.00
      Y: +0.00       B: +0.00
      Z: +0.00       C: +0.00
     L1: +0.00      L2: +0.00
    FL1:     0     FL2:     0
 MOVE  TEACH  123  Prev  Next  ⇒
```

Selects the object variable.

```
<POS.>   JNT        50%   P1
      X: +0.00       A: +0.00
      Y: +0.00       B: +0.00
      Z: +0.00       C: +0.00
     L1: +0.00      L2: +0.00
    FL1:     0     FL2:     0
 MOVE  TEACH  123  Prev  Next  ⇒
```

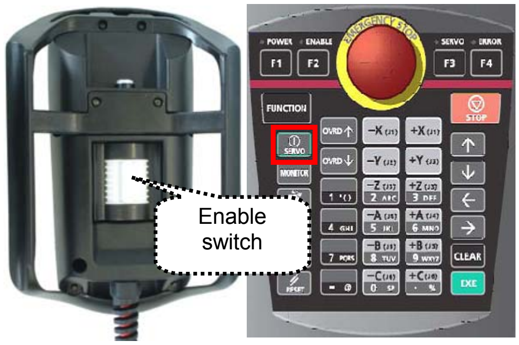
［TEACH･F2］

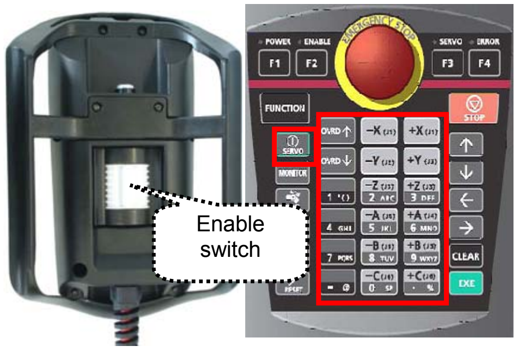
```
<POS.EDIT>  1

P5
RECORD CURRENT POSITION.
OK?
 Yes          123          No
```

[Yes]
[F1]

```
<POS.>   JNT        50%           P1
      X:+977.45      A:-180.00
      Y:  +0.00      B: +89.85
      Z:+928.24      C:+180.00
     L1:            L2:
    FL1:     7     FL2:     0
 MOVE  TEACH  123  Prev  Next  ⇒
```

(2) Move the robot to the evacuation position.
Switch On the robot servo power supply and move the robot with Jog operation.
(a) Press the [Enable switch] + [SERVO] to turn the servo power ON.
(b) Press the [Enable switch] + [each axis ± key (+X/-X, etc.)] to move the robot to the evacuation point.





(3) When this work is complete, press the teaching pendant [CLOSE] button to store the robot program.
(4) Turn OFF the teaching pendant (R32TB) TB ENABLE switch.

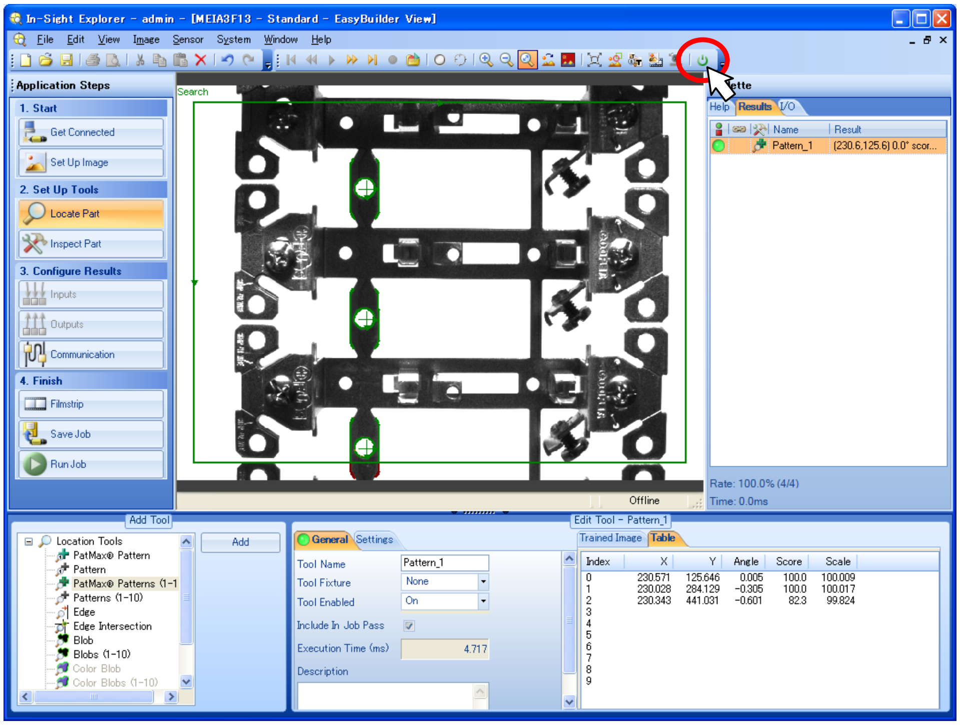
6.5. Executing the automatic operation test

This section explains automatic operation that starts the program created with “6.2 Samle Robot Program Writing” and transports the work recognized with the vision sensor.

6.5.1. Switches the vision sensor to online.

In order for the robot controller to control the vision sensor, it is necessary to switch the vision sensor to "online".

Click ⏻ on the toolbar button.



Name	Result
Pattern_1	(230.6,125.6) 0.0° scor...

Rate: 100.0% (4/4)
Time: 0.0ms

Tool Name: Pattern_1
Tool Fixture: None
Tool Enabled: On
Include In Job Pass: ☑
Execution Time (ms): 4.717
Description:

Index	X	Y	Angle	Score	Scale
0	230.571	125.646	0.005	100.0	100.009
1	230.028	284.129	-0.305	100.0	100.017
2	230.343	441.031	-0.601	82.3	99.824
3					
4					
5					
6					
7					
8					
9					

6.5.2. Test by executing each step

Open the robot program created with the teaching box and while executing one line at a time, check the robot program operations.

For details on the step execution method, see "Detailed explanations of functions and operations (BFP-A8661)" "3.6 Debugging Operations".

⚠ **Caution**

There are command words not completed in a single step execution.

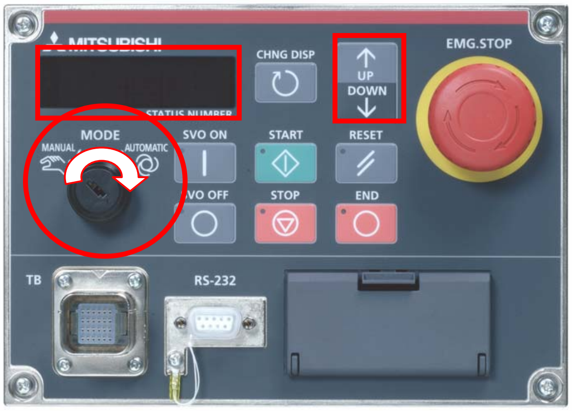
When execution does not move to the next step when the step is executed one time, execute the step again.

Example: NVOpen requires at least seven repetitions of step execution.

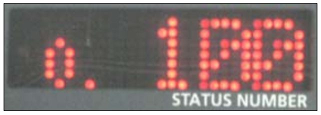
6.5.3. Starting a Robot Program

This section explains the work for starting the stored robot program "1" with the robot controller operation panel.

(1) Turn the controller's MODE select switch to [AUTOMATIC].

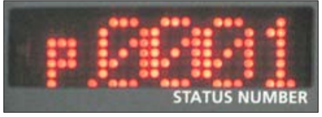


(2) Press the [CHNG DISP] button to display the override at the Status Number.



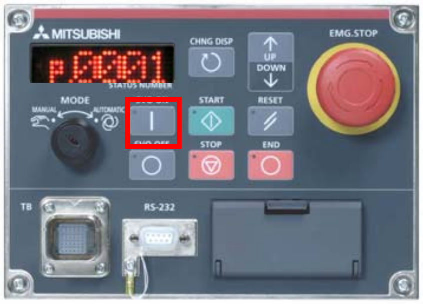
(3) Press the [UP/DOWN] button to set the Status Number display to **"o.010"**. (This sets the robot override to 10%.)

(4) Press the [CHNG DISP] button to display the robot program number at the Status Number.



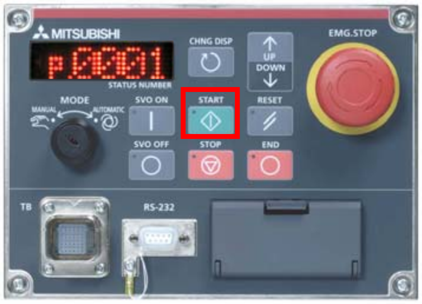
(5) Press the [UP/DOWN] button to set the Status Number display to "P.0001".
(This selects robot program 1.)

(6) Press the [SVO ON] button to switch On the robot servo power supply.

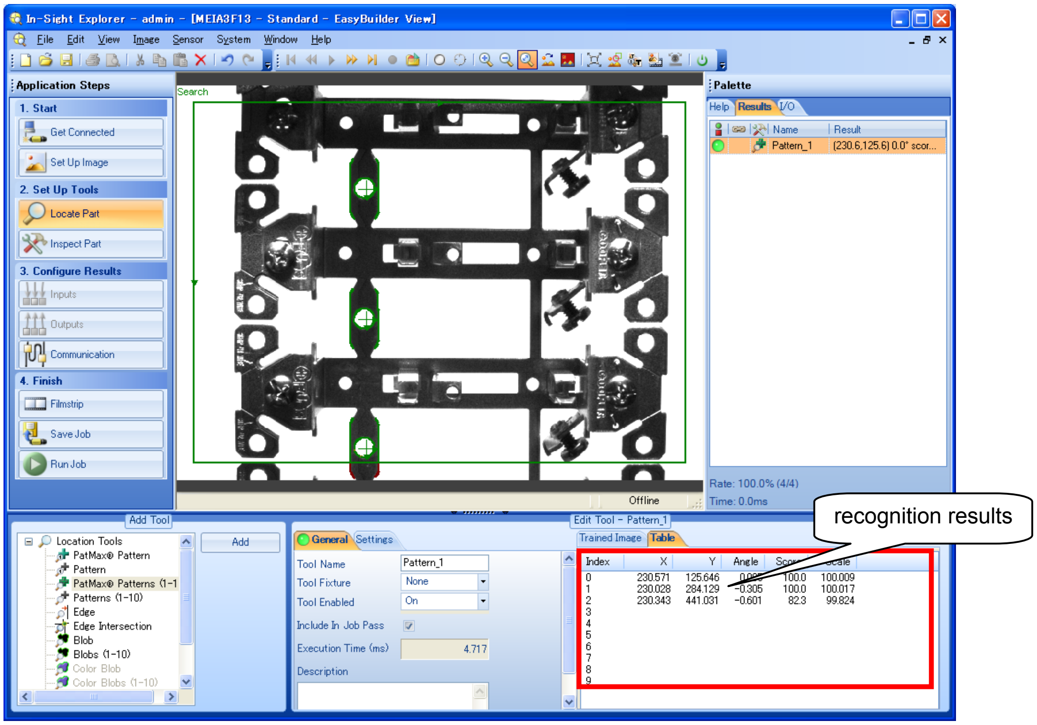


(7) Check around the robot to make sure that everything will be safe even if the robot operates.

(8) Press the [START] button



(9) The robot transports all the work recognized by the vision sensor. After transporting, the robot program stops.



Name	Result
Pattern_1	(230.6,125.6) 0.0° scor...

Rate: 100.0% (4/4)
Time: 0.0ms

Tool Name: Pattern_1
Tool Fixture: None
Tool Enabled: On
Include In Job Pass: ☑
Execution Time (ms): 4.717
Description:

recognition results

Index	X	Y	Angle	Score	Scale
0	230.571	125.646	0.0[illegible]5	100.0	100.009
1	230.028	284.129	-0.305	100.0	100.017
2	230.343	441.031	-0.601	82.3	99.824
3					
4					
5					
6					
7					
8					
9					

7. Detailed Explanation of Functions

This chapter explains the functions of this product in detail.

7.1. Vision Sensor Dedicated Commands and Status Variables

The robot controller has status variables and dedicated commands for controlling vision sensors. This section explains these dedicated commands and status variables.

7.1.1. How to Read Items

[Function] : Shows the command word function.
[Format] : Shows the command word argument input method.
＜＞ indicates an argument.
[ ] indicates that it can be omitted.
□ indicates that a space is required.
[Term] : Shows the argument meaning, range, etc.
[Sample sentence] : Shows a sample sentence.
[Explanation] : Shows the functions in detail and caution item.
[Error] : Shows an error generated when the command word is executed.

7.1.2. MELFA-BASIC V Commands

Here are the dedicated vision sensor commands.

Table 7-1 List of Dedicated Vision Sensor Commands

	Command word	Contents	compliant version of the robot controller
(1)	NVOpen	Connects with the vision sensor and logs on to the vision sensor.	CRnQ-700 series : Ver. N1 or later CRnD-700 series : Ver. P1 or later
(2)	NVClose	Cuts off the connection with vision sensor.	
(3)	NVLoad	Puts the specified vision program into the state in which it can be started.	
(4)	NVRun	Starts the specified vision program.	
(5)	NVTrg	Requests the vision sensor to capture an image and acquires the encoder value after the specified time.	
(6)	EBRead	Reads the data for which the tag name of the vision sensor is specified.	CRnQ-700 series : Ver. R1 or later CRnD-700 series : Ver. S1 or later

Additional command word details are shown below.

(1) NVOpen (network vision sensor line open)

[Function]

Connects with the specified vision sensor and logs on to that vision sensor.

[Format]

```
NVOpen□“<COM number>”□As□#<Vision sensor number>
```

[Term]

<Com number> (Can not be omitted):

Specify the communications line number in the same way as for the Open command.
"COM1:" can not be specified by it is monopolized by the operation panel front RS-232C.
Setting range: "COM2:" – "COM8:"

<Vision sensor number> (Can not be omitted)

Specifies a constant from 1 to 8 (the vision sensor number). Indicates the number for the vision sensor connection to the COM specified with the <COM number>.
Be careful. This number is shared with the <file number> of the Open command.
Setting range: 1 – 8

[Sample sentence]

```
1 If M_NVOpen(1)<>1 Then   'If vision sensor number 1 log on is not complete
2    NVOpen “COM2:” As#1    ' Connects with the vision sensor connected to COM2 and sets its number
                            as number 1.
3 ENDIf
4 Wait M_NVOpen(1)=1        ' Connects with vision sensor number 1 and waits for logon to be completed.
```

[Explanation]

1) Connects with the vision sensor connected to the line specified with the <COM number> and logs on to that vision sensor.
2) It is possible to connect to a maximum of 7 vision sensors at the same time. <Vision sensor numbers> are used in order to identify which vision sensor is being communicated with.
3) When used together with the Open command, the Open command <COM number> and <File number> and the <COM number> and <Vision sensor number> of this command are shared, so use numbers other than those specified with the Open command <COM number> and <File number>.

```
Example: Normal example          Error example
 1 Open “COM1:” As #1         1 Open “COM2:” As #1
 2 NVOpen “COM2:” As #2       2 NVOpen “COM2:” As #2 => <COM number> used
 3 NVOpen “COM3:” As #3       3 NVOpen “COM3:” Ass#1 => <Vision sensor number>
                                                      Used
```

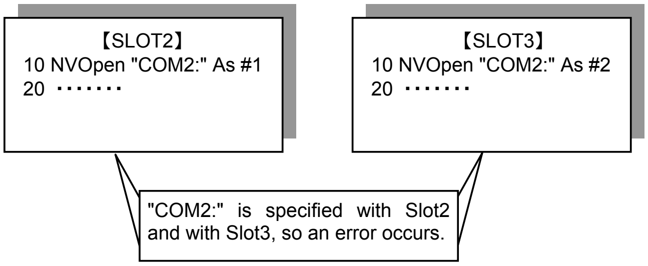
It is not possible to open more than one line in a configuration with one robot controller and one vision sensor. If the same IP address is set as when the [NETHSTIP] parameter was set, an "Ethernet parameter NETHSTIP setting" error occurs.

4) Logging on to the vision sensor requires the "User name" and "Password". It is necessary to set a user name for which full access is set in the vision sensor and the password in the robot controller [NVUSER] and [NVPSWD] parameters.
The user name and password can each be any combination of up to **15** numbers (0-9) and letters (A-Z). **(T/B only supports uppercase letters, so when using a new user, set the password set in the vision sensor with uppercase letters.**)
The user name with full access rights when the network vision sensor is purchased is "admin". The password is "". Therefore, the default values for the [NVUSER] and [NVPSWD] parameters are [NVUSER] = "admin" and [NVPSWD] = "".
When the "admin" password is changed with MELFA-Vision or a new user is registered, change the [NVUSER] and [NVPSWD] parameters. When such a change is made, when the content of the [NVPSWD] parameter is displayed, "****" is displayed. If the vision sensor side password is changed, open the [NVPSWD] parameter and directly change the displayed "****" value. After the making the change, reset the robot controller power.
[Caution]
When multiple vision sensors are connected to one robot controller, set the same user name and password for all of them.
5) The state of communications with the network vision sensor when this command is executed can be checked with M_NVOpen. For details, see the explanation of M_NVOpen.
6) If the program is cancelled while this command is being executed, it stops immediately. In order to log

on to the vision sensor, it is necessary to reset the robot program, then start.

7) When this command is used with multi-tasking, there are the following restrictions. The <COM number> and <Vision sensor number> must not be duplicated in different tasks.

   (a) If the same <COM number> is used in another task, the **"attempt was made to open an already open communication file"** error occurs.

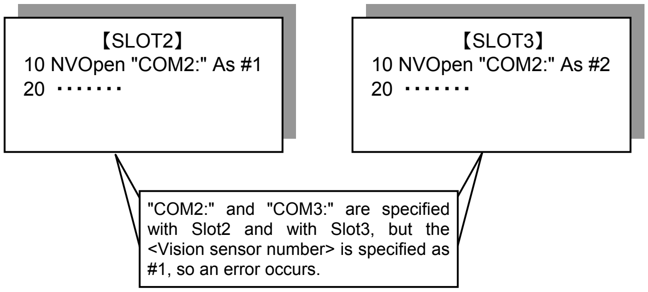


```
【SLOT2】
10 NVOpen "COM2:" As #1
20 ･･･････
```

```
【SLOT3】
10 NVOpen "COM2:" As #2
20 ･･･････
```

"COM2:" is specified with Slot2 and with Slot3, so an error occurs.

   (b) If the same vision sensor number is used in another task, the **"attempt was made to open an already open communication file"** error occurs.



```
【SLOT2】
10 NVOpen "COM2:" As #1
20 ･･･････
```

```
【SLOT3】
10 NVOpen "COM2:" As #2
20 ･･･････
```

"COM2:" and "COM3:" are specified with Slot2 and with Slot3, but the <Vision sensor number> is specified as #1, so an error occurs.

8) A program start condition of "Always" and the continue function are not supported.

9) Three robots can control the same vision sensor at the same time. If a fourth robot logs on, the line for the first robot is cut off, so be careful when constructing the system.

10) The line is not closed with an End command in a program called out with a Callp command, but the line is closed with a main program End command. The line is also closed by a program reset.

11) If an interrupt condition is established while this command is being executed, the interrupt processing is executed immediately even during processing of this command.

[Error]

1) If data type for an argument is incorrect, the **"syntax error in input command"** error is generated.
2) If there is an abnormal number of command arguments (too many or too few), the **"incorrect argument count"** error occurs.
3) If the character specified in <COM number> is anything other than "COM2:" through "COM8:", the **"argument out of range"** error occurs.
4) If the value specified as the <vision sensor number> is anything other than "1" through "8", the **"argument out of range"** error occurs.
5) If a <COM number> for which the line is already connected is specified (including the <File number> for which the line has been opened with an Open command), the **"attempt was made to open an already open communication file"** error occurs.
6) If the vision sensor is not connected before the line is opened, the **"vision sensor not connected"** error occurs. (The same set manufacturer parameter [COMTIMER] as in the Ethernet specifications is used. Currently "1s")
7) If the same <COM number> or the same <vision sensor number> is specified in another task, the **"attempt was made to open an already open communication file"** error occurs.
8) If the user name or password specified in the [NVUSER] parameter (user name) and [NVPSWD] (password) is wrong, the **"wrong password"** error occurs.
9) If the communications line is cut while this command is being executed, the **"abnormal communications"** error occurs and the robot controller side line is closed.
10) If a program is used for which the starting condition is "Always", the **"this command can not be used if the start condition is ERR or ALW"** error occurs.

(2) NVClose(network vision sensor line close)

[Function]

Cuts the line with the specified vision sensor.

[Format]

```
NVClose□[[#]<Vision sensor number>[,[[#]<Vision sensor number>･･･]
```

[Term]

<Vision sensor number> (Can be omitted)

Specifies a constant from 1 to 8 (the vision sensor number). Indicates the number for the vision sensor connection to the COM specified with the <COM number>.
When this parameter is omitted, all the lines (vision sensor lines) opened with an NVOpen command are closed.
Also, up to 8 <vision sensor numbers> can be specified. They are delimited with commas.
Setting range: 1 - 8

[Sample sentence]

```
1 If M_NVOpen(1)<>1 Then     ' When logon has not been completed for vision sensor number 1
2    NVOpen “COM2:” ASs#1    ' Connects with the vision sensor connected to COM2 and sets its number
                             as number 1.
3 EneIf
4 Wait M_NVOpen(1)=1         'Connects with vision sensor number 1 and waits for logon to be completed.
5 ･････
 :
100 NVClose #1               'Cuts the line with the vision sensor connected to COM2.
```

[Explanation]

1) Cuts the line with the vision sensor connected with the NVOpen command.
2) If the <vision sensor number> is omitted, cuts the line with all the vision sensors.
3) If a line is already cut, execution shifts to the next step.
4) Because up to seven vision sensors can be connected at the same time, <Vision sensor numbers> are used in order to identify which vision sensor to close the line for.
5) If the program is cancelled while this command is being executed, execution continues until processing of this command is complete.
6) When this command is used with multi-tasking, in the task using this command, it is necessary to close only the lines opened by executing an NVOpen command . At this time, use the <Vision sensor number> specified with the NVOpen command.
7) A program start condition of "Always" and the continue function are not supported.
8) If an End command is used, all the lines opened with an NVOpen command or Open command are closed. However, lines are not closed with an End command in a program called out with a CAllp command.
   Lines are also closed by a program reset, so when an End command or a program reset is executed, it is not necessary to close lines with this command.
9) The continue function is not supported.
10) If an interrupt condition is established while this command is being executed, the interrupt processing is executed after this command is completed.

[Errors]

1) If the value specified as the <vision sensor number> is anything other than "1" through "8", the **"argument out of range"** error occurs.
2) If there are more than eight command arguments, an "incorrect argument count" error occurs.

(3) NVLoad (network vision sensor load)

[Function]

Loads the specified vision program into the vision sensor.

[Format]

```
NVLoad□#<Vision sensor number>,”<Vision program (job) name>”
```

[Term]

<Vision sensor number> (Can not be omitted)

This specifies the number of the vision sensor to control.
Setting range: 1 - 8

<Vision program (job) name> (Can not be omitted)

Specifies the name of the vision program to start.
The vision program extension (.job) can be omitted.
The only characters that can be used are "0" - "9", "A" - "Z", "a" - "z", "-", and "_".

[Sample sentence]

```
100 If M_NVOpen(1)<>1 Then       'If vision sensor number 1 log on is not complete
110     NVOpen “COM2:” As #1     'Connects with the vision sensor connected to COM2.
120 EndIf
130 Wait M_NVOpen(1)=1          ' Connects with vision sensor number 1 and waits for logon to be
                                 completed.
140 NVLoad #1,”TEST”            'Loads the "Test".
150 NVPst #1, ””,”E76”,”J81”,”L84”,0,10
'Receives the recognition count recognized with the "Test" program from the E76 cell and the recognition
                                 results from cells J81 through L84, and stores them in P_NvS1().
160 ・・・・
   :
300 NVClose #1     'Cuts the line with the vision sensor connected to COM2.
```

[Explanation]

1) Loads the specified vision program into the specified vision sensor.
2) This command moves to the next step at the point in time when the vision program is loaded into the vision sensor.
3) If the program is cancelled while this command is being executed, it stops immediately.
4) If the specified <vision program name> is already loaded, the command ends with no processing.
5) When this command is used with multi-tasking, it is necessary to execute the NVOpen command in the task using this command. Also, use the <vision sensor number> specified with the NVOpen command.
6) A program start condition of "Always" and the continue function are not supported.
7) If an interrupt condition is established while this command is being executed, the interrupt processing is executed immediately.

[Errors]

1) If data type for an argument is incorrect, a **"syntax error in input command statement"** error is generated.
2) If there is an abnormal number of command arguments (too many or too few), an **"incorrect argument count"** error occurs.
3) If the <vision sensor number> is anything other than "1" through "8", an **"argument out of range"** error occurs.
4) If the NVOpen command is not opened with the number specified as the <vision sensor number>, an **"abnormal vision sensor number specification"** error occurs.
5) If the <vision program name> exceeds 15 characters, an **"abnormal vision program name"** error occurs.
6) If a <vision program name> uses a character other than "0" – "9", "A" – "Z", "-", or "_" (including lowercase letters), an **"abnormal vision program name"** error occurs.
7) If the program specified in the <vision program name> is not in the vision sensor, a **"vision program does not exist"** error occurs.
8) If the vision sensor is "offline", the **"Put online"** error occurs, so put the vision sensor "Online".
9) If the communications line is cut while this command is being executed, an **"abnormal communications"** error occurs and the robot controller side line is closed.

(4) NVRun (network vision sensor run)

[Function]

Starts the specified vision program.

[Format]

```
NVRun□#<Vision sensor number >,< Vision program (job) name >
```

[Term]

<Vision sensor number> (Can not be omitted)

This specifies the number of the vision sensor to control.
Setting range: 1 - 8

<Vision program (job) name> (Can not be omitted)

Specifies the name of the vision program to start.
The vision program extension (.job) can be omitted.
The only characters that can be used are "0" - "9", "A" - "Z", "a" - "z", "-", and "_".

[Sample sentence]

```
100 If M_NvOpen(1)<>1 Then   'If vision sensor number 1 log on is not complete
110   NVOpen "COM2:" As #1     'Connects with the vision sensor connected to COM2.
120 End If
130 Wait M_NvOpen(1)=1          ' Connects with vision sensor number 1 and waits for logon to be
completed.
140 NVLoad #1,"TEST"                    'Loads the "Test" program.
150 NVRun #1,"TEST"                     'Starts the "Test" program..
160 EBRead #1,,MNUM,PVS1,PVS2 'The data of "Job.Robot.FormatString" tag is read,
                                             and they are preserved in the variable MNUM, PVS1, and PVS2.
170   :
300 NVClose #1                     'Cuts the line with the vision sensor connected to COM2.
```

[Explanation]

1) Loads the specified vision program into the specified vision sensor.
2) **The timing of completing this command processing is different depending on the value of parameter NVTRGTMG. When the parameter NVTRGTMG is a factory shipment setting, the next command is executed after completing the communication of image processing instruction (taking picture demand) between the vision sensor and robot controller.**
3) If the program is cancelled while this command is being executed, it stops immediately.
4) If the specified <vision program name> is already loaded, only image capture and image processing are executed. (The vision program is not loaded.)）
5) For receiving data from the vision sensor, use the EBRead command.
6) When this command is used with multi-tasking, it is necessary to execute the NVOpen command in the task using this command. Also, use the <vision sensor number> specified with the NVOpen command.
7) A program start condition of "Always" and the continue function are not supported.
8) Please set the trigger of EasyBuilder's image capture to "External trigger", "Manual triggeroperation" or "Network".
   (It is possible to use "Camera" when the value of paramete NVTRGTMG is 0 or 2.)
9) Up to three robots can control the same vision sensor at the same time, but this command can not be used by more than one robot at the same time. Use this command on any one of the robots.
10) If an interrupt condition is established while this command is being executed, the interrupt processing is executed immediately.

[Errors]

1) If data type for an argument is incorrect, a **"syntax error in input command statement"** error is generated.
2) If there is an abnormal number of command arguments (too many or too few), an **"incorrect argument count"** error occurs.
3) If the <vision sensor number> is anything other than "1" through "8", an **"argument out of range"** error occurs.
4) If the NVOpen command is not opened with the number specified as the <vision sensor number>, an **"abnormal vision sensor number specification"** error occurs.
5) If the <vision program name> exceeds 15 characters, an **"abnormal vision program name"** error occurs.
6) If a <vision program name> uses a character other than "0" – "9", "A" – "Z", "-", or "_" (including lowercase letters), an **"abnormal vision program name"** error occurs.
7) If the program specified in the <vision program name> is not in the vision sensor, a **"vision program does not exist"** error occurs.
8) If EasyBuilder's image capture specification is set to anything other than "Camera" (all trigger command), "External trigger", or "Manual trigger", an "abnormal image capture specification" error occurs.
   (When the value of parameter NVTRGTMG is 1 and the image capture specification is set to "Camera", the same error occurs.)
9) If the vision sensor is "offline", the **"Put online"** error occurs, so put the vision sensor "Online".
10) If the communications line is cut while this command is being executed, an **"abnormal communications"** error occurs and the robot controller side line is closed。

(5) NVTrg (network vision sensor trigger)

[Function]

Requests the specified vision program to capture an image.

[Format]

```
NVTrg□#<Vision sensor number>,<delay time>, <encoder 1 value read-out variable>
           [,[<encoder 2 read-out variable>][,[<encoder 3 value read-out variable>]
           [,[<encoder 4 read-out variable>][,[<encoder 5 value read-out variable>]
           [,[<encoder 6 read-out variable>][,[<encoder 7 value read-out variable>]
           [,[<encoder 8 read-out variable>]
```

[Term]

<Vision sensor number> (Can not be omitted)

This specifies the number of the vision sensor to control.
Setting range: 1 - 8

<Delay time> (Can not be omitted)

This specifies the delay time (in ms) from when the image capture request is output to the vision sensor until the encoder value is obtained.
Setting range: 0 - 150 ms

<Encoder n value read-out variable> (Can be omitted from the second one on)

Specifies the double precision numeric variable into which the read out external encoder n value is set.
Note: n is 1 - 8.

[Sample sentence]

```
1 If M_NVOpen(1)<>1 Then   'If vision sensor number 1 logon is not complete
2    NVOpen “COM2:” As #1    'Connects with the vision sensor connected to COM2.
3 EndIf
4 Wait M_NVOpen(1)=1        ' Connects with vision sensor number 1 and waits for logon to be completed.
5 NVRun #1,”TEST”             'Starts the "Test" program.
6 NVTrg #1,15,M1#,M2#        'Requests the vision sensor to capture an image and acquires encoders 1
                             and 2 after 15 ms.
7 NVIn #1, ”TEST”,”E76”,”J81”,”L84”,0,10
'Receives the recognition count recognized with the "Test" program from the E76 cell and the recognition
                             results from cells J81 through L84, and stores this in P_NvS1 ().
8 ････
 :
100 NVClose #1    'Cuts the line with the vision sensor connected to COM2.
```

[Explanation]

1) Outputs the image capture request to the specified vision sensor and acquires the encoder value after the specified time. The acquired encoder value is stored in the specified numeric variable.
2) This command moves to the next step at the point in time when the encoder value is acquired the specified time after the image capture request to the vision sensor.
3) If the program is cancelled while this command is being executed, it stops immediately.
4) For receiving data from the vision sensor, use the NVIn command.
5) When this command is used with multi-tasking, it is necessary to execute the NVOpen command in the task using this command. Also, use the <vision sensor number> specified with the NVOpen command.
6) A program start condition of "Always" and the continue function are not supported.
7) Up to three robots can control the same vision sensor at the same time, but this command can not be used by more than one robot at the same time. Use this command on any one of the robots.
8) If an interrupt condition is established while this command is being executed, the interrupt processing is executed immediately.

[Errors]

1) If data type for an argument is incorrect, a **"syntax error in input command statement"** error is generated.
2) If there is an abnormal number of command arguments (too many or too few), an **"incorrect argument count"** error occurs.
3) If the <vision sensor number> is anything other than "1" through "8", an **"argument out of range"** error occurs.
4) If the NVOpen command is not opened with the number specified as the <vision sensor number>, an **"abnormal vision sensor number specification"** error occurs.
5) If the vision program's image capture specification is set to anything other than "Camera" (all trigger command), "External trigger", or "Manual trigger", an **"abnormal image capture specification"** error occurs.
6) If the vision sensor is "offline", the **"Put online"** error occurs, so put the vision sensor "Online".
7) If the communications line is cut while this command is being executed, an **"abnormal communications"** error occurs and the robot controller side line is closed.

(6) EBRead(EasyBuilder read)

[Function]

Reads out the data by specifying the tag name of the vision sensor.

The data read from the vision sensor is stored in the specified variable.

Please read out data specifying the tag name by using this command when the vision program (job) is made with the vision tool EasyBuilder made by Cognex Corporation.

[Format]

```
EBRead□#<Vision sensor number>,[<Tag name>],<variable name 1> [,<variable name 2>]..[,<Time out>]
```

[Term]

<Vision sensor number> (Can not be omitted)

This specifies the number of the vision sensor to control.
Setting range: 1 - 8

<Tag name> (Can be omitted)

Specifies the name of symbolic tag where data read out by the vision sensor is stored .
When omitting it, the value of paraemter EBRDTAG (initial value is the custom format tag name "Job.Robot.FormatString") is set to it.

<variable name>(Can not be omitted)

Specifies the variable where the data read from the vision sensor is stored.
It is possible to use two or more variables by delimited with commas.
It is possible to specify the Numeric value variable, Position variable or String variable.
When the Position variable is specified, the value is set to X, Y, and C element, and 0 is set to other elements.

<Time out> (If omitted, 10)

Specifies the time-out time (in seconds).
Specification range: Integer 1-32767

[Sample sentence]

```
100 If M_NvOpen(1)<>1 Then      'If vision sensor number 1 log on is not complete
110    NVOpen "COM2:" As #1      'Connects with the vision sensor connected to COM2.
120 End If
130 Wait M_NvOpen(1)=1       ' Connects with vision sensor number 1 and waits for logon to be completed.
140 NVLoad #1,"TEST"          'Loads the "Test" program.
150 NVRun #1,"TEST"           'Starts the "Test" program..
160 EBRead #1,,MNUM,PVS1,PVS2      'The data of "Job.Robot.FormatString" tag is read,
                                    and they are preserved in the variable MNUM, PVS1, and PVS2.
170 --------
  :
300 NVClose #1                    'Cuts the line with the vision sensor connected to COM2.
```

[Explanation]

1) Gets the data by specifying the tag name from an active vision program in the specified vision sensor.
2) The data read from the vision sensor is stored in the specified variable.
3) When the specified variable identifier is delimited by comma and enumerated when the data of the vision sensor is two or more values (character string) delimited by comma, data is stored in order of describing the variable identifier. In this case, the type of the object data should be the same as the type of the variable.
4) When the position variable is specified, the vision data is stored in X, Y, and C element. And the value of other elements are 0.
   The value converted into the radian is set to C element.
5) The value of receiving data are set only to the specified variables when the number of specified variables is less than that of receive data.
6) The variable more than the number of receiving data is not updated when the number of specified variables is more than that of receive data.
7) When the tag name is omitted, the value of parameter EBRDTAG is set instead of the tag name. (The factory shipment setting is " Job.Robot.FormatString".)
8) It is possible to specify the timeout time by the numerical value. Within the timeout time, does not move to the next step until the results are received from the vision sensor. However, if the robot program is stopped, this command is immediately cancelled. Processing is continued with a restart.
9) When this command is used with multi-tasking, it is necessary to execute the NVOpen command and the NVRun command in the task using this command. In this case, use the <vision sensor number> specified with the NVOpen command.
10) A program start condition of "Always" and the continue function are not supported.
11) If an interrupt condition is established while this command is being executed, the interrupt processing is executed immediately even during processing of this command. The processing is executed after completing the interrupt processing.
12) **In order to shorten the tact time, it is possible to do other work after executing the NVRun command and execute EBRead when it is required.**
13) **Please set 1 to the parameter NVTRGTMG if the EBRead command is executed immediately after the NVRun command.**
    When parameter NVTRGTMG is factory shipment setting,the next process of the NVRun command is executed without waiting for the completion of the vision recognition processing. Therefore, the last recognition might be taken when the EBRead command is continuously executed.
14) Note that if the program stops between NVRun and EBRead, the results when NVRun is executed and the results when EBRead is executed may be different.

< Value of the variable>

The variable by executing the EBRead command is as follows.

(1) Content of specified tag (Pattern_1.Number_Found) is 10

(a) The value when “EBRead #1,"Pattern_1.Number_Found",MNUM” is executed is :
-> MNUM=10

(b) The value when “EBRead #1,"Pattern_1.Number_Found",CNUM” is executed is :
-> CMNUM="10"

(2) Content of specified tag (Job.Robot.FormatString) is 2, 125.75, 130.5, -117.2, 55.1, 0, 16.2

(a) The value when “EBRead #1,,MNUM,PVS1,PVS2” is executed is :

```
-> MNUM=2
        PVS1.X=125.75    PVS1.Y=130.5    PVS1.C=-117.2
        PVS2.X=55.1      PVS2.Y=0,       PVS2.C=16.2
            * The element (Excluding X and Y element) that the vision data is not set is 0.
```

(b) The value when “EBRead #1,,MNUM,MX1,MY1,MC1,MX2,MY2,MC2” is executed is :

```
-> MNUM=2
        MX1=125.75    MY1=130.5    MC1=-117.2
        MX2=55.1      MY2=0        MC2=16.2
```

(c) The value when “EBRead #1,,CNUM,CX1,CY1,CC1,CX2,CY2,CC2” is executed is :

```
-> CNUM="2"
        CX1="125.75"    CY1="130.5"    CC1="-117.2"
        CX2="55.1"      CY2="0"        CC2="16.2"
```

(3) Content of specified tag (Job.Robot.FormatString) is 2, 125.75, 130.5

(a) The value when “EBRead #1,,MNUM,PVS1” is executed is :

-> MNUM=2

PVS1.X=125.75 PVS1.Y=130.5

* The element (Excluding X and Y element) that the vision data is not set is 0.

[Errors]

1) If data type for an argument is incorrect, a **"syntax error in input command statement"** error is generated.
2) If there is an abnormal number of command arguments (too many or too few), an **"incorrect argument count"** error occurs.
3) If the <vision sensor number> is anything other than "1" through "8", an **"argument out of range"** error occurs.
4) If the NVOpen command is not opened with the number specified as the <vision sensor number>, an **"The NVOpen command is not executed."** error occurs.
5) If data type of the strings data received from the vision sensor and the variable substituted for it is difference, a **" Illegal Receive data(EBREAD)** " error is generated.
6) If the <Timeout> is other than "1" – "32767", an **"argument out of range"** error occurs.
7) If the vision sensor does not respond without the time specified as the <Timeout> or within the first 10 seconds if the <Timeout> parameter is omitted, a **"vision sensor response timeout"** error occurs.
8) If the communications line is cut while this command is being executed, an **"abnormal communications"** error occurs and the robot controller side line is closed.
9) If the specified tag name does not exist in the active vision program, a **“Vision Tag name is abnormal”** error is generated.
10) Please specify 31 variables or less
('number of the recognition' +' position in the coordinate (X,Y,Z)' x 10) .
If 32 variables or more are specified, a **"syntax error in input command statement"** error is generated.
11) If the <vision program name> exceeds 15 characters, an "abnormal vision program name" error occurs.
12) If a <vision program name> uses a character other than "0" – "9", "A" – "Z", "-", or "_" (including lowercase letters), an "abnormal vision program name" error occurs.
13) If the program specified in the <vision program name> is not in the vision sensor, a **"vision program does not exist"** error occurs.
14) If the program specified in the <vision program name> is not started by an NVRun command, a **"abnormal vision program name"** error occurs.
15) If the <Recognition count cell>, <Start cell>, or <End cell> contains a number other than "0" – "399" or a letter other than "A – "Z", an **"argument out of range"** error occurs.
16) If there is no value in the cell specified in "Recognition count cell", an **"invalid value in specified for recognition count cell"** error occurs.
17) If the <Start cell> and <End cell> are reversed, a **"specified cell value out of range"** error occurs.
18) If the number of data included in the cell which specifies it by <Start cell> and <End cell> exceeds 90, a**"specified cell value out of range"** error occurs.
19) If the range specified by <Star cell> and <End cell> exceeds line 30 and row 10, a**"specified cell value out of range"** error occurs.
20) If the <Type> is other than "0" - "7", an **"argument out of range"** error occurs.

7.1.3. Robot status variables

Here are the status variables for vision sensors.
Be careful. The data for these status variables is not backed up by the RT ToolBox2 backup function.
These status variables can be used by the following software versions.
CRnQ-700 series: N1 or later
CRnD-700 series: P1 or later

Table 7-2 List of Robot status variables for vision sencer

Variable name	Array elements	Contents	Attribute (*)	Data type
M_NvOpen	8	Line connection status	R	Integer type

(*1) R indicates that a status variable is read-only.

The details of the status variables are as follows.

(1) M_NvOpen

[Function]
Indicates the vision sensor line connection status.

[Array meaning]
Array elements (1 - 8) : Vision sensor numbers

[Explanation of values returned]
0: Line connecting (logon not complete)   1: Logon complete   -1: Not connected

[Usage]
After an NVOpen command is executed, checks whether or not the line with the vision sensor is connected and the vision sensor logged onto.

[Sample sentence]

```
1 If M_NVOpen(1)<>1 Then    ' If vision sensor number 1 is not connected
2    NVOpen “COM2:” As #1   ' Connects with the vision sensor connected to COM2 and sets its number
as number 1.
3 EndIf
4 Wait M_NVOpen(1)=1         ' Connects with vision sensor number 1 and waits for the logon state.
5 ･････
 :
100 NVClose #1               'Cuts the line with the vision sensor connected to COM2.
```

[Explanation]

1) Indicates the status of a line connected with a network vision sensor with an NVOpen command when the line is opened.
2) The initial value is "-1". At the point in time that the NVOpen command is executed and the line is connected, the value becomes "0" (line connecting). At the point in time that the network vision sensor logon is completed, the value becomes "1" (logon complete).
3) This variable strongly resembles the status of status variable M_OPEN, but whereas M_Open
4) becomes "1" when the connection is verified, M_NVOpen becomes "1" when the vision sensor logon is complete.

[Errors]

(1) If the type of data specified as an array element is incorrect, a **"syntax error in input command statement"** error occurs.
(2) If there is an abnormal number of array elements (too many or too few), an **"incorrect argument type"** error occurs.
(3) If an array element other than "1" through "8" is specified, an **"array element mistake"** error occurs.

7.1.4. Parameter for the vision sensor

The parameter of the robot for the vision sensor is as follows.

Table 7-3 List of the parameters for the vision sensor

<table>
<tr><th>Parameter</th><th>Paraemter name</th><th>No. of arrays</th><th>Details explanation</th><th>Factory setting</th></tr>
<tr><td>User name</td><td>NVUSER</td><td>Character string 1</td><td>The user name to log on the vision sensor is set.
(no more than 15 characters)</td><td>"admin"</td></tr>
<tr><td>Password</td><td>NVPSWD</td><td>Character string 1</td><td>The password to log on the vision sensor is set.
(no more than 15 characters)</td><td>""</td></tr>
<tr><td>Timing of trigger
Notes)
The software version that can be used is as follows.
- SQ series : version N2B or later
- SD series : Version P2B or later</td><td>NVTRGTMG</td><td>Integer1</td><td>Defines about processing of NVRun command and NVTrg command.
The content of each set value is as follows.
<table>
<tr><th>value</th><th>NVRun</th><th>NVTrg</th></tr>
<tr><td>0</td><td>Trigger</td><td>Trigger</td></tr>
<tr><td>1</td><td>Trigger+
Image processing</td><td>Trigger+
Image processing</td></tr>
<tr><td>2</td><td>Trigger</td><td>Trigger+
Image processing</td></tr>
</table>
- Trigger : The next command is executed after the communication of the image processing instruction (taking picture demand) and the vision sensor is completed.
It is possible to shorten the tact time by setting this value when another process of the robot is executed while the vision sensor is processing the image.

-Trigger + Image processing : The robot controller request the image processing (capture request) to the vision sensor. And, the next command of the robot is executed after the vision image processing is completed.
When the robot requests the image processing to the vision sensor and gets the recognition results (EBRead command is executed) with next step, please set this</td><td>2</td></tr>
<tr><td>Initial value of tag name specified by EBRead command
Notes)
The software version that can be used is as follows.
- SQ series : version R1 or later
- SD series : Version S1 or later</td><td>EBRDTAG</td><td>Character string 1</td><td>Sets up an initial value of the "symbolic tag name" used with EBRead command (it is 128 characters or less)
When the tag name of EBRead command is omitted, the value of this parameter is specified.</td><td>"Job.Robot.FormatString"</td></tr>
</table>

8. Troubleshooting

This chapter lists the errors that can occur in using network vision sensors and explains the causes of and solutions to these errors.

8.1. Error list

Below are the messages for error numbers, their causes, and the solutions.
The meanings of the error levels in the table are as follows.

Table 8-1 Error Category List

Level	Explanation
H High-level error	The servo is switched Off and program execution stopped.
L Low-level error	Program execution is stopped.
C Warning	Program execution is continued.

Table 8-2 List of Errors for Vision Sensor Use

Level	Error No.	Error contents	Cause	Solution
L	3110	Argument out of range	One of the argument values specified in a command is out of its range.	Check the argument range and re-input.
L	3120	Incorrect argument count	The number of arguments in the executed command is incorrect.	Check the argument count and re-input.
L	3130	Attempt was made to open an already open communication file.	The communications line that was the subject of the attempted opening is already open.	Check the COM number and vision sensor number and re-execute. Or check the communications parameters.
L	3141	The NVOpen command is not executed.	No NVOpen command was executed before execution of a command communicating with the vision sensor.	Revise the robot program to execute the NVOpen command.
L	3142	The communication line can not be opened.	The line for communication with the vision sensor can not be opened.	Check the communication cable or the communications parameters.
L	3287	This command can not be used if the start condition is ERR or ALW.	This command can not be used if the start condition is ERR or ALW.	Revise the program.
L	3501	Illegal Receive data(EBREAD)	The type of the data received by EBRead command and the type of specified variable are different.	Revise the program.
L	3810	Incorrect argument type	The arithmetic calculation, single-item calculation, comparison, or function argument type is incorrect.	Specify the correct argument. Please confirm the content of the data of specified tag of the vision sensor.
L	4220	Syntax error in input command	There is a mistake in the structure of the input command.	Check the program contents, then re-input with the correct syntax.
L	4370	Array element mistake	1. An array element is outside the defined range. 2. A variable was specified that can not be arrayed.	1. Revise the array elements to be within 1 – maximum element. 2. Stop array element specification.
L	7810	Abnormal Ethernet parameter setting	The parameter setting is incorrect.	Check the NETHSTIP, NETPORT, NETMODE, and other such parameters.

Table 8-3 List of Errors Only for Vision Sensors

Level	Error No.	Error contents	Cause	Solution
L	8600	Vision sensor not connected	There is no vision sensor connected to the specified COM number.	Check the specified vision program number, "COMDEV" parameter, etc. settings.
L	8601	Logon not possible	The communication line was opened, but there is no response from the vision sensor.	Reset the program and start it again.
L	8602	Wrong password	The password for the user set with the "NVUSER" password is not set in the "NVPSWD" parameter.	Set the correct password.
L	8603	Parameter abnormality	The user name or password parameter is abnormal.	Check the NVUSER and NVPSWD parameters.
L	8610	Abnormal communications	Communication with the vision sensor was cut off before or during command execution.	Check the communication cable between the robot and vision sensor.
L	8620	Abnormal vision sensor number specification	The specified vision sensor number is not defined with an NVOpen command.	Check that the specified vision sensor number is correct. Also, check that that number is defined with an NVOpen command.
L	8621	Abnormal vision program name	The specified vision program name is more than 15 characters.	Specify a vision program name with no more than 15 characters.
L	8622	Vision program not present.	The specified program does not exist in the specified vision sensor.	Check whether the specified vision program exists in the specified vision sensor. Also check that the vision program name specified is correct.
L	8630	Incorrect value in recognition count cell	The recognition count value was not in the cell specified as the recognition count cell.	Check that the correct cell is specified.
L	8631	Specified cell value out of range	Corresponding to either the following. ・The values specified for the start cell and end cell are reversed. ・The range specified by Start Cell and End Cell exceeds line 30 and row 10. ・The number of data included in the cell which specifies it by Start Cell and End Cell exceeds 90.	・Check that the correct cell is specified. ・ Check the number of data acquired from the cell which specifies it by Start Cell and End Cell.
L	8632	Vision sensor response timeout	There is no response from vision sensor within the specified time or within a specific time.	Check that the specified time is correct. Or check that the vision sensor settings are correct.
L	8633	NVTRG response timeout	No response to image capture request.	Check the communications cable.
L	8636	Vision Tag name is abnormal	The active specified symbolic tag does not exist in the vision program.	Please confirm the name of symbolic tag of Easy Builder is corresponding to the tag name specified by the robot program, and correct the tag name.
L	8640	Abnormal image capture specification	The image capture specification is other than "Network", "external", and "manual".	Specify an image capture specification of "Network", "external", or "manual".
L	8650	Put online.	The vision sensor is offline.	Put the vision sensor online to enable control from the outside.
L	8660	Not permitted to control vision sensor	The NVUSER and NVPSWD parameters set for logging on to the vision sensor do not have the right to full access to the vision sensor.	Check the vision sensor side user list registration and specify the name of a user with full access in NVUSER and their password in NVPSWD.

L	8670	Restart not possible after stop	After the program was stopped, it was started without being reset.	Reset the robot program, then start it.

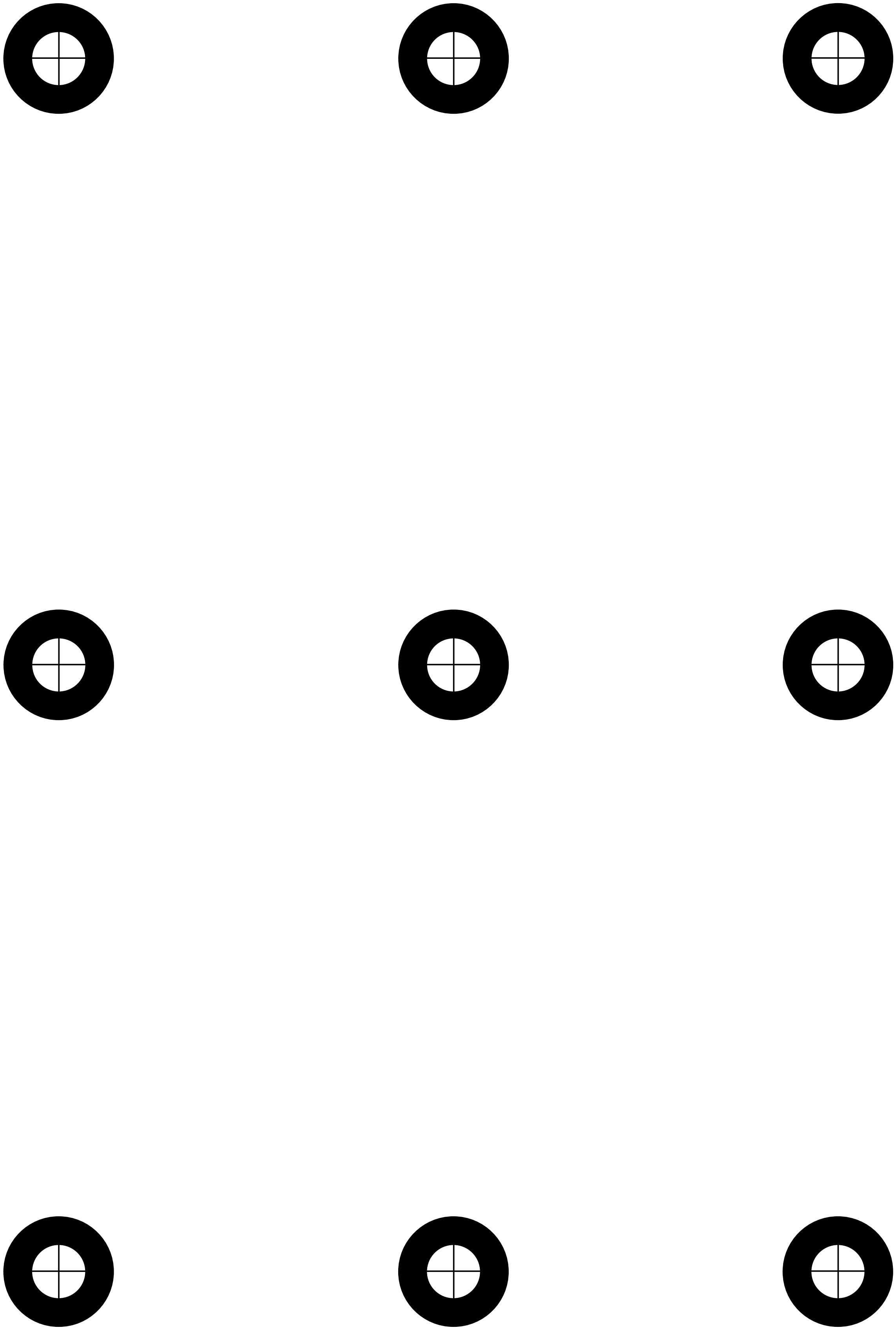
9. Appendix

9.1. Calibration No. marking sheet

This is a marking sheet used in calibration work. Please use this sheet in your calibration work.

Enlarge or reduce it as necessary to match the size of the field of vision of the image.

When changing the size of the sheet, or calibrating in more points, you can photocopy the sheet.



MITSUBISHI ELECTRIC CORPORATION

HEAD OFFICE : MITSUBISHI DENKI BLDG MARUNOUCHI TOKYO 100-8310
NAGOYA WORKS : 1-14, YADA-MINAMI 5, HIGASHI-KU, NAGOYA, JAPAN