MITSUBISHI

Mitsubishi Industrial Robot

CR750/CR751 series controller
CRn-700 series controller

# Tracking Function INSTRUCTION MANUAL

MELFA

BFP-A8664-H

## ⚠ Safety Precautions

Always read the following precautions and separate "Safety Manual" carefully before using robots, and take appropriate action when required.

**⚠ Caution** Teaching work should only be performed by those individuals who have undergone special training.
(The same applies to maintenance work with the robot power ON.)
→ Conduct safety education.

**⚠ Caution** Prepare work regulations indicating robot operation methods and procedures, and measures to be taken when errors occur or when rebooting robots, and observe these rules at all times.
(The same applies to maintenance work with the robot power ON.)
→ Prepare work regulations.

**⚠ Warning** Only perform teaching work after first equipping the controller with a device capable of stopping operation immediately.
(The same applies to maintenance work with the robot power ON.)
→ Equip with an EMERGENCY STOP button.

**⚠ Caution** Notify others when teaching work is being performed by affixing a sign to the START switch, etc.
(The same applies to maintenance work with the robot power ON.)
→ Indicate that teaching work is being performed.

**⚠ Warning** Install fences or enclosures around robots to prevent contact between robots and workers during operation.
→ Install safety fences.

**⚠ Caution** Stipulate a specific signaling method to be used among related workers when starting operation.
→ Operation start signal

**⚠ Caution** As a rule, maintenance work should be performed only after turning OFF the power, and other workers should be notified that maintenance is being performed by affixing a sign to the START switch, etc.
→ Indicate that maintenance work is being performed.

**⚠ Caution** Before starting operation, conduct an inspection of robots, EMERGENCY STOP buttons, and any other related devices to ensure that there are no abnormalities.
→ Inspection before starting operation

The following precautions are taken from the separate "Safety Manual".
Refer to the "Safety Manual" for further details.

| | |
|---|---|
| ⚠ Caution | Use robots in an environment stipulated in the specifications.<br>Failure to observe this may result in decreased reliability or breakdown.<br>(Temperature, humidity, atmosphere, noise environment, etc.) |
| ⚠ Caution | Only transport robots in the manner stipulated.<br>Failure to observe this may result in bodily injury or breakdown if the robot is dropped. |
| ⚠ Caution | Install and use the robot on a secure and stable platform.<br>Positional displacement or vibrations may occur if the robot is unstable. |
| ⚠ Caution | Ensure that cables are kept as far apart from noise sources as possible.<br>Positional displacement or malfunction may occur if in close contact with one another. |
| ⚠ Caution | Do not apply too much force to connectors, or bend cables too much.<br>Failure to observe this may result in contact defects or wire damage. |
| ⚠ Caution | Ensure that the weight of the workpiece, including the hand, does not exceed the rated load or allowable torque.<br>Failure to observe this may result in alarms or breakdown. |
| ⚠ Warning | Attach hands and tools, and grip workpieces securely.<br>Failure to observe this may result in bodily injury or property damage if objects are sent flying or released during operation. |
| ⚠ Warning | Ground the robot and controller properly.<br>Failure to observe this may result in malfunction due to noise, or even electric shock. |
| ⚠ Caution | Always indicate the robot operating status during movement.<br>If there is no indication, operators may approach the robot, potentially leading to incorrect operation. |
| ⚠ Warning | If performing teaching work inside the robot movement range, always ensure complete control over the robot beforehand. Failure to observe this may result in bodily injury or property damage if able to start the robot with external commands. |
| ⚠ Caution | Jog the robot with the speed set as low as possible, and never take your eyes off the robot. Failure to observe this may result in collision with workpieces or surrounding equipment. |
| ⚠ Caution | Always check robot movement in step operation before commencing auto operation following program editing. Failure to observe this may result in collision with surrounding equipment due to programming mistakes, etc. |
| ⚠ Caution | If attempting to open the safety fence door during auto operation, ensure that the door is locked, or that the robot stops automatically. Failure to observe this may result in bodily injury. |

⚠ Caution

Do not perform unauthorized modifications or use maintenance parts other than those stipulated. Failure to observe this may result in breakdown or malfunction.

⚠ Warning

If moving the robot arm by hand from outside the enclosure, never insert hands or fingers in openings. Depending on the robot posture, hands or fingers may become jammed.

⚠ Caution

Do not stop the robot or engage the emergency stop by turning OFF the robot controller main power.
Robot accuracy may be adversely affected if the robot controller main power is turned OFF during auto operation. Furthermore, the robot arm may collide with surrounding equipment if it falls or moves under its own inertia.

⚠ Caution

When rewriting internal robot controller information such as programs or parameters, do not turn OFF the robot controller main power.
If the robot controller main power is turned OFF while rewriting programs or parameters during auto operation, the internal robot controller information may be destroyed.

⚠ Warning

Horizontal multi-joint robots
The hand may drop under its own weight while the robot brake release switch is pressed, and therefore due care should be taken. Failure to observe this may result in collision between the hand and surrounding equipment, or hands or fingers becoming jammed if the hand falls.

⚠ Caution

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

⚠ Caution

Do not look directly at light emitted from the tip of SSCNET III connectors or SSCNET III cables. Eye discomfort may be felt if exposed to the light. (SSCNET III employs a Class 1 or equivalent light source as specified in JISC6802 and IEC60825-1.)

Revision history

| Date of print | Specifications No. | Details of revisions |
|---|---|---|
| 2009-02-10 | BFP-A8664-* | First print |
| 2009-10-23 | BFP-A8664-A | The EC Declaration of Conformity was changed.<br>(Correspond to the EMC directive; 2006/42/EC) |
| 2010-04-30 | BFP-A8664-B | The tracking function is realized to SQ series. |
| 2010-10-18 | BFP-A8664-C | The notes were added about physical encoder number (List 1-1) and No.9 (List 1-2). |
| 2012-03-01 | BFP-A8664-D | CR750/CR751 series controller were added.<br>The note was added to Trk command. |
| 2012-10-19 | BFP-A8664-E | The explanation of vision was changed from MELFA-Vision to In-Sight Explorer for EasyBuilder.<br>Sample program for RH-3S*HR was added.<br>The explanation of parameter "TRPACL" and "TRPDCL" was added.<br>"Troubleshooting" is enhanced. |
| 2013-01-22 | BFP-A8664-F | The statement about trademark registration was added. |
| 2013-05-27 | BFP-A8664-G | “Table 21-3 Connectors: CNENC/CNUSR Pin Assignment” was corrected. |
| 2014-02-13 | BFP-A8664-H | The explanations about Encoder distribution unit (option) were added. |

■Preface

Thank you very much for purchasing Mitsubishi Electric Industrial Robot.
The tracking function allows robots to follow workpieces on a conveyer or transport, line up and process the workpieces without having to stop the conveyer. The conveyor tracking function is the standard function in the controller. It can use only by having the parameter "TRMODE" changed into "1."

Please be sure to read this manual carefully and understand the contents thoroughly before starting to use the equipment in order to make full use of the tracking function.
Within this manual, we have tried to describe all ways in which the equipment can be handled, including non-standard operations, to the greatest extent possible. Please avoid handling the equipment in any way not described in this manual.
Tracking function is installed as standard for the controller, and the function can be used only by changing parameter "TRMODE" from “0" to “1". However, there are different parts in the system configuration and the way of programming in the CR750-Q/CR751-Q, CRnQ-700 series and the CR750-D/CR751-D, CRnD-700 series. Please give the attention that this manual explains these differences between CR750-Q/CR751-Q, CRnQ-700 series and CR750-D/CR751-D, CRnD-700SD series.

Note that this manual is written for the following software version.

CR750-Q/CR751-Q series : Ver. R3 or later
CR750-D/CR751-D series : Ver. S3 or later
CRnQ-700 series : Ver. R1 or later
CRnD-700 series : Ver. P1a or later

# [Contents]

# [Part 1] Overview

## 1. Overview

### 1.1. What is the Tracking Function?

The tracking function allows a robot to follow workpieces moving on a conveyer. With this function, it becomes possible to transport, line up and process workpieces without having to stop the conveyer. It also eliminates the need for mechanical fixtures and so forth required to fix workpiece positions.
The features of this function are described below.

1) It is possible to follow lined-up workpieces moving on a conveyer while working on them (conveyer tracking making use of photo electronic sensors).
2) It is possible to follow workpieces that are not in a line moving on a conveyer while working on them, even in the case of different types of workpieces (vision tracking combined with vision sensors).
3) It is possible to follow changes of movement speed due to automatic calculation of conveyer movement speed.
4) Tracking function can be easily achieved by using Mitsubishi's robot command MELFA-BASIC V.
5) System construction is made easy by use of sample programs.

## 1.2. Applications

Tracking is primarily intended for applications such as the following.

**(1) Transfer of processed food pallets**

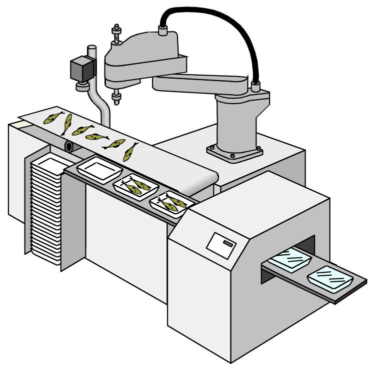


**Figure 1–1 Example of Processed Food Pallet Transfer**

**(2) Lining up parts**

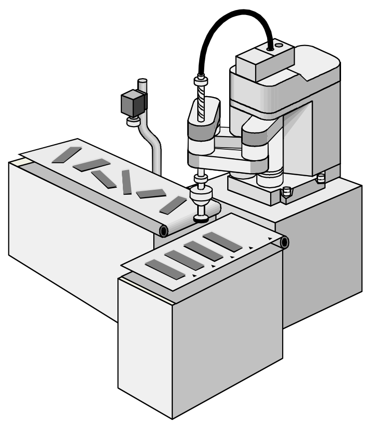


**Figure 1–2 Example of Parts Lineup**

**(3) Assembly of small electrical products**

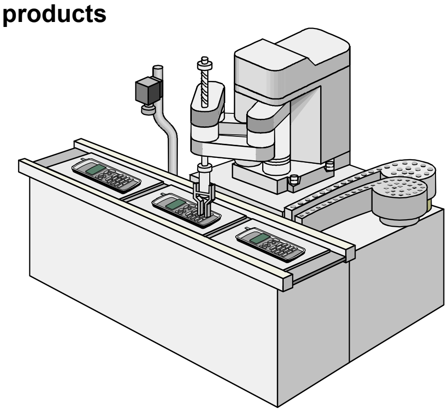


**Figure 1–3 Example of Small Electrical Products Assembly**

## 1.3. Contents of this manual

This manual explains the operation procedure when the customer use conveyer tracking system and vision tracking system using Mitsubishi robot. The robot model are CR750-Q/CR751-Q/CRnQ-700 series and CR750-D/CR751-D/CRnD-700 series, however there are H/W differences. Please read as following.

CR750-Q/CR751-Q/CRnQ-700 series

Part.2 System Configuration CR750-Q/CR751-Q/CRnQ-700 series（**2～6**）
System Configuration/ systemup/ Setting option parts/
Connection to encoder/ Parameter setting

↓

Part.4 Tracking Control（**12～21**）
Sample program/ Teaching/ Automatic operation/ Trouble shooting

CR750-D/CR751-D/CRnD-700Series

Part.3 System Configuration CR750-D/CR751-D/CRnD-700 series（**7～11**）
System Configuration/ systemup/ Setting option parts/
Connection to encoder/ Parameter setting

↓

Part.4 Tracking Control（**12～21**）
Sample program/ Teaching/ Automatic operation/ Trouble shooting

## 1.4. The generic name and abbreviation

**List 1-1 generic name and abbreviation**

| Generic name and abbreviation | Contents |
|---|---|
| Tracking function | The tracking function allows a robot to follow workpieces moving on a conveyer. With this function, it becomes possible to transport line up and process workpieces without having to stop the conveyer. |
| Conveyer tracking | The conveyer tracking allows a robot to follow workpieces lining up on a conveyer. With this function, it becomes possible to transport, process workpieces. |
| Vision tracking | The vision tracking allows a robot to follow workpieces not lining up on a conveyer. With this function, it becomes possible to transport line up and process workpieces. |
| Network vision sensor | The network vision sensor is an option which makes it possible to inspect or find the workpieces by using with robot controller and processing the image. |
| Q173DPX unit | Q173DRX unit is manual pulser input unit for motion controller. At Q series CPU, it is used as intelligent function unit ( occupation 32 points）<br>Each encoder figure can be got by connection with 1 pc the manual pulser machine（MR-HDP01）or 3pcs the incremental encoder. |
| Physical encoder number | Physical encoder numbers a number of the encoder physically allocated according to a certain rule.<br>In the CR750-Q/CR751-Q/CRnQ-700 series, the number is allocated by arranging the encoder connected with Q173DPX unit.<br>The encoder which connected with CH1 of the Q173DPX unit specified for parameter “ENC UNIT1” is the first, the encoder which connected with CH2 is the second and with CH3 is the third.<br>It becomes from 4 to 6 for the Q173DPX unit specified for parameter”ENCUNIT2”.<br>It becomes from 7 to 8 for the Q173DPX unit specified for parameter”ENCUNIT3”.<br>Note) The 3rd set of Q173DPX units can use only the two channels. |
| Logical encoder number | The physical encoder number change to the logical encoder number by parameter “EXTENC”. The purpose of this is to change freely number by the parameter for the encoder physically arranged. This logical encoder number is used with the instruction and the state variable of the robot program. |
| TREN signal | tracking enable signal |

## 1.5. System that can achieve

With the tracking function of CR750-Q/CR751-Q/CRnQ-700 series, CR750-D/CR751-D/CRnD-700 series, the example of the system that can be achieved is shown as following.

**List 1-2 Example of system that can be achieved by the tracking function**

| No. | CR750-Q<br>CR751-Q<br>CRnQ-700 | CR750-D<br>CR751-D<br>CRnD-700 | Example of the system |
|---|---|---|---|
| 1 | ● | ● | When a robot picks the workpieces moving on a conveyer, it is tracking.（transportation） |
| 2 | ● | ● | When a robot places workpieces which taken out from the pallet to a conveyer, it is tracking (transportation). It is also possible to hang workpieces on S character hook that moves the above of the robot. |
| 3 | ● | ● | A robot decorates (processing) the workpieces moving on a conveyer while tracking. |
| 4 | ● | ● | A robot attaches the parts (assembling) with the workpieces moving on a conveyer while tracking. |
| 5 | ● | ● | A robot has the vision sensor (hand eye) and it checks the workpieces moving on a conveyer. (inspection) It also can check and push the button while tracking, not the vision sensor. |
| 6 | ● | ● | When a robot picks the workpieces moving on a conveyer A, the tracking is done and a robot places the workpieces while tracking to marking on a conveyer B. |
| 7 | ● | ● | The tracking is done with an encoder of line driver (differential motion) output type. |
| 8 | ● | (●) Note1) | The tracking is done with an encoder of voltage output/open collector type. |
| 9 | ● | - | In case of multi CPU system, it makes possible to add max 9 pcs Q173DPX units（3 units per 1 CPU）. However, in each CPU, only the two channels can be used at the 3rd set of Q173DPX units. |

Note1) This system requires the Encoder distribution unit. Please refer to the Encoder Distribution Unit Manual (BFP-A3300) for details.

# [Part 2] System Configuration and Setting (CR750-Q/CR751-Q series, CRnQ-700 series)

## 2. System Configuration

### 2.1. Components

#### 2.1.1. Robot controller enclosure products

The product structure of the tracking functional relation enclosed by the robot controller is shown in the Table 2–1.

**Table 2–1 List of Configuration in the tracking functional-related product**

| Product name | Model name | Remark |
|---|---|---|
| Tracking Function INSTRUCTION MANUAL | BFP-A8664 | This manual is included in instruction-manual CD-ROM attached to the product. |
| Sample program | – | Please refer to "12 Sample Robot Programs" for the sample robot program. |

#### 2.1.2. Devices Provided by Customers

When configuring the system, the customers must have certain other devices in addition to this product. The table below shows the minimum list of required devices. Note that different devices are required depending on whether conveyer tracking or vision tracking is used. Please refer to “Table 2–2 List of Devices Provided by Customers (Conveyer Tracking)” and “Table 2–3 List of Devices Provided by Customers (Vision Tracking)” for further details.

**Table 2–2 List of Devices Provided by Customers (Conveyer Tracking)**

| Name of devices to be provided by customers | Model | Quantity | Remark |
|---|---|---|---|
| **Robot part** | | | |
| Teaching pendant | R32TB/R33TB or R56TB/R57TB | 1 | |
| Hand | – | | |
| Hand sensor | – | (1) | Used to confirm that workpieces are gripped correctly. Provide as necessary. |
| Solenoid valve set | See the Remark column | | Different models are used depending on the robot used. Check the robot version and provide as necessary. |
| Hand input cable | | | |
| Air hand interface | 2A-RZ365 or 2A-RZ375 | | (CRnQ-700/CRnD-700 series controller) Provide as necessary. |
| Calibration jig | – | | This is a jig with a sharp tip that is attached to the mechanical interface of the robot arm and used for calibration tasks. It is recommended to use the jig if high precision is required. |
| Encoder pulse unit | Q173DPX | More than 1 | Manual pulser input unit for motion controller **[*]This unit cannot be connected with two or more robot CPU. Please prepare for unit necessary in each robot CPU** |

| Name of devices to be provided by customers | Model | Quantity | Remark |
|---|---|---|---|
| **Conveyer part** | | | |
| Conveyer (with encoder) | – | 1 | Encoder:<br>Voltage output/open collector type<br>Line driver output<br>[Confirmed operation product]<br>Omron encoder (E6B2-CWZ1X-1000 or -2000)<br>Encoder cable (Recommended product) :<br>2D-CBL05/2D-CBL15<br>**[*]The Q173DPX unit supplies 5V power supply to the encoder.** |
| Photo electronic sensor | – | | Used to synchronize tracking |
| 24V power supply | – | | +24 VDC (±10%) : For the Photo electronic sensor |
| Encoder distribution unit | 2F-YZ581 | (1) | The Encoder distribution unit is required when two or more manual pulser input units are connected to the one encoder. Provide this unit as necessary. Refer to the Encoder Distribution Unit Manual (BFP-A3300) for details. |
| **Personal computer part** | | | |
| Personal computer | – | 1 | Please refer to the instruction manual of RT ToolBox2 for the details of the personal computer specifications. |
| RT ToolBox2 (Personal computer support software) | 3D-11C-WINE<br>3D-12C-WINE | | |

**Table 2–3 List of Devices Provided by Customers (Vision Tracking)**

| Name of devices to be provided by customers | Model | Quantity | Remark |
|---|---|---|---|
| **Robot part** | | | |
| Teaching pendant | R32TB/R33TB<br>or<br>R56TB/R57TB | 1 | |
| Hand | – | | |
| Hand sensor | – | (1) | Used to confirm that workpieces are gripped correctly. Provide as necessary. |
| Solenoid valve set | See the Remark column | | Different models are used depending on the robot used. Check the robot version and provide as necessary. |
| Hand input cable | | | |
| Air hand interface | 2A-RZ365 or<br>2A-RZ375 | | (CRnQ-700/CRnD-700 series controller)<br>Provide as necessary. |
| Calibration jig | – | | This is a jig with a sharp tip that is attached to the mechanical interface of the robot arm and used for calibration tasks. It is recommended to use the jig if high precision is required. |
| Encoder pulse unit | Q173DPX | More than 1 | manual pulser input unit for motion controller<br>**【*】This unit cannot be connected with two or more robot CPU. Please prepare for unit necessary in each robot CPU.** |

| Name of devices to be provided by customers | Model | Quantity | Remark |
|---|---|---|---|
| **Conveyer part** | | | |
| Conveyer<br>(with encoder) | – | 1 | Encoder:<br>Voltage output/open collector type<br>Line driver output<br>[Confirmed operation product]<br>Omron encoder (E6B2-CWZ1X-1000 or -2000)<br>Encoder cable (Recommended product) :<br>2D-CBL05/2D-CBL15<br>**[*]The Q173DPX unit supplies 5V power supply to the encoder.** |
| Photo electronic sensor | – | | Used to synchronize tracking |
| 24V power supply | – | | +24 VDC (±10%) :<br>For the Photo electronic sensor and Vision sensor |
| Encoder distribution unit | 2F-YZ581 | (1) | The Encoder distribution unit is required when two or more manual pulser input units are connected to the one encoder. Provide this unit as necessary. Refer to the Encoder Distribution Unit Manual (BFP-A3300) for details. |
| **Vision sensor part** | | | |
| Basic network vision sensor set | 4D-2CG5xxxx-PKG | 1 | See the instruction manual of the network vision sensor for details |
| In-Sight 5000 series<br>In-Sight Micro<br>In-Sight EZ | – | | COGNEX Vision sensor |
| Lens | – | | C-mount lens |
| Lighting installation | – | (1) | Provide as necessary. |
| **Connection part** | | | |
| Hub | – | 1 | |
| Ethernet cable (straight) | – | 2 | Between Robot controller and Hub<br>Between Personal computer and Hub |
| **Personal computer part** | | | |
| Personal computer | – | 1 | Please refer to the instruction manual of RT ToolBox2 or the instruction of the network vision sensor for details of the personal computer specifications. |
| RT ToolBox2<br>(Personal computer support software) | 3D-11C-WINE<br>3D-12C-WINE | | Please refer to the instruction manual of RT ToolBox2 for the details of the personal computer specifications. |

## 2.2. Example of System Configuration

The following figure shows examples of conveyer tracking systems and vision tracking systems.

### 2.2.1. Configuration Example of Conveyer Tracking Systems

The following figure shows a configuration example of a system that recognizes lined-up workpieces on a conveyer passing a photo electronic sensor and follows the workpieces.

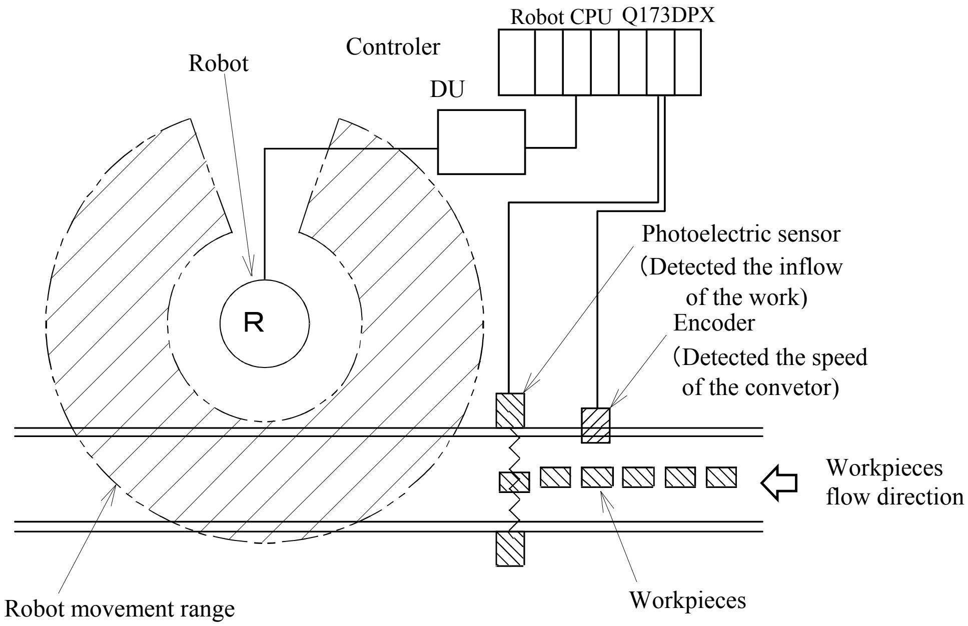



**Figure 2–1 Configuration Example of Conveyer Tracking (Top View)**

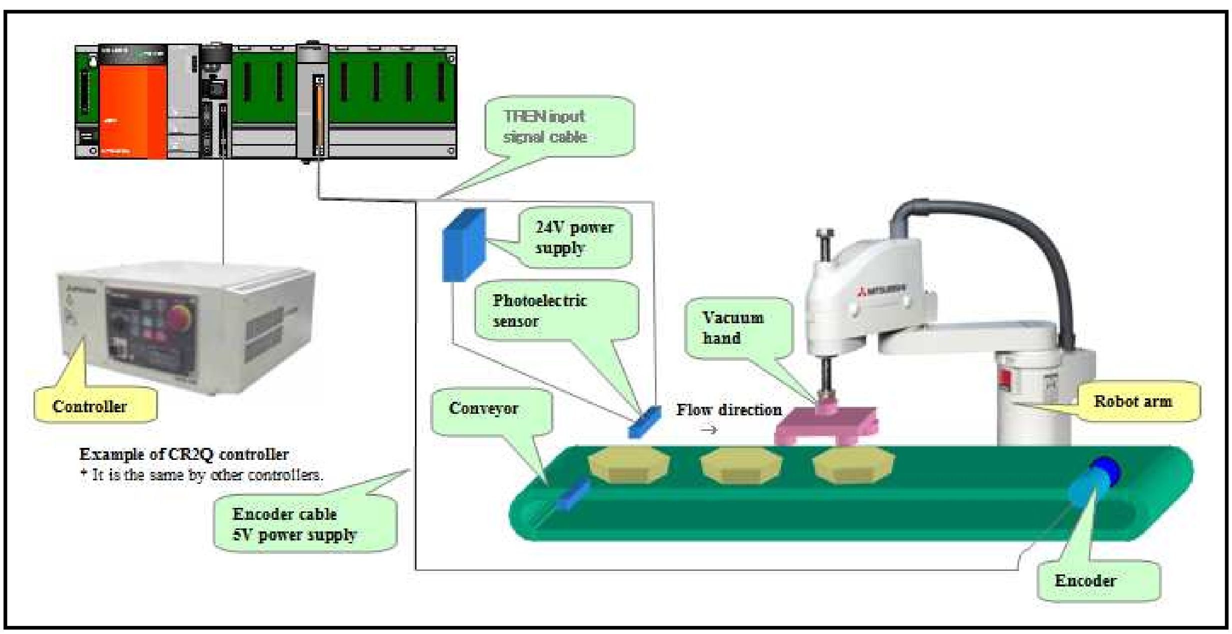



**Figure 2–2 Configuration Example of Conveyer Tracking**

### 2.2.2. Configuration Example of Vision Tracking Systems

The following figure shows a configuration example of a system that recognizes positions of workpieces that are not lined up on a conveyer with a vision sensor and follows the workpieces.

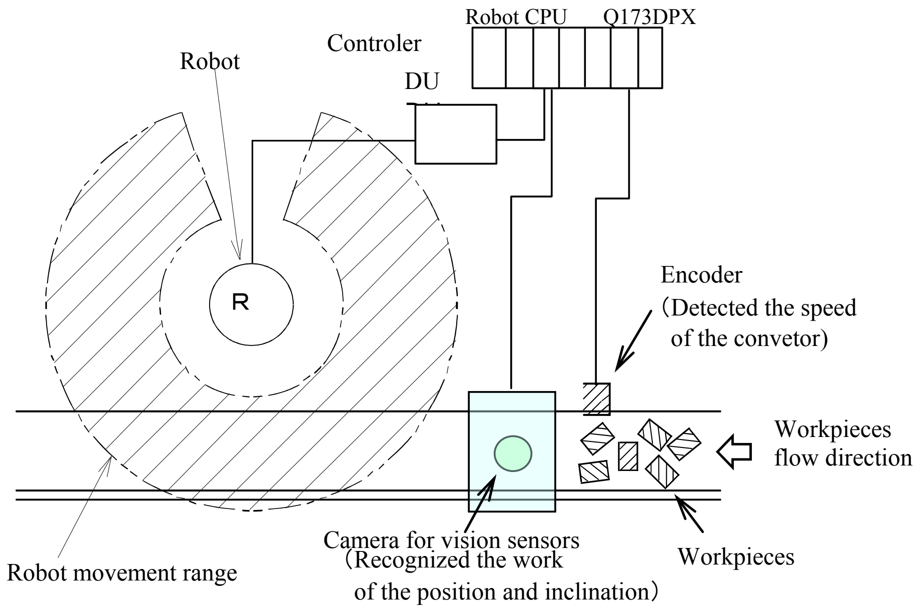



**Figure 2–3 Configuration Example of Vision Tracking (Top View)**

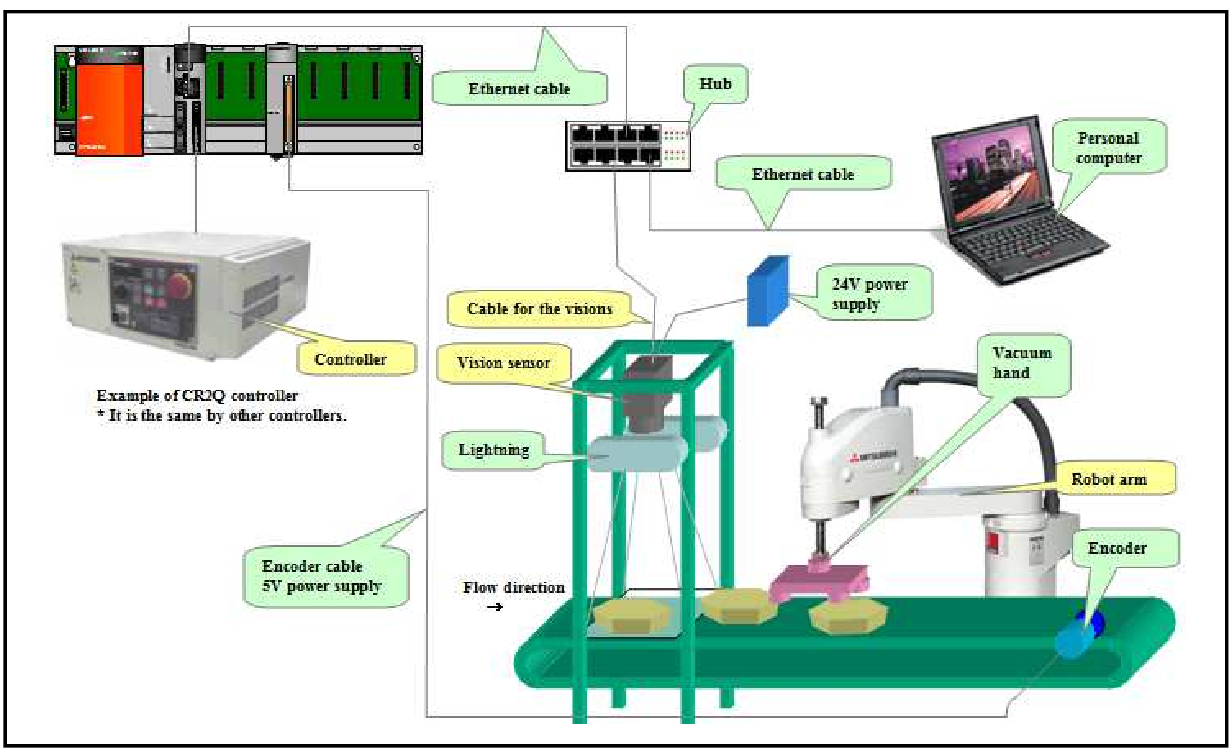



**Figure 2–4 Configuration Example of Vision Tracking**

# 3. Specification

## 3.1. Tracking Specifications and Restriction matter

”Table 3−1 CR750-Q/CR751-Q Series, CRnQ-700 Series Controller Tracking Function Specifications” shows the tracking specifications.
Please refer to “Standard Specifications Manual” for the specifications of the robot arm and controller to be used.

**Table 3−1 CR750-Q/CR751-Q Series, CRnQ-700 Series Controller Tracking Function Specifications**

| Item | | Specification and Restriction matter |
|---|---|---|
| Supported robots (*8) | | RH-SQH series / RV-SQ series<br>RH-FH-Q series / RV-F-Q series |
| Applicable robot controller | | CR1Q / CR2Q / CR3Q controller<br>CR750-Q/CR751-Q series controller |
| Robot program language | | Load commands dedicated for the tracking function |
| Conveyer | Number of conveyer (*6) | Max 8pcs （in case 1pc encoder connect to 1 pc conveyer）<br>Encoder 3 pcs / Q173DPX unit 1pc<br>Q173DPX unit 3pcs / system |
| | Movement Speed（*1） | Possible to support up to 300mm/s (When the robot always transport the workpieces)<br>Possible to support up to 500mm/s when the interval of workpiece is wide. |
| | Encoder | Output aspect ： A、$\overline{A}$、B、$\overline{B}$、Z、$\overline{Z}$<br>Output form ： Voltage output/open collector type（*7）<br>Line driver output（*2）<br>Resolution(pulse/rotation)）： Up to 2000（4000 and 8000 uncorrespond)）<br>Confirmed operation product ：Omuron E6B2-CWZ1X-1000<br>E6B2-CWZ1X-2000 |
| | Encoder cable | Option:<br>2D-CBL05(External I/O cable 5m)<br>2D-CBL15(External I/O cable 15m)<br>Conductor size: AWG#28 |
| Encoder unit | | Only Q173DPX unit<br>[*] Two or more robots CPU cannot share one Q173DPX.<br>One Q173DPX is necessary for each robot CPU. |
| Photoelectronic sensor (*3) | | Used to detect workpieces positions in conveyer tracking.<br>Output signal of sensor need to be connected to TREN terminal of Q173DPX unit.（Input signal number 810～817）<br>And a momentary encoder value that the input enters is preserved in state variable "M_EncL". |
| Vision sensor（*4） | | Mitsubishi’s network vision sensor |
| Precision at handling position (*5) | | Approximately ±2 mm (when the conveyer speed is approximately 300 mm/s)<br>(Photoelectronic sensor recognition accuracy, vision sensor recognition accuracy, robot repeatability accuracy and so on) |

（*1） The specification values in the table should only be considered guidelines. The actual values depend on the specific operation environment, robot model, hand and other factors.
（*2） The line driver output is a data transmission circuit in accordance with RS-422A. It enables the long-distance transmission.
（*3） Please connect the output signal of a photoelectric sensor with the terminal TREN of the Q173DPX unit. This input can be confirmed, by the input signal 810th-817th.
（*4） In the case of vision tracking, please refer to the instruction manual of network vision sensor.
（*5） The precision with which workpieces can be grabbed is different from the repeatability at normal transportation due to the conveyer speed, sensor sensitivity, vision sensor recognition accuracy and other factors. The value above should only be used as a guideline.
（*6） The encoder connected with the third channel of the Q173DPX unit specified for parameter "ENCUNIT3" cannot be used.
（*7） Voltage output/open collector type is an output circuit with two output transistors of NPN and PNP.
（*8） The sample program doesn't correspond to the RV-5 axis robot.

# 4. Operation Procedure

This chapter explains the operation procedure for constructing a conveyer tracking system and a vision tracking system using Mitsubishi Electric industrial robots CR750-Q/CR751-Q series, CRnQ-700 series.

1. Start of operation

⬇

2. Connection of Equipment ···················· Refer to “Chapter 5.”
   It explains Q173DPX (manual pulser input) unit preparation and the connection with the encoder.

⬇

3. Parameter Setting ···················· Refer to “Chapter 6.”
   Chapter 6 explains assignment of signals and setting of parameters related to tracking to allow an external device to control a robot.

⬇

4. Sample Robot Programs ···················· Refer to “Chapter 12.”
   Chapter 12 explains functions related to supplemental sample programs.

⬇

5. Calibration of Conveyer and Robot Coordinate Systems (“A1” program) ········ Refer to “Chapter 13.”
   Chapter 13 explains how to calculate the amount of robot movement per encoder pulse.

⬇

6. Calibration of Vision Coordinate and Robot Coordinate Systems (“B1” program) ··· Refer to “Chapter 14.”
   Chapter 14 explains how to display the position of a workpiece recognized by the vision sensor in the robot coordinate system.

⬇

7. Workpiece Recognition and Teaching (“C1” program) ···················· Refer to “Chapter 15.”
   Chapter 15 explains how to calculate the relationship between the position of a workpiece recognized by the vision sensor and the position at which the robot grabs the workpiece.

⬇

8. Teaching and Setting of Adjustment Variables (“1” Program) ···················· Refer to “Chapter 16.”
   Chapter 16 explains how to make settings such that the robot can follow workpieces moving by on a conveyer and how to teach the robot origin and transportation destination at system start-up.

⬇

9. Automatic Operation ···················· Refer to “Chapter 18.”
   In automatic operation, the robot operates via commands from the conveyer control.

⬇

End of operation

10. Maintenance ···················· Refer to “Chapter 19.”

11. Troubleshooting ···················· Refer to “Chapter 20.”

# 5. Connection of Equipment

This section explains how to connect each of the prepared pieces of equipment.

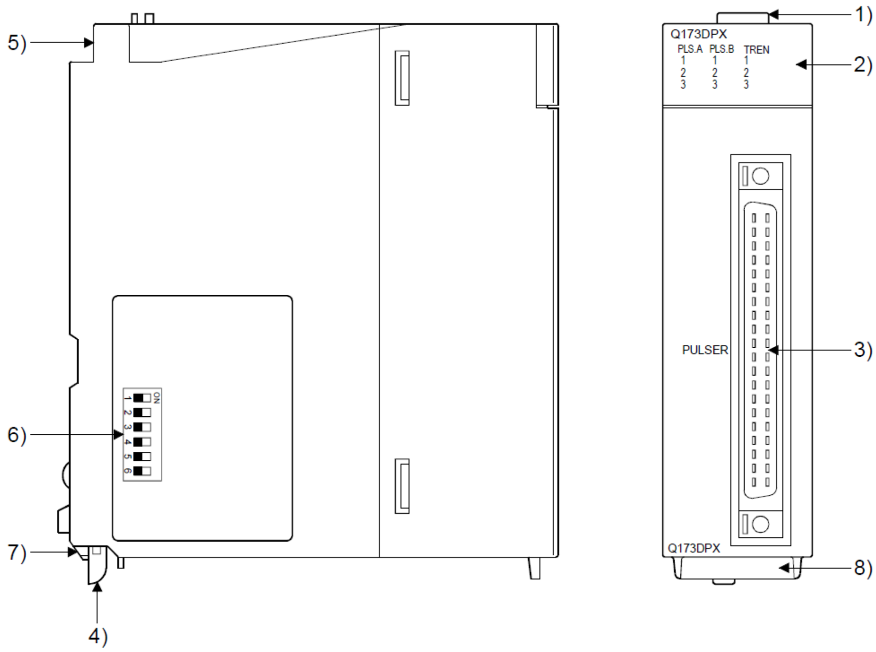
## 5.1. Preparation of Equipment

Prepare equipment by referring to “Table 2–2 List of Devices Provided by Customers (Conveyer Tracking)” to construct a conveyer tracking system and “Table 2–3 List of Devices Provided by Customers (Vision Tracking)” to construct a vision tracking system.

### 5.1.1. Q173DPX（manual pilser input）unit specification

Add Q173DPX unit into PLC base unit （Q3□DB） when the customer use CR750-Q/CR751-Q series, CRnQ-700 series tracking function. Please refer to
"Q173DCPU/Q172DCPU user's manual" about details of this unit.

(1) External and name of Q173DPX unit



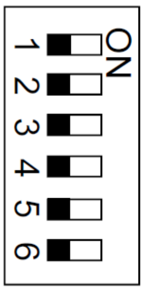

| No. | Name | Application |
|---|---|---|
| 1) | Module fixing hook | Hook used to fix the module to the base unit.<br>(Single-motion installation) |
| 2) | Mode judging LED | Display the input status from the external equipment.<br><table><tr><th>LED</th><th>Details</th></tr><tr><td>PLS.A 1 to 3<br>PLS.B 1 to 3</td><td>Display for input signal status of manual pulse generator/incremental synchronous encoder phases A, B</td></tr><tr><td>TREN 1 to 3</td><td>Display for signal status of tracking enable.</td></tr></table>The manual pulse generator/incremental synchronous encoder phases A, B and tracking enable signal does not turn ON without setting Q173DPX in the system setting. |
| 3) | PULSER connector | Input connector of the Manual pulse generator/Incremental synchronous encoder. |
| 4) | Module mounting lever | Used to install the module to the base unit. |
| 5) | Module fixing screw hole | Hole for the screw used to fix to the base unit<br>(M3×12 screw : Purchase from the other supplier) |

**Figure 5–1 Externals of Q173DPX unit**

(2) Dip switch

By setting the dip switch, the condition of the tracking enable signal is decided.

List **5-1 Item of dip switch**

| No. | Name | Application | |
|---|---|---|---|
| 6) | Dip switches (Note-1)  (Factory default in OFF position) | Dip switch 1<br>Dip switch 2 | Detection setting of TREN1 signal<br>SW1 SW2<br>OFF OFF<br>ON ON<br>ON OFF } TREN is detected at leading edge of TREN signal.<br>OFF ON TREN is detected at trailing edge of TREN signal. |
| | | Dip switch 3<br>Dip switch 4 | Detection setting of TREN2 signal<br>SW3 SW4<br>OFF OFF<br>ON ON<br>ON OFF } TREN is detected at leading edge of TREN signal.<br>OFF ON TREN is detected at trailing edge of TREN signal. |
| | | Dip switch 5<br>Dip switch 6 | Detection setting of TREN3 signal<br>SW5 SW6<br>OFF OFF<br>ON ON<br>ON OFF } TREN is detected at leading edge of TREN signal.<br>OFF ON TREN is detected at trailing edge of TREN signal. |
| 7) | Module fixing projection | Projection used to fix to the base unit. | |
| 8) | Serial number display | Display the serial number described on the rating plate. | |

(Note-1) : The function is different according to the operating system software installed.

## ⚠CAUTION

- Before touching the DIP switches, always touch grounded metal, etc. to discharge static electricity from human body. Failure to do so may cause the module to fail or malfunction.
- Do not directly touch the module's conductive parts and electronic components. Touching them could cause an operation failure or give damage to the module.

(3) Specification of hardware

## (a) Module specifications

| Item | Specifications |
|---|---|
| Number of I/O occupying points | 32 points(I/O allocation: Intelligent, 32 points) |
| Internal current consumption(5VDC)[A] | 0.38 |
| Exterior dimensions [mm(inch)] | 98(H)×27.4(W)×90(D)<br>(3.86(H)×1.08(W)×3.54(D) ) |
| Mass [kg] | 0.15 |

## (b) Tracking enable signal input

| Item | | Specifications |
|---|---|---|
| Number of input points | | Tracking enable signal : 3 points |
| Input method | | Sink/Source type |
| Isolation method | | Photocoupler |
| Rated input voltage | | 12/24VDC |
| Rated input current | | 12VDC 2mA/24VDC 4mA |
| Operating voltage range | | 10.2 to 26.4VDC<br>(12/24VDC +10/ -15%, ripple ratio 5% or less) |
| ON voltage/current | | 10VDC or more/2.0mA or more |
| OFF voltage/current | | 1.8VDC or less/0.18mA or less |
| Input resistance | | Approx. 5.6kΩ |
| Response time | OFF to ON | **7.1ms** |
| | ON to OFF | |
| Common terminal arrangement | | 1 point/common(Common contact: TREN.COM) |
| Indicates to display | | ON indication(LED) |

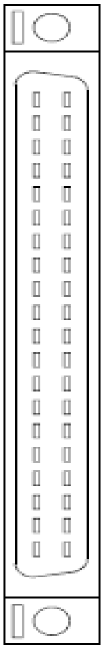
(Note): Functions are different depending on the operating system software installed.

## (c) Manual pulse generator/Incremental synchronous encoder input

| Item | | Specifications |
|---|---|---|
| Number of modules | | 3/module |
| Voltage-output/<br>Open-collector type | High-voltage | 3.0 to 5.25VDC |
| | Low-voltage | 0 to 1.0VDC |
| Differential-output type<br>(26LS31 or equivalent) | High-voltage | 2.0 to 5.25VDC |
| | Low-voltage | 0 to 0.8VDC |
| Input frequency | | Up to 200kpps (After magnification by 4) |
| Applicable types | | Voltage-output type/Open-collector type (5VDC),<br>Recommended product: MR-HDP01,<br>Differential-output type: (26LS31 or equivalent) |
| External connector type | | 40 pin connector |
| Applicable wire size | | 0.3mm$^2$ |
| Applicable connector for the external connection | | A6CON1 (Attachment)<br>A6CON2, A6CON3, A6CON4 (Optional) |
| Cable length | Voltage-output/<br>Open-collector type | 30m (98.43ft.)<br>(Open-collector type: 10m (32.81ft.) ) |
| | Differential-output type | |

(4) Wiring

The pin layout of the Q173DPX PULSER connecter viewed from the unit is shown below.



PULSER connector

| | Pin No. | Signal Name | Pin No. | Signal Name | |
|---|---|---|---|---|---|
| 2) | B20 | HB1 | A20 | HA1 | 2) |
| | B19 | SG | A19 | SG | |
| | B18 | 5V | A18 | HPSEL1 | 1) |
| 3) | B17 | HA1N | A17 | HA1P | 3) |
| 3) | B16 | HB1N | A16 | HB1P | 3) |
| 2) | B15 | HB2 | A15 | HA2 | 2) |
| | B14 | SG | A14 | SG | |
| | B13 | 5V | A13 | HPSEL2 | 1) |
| 3) | B12 | HA2N | A12 | HA2P | 3) |
| 3) | B11 | HB2N | A11 | HB2P | 3) |
| 2) | B10 | HB3 | A10 | HA3 | 2) |
| | B9 | SG | A9 | SG | |
| | B8 | 5V | A8 | HPSEL3 | 1) |
| 3) | B7 | HA3N | A7 | HA3P | 3) |
| 3) | B6 | HB3N | A6 | HB3P | 3) |
| | B5 | No connect | A5 | No connect | |
| | B4 | TREN1 − | A4 | TREN1 + | |
| | B3 | TREN2 − | A3 | TREN2 + | |
| | B2 | TREN3 − | A2 | TREN3 + | |
| 4) | B1 | FG | A1 | FG | 4) |

Applicable connector model name

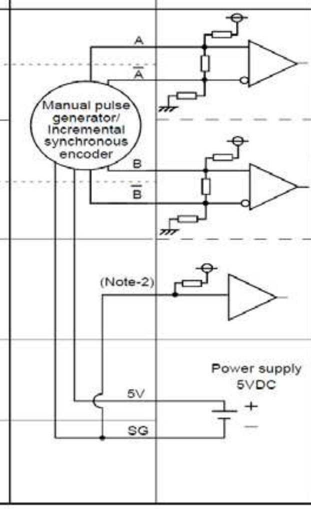
A6CON1 type soldering type connector
FCN-361J040-AU connector (FUJITSU COMPONENT LIMITED)
FCN-360C040-B connector cover
} (Attachment)

A6CON2 type Crimp-contact type connector
A6CON3 type Pressure-displacement type connector
A6CON4 type soldering type connector
} (Optional)

**Figure 5–2 Pin assignment of the PULSER connector**

Interface between PULSER connecter and manual pulse generator (Differential-output type)/ Incremental synchoronous encoder

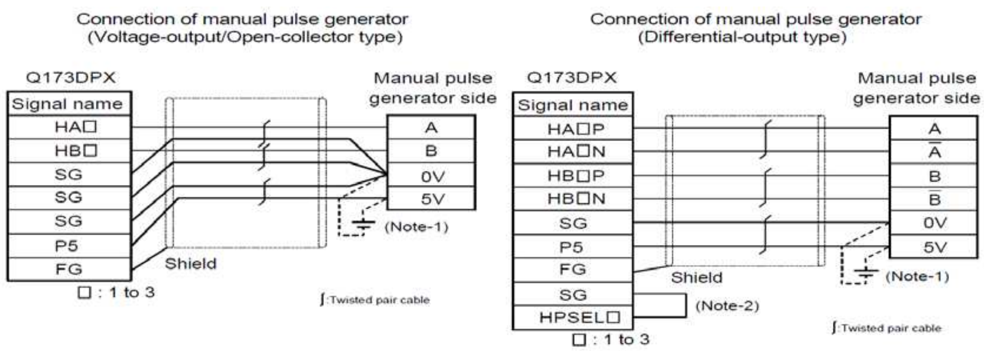
### Interface between Manual pulse generator (Differential-output type)/ Incremental synchronous encoder

| Input or Output | Signal name | | Pin No. PULSER connector Differential-output type 1 | 2 | 3 | Wiring example | Internal circuit | Specification | Description |
|---|---|---|---|---|---|---|---|---|---|
| Input | Manual pulse generator, phase A | A+ HA□P | A17 | A12 | A7 |  | | • Rated input voltage 5.5VDC or less<br>• HIGH level 2.0 to 5.25VDC<br>• LOW level 0.8VDC or less<br>• 26LS31 or equivalent | For connection manual pulse generator Phases A, B<br>• Pulse width 20μs or more; 5μs or more, 5μs or more (Duty ratio: 50%±25%)<br>• Leading edge, Trailing edge time ••• 1μs or less.<br>• Phase difference: Phase A, Phase B, 2.5μs or more<br>(1) Positioning address increases if Phase A leads Phase B.<br>(2) Positioning address decreases if Phase B leads Phase A. |
| | | A− HA□N | B17 | B12 | B7 | | | | |
| | Manual pulse generator, phase B | B+ HB□P | A16 | A11 | A6 | | | | |
| | | B− HB□N | B16 | B11 | B6 | | | | |
| | Select type signal HPSEL□ | | A18 | A13 | A8 | | | | |
| Power supply | P5(Note-1) | | B18 | B13 | B8 | | | | |
| | SG | | A19<br>B19 | A14<br>B14 | A9<br>B9 | | | | |



(Note-1) : The 5V(P5)DC power supply from the Q173DPX must not be connected if a separated power supply is used as the Manual pulse generator/Incremental synchronous encoder power supply. Use a 5V stabilized power supply as a separated power supply. Any other power supply may cause a failure.

(Note-2) : Connect HPSEL□ to the SG terminal if the manual pulse generator (differential-output type) /incremental synchronous encoder is used.




(Note-1) : The 5V(P5)DC power supply from the Q173DPX must not be connected if a separated power supply is used as the Manual pulse generator/Incremental synchronous encoder power supply.
Use a 5V stabilized power supply as a separated power supply. Any other power supply may cause a failure.

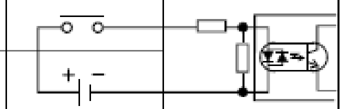
(Note-2) : Connect HPSEL□ to the SG terminal if the manual pulse generator (differential-output type)/incremental synchronous encoder is used.

**Figure 5–3 Wiring connection with rotary encoder**

As above image, because DC5V voltage is output from Q173DPX unit, it makes possible to supply 5V from Q173DPX unit to rotary encoder. When 24V encoder type of power supply is used, it makes possible to use 24V output from PLC power unit.

The interface between tracking enable signal is shown follow.
This signal is used for input signal when the photoelectronic sensor is used to find workpieces so please connect output signal of photoelectronic sensor.

Interface between tracking enable signal

<table>
<tr><th rowspan="3">Input or Output</th><th rowspan="3" colspan="2">Signal name</th><th colspan="3">Pin No.</th><th rowspan="3">Wiring example</th><th rowspan="3">Internal circuit</th><th rowspan="3">Specification</th><th rowspan="3">Description</th></tr>
<tr><th colspan="3">PULSER connector</th></tr>
<tr><th>1</th><th>2</th><th>3</th></tr>
<tr><td rowspan="2">Input</td><td rowspan="2">Tracking enable</td><td>TREN□+</td><td>A4</td><td>A3</td><td>A2</td><td rowspan="2" colspan="2"></td><td rowspan="2"></td><td rowspan="2">Tracking enable signal input.</td></tr>
<tr><td>TREN□−</td><td>B4</td><td>B3</td><td>B2</td></tr>
</table>

(Note) : As for the connection to tracking enable (TREN□+, TREN□–), both "+" and "–" are possible.

**Figure 5–4 Connected composition of tracking enable signal**

## ⚠CAUTION

- If a separate power supply is used as the manual pulse generator/incremental synchronous encoder power supply, use a 5V stabilized power supply. Any other power supply may cause a failure.
- Always wire the cables when power is off. Not doing so may damage the circuit of modules.
- Wire the cable correctly. Wrong wiring may damage the internal circuit.

## 5.2. Connection of Equipment

The connection with each equipments is explained as follow.

### 5.2.1. Connection of Unit

Q173DPX unit is connected to base unit（Q3□DB）or Q6□B increase base unit.

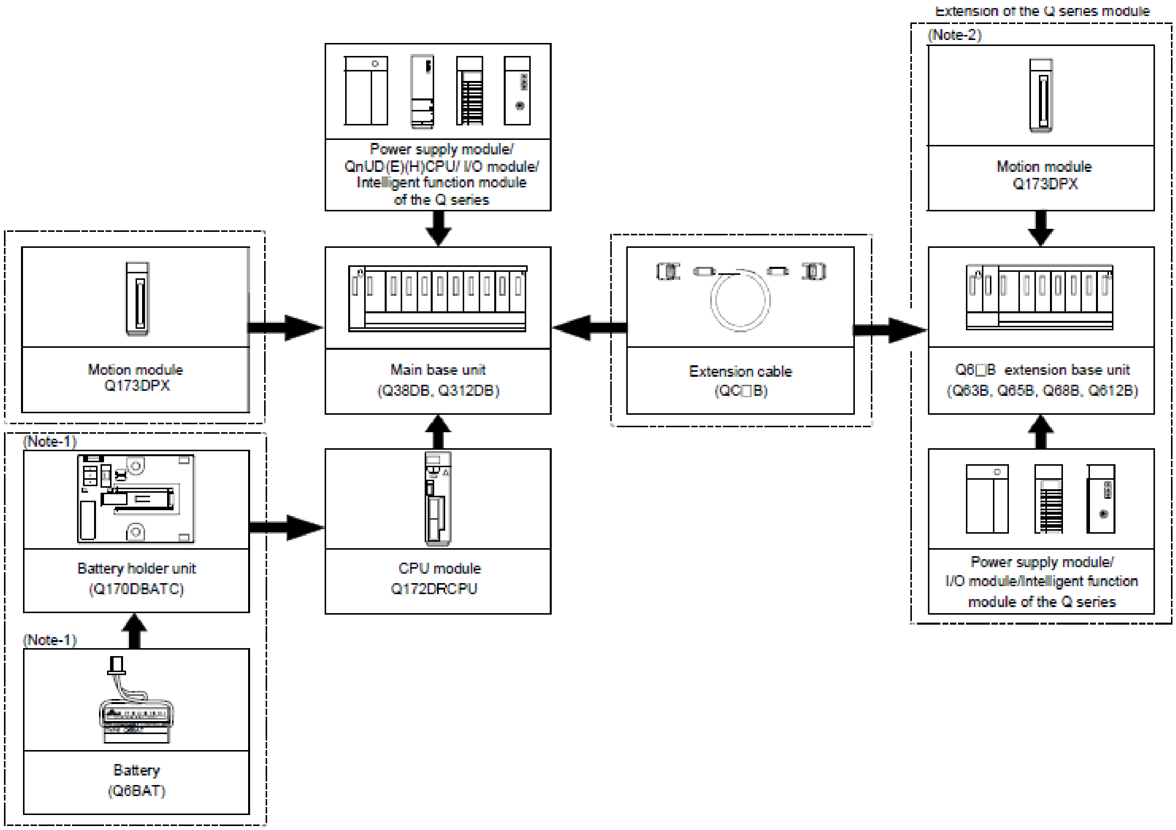



**Figure 5–5 Connected composition of units**

The connection robot system with Q173DPX unit is shown as follow.

List **5-2 Spec list of Q173DPX in robot system**

| Item | Spec and Remark |
|---|---|
| Encoder | Incremental synchronous encoder 3pcs |
| Tracking input points | 3points<br>Three points can be input to ± TREN1-3 in the pin assignment of the unit.<br>When the input of a photoelectric sensor is put, this input is used. |
| Slot that can be connected | Connection with the base unit Possible to install I/O slot since 3<br>（Impossible to install CPU slot or I/O slot 0 to 2）<br>Connection with additional base unit Possible to install all slots. |
| Robot CPU unit that can be managed | Q173DPX unit 3pcs |
| Robot CPU encoder that can be managed | Max 8pcs<br>Impossible to use the third channel of the third Q173DPX unit.<br>And impossible to use the encoder connected with the third channel of the unit specified for parameter「ENCUNIT3」. |

### 5.2.2. Connection with encoder for conveyer and encoder cable

E6B2-CWZ1X (made by Omron) is used, and the wiring for the encoder and the encoder cable for the conveyer is shown in "Figure 5-2 the encoder for the conveyer and the wiring diagram of the encoder cable".
The encoder for the conveyer up to 3 pcs can be connected per Q173DP unit 1pc. The signal cabels needed in case of the connection are power supply （＋、－） and encoder A,B,Z each ＋、－, total 8 cables. Please refer to the manual of the encoder, please connect signal cable correctly. Also please ground shield line (SLD).

| ⚠CAUTION |
| --- |
| ● When fabricating the encoder cable, do not make incorrect connection. Wrong connection will cause runaway or explosion. |

Pin assignment of the
PULSER connector

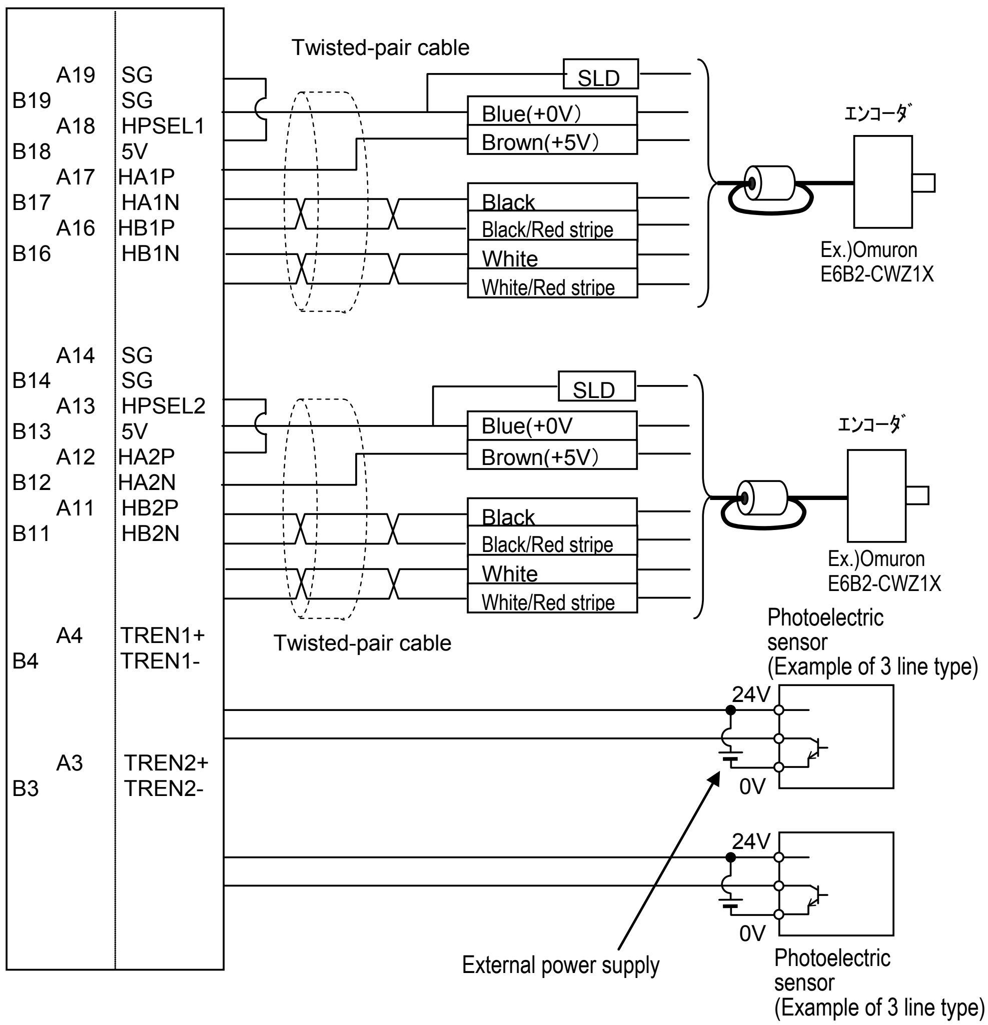



**Figure 5–6 the encoder for the conveyer and the wiring diagram of the encoder cable**

※Please refer to "Figure 5–2 Pin assignment of the PULSER connector" with the pin crack of the PULSER connector that arrives at the unit.

The wiring example by the thing is shown below.
(Please note that the connector shape is different depending on the controller. )

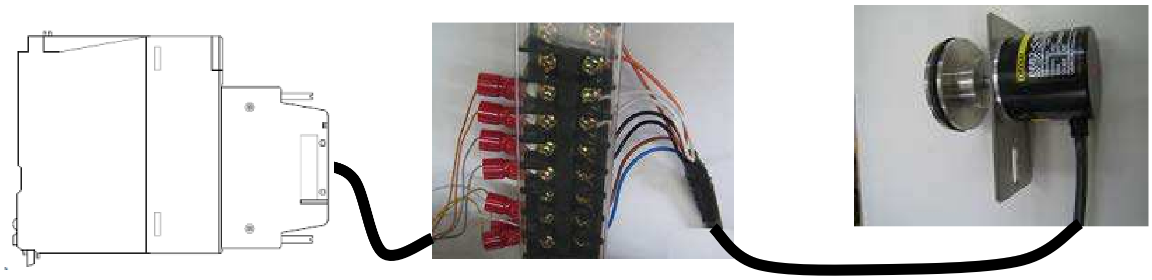


**Figure 5–7 Wiring example (CR75x-Q/ CRnQ-700 series controller)**

### 5.2.3. Connection of Photoelectronic Sensor

If a photoelectronic sensor is used for detection of workpieces, connect the output signal of the photoelectronic sensor to a tracking enable signal of the Q173DPX unit.
In this section, a connection example where the photoelectronic sensor signal is connected to the tracking enable signal is shown in “

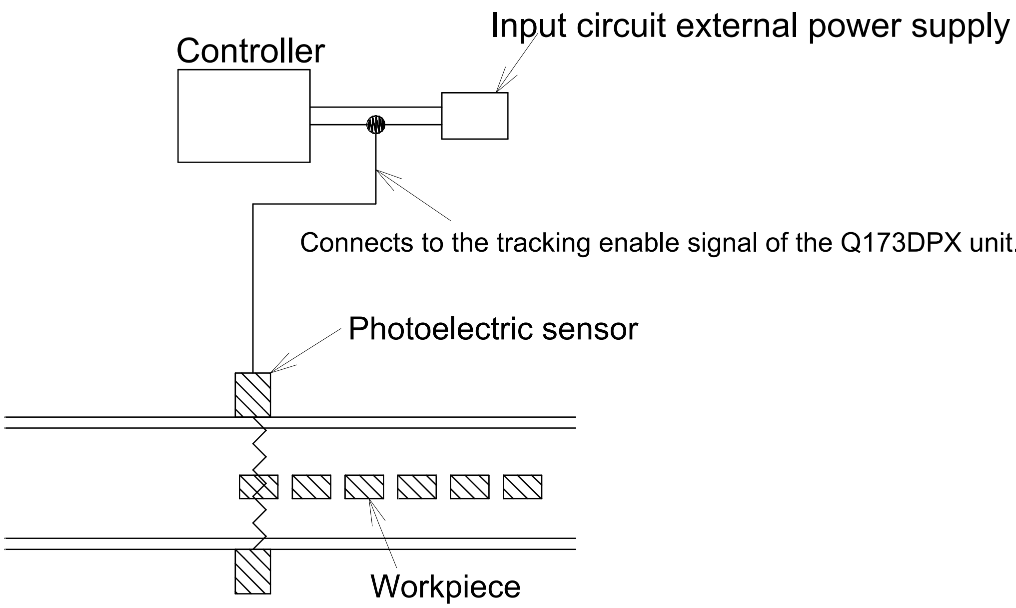



**Figure 5–8 Photoelectronic Sensor Arrangement Example**

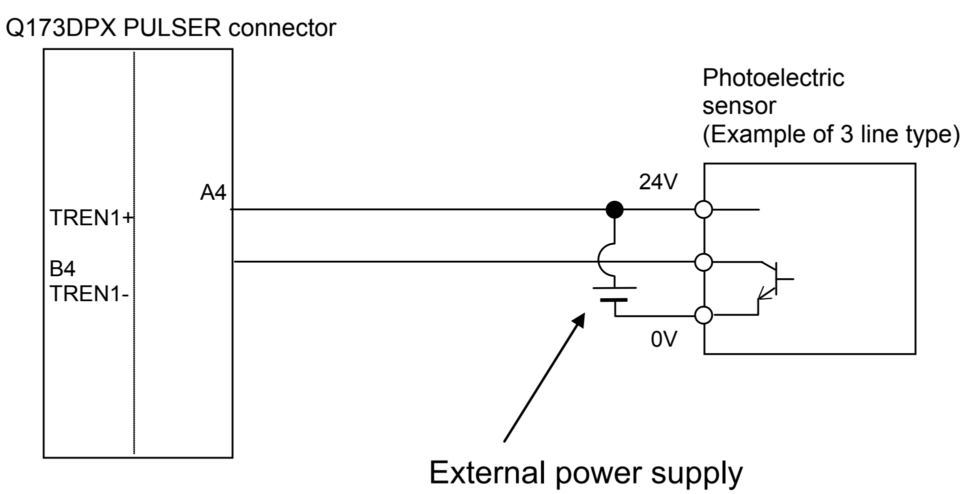



**Figure 5–9 Photoelectronic Sensor Connection Example (6th General Input Signal is Used)**

Note) The external power supply and photoelectric sensor must be prepared

The tracking enable signal is connected to the robot input signal as follows.

List **5-3 List with signal crack of tracking enable signal (TREN)**

| Encoder physics number | Connection channel CR750-Q/CR751-Q series, CRnQ-700 series | Robot Input signal number |
|---|---|---|
| 1 | 1st channel of Parameter ENCUNIT1 | 810 |
| 2 | 2nd channel | 811 |
| 3 | 3rd channel | 812 |
| 4 | 1st channel of Parameter ENCUNIT2 | 813 |
| 5 | 2nd channel | 814 |
| 6 | 3rd channel | 815 |
| 7 | 1st channel of Parameter ENCUNIT3 | 816 |
| 8 | 2nd channel | 817 |

# 6. Parameter Setting

This chapter explains how to set dedicated input/output signals that play the role of interface between a robot and an external device (e.g., a Programmable Logic Controller) and parameters related to the tracking function. Please refer to “Detailed Explanations of Functions and Operations” for how to set the parameters.

## 6.1. Dedicated Input/Output Parameters

“Table 11–1 List of Dedicated Input/Output Parameters” lists the setting items of dedicated input/output parameters used to operate the robot via instructions from an external device. Set the signal numbers according to your system using the setting values in the table as reference. **It is not necessary to set these parameters if the robot operates by itself, rather than via instructions from an external device.**

**Table 6–1 List of Dedicated Input/Output Parameters**

| Input name/output name (parameter name) | Explanation | Setting Example (*1) |
|---|---|---|
| Stop/pausing (**STOP**) or (**STOP2**) | Input: Stop a program<br>Output: Output program standby status | **10000 , -1** |
| Servo OFF/servo ON disabled (**SRVOFF**) | Input: Turn the servo off<br>Output: Output servo ON disabled status | **10011 , -1** |
| Error reset/error occurring (**ERRRESET**) | Input: Cancel error status<br>Output: Output error status | **10009 , -1** |
| Start/operating (**START**) | Input: Start automatic operation<br>Output: Output program running status | **10006 , 1** |
| Servo ON/turning servo ON (**SRVON**) | Input: Turn the servo on<br>Output: Output servo on status | **10010 , 0** |
| Operation right/operation right enabled (**IOENA**) | Input: Enable/disable operation right of external signal control<br>Output: Output external signal control operation enabled status | **10005 , -1** |
| Program reset/program selectable (**SLOTINIT**) | Input: Initiate a program. The program execution returns to the first step.<br>Output: Output a status where program No. can be changed | **10008 , -1** |
| General output signal reset (**OUTRESET**) | Input: Reset a general output signal | **10015 , -1** |
| User specification area 1 (**USRAREA**) | Output an indication that the robot is in an area specified by a user<br>Set the start number and end number | **10064, 10071** |

**(*1) “-1” in the Setting value column means “not set.”**

## 6.2. Operation Parameters

“Table 11–2 List of Operation Parameter” lists the setting items of parameters required to operate the robot at the optimal acceleration/deceleration.

**Table 6–2 List of Operation Parameter**

| Parameter name | Explanation | Reference value |
|---|---|---|
| Optimal acceleration/ deceleration hand data (**HANDDAT1**) | Specify hand weight and so on to make settings that allow optimal acceleration/deceleration operations.<br>For example, if the hand weighs 3 kg, changing the weight setting value from 10 kg to 3 kg makes the robot movement faster.<br>(Hand weight (kg), size (mm) X, Y, Z, gravity (mm) X, Y, Z) | (3,0,0,0,0,0,0)<br>The setting values are different for each robot model.<br>**Use these values as reference only**. |
| Optimal acceleration/ deceleration workpiece data (**WRKDAT1**) | Specify workpiece weight and so on to make settings that allow optimum acceleration/deceleration operations.<br>If a workpiece is grabbed via the HClose instruction, the acceleration/deceleration becomes slower. If a workpiece is released via the HOpen instruction, acceleration/deceleration becomes faster.<br>(Workpiece weight (kg), size (mm) X, Y, Z, gravity (mm) X, Y, Z) | (1,0,0,0,0,0,0)<br>The setting values are different for each robot model.<br>**Use these values as reference only.** |

## 6.3. Tracking Parameter Setting

Specify to which channel of the encoder connector (CNENC) an encoder of conveyer is connected.
“Table 6-3 Tracking Parameter Setting” lists the parameters to be set. Other parameters are shown in “Table 21-1 List of Tracking Parameters”, make settings as required.

### 6.3.1. Robot Parameter Setting

After the installation of Q173DPX module and connection with the encoder are complete, use the following steps to establish robot CPU parameters.
(1) Using parameter ENCUNT* (*=1~3), designate the slot in which Q173DPX module under the control of robot CPU is installed.
(2) Change the number of the incremental synchronization encoder being physically wired into a logic number, using parameter EXTENC.

**Table 6-3 Tracking Parameter Setting**

| Parameter | Parameter name | Number of elements | Explanation | Value set at factory shipping |
|---|---|---|---|---|
| Tracking mode | TRMODE | 1 integer | Enable the tracking function<br>Please set it to “1" when you use the tracking function.<br>0: Disable/1: Enable | 0 |
| first Q173DPX | ENCUNIT1 | | The base unit-number of the first Q173DPX unit (element 1) that robot CPU manages and slot number (element 2) are set.<br>【Element １】<br>－1：No connection<br>０：Basic base unit<br>1～７：Increase base unit<br>【Element ２】<br>０～１1： I/O Slot number<br>*** This parameter is valid in the following software versions.**<br>**・CRnQ-700 series: Ver. R1 or later** | -1,0 |
| Second Q173DPX | ENCUNIT2 | | The base unit-number of the second Q173DPX unit (element 1) that robot CPU manages and slot number (element 2) are set.<br>【Element １】<br>－１：No connection<br>０：Basic base unit<br>～７：Increase base unit<br>【Element ２】<br>０～１１ ： I/O slot number<br>*** This parameter is valid in the following software versions.**<br>**・CRnQ-700 series: Ver. R1 or later** | -1,0 |
| third Q173DPX | ENCUNIT3 | | The base unit-number of the third Q173DPX unit (element 1) that robot CPU manages and slot number (element 2) are set.<br>【Element １】<br>－１：No connection<br>０：Basic base unit<br>～７：Increase base unit<br>【Element ２】<br>０～１１ ： I/O slot number<br>*** This parameter is valid in the following software versions.**<br>**・CRnQ-700 series: Ver. R1 or later** | -1,0 |

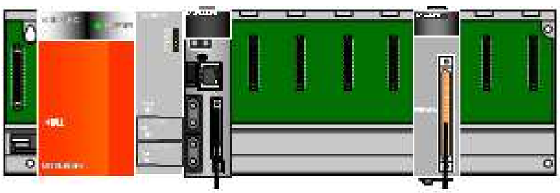
<table>
<tr><th>Parameter</th><th>Parameter name</th><th>Number of element</th><th>Explanation</th><th>Value set at factory shipping</th></tr>
<tr><td>Encoder number allocation</td><td>EXTENC</td><td>8 integers</td><td>Set connection destinations on the connector for encoder numbers 1 to 8.<br>Parameter elements correspond to encoder number 1, encoder number 2 … encoder number 8 from the left.<br>Setting value is iuput encoder physics number from below list.<br>In case of CR750-D/CR751-D and CRnD-700 series, CH1 and CH2 of slot 1 to 3 are reservation. At present, it cannot be used.<br>【In case of CR750-Q/CR751-Q, CRnQ-700 series】
<table>
<tr><th>Encoder physics number</th><th>Connection channel (CR750-Q/CR751-Q, CRnQ-700 series)</th></tr>
<tr><td>1</td><td>1st channel of Parameter ENCUNIT1</td></tr>
<tr><td>2</td><td>2nd channel</td></tr>
<tr><td>3</td><td>3rd channel</td></tr>
<tr><td>4</td><td>1st channel of Parameter ENCUNIT2</td></tr>
<tr><td>5</td><td>2nd channel</td></tr>
<tr><td>6</td><td>3rd channel</td></tr>
<tr><td>7</td><td>1st channel of Parameter ENCUNIT3</td></tr>
<tr><td>8</td><td>2nd channel</td></tr>
</table>
It is convenient to check the status variable “M_Enc” when determining the setting value of the “EXTENC” parameter.<br>Please refer to "19.1.2 List of Robot Status Variables” for the explanation of state variable “M_Enc".<br>Please refer to “Detailed Explanations of Functions and Operations” for how to check the status variable.</td><td>1,2,3,4,<br>5,6,7,8</td></tr>
<tr><td>Tracking Workpiece judgement distance</td><td>TRCWDST</td><td>1 integer</td><td>Distance to judge that the same workpiece is being tracked (mm)<br>The sensor reacts many times when the workpiece with the ruggedness passes the sensor. Then, the robot controller judged that one workpiece is two or more pieces.<br>The sensor between values [mm] set to this parameter does not react after turning on the sensor.<br>To set the measure of workpieces flow is recommended.</td><td>5.00</td></tr>
</table>

### 6.3.2. Sequencer CPU Parameter Setting

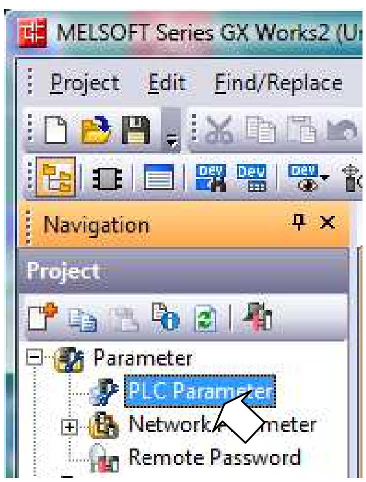
It is necessary to set multi CPU related parameters for both the sequencer CPU and robot CPU In order to use the sequencer link function.

a) Multiple CPU setting : Set the number of CPU units.
b) I/O assignment : Select I/O units and/or Intelligent units.
c) Control PLC setting : Set the CPU Unit numbers which control the Q173DPX unit.

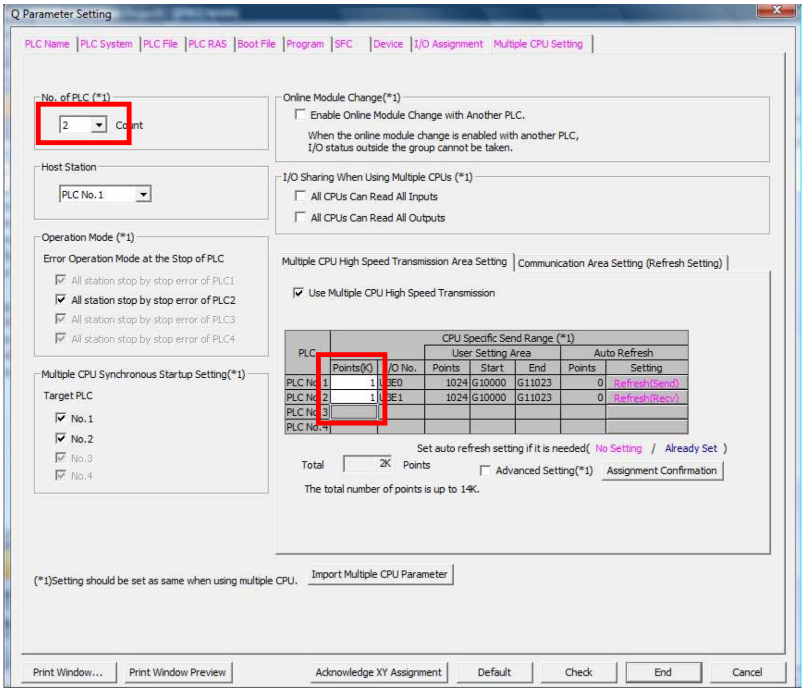
The setting procedure of the parameter is as below.
The following explanation assumes the case that attached Q173DPX unit to the fifth slot of baseboard.



(1) Execute the GX Works2 and select the project file.
(2) Double-click the “PLC Parameter”, then the “Q Parameter Setting” is displayd.



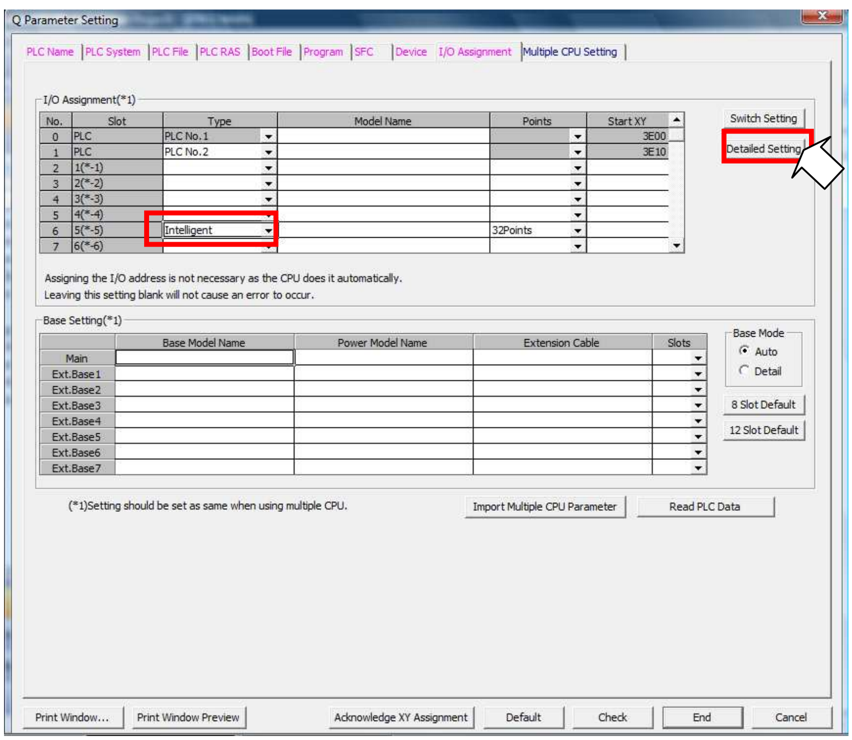
(3) Double-click the “Multiple CPU Setting”



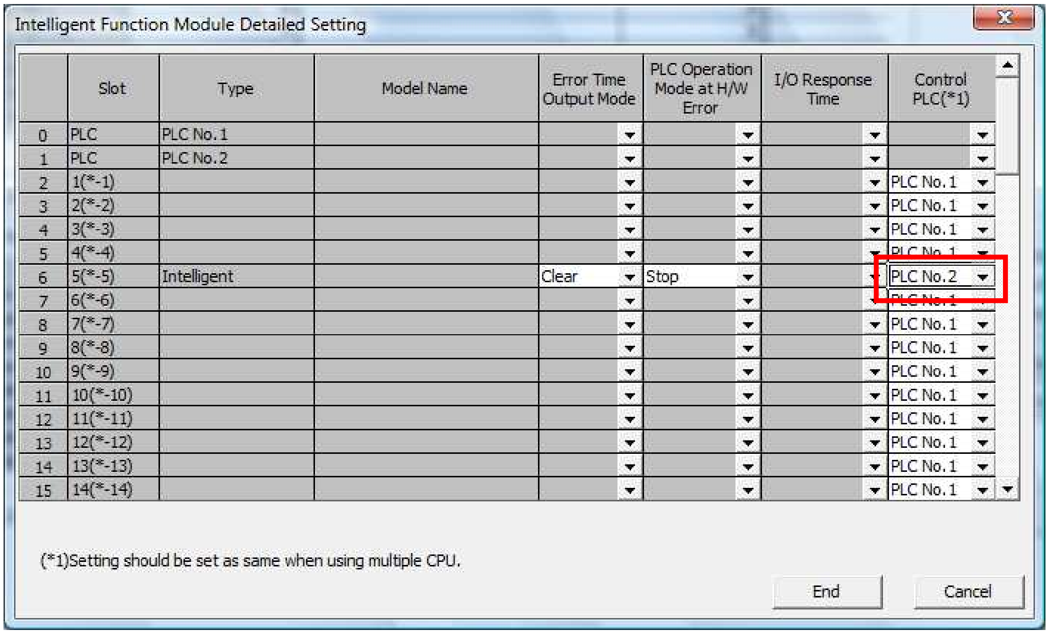

Set the number of CPU and this system area size (K Points)

(4) Double-click the “I/O assignment”

When Q173DPX unit is attached to fifth slot, change the type of slot 5 to the “Intelligent”.



(5) Click the “Detailed Setting” button.




Because the robot CPU manages the Q173DPX unit, change the Control PLC of slot 5 to the “PLC No.2” (Robot CPU).

6) Click the “END” button.
The Parameters are memorized into the sequencer CPU.

**The following work is confirming the operation of the robot by the sample program.**
**Please confirm "[Part 4] Tracking Control".**

# [Part 3] System Configuration and Setting (CR750-D/CR751-D series, CRnD-700 series)

## 7. System Configuration

### 7.1. Components

#### 7.1.1. Robot controller enclosure products

The product structure of the tracking functional relation enclosed by the robot controller is shown in the Table 2–1.

**Table 7–1 List of Configuration in the tracking functional-related product**

| Product name | Model name | Remark |
|---|---|---|
| Tracking Function INSTRUCTION MANUAL | BFP-A8664 | This manual is included in instruction-manual CD-ROM attached to the product. |
| Sample program | – | Please refer to "12 Sample Robot Programs" for the sample robot program. |

#### 7.1.2. Devices Provided by Customers

When configuring the system, the customers must have certain other devices in addition to this product. The table below shows the minimum list of required devices. Note that different devices are required depending on whether conveyer tracking or vision tracking is used. Please refer to “Table 2–2 List of Devices Provided by Customers (Conveyer Tracking)” and “Table 2–3 List of Devices Provided by Customers (Vision Tracking)” for further details.

**Table 7–2 List of Devices Provided by Customers (Conveyer Tracking)**

| Name of devices to be provided by customers | Model | Quantity | Remark |
|---|---|---|---|
| **Robot part** | | | |
| Teaching pendant | R32TB/R33TB or R56TB/R57TB | 1 | |
| Hand | – | | |
| Hand sensor | – | (1) | Used to confirm that workpieces are gripped correctly. Provide as necessary. |
| Solenoid valve set | See the Remark column | | Different models are used depending on the robot used. Check the robot version and provide as necessary. |
| Hand input cable | | | |
| Air hand interface | 2A-RZ365 or 2A-RZ375 | | (CRnQ-700/CRnD-700 series controller)<br>Provide as necessary. |
| Calibration jig | – | | This is a jig with a sharp tip that is attached to the mechanical interface of the robot arm and used for calibration tasks. It is recommended to use the jig if high precision is required. |
| **Conveyer part** | | | |
| Conveyer (with encoder) | – | 1 | Encoder:<br>Line driver output<br>[Confirmed operation product]<br>Omron encoder (E6B2-CWZ1X-1000 or -2000)<br>Encoder cable. Twisted-pair cable with the shield.<br>(CRnD-700 series controller)<br>Recommended connector for encoder input terminal:<br>10120-3000PE plug made by 3M<br>10320-52F0-008 shell made by 3M |
| 5V power supply | – | | +5 VDC (±10%) : For the encoder |
| Photoelectronic sensor | – | | Used to synchronize tracking |
| 24V power supply | – | | +24 VDC (±10%) : For the Photoelectronic sensor |

| Name of devices to be provided by customers | Model | Quantity | Remark |
|---|---|---|---|
| Encoder distribution unit | 2F-YZ581 | (1) | The Encoder distribution unit is required when two or more robot controllers are connected to the one encoder. Provide this unit as necessary.<br>If the Encoder distribution unit is used, a 5V power source for the encoder is not necessary.<br>Refer to the Encoder Distribution Unit Manual (BFP-A3300) for details. |
| **Personal computer part** | | | |
| Personal computer | – | 1 | Please refer to the instruction manual of RT ToolBox2 for the details of the personal computer specifications. |
| RT ToolBox2 (Personal computer support software) | 3D-11C-WINE<br>3D-12C-WINE | | |

**Table 7–3 List of Devices Provided by Customers (Vision Tracking)**

| Name of devices to be provided by customers | Model | Quantity | Remark |
|---|---|---|---|
| **Robot part** | | | |
| Teaching pendant | R32TB/R33TB<br>or<br>R56TB/R57TB | 1 | |
| Hand | – | | |
| Hand sensor | – | (1) | Used to confirm that workpieces are gripped correctly. Provide as necessary. |
| Solenoid valve set | See the Remark column | | Different models are used depending on the robot used. Check the robot version and provide as necessary. |
| Hand input cable | | | |
| Air hand interface | 2A-RZ365 or<br>2A-RZ375 | | (CRnQ-700/CRnD-700 series controller)<br>Provide as necessary. |
| Calibration jig | – | | This is a jig with a sharp tip that is attached to the mechanical interface of the robot arm and used for calibration tasks. It is recommended to use the jig if high precision is required. |
| **Conveyer part** | | | |
| Conveyer (with encoder) | – | 1 | Encoder:<br>Line driver output<br>[Confirmed operation product]<br>Omron encoder (E6B2-CWZ1X-1000 or -2000)<br>Encoder cable. Twisted-pair cable with the shield.<br>(CRnD-700 series controller)<br>Recommended connector for encoder input terminal:<br>10120-3000PE plug made by 3M<br>10320-52F0-008 shell made by 3M |
| 5V power supply | – | | +5 VDC (±10%) : For the encoder |
| Photoelectronic sensor | – | | Used to synchronize tracking |
| 24V power supply | – | | +24 VDC (±10%) :<br>For the Photoelectronic sensor and Vision sensor |
| Encoder distribution unit | 2F-YZ581 | (1) | The Encoder distribution unit is required when two or more robot controllers are connected to the one encoder. Provide this unit as necessary.<br>If the Encoder distribution unit is used, a 5V power source for the encoder is not necessary.<br>Refer to the Encoder Distribution Unit Manual (BFP-A3300) for details. |

| Name of devices to be provided by customers | Model | Quantity | Remark |
|---|---|---|---|
| **Vision sensor part** | | | |
| Basic network vision sensor set | 4D-2CG5xxxx-PKG | 1 | See the instruction manual of the network vision sensor for details |
| In-Sight 5000 series<br>In-Sight Micro series<br>In-Sight EZ series | – | | COGNEX Vision sensor |
| Lens | – | | C-mount lens |
| Lighting installation | – | (1) | Provide as necessary. |
| **Connection part** | | | |
| Hub | – | 1 | |
| Ethernet cable (straight) | – | 2 | Between Robot controller and Hub<br>Between Personal computer and Hub |
| **Personal computer part** | | | |
| Personal computer | – | 1 | Please refer to the instruction manual of RT ToolBox2 or the instruction of the network vision sensor for details of the personal computer specifications. |
| RT ToolBox2<br>(Personal computer support software) | 3D-11C-WINE<br>3D-12C-WINE | | Please refer to the instruction manual of RT ToolBox2 for the details of the personal computer specifications. |

## 7.2. Example of System Configuration

The following figure shows examples of conveyer tracking systems and vision tracking systems.

### 7.2.1. Configuration Example of Conveyer Tracking Systems

The following figure shows a configuration example of a system that recognizes lined-up workpieces on a conveyer passing a photoelectronic sensor and follows the workpieces.

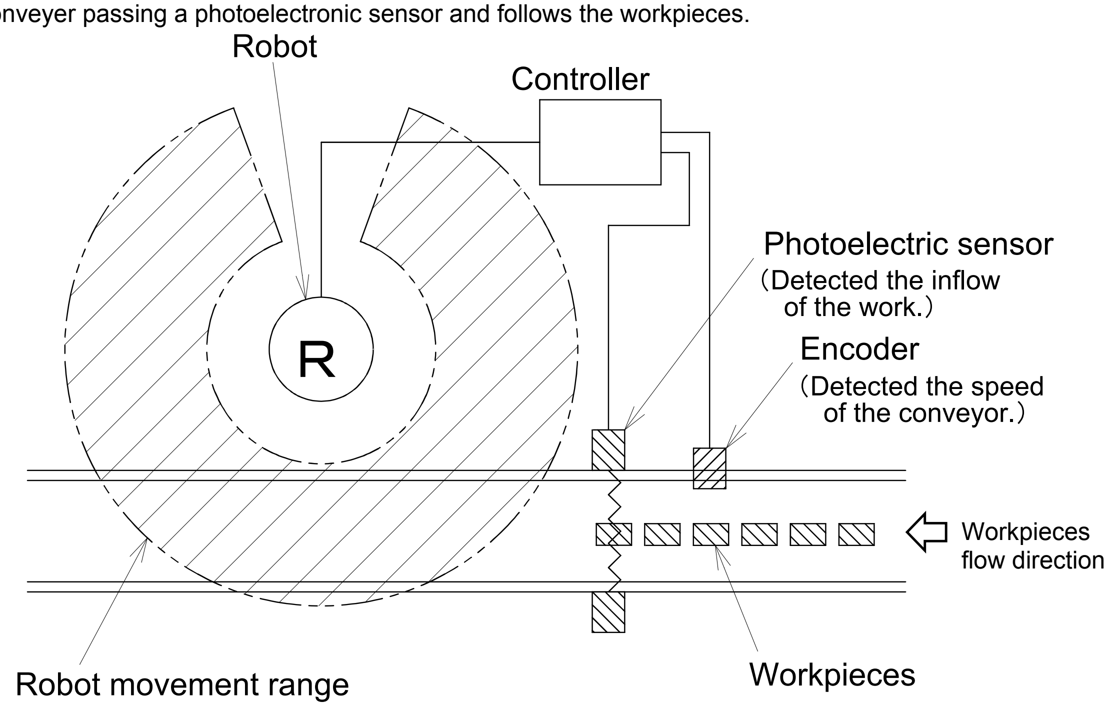



**Figure 7-1 Configuration Example of Conveyer Tracking (Top View)**

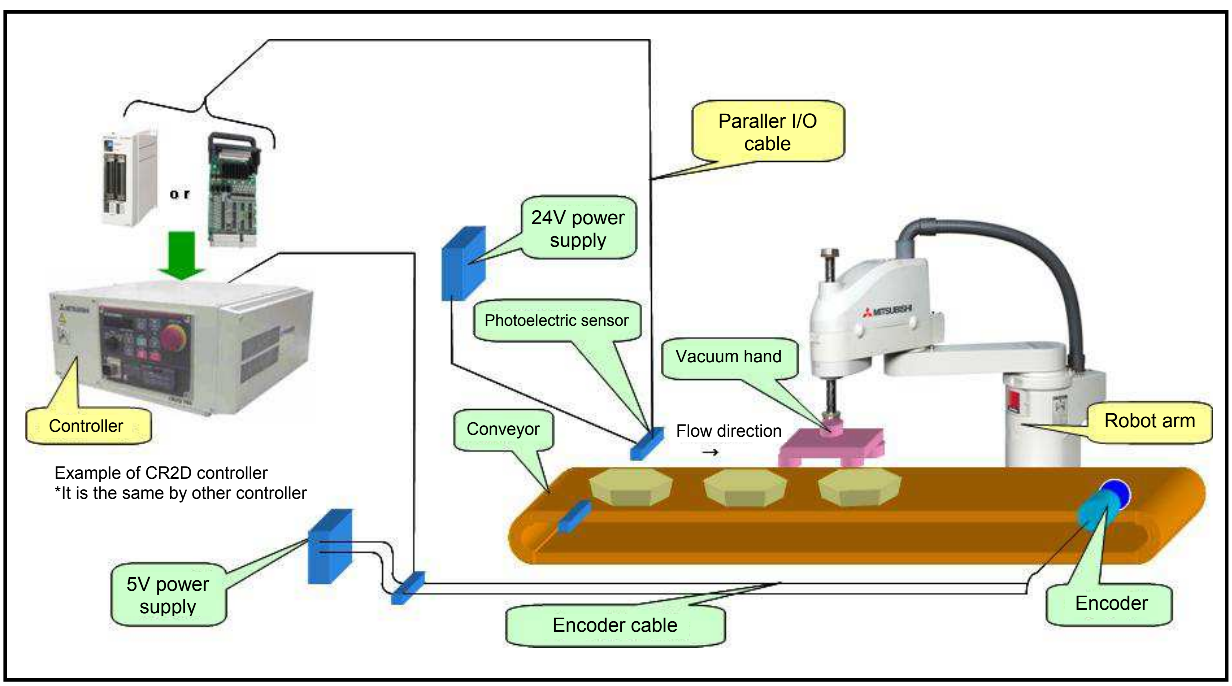



**Figure 7-2 Configuration Example of Conveyer Tracking**

### 7.2.2. Configuration Example of Vision Tracking Systems

The following figure shows a configuration example of a system that recognizes positions of workpieces that are not lined up on a conveyer with a vision sensor and follows the workpieces.

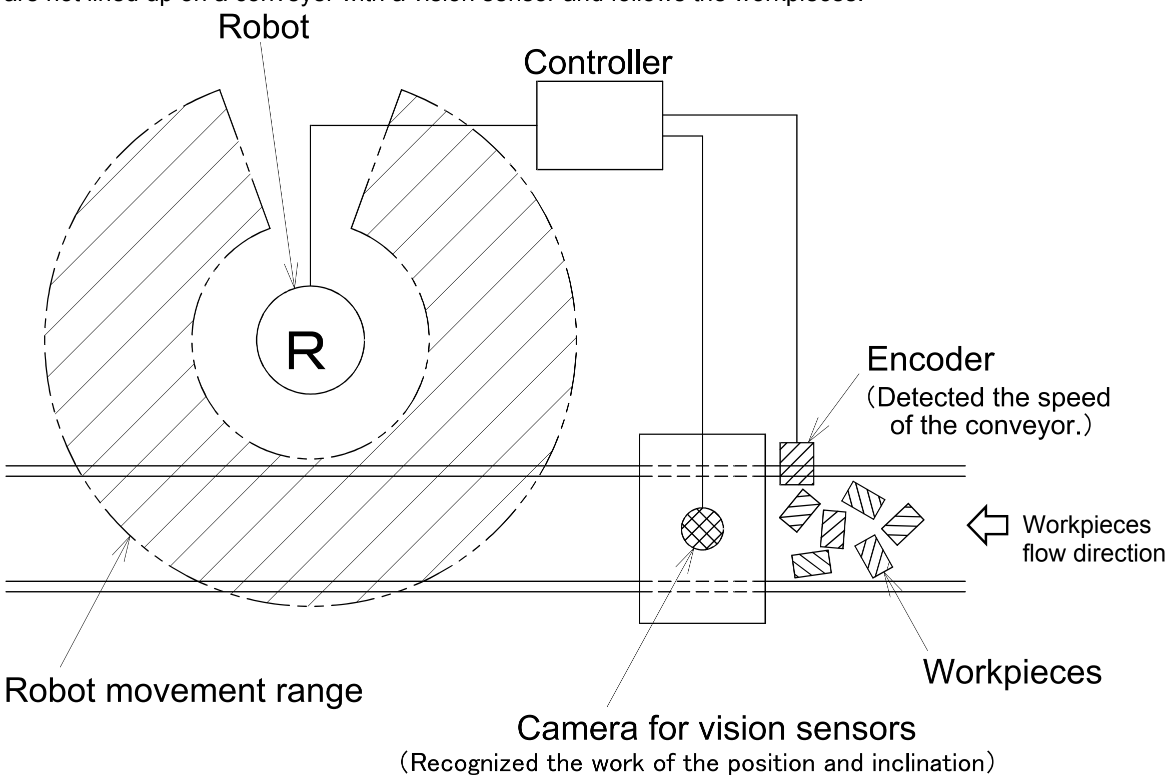



**Figure 7–3 Configuration Example of Vision Tracking (Top View)**

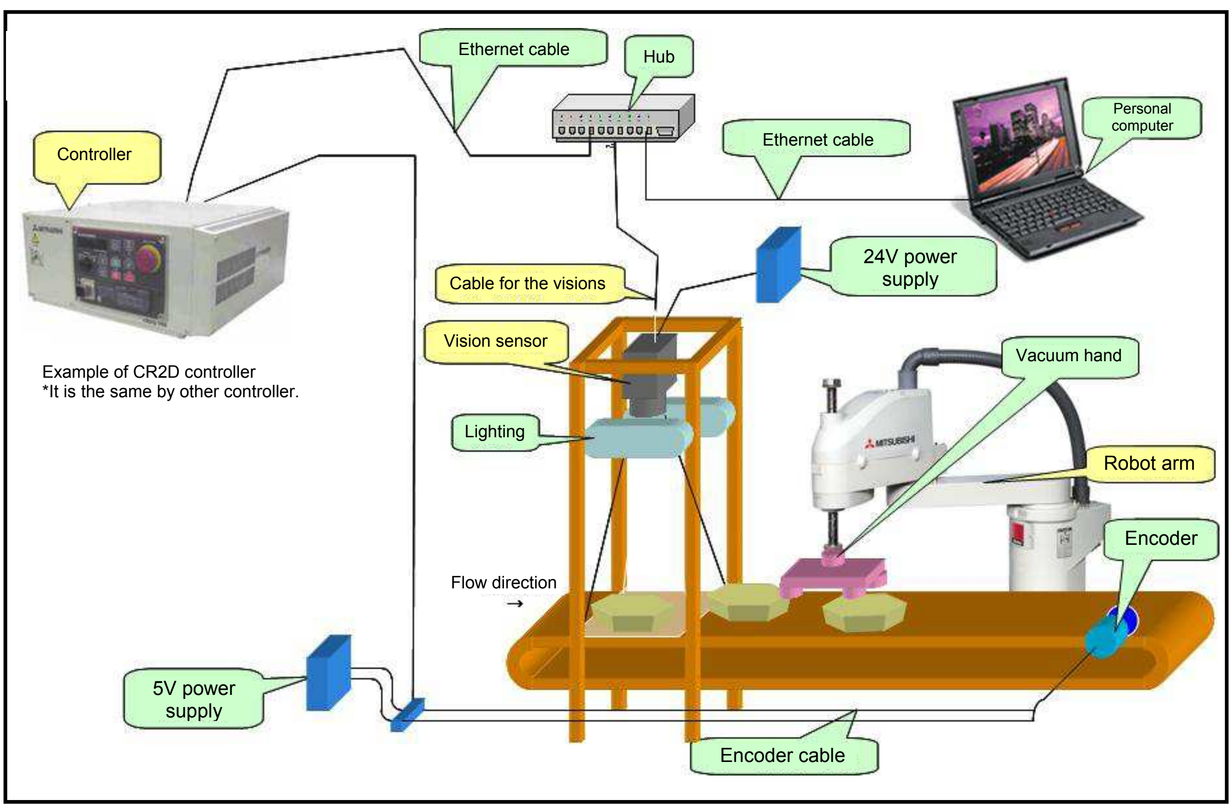



**Figure 7–4 Configuration Example of Vision Tracking**

# 8. Specification

## 8.1. Tracking Specifications and Restriction matter

"Table 3−1 CR750-Q/CR751-Q Series, CRnQ-700 Series Controller Tracking Function Specifications" shows the tracking specifications.

Please refer to "Standard Specifications Manual" for the specifications of the robot arm and controller to be used.

**Table 8−1 CR750-D/CR751-D Series, CRnD-700 Series Tracking Function Specifications**

| Item | | Specification and Restriction matter |
|---|---|---|
| Supported robots (*6) | | RH-SDH series / RV-SD series<br>RH-FH-D series / RV-F-D series |
| Applicable robot controller | | CR1D/ CR2D/CR3D contoller<br>CR750-D/CR751-D series controller |
| Robot program language | | Load commands dedicated for the tracking function |
| Conveyer | Number of conveyer | Max 2pcs （in case 1pcs encoder connect to 1pcs conveyer）<br>Encoder 2pcs / Robot controller 1pcs<br>The robot controller can correspond to two conveyers by the standard specification. |
| | Movement speed (*1) | Possible to support up to 300 mm/s (When the robot always transport the workpieces)<br>Possible to support up to 500 mm/s when the interval of workpiece is wide.<br>Possible to support two conveyers by one Robot controller. |
| | Encoder | Output aspect : A, $\overline{A}$, B, $\overline{B}$, Z, $\overline{Z}$<br>Output form : line driver output (*2)<br>Highest response frequency: 100 kHz<br>Resolution(pulse/rotation) : Up to 2000 (4000 and 8000 uncorrespond)<br>Confirmed operation product : Omron E6B2-CWZ1X-1000<br>E6B2-CWZ1X-2000 |
| | Encoder cable | Shielded twisted-pair cable<br>Outside dimension : Maximum phi6mm<br>Conductor size: 24AWG (0.2 $mm^2$) Cable length: Up to 25 m |
| Photoelectronic sensor (*3) | | Used to detect workpieces positions in conveyer tracking. |
| Vision sensor (*4) | | Mitsubishi's network vision sensor |
| Precision at handling position (*5) | | Approximately ±2 mm (when the conveyer speed is approximately 300 mm/s)<br>(Photoelectronic sensor recognition accuracy, vision sensor recognition accuracy, robot repeatability accuracy and so on) |

(*1) The specification values in the table should only be considered guidelines. The actual values depend on the specific operation environment, robot model, hand and other factors.

(*2) The line driver output is a data transmission circuit in accordance with RS-422A. It enables the long-distance transmission.

(*3) The output signal of a photoelectronic sensor must be connected to a general input signal (arbitrary) of the robot controller.

(*4) In the case of vision tracking, please refer to the instruction manual of network vision sensor.

(*5) The precision with which workpieces can be grabbed is different from the repeatability at normal transportation due to the conveyer speed, sensor sensitivity, vision sensor recognition accuracy and other factors. The value above should only be used as a guideline.

(*6) The sample program doesn't correspond to the RV-5 axis robot.

# 9. Operation Procedure

This chapter explains the operation procedure for constructing a conveyer tracking system and a vision tracking system using Mitsubishi Electric industrial robots CR750-D/CR751-D series, CRnD-700 series.

1. Start of operation

⬇

2. Connection of Equipment ········ Refer to “Chapter 10.”
   Chapter 10 explains installation of option cards and connection of an encoder.

⬇

3. Parameter Setting ········ Refer to “Chapter 11.”
   Chapter 11 explains assignment of signals and setting of parameters related to tracking to allow an external device to control a robot.

⬇

4. Sample Robot Programs ········ Refer to “Chapter 12.”
   Chapter 12 explains functions related to supplemental sample programs.

⬇

5. Calibration of Conveyer and Robot Coordinate Systems (“A1” program) ········ Refer to “Chapter 13.”
   Chapter 13 explains how to calculate the amount of robot movement per encoder pulse.

⬇

6. Calibration of Vision Coordinate and Robot Coordinate Systems (“B1” program) ··· Refer to “Chapter 14.”
   Chapter 14 explains how to display the position of a workpiece recognized by the vision sensor in the robot coordinate system.

⬇

7. Workpiece Recognition and Teaching (“C1” program) ········ Refer to “Chapter 15.”
   Chapter 15 explains how to calculate the relationship between the position of a workpiece recognized by the vision sensor and the position at which the robot grabs the workpiece.

⬇

8. Teaching and Setting of Adjustment Variables (“1” Program) ········ Refer to “Chapter 16.”
   Chapter 16 explains how to make settings such that the robot can follow workpieces moving by on a conveyer and how to teach the robot origin and transportation destination at system start-up.

⬇

9. Automatic Operation ········ Refer to “Chapter 18.”
   In automatic operation, the robot operates via commands from the conveyer control.

⬇

End of operation

10. Maintenance ········ Refer to “Chapter 19.”

11. Troubleshooting ········ Refer to “Chapter 20.”

# 10. Connection of Equipment

This section explains how to connect each of the prepared pieces of equipment.

## 10.1. Preparation of Equipment

Prepare equipment by referring to “Table 2−2 List of Devices Provided by Customers (Conveyer Tracking)” to construct a conveyer tracking system and “Table 2−3 List of Devices Provided by Customers (Vision Tracking)” to construct a vision tracking system.

## 10.2. Connection of Equipment

This section explains how to connect each of the prepared pieces of equipment.

### 10.2.1. Connection of Conveyer Encoder

Wiring of the encoder for the conveyors and the encoder cable is shown in the "Figure 10−1” (CRnD-700 series) or “Figure 10−3” (CR750-D/CR751-D series). Those shows the connection between a Expansion serial interface card connector and an encoder.(The cable uses E6B-2-CWZ1X (by OMRON).)
The a maximum of two encoders for the conveyors are connectable as standard specification. A total of 8 signal wires are required for the connection for the power supply (+ and - terminals) and the + and - terminals of the differential encoders' A, B and Z phases. Refer to the instruction manual of the encoders to be used and connect the signal wires correctly. Note that shielded wires (SLD) should be connected to the ground of the controller and system.

> ⚠ **CAUTION**
>
> ***Be sure to mount ferrite cores on all encoder cables.***
>
> **Be sure to mount the ferrite cores on the encoder cables at a position near the robot controller. If ferrite cores are not mounted, the robot may malfunction due to the influence of noise.**

> ⚠ **CAUTION**
>
> ***There is one robot controller connectable with the one encoder.***
>
> **If two or more robot controllers are connected to the one encoder, the waveform of the encoder falls and the exact encoder value may be unable to be acquired. If you want to connect two or more robot controller to the one encoder, the Encoder distribution unit (model: 2F-YZ581) is required. Refer to the Encoder Distribution Unit Manual (BFP-A3300) for details.**

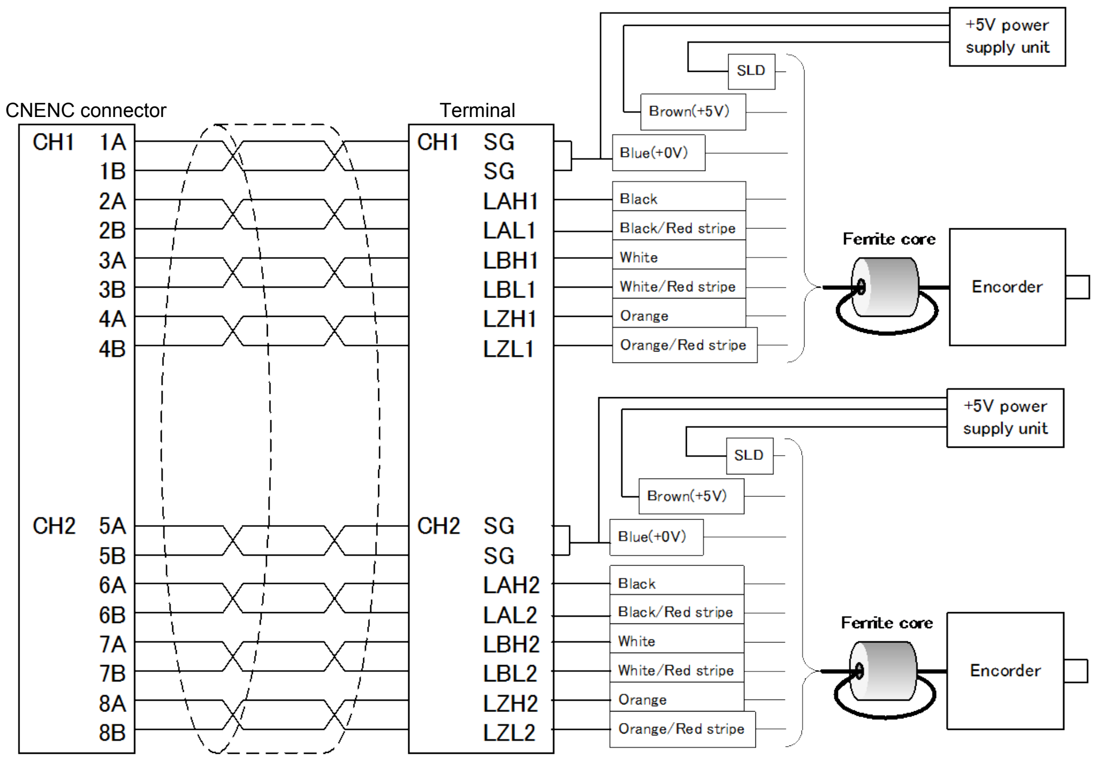



**Figure 10–1 Wiring of the encoder for conveyors and encoder cable (CRnD-700 series controller)**

Refer to "Table 21–3 Connectors: CNENC/CNUSR Pin Assignment" with pin assignment of connector CNENC.

The wiring example by the thing is shown below.
(Please note that the connector shape is different depending on the controller. )

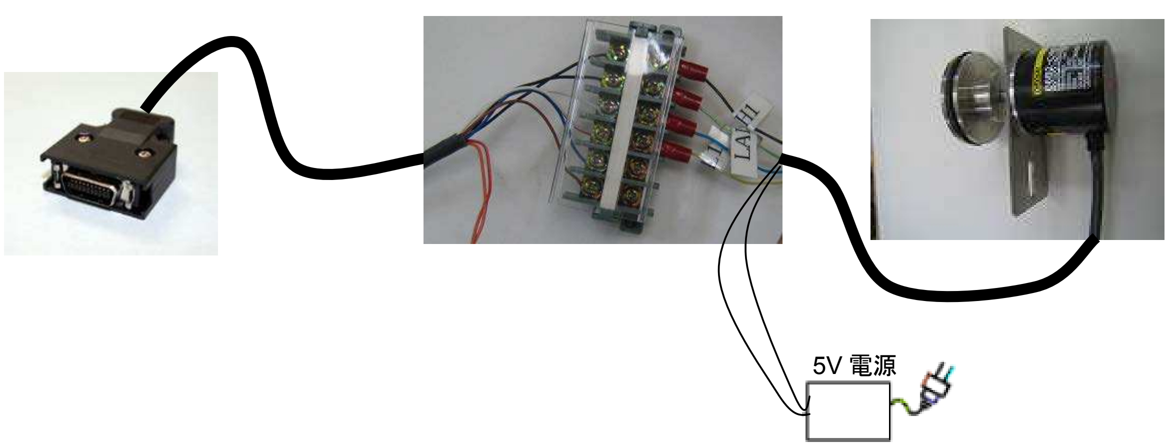



**Figure 10–2 Wiring example (CRnD-700 series controller)**

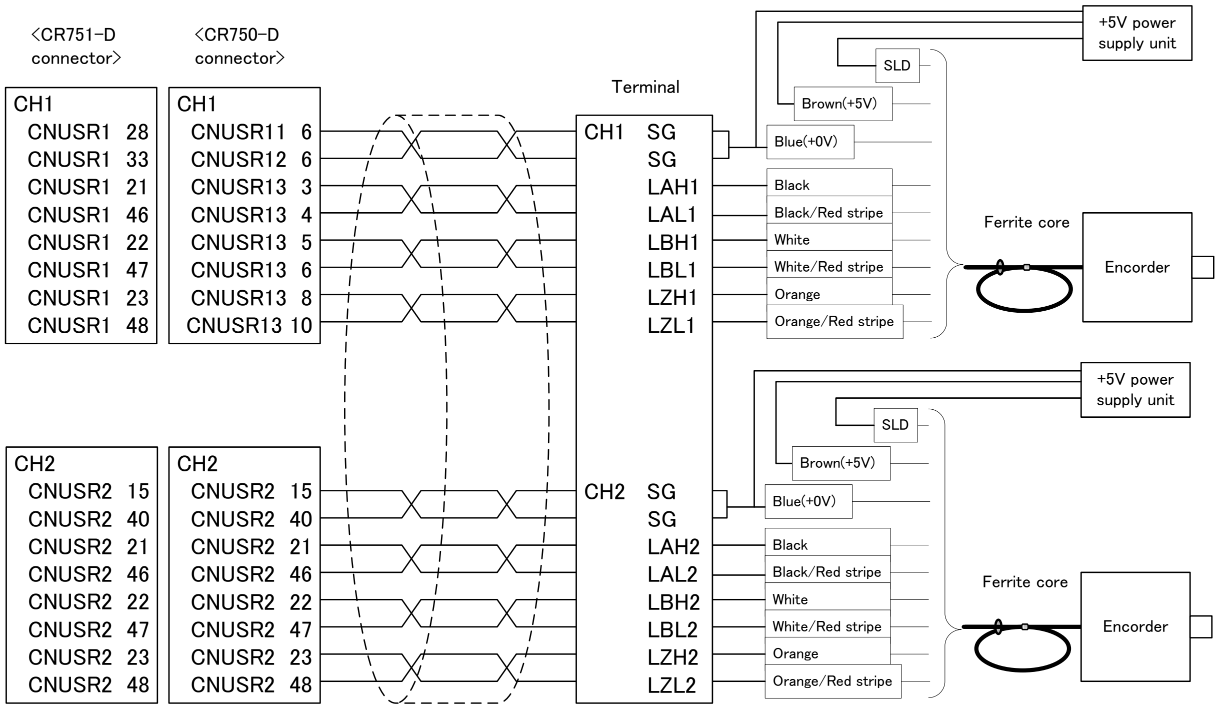



**Figure 10–3 Wiring of the encoder for conveyors and encoder cable (CR750-D/CR751-D series controller)**

Refer to "Table 21–3 Connectors: CNENC/CNUSR Pin Assignment" with pin assignment of connector CNUSR.

The wiring example by the thing is shown below.
(Please note that the connector shape is different depending on the controller. )

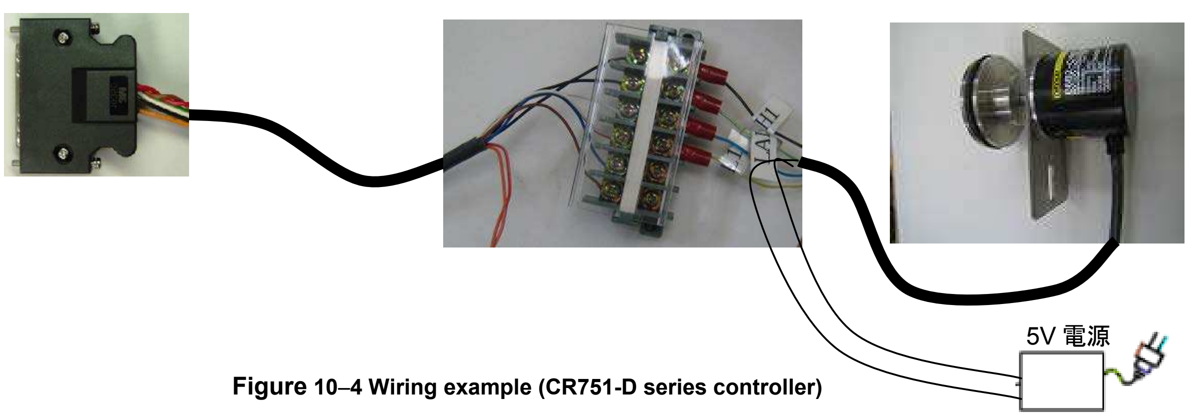



**Figure 10–4 Wiring example (CR751-D series controller)**

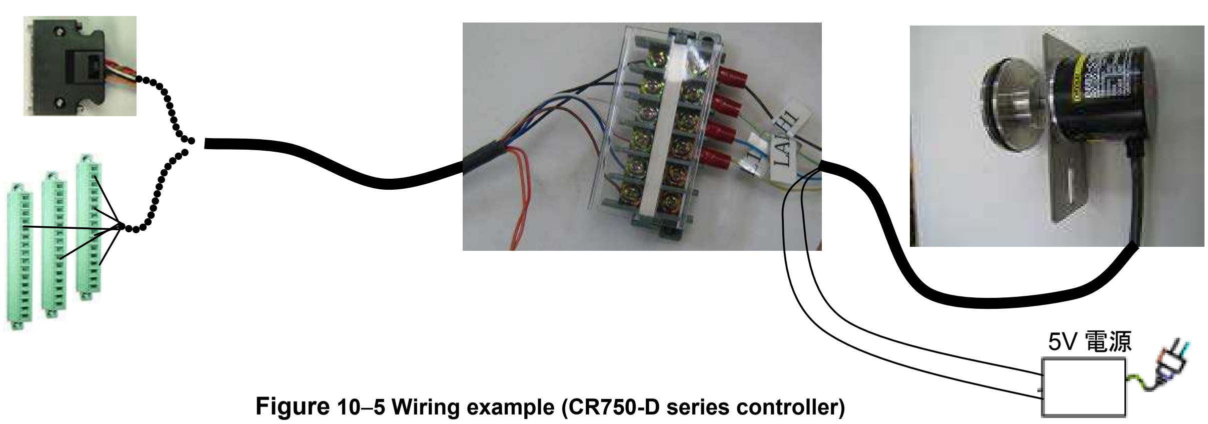



**Figure 10–5 Wiring example (CR750-D series controller)**

### 10.2.2. Installation of encoder cable

The installation method of the encoder cable is shown by controller to be used.
*CR750-D series: "Figure 10-6 Installation of encoder cable (CR750-D series) "
*CR751-D series: "Figure 10-7 Installation of encoder cable (CR751-D series)"
*CR1D-700 series: "Figure 10-8Installation of encoder cable (CR1D-700 series) "
*CR2D-700 series: "Figure 10-9Installation of encoder cable (CR2D-700 series) "
*CR3D-700 series: "Figure 10-10Installation of encoder cable (CR3D-700 series) "

And, the description about the measures against the noise is shown in the figure "Figure 10-11 Example of noise measures of tracking system".

#### (1)CR750-D series

<CR750-D series controller (rear)>

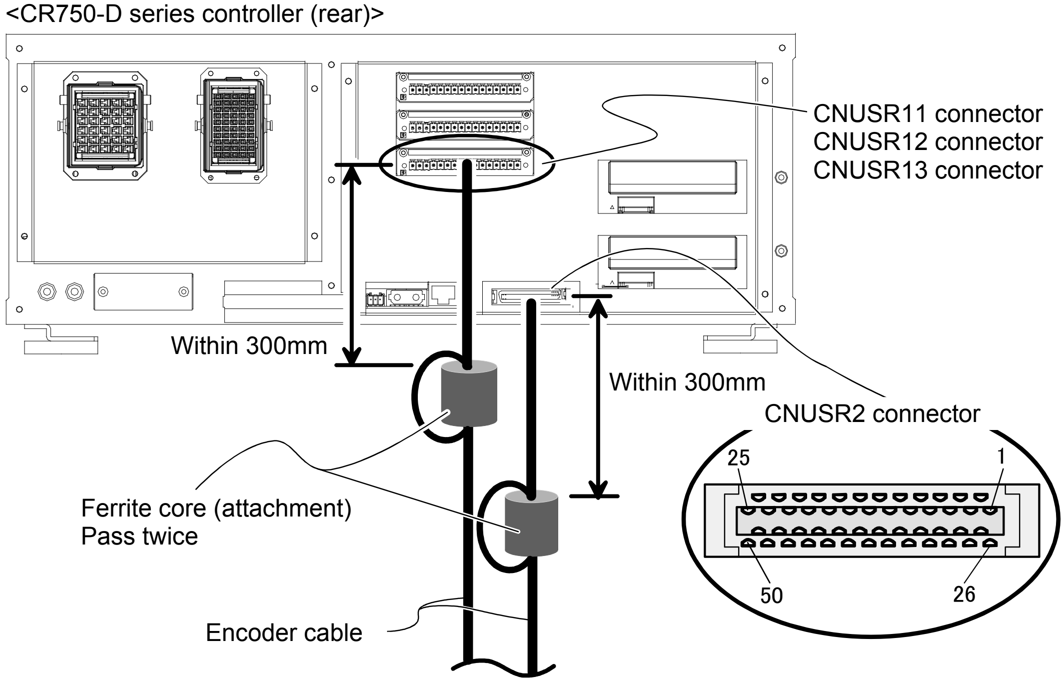



**Figure 10-6 Installation of encoder cable (CR750-D series)**

### (2)CR751-D series

<CR750-D series controller (front)>

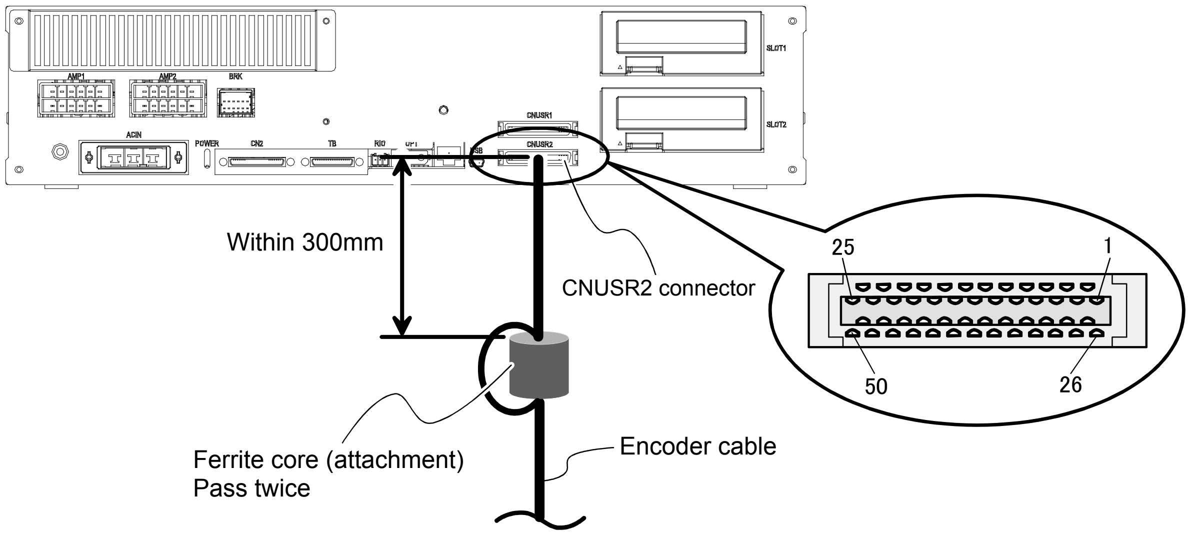



**Figure 10-7 Installation of encoder cable (CR751-D series)**

### (3)CR1D-700 series

Connect the encoder cable to the connector of the [CNENC] display. And, the ground of the cable uses the rear cover.

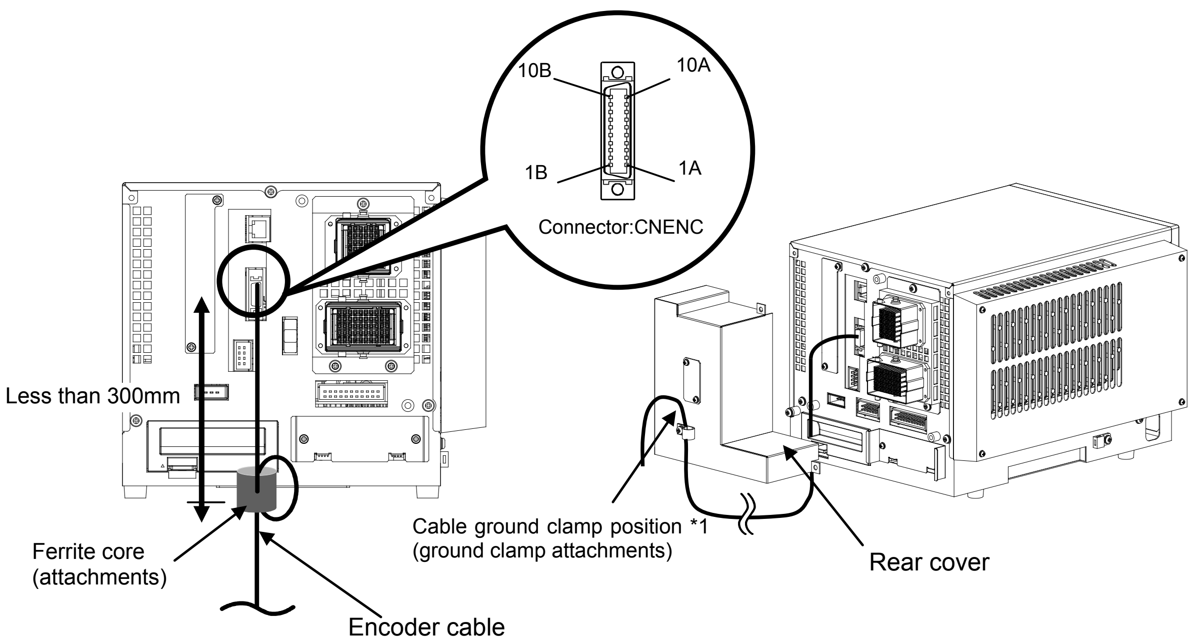



**Figure 10-8 Installation of encoder cable (CR1D-700 series)**

### (4)CR2D-700 series

Connect the encoder cable to the connector of the [CNENC] display. And, the ground of the cable uses the rear cover.

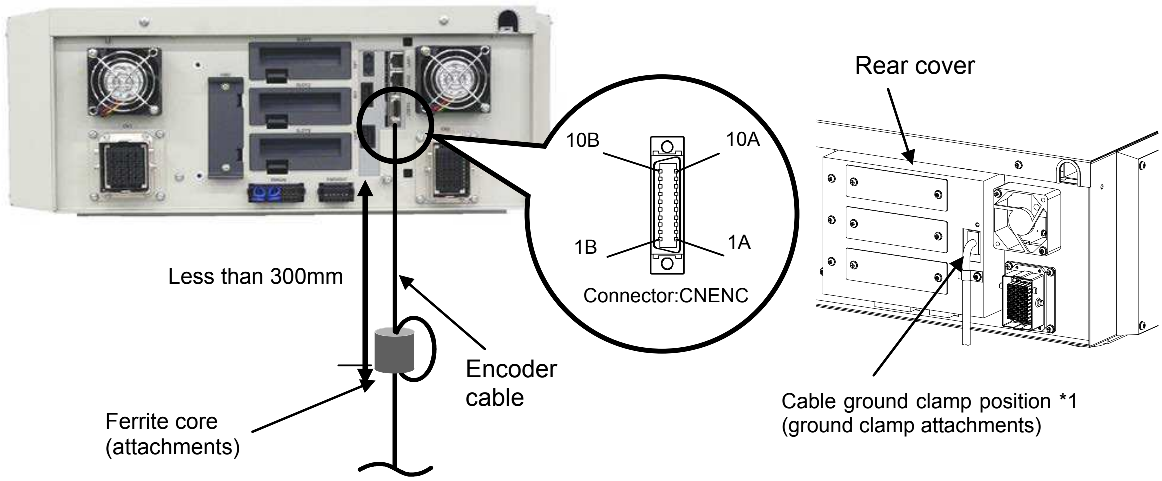



**Figure 10-9 Installation of encoder cable (CR2D-700 series)**

### (5)CR3D-700 series

Connect the encoder cable to the connector of the [CNENC] display. And, the ground of the cable uses the rear cover.

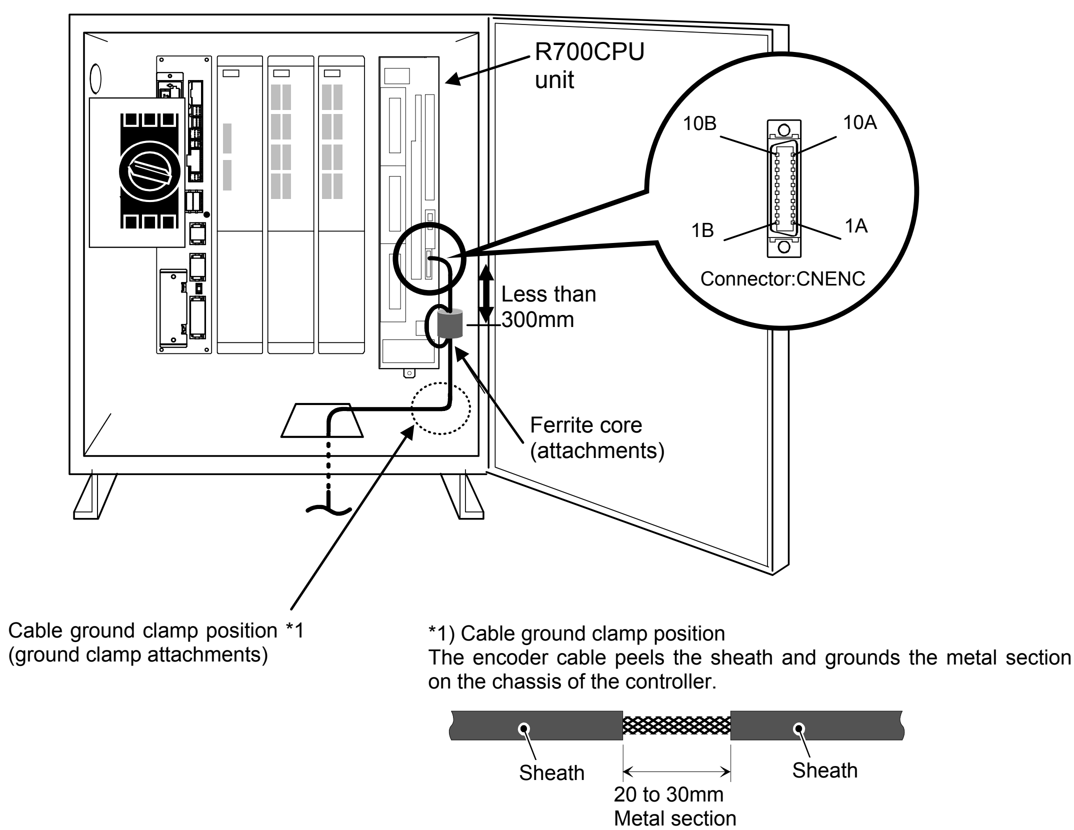



**Figure 10-10 Installation of encoder cable (CR3D-700 series)**

**(6)Measures against the noise**

The example of noise measures of the tracking system is shown in the following.
Please implement the measures against the noise if needed in the power supply periphery section for the encoders which prepared of the customer.

1) Please insert AC line filter (recommendation: MXB-1210-33 * Densei-Lambda) in the AC input side cable of the power supply for the encoders.
2) Please insert the ferrite core (recommendation: E04SR301334 * SEIWA ELECTRIC MFG.) in the DC output side cable of the power supply for the encoders.
3) Please connect the power supply case for the encoders to the installation operator control panel, connect the earth wire to grounding or the case, and insert the ferrite core (recommendation: E04SR301334 * SEIWA ELECTRIC MFG.).

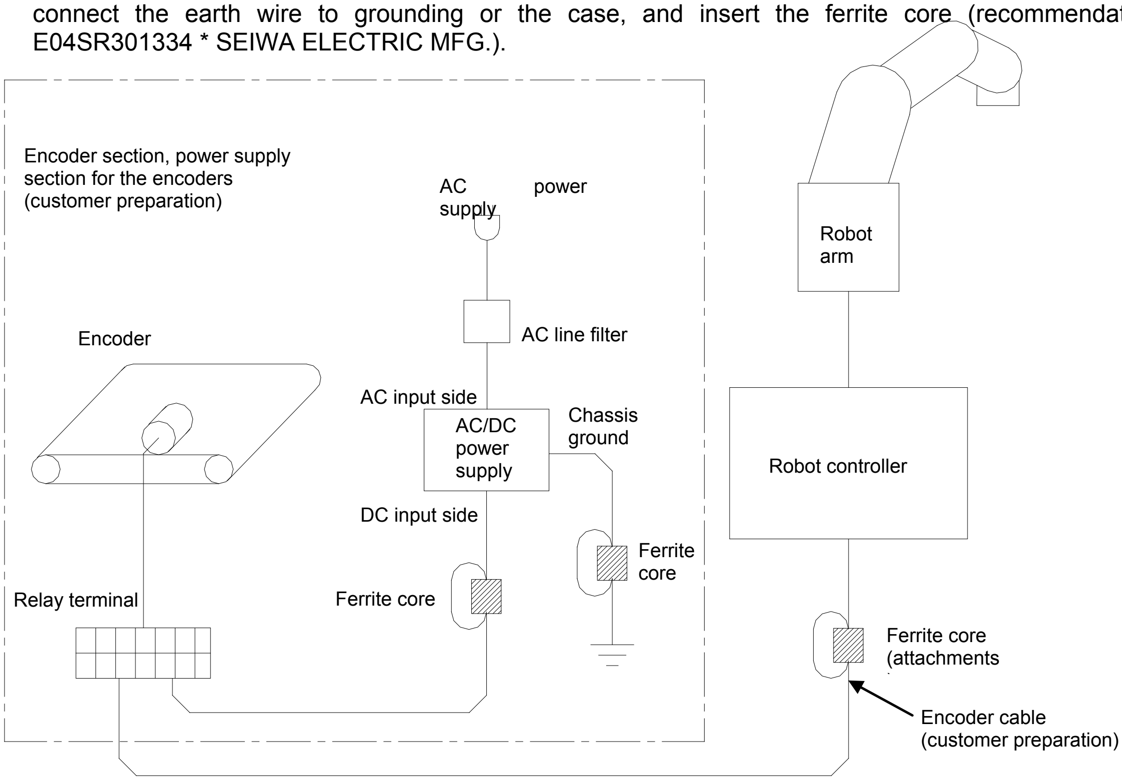



**Figure 10-11 Example of noise measures of tracking system**

### 10.2.3. Connection of Photoelectronic Sensor

If a photoelectronic sensor is used for detection of workpieces, connect the output signal of the photoelectronic sensor to a general input signal of the robot controller. Any general input signal number of the robot controller can be selected.
In this section, a connection example where the photoelectronic sensor signal is connected to the 6th general input signal is shown in "Figure 10–13 Photoelectronic Sensor Connection Example (6th General Input Signal is Used)."

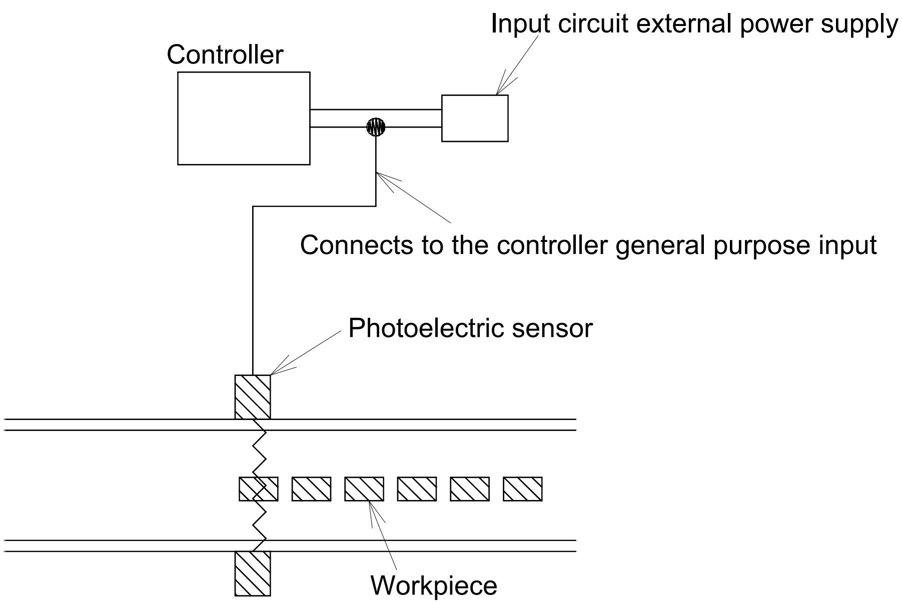
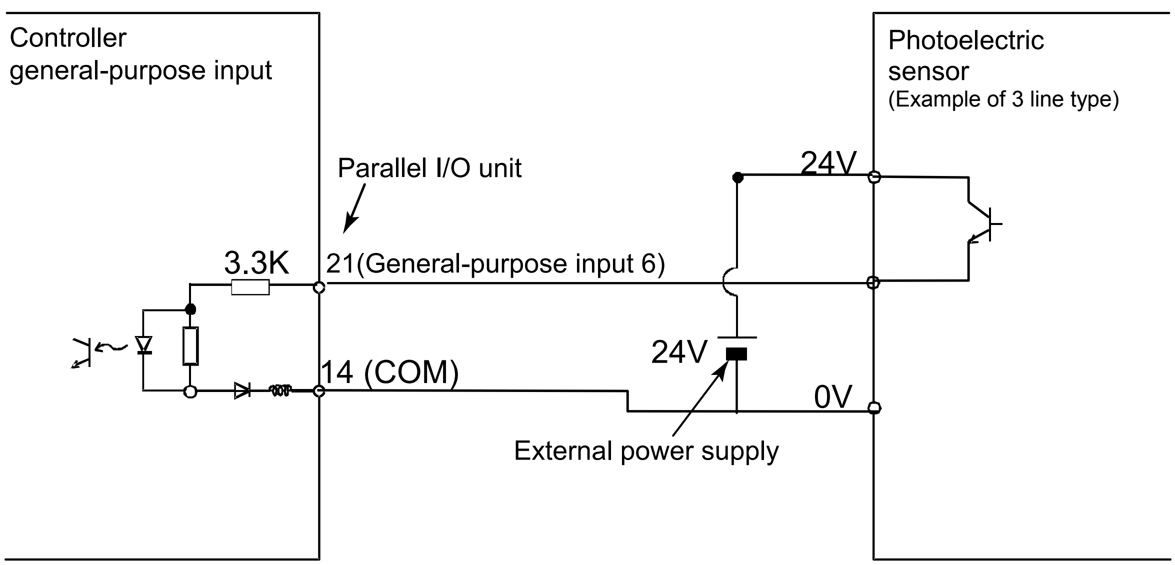



**Figure 10–12 Photoelectronic Sensor Arrangement Example**




Note) The external power supply and photoelectric sensor must be prepared by the customer.

Note) This connection example shows the connection of the source type.

**Figure 10–13 Photoelectronic Sensor Connection Example (6th General Input Signal is Used)**

# 11. Parameter Setting

This chapter explains how to set dedicated input/output signals that play the role of interface between a robot and an external device (e.g., a Programmable Logic Controller) and parameters related to the tracking function. Please refer to “Detailed Explanations of Functions and Operations” for how to set the parameters.

## 11.1. Dedicated Input/Output Parameters

“Table 11−1 List of Dedicated Input/Output Parameters” lists the setting items of dedicated input/output parameters used to operate the robot via instructions from an external device. Set the signal numbers according to your system using the setting values in the table as reference. **It is not necessary to set these parameters if the robot operates by itself, rather than via instructions from an external device.**

**Table 11−1 List of Dedicated Input/Output Parameters**

| Input name/output name (parameter name) | Explanation | Setting Example (*1) |
|---|---|---|
| Stop/pausing<br>(**STOP**) or (**STOP2**) | Input: Stop a program<br>Output: Output program standby status | **0 , -1** |
| Servo OFF/servo ON disabled<br>(**SRVOFF**) | Input: Turn the servo off<br>Output: Output servo ON disabled status | **1 , -1** |
| Error reset/error occurring<br>(**ERRRESET**) | Input: Cancel error status<br>Output: Output error status | **2 , -1** |
| Start/operating<br>(**START**) | Input: Start automatic operation<br>Output: Output program running status | **3 , 1** |
| Servo ON/turning servo ON<br>(**SRVON**) | Input: Turn the servo on<br>Output: Output servo on status | **4 , 0** |
| Operation right/operation right enabled (**IOENA**) | Input: Enable/disable operation right of external signal control<br>Output: Output external signal control operation enabled status | **5 , -1** |
| Program reset/program selectable<br>(**SLOTINIT**) | Input: Initiate a program. The program execution returns to the first step.<br>Output: Output a status where program No. can be changed | **10 , -1** |
| General output signal reset<br>(**OUTRESET**) | Input: Reset a general output signal | **11 , -1** |
| User specification area 1<br>(**USRAREA**) | Output an indication that the robot is in an area specified by a user<br>Set the start number and end number | **8 , 8** |

**(*1) “-1” in the Setting value column means “not set.”**

## 11.2. Operation Parameters

“Table 11−2 List of Operation Parameter” lists the setting items of parameters required to operate the robot at the optimal acceleration/deceleration.

**Table 11−2 List of Operation Parameter**

| Parameter name | Explanation | Reference value |
|---|---|---|
| Optimal acceleration/ deceleration hand data<br>(**HANDDAT1**) | Specify hand weight and so on to make settings that allow optimal acceleration/deceleration operations.<br>For example, if the hand weighs 3 kg, changing the weight setting value from 10 kg to 3 kg makes the robot movement faster.<br>(Hand weight (kg), size (mm) X, Y, Z, gravity (mm) X, Y, Z) | (3,0,0,0,0,0,0)<br>The setting values are different for each robot model.<br>**Use these values as reference only**. |
| Optimal acceleration/ deceleration workpiece data<br>(**WRKDAT1**) | Specify workpiece weight and so on to make settings that allow optimum acceleration/deceleration operations.<br>If a workpiece is grabbed via the HClose instruction, the acceleration/deceleration becomes slower. If a workpiece is released via the HOpen instruction, acceleration/deceleration becomes faster.<br>(Workpiece weight (kg), size (mm) X, Y, Z, gravity (mm) X, Y, Z) | (1,0,0,0,0,0,0)<br>The setting values are different for each robot model.<br>**Use these values as reference only.** |

## 11.3. Tracking Parameter Setting

Specify to which channel of a Encoder connector(CNENC) an encoder of a conveyer is connected.
“Table 11–3 Tracking Parameter Setting” lists the parameters to be set. Other parameters are shown in “Table 16-1 List of Tracking Parameters”; make settings as required.

**Table 11–3 Tracking Parameter Setting**

<table>
<tr><th>Parameter</th><th>Parameter name</th><th>Number of elements</th><th>Explanation</th><th>Value set at factory shipping</th></tr>
<tr><td>Tracking mode</td><td>TRMODE</td><td>1 integer</td><td>Enable the tracking function<br>Please set it to “1" when you use the tracking function.<br>0: Disable/1: Enable</td><td>0</td></tr>
<tr><td>Encoder number allocation</td><td>EXTENC</td><td>8 integers</td><td>Set connection destinations on the connector for encoder numbers 1 to 8.<br>Parameter elements correspond to encoder number 1, encoder number 2 … encoder number 8 from the left.<br>In addition, the encoder physics numbers 3-8 are the reservation number for extension. At present, it cannot be used.
<table>
<tr><th>Connection channel</th><th>Encoder physics number</th><th></th></tr>
<tr><td>Standard CH1</td><td>1</td><td></td></tr>
<tr><td>Standard CH2</td><td>2</td><td></td></tr>
<tr><td>Slot1 CH1</td><td>3</td><td rowspan="6">Reservation number for future extension</td></tr>
<tr><td>Slot1 CH2</td><td>4</td></tr>
<tr><td>Slot2 CH1</td><td>5</td></tr>
<tr><td>Slot2 CH2</td><td>6</td></tr>
<tr><td>Slot3 CH1</td><td>7</td></tr>
<tr><td>Slot3 CH2</td><td>8</td></tr>
</table>
The value of the encoder which wired the channel 1 in case of the standard encoder input connector [CNENC] for the robot controller is equipped with the encoder cable with initial setting,The value of the encoder which wired the channel 2 by the status variable "M_Enc (1)", "M_Enc (3)", "M_Enc (5)", and "M_Enc (7)",It can confirm by the status variable "M_Enc (2)", "M_Enc (4)", "M_Enc (6)", and "M_Enc (8)."<br><br>It is convenient to check the status variable “M_Enc” when determining the setting value of the “EXTENC” parameter.<br>Please refer to "19.1.2 List of Robot Status Variables” for the explanation of state variable “M_Enc".<br>Please refer to “Detailed Explanations of Functions and Operations” for how to check the status variable “M_Enc.”</td><td>1,2,3,4,1,2,3,4</td></tr>
<tr><td>Tracking Workpiece judgement distance</td><td>TRCWDST</td><td>1 integer</td><td>Distance to judge that the same workpiece is being tracked (mm)<br>The sensor reacts many times when the workpiece with the ruggedness passes the sensor. Then, the robot controller judged that one workpiece is two or more pieces.<br>The sensor between values [mm] set to this parameter does not react after turning on the sensor.</td><td>5.00</td></tr>
</table>

# [Part 4] Tracking Control (common function between series)

(Take note that there are some aspects which differ between CR750-Q, CR751-Q, CRnQ-700 series and CR750-D, CR751-D, CRnD-700 series.)

## 12. Sample Robot Programs

This chapter explains the structure of the sample robot programs.

Two types of sample robot programs are provided; for conveyer tracking and for vision tracking. Their program structures are shown in “Table 12–1 List of Sample Robot Programs (Conveyer Tracking)” and “Table 12–2 List of Sample Robot Programs (Vision Tracking)” respectively.
Refer to “RT ToolBox2 Robot Total Engineering Support Software Instruction Manual” for how to install programs to the robot controller.

**Table 12–1 List of Sample Robot Programs (Conveyer Tracking)**

| Program name | Description | Explanation |
|---|---|---|
| **A1** | Conveyer - robot coordinate system calibration program | This program matches the coordinate systems of the conveyer and robot and calculates the amount of robot movement per encoder pulse. |
| **C1** | Workpiece coordinate system - robot coordinate system matching program | This program calculates the coordinates at which the robot grabs a workpiece based on the coordinates at which a sensor is activated. |
| **1** | Operation program | This program handles transporting workpieces while following recognized workpieces.<br>(1) Movement to the robot origin<br>(2) Workpiece suction and transportation operation while following movement |
| **CM1** | Workpiece coordinate monitor program | This program monitors encoder values and stores workpiece coordinates. |

**Table 12–2 List of Sample Robot Programs (Vision Tracking)**

| Program name | Description | Explanation |
|---|---|---|
| **A1** | Conveyer - robot coordinate system calibration program | This program matches the coordinate systems of the conveyer and robot and calculates the amount of robot movement per encoder pulse. |
| **B1** | Vision coordinate system – robot coordinate system calibration program | This program matches the vision coordinate system and the robot coordinate system. |
| **C1** | Workpiece coordinate system - robot coordinate system matching program | This program calculates the coordinates at which the robot grabs a workpiece based on the coordinates at which a vision sensor has detected the workpiece. |
| **1** | Operation program | This program handles transporting workpieces while following recognized workpieces.<br>(1) Movement to the robot origin<br>(2) Workpiece suction and transportation operation while following movement |
| **CM1** | Workpiece coordinate monitor program | This program monitors encoder values and stores workpiece coordinates. |

# 13. Calibration of Conveyer and Robot Coordinate Systems (“A1” program)

This chapter explains the tasks carried out by using “A1” program.

*** “A1” program contains operations required for both conveyer tracking and vision tracking.**

Calibration of a conveyer refers to determining the movement direction of the conveyer in the robot coordinate system and the amount of movement of the robot per encoder pulse. This amount of movement is stored in the robot’s status variable “P_EncDlt.”

“A1” Program performs specified tasks and automatically calculates the amount of movement of the robot per encoder pulse mentioned above.

The procedures of operations specified by “A1” program and items to be confirmed after the operations are explained below.

Please refer to “Detailed Explanations of Functions and Operations” for the steps involved in each operation.

Please monitor status variable “M_Enc(1)" to “M_Enc(8)" before it works, rotate the encoder, and confirm the value changes.

## 13.1. Operation procedure

1) Mount a calibration jig on the mechanical interface of a robot. Connect a personal computer on which RT ToolBox2(option) is installed to the robot controller.

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

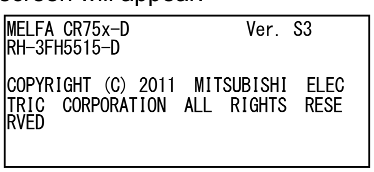
O/P

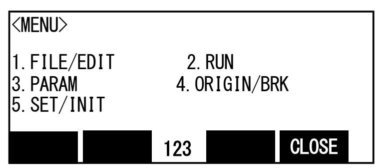

T/B

Up : DISABLE
Down : ENABLE
*Lamp lighting

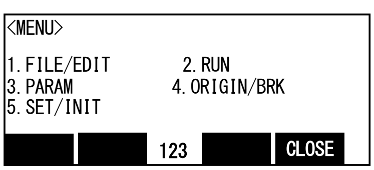
T/B rear

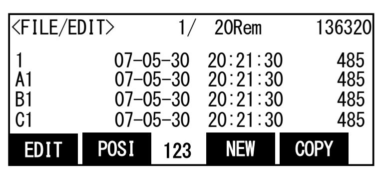
3) Press one of the keys (example, [EXE] key) while the <TITLE> screen is displayed. The <MENU> screen will appear.



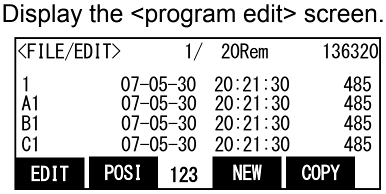



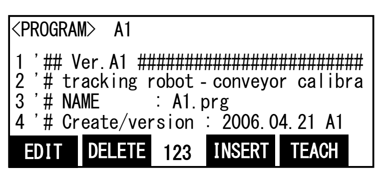

4) Select "1. FILE /EDIT" screen on the <MENU> screen.



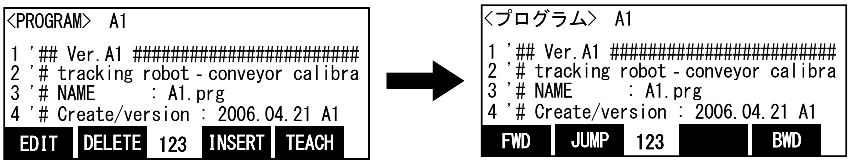




5) Press the arrow key, combine the cursor with the program name "A1" and press the [EXE] key. Display the <program edit> screen.

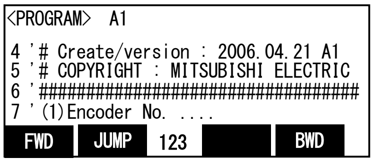


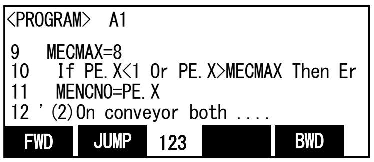
6) Press the [FUNCTION] key, and change the function display



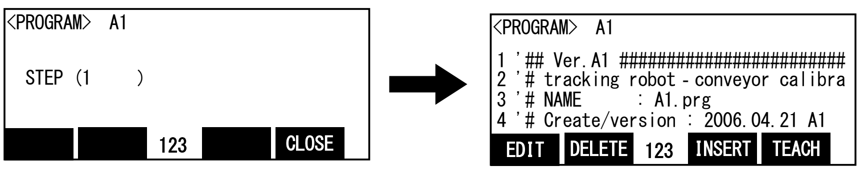

7) Press the [F1] (FWD) key and execute step feed. "(1)Encoder No ......." is displayed



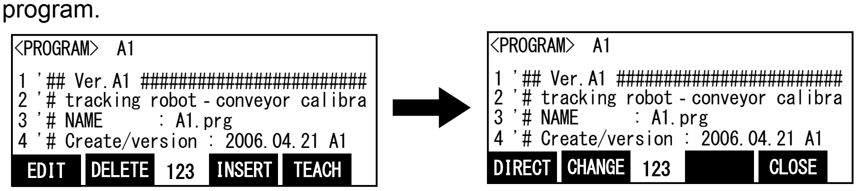

8) Work according to the comment directions in the robot program.
9) Next "' (2) On conveyor both .. Execute step feed to ".




10) Repeat (7) - (8) and execute step feed to "End."
11) Press the [F2] (JUMP) key and input the step number. Press the [EXE] key. Then returns to first step




12) Press the [FUNCTION] key, and change the function display. Press the [F4] (close) key and close the program.



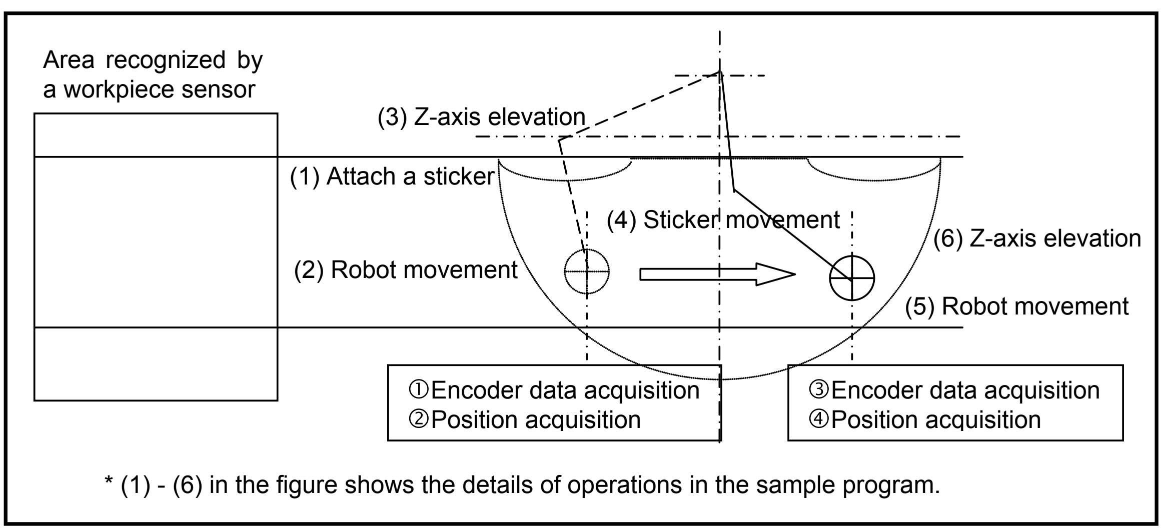
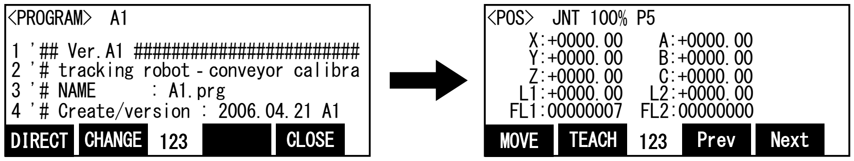




**Figure 13–1 Conveyer and Robot Calibration Operation Diagram**

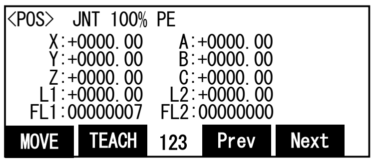
## 13.2. Tasks

1) Set the encoder number to the X coordinates value of position variable: "PE."

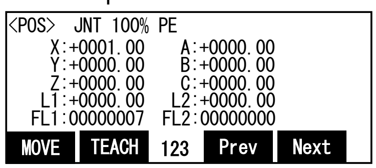
(a) Press the function key ([F2]) corresponding to "the change", and display the position edit screen.



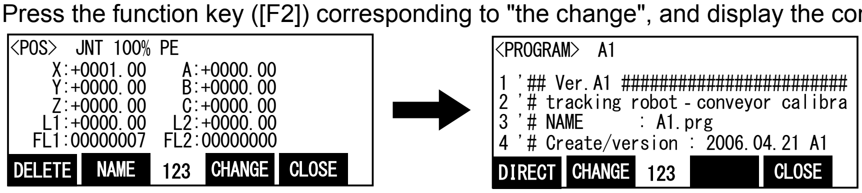

(b) The [F3] (Prev) key or the [F4] (Next) key is pressed, change the target variable, and display "PE" on the position name.




(c) X coordinates are selected by the arrow key, press the [CLEAR] key for a long time, and delete the details. Input the encoder number into X coordinates.




(d) Press the function key ([F2]) corresponding to "the change", and display the command edit screen.



2) Attach a marking sticker on the conveyer (a sticker with an X mark is the best choice for the marking sticker).
   Drive the conveyer and stop it when the marking sticker comes within the robot movement range.

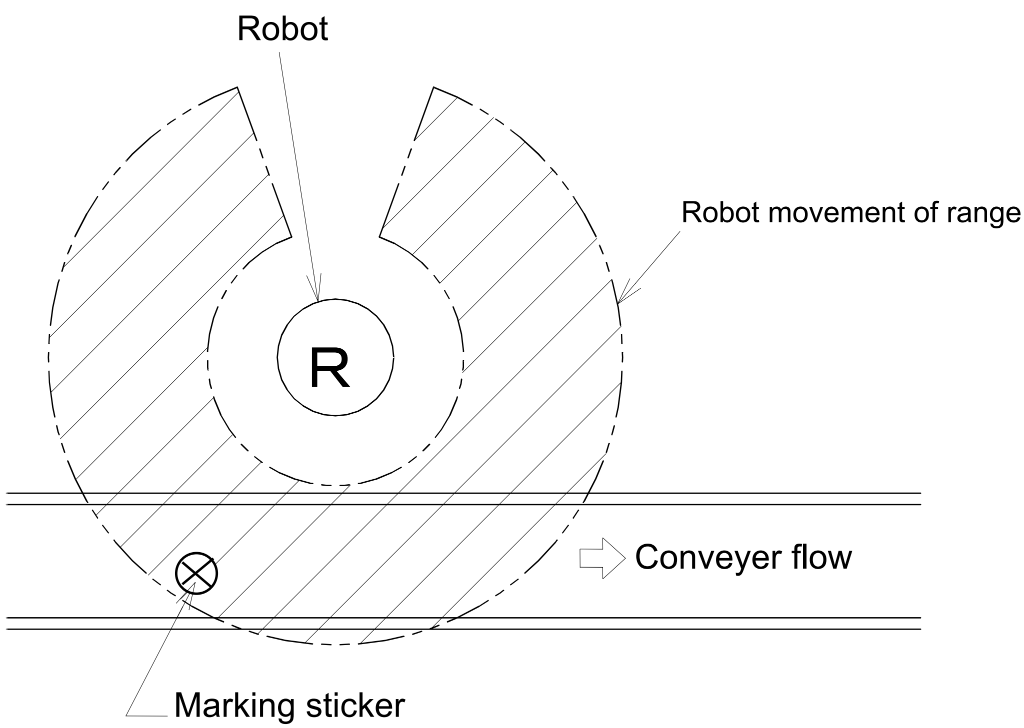



**Figure 13–2 Position of Marking Sticker on Conveyer**

3) Move the robot to the position right at the center of the marking sticker on the conveyer.
   *** With this operation, encoder data and robot position are acquired.**

> ⚠ **CAUTION**
>
> ***Move the robot to an accurate position.***
> Be sure to move the robot to the position exactly at the center of the marking sticker because the amount of robot movement per encoder pulse is determined by the robot positions specified for the first and second times. Moreover, pay attention to the robot height as well because this amount of movement includes changes of robot position in the Z axis direction.

4) Raise the robot.
5) Drive the conveyer and stop at a position where the marking sticker is immediately outside the robot movement range.

> ⚠ **CAUTION**
>
> ***The marking sticker should be moved for the maximum amount of movement allowed by the robot movement range.***
> If the amount of movement is too small, errors in the amount of robot movement per encoder pulse will become large due to the error of the position specified for the robot.

6) Move the robot to the position right above the center of the marking sticker on the moved conveyer.
   *** With this operation, encoder data and robot position are acquired.**
7) Raise the robot.
8) Perform step operation until "End."
   *** The amount of robot movement per encoder pulse is calculated based on this operation.**

## 13.3. Confirmation after operation

Check the value of “P_EncDlt” using T/B.

*** This value indicates the movement of each coordinate (mm) of the robot coordinate system, corresponding to the movement of the conveyer per pulse.**

Example) If ”0.5” is displayed for the Y coordinate only

This means that if the conveyer moves for 100 pulses, the workpiece moves 50 mm (0.5 x 100 = 50) in the +Y direction in the robot coordinate system.

When backing up, the data of “P_EncDlt" is not backed up.
Please work referring to "20.3.5 Restore backup data to another controller" when you restore data to another tracking system.

## 13.4. When multiple conveyers are used

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

Example) When using conveyer 2 (encoder number 2):

(a) Enter “2” for the encoder number specified for the X coordinate of the position variable “PE” in the program.

(b) Check the value of “P_EncDlt(2)” using RT ToolBox2 when confirming the data after operation.

**Refer to “RT ToolBox2 Robot Total Engineering Support Software Instruction Manual” for how to check variable values using RT ToolBox2.**

# 14. Calibration of Vision Coordinate and Robot Coordinate Systems (“B1” program)

This chapter explains the tasks carried out by using “B1” program.

*** “B1” program only contains operations required when constructing a vision tracking system. These operations are not necessary when constructing a conveyer tracking system.**

Calibration of a vision sensor refers to converting the position of a workpiece recognized by the vision sensor to the corresponding position in the robot coordinate system.

This calibration operation is easily performed by the “Mitsubishi robot tool” in In-Sight Explorer. Refer to "Mitsubishi robot tool manual for EasyBuilder" for the details of this function.

“B1” program performs specified tasks and allows acquiring the workpiece coordinates recognized by the vision sensor in the robot coordinate system (position coordinates of robot movement).

The procedures of operations specified by “B1” program and items to be confirmed after the operations are explained below.

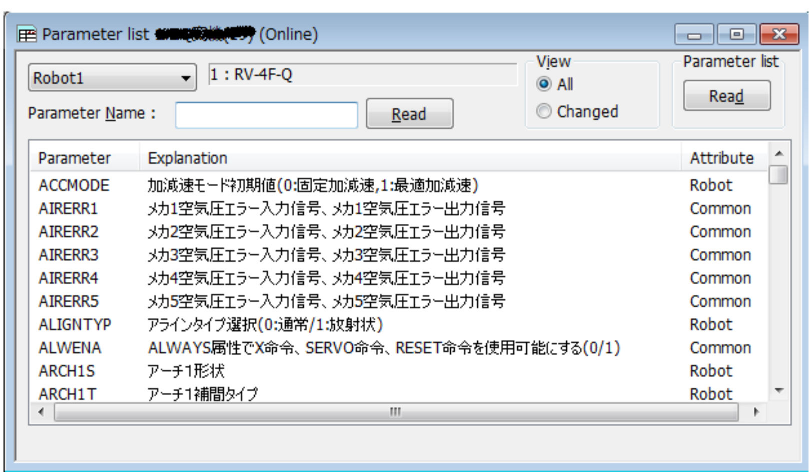
This chapter explains on the assumption that "Mitsubishi robot tool" is used.

Please refer to “Detailed Explanations of Functions and Operations” for the steps involved in each operation.

## 14.1. Operation procedure

1) To communicate the Mitsubishi robot tool and the vision sensor, set a necessary parameter by using RT ToolBox2.
   A necessary parameter is three (“NETIP", "Element 9 of NETTERM", and “CTERME19").
   In RT ToolBox2, select [Online]-[parameter]-[parameter list].
   Input the following parameters to "Parameter Name" of the displayed "Parameter list" screen and change a “Setting value”.

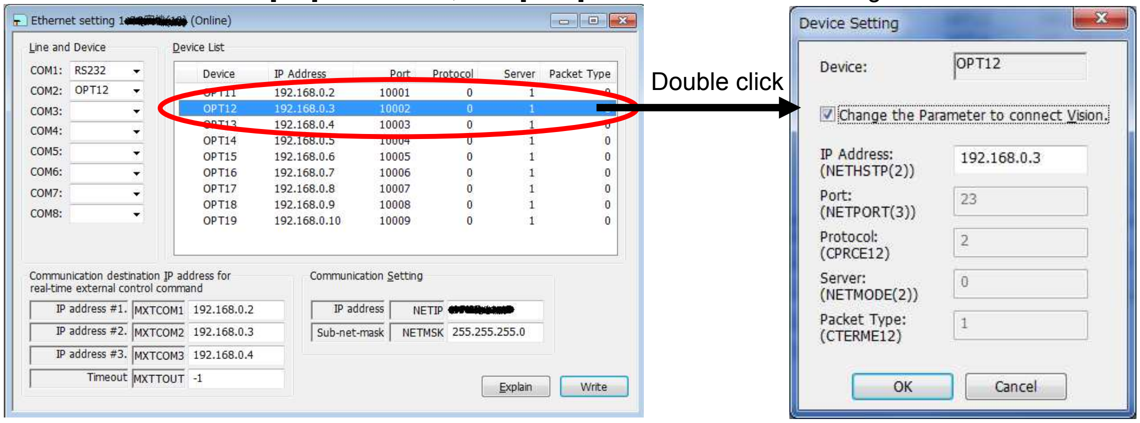
| Parameter Name | Initial value | Setting value | Explanation |
|---|---|---|---|
| NETIP | Q type : 192.168.100.1<br>D type : 192.168.0.20 | xxx.xxx.xxx.xxx | IP address of robot controller |
| NETTERM(Element 9) | 0 | 1 | The end code is added with communication. |
| CTERME19 | 0 | 1 | The end code of port 10009 is changed to “CR+LF”. |




Please confirm whether the following parameters are initial values.

| Parameter Name | Initial value | Explanation |
|---|---|---|
| NETPORT(Element 10) | 10009 | Port number allocated to device OPT19 |
| CPRCE19 | 0 | The protocol used is “Non-procedure” |
| NETMODE(Element 9) | 1 | Opens as “Server”. |

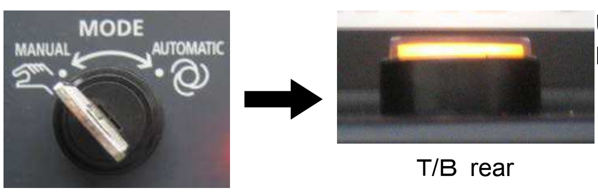
In RT ToolBox2, select [Online]-[parameter]-[Ethernet setting].
“OPT12" is selected “COM2:" that exists in "Line and Device" column on the displayed "Ethernet setting" screen. Double-click “OPT12" that exists in "Device List" .
Check "Change the parameter to connect Vision", and Input IP address of the vision sensor to "IP Address:" column. Click [OK] button. And, click [write] button on “Ethernet setting” screen.




Turn on robot controller's power supply again to make the set parameter effective.

2) Open “B1” program using T/B.
Set the controller mode to "MANUAL". Set the T/B to "ENABLE".

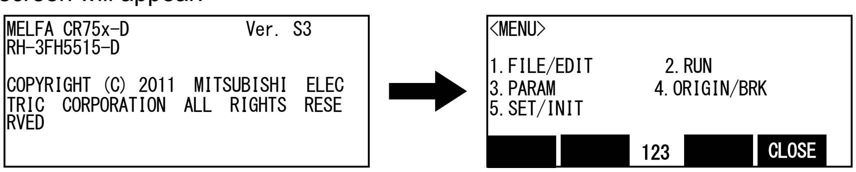
O/P T/B



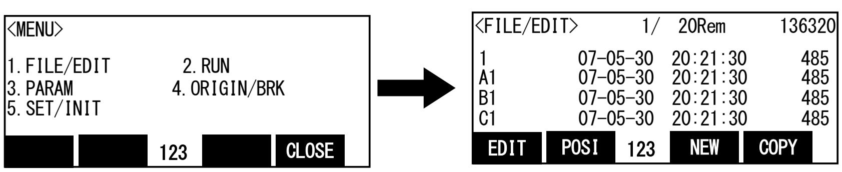

Up : DISABLE
Down : ENABLE
*Lamp lighting

T/B rear

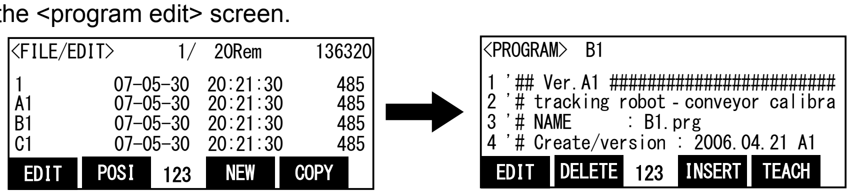
3) Press one of the keys (example, [EXE] key) while the <TITLE> screen is displayed. The <MENU> screen will appear.




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




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



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

```
<PROGRAM>  B1

1 '## Ver.A1 ########################
2 '# tracking robot - conveyor calibra
3 '# NAME      : B1.prg
4 '# Create/version : 2006.04.21 A1
 EDIT  DELETE  123  INSERT  TEACH
```

→

```
<PROGRAM>  B1

1 '## Ver.A1 ########################
2 '# tracking robot - conveyor calibra
3 '# NAME      : B1.prg
4 '# Create/version : 2006.04.21 A1
 FWD   JUMP   123          BWD
```

7) Press the [F1] (FWD) key and execute step feed. "(1)Encoder No ......." is displayed

```
<PROGRAM>  B1

4 '# Create/version : 2006.04.21 A1
5 '# COPYRIGHT : MITSUBISHI ELECTRIC
6 '##################################
7 '(1)Encoder No. ....
 FWD   JUMP   123          BWD
```

8) Work according to the comment directions in the robot program.
9) Next "' (2) Vision sensor .. Execute step feed to ".

```
<PROGRAM>  B1

9   MECMAX=8
10   If PE.X<1 Or PE.X>MECMAX Then Er
11   MENCNO=PE.X
12 '(2) Vision sensor....
 FWD   JUMP   123          BWD
```

10) Repeat (7) - (8) and execute step feed to "End."
11) Press the [F2] (JUMP) key and input the step number. Press the [EXE] key. Then returns to first step

```
<PROGRAM>  B1


  STEP (1     )


              123          CLOSE
```

→

```
<PROGRAM>  B1

1 '## Ver.A1 ########################
2 '# tracking robot - conveyor calibra
3 '# NAME      : B1.prg
4 '# Create/version : 2006.04.21 A1
 EDIT  DELETE  123  INSERT  TEACH
```

12) Press the [FUNCTION] key, and change the function display. Press the [F4] (close) key and close the program.

```
<PROGRAM>  B1

1 '## Ver.A1 ########################
2 '# tracking robot - conveyor calibra
3 '# NAME      : B1.prg
4 '# Create/version : 2006.04.21 A1
 EDIT  DELETE  123  INSERT  TEACH
```

→

```
<PROGRAM>  B1

1 '## Ver.A1 ########################
2 '# tracking robot - conveyor calibra
3 '# NAME      : B1.prg
4 '# Create/version : 2006.04.21 A1
 DIRECT CHANGE 123          CLOSE
```

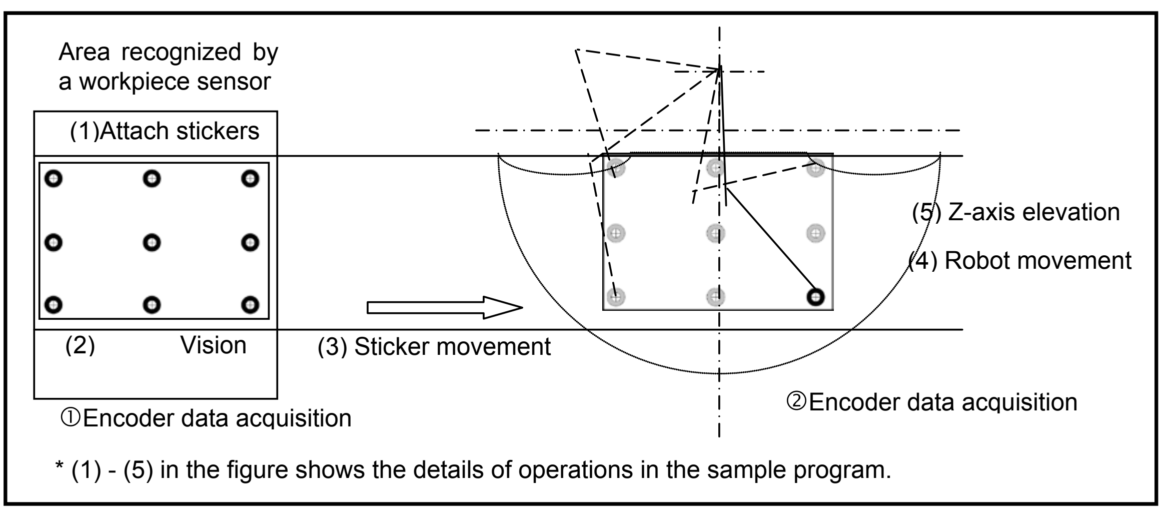
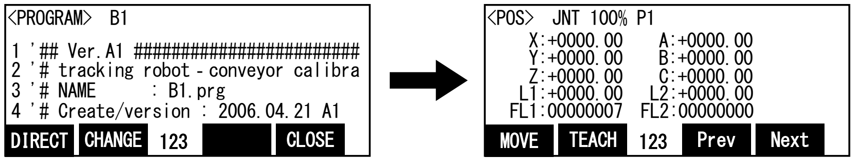



**Figure 14–1 Vision Sensor and Robot Calibration Operation Procedure Diagram**

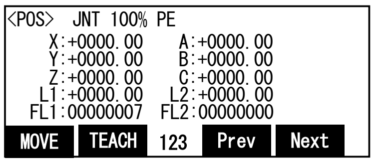
## 14.2. Tasks

1) Set the encoder number to the X coordinates value of position variable: "PE."

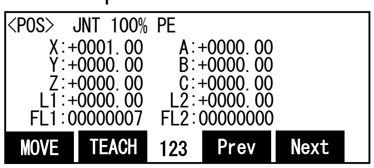
   (a) Press the function key ([F2]) corresponding to "the change", and display the position edit screen.



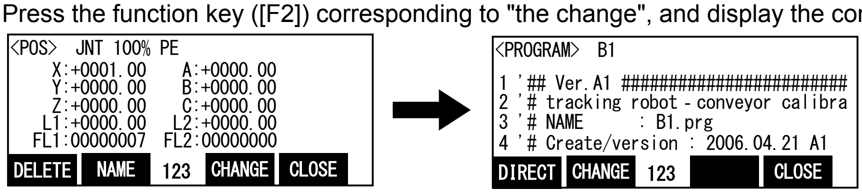

   (b) The [F3] (Prev) key or the [F4] (Next) key is pressed, change the target variable, and display "PE" on the position name.




   (c) X coordinates are selected by the arrow key, press the [CLEAR] key for a long time, and delete the details. Input the encoder number into X coordinates.




   (d) Press the function key ([F2]) corresponding to "the change", and display the command edit screen.




2) Start In-Sight Explorer and make the vision sensor into the off-line. Select the [Live Video] of “Set Up Image” in “Application Steps” Menu and display the picture which the vision sensor picturized on real time. Refer to the manual obtained from the Cognex for operation of In-Sight Explorer.

3) Paste appendix calibration seat to "Mitsubishi robot tool manual for EasyBuilder" on the conveyer. Paste calibration seat within the field of vision checking the live images of In-Sight Explorer.

*** With this operation, encoder data is acquired.**

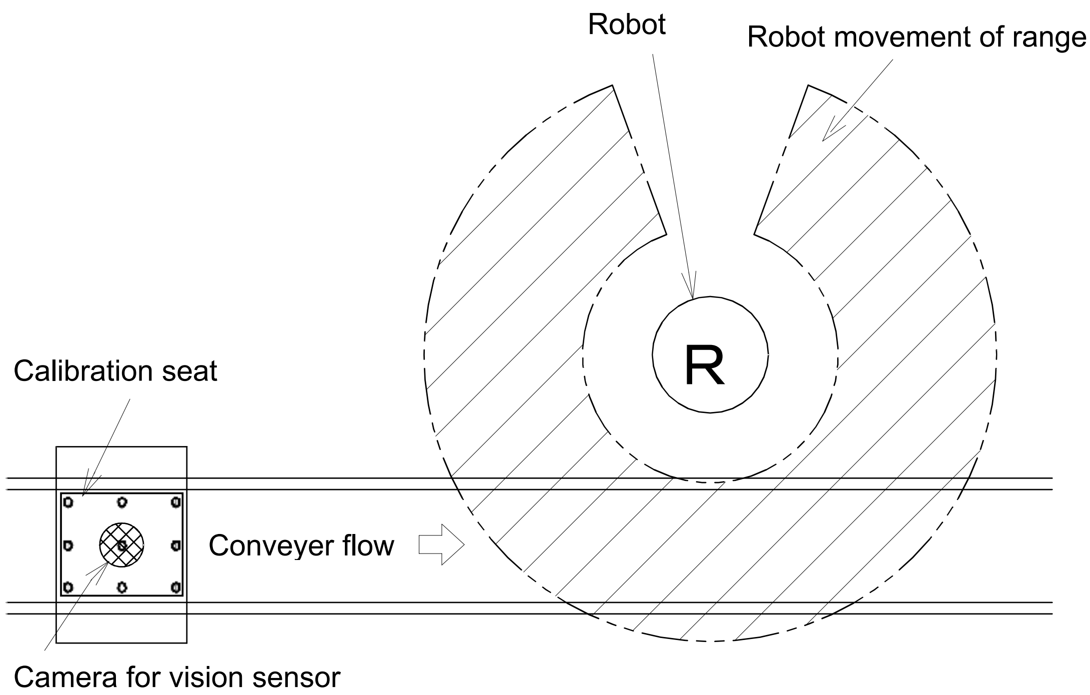



**Figure 14–2 Pasting Calibration seat**

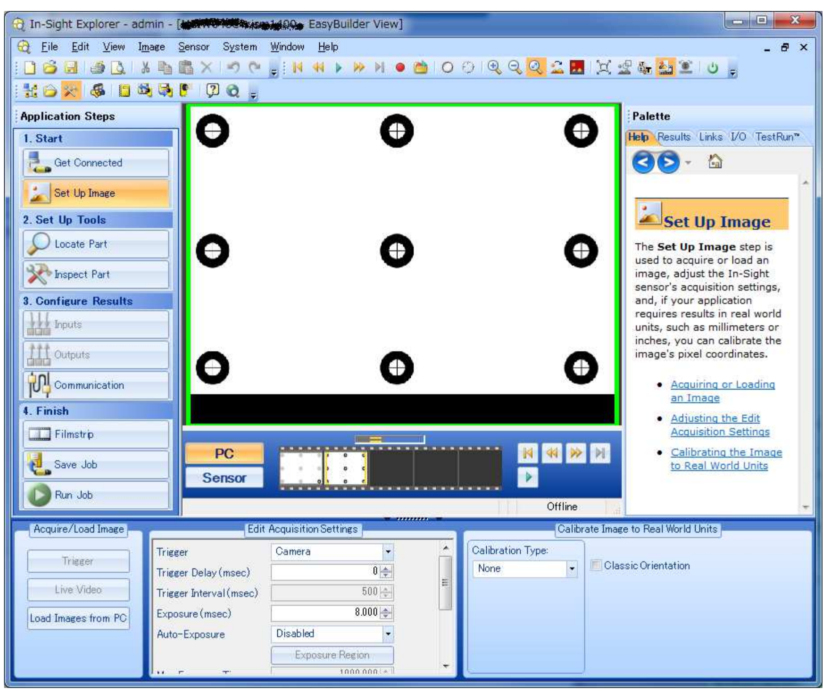



**Figure 14–3 Screen of In-Sight Explorer from which calibration seat is taken picture**

4) End [Live Video] of In-Sight Explorer, and select [Inspect Part] button of “Application Steps”.
5) Select [Geometry Tools] - [User-Defined Point] in “Add tool”.

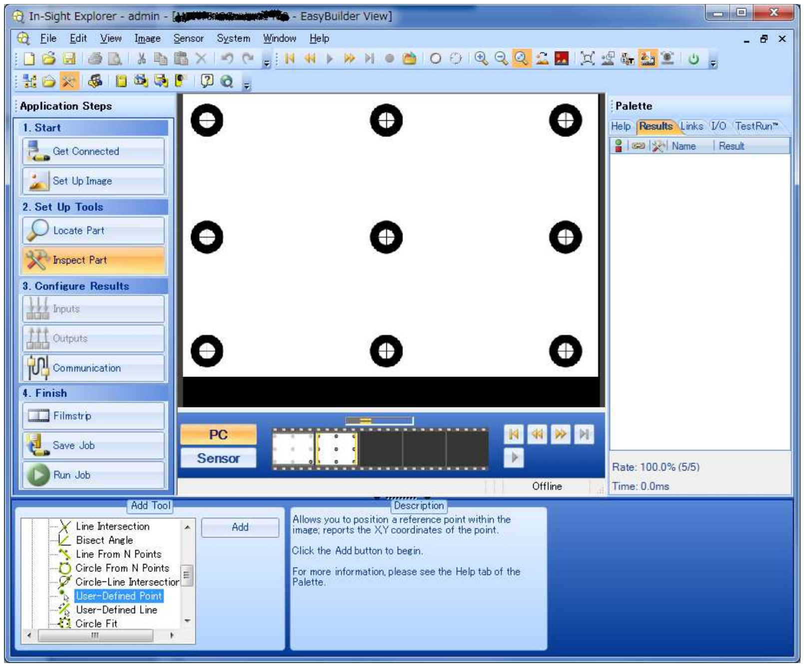
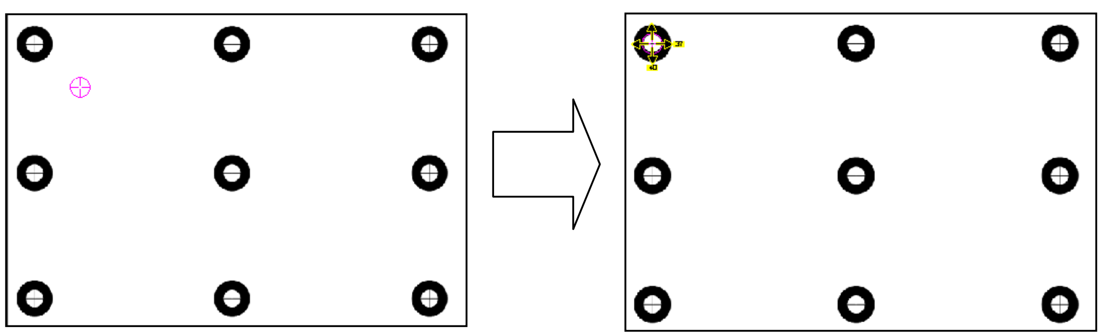
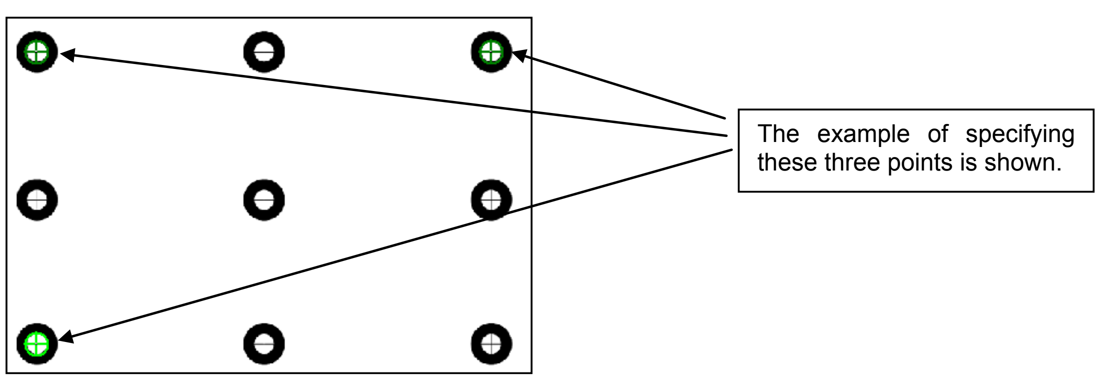



**Figure 14–4 Screen of In-Sight Explorer from which calibration seat is taken picture**

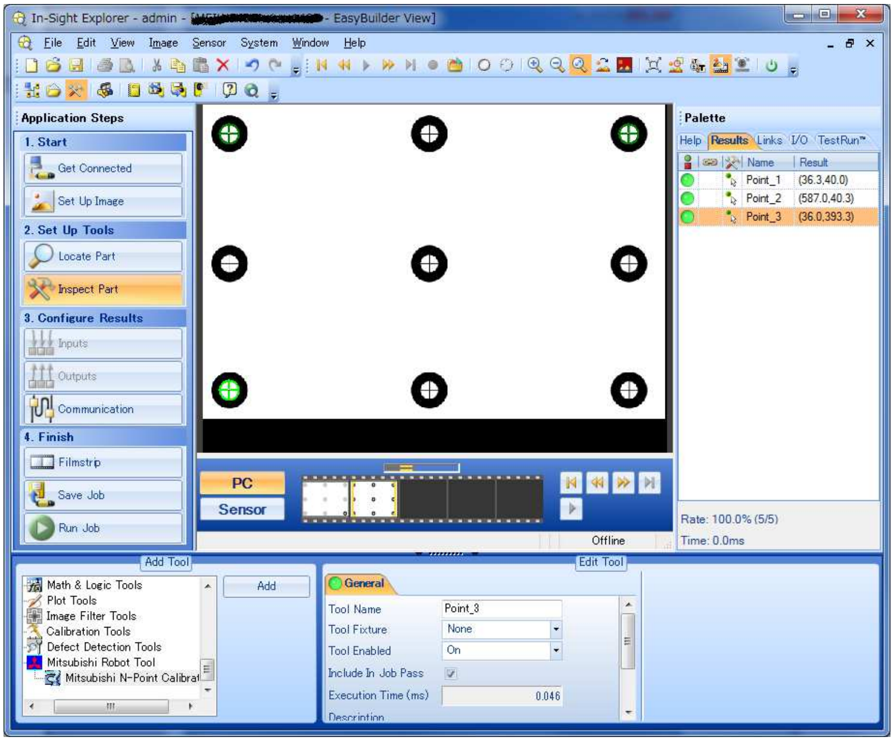
6) Click [Add] button. Then, the cross sign enclosed with circle on the screen is displayed. Move it to the mark of the calibration seat, and click [OK] button.



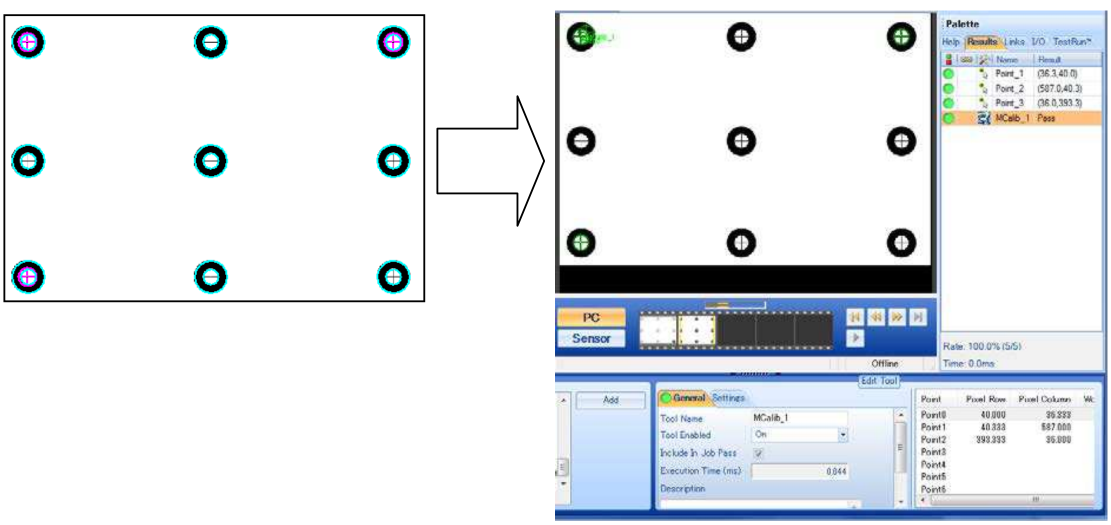
7) Specify the “User-Defined point” in three points or more repeating the above-mentioned work.



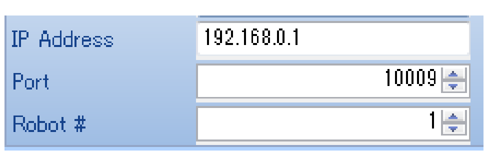
8) Select [Mitsubishi Robot Tool] – [Mitsubishi N-point calibration] in “Add Tool” column of this tool.




9) Click [Add] button. Select “User-Defined point” three points specified ahead from nine displayed marks. Then, Click [OK] button.

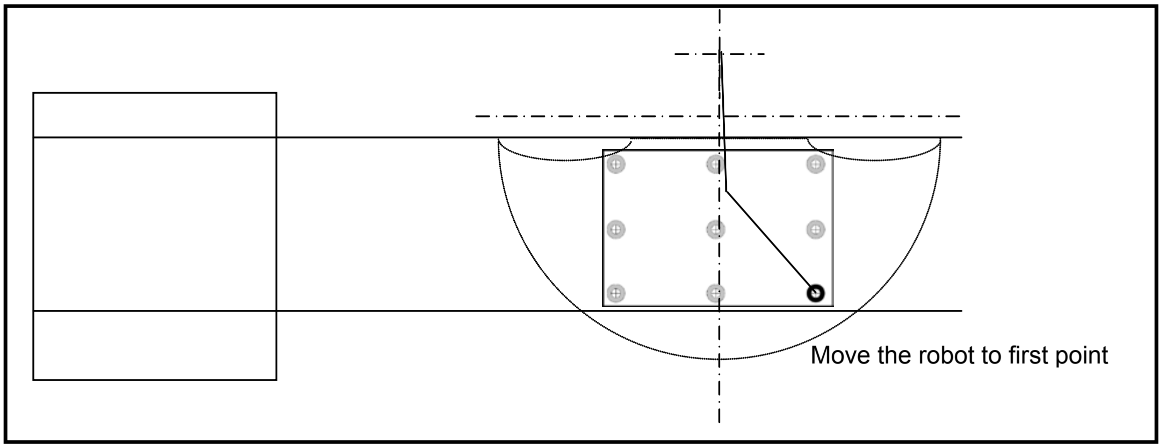


10) Open the [Settings] tab screen from the “Edit Tool”, and input IP address set to "Robot IP address".

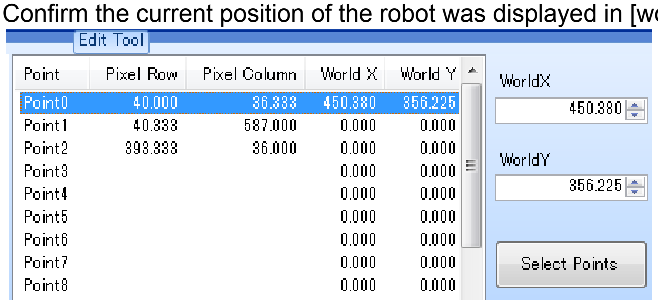



11) Make the vision sensor online.

12) Move the calibration seat by starting the conveyer within the robot movement range.
13) Move the robot to the position right above the first mark on the conveyer.



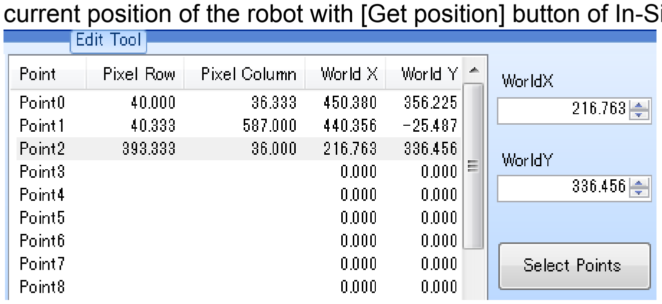

14) Click [Get position] button in “Edit Tool” column of In-Sight Explorer.
Confirm the current position of the robot was displayed in [world X] and [world Y].



| Point | Pixel Row | Pixel Column | World X | World Y |
|---|---|---|---|---|
| Point0 | 40.000 | 36.333 | 450.380 | 356.225 |
| Point1 | 40.333 | 587.000 | 0.000 | 0.000 |
| Point2 | 393.333 | 36.000 | 0.000 | 0.000 |
| Point3 | | | 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 |



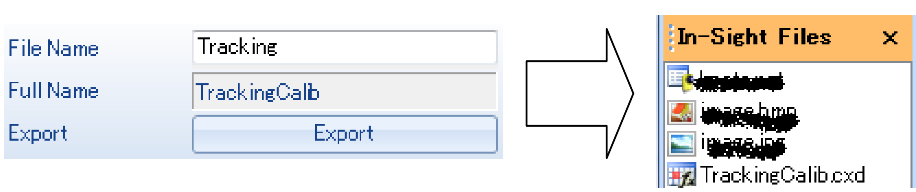
15) Similarly, move the robot hand to the mark of the second point and the third point, and acquire the current position of the robot with [Get position] button of In-Sight Explorer.



| Point | Pixel Row | Pixel Column | World X | World Y |
|---|---|---|---|---|
| Point0 | 40.000 | 36.333 | 450.380 | 356.225 |
| Point1 | 40.333 | 587.000 | 440.356 | -25.487 |
| Point2 | 393.333 | 36.000 | 216.763 | 336.456 |
| Point3 | | | 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 |



16) Input an arbitrary name to "File name" in the tool edit column of In-Sight Explorer, and click the export button. And, confirm the calibration file of the specified name was made in the vision sensor.




17) Raise the robot.
*** With this operation, encoder data is acquired.**

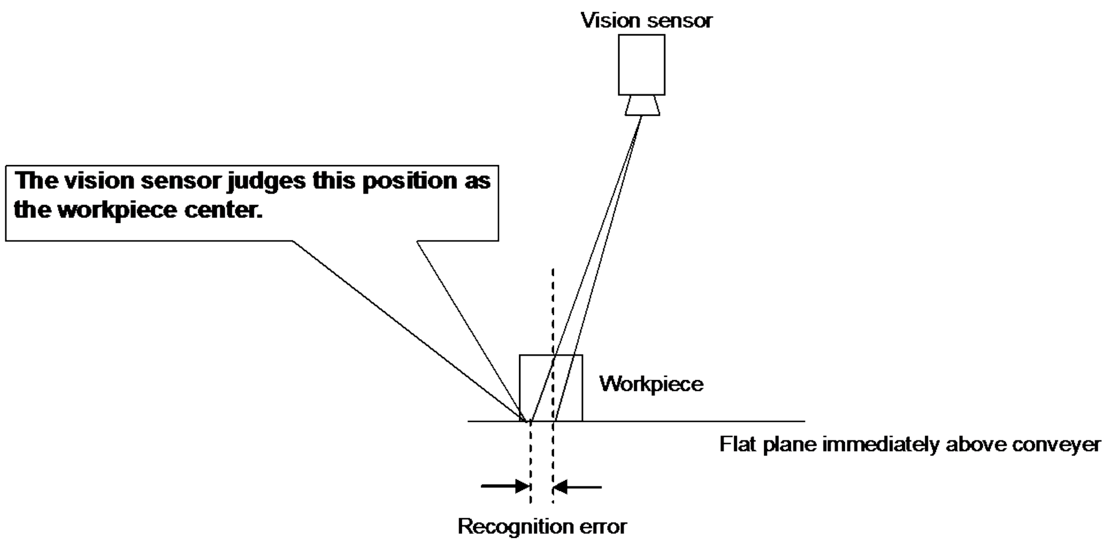
## 14.3. Confirmation after operation

Check the value of “M_100()” using T/B.
Enter the **encoder number** in the array element.
Confirm that the differences between the encoder values acquired on the vision sensor side and the encoder values acquired on the robot side are set in “M_100().”

### ⚠ CAUTION

***If precision is highly important, use four workpieces instead of marking stickers to specify 4 points at which they are grabbed.***

When marking stickers are used, a vision sensor calculates the robot position on a flat plane immediately above the conveyer. If the workpiece height is large, the robot coordinate values may deviate from the actual workpiece center displayed when the center of the workpiece is recognized.
For this reason, it is recommended to calibrate the robot using workpieces in order to make sure that the robot calculates the coordinates correctly, based on a flat plane immediately above the workpieces.



# 15. Workpiece Recognition and Teaching ("C1" program)

This chapter explains the tasks carried out by using "C1" program.

* **"C1" program contains operations required for both conveyer tracking and vision tracking, but different operations are performed. Refers to "15.1Program for Conveyer Tracking" for operations in the case of conveyer tracking and "15.2Program for Vision Tracking" for operations in the case of vision tracking.**

Please refer to "Detailed Explanations of Functions and Operations" for the steps involved in each operation.

## 15.1. Program for Conveyer Tracking

In "C1" program for conveyer tracking, encoder data at the positions where a sensor is activated and where the robot suctions a workpiece is acquired so that the robot can recognize the workpiece coordinates when the sensor is activated at later times.

The operation procedure and items to be confirmed after operation in "C1" program for conveyer tracking are explained below.

### (1) Operation procedure

1) Open "C1" program using T/B.
2) Set the controller mode to "MANUAL". Set the T/B to "ENABLE".

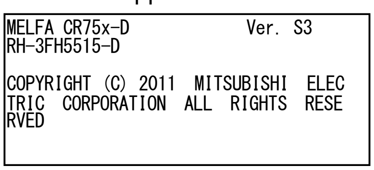
O/P

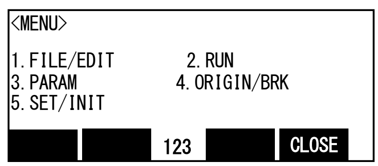

T/B

Up : DISABLE
Down : ENABLE
*Lamp lighting

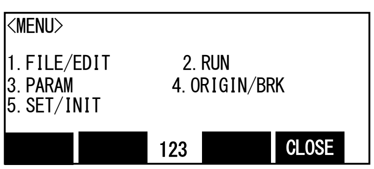
T/B rear

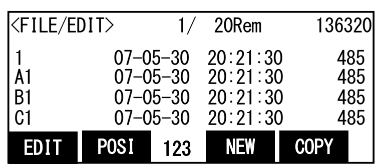
3) Press one of the keys (example, [EXE] key) while the <TITLE> screen is displayed. The <MENU> screen will appear.



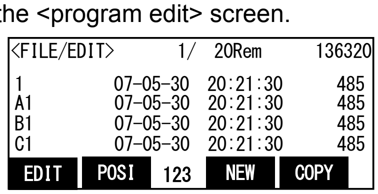



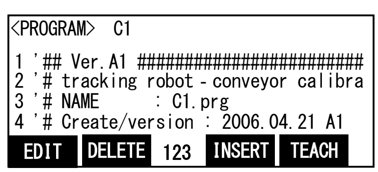

4) Select "1. FILE /EDIT" screen on the <MENU > screen.



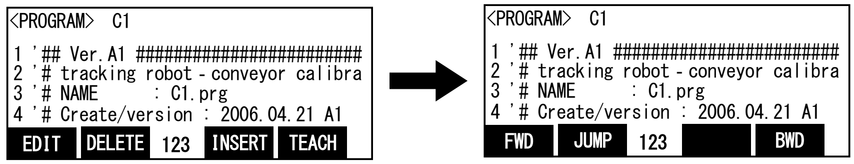




5) Press the arrow key, combine the cursor with the program name "C1" and press the [EXE] key. Display the <program edit> screen.

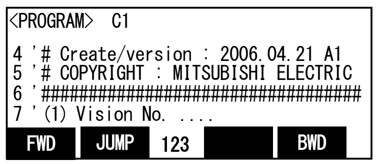


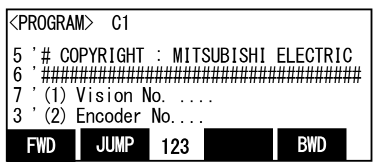
6) Press the [FUNCTION] key, and change the function display



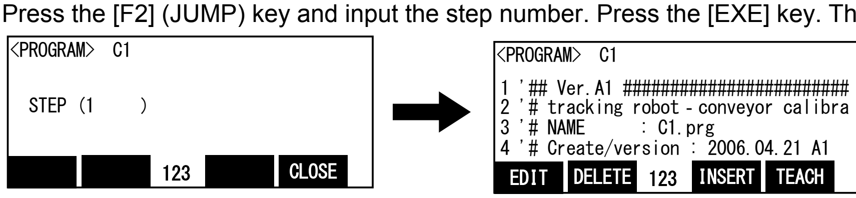

7) Press the [F1] (FWD) key and execute step feed. "(1)Vision No ......." is displayed



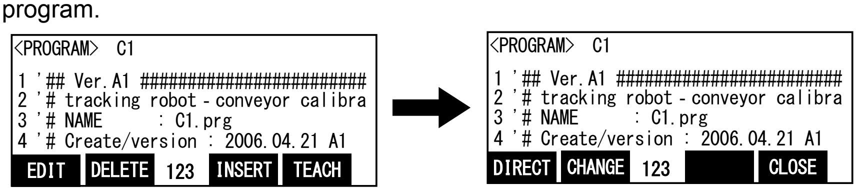

8) Work according to the comment directions in the robot program.
9) Next "' (2) Encoder No.. Execute step feed to ".




10) Repeat (7) - (8) and execute step feed to "End."
11) Press the [F2] (JUMP) key and input the step number. Press the [EXE] key. Then returns to first step




12) Press the [FUNCTION] key, and change the function display. Press the [F4] (close) key and close the program.



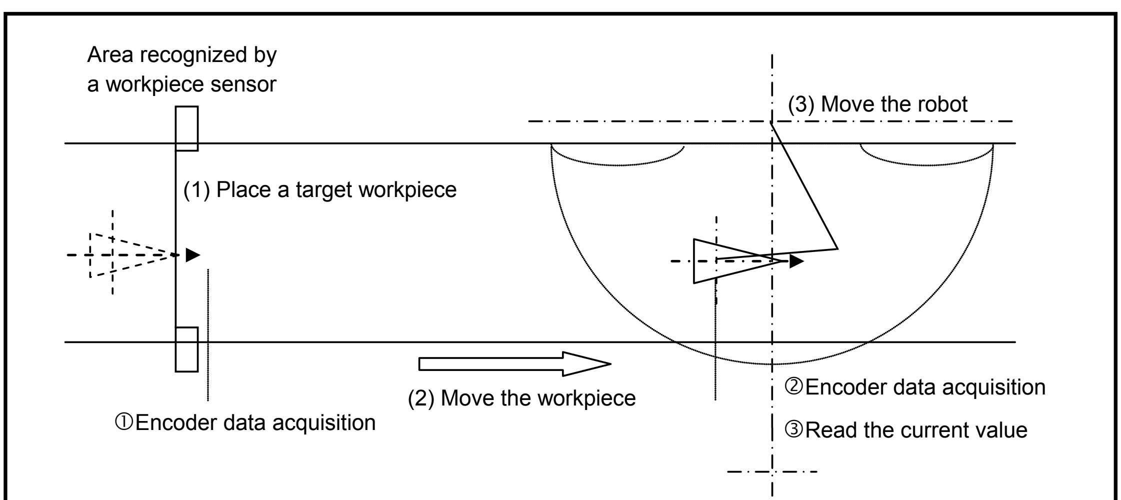
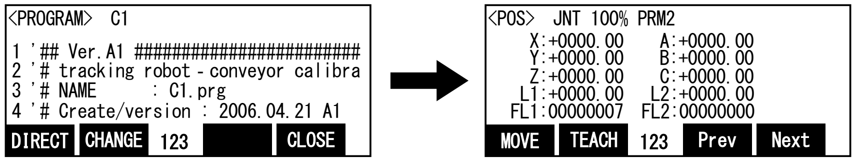




**Figure 15–1 Operation for Matching Workpiece Coordinates and Robot Coordinates**

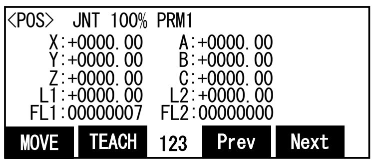
**(2) Tasks**

1) Enter the model number, encoder number and number of the sensor that monitors the workpieces in the X, Y and Z coordinates of the position variable “**PRM1**” in the program.

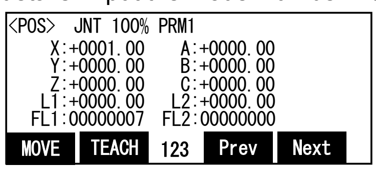
(a) Press the function key ([F2]) corresponding to "the change", and display the position edit screen.



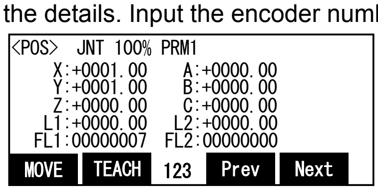

(b) The [F3] (Prev) key or the [F4] (Next) key is pressed, change the target variable, and display "PRM1" on the position name.




(c) X coordinates are selected by the arrow key, press the [CLEAR] key for a long time, and delete the details. Input the model number into X coordinates.

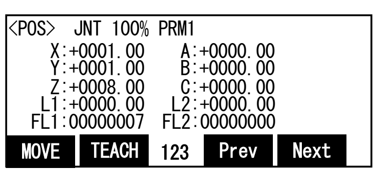



(d) Y coordinates are selected by the arrow key, press the [CLEAR] key for a long time, and delete the details. Input the encoder number into Y coordinates.



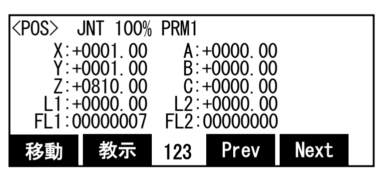

(e) Z coordinates are selected by the arrow key, press the [CLEAR] key for a long time, and delete the details. Input the number of the sensor that monitors the workpieces into Z coordinates.

SD series



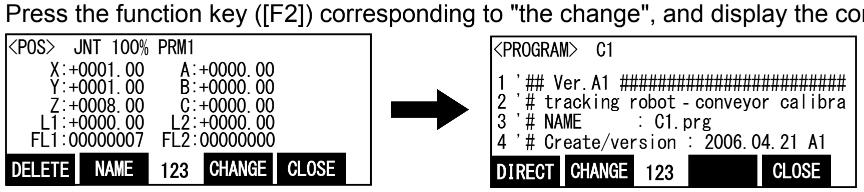

Example) Input signal number is 8

SQ series




Example)Traking enable signal number is 810.

(f) Press the function key ([F2]) corresponding to "the change", and display the command edit screen.



2) Move a workpiece to the location where the sensor is activated.
   *** With this operation, encoder data is acquired.**
3) Drive the conveyer to move the workpiece within the robot movement range.
4) Move the robot to the position where it suctions the workpiece.
   *** With this operation, encoder data and robot position are acquired.**
5) Perform step operation until "End."
   *** With this operation, the robot is able to calculate the position of a workpiece as soon as the sensor is activated.**

**(3) Confirmation after operation**

Confirm the values of "M_101()," "P_100()" and "P_102()" using T/B.
Enter **encoder numbers** in array elements.

- "M_101()": Differences between the encoder values acquired at the position of the photoelectronic sensor and the encoder values acquired on the robot side.
- "P_100()": Position at which workpieces are suctioned
- "P_102()": The value of the variable "PRM1" set in step (1)

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

## 15.2. Program for Vision Tracking

Vision tracking “C1” program acquires encoder data at the position where the vision sensor recognizes workpieces and where the robot suctions workpieces such that the robot can recognize the work coordinates recognized by the vision sensor. The following explains the operation procedure and items to confirm after operation in vision tracking “C1” program.

**(1) Operation procedure**

1) Register workpieces to be recognized by a vision sensor and create a vision program.
   Please refer to "In-Sight Explorer manual" for the method of making the vision program.
2) Open “C1” program using T/B.
3) Set the controller mode to "MANUAL".  Set the T/B to "ENABLE".

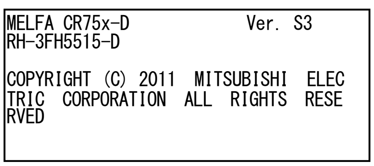
O/P

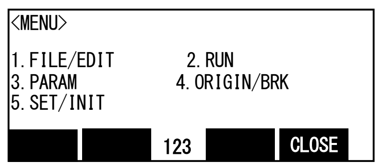

T/B

Up　:DISABLE
Down:ENABLE
★Lamp lighting

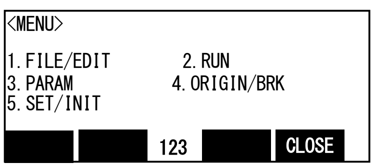
T/B rear

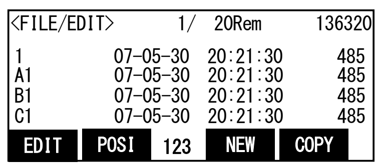
4) Press one of the keys (example, [EXE] key) while the <TITLE> screen is displayed. The <MENU> screen will appear.



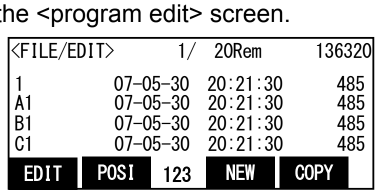



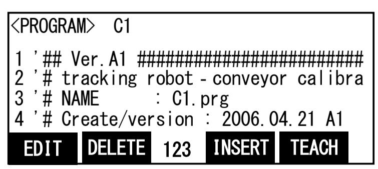

5) Select "1. FILE /EDIT" screen on the <MENU > screen.




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



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

```
<PROGRAM>  C1
1 '## Ver.A1 ########################
2 '# tracking robot-conveyor calibra
3 '# NAME      : C1.prg
4 '# Create/version : 2006.04.21 A1
EDIT DELETE 123 INSERT TEACH
```

→

```
<PROGRAM>  C1
1 '## Ver.A1 ########################
2 '# tracking robot-conveyor calibra
3 '# NAME      : C1.prg
4 '# Create/version : 2006.04.21 A1
FWD JUMP 123  BWD
```

8) Press the [F1] (FWD) key and execute step feed. "(1)Vision No ......." is displayed

```
<PROGRAM>  C1
4 '# Create/version : 2006.04.21 A1
5 '# COPYRIGHT : MITSUBISHI ELECTRIC
6 '##################################
7 '(1) Vision No. ....
FWD JUMP 123  BWD
```

9) Work according to the comment directions in the robot program.
10) Next "' (2) Encoder No.. Execute step feed to ".

```
<PROGRAM>  C1
5 '# COPYRIGHT : MITSUBISHI ELECTRIC
6 '##################################
7 '(1) Vision No. ....
3 '(2) Encoder No....
FWD JUMP 123  BWD
```

11) Repeat (7) - (8) and execute step feed to "End."
12) Press the [F2] (JUMP) key and input the step number. Press the [EXE] key. Then returns to first step

```
<PROGRAM>  C1

  STEP (1     )

  123  CLOSE
```

→

```
<PROGRAM>  C1
1 '## Ver.A1 ########################
2 '# tracking robot-conveyor calibra
3 '# NAME      : C1.prg
4 '# Create/version : 2006.04.21 A1
EDIT DELETE 123 INSERT TEACH
```

13) Press the [FUNCTION] key, and change the function display. Press the [F4] (close) key and close the program.

```
<PROGRAM>  C1
1 '## Ver.A1 ########################
2 '# tracking robot-conveyor calibra
3 '# NAME      : C1.prg
4 '# Create/version : 2006.04.21 A1
EDIT DELETE 123 INSERT TEACH
```

→

```
<PROGRAM>  C1
1 '## Ver.A1 ########################
2 '# tracking robot-conveyor calibra
3 '# NAME      : C1.prg
4 '# Create/version : 2006.04.21 A1
DIRECT CHANGE 123  CLOSE
```

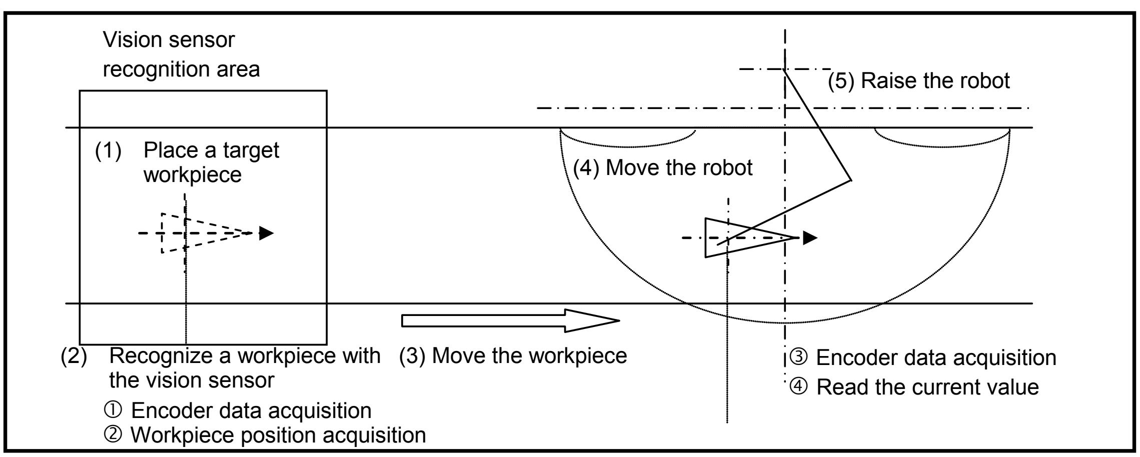
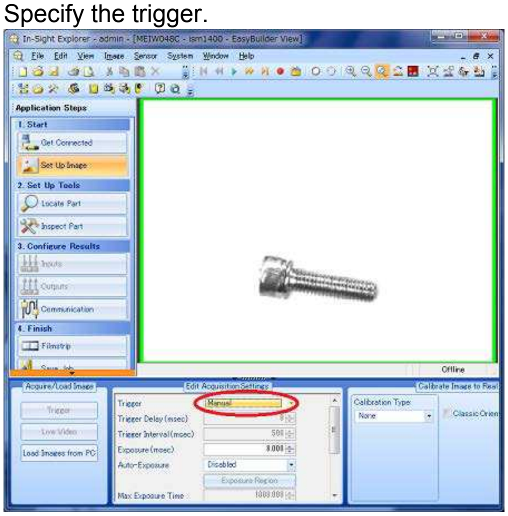
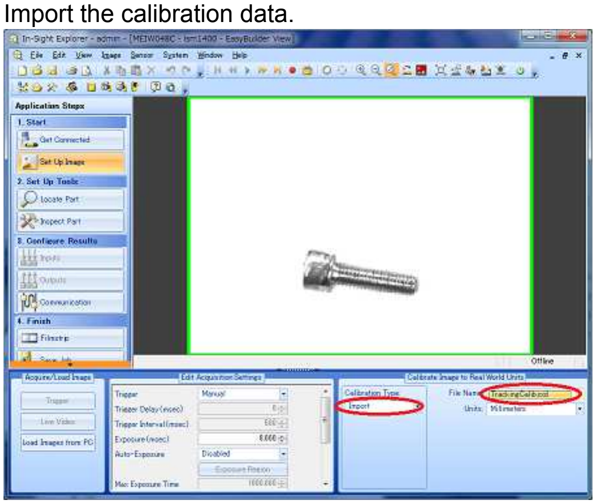
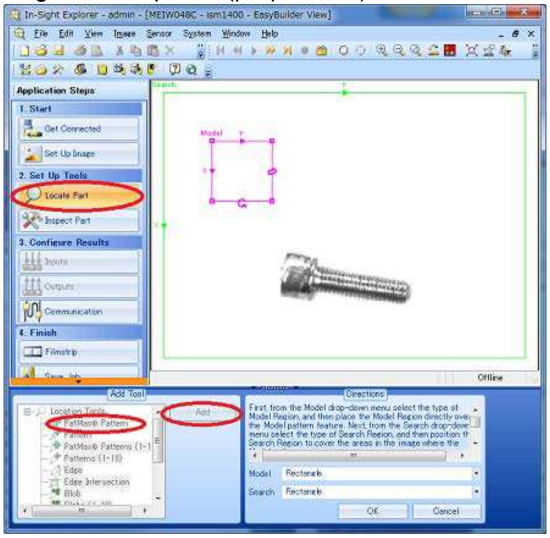
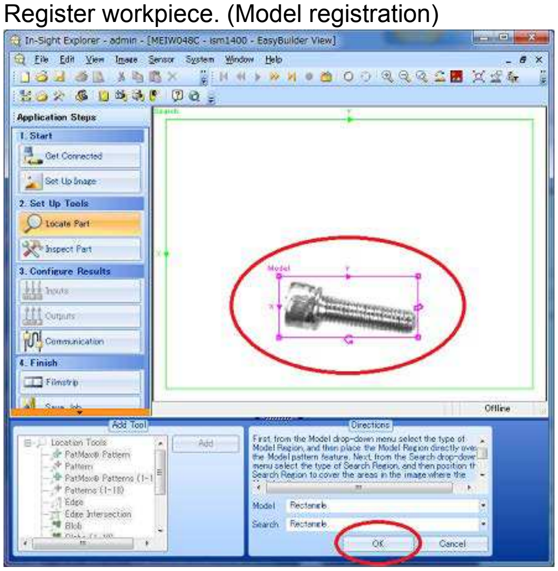
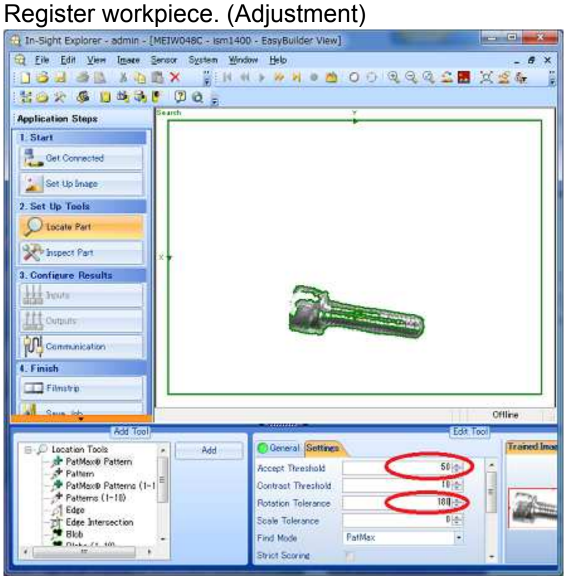


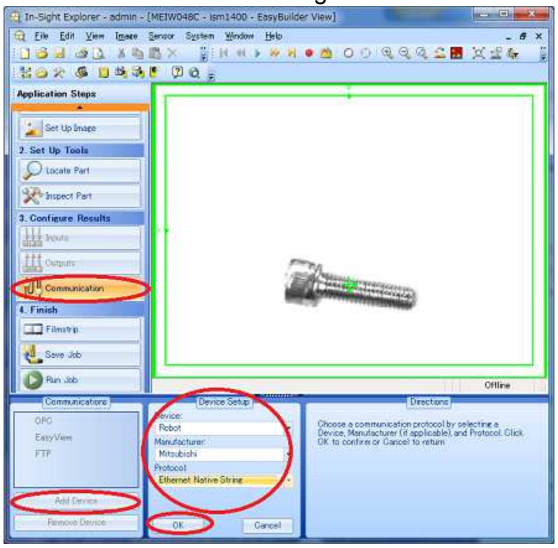
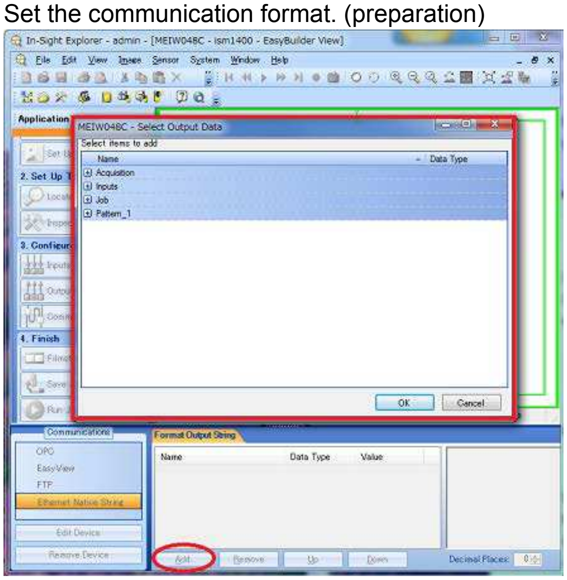
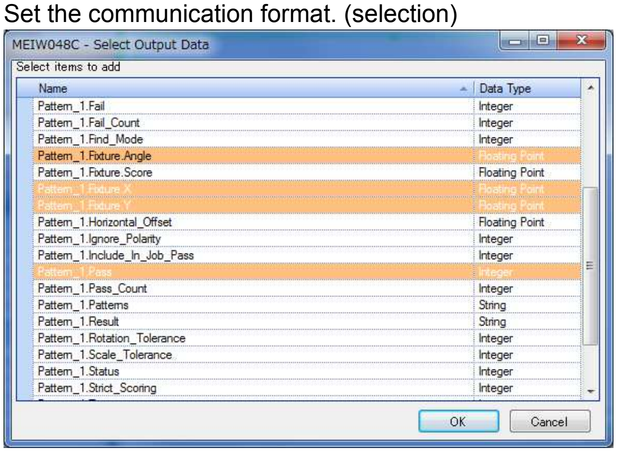

**Figure 15–2 Operation for Matching Workpiece Coordinates and Robot Coordinates**

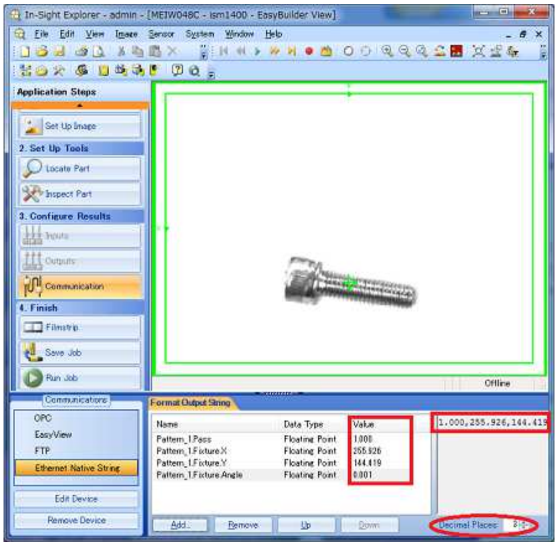
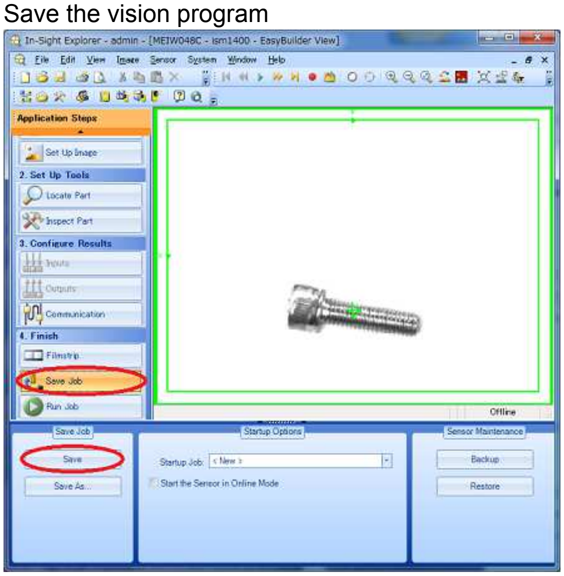
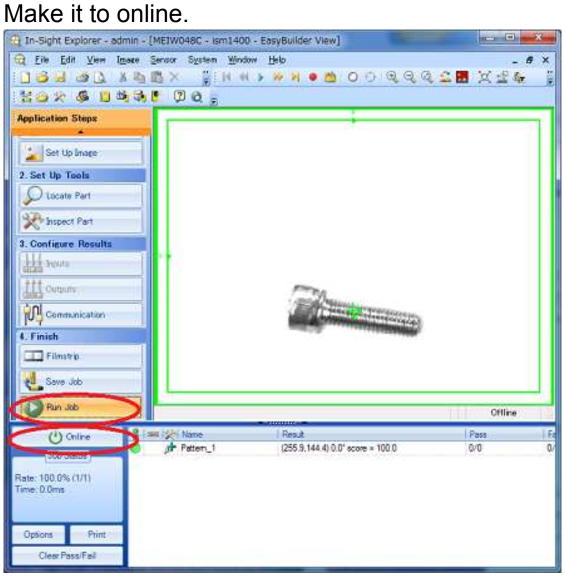
**(2) Tasks**

1) Make the vision program.

| | |
|---|---|
| Take picture of workpiece.<br> | Select [File] – [New Job] from the menu.<br><br>Click [Set Up Image] button from “Application Steps”.<br><br>Click [Live Video] button.<br><br>Take picture of workpiece that does the tracking.<br><br>Again, stop a live image clicking [Live Video] button. |
| Specify the trigger.<br> | **Change [Trigger] from "Camera" to "Manual”.**<br><br>8640(The image trigger is abnormal) error occurs when the robot controller outputs the taking picture demand to the vision sensor when you do not change. |
| Import the calibration data.<br> | In [Calibration type], select "Import".<br><br>In [File Name], select “TrackingCalib.cxd" registered when working about the B1 program. |

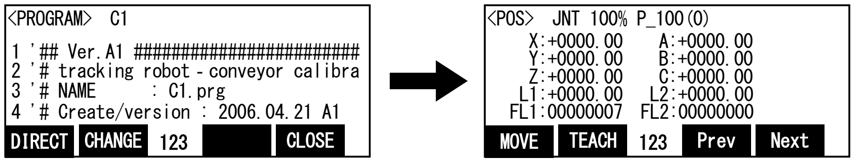
| | |
|---|---|
| Register workpiece. (preparation)<br> | Click [Locate Part] from “Application Steps”.<br><br>Select "PatMax Pattern" from “Add Tool”, and click [Add] button. |
| Register workpiece. (Model registration)<br> | Move the displayed "Model" frame, and enclose workpiece.<br><br>Click [OK] button in “Directions”. |
| Register workpiece. (Adjustment)<br> | Click [Settings] tab from "Edit Tool", and change the [Rotation Tolerance] value to “180".<br><br>(The vision sensor can recognize workpiece up to ±180 degrees. )<br><br>Change the [Accept Threshold], and adjust the recognition rate of workpiece. |

| | |
|---|---|
| Do the communication setting.<br> | Click [Communication] from “Application Steps”.<br><br>Click [Addi Device] from "Communications".<br><br>Select the following from "Device Setup".<br><br>[Device:] "Robot"<br>[Manufacturer:] "Mitsubishi”<br>[Protocol:] "Ethernet Native String"<br><br>Click [OK] button. |
| Set the communication format. (preparation)<br> | Click [Add] button from "Format Output String".<br>-> "Select Output Data" screen opens. |
| Set the communication format. (selection)<br>  | Click [+] sign of “Pattern_1”, and select it in the following order while pushing the [Ctrl] key.<br><br>(1) Pattern_1. Pass<br>(2) Pattern_1. Fixture.X<br>(3) Pattern_1. Fixture.Y<br>(4) Pattern_1. Fixture.Angle<br><br>Click [OK] button. |

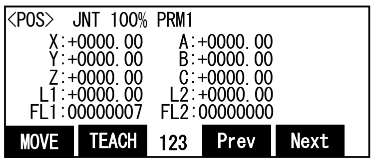
| | |
|---|---|
| Confirmation of communication format<br> | Confirm the value enclosed with a square frame.<br><br>Data sent to the robot controller is shown in a right square frame.<br><br>Change the value of [Decimal Places], and change the number of decimal positions of transmitted data. |
| Save the vision program<br> | Click [Save Job] from “Application Steps”.<br><br>Click [Save] from "Save Job".<br><br>Make the name of the job that saves it **<u>“TRK"</u>**.<br><br>Change the line of “CPRG$=" C1 program when not assuming “TRK". |
| Make it to online.<br> | Click [Run Job] from “Application Steps”.<br><br>Click [Online] on "Job Status". |

2) Enter the model number and encoder number in the X and Y coordinates of the position variable “**PRM1**” in the program.

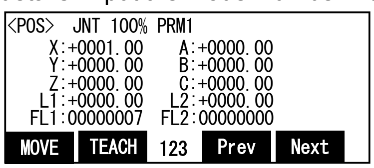
   (a) Press the function key ([F2]) corresponding to "the change", and display the position edit screen.



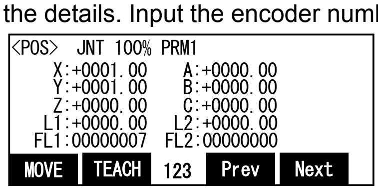

   (b) The [F3] (Prev) key or the [F4] (Next) key is pressed, change the target variable, and display "PRM1" on the position name.



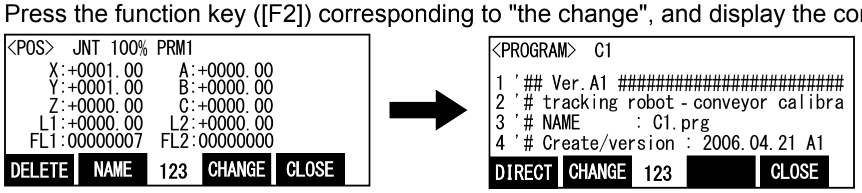

   (c) X coordinates are selected by the arrow key, press the [CLEAR] key for a long time, and delete the details. Input the model number into X coordinates.




   (d) Y coordinates are selected by the arrow key, press the [CLEAR] key for a long time, and delete the details. Input the encoder number into Y coordinates.



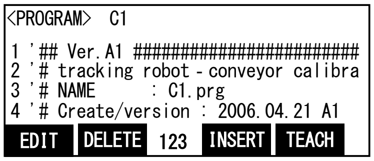

   (f) Press the function key ([F2]) corresponding to "the change", and display the command edit screen.



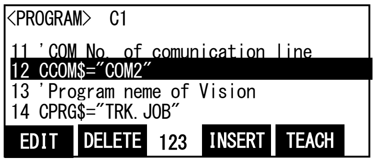

3) Start In-Sight Explorer and make the vision sensor into the off-line. Select the [Live Video] of “Set Up Image” in “Application Steps” Menu and display the picture which the vision sensor picturized on real time. Check the images and set the field of vision in the moving direction of the conveyer (mm) and the length of workpieces detected by the vision sensor (length in the moving direction of the conveyer) in the **X and Y** coordinates of the position variable “**PRM2**” in the program, respectively.

   (a) Open the [Position data Edit] screen.
   (b) Display “PRM2” at the position name.
   (c) Enter the field of vision in the moving direction of the conveyer (mm) in the X coordinate.
   (d) Enter the workpiece length detected by the vision sensor (length in the moving direction of the conveyer (mm)) in the Y coordinate.
   (d) Return to the [Command edit] screen.

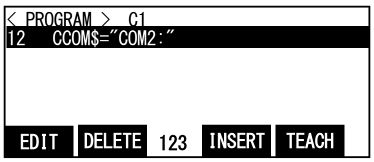
4) Specify a communication line to be connected with the vision sensor.
   (a) Open the [Command edit] screen.

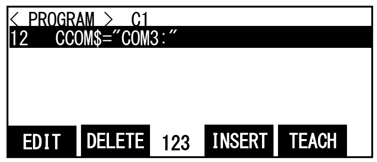



   (b)Display the command step shown in the following

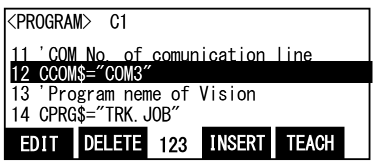



   (c) Press [F1] (edit) key and specify the line opened for the robot controller may connect with the vision sensor to the variable "CCOM$."
   example) Open COM3:




   (d)Press the [EXE] key and edit is fixed.




5) Specify a vision program to be started.
   In the same way as in step 3), change the vision program name entered after “CPRG$=” in the program.
6) Place a workpiece to be recognized within the area that the vision sensor can recognize.
7) Using In-Sight Explorer, place the vision sensor in the online status.

8) Using T/B, close the opened “C1” program once and then run the modified “C1” program automatically with the robot controller.

Note) When your controller has no operation panel, use the dedicated external signals corresponding to the following step to operate the robot.
Although the image of the operation panel is the CRnD-700 controller, the operation method is the same in other controllers.

T/B disabled

Set the T/B [ENABLE] switch to "DISABLE".

Controller enabled


Set the controller [MODE] switch to "AUTOMATIC".

**Servo ON**


Press the [SVO ON] key, the servo will turn ON, and the SVO ON lamp will light.

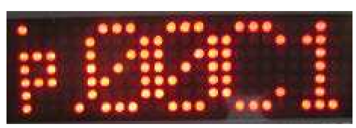
**Selection of a program number**

Display of a program number



Press the [CHNG DISP] key and display "PROGRAM NO." on the STATUS NUMBER display.

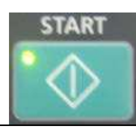
Selection of a program number

Press the [UP] or the [DOWN] key and display program name"C1"

**Start of automatic operation**

Start




Press the [START] key.

After automatic operation, “C1” program automatically stops and the LED of the [STOP] button is turned on. Open “C1” program again with T/B. Press the [F1](FWD) key to display the subsequent operation messages.

* **With this operation, encoder data and workpiece position recognized by the vision sensor are acquired.**

9) Rotate the conveyer forward and move a workpiece within the vision sensor recognition area into the robot movement range.
10) Move the robot to the position where it is able to suction the workpiece.
    * **With this operation, encoder data and robot position are acquired.**
11) Perform step operation until “End.”
    * **With this operation, the robot becomes able to recognize the position of the workpiece recognized by the vision sensor.**

**(3) Confirmation after operation**

Check the values of the following variables using T/B.
Enter the model number for the array number.

- Value of “M_101()”: Differences between encoder values when a workpiece is within the vision sensor area and when the workpiece is on the robot side
- Value of “P_102()”: Data in the variable “PRM1” (model number/encoder number)
- Value of “P_103()”: Data in the variable “PRM2” (recognition field of image view/workpiece size)
- Value of “C_100$()”: COM number
- Value of “C_101$()”: Vision program name

Confirm that each of the above values is entered.

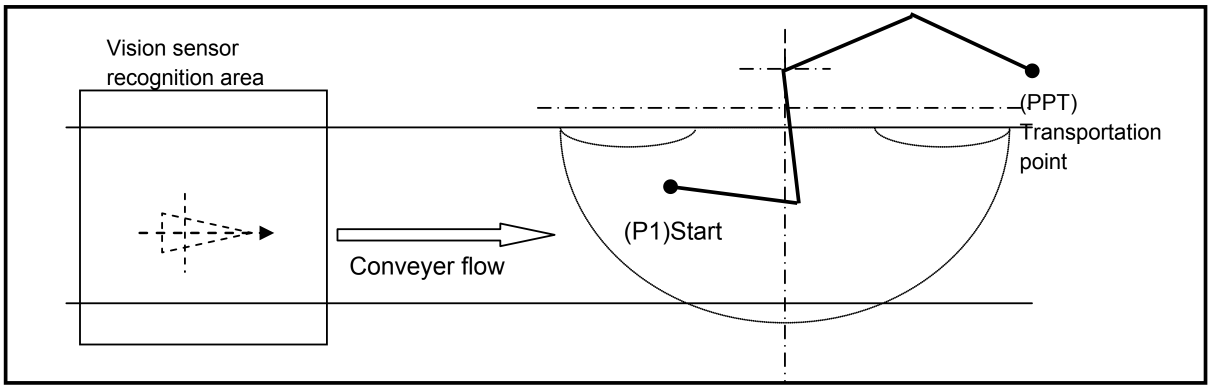
# 16. Teaching and Setting of Adjustment Variables (“1” Program)

This chapter explains operations required to run “1” program.
*** “1” program settings are required for both conveyer tracking and vision tracking.**
“1” program instructs the robot to follow and grab workpieces recognized by a photoelectronic sensor or vision sensor and transport the workpieces.
The teaching positions required by “1” program are explained below, along with how to set adjustment variables prepared in the program.

## 16.1. Teaching

The teaching of "Starting point position (position in which it is waited that workpiece arrives)" and "Transportation destination (position in which the held workpiece is put)" is executed.

For instance, the teaching does the following positions.




Teach the origin position and transportation destination. The following explains how to perform these operations.

1) Open “1” program using T/B.
2) Open the [Position data Edit] screen.
3) Display “P1” in order to set the robot origin position when the system is started.
4) Move the robot to the origin position and teach it the position.
5) Display “PPT” in order to set the transportation destination position (the location where workpieces are placed).
6) Move the robot to the transportation destination and teach it the position.

Confirm whether workpiece can be transported at the position in which the teaching was done.

7) Display “P1" at the starting point position on the [Position data Edit] screen. Turn on the servo by gripping the deadman switch.
8) Move the robot to the position of “P1" pushing F1 (MOVE).

```
<POS>  JNT 100% P1
    X:  +300.00    A:+0000.00
    Y:  +500.00    B:   +90.00
    Z:  +400.00    C:  +150.00
   L1:+0000.00    L2:+0000.00
  FL1:00000007   FL2:00000000
 MOVE  TEACH  123  Prev  Next
```

9) Move the robot to an arbitrary position (position in which workpiece flows) by the jog operation.
10) Display “PPT" at the transportation point position on the [Position data Edit] screen. Turn on the servo by gripping the deadman switch.

```
<POS>  JNT 100% PPT
    X:   +50.00    A:+0000.00
    Y:  +500.00    B:   +90.00
    Z:  +400.00    C:   +45.00
   L1:+0000.00    L2:+0000.00
  FL1:00000007   FL2:00000000
 MOVE  TEACH  123  Prev  Next
```

11) Move the robot to the position of “PPT" pushing F1 (MOVE).

## 16.2. Setting of adjustment variables in the program

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

**Table 16–1 List of Adjustment Variables in Programs**

| Variable name | Explanation | Setting example |
|---|---|---|
| PWK | Set the model number.<br>X = model number (1 to 10) | When you set 1 to the model number:<br>(X, Y, Z, A, B, C) =(+1,+0,+0,+0,+0,+0) |
| PRI | "1" program and "CM1" program are run simultaneously (multitasking). "1" program moves the robot, and "CM1" program observes the sensor.<br>It is possible to specify which program is processed with a higher priority, rather than performing the same amount of processing at the same time.<br>X = Set the line numbers of "1" program to be performed (1 to 31).<br>Y = Set the line numbers of "CM1" program to be performed (1 to 31). | When you set to run "1" program by one line and run "CM1" program by 10 lines:<br>(X, Y, Z, A, B, C) = (+1,+10,+0,+0,+0,+0) |
| PUP1 | When operating by the adsorption of workpiece, set the height that the robot works.<br>Height sets the amount of elevation (mm) from the position where workpiece is adsorbed.<br>X = Amount of elevation of the position where a robot waits until a workpiece arrives. (mm)<br>Y = Amount of elevation from the workpiece suction position (before suctioning) (mm)<br>Z = Amount of elevation from the workpiece suction position (after suctioning) (mm)<br><br>*** Since the Y and Z coordinates indicate distances in the Z direction in the tool coordinate system, the sign varies depending on the robot model.** | When the following values are set:<br>Amount of elevation of the position where a robot waits until a workpiece arrives<br>: 50 mm<br>Amount of elevation from the workpiece suction position (before suctioning)<br>: -50 mm<br>Amount of elevation from the workpiece suction position (after suctioning)<br>: -50 mm<br><br>(X, Y, Z, A, B, C) = (+50,-50,-50,+0,+0,+0) |
| PUP2 | When operating in putting workpiece, set the height that the robot works.<br>Height sets the amount of elevation (mm) from the position where workpiece is adsorbed.<br>Y = Amount of elevation from the workpiece release position (before release). (mm)<br>Z = Amount of elevation from the workpiece release position (after release). (mm)<br>*** Since these values are distances in the Z direction of the tool coordinate system, the sign varies depending on the robot model.** | When the following values are set:<br>Amount of elevation from the workpiece release position (before release)<br>: -50 mm<br>Amount of elevation from the workpiece release position (after release)<br>: -50 mm<br>(X, Y, Z, A, B, C) = (+0,-50,-50,+0,+0,+0) |
| PAC1 | When operating by the adsorption of workpiece, the acceleration and the deceleration when moving to the position on the workpiece are set.<br>X = The acceleration until moving to the position on the workpiece. (1 to 100) (%)<br>Y = The deceleration until moving to the position on the workpiece. (1 to 100) (%)<br>*** The value set by X coordinates and Y coordinates of "PAC*" is used for <acceleration ratio(%)> of the Accel instruction and <deceleration ratio(%)>.<br>The value is reduced when the speed of time when the robot vibrates and the robot is fast.** | When the following values are set:<br>Acceleration until moving to the position on the workpiece. : 100%<br>Deceleration until moving to the position on the workpiece. : 100%<br>(X, Y, Z, A, B, C) =<br>(+100,+100,+0,+0,+0,+0) |

| | | |
|---|---|---|
| PAC2 | When operating by the adsorption of workpiece, the acceleration and the deceleration when moving to the workpiece suction position are set.<br>X = The acceleration until moving to the workpiece suction position. (1 to 100) (%)<br>Y = The deceleration until moving to the workpiece suction position. (1 to 100) (%) | When the following values are set:<br>Acceleration until moving to the workpiece suction position. : 10%<br>Deceleration until moving to the workpiece suction position. : 20%<br>(X, Y, Z, A, B, C) =(+10,+20,+0,+0,+0,+0) |
| PAC3 | When operating by the adsorption of workpiece, the acceleration and the deceleration when moving toward the position on the workpiece are set.<br>X = The acceleration until moving to the position on the workpiece. (1 to 100) (%)<br>Y = The deceleration until moving to the position on the workpiece. (1 to 100) (%) | When the following values are set:<br>Acceleration until moving to the position on the workpiece. : 50%<br>Deceleration until moving to the position on the workpiece. : 80%<br>(X, Y, Z, A, B, C) =(+50,+80,+0,+0,+0,+0) |
| PAC11 | When operating by the release of workpiece, the acceleration and the deceleration when moving to the position on the workpiece are set.<br>X = The acceleration until moving to the position release position. (1 to 100) (%)<br>Y = The deceleration until moving to the position release position. (1 to 100) (%) | When the following values are set:<br>Acceleration until moving to the position on the workpiece : 80%<br>Deceleration until moving to the position on the workpiece : 70%<br>(X, Y, Z, A, B, C) =(+80,+70,+0,+0,+0,+0) |
| PAC12 | When operating by the release of workpiece, the acceleration and the deceleration when moving to the workpiece release position are set.<br>X = The acceleration until moving to the workpiece release position. (1 to 100) (%)<br>Y = The deceleration until moving to the workpiece release position. (1 to 100) (%) | When the following values are set:<br>Acceleration until moving to the workpiece release position. : 5%<br>Deceleration until moving to the workpiece release position. : 10%<br>(X, Y, Z, A, B, C) = (+5,+10,+0,+0,+0,+0) |
| PAC13 | When operating by the release of workpiece, the acceleration and the deceleration when moving toward the position on the workpiece are set.<br>X = The acceleration until moving to the position on the workpiece. (1 to 100) (%)<br>Y = The deceleration until moving to the position on the workpiece. (1 to 100) (%) | When the following values are set:<br>Acceleration until moving to the position on the workpiece. : 100%<br>Deceleration until moving to the position on the workpiece. : 100%<br>(X, Y, Z, A, B, C) =<br>(+100,+100,+0,+0,+0,+0) |
| PDLY1 | Set the suction time.<br>X: Suction time (s). | When setting 0.5 second for the sucking time:<br>(X, Y, Z, A, B, C) = (+0.5,+0,+0,+0,+0,+0) |
| PDLY2 | Set the release time.<br>X: Release time (s). | When setting 0.3 second for the release time:<br>(X, Y, Z, A, B, C) = (+0.3,+0,+0,+0,+0,+0) |
| POFSET | When the adsorption position shifts, the gap can be corrected. Set the correction value.<br>*** The direction of the correction is a direction of the hand coordinate system. Please decide the correction value after changing the job mode to "Tool", pushing the [+X] key and the [+Y] key, and confirming the operation of the robot.** | |

<table>
<tr><td>PTN</td><td>Set the position of the robot and conveyer, and the direction where the workpiece moves.<br>X = The following values. (1 to 6)<br><table>
<tr><th>Setting value</th><th>Conveyer position</th><th>Conveyer direction</th></tr>
<tr><td>1</td><td>Front</td><td>Right to Left</td></tr>
<tr><td>2</td><td>Front</td><td>Left to Right</td></tr>
<tr><td>3</td><td>Left side</td><td>Right to Left</td></tr>
<tr><td>4</td><td>Left</td><td>Left to Right</td></tr>
<tr><td>5</td><td>Right side</td><td>Right to Left</td></tr>
<tr><td>6</td><td>Right side</td><td>Left to Right</td></tr>
</table></td><td>When a conveyer is placed in front of the robot and the workpiece moves from the left to right: (When in view of the robot)<br>(X, Y, Z, A, B, C) = (+1,+0,+0,+0,+0,+0)<br><br>The relationship between PRNG and PTN is shown in "Figure 16–3 Diagram of Relationship between Adjustment Variables "PRNG" and "PTN" in the Program".</td></tr>
<tr><td>PRNG</td><td>Set range of motion where the robot judges workpiece to be able to follow.<br>X = The start distance of the range in which the robot can follow a workpiece :(mm)<br>Y = The end distance of the range in which the robot can follow a workpiece :(mm)<br>Z = The distance in which follow is canceled :(mm)</td><td>The relationship between PRNG and PTN is shown in "Figure 16–3 Diagram of Relationship between Adjustment Variables "PRNG" and "PTN" in the Program".</td></tr>
<tr><td>P3HR</td><td>(For RH-3S*HR)<br>The singular point neighborhood can be moved in RH-3S*HR at the joint operation.<br>However, when the tracking operation passes over the singular point neighborhood for straight line operation, the J1 axis accelerates rapidly and speed limit (H213x error :x= axis number) is generated.<br>Then, the singular point neighborhood is limited to the tracking by setting this parameter.<br>X = The Time in which the robot can move over the workpiece :(ms)<br>Y = The Maximal speed for J3 axis :(mm/s)<br>Z = The radius of area made singular point neighborhood :(mm)</td><td>(X, Y, Z, A, B, C) =<br>(+800,+1500,+60,+0,+0,+0)<br><br>Refer to "Figure 16–1 Diagram of Relationship between Adjustment Variables "PRNG" and "P3HR" in the Program"</td></tr>
</table>

<Restrictions of RH-3S*HR when using the tracking function>
The RH-3S*HR can not pass over the singular adjustment point while the tracking operation.
It is necessary to avoid singular adjustment point and place the conveyer.
As shown in Figure 16–1 or Figure 16–2, If the conveyer is installed at right under the robot, the operation range of tracking must been setting out of range of singular adjustment point.

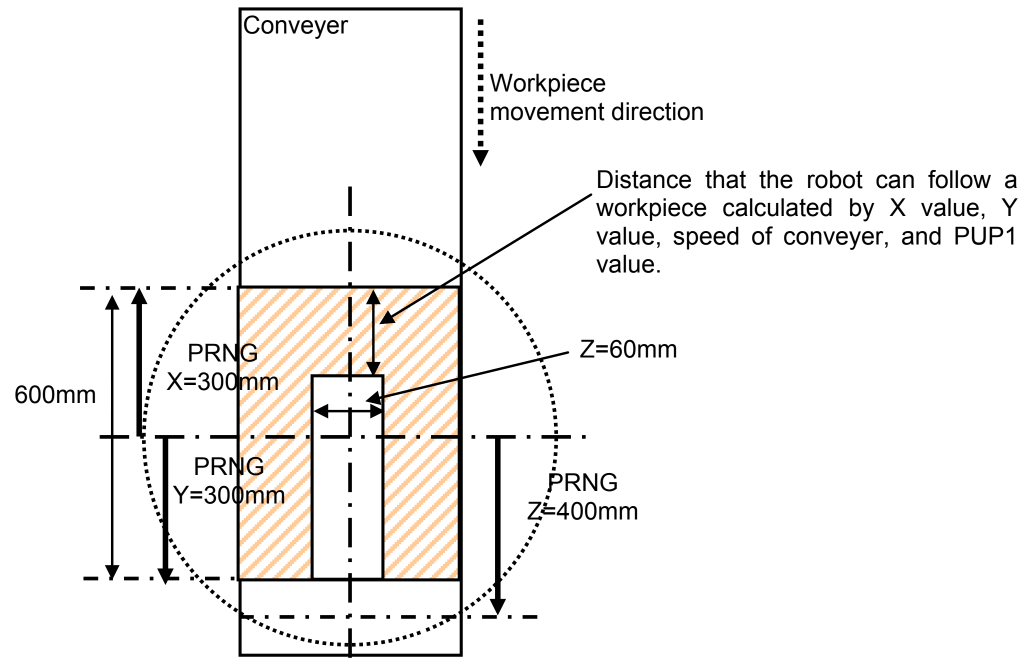
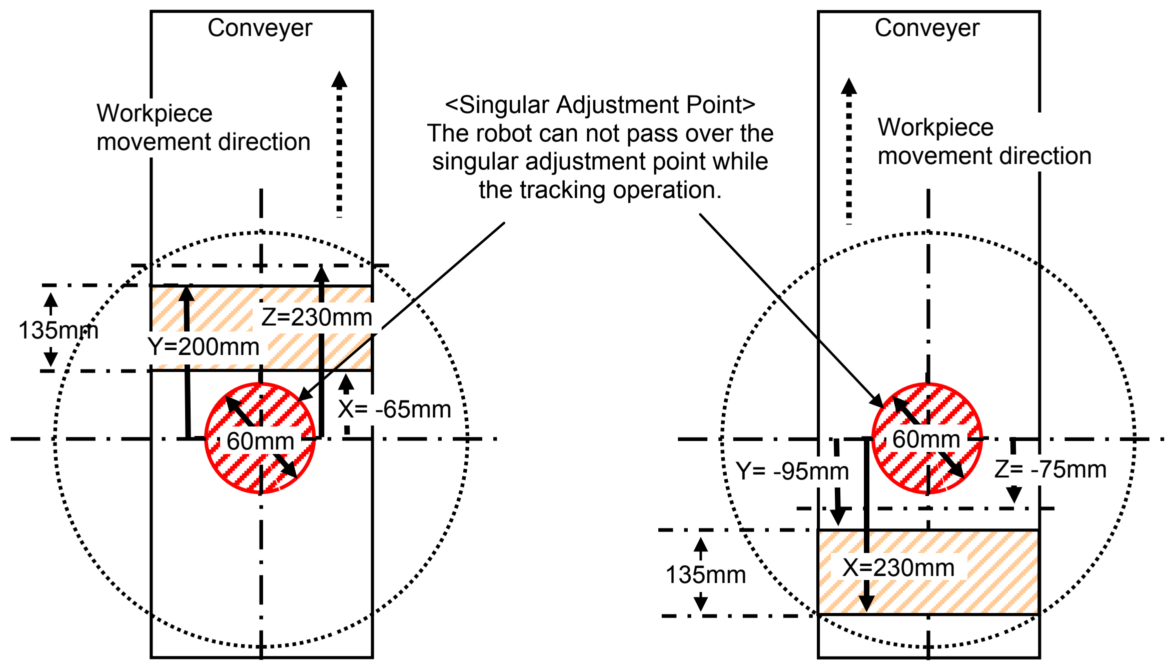



**Figure 16–1 Diagram of Relationship between Adjustment Variables “PRNG” and “P3HR” in the Program**




When the conveyer is placed at the **right under the robot** and the workpiece moves from the **rear to front** (= the X coordinate of PTN is **"5"**) and the operation range of tracking is set at front of the robot.
PRNG は、(X,Y,Z)=**(-65,+200,+230)**

When the conveyer is placed at the **right under the robot** and the workpiece moves from the **rear to front** (= the X coordinate of PTN is **"5"**) and the operation range of tracking is set at backward the robot.
PRNG は、(X,Y,Z)=**(+230,-95,-75)**

**Figure 16–2 Relationship of singular point neighborhood and tracking area**

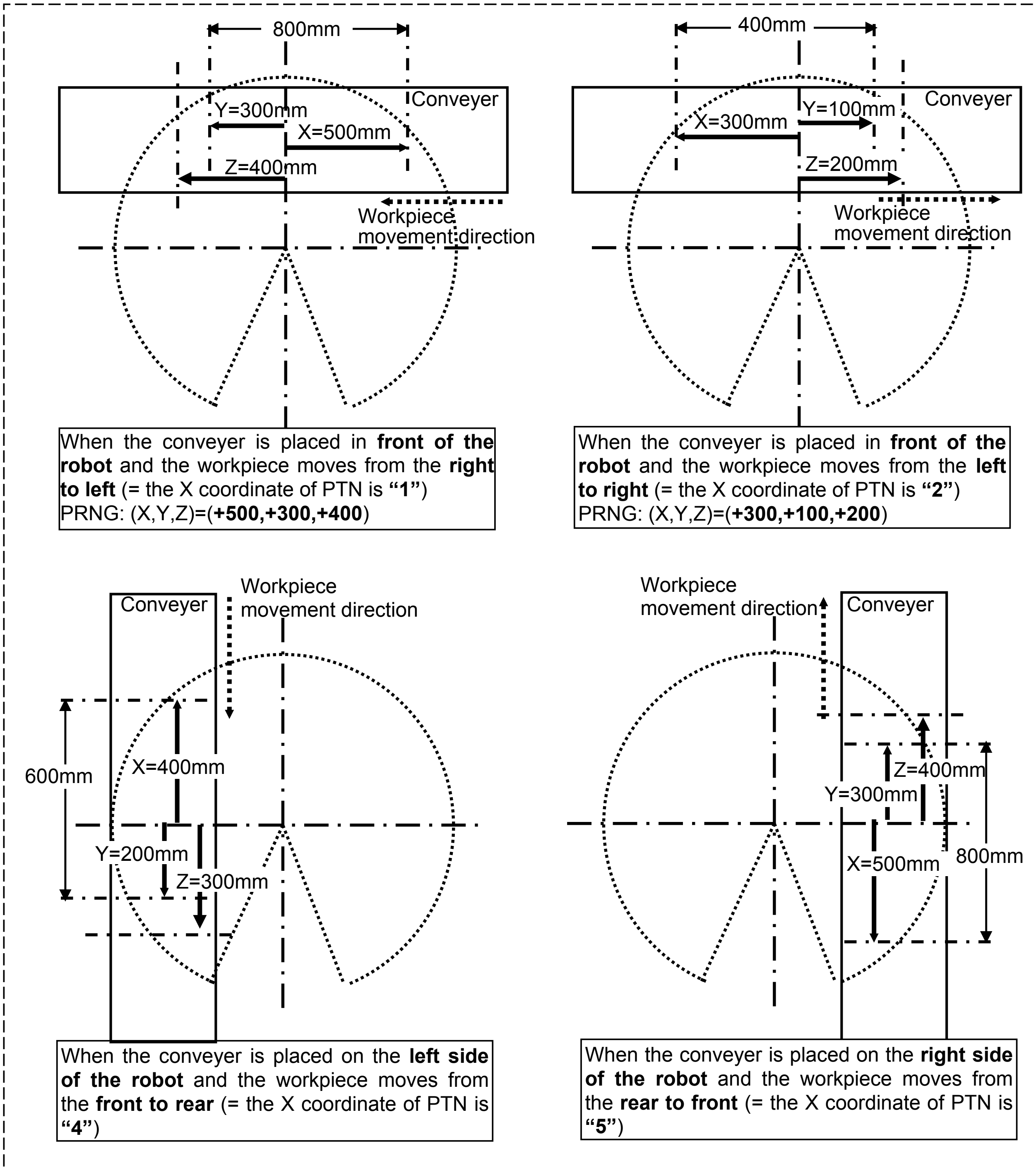



**Figure 16–3 Diagram of Relationship between Adjustment Variables "PRNG" and "PTN" in the Program**

# 17. Sensor Monitoring Program ("CM1" Program)

This chapter provides an overview of "CM1" program, which is run in parallel, when "1" program is run. Different types of "CM1" programs are used for conveyer tracking and vision tracking, and different processing is performed for them. These programs are explained in the following.

## 17.1. Program for Conveyer Tracking

"CM1" program calculates the workpiece coordinates in the robot coordinate system at the moment where a photoelectronic sensor is activated based on the following data acquired with "A1" program and "C1" program, and then stores the coordinates in the tracking buffer(Storage area to preserve data temporarily).
<Acquired data>

- Amount of robot movement per encoder pulse (P_EncDlt)
- Difference between the encoder value when a photoelectronic sensor is activated and the encoder value when teaching is performed on a robot
- Position at which the robot is taught to grab a workpiece

## 17.2. Program for Vision Tracking

"CM1" program converts the workpiece position recognized by the vision sensor to the corresponding coordinates in the robot coordinate system based on the following data acquired with "A1" program, "B1" program and "C1" program, and then stores the coordinates in the tracking buffer.
<Acquired data>

- Amount of robot movement per encoder pulse(P_EncDlt)
- Difference between the encoder value when a marking sticker is on the vision sensor side and the encoder value when the marking sticker is on the robot side
- Workpiece position recognized by the vision sensor
- Difference between the encoder value when the vision sensor recognizes a workpiece and the encoder value when teaching on the workpiece position was performed on the robot
- Position at which the robot is taught to grab a workpiece

The timing at which the vision sensor acquires images is calculated such that images of the same workpiece are taken at least once or up to twice by the following data specified in "C1" program.
<Data specified in "C1" program>

- Field of view in the conveyer movement direction
- Length of workpieces detected by a vision sensor (length in the conveyer movement direction)

**POINT**

***"1" program follows workpieces on a conveyer based on the workpiece information stored in the tracking buffer in "C" program.***

"C" program performs processing until the recognized workpiece position is stored in the tracking buffer. The workpiece information stored in the tracking buffer is read by "1" program and the robot follows workpieces on the conveyer based on the information.

# 18. Automatic Operation

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

## 18.1. Preparation

1) Check that there is no interfering object within the robot movement range.
2) Prepare to run the desired program.

Note) When your controller has no operation panel, use the dedicated external signals corresponding to the following step to operate the robot.
Although the image of the operation panel is the robot controller, the operation method is the same in other controllers.

T/B disabled

Set the T/B [ENABLE] switch to "DISABLE".

Controller enabled


Set the controller [MODE] switch to "AUTOMATIC".

**Servo ON**


Press the [SVO ON] key, the
servo will turn ON, and the SVO ON lamp will light.

**Selection of a program number**

Display of
a program number



Press the [CHNG DISP] key and display "PROGRAM NO." on the STATUS NUMBER display.

Selection of
a program number



Press the [UP] or the [DOWN] key and display program name"1."

## 18.2. Execution

1)Be sure that you are ready to press the [Emergency Stop] button of T/B in the case of any unexpected movement of the robot.
2)Run the program from the operation panel of the robot controller.

Note) The robot of the specification without the operation panel of the controller operates by the external signal corresponding to the following step.
Although the image of the operation panel is the robot controller, the operation method is the same in other controllers.

| **Start of automatic operation** | |
|---|---|
| Start  | Press the [START] key. |



## 18.3. At error occurrence

If the robot moves erroneously, refer to separate manual “Troubleshooting”.

## 18.4. Ending

The robot does not move unless a sensor that monitors workpieces is activated or a vision sensor recognizes a workpiece. Stop the flow of workpieces from the upstream and press the [STOP] button of the operation panel of the robot controller. Confirm that the [STOP] lamp is turned on.

Note) The robot of the specification without the operation panel of the controller is stopped by the external signal.

## 18.5. Adjusting method

You can confirm the follow operation by automatic driving.
Refer to "Maintenance of robot program" in Chapter 19 when you want to adjust it.
And, refer to "In such a case (improvement example)" in Chapter 20.3.

# 19. Maintenance of robot program

This chapter explains information required when maintaining the sample programs (robot program language MELFA-BASIC V and dedicated input/output signals).

## 19.1. MELFA-BASIC V Instructions

The lists of instructions, status variables and functions related to tracking operation are shown below.
Please refer to the separate manual “Detailed Explanations of Functions and Operations” for further information about MELFA-BASIC V.

### 19.1.1. List of Instructions

**Table 19–1 List of Instructions**

| Instruction name | Function |
|---|---|
| TrBase | Specify the workpiece coordinate origin of teaching data and tracking external encoder logic number. |
| TrClr | Clear the tracking data buffer. |
| Trk | Declare start and end of the tracking mode. |
| TrOut | Output signals from a general-purpose output and read the encoder values. |
| TrRd | Read workpiece data from the tracking data buffer. |
| TrWrt | Write workpiece data in the tracking data buffer. |

### 19.1.2. List of Robot Status Variables

**Table 19–2 List of Robot Status Variables**

| Variable name | Number of arrays | Function | Attribute (*1) | Data type |
|---|---|---|---|---|
| M_Enc | number of encoders 1 to 8 | External encoder data<br>External encoder data can be rewritten.<br>**If this state variable does not set parameter “TRMODE" to “1", the value becomes like “0".** | R/W | Double-precision real number |
| M_EncL | Number of encoder 1to8 | The stored encoder data<br>※ **Possible to use from R1 and S1**<br>※ **0 always returns in S1.** | R/W | Double-precision real number |
| P_EncDlt | number of encoders 1 to 8 | Amount of robot movement per encoder pulse<br>***This state variable is made by sample “A1” program.** | R/W | Position |
| M_Trbfct | buffer No. 1 to The first argument of parameter [TRBUF] | Number of data items stored in the tracking buffer | R | Integer |
| P_Cvspd | number of encoders 1 to 8 | Conveyer speed (mm, rad/sec) | R | Position |
| M_EncMax | number of encoders 1 to 8 | The maximum value of external encoder data | R | Double-precision real number |
| M_EncMin | number of encoders 1 to 8 | The minimum value of external encoder data | R | Double-precision real number |
| M_EncSpd | number of encoders 1 to 8 | External encoder speed(Unit: pulse/sec) | R | Single-precision real number |
| M_TrkCQ | mechanism No. 1 to 3 | Tracking operation status of specified mechanism<br>1: Tracking<br>0: Not tracking | R | Integer |

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

### 19.1.3. List of Functions

**Table 19–3 List of Functions**

| Function name | Function | Result |
|---|---|---|
| Poscq(<position>) | Check whether the specified position is within the movement range.<br>1: Within the movement range<br>0: Outside the movement range | Integer |
| TrWcur(<encoder number>, <position>,<encoder value>) | Obtain the current position of a workpiece.<br><br><number of encoders><br>1 to 8 | Position |
| TrPos(<position>) | Acquire the coordinate position of a workpiece being tracked.<br>Trk On P0,P1,1,M1#<br>PC2=TrPos(P2)<br>PC2 above is obtained in the following manner.<br>PC1=P1+P_EncDlt*(M_Enc-M1#) ‘ The current position of P1<br>PC2=PC1*(P_Zero/P0*P2) | Position |

### 19.1.4. Explanation of Tracking Operation Instructions

The instructions related to tracking operations are explained in details below.

The explanations of instructions are given using the following format.

[Function] : Describes the function of an instruction.
[Format] : Describes the entry method of arguments of an instruction.
< > indicate an argument.
[ ] indicates that entry can be omitted.
☐ indicate that space is required.
[Term] : Describes meaning, range and so on of an argument.
[Example] : Presents statement examples.
[Explanation] : Provides detailed function descriptions and precautions.

## ***TrBase (tracking base)***

[Function]

Specify the workpiece coordinate system origin during the teaching operation and the encoder logic number of an external encoder used in tracking operation.

[Format]

| |
|---|
| TrBase □ <Reference position data> [ , <Encoder logic number>] |

[Term]

**<Reference position data>** (can be omitted):

Specify the origin position of position data to be followed during the tracking mode.

**<Encoder logic number>** (can be omitted):

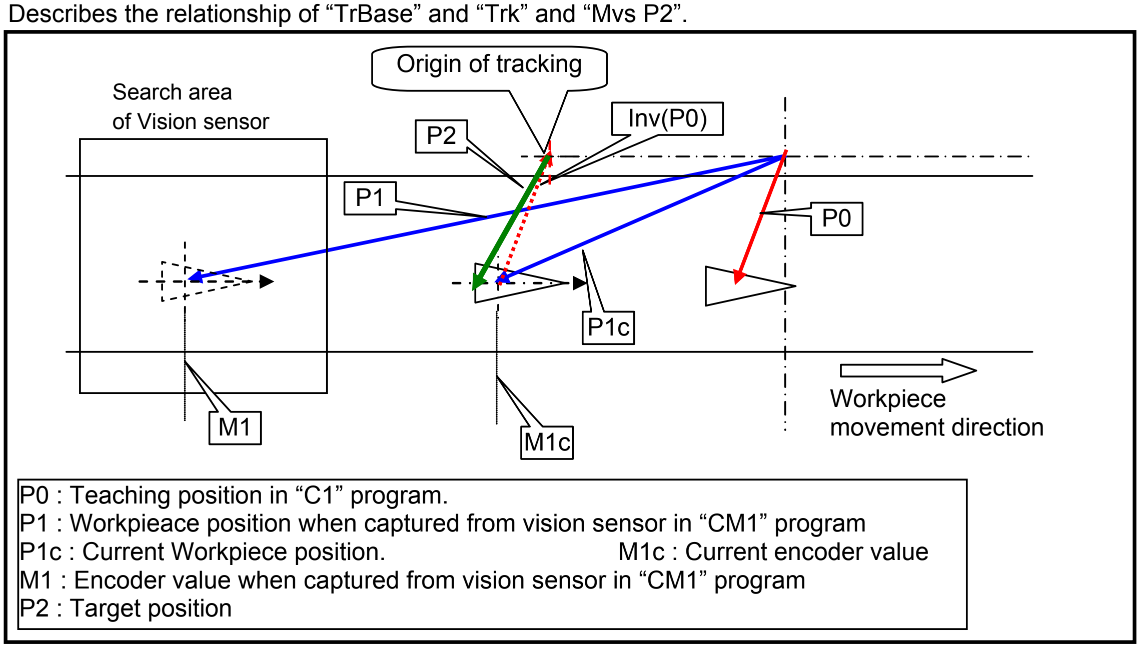
This is a logic number indicating the external encoder that performs tracking operation.
1 is set when this argument is omitted.
Setting range: 1 to 8

[Example]

```
1 TrBase P0          ' Specify the workpiece coordinate origin at the teaching position.
2 TrRd P1,M1,MKIND   ' Read the workpiece position data from the data buffer.
3 Trk On,P1,M1       ' Start tracking of a workpiece whose position measured by a sensor is P1 and
                       encoder value at that time is M1.
4 Mvs P2             ' Setting the current position of P1 as P1c, make the robot operate while following
                       workpieces with the target position of P1c*P_Zero/P0*P2.
5 HClose 1           ' Close hand 1.
6 Trk Off            ' End the tracking operation.
```

[Explanation]

- Specify the workpiece coordinate system origin during the teaching operation and the logic number of an external encoder used in tracking operation.
- If an encoder logic number is omitted, the previously specified value 1 is set.
- The reference position data and encoder number are set to their initial values until they are specified by the TrBase instruction or the Trk On instruction. The initial value is P_Zero for the reference position data and 1 for the encoder number.
- Describes the relationship of “TrBase” and “Trk” and “Mvs P2”.



## ***TrClr (tracking data clear)***

[Function]

Clears the tracking data buffer.

[Format]

```
TrClr □ [<Buffer number>]
```

[Term]

**<Buffer number>** (cannot be omitted):

Specify the number of a general-purpose output to be output.
Setting range:1 to 4 (The first argument of parameter [TRBUF])

[Example]

```
1 TrClr 1                          ' Clear tracking data buffer No. 1.
2 *LOOP
3 If M_In(8)=0 Then GoTo *LOOP     ' Jump to *LOOP if input signal No. 8, to which a photoelectronic
                                     sensor is connected, is OFF.
4 M1#=M_Enc(1)                     ' Acquire data of encoder number 1 at the time when input signal
                                     No. 8 is turned on and store it in M1#.
5 TrWrt P1, M1#,MK                 ' Write workpiece position data P1, encoder value M1# at the time
                                     an image is acquired and model number MK in the buffer.
```

[Explanation]

- Clear information stored in specified tracking buffer (1 to 4).
- Execute this instruction when initializing a tracking program.

## *Trk (tracking function)*

[Function]

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

[Format]

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

[Term]

**<Measurement position data>** (can be omitted):

Specify the workpiece position measured by a sensor.

**<Encoder data>** (can be omitted):

Specify a value of an encoder installed on a conveyer when a workpiece is measured.

**<Reference position data>** (can be omitted):

Specify the origin position of position data to be followed during the tracking mode.
If this argument is omitted, the robot follows the conveyer using the position specified by the TrBase instruction as the origin.
The initial value is PZERO.

**<Encoder logic number>** (can be omitted):

This is a logic number indicating the external encoder that performs tracking operation.
1 is set when this argument is omitted.
Setting range: 1 to 8

[Example]

```
1 TrBase P0          ' Specify the workpiece coordinate origin at the teaching position.
2 TrRd P1,M1,MKIND   ' Read the workpiece position data from the data buffer.
3 Trk On,P1,M1       ' Start tracking of a workpiece whose position measured by a sensor is P1 and
                       encoder value at that time is M1.
4 Mvs P2             ' Setting the current position of P1 as P1c, make the robot operate while following
                       workpieces with the target position of P1c*P_Zero/P0*P2 (P2 indicates the
                       workpiece grabbing position).
5 HClose 1           ' Close hand 1.
6 Trk Off            ' End the tracking operation.
```

[Explanation]

- Specify the position relative to the position data specified by Trk On as show in line 20 of the statement example for the target position of the movement instruction during tracking operation.

⚠CAUTION

A target position that moves in the tracking is calculated based on the workpiece position when Trk On.
The H2802 error might occur when a target position doesn't exist in the robot range at the time of Trk On.
Please execute Trk Off before the movement to the target position when the error occurs.
And, please execute Trk On again.

- “P_Zero/P0” in ”P1c*P_Zero/P0*P2” in [Example] can be replaced with INV(P0).

⚠CAUTION

•S/W Ver.R1 or later (SQ series) ,S1 or later (SD series), CR750/CR751 series.
When Hlt command is executed during tracking movement, tracking movement will be stopped (an equivalent for the Trk Off command) and execution of the program will be interrupted. In use of the multi-mechanism, tracking movement is stopped to the robot of the mechanism number got by the GetM command. When you continue tracking movement by the restart (continuation), please create the program to execute the Trk On command.

•S/W Ver. before R1 (SQ series), before S1 (SD series)
When Hlt command is executed during tracking movement, execution of the program will stop, but continue the conveyor tracking movement. When you stop tracking movement, please execute the Trk Off command before executing Hlt command.

## ***TrOut (reading tracking output signal and encoder value)***

[Function]

Read a tracking output value specified by a general-purpose output and read the value of an external encoder synchronously with the output.

[Format]

```
TrOut □ <Output number>, <Encoder 1 value read variable> [ , [<Encoder 2 value read variable>]
                         [ , [<Encoder 3 value read variable>] [ , [<Encoder 4 value read variable>]
                         [ , [<Encoder 5 value read variable>] [ , [<Encoder 6 value read variable>]
                         [ , [<Encoder 7 value read variable>] [ , [<Encoder 8 value read variable>] ]]]]]]
```

[Term]

**<Output number>** (cannot be omitted):
Specify the number of a general-purpose output to be output.

**<Encoder n value read variable>** (can be omitted):
Specify a double-precision value variable in which read values of an external encoder are stored.
Note) n is a value in the range from 1 to 8.

[Example]

```
1 *LOOP1
2 If M_In(10) <> 1 GoTo *LOOP1  ' Check whether a photoelectronic sensor is activated.
3 TrOut 20, M1# , M2#          ' Output from general-purpose output No. 20 and store the value of
                                 external encoder No.1 in M1#, and store the value of external encoder
                                 No.2 in M2# synchronously with the output.
4 *LOOP2
5 If M_In(21) <> 1 GoTo *LOOP2  ‘ Wait until the signal (general-purpose input No.21) which shows
                                 acquiring image from the vision sensor is turned on.
6 M_Out(20)=0                  ‘ Turn off the No.20 general-purpose output.
```

[Explanation]

- This instruction is used when triggering the vision sensor that calculates positions of workpieces to be tracked.
- It is possible to know the position where workpiece images are acquired by obtaining the external encoder values synchronously with the output.
- The general-purpose output signal specified <Output number> is maintained. Therefore, please turn off the signal by using the M_Out state ariable when you confirm acquiring of the vision sensor.

## TrRd (reading tracking data)

[Function]

Read position data for tracking operation, encoder data and so on from the data buffer.

[Format]

```
TrRd □ <Position data> [ , <Encoder data>] [ , [<Model number>] [ , [<Buffer number>] [ , <Encoder number>] ] ] ]
```

[Term]

**<Position data>** (cannot be omitted):
Specify a variable that contains workpiece positions read from the buffer.

**<Encoder data>** (can be omitted):
Specify a variable that contains encoder values read from the buffer.

**<Model number>** (can be omitted):
Specify a variable that contains model numbers read from the buffer.

**<Buffer number>** (can be omitted):
Specify a number of a buffer from which data is read.
1 is set if the argument is omitted.
Setting range: 1 to 4(The first argument of parameter [TRBUF])

**<Encoder number>** (can be omitted):
Specify a variable that contains values of external encoder numbers read from the buffer.

[Example]

(1) Tracking operation program

```
1 TrBase P0          ' Specify the workpiece coordinate origin at the teaching position.
2 TrRd P1,M1,MK      ' Read the workpiece position data from the data buffer.
3 Trk On,P1,M1       ' Start tracking of a workpiece whose measured position is P1 and encoder value
                       at the time of measurement is M1.
4 Mvs P2             ' Setting the current position of P1 as P1c, make the robot operate while following
                       workpieces with the target position of P1c*P_Zero/P0*PW2.
5 HClose 1           ' Close hand 1.
6 Trk Off            ' End the tracking operation.
```

(2) Sensor data reception program

```
1 *LOOP
2 If M_In(8)=0 Then GoTo *LOOP     ' Jump to *LOOP if input signal No. 8, to which a
                                     photoelectronic sensor is connected, is OFF.
3 M1#=M_Enc(1)                     ' Acquire data of encoder number 1 at the time when input
                                     signal No. 8 is turned on and store it in M1#.
4 TrWrt P1, M1#,MK                 ' Write workpiece position data P1, encoder value M1# at the
                                     time an image is acquired and model number MK in the buffer.
```

[Explanation]

- Read the workpiece position (robot coordinates), encoder value, model number and encoder number stored by the TrWrt instruction from the specified buffer.
- If the TrRd instruction is executed when no data is stored in the specified buffer, Error 2540(There is no read data) occurs.

## ***TrWrt (writing tracking data)***

[Function]

Write position data for tracking operation, encoder data and so on in the data buffer.

[Format]

```
TrWrt □ <Position data> [ , <Encoder data>] [ , [<Model number>] [ , [<Buffer number>] [ , <Encoder number>] ] ] ]
```

[Term]

**<Position data>** (cannot be omitted):
Specify the workpiece position measured by a sensor.

**<Encoder data>** (can be omitted):
Specify the value of an encoder mounted on a conveyer at the time a workpiece is measured.
The encoder value acquired in the M_Enc() state variable and the TrOut instruction is specified usually.

**<Model number>** (can be omitted):
Specify the model number of workpieces.
Setting range: 1 to 65535

**<Buffer number>** (can be omitted):
Specify a data buffer number.
1 is set if the argument is omitted.
Setting range: 1 to 4(The first argument of parameter [TRBUF])

**<Encoder number>** (can be omitted):
Specify an external encoder number.
The same number as the buffer number is set if the argument is omitted.
Setting range: 1 to 8

[Example]

(1) Tracking operation program

```
1 TrBase P0          ' Specify the workpiece coordinate origin at the teaching position.
2 TrRd P1,M1,MKIND   ' Read the workpiece position data from the data buffer.
3 Trk On,P1,M1       ' Start tracking of a workpiece whose measured position is P1 and encoder
                       value at the time of measurement is M1.
4 Mvs P2             ' Setting the current position of P1 as P1c, make the robot operate while
                       following workpieces with the target position of P1c*P_Zero/P0*PW2.
5 HClose 1           ' Close hand 1.
6 Trk Off            ' End the tracking operation.
```

(2) Sensor data reception program

```
1 *LOOP
2 If M_In(8)=0 Then GoTo *LOOP   ' Jump to +LOOP if input signal No. 8, to which a
                                   photoelectronic sensor is connected, is OFF.
3 M1#=M_Enc(1)                   ' Acquire data of encoder number 1 at the time when input
                                   signal No. 8 is turned on and store it in M1#.
4 TrWrt P1, M1#,MK               ' Write workpiece position data P1, encoder value M1# at the
                                   time an image is acquired and model number MK in the buffer.
```

[Explanation]

- This function stores the workpiece position (robot coordinates) at the time when a sensor recognizes a workpiece, encoder value, model number and encoder number in the specified buffer.
- Arguments other than the workpiece position (robot coordinates) can be omitted. If any of the arguments are omitted, the robot operates while following changes of position data.
- Workpieces within the same workpiece judgment distance set in the “TRCWDST” parameter are regarded as the same workpiece. Even if the data is written twice in the buffer with the TrWrt instruction, only one data set is stored in the buffer. For this reason, data for one workpiece only is read with the TrRd instruction even if images of the same workpiece are acquired twice with a vision sensor.

## M_EncL (Latched Encoder data)

[Function]
At the instant of receipt of a TREN signal for Q17EDPX module, a stored encoder data is read.
Also, 0 is written to clear the stored encoder data to zero.

[Format]

| | |
|---|---|
| Example)<Numeric Variable>=M_EncL[(<logic encoder number>)] | --------referencing |
| M_EncL[(<logic encoder number>)]=<Constants> | --------writing |

[Terminology]
<Numeric Variable> Specify the numerical variable to substitute.
Available argument type

| | Numeric value | | | Position | Joint | Character string |
|---|---|---|---|---|---|---|
| | Integer | Real number | Double-precision real number | | | |
| Variable | ○ | ○ | ○ | ○ (member data) | ○ (member data) | - Error 4220 |

○:Available -:Not available(syntax error at input time)

<logic encoder number> (can be omitted) Specify the value of an logic encoder number
Available argument type

| | Numeric value | | | Position | Joint | Character string |
|---|---|---|---|---|---|---|
| | Integer | Real number | Double-precision real number | | | |
| Constants | ○ | ○ Rounding | ○ Rounding | - | - | - Error 4220 |
| Variable | ○ | ○ Rounding | ○ Rounding | ○ (member data) | ○ (member data) | - Error 4220 |

○:Available -:Not available(syntax error at input time)

<Constants> Specify the stored encoder data to initial value(zero or other).
Available argument type

| | Numeric value | | | Position | Joint | Character string |
|---|---|---|---|---|---|---|
| | Integer | Real number | Double-precision real number | | | |
| Constants | ○ | ○ | ○ | - Error 4220 | - Error 4220 | - Error 4220 |
| Variable | ○ | ○ | ○ | ○ (member data) | ○ (member data) | - Error 4220 |

○:Available -:Not available(syntax error at input time)

[Reference Program]

| | |
|---|---|
| 1 MENC1#=M_EncL(1) | At logic encoder number 1, assign encoder data stored at the time of receipt of a TREN signal to the variable MENC1#. |
| 2 MENC2#=M_EncL(M1%) | At a logic encoder number specified in the variable M1%, assign encoder data stored at the time of receipt of a TREN signal to the variable MENC2#. |
| 3 TrWrt P1, MEncL(1), MK | Write work position data P1, encoder value M_EncL(1) present at the time of receipt of a TREN signal and work category number MK onto the buffer 1 for tracking. |
| 4 M_EncL(1)=0 | Use latched data to clear the encoder to zero as it is not required until next latched data is used. |

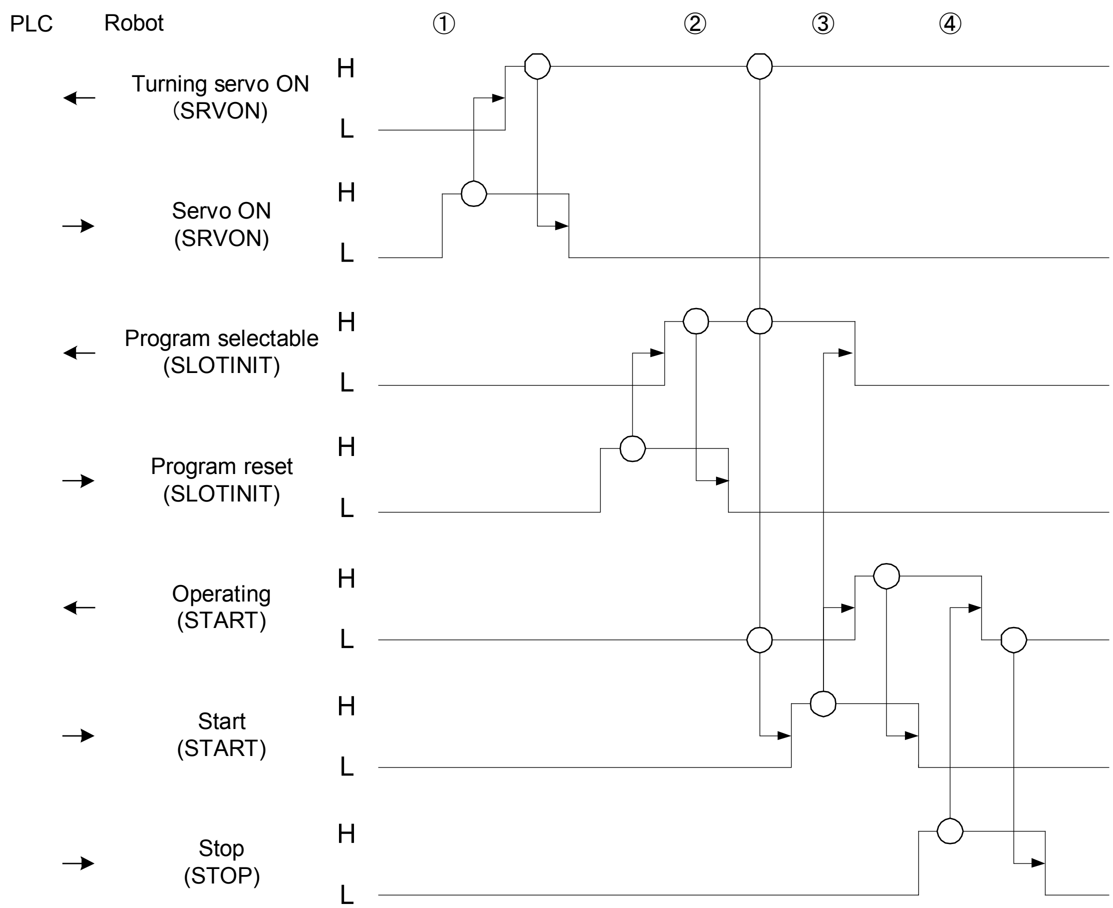
[Explanation]
• Stored encoder value corresponding to the encoder number being specified in <logical encoder number> is acquired.
Encoder value is stored in memory at a low-to-high or high-to-low transition of TREN signal which has been specified with a DIP switch on Q17EDPX module.
Encoder value thus acquired is written onto the buffer for tracking by using a TrWr command so as to perform tracking operations.

- As encoder value is in double-precision real number, specify <numerical variable> with a variable which is of double-precision real-number type.
- You can omit the step to specify <logic encoder number> . When it is omitted, logic encoder number will be treated as "1."
- Number which you can enter to specify <logic encoder number> is an integer in the range of "1" to "8." Entering anything else causes L3110 (Out-of-range Argument) error to occur.
  * If a number having a decimal part is entered, the fraction of 0.5 or over will be counted as one and the rest will be cut away.
- As latched encoder data represents a value present at a low-to-high or high-to-low transition of TREN signal, you should check input corresponding to input number in 810 to 817 range which has been assigned to TREN signal when reading it out.
- You can clear the encoder to zero by typing "0" after having used latched encoder data. This step may be performed as a precaution against using previously latched data.

## 19.2. Timing Diagram of Dedicated Input/Output Signals

### 19.2.1. Robot Program Start Processing

The signal timing when a robot program is started from an external device is shown below.




① PLC sets “servo ON H” when it detects “turning servo ON L.” The robot turns the servo power supply on and sets “turning servo ON H.” PLC acknowledges “turning servo ON H” and sets “servo ON L.”

② PLC sets “program reset H” upon receiving “program selectable L.” The robot returns to the beginning of the program and sets “program selectable H” when the program becomes ready to be started. PLC sets “program reset L” when it detects “program selectable H.”

③ PLC acknowledges “turning servo ON H,” “program selectable H” and “operating L” and sets “start H.” The robot sets “program selectable L” and “operating H” when it detects “start H.” PLC confirms “operating H” and sets “start L.”

④ If a stop signal is input, the following processing is performed.
Upon receiving “stop H” from PLC, the robot sets “operating L.”

# 20. Troubleshooting

This section explains causes of error occurrence and actions to be taken.

## 20.1. Occurrence of Error Numbers in the Range from 9000 to 9999

This section describes causes of errors that may occur while starting a program and how to handle them.

**Table 20–1 List of Errors in Sample Programs**

| Error number | Error description | Causes and actions |
|---|---|---|
| 9100 | Communication error | [Causes]<br>The network vision sensor and the robot cannot be connected by the "C1" program or the robot cannot log on the vision sensor.<br>[Actions]<br>1) Check the Ethernet cable which connects the robot with the network vision sensor. |
| 9101 | Encoder number out of range | [Causes]<br>The encoder number specified in "A1" program to "C1" program is "0" or "9" or larger.<br>[Actions]<br>1) Check the X coordinate of the position variable "PE" in the programs. |
| 9102 | Model number out of range | [Causes]<br>The model number specified in "C1" program is "0" or "10" or larger.<br>[Actions]<br>1) Check the X coordinate of the position variable "PRM1" in "C1" program.<br>2) If there are more than 11 models, change "MWKMAX=10" line in "C1" program. |
| 9110 | Position accuracy out of range | [Causes]<br>The workpiece position calculated by operations in "A1" program to "C1" program is very different from the theoretical value.<br>The example is shown in (*1).<br>[Actions]<br>1) Check the X and Y coordinates of the position variable "PVTR" in "CM1" program. These values represent the difference from the theoretical value.<br>2) If the difference stored in "PVTR" is large, run "A1" program to "C1" program again.<br>3) Please add the value of positional variable "PCHK" in the 'CM1' program when the hand offsets from time when the calibration was executed and add the amount of the offset.<br>4) Check that the X and Y coordinates of the position variable "PCHK" in "CM1" program are not "0." If they are "0," change the difference from the theoretical value to an allowable value. |
| 9199 | Program error | [Causes]<br>A return value cannot be created by the *S50WKPOS function of "1" program.<br>[Actions]<br>1) Check the reason why "MY50STS" of the *S50WKPOS function in "1" program does not change from"0". |

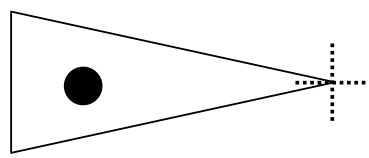
### (*1) About the factor that the L9110 error occurs

Positional variable "PVTR" in 'CM1' program is calculated based on the setting of the A1-C1 program.
The calculation result is a difference between the position of [+] mark set with the vision sensor and the position taught by the 'C1' program.
And, the L9110 error occurs when the difference exceeds the numerical value specified for positional variable "PCHK".

Therefore, there is a possibility that the L9110 error occurs in the following cases.

[a] The position taught by the 'C1' program shifts to [+] mark specified with the vision sensor.

For instance, when the vision sensor output the triangular top, ● sign was taught in the 'C1' program.
In this case, the difference is recognized as a gap.



[b] There is a difference to the flange and each hand of the robot in the gap for the multi hand.

The calibration executed by using the 'B1' program, the calibration treatment device is used.
It is installed in the flange of the robot. The position that the vision sensor outputs becomes the flange position of the robot.
However, when teaching by the 'C1' program, the gap is caused there to use and to teach the hand.

[c] In the setting of 'A1' - 'C1' program, some mistakes are found.

"P_EncDlt()" (the amount of the movement of the robot per a pulse) in the 'A1' program is an unexpected value.
Or, in the 'B1' program, the direction of three points specified by the calibration was different or it was the inputting error of coordinates.

## 20.2. Occurrence of Other Errors

**Table 20–2 List of Tracking relation Errors**

| Error number | Error description | Causes and actions |
|---|---|---|
| L2500 | Tracking encoder data error | [Causes]<br>The data of the tracking encoder is abnormal.<br>(The amount of the change is 1.9 times or more.)<br>[Actions]<br>1) Check the conveyor rotates at the fixed velocity.<br>2) Check the connection of the encoder.<br>3) Check the earth of the earth wire. |
| L2510 | Tracking parameter reverses | [Causes]<br>Tracking parameter[EXCRGMN] and [EXCRGMX] Setting value reverses<br>[Actions]<br>1) Check the value of [ENCRGMX] and [ENCRGMN] parameters. |
| L2520 | Tracking parameter is range over | [Causes]<br>The set value is outside the range parameter [TRBUF]. The first argument is 1 to 8, and the second argument is 1 to 64.<br>[Actions]<br>1) Check the value of [TRBUF] parameter. |
| L2530 | There is no area where data is written | [Causes]<br>The data of the size or more of the buffer in which the TrWrt command was continuously set to the second argument of parameter [TRBUF] was written.<br>[Actions]<br>1) Check the execution count of the TrWrt command is correct.<br>2) Check the value of the second argument of parameter [TRBUF] is correct.<br>3) Check that the X and Y coordinates of the position variable ”PCHK” in “CM1” program are not ”0.” If they are ”0,” change the difference from the theoretical value to an allowable value. |
| L2540 | There is no read data | [Causes]<br>The TrRd command was executed in state the data is not written in tracking buffer.<br>[Actions]<br>1) Execute the TrRd command after confirming whether the buffer has the data with the state variable [M_Trbfct].<br>2) Confirm whether the buffer number specified by the buffer number specified in TrWrt Mende and the TrRd command is in agreement. |
| L2560 | Illegal parameter of Tracking | [Causes]<br>The value set as the parameter [EXTENC] is outside the range. The ranges are 1-8.<br>[Actions]<br>Please confirm the value set to Parameter [EXTENC].<br>Please confirm whether the Q173DPX unit is installed in the slot specified for parameter "ENCUNITn" (n=1-3).<br>Please confirm whether slot 0-2 of a basic base is not specified by setting the parameter.<br>Please confirm whether the setting of "Management CPU" that exists in "I/O unit and intelligent function unit details setting" of the parameter of the sequencer and specification of parameter "ENCUNITn" (n=1-3) are corresponding. There is a possibility Q173DPX is not robot CPU management. |
| L2570 | Installation slot error. | [Causes]<br>Q173DPX is installed in slot 0-2 of a basic base.<br>[Actions]<br>Slot 0-2 of the basic base is basically only for CPU. Please install Q173DPX since slot3. |

| Error number | Error description | Causes and actions |
|---|---|---|
| L3982 | Cannot be used (singular point) | [Causes]<br>1) This robot does not correspond to the singular point function<br>2) Cmp command is executed<br>3) A synchronous addition axis control is effective<br>4) Tracking mode is effective<br>5) Pre-fetch execution is effective<br>6) This robot is a setting of the multi mechanism<br>7) ColChk On command is executed<br>[Actions]<br>1) Check the argument of Type specification<br>2) Invalidate a compliance mode (execute Cmp Off)<br>3) Invalidate a synchronous addition axis control<br>4) Invalidate a tracking mode (execute Trk Off)<br>5) Invalidate a pre-fetch execution<br>6) Do not use the function of passage singular point<br>7) Invalidate a collision detection (execute ColChk Off) |
| L6632 | Input TREN signal cannot be written | [Causes]<br>During the actual signal input mode, external output signal 810 to 817 (TREN signal) cannot be written.<br>[Actions]<br>1) Use an real input signal (TREN signal) |

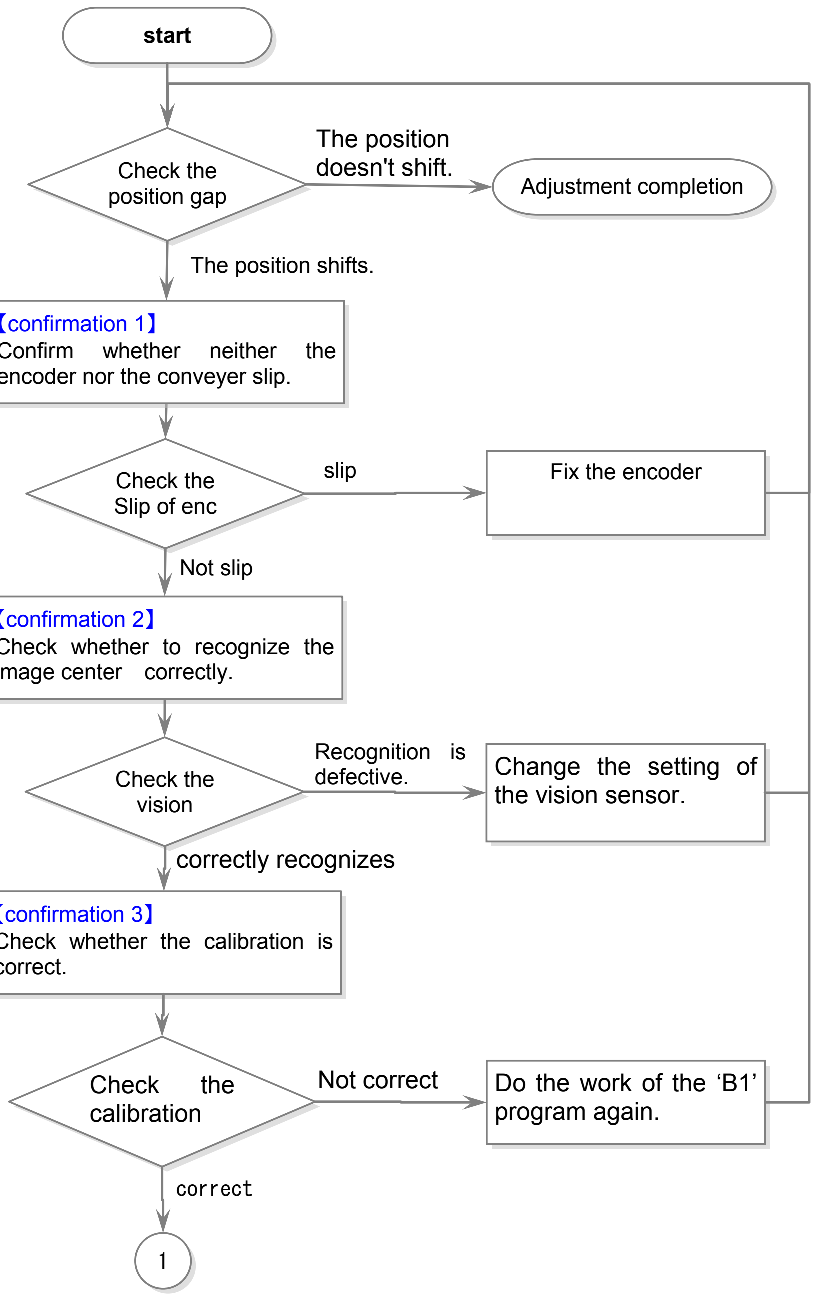
Please refer to separate manual “Troubleshooting”.

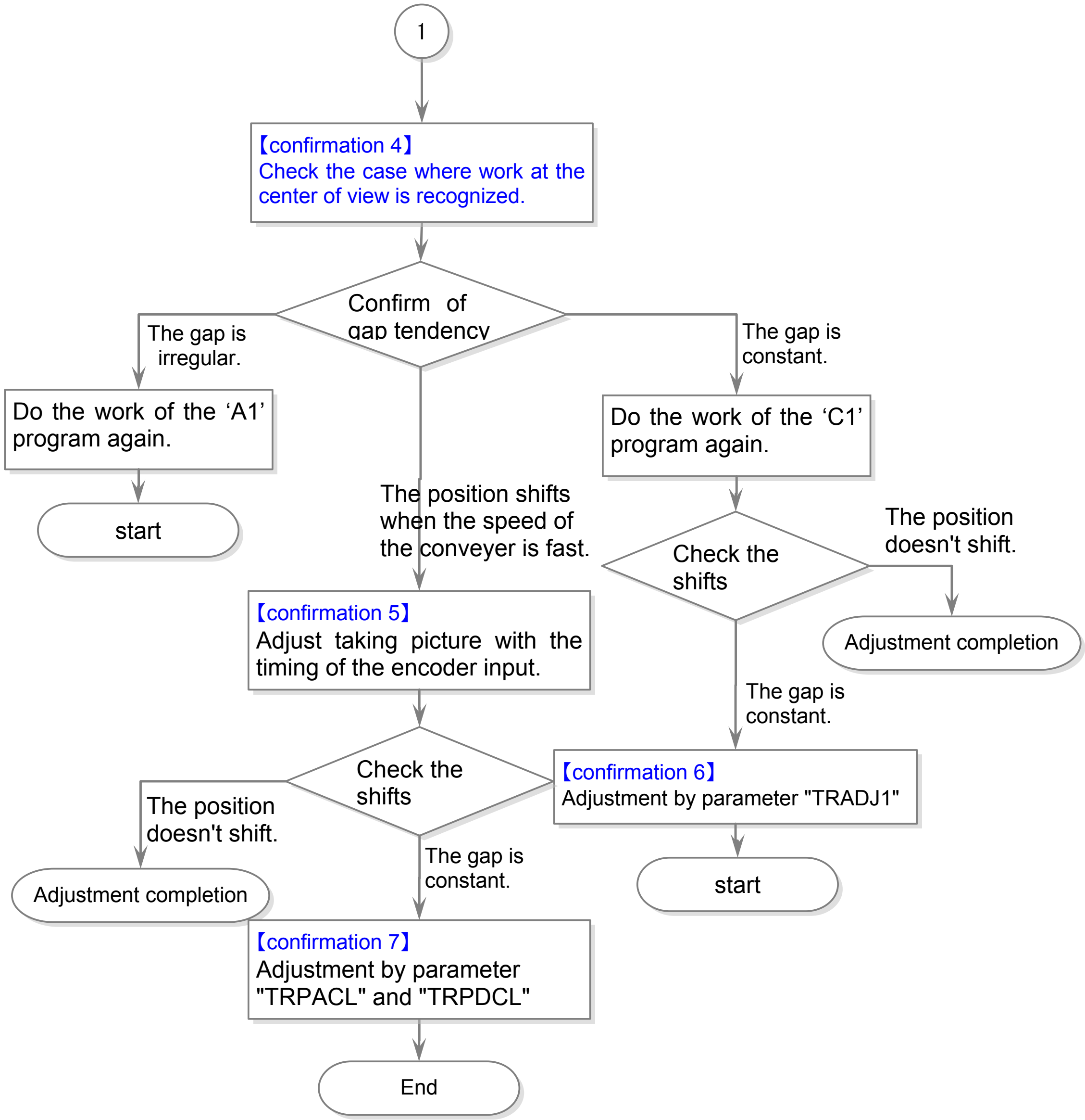
## 20.3. In such a case (improvement example)

Explain the improvement example, when building the tracking system using the sample robot program.

### 20.3.1. The adsorption position shifts.

When the place that shifts from the specified adsorption position has been adsorbed, the cause is investigated according to the following procedures.






【confirmation 1】

1) Stop the conveyer.
2) Confirm the disk installed in the rotary encoder has come in contact with the conveyer.
3) Confirm whether the disk installed in the encoder rotates when the conveyer is made to work.

【confirmation 2】

1) Stop the conveyer.
2) Put workpiece on the center of the vision view.
3) In In-Sight Explorer(EasyBuilder), click the "Set Up Image" from the "Application Steps". And, set "Calibration Type" displayed in the lower right of the screen to "None".
4) Confirm workpiece is recognized by starting the job, and the recognition result (pixel level) is correct.
   (example)
   When the center of view is recognized, the result of (320,240) is displayed when pixels are 640×480 vision sensors.
5) Arrange workpieces on four corners.
6) Confirm whether the workpieces put on four corners of the image is recognized similar and correctly.

【confirmation 3】

1) Stop the conveyer.
2) Put workpiece on the center of the vision view.
3) In In-Sight Explorer(EasyBuilder), click the “Set Up Image” from the “Application Steps”.
   Set "Calibration Type" displayed in the lower right of the screen to "Import".
   Specify the file that exported when the calibration is done to "File Name".
4) Confirm workpiece is recognized by starting the job, and the recognition result (robot coordinate) is correct.
   (example)
   (+0, +0) is displayed as a recognition result when assuming that the robot coordinates are set as follows when the calibration is done by using the calibration seat, and using a ○ sign in four corners.
   (the first point xy) (the second point xy)(the third point xy)(the fourth point xy)
   = (+100,+100), (+100,-100), (-100,+100), and (-100,-100)
5) Arrange workpieces on four corners.
6) Confirm whether the workpieces put on four corners of the image is recognized similar and correctly.
   The recognition result becomes (+100,+100), (+100,-100), (-100,+100), and (-100,-100).

【confirmation 4】

1) Stop the conveyer.
2) Put workpiece on the center of the vision view.
3) Change X coordinates of PDLY1 in ‘1’ program to a big value like the “10" second etc.
4) Start ‘1’ program, and start the conveyer in low-speed.
5) Stop the conveyer because it keeps following during the “10" second in the place where the robot moved to the adsorption position. And, stop ‘1’ program.
6) Confirm whether the position in which the robot adsorbs workpiece is correct.
7) Confirm the tendency to a positional gap repeating this work several times.

【confirmation 5】

1) Stop the conveyer.
2) Start the ‘1’ program, and start the conveyer in the speed that you want.
3) Flow workpiece.
4) Stop the conveyer because it keeps following during the “10" second in the place where the robot moved to the adsorption position. And, stop ‘1’ program.
5) Confirm the position in which the robot adsorbs workpiece.

<The position shifts in shape to adsorb the rear side of work >

Please adjust < delay time of NvTrg command used because of the 'CM1' program >.
Please adjust the encoder value specified by the TrWrt command as < delay time > “0" when the adjustment by < delay time of NvTrg command > is difficult.
For instance, the 'CM1' program is changed as follows and the numerical value (for instance, following “500") is adjusted.

```
MENCDATA#=MTR1#+500
TrWrt PRW, MENCDATA#, MWKNO,1,MENCNO
```

【confirmation 6】

1) Change parameter "TRADJ1", and adjust a positional gap.

【confirmation 7】

1) Change parameter "TRPACL" and "TRPDCL" to make the follow speed of the tracking fast.
   Note it though the load factor of each axis of the robot goes up.
   Confirm the state of the load of each axis by "Load factor monitor" of RT ToolBox2.

### 20.3.2. Make adsorption and release of the work speedy

In the tracking system, adsorption confirmation of the work may be unnecessary. In that case, processing of adsorption and release can be made speedy by the following methods.

(1) Adjust adsorption time and release time.
Adjust the adjustment variable "PDLY1", and the value of X coordinates of "PDLY2" of the program 1. Refer to "Table 16−1 List of Adjustment Variables in Programs" for the adjustment method.

### 20.3.3. Make movement of the robot speedy.

Adjust the following setting to make movement of the robot speedy.

(1) Adjust the acceleration and the deceleration time for the tracking by using the parameter.
Acceleration and the deceleration of the follow operation can be done fast by reducing the value of each element of parameter "TRPACL" and "TRPDCL".
(example)
For the robot of the RH type (X,Y,Z,A,B,C) = (0.2, 0.2, 1.0, 1.0, 1.0, 1.0) : X and Y are changed.
For the robot of the RV type (X,Y,Z,A,B,C) = (0.2, 0.2, 0.2, 1.0, 1.0, 1.0) : X, Y, and Z are changed.

(2) Adjustment of the optimal acceleration-and-deceleration setting
Set mass, size, and center of gravity of the hand installed in the robot as the parameter "HNDDAT1." And, set mass, size, and center of gravity of the work as the parameter "WRKDAT1."
By this setting, the robot can move with the optimal acceleration and deceleration and speed.
Refer to "Table 11−2 List of Operation Parameter" for setting method.

(3) Adjustment of carrying height
By making low distance at adsorption and release of robot, the moving distance decreases and motion time can be shortened as a result. Refer to the adjustment variable of "PUP1"and "PUP2" in the "Table 16−1 List of Adjustment Variables in Programs" for change of rise distance.

### 20.3.4. The robot is too speedy and drops the work.

Since the robot's acceleration and deceleration are speedy, drop the work, adjustment is necessary. Refer to the adjustment variable of「PAC1」to「PAC3」and「PAC11」to「PAC13」in the "Table 16−1 List of Adjustment Variables in Programs" for the adjustment method of the acceleration and deceleration.

### 20.3.5. Restore backup data to another controller

The status variable "P_EncDlt" is not saved in the backup data from tracking system robot controller.
To generate the value of "P_EncDlt", execute the "P_EncDlt(MENCNO) =PY10ENC" command of "Program A" by step forward. (Moving distance per one pulse)

### 20.3.6. Circle movement in tracking.

Screw fastening and decoration on the work, etc are available in the tracking system. Here, explain the example which draws the circle on the basis of the adsorption position.

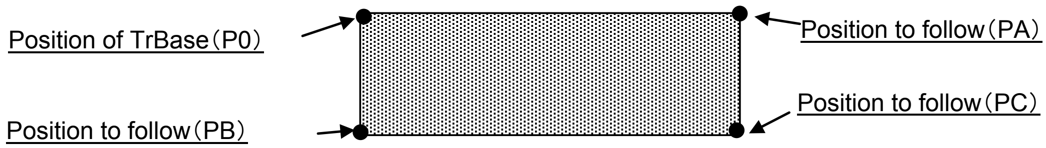
<Conditions>
*The adsorption position taught by Program C is the starting point of the circle.
*The offset from the adsorption position of pass and end position of circle decided as follows.
POF1=(+50,+50,0,0,0,0,0,0)(0,0)......Relative distance to pass position from adsorption position.
POF2=(0,+100,0,0,0,0,0,0)(0,0)........Relative distance to end position from adsorption position
*Create PGT1 (pass point) and PGT2 (end point) from the relative distance.
*Use the Mvr command (circle command) and move on the circle of PGT->PGT1 ->PGT2.

The example of program change of the above <conditions> is shown in the following.

| Before sample program change | After sample program change |
|---|---|
| 81 Trk On,PBPOS,MBENC#,PTBASE･･･ | 81 Trk On,PBPOS,MBENC#,PTBASE･･･ |
| 82 Mov PGT,PUP1.Y Type 0,0 | 82 Mov PGT,PUP1.Y Type 0,0 |
| 83 Accel PAC2.X,PAC2.Y | **83 POF1=(+50,+50,0,0,0,0,0,0)(0,0) '** |
| 84 Mvs PGT | **84 POF2=(0,+100,0,0,0,0,0,0)(0,0) '** |
| 85 HClose 1 | **85 PGT1=PGT*POF1 'Pass position** |
| | **86 PGT2=PGT*POF2 'End position** |
| | 87 Accel PAC2.X,PAC2.Y |
| | 88 Mvs PGT |
| | **89 Mvr PGT,PGT1,PGT2 ' Circle movement** |
| | 90 HClose 1 |

### 20.3.7. Draw the square while doing the tracking.

Here, explain the example which draws the outline of the following square workpiece on the basis of the adsorption position.




The robot traces the outline of workpiece clockwise based on the position specified that the following programs are executed by the TrBase instruction.

```
1 TrBase P0                 ' Specify the workpiece coordinate origin at the teaching position.
2 TrRd P1,M1,MKIND          ' Read the workpiece position data from the data buffer.
3 Trk On,P1,M1              ' Start tracking of a workpiece whose position measured by a sensor is
                              P1 and encoder value at that time is M1.
4 Cnt 0
5 Mov P0, +20 ← Please specify -20 for RV robot though RH(SCARA) robot is +20.
6 Mvs P0
7 Mvs PA
8 Mvs PB
9 Mvs PC
10 Mvs PC, +20 ← Please specify -20 for RV robot though RH(SCARA) robot is +20.
11 Trk Off                  ' End the tracking operation.
```

# 21. Appendix

This appendix provides a list of parameters related to tracking and describes Expansion serial interface connector pin assignment as well as sample programs for conveyer tracking and vision tracking.

## 21.1. List of Parameters Related to Tracking

**Table 21–1 List of Parameters Related to Tracking**

| Parameter | Parameter name | Number of elements | Description | Setting value at factory shipment |
|---|---|---|---|---|
| Tracking buffer | TRBUF | 2 integers | Number of tracking buffers and their sizes (KB)<br><Buffer number><br>Specify the number of buffers where the tracking data is stored.<br>Mainly the tracking data for each conveyors is saved at the buffer. Change the set value, when the conveyor for tracking is increased.<br>However, if the value is enlarged, the memory area where the tracking data is saved will be secured. Be careful because the program number which can be saved decreases.<br>Setting range: 1 to 8<br><br><Buffer size><br>Specify the size in which the tracking data is preserved.<br>Change this element when there is larger tracking data saved by TrWrt command than reading by TrRd command.<br>Be careful because the memory is secured like the above-mentioned [Buffer number].<br><br>Setting range: 1 to 200 | 2 , 64 |
| Minimum external encoder value | ENCRGMN | 8 integers | The minimum external encoder data value (pulse)<br><br>The range of the encoder value which can be acquired in state variable “M_Enc"<br>(minimum value side) | 0,0,0,0,0,0,0,0 |
| Maximum external encoder value | ENCRGMX | 8 integers | The maximum external encoder data value (pulse)<br><br>The range of the encoder value which can be acquired in state variable “M_Enc"<br>(maximum value side) | 100000000,<br>100000000,<br>100000000,<br>100000000,<br>100000000,<br>100000000,<br>100000000,<br>100000000 |
| Tracking buffer | TRBUF | 2 integers | Number of tracking buffers and their sizes (KB)<br><Buffer number><br>Specify the number of buffers where the tracking data is stored.<br>Setting range: 1 to 8<br><Buffer size><br>Specify the size in which the tracking data is preserved.<br>Setting range: 1 to 64 | 4 , 64 |

| Parameter | Parameter name | Number of elements | Description | Setting value at factory shipment |
|---|---|---|---|---|
| Tracking adjustment coefficient 1 | TRADJ1 | 8 real numbers (X,Y,Z, A,B,C, L1,L2) | Tracking adjustment coefficient 1<br>Set the amount of delay converted to the conveyer speed. Convert to 100 mm/s.<br>Example)<br>• If the delay is 2 mm when the conveyer speed is 50 mm/s:<br>Setting value = 4.0 (2 / 50 * 100)<br>• If the advance is 1 mm when the conveyer speed is 50 mm/s:<br>Setting value = -2.0 (-1 / 50 * 100) | 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00 |
| Tracking acceleration | TRPACL | 8 real numbers (X,Y,Z, A,B,C, L1,L2) | Tracking acceleration.<br>Acceleration during execution of tracking movement. | 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0 |
| Tracking deceleration | TRPDCL | 8 real numbers (X,Y,Z, A,B,C, L1,L2) | Tracking deceleration.<br>Deceleration during execution of tracking movement. | 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0 |

## 21.2. Shine of changing parameter

When the tracking function is used, the parameter need to be changed depens on operation phase. List of the parameter is shown as follow.

List **21-2 List of the user shine of changing parameter**

| No. | Operation phase | Model | | Parameter name | Example | Explanation |
|---|---|---|---|---|---|---|
| | | CR750-Q<br>CR751-Q<br>CRnQ-700 | CR750-D<br>CR751-D<br>CRnD-700 | | | |
| 1 | Power on<br>Setting orgin<br>JOG operation | — | — | — | — | |
| 2 | Attach option<br>Connection with peripherals | ● | — | ENCUNIT1<br>ENCUNIT2<br>ENCUNIT3 | 0, 5<br>-1, 0<br>-1, 0 | It is set to have installed Q173DPX unit into 5 I/O slot of the base unit.<br>By setting it, incremental three encoders connected with Q173DPX unit are recognized physical encoder number 1 to 3. |
| 3 | | ● | ● | TRMODE | 1 | It makes tracking function valid. By being valid, incremental encoder value can be got. |
| 4 | In case of robot programming | ● | ● | EXTENC | 1, 2,<br>3, 1,<br>2, 3,<br>1, 2 | About EXTENC, because initial value is 1,2,1,2,1,2,1,2, physical encoder number 1 and 2 are allocated to logic encoder(physical encoder number3) number 1 to 8. At this time, the encoder connected with CH3 of Q173DPX unit is not allocated to logic encoder number. So by changing this parameter to 1,2,3,1,2,3,1,2, the encoder of CH3 is allocated to logic encoder number 3 and 6. Also it is possible in following case. 3 pcs encoder are connected with Q173DPX unit and attach each encoder to conveyer 1 to 3. If conveyer1 connect to encoder3, conveyer 3 connect to encoder 1, it is not effective to change encoder, so by changing this parameter to 3,2,1,3,2,1,1,2, encoder attached with conveyer 1 becomes logic encoder1. |

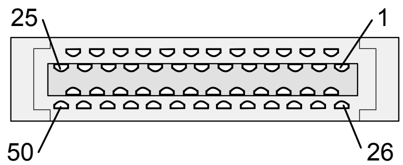
| No. | Operation phase | Model | | Parameter name | Example | Explanation |
|---|---|---|---|---|---|---|
| | | CR750-Q<br>CR751-Q<br>CRnQ-700 | CR750-D<br>CR751-D<br>CRnD-700 | | | |
| 5 | In case of system debag | ● | ● | TRCWDST | 20.0 | In case of vision tracking, if there is a workpiece not recognized well by vision sensor, it might reply over one recognition results to one workpiece. In this case, it makes possible to get only one recognition result excluding the results with the distance which is shorter than the distance set by this parameter. For example, it is recognized that 3 vision sensors exist for 1 workpieces. This one workpiece is got and another 2 workpieces are not got because the distance of result is shorter than it set 20mm. |
| 6 | In case of system debug | ● | ● | TRADJ1 | +0.00,<br>+4.00,<br>+0.00,<br>+0.00,<br>+0.00,<br>+0.00,<br>+0.00,<br>+0.00,<br>+0.00 | It is possible to adjust the gap by using this parameter when this gap is caused every time in the same direction when the tracking operates.<br>For example, the speed of conveyer is 50mm/s and there is +2mm gap (＋Y direction) +2mm,<br>Set value = 4.0 (2 / 50 * 100 )<br>+4.0 is set to the second element that shows Y coordinates. |
| 7 | | ● | ● | TRBUF | 3, 100 | When three kinds of workpieces flow respectively on the three conveyers for one robot controller, three tracking buffers where workpiece information is preserved are needed. In this case, the first element of this parameter is changed to three. Moreover, when TrWrt command is frequently executed and TrRd command is slow, workpiece information collects in the tracking buffer. Because the error occurs when 64 workpieces information or more on an initial value collects, it is necessary to increase the number in which work information is preserved. Then, the second element of this parameter is changed to 100. |

| No. | Operation phase | Model | | Parameter name | Example | Explanation |
|---|---|---|---|---|---|---|
| | | CR750-Q<br>CR751-Q<br>CRnQ-700 | CR750-D<br>CR751-D<br>CRnD-700 | | | |
| 8 | Others | ● | ● | ENCRGMN | 0,0,0,0,<br>0,0,0,0 | This parameter is a parameter that sets the range of the value of state variable M_Enc.<br>M_Enc becomes the range of 0-100000000, and next to 100000000, it becomes 0 encoder rotates in case of an initial value.<br>Though this range is changed by this parameter, tracking sample program is made on the assumption that it is used within this range, so do not change this parameter. |
| 9 | | ● | ● | ENCRGMX | 100000000,<br>100000000,<br>100000000,<br>100000000,<br>100000000,<br>100000000,<br>100000000,<br>100000000 | |

## 21.3. Expansion serial interface Connector Pin Assignment (CR750-D/CR751-D, CRnD-700 series controller)

"Figure 21−1 Connector Arrangement" shows the connector arrangement and "Table 21−3 Connectors: CNENC/CNUSR Pin Assignment" shows pin assignment of each connector.

CNUSR2（CR750-D/CR751-D）
Encoder

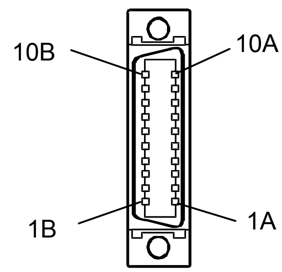



Connector: CNUSR2

CNUSR11/12/13（CR750-D）
Encoder


Connector: CNUSR11/12/13

CNENC（CRnD-700）
Encoder x 2CH




Connector: CNENC

**Figure 21−1 Connector Arrangement**

**Table 21–3 Connectors: CNENC/CNUSR Pin Assignment**

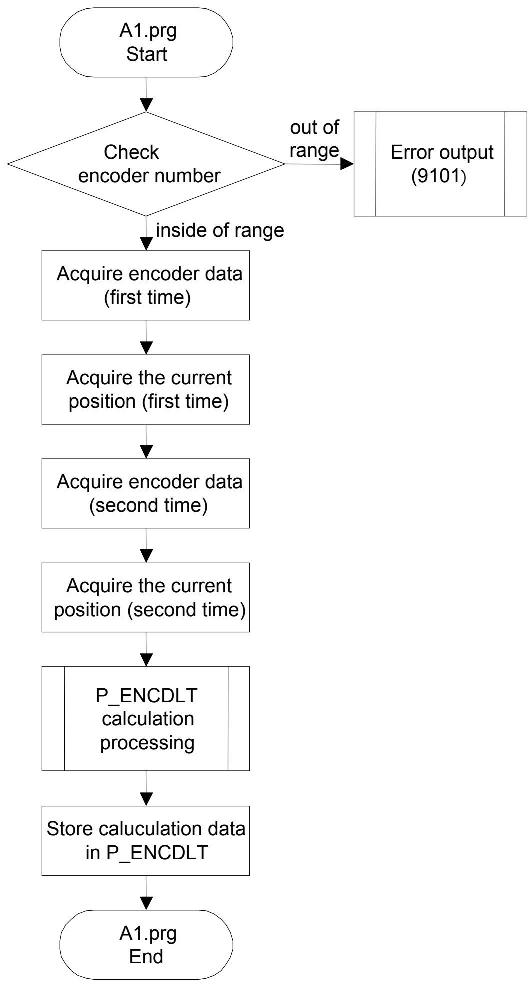
| Pin NO. | | | Signal name | Explanation | Input/output | Remark |
|---|---|---|---|---|---|---|
| CRnD-700 controller (CNENC) | Connector name – Pin name | | | | | |
| | CR751-D controller | CR750-D controller | | | | |
| 1A | CNUSR1-28 | CNUSR11-6 | SG | Control power supply 0 V | GND | |
| 2A | CNUSR1-21 | CNUSR13-3 | LAH1 | + terminal of differential encoder A-phase signal | Input | |
| 3A | CNUSR1-22 | CNUSR13-5 | LBH1 | + terminal of differential encoder B-phase signal | Input | CH1 |
| 4A | CNUSR1-23 | CNUSR13-8 | LZH1 | + terminal of differential encoder Z-phase signal | Input | |
| 5A | CNUSR1-33 | CNUSR12-6 | SG | Control power supply 0 V | GND | |
| 6A | CNUSR2-21 | CNUSR2-21 | LAH2 | + terminal of differential encoder A-phase signal | Input | |
| 7A | CNUSR2-22 | CNUSR2-22 | LBH2 | + terminal of differential encoder B-phase signal | Input | CH2 |
| 8A | CNUSR2-23 | CNUSR2-23 | LAH2 | + terminal of differential encoder Z-phase signal | Input | |
| 9A | - | - | - | Empty | – | |
| 10A | - | - | - | Empty | – | |
| 1B | CNUSR2-15 | CNUSR2-15 | SG | Control power supply 0 V | GND | |
| 2B | CNUSR1-46 | CNUSR13-4 | LAL1 | - terminal of differential encoder A-phase signal | Input | |
| 3B | CNUSR1-47 | CNUSR13-6 | LBL1 | - terminal of differential encoder B-phase signal | Input | CH1 |
| 4B | CNUSR1-48 | CNUSR13-10 | LZL1 | - terminal of differential encoder Z-phase signal | Input | |
| 5B | CNUSR2-40 | CNUSR2-40 | SG | Control power supply 0 V | GND | |
| 6B | CNUSR2-46 | CNUSR2-46 | LAL2 | - terminal of differential encoder A-phase signal | Input | |
| 7B | CNUSR2-47 | CNUSR2-47 | LBL2 | - terminal of differential encoder B-phase signal | Input | CH2 |
| 8B | CNUSR2-48 | CNUSR2-48 | LZL2 | - terminal of differential encoder Z-phase signal | Input | |
| 9B | - | - | - | Empty | – | |
| 10B | - | - | - | Empty | – | |

## 21.4. Chart of sample program

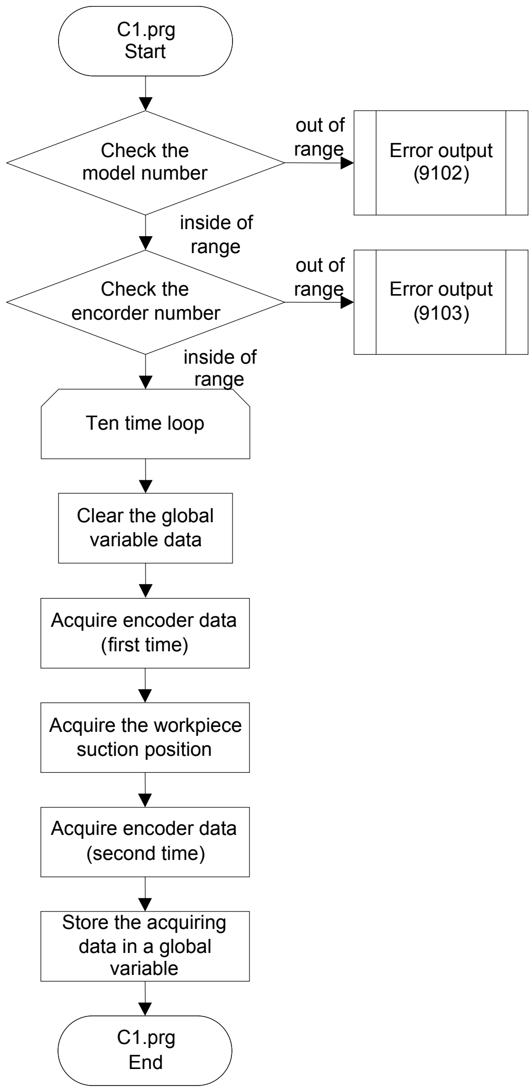
The chart of the sample program is shown below.

### 21.4.1. Conveyer tracking

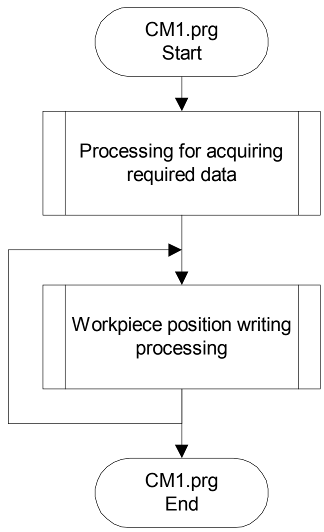
#### (1) A1.prg

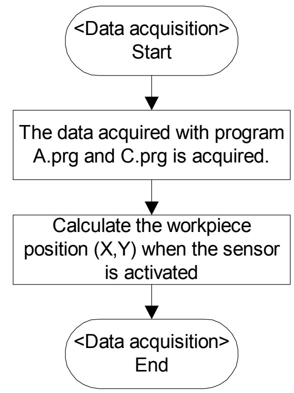


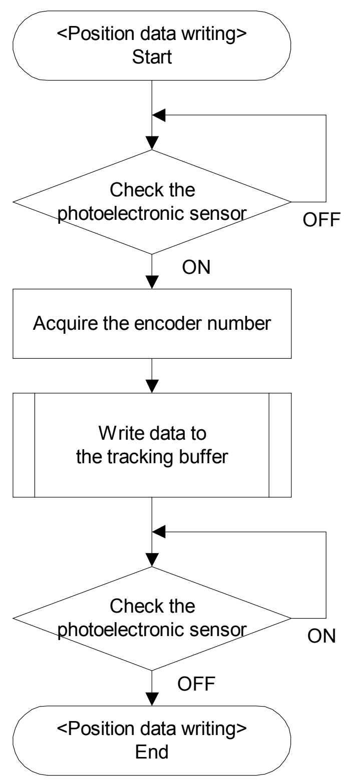
### (2) C1.prg


C1.prg
Start
Check the
model number
out of
range
Error output
(9102)
inside of
range
Check the
encorder number
out of
range
Error output
(9103)
inside of
range
Ten time loop
Clear the global
variable data
Acquire encoder data
(first time)
Acquire the workpiece
suction position
Acquire encoder data
(second time)
Store the acquiring
data in a global
variable
C1.prg
End

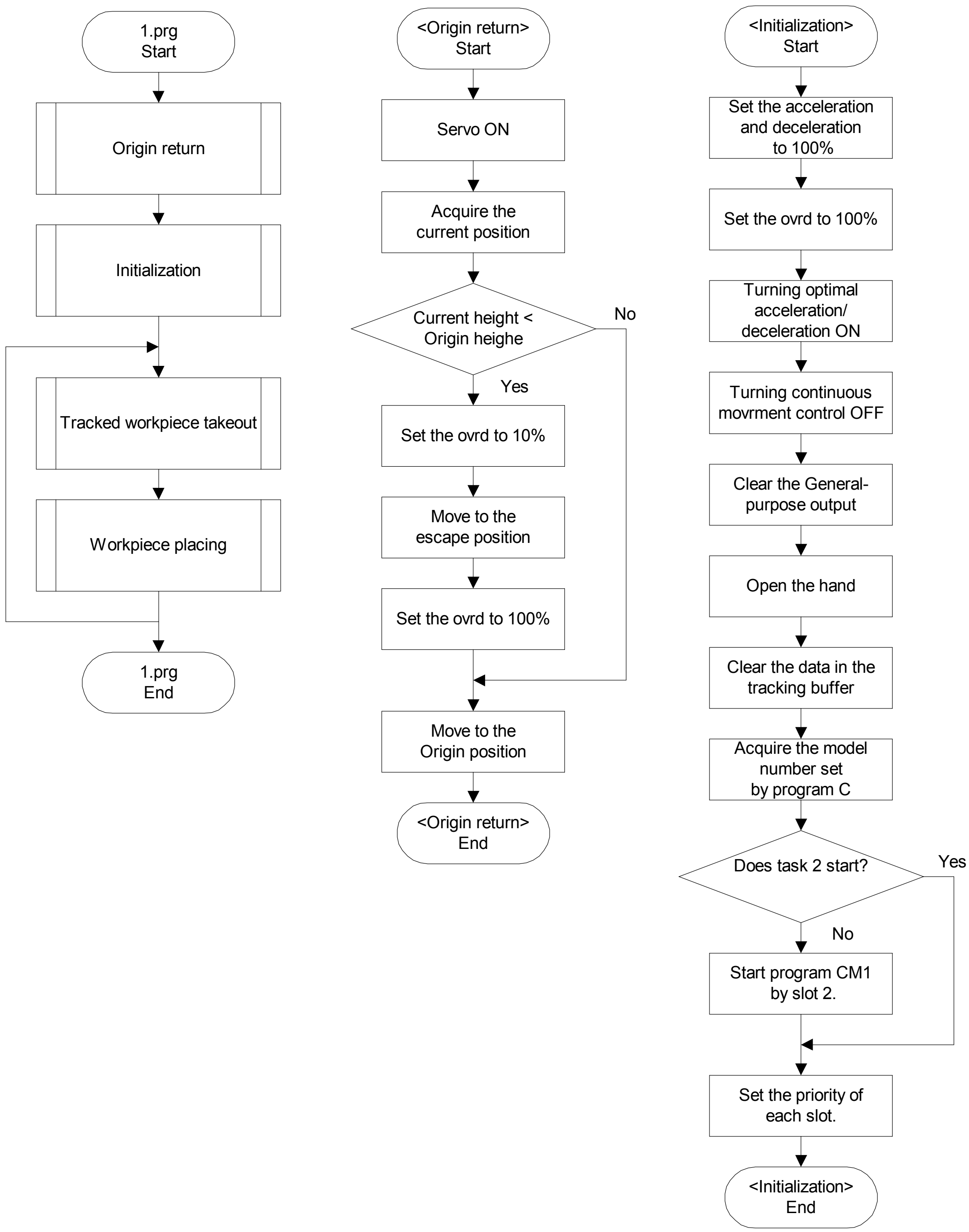
### (3) CM1.prg

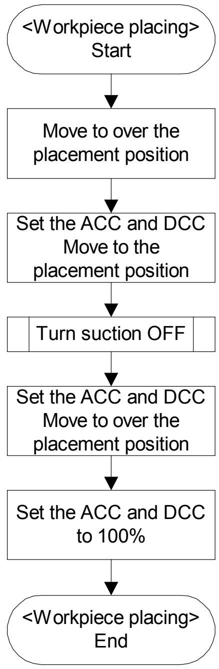

CM1.prg
Start
Processing for acquiring required data
Workpiece position writing processing
CM1.prg
End

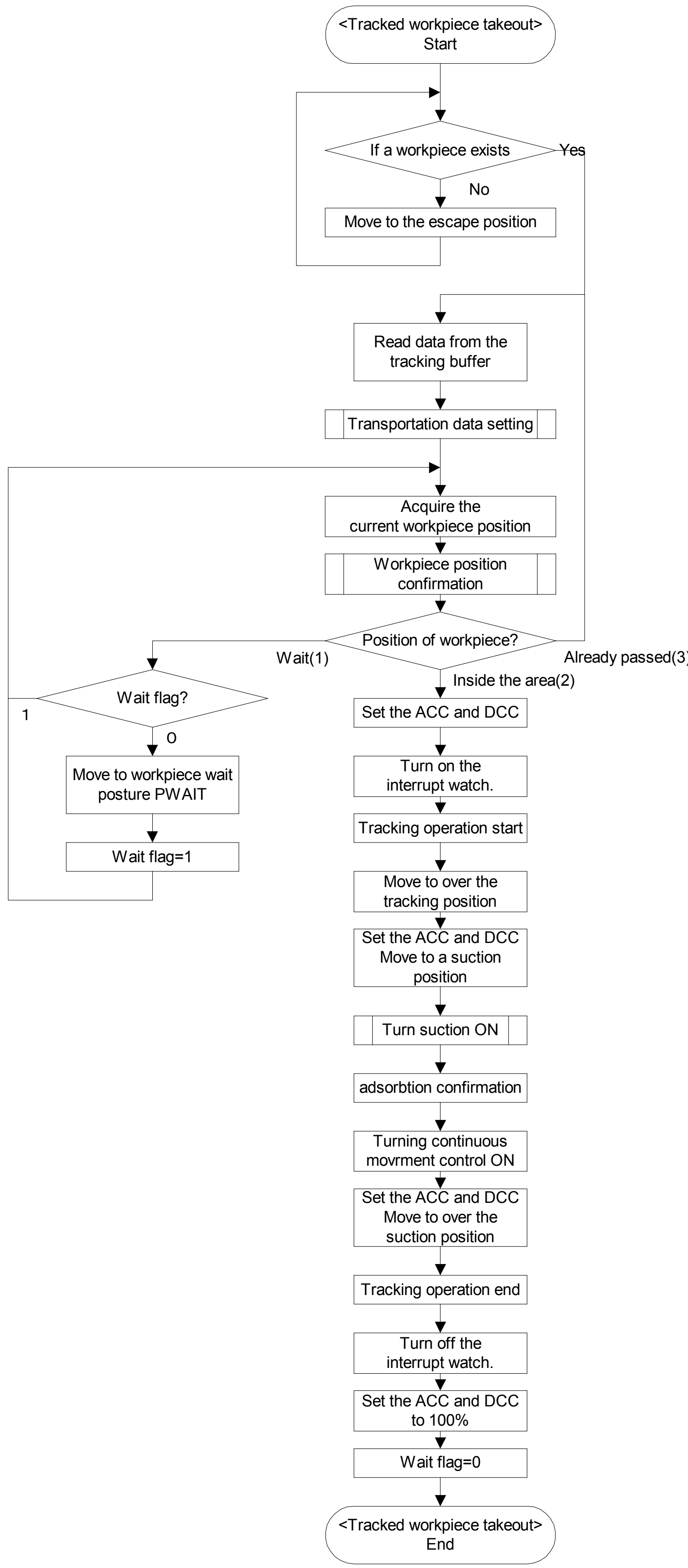

<Data acquisition>
Start
The data acquired with program A.prg and C.prg is acquired.
Calculate the workpiece position (X,Y) when the sensor is activated
<Data acquisition>
End

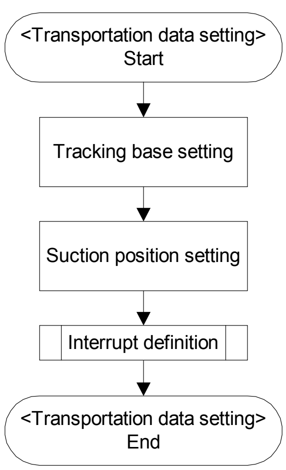

<Position data writing>
Start
Check the photoelectronic sensor
OFF
ON
Acquire the encoder number
Write data to the tracking buffer
Check the photoelectronic sensor
ON
OFF
<Position data writing>
End

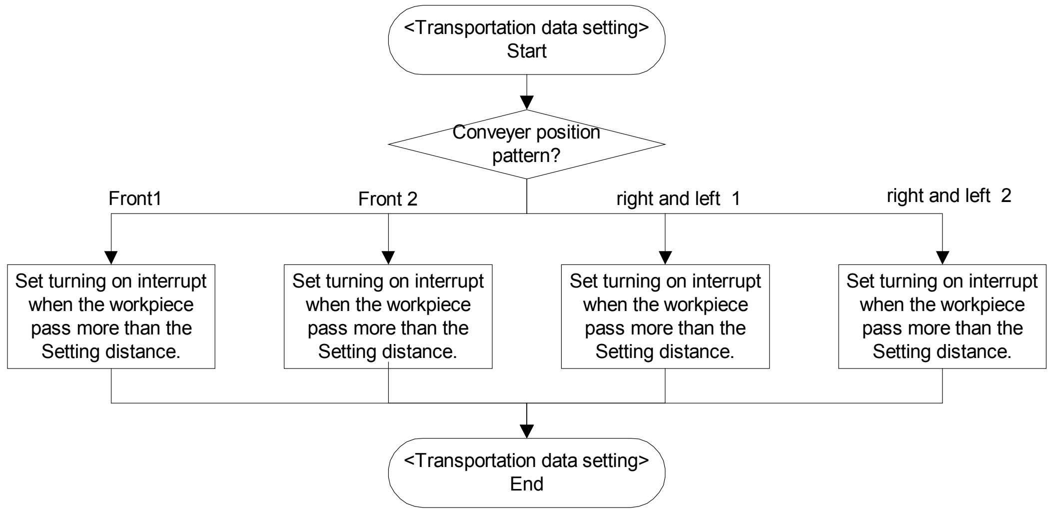
### (4) 1.prg


1.prg
Start
Origin return
Initialization
Tracked workpiece takeout
Workpiece placing
1.prg
End
<Origin return>
Start
Servo ON
Acquire the current position
Current height < Origin heighe
No
Yes
Set the ovrd to 10%
Move to the escape position
Set the ovrd to 100%
Move to the Origin position
<Origin return>
End
<Initialization>
Start
Set the acceleration and deceleration to 100%
Set the ovrd to 100%
Turning optimal acceleration/ deceleration ON
Turning continuous movrment control OFF
Clear the General-purpose output
Open the hand
Clear the data in the tracking buffer
Acquire the model number set by program C
Does task 2 start?
Yes
No
Start program CM1 by slot 2.
Set the priority of each slot.
<Initialization>
End

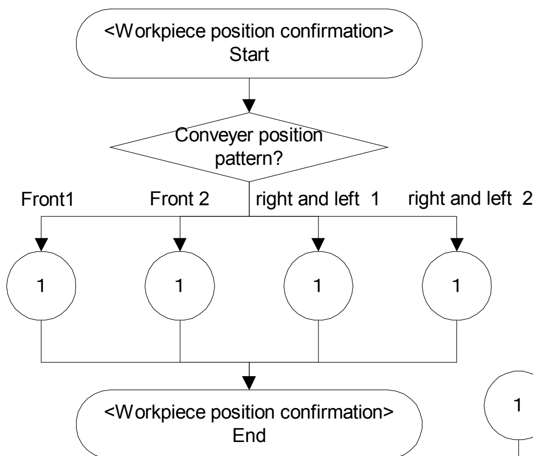

<Workpiece placing> Start
Move to over the placement position
Set the ACC and DCC Move to the placement position
Turn suction OFF
Set the ACC and DCC Move to over the placement position
Set the ACC and DCC to 100%
<Workpiece placing> End


<Tracked workpiece takeout> Start
If a workpiece exists
Yes
No
Move to the escape position
Read data from the tracking buffer
Transportation data setting
Acquire the current workpiece position
Workpiece position confirmation
Position of workpiece?
Wait(1)
Already passed(3)
Inside the area(2)
Wait flag?
1
0
Move to workpiece wait posture PWAIT
Wait flag=1
Set the ACC and DCC
Turn on the interrupt watch.
Tracking operation start
Move to over the tracking position
Set the ACC and DCC Move to a suction position
Turn suction ON
adsorbtion confirmation
Turning continuous movrment control ON
Set the ACC and DCC Move to over the suction position
Tracking operation end
Turn off the interrupt watch.
Set the ACC and DCC to 100%
Wait flag=0
<Tracked workpiece takeout> End

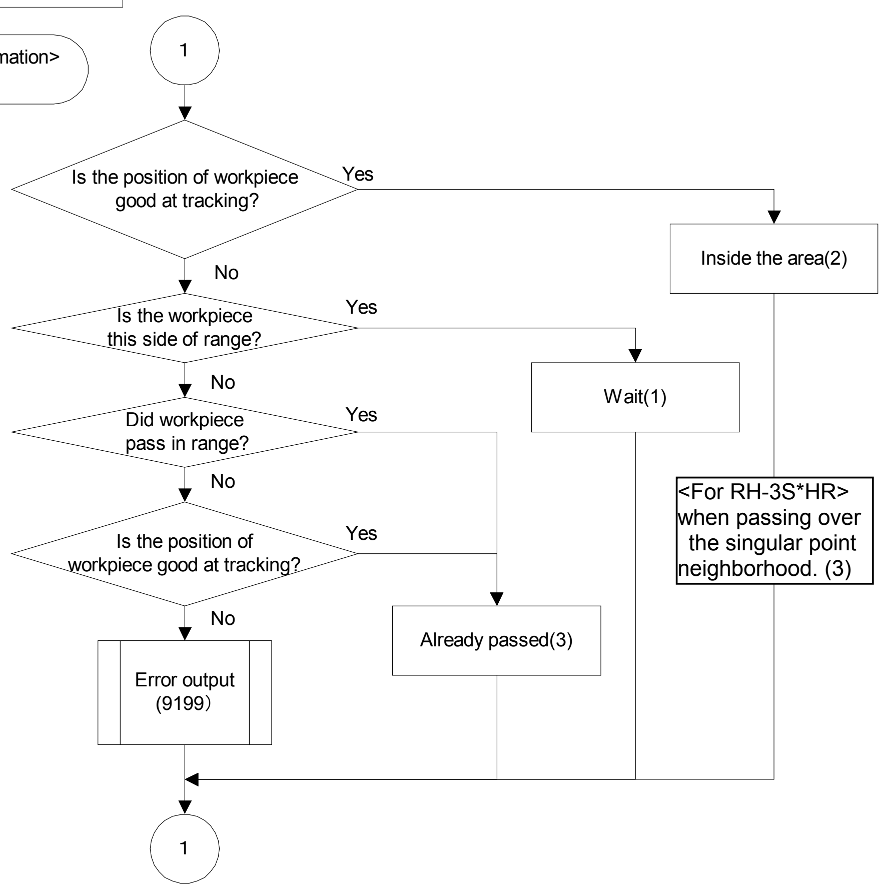

<Transportation data setting> Start
Tracking base setting
Suction position setting
Interrupt definition
<Transportation data setting> End



<Transportation data setting> Start
Conveyer position pattern?
Front1
Front 2
right and left 1
right and left 2
Set turning on interrupt when the workpiece pass more than the Setting distance.
Set turning on interrupt when the workpiece pass more than the Setting distance.
Set turning on interrupt when the workpiece pass more than the Setting distance.
Set turning on interrupt when the workpiece pass more than the Setting distance.
<Transportation data setting> End

<Workpiece position confirmation> Start
Conveyer position pattern?
Front1
Front 2
right and left 1
right and left 2
1
1
1
1
<Workpiece position confirmation> End

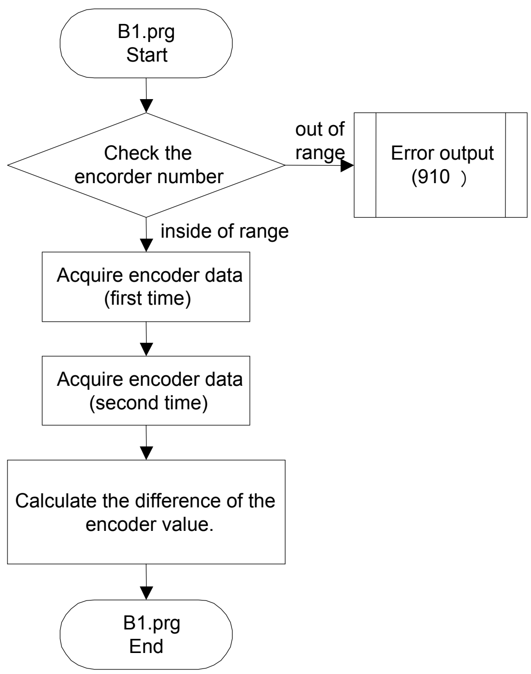
1
Is the position of workpiece good at tracking?
Yes
No
Inside the area(2)
Is the workpiece this side of range?
Yes
No
Wait(1)
Did workpiece pass in range?
Yes
No
<For RH-3S*HR> when passing over the singular point neighborhood. (3)
Is the position of workpiece good at tracking?
Yes
No
Already passed(3)
Error output (9199)
1

### 21.4.2. Vision Tracking

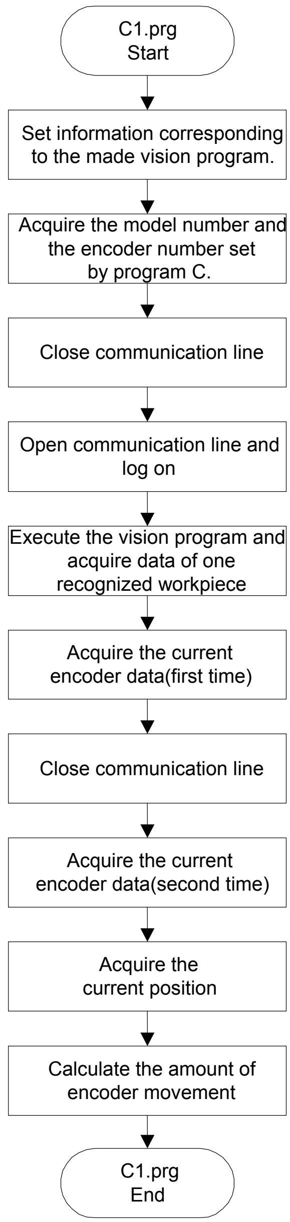
#### (1) A1.prg

The same program as the conveyer tracking.

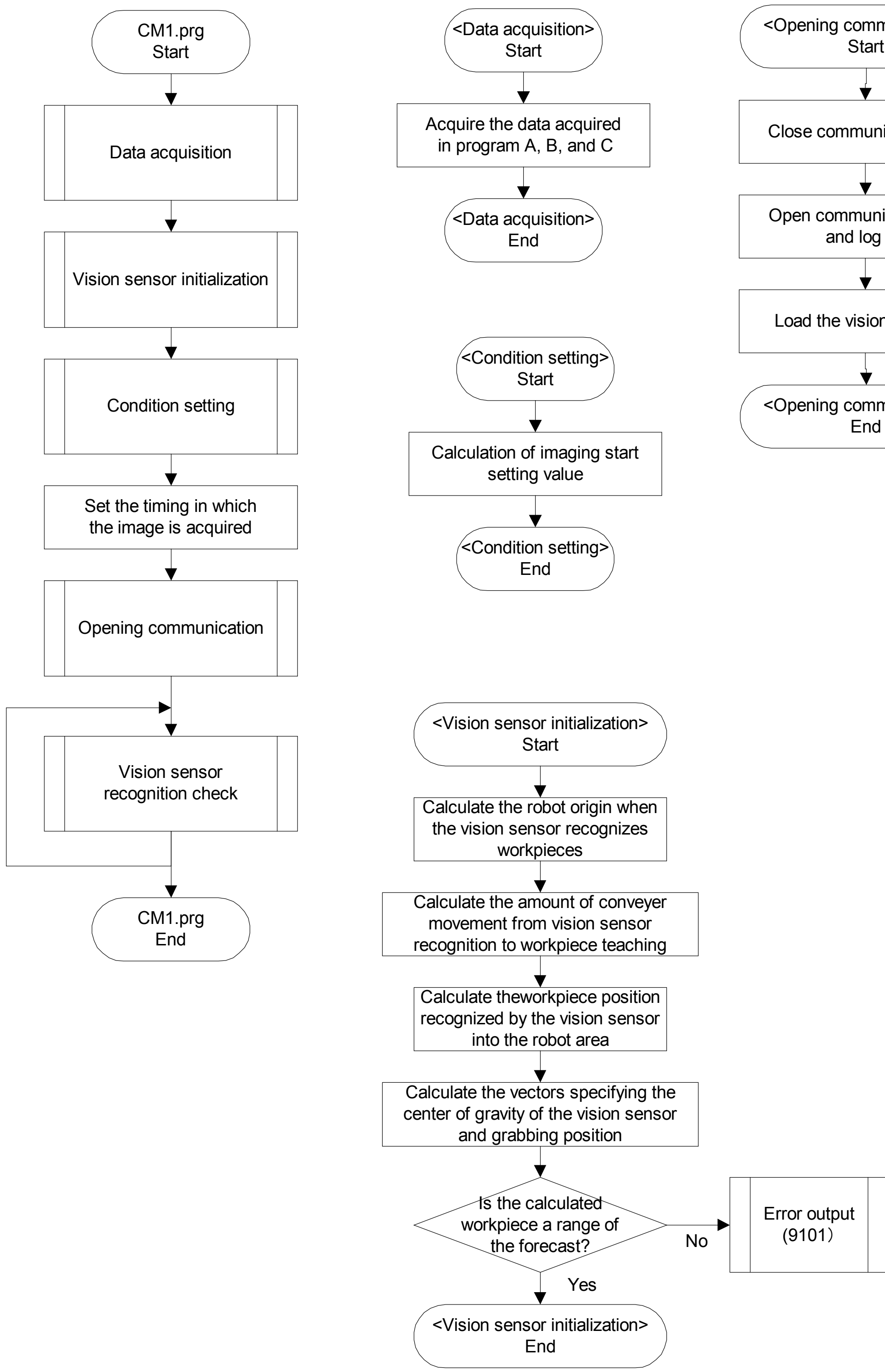
#### (2) B1.prg



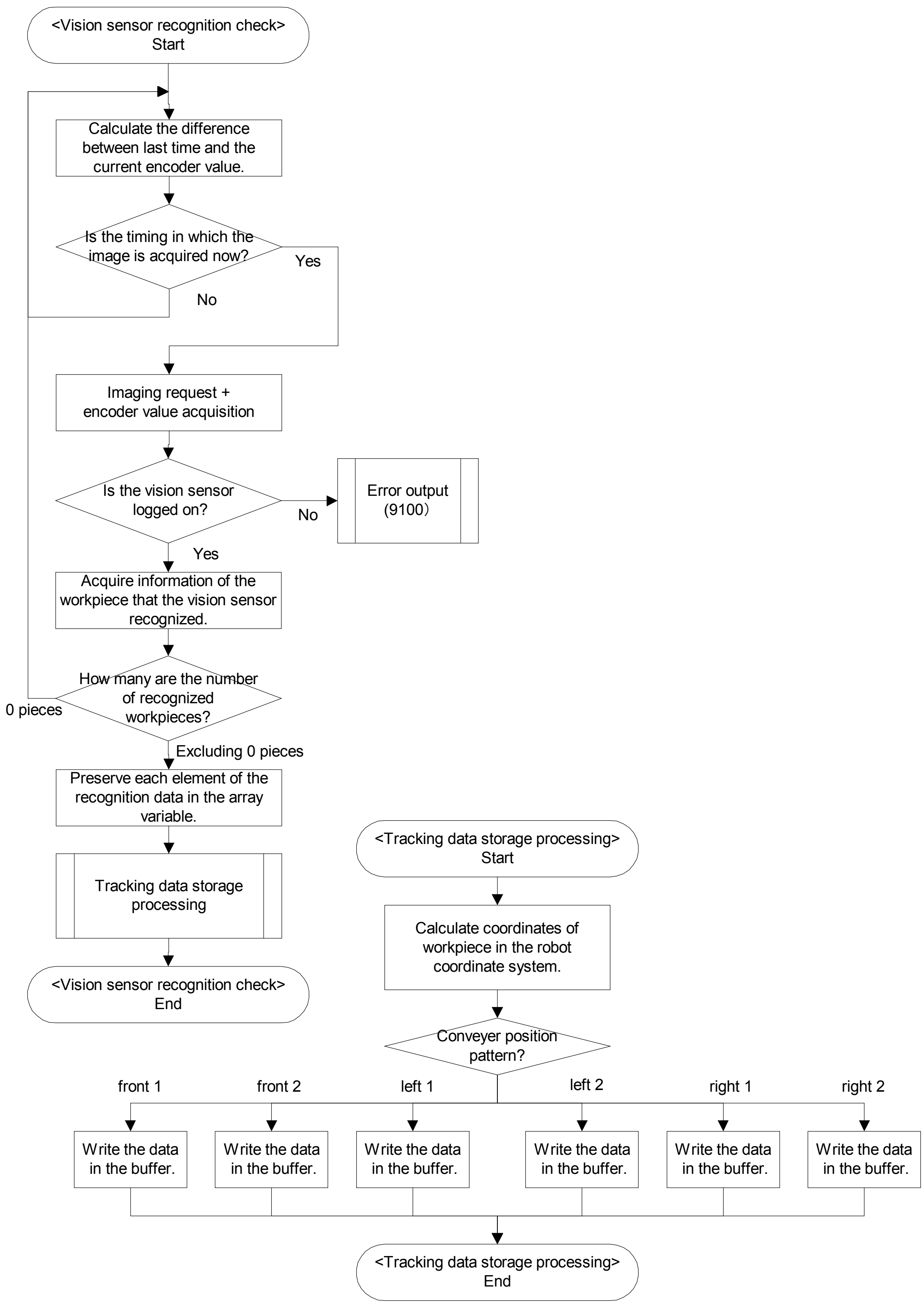
### (3) C.prg


C1.prg
Start
Set information corresponding to the made vision program.
Acquire the model number and the encoder number set by program C.
Close communication line
Open communication line and log on
Execute the vision program and acquire data of one recognized workpiece
Acquire the current encoder data(first time)
Close communication line
Acquire the current encoder data(second time)
Acquire the current position
Calculate the amount of encoder movement
C1.prg
End

## (4) CM1.prg



CM1.prg Start
Data acquisition
Vision sensor initialization
Condition setting
Set the timing in which the image is acquired
Opening communication
Vision sensor recognition check
CM1.prg End
<Data acquisition> Start
Acquire the data acquired in program A, B, and C
<Data acquisition> End
<Condition setting> Start
Calculation of imaging start setting value
<Condition setting> End
<Opening communication> Start
Close communication line
Open communication line and log on
Load the vision program
<Opening communication> End
<Vision sensor initialization> Start
Calculate the robot origin when the vision sensor recognizes workpieces
Calculate the amount of conveyer movement from vision sensor recognition to workpiece teaching
Calculate theworkpiece position recognized by the vision sensor into the robot area
Calculate the vectors specifying the center of gravity of the vision sensor and grabbing position
Is the calculated workpiece a range of the forecast?
No
Error output (9101)
Yes
<Vision sensor initialization> End




### (5) 1.prg

The same program as the conveyer tracking.

## 21.5. Sample Programs

### 21.5.1. Conveyer Tracking

**(1) A1.Prg**

```
'## Ver.A3 #######################################
'# Program for calibration between tracking robot and conveyer
'# Program type : A1.prg
'# Date of creation/version : 2012.07.31 A3
'# COPYRIGHT : MITSUBISHI ELECTRIC CORPORATION.
'################################################
'(1) Register an encoder number to the X coordinate of the "PE" variable/
'Check the setting value
  MECMAX=8                                   'The maximum encoder number value (for checking)
   If PE.X<1 Or PE.X>MECMAX Then Error 9101   'Encoder number out of range
   MENCNO=PE.X                                'Acquire the encoder number
'(2) Attach a marking sticker on the conveyer upstream side/
'(3) Move the robot to the position right at the center of the attached sticker/
   MX10EC1#=M_Enc(MENCNO)                        'Acquire encoder data (first time)
   PX10PS1=P_Zero                            'Set all elements to ZERO
   PX10PS1=P_Fbc(1)                          'Acquire the current position (first time)
'(4) Raise the robot/
'(5) Move the sticker in the forward direction of the conveyer/
'(6) Move the robot to the position right at the center of the moved sticker/
   MX10EC2#=M_Enc(MENCNO)                        'Acquire encoder data (second time)
   PX10PS2=P_Zero                            'Set all elements to ZERO
   PX10PS2=P_Fbc(1)                          'Acquire the current position (second time)
'(7) Raise the robot/
'(8) Perform step operation until END/
   GoSub *S10ENC                             'P_ENCDLT calculation processing
   P_EncDlt(MENCNO)=PY10ENC                     'Store data in P_ENCDLT
End
'
'##### Processing for obtaining P_ENCDLT #####
    'MX10EC1: Encoder data 1
    'MX10EC2: Encoder data 2
    'PX10PS1: Position 1
    'PX10PS2: Position 2
    'PY10ENC: P_ENCDLT value
*S10ENC
   M10ED#=MX10EC2#-MX10EC1#
   If M10ED#>800000000.0# Then M10ED#=M10ED#-1000000000.0#
   If M10ED#<-800000000.0# Then M10ED#=M10ED#+1000000000.0#
   PY10ENC.X=(PX10PS2.X-PX10PS1.X)/M10ED#
   PY10ENC.Y=(PX10PS2.Y-PX10PS1.Y)/M10ED#
   PY10ENC.Z=(PX10PS2.Z-PX10PS1.Z)/M10ED#
   PY10ENC.A=(PX10PS2.A-PX10PS1.A)/M10ED#
   PY10ENC.B=(PX10PS2.B-PX10PS1.B)/M10ED#
   PY10ENC.C=(PX10PS2.C-PX10PS1.C)/M10ED#
   PY10ENC.L1=(PX10PS2.L1-PX10PS1.L1)/M10ED#
   PY10ENC.L2=(PX10PS2.L2-PX10PS1.L2)/M10ED#
Return
'
'This program "computes how much a robot moves per 1 pulse and stores the result in P_ENCDLT."
PE=(+1.00,+0.00,+0.00,+0.00,+0.00,+0.00,+0.00,+0.00)(0,0)
PX10PS1=(+0.00,+0.00,+0.00,+0.00,+0.00,+0.00,+0.00,+0.00)(0,0)
PX10PS2=(+0.00,+0.00,+0.00,+0.00,+0.00,+0.00,+0.00,+0.00)(0,0)
PY10ENC=(+0.00,+0.00,+0.00,+0.00,+0.00,+0.00,+0.00,+0.00)(0,0)
```

**(2) C1.Prg**

```
1 '## Ver.A3 ###########################################
2 '# Conveyer tracking, workpiece suction position registration program
3 '# Program type : C1.prg
4 '# Date of creation/version : 2012.07.31 A3
5 '# COPYRIGHT : MITSUBISHI ELECTRIC CORPORATION.
6 '####################################################
7 '(1) Register a model number in the X coordinate of the "PRM1" variable/
8 '(2) Register an encoder number in the Y coordinate of the "PRM1" variable/
9 '(3) Register the number of the sensor that monitors workpieces in the Z coordinate of the "PRM1" variable /
10 'Check the conditions set in the "PRM1" variable
11    MWKMAX=10                              'The maximum model number value (for checking)
12    MECMAX=8                               'The maximum encoder number value (for checking)
13    MWKNO=PRM1.X                           'Acquire a model number
14    MENCNO=PRM1.Y                          'Acquire an encoder number
15    If MWKNO<1 Or MWKNO>MWKMAX Then Error 9102     'Model number out of range
16    If MENCNO<1 Or MENCNO>MECMAX Then Error 9101 'Encoder number out of range
17    For M1=1 To 10                         'Clear the information
18      P_100(M1)=P_Zero                     'A variable that stores workpiece positions
19      P_102(M1)=P_Zero                     'A variable that stores operation conditions
20      M_101#(M1)=0                         'A variable that stores encoder value differences
21    Next M1
22 '(4) Move a workpiece to the position where the photoelectronic sensor is activated/
23    ME1#=M_Enc(MENCNO)                     'Acquire encoder data (first time)
24 '(5) Move a workpiece on the conveyer into the robot operation area/
25 '(6) Move the robot to the suction position/
26    ME2#=M_Enc(MENCNO)                     'Acquire encoder data (second time)
27    P_100(MWKNO)=P_Fbc(1)                  'Acquire the workpiece suction position (current position)
28 '(7) Perform step operation until END/
29    MED#=ME2#-ME1#                         'Calculate the difference of the encoder value.
30    If MED# > 800000000.0# Then MED# = MED#-1000000000.0#
31    If MED# < -800000000.0# Then MED# = MED#+1000000000.0#
32 '
33    M_101#(MWKNO)=MED#                     'Store the amount of encoder movement in a global variable
34    P_102(MWKNO).X=PRM1.Y                  'Store encoder numbers in a global variable
35    P_102(MWKNO).Y=PRM1.Z                  'Store the sensor number in a global variable
36 End
37 '
38 'This program is "the relation between the position at which the sensor is reacted and the position at which
39 'the robot absorbs workpieces.
PRM1=(+1.00,+1.00,+810.00,+0.00,+0.00,+0.00,+0.00,+0.00)(,)
```

**(3) 1.Prg**

```
'### Ver.A3 ##################################
'# Conveyer tracking, robot operation program
'# Program type : 1.prg
'# Date of creation/version : 2012.07.31 A3
'# MITSUBISHI ELECTRIC CORPORATION.
'#############################################
'
'### Main processing ###
*S00MAIN
    GoSub *S90HOME                                      'Origin return processing
    GoSub *S10INIT                                      'Initialization processing
*LOOP
    GoSub *S20TRGET                                     'Tracked workpiece takeout processing
    GoSub *S30WKPUT                                     'Workpiece placing processing
    GoTo *LOOP
End
'
'###   Initialization processing ###
*S10INIT
'/// Speed related ///
    Accel 100,100                                       'Acceleration/deceleration setting
    Ovrd 100                                            'Speed setting
    Loadset 1,1                                         'Optimal acceleration/deceleration specification
    OAdl On                                             'Turning optimal acceleration/deceleration ON
    Cnt 0
    Clr 1
    HOpen 1
'/// Initial value setting ///
    TrClr 1                                             'Clear tracking buffer 1
    MWAIT1=0                                            'Clear workpiece wait flag 1
'/// Multitask startup ///
    M_09#=PWK.X                                         'Model number specification
    If M_Run(2)=0 Then                                  'Confirmation of conveyer 1 multitasking
      XRun 2,"CM1",1                                    'Multitasking setting
      Wait M_Run(2)=1
    EndIf
    Priority PRI.X,1
    Priority PRI.Y,2
Return
'
'### Tracked workpiece takeout processing ###
*S20TRGET
'/// Tracking buffer check ///
*LBFCHK
    If M_Trbfct(1)>=1 Then GoTo *LREAD                  'If a workpiece exists
    Mov P1                                              'Move to the pull-off location
    MWAIT1=0
    GoTo *LBFCHK
'/// Workpiece data acquisition ///
*LREAD
    TrRd PBPOS,MBENC#,MBWK%,1,MBENCNO%                  'Read data from the tracking buffer
    GoSub *S40DTSET                                     'Transportation data setting
'/// Workpiece position confirmation ///
*LNEXT
    PX50CUR=TrWcur(MBENCNO%,PBPOS,MBENC#)               'Acquire the current workpiece position
    MX50ST=PRNG.X                                       'Start distance of the range where the robot can
follow a workpiece
```

```
57     MX50ED=PRNG.Y                    'End distance of the range where the robot can follow a workpiece
58     MX50PAT=PTN.X                    'Conveyer position pattern number
59     GoSub *S50WKPOS                  'Workpiece position confirmation processing
60     If MY50STS=3 Then GoTo *LBFCHK   'Already passed. Go to the next workpiece
61     If MY50STS=2 Then GoTo *LTRST    'Operable: start tracking
62     If MWAIT=1 Then GoTo *LNEXT      'Wait for incoming workpieces
63 '/// To standby position ///
64     PWAIT=P1                         'Change to workpiece wait posture
65     Select PTN.X                     'Conveyer position pattern number
66     Case 1 To 2                      'When the conveyer is the front of the robot
67       PWAIT.X=PX50CUR.X              'X coordinates of the robot are matched to workpiece.
68     Case 3 To 6
69       PWAIT.Y=PX50CUR.Y              'Y coordinates of the robot are matched to workpiece.
70     End Select
71     PWAIT.Z=PX50CUR.Z+PUP1.X
72     PWAIT.C=PX50CUR.C
73     Mov PWAIT                        'Move to workpiece wait posture PWAIT
74     MWAIT1=1                         'Set workpiece wait flag
75     GoTo *LNEXT
76 '/// Start tracking operation ///
77 *LTRST
78     Accel PAC1.X,PAC1.Y
79     Cnt 1,0,0
80     Act 1=1                          'Monitor the robot following workpieces too far
81     Trk On,PBPOS,MBENC#,PTBASE,MBENCNO%     'Tracking operation start setting
82     Mov PGT,PUP1.Y Type 0,0          'Move to tracking midair position
83     Accel PAC2.X,PAC2.Y
84     Mov PGT Type 0,0                 'Move to a suction position
85     GoSub *S85CLOSE                  'Turn suction ON
86     MX80ENA=PHND.X                   'Check instruction
87     MX80SIG=PHND.Y                   'Check signal number
88     MX80SEC=PDLY1.X                  'Check second number(s)
89     GoSub *S80CWON                   'adsorbtion confirmation
90     Cnt 1
91     Accel PAC3.X,PAC3.Y
92     Mov PGT,PUP1.Z Type 0,0          'Move to tracking midair position
93     Trk Off                          'Tracking operation end setting
94     Act 1=0
95     Accel 100,100
96     MWAIT = 0
97 Return
98 '
99 '### Workpiece placing processing ###
100 *S30WKPUT
101    Accel PAC11.X,PAC11.Y
102    Mov PPT,PUP2.Y                   'Move to over the placement position
103    Accel PAC12.X,PAC12.Y
104    Cnt 1,0,0
105    Mov PPT Type 0,0                 'Move to the placement position
106    GoSub *S86OPEN                   'Turn suction OFF
107    MX81ENA=PHND.X                   'Check instruction
108    MX81SIG=PHND.Z                   'Check signal number
109    MX81SEC=PDLY2.X                  'Check second number(s)
110    GoSub *S81CWOFF                  'Release confirmation
111    Cnt 1
112    Accel PAC13.X,PAC13.Y
113    Mov PPT,PUP2.Z Type 0,0          'Move to over the placement position
```

```
    Accel 100,100
Return
'
'### Transportation data setting processing ###
*S40DTSET
    PTBASE=P_100(PWK.X)                          'Create reference position
    TrBase PTBASE,MBENCNO%                        'Tracking base setting
    PGT=PTBASE*POFSET                             'Suction position setting
    GoSub *S46ACSET                              'Interrupt definition
Return
'
'### Interrupt definition processing 1 ###
*S46ACSET
    Select PTN.X                                'Conveyer position pattern number
    Case 1 'Front right -> left
      MSTP1=PRNG.Z                               'Following stop distance
      Def Act 1,P_Fbc(1).Y>MSTP1 GoTo *S91STOP    'To *S91STOP if followed far long
      Break
    Case 2 'Front left -> right
      MSTP1=-PRNG.Z
      Def Act 1,P_Fbc(1).Y<MSTP1 GoTo *S91STOP
      Break
    Case 3 'Left side rear -> front
    Case 5 'Right side rear -> front
      MSTP1=PRNG.Z
      Def Act 1,P_Fbc(1).X>MSTP1 GoTo *S91STOP
      Break
    Case 4 'Left side front -> rear
    Case 6 'Right side front -> rear
      MSTP1=-PRNG.Z
      Def Act 1,P_Fbc(1).X<MSTP1 GoTo *S91STOP
      Break
    End Select
Return
'
'### Workpiece position confirmation processing ###
     'PX50CUR:Current workpiece position
     'MX50ST:Tracking start range
     'MX50ED:Tracking end range
     'MX50PAT:Conveyer position pattern number
     'MY50STS:Result (1: Wait/2: Start tracking/3: Next workpiece)
*S50WKPOS
    MY50STS=0                                   'Clear return value
    Select MX50PAT                              'Conveyer pattern
    Case 1 'Front right -> left
      M50STT=-MX50ST                             'The start side has a negative value
      M50END=MX50ED
      If PosCq(PX50CUR)=1 And PX50CUR.Y>=M50STT And PX50CUR.Y<=M50END Then
        MY50STS=2                                'Tracking possible
      Else 'If tracking not possible
        If PX50CUR.Y<0 Then MY50STS=1              'Wait
        If PX50CUR.Y>M50END Then MY50STS=3          'Move onto the next workpiece
        If PosCq(PX50CUR)=0 And PX50CUR.Y>=M50STT And PX50CUR.Y<=M50END Then MY50STS=3 'Outside the movement range
      EndIf
      Break
    Case 2 'Front left -> right
      M50STT=MX50ST
      M50END=-MX50ED                              'The end side has a negative value
      If PosCq(PX50CUR)=1 And PX50CUR.Y<=M50STT And PX50CUR.Y>=M50END Then
```

```
      MY50STS=2                                      'Tracking possible
    Else 'If tracking not possible
      If PX50CUR.Y>0 Then MY50STS=1                   'Wait
      If PX50CUR.Y<0 Then MY50STS=3                   'Move onto the next workpiece
      If PosCq(PX50CUR)=0 And PX50CUR.Y<=M50STT And PX50CUR.Y>=M50END Then MY50STS=3 'Outside the movement range
    EndIf
    Break
  Case 3 'Left side rear -> front
  Case 5 'Right side rear -> front
    M50STT=-MX50ST                                   'The start side has a negative value
    M50END=MX50ED
    If PosCq(PX50CUR)=1 And PX50CUR.X>=M50STT And PX50CUR.X<=M50END Then
      MY50STS=2                                      'Tracking possible
    Else 'If tracking not possible
      If PX50CUR.X<0 Then MY50STS=1                   'Wait
      If PX50CUR.X>0 Then MY50STS=3                   'Move onto the next workpiece
      If PosCq(PX50CUR)=0 And PX50CUR.X>=M50STT And PX50CUR.X<=M50END Then MY50STS=3 'Outside the movement range
    EndIf
    Break
  Case 4 'Left side front -> rear
  Case 6 'Right side front -> rear
    M50STT=MX50ST
    M50END=-MX50ED                                   'The end side has a negative value
    If PosCq(PX50CUR)=1 And PX50CUR.X<=M50STT And PX50CUR.X>=M50END Then
      MY50STS=2                                      'Tracking possible
    Else 'If tracking not possible
      If PX50CUR.X>0 Then MY50STS=1                   'Wait
      If PX50CUR.X<0 Then MY50STS=3                   'Move onto the next workpiece
      If PosCq(PX50CUR)=0 And PX50CUR.X<=M50STT And PX50CUR.X>=M50END Then MY50STS=3 'Outside the movement range
    EndIf
    Break
  End Select
  If MY50STS=0 Then Error 9199                       'Program modification required
Return
'
'### Origin return processing ###
*S90HOME
  Servo On                                           'Servo ON
  P90CURR=P_Fbc(1)                                   'Acquire the current position
  If P90CURR.Z<P1.Z Then                             'If the current height is below the origin
    Ovrd 10
    P90ESC=P90CURR                                   'Create an escape position
    P90ESC.Z=P1.Z
    Mvs P90ESC                                       'Move to the escape position
    Ovrd 100
  EndIf
  Mov P1                                             'Move to the origin
Return
'
'### Tracking interruption processing ###
*S91STOP
  Act 1=0
  Trk Off
  GoSub *S86OPEN                                     'Release suction
  P91P=P_Fbc(1)                                      'Acquire the current position
  P91P.Z=P1.Z
  Mvs P91P Type 0,0                                  'Raise
```

```
230    Mov P1                                   'Return to the origin once
231    GoTo *LBFCHK
232 '
233 '##### Suction of substrates #####
234 *S85CLOSE
235    HClose 1                                 'Turn suction ON
236 Return
237 '##### Suction/release of substrates #####
238 *S86OPEN
239    HOpen 1                                  'Turn suction OFF
240 Return
241 '
242 '##### Turning on the signal is waited for #####
243    'MX80ENA:ENABLE/DISABLE of check(1/0)
244    'MX80SIG:Check signal number
245    'MX80SEC:Check second number(S)
246    'MY80SKP:OK/TIMEOUT(1/0)
247 *S80CWON
248    If MX80ENA=1 Then                        'If the signal check is ENABLE
249      M_Timer(1)=0
250      MY80SKP=0
251      MX80SEC=MX80SEC * 1000                 'Second -> Millisecond
252 *L80LOP
253      If (M_Timer(1)>MX80SEC) Or (MY80SKP<>0) Then *L80END
254        If M_In(MX80SIG)=1 Then MY80SKP=1    'If the signal specified is turned on
255      GoTo *L80LOP
256    Else                                     'If the signal check is DISABLE
257      Dly MX80SEC                            'Wait at the specified check time
258      MY80SKP=1                              'OK
259    EndIf
260 *L80END
261 Return
262 '
263 '##### Turning off the signal is waited for #####
264    'MX81ENA:ENABLE/DISABLE of check(1/0)
265    'MX81SIG:Check signal number
266    'MX81SEC:Check second number(S)
267    'MY81SKP:OK/TIMEOUT(1/0)
268 *S81CWOFF
269    If MX81ENA=1 Then                        'If the signal check is ENABLE
270      M_Timer(1)=0
271      MY81SKP=0
272      MX81SEC=MX81SEC * 1000                 'Second -> Millisecond
273 *L81LOP
274      If (M_Timer(1)>MX81SEC) Or (MY81SKP<>0) Then *L81END
275        If M_In(MX81SIG)=0 Then MY81SKP=1    'If the signal specified is turned off
276      GoTo *L81LOP
277    Else                                     'If the signal check is DISABLE
278      Dly MX80SEC                            'Wait at the specified check time
279      MY81SKP=1 'OK
280    EndIf
281 *L81END
282 Return
PWK=(+1.00,+0.00,+0.00,+0.00,+0.00,+0.00,+0.00,+0.00)(0,0)
PRI=(+1.00,+1.00,+0.00,+0.00,+0.00,+0.00,+0.00,+0.00)(0,0)
P1=(+0.00,+0.00,+0.00,+0.00,+0.00,+0.00,+0.00,+0.00)(0,0)
PBPOS=(+0.00,+0.00,+0.00,+0.00,+0.00,+0.00,+0.00,+0.00)(0,0)
PX50CUR=(+0.00,+0.00,+0.00,+0.00,+0.00,+0.00,+0.00,+0.00)(0,0)
PRNG=(+300.00,+200.00,+400.00,+0.00,+0.00,+0.00,+0.00,+0.00)(0,0)
PTN=(+1.00,+0.00,+0.00,+0.00,+0.00,+0.00,+0.00,+0.00)(0,0)
```

```
PWAIT=(+0.00,+0.00,+0.00,+0.00,+0.00,+0.00,+0.00,+0.00)(0,0)
PUP1=(+50.00,+0.00,+0.00,+0.00,+0.00,+0.00,+0.00,+0.00)(0,0)
PAC1=(+100.00,+100.00,+0.00,+0.00,+0.00,+0.00,+0.00,+0.00)(0,0)
PTBASE=(+0.00,+0.00,+0.00,+0.00,+0.00,+0.00,+0.00,+0.00)(0,0)
PGT=(+0.00,+0.00,+0.00,+0.00,+0.00,+0.00,+0.00,+0.00)(0,0)
PAC2=(+100.00,+100.00,+0.00,+0.00,+0.00,+0.00,+0.00,+0.00)(0,0)
PHND=(+0.00,+900.00,+900.00,+0.00,+0.00,+0.00,+0.00,+0.00)(0,0)
PDLY1=(+1.00,+0.00,+0.00,+0.00,+0.00,+0.00,+0.00,+0.00)(0,0)
PAC3=(+100.00,+100.00,+0.00,+0.00,+0.00,+0.00,+0.00,+0.00)(0,0)
PAC11=(+100.00,+100.00,+0.00,+0.00,+0.00,+0.00,+0.00,+0.00)(0,0)
PPT=(+0.00,+0.00,+0.00,+0.00,+0.00,+0.00,+0.00,+0.00)(0,0)
PUP2=(+0.00,+0.00,+0.00,+0.00,+0.00,+0.00,+0.00,+0.00)(0,0)
PAC12=(+100.00,+100.00,+0.00,+0.00,+0.00,+0.00,+0.00,+0.00)(0,0)
PDLY2=(+1.00,+0.00,+0.00,+0.00,+0.00,+0.00,+0.00,+0.00)(0,0)
PAC13=(+100.00,+100.00,+0.00,+0.00,+0.00,+0.00,+0.00,+0.00)(0,0)
POFSET=(+0.00,+0.00,+0.00,+0.00,+0.00,+0.00,+0.00,+0.00)(0,0)
P90CURR=(+0.00,+0.00,+0.00,+0.00,+0.00,+0.00,+0.00,+0.00)(0,0)
P90ESC=(+0.00,+0.00,+0.00,+0.00,+0.00,+0.00,+0.00,+0.00)(0,0)
P91P=(+0.00,+0.00,+0.00,+0.00,+0.00,+0.00,+0.00,+0.00)(0,0)
```

**(4) CM1.Prg**

```
1 '## Ver.A3 ######################################
2 '# Conveyer tracking, sensor monitoring program
3 '# Program type : CM1.prg
4 '# Date of creation/version : 2012.07.31 A3
5 '# COPYRIGHT : MITSUBISHI ELECTRIC CORPORATION.
6 '##############################################
7 '
8 '##### Main processing #####
9 *S00MAIN
10    GoSub *S10DTGET                    'Processing for acquiring required data
11 *LOOP
12    GoSub *S20WRITE                    'Workpiece position writing processing
13    GoTo *LOOP
14 End
15 '##### Data acquisition processing #####
16 *S10DTGET
17 'Acquire the suction position, amount of encoder movement and encoder number set with program C
18    MWKNO=M_09#                        'Acquire model number
19    M10ED#=M_101#(MWKNO)               'Amount of encoder movement
20    MENCNO=P_102(MWKNO).X              'Encoder number
21    MSNS=P_102(MWKNO).Y                'Sensor number
22 'Calculate the workpiece position (X,Y) when the sensor is activated
23    PWPOS=P_100(MWKNO)-P_EncDlt(MENCNO)*M10ED#
24 Return
25 '##### Position data writing processing #####
26 *S20WRITE
27    If M_In(MSNS)=0 Then GoTo *S20WRITE   'Wait for a workpiece to activate the photoelectronic sensor
```

CR750-Q/CR751-Q series, CRnQ-700 series controller

```
28    MENC#=M_EncL(MENCNO)               'Encoder number
```

CR750-D/CR751-D series, CRnD-700 series controller

```
28    MENC#=M_Enc(MENCNO)                'Encoder number
```

（Note）
The command is deferent between iQ Platform controller (CR750-Q/CR751-Q series, CRnQ-700 series) and stand alone type controller (CR750-D/CR751-D series, CRnD-700 series).
In the CR750-Q/CR751-Q series, CRnQ-700 series series, it is necessary to use the latch encoder data (M_ENCL) after confirmation with an input signal.

```
29    TrWrt PWPOS,MENC#,MWKNO,1,MENCNO   'Write data (workpiece position and encoder value) to the tracking buffer
30 *L20WAIT
31    If M_In(MSNS)=1 Then GoTo *L20WAIT
32 Return
```

### 21.5.2. Vision Tracking

**(1) A1.Prg**

The same program as the conveyer tracking.

**(2) B1.Prg**

```
1 '### Ver.A3 ##########################################
2 '# Network vision tracking, calibration between robot and vision sensor
3 '# Program type          : B1.prg
4 '# Date of creation    : 2012.07.31 A3
5 '# COPYRIGHT : MITSUBISHI ELECTRIC CORPORATION.
6 '#####################################################
7 '(1) Register an encoder number to the X coordinate of the “PE” variable/
8 'Check the setting value
9     MECMAX=8                                        'The maximum encoder number value (for checking)
10     If PE.X<1 Or PE.X>MECMAX Then Error 9101    'Encoder number out of range
11     MENCNO=PE.X                                   'Acquire the encoder number
12 '(2) Place the calibration sheet within the vision sensor recognition area/
13 '(3) Check that the calibration sheet positions are correct by looking at vision images/
14     ME1#=M_Enc(MENCNO)                            'Acquire encoder data (first time)
15 '(4) Specify the mark in three points or more by using "Mitsubishi Robot Tool" on "In-Sight Explorer"/
16 '(5) Move the calibration sheet until they are within the robot operation area/
17 '(6) Move the robot hand to the position right at the center of mark 1/
18 '(7) Acquire the robot present position by using "In-Sight Explorer"/
19 '(8) Acquire the position of the robot in three points or more repeating work/
20 '(9) Click the Export button. Then, the calibration data can be made/
21 '(10) Raise the robot arm/
22     ME2#=M_Enc(MENCNO)                            'Acquire encoder data (second time)
23     MED#=ME1#-ME2#                                'Calculate the difference of the encoder value.
24     If MED# > 800000000.0# Then MED# = MED#-1000000000.0#
25     If MED# < -800000000.0# Then MED# = MED#+1000000000.0#
26     M_100#(MENCNO)=MED#
27 End
PE=(+1.00,+0.00,+0.00,+0.00,+0.00,+0.00,+0.00,+0.00)(0,0)
```

**(3) C1.Prg**

```
'### Ver.A3 ##########################################
'# Network vision tracking, workpiece suction position registration program
'# Program type                    : C1.prg
'# Date of creation/version    : 2012.07.31 A3
'# COPYRIGHT : MITSUBISHI ELECTRIC CORPORATION.
'####################################################
'(1) Store a model number in the X coordinate of the "PRM1" variable/
'(2) Store an encoder number in the Y coordinate of the "PRM1" variable/
'(3) Check live images and register the length in the movement direction to the X coordinate of the "PRM2" variable/
'(4) Store the workpiece length in the Y coordinate of the "PRM2" variable/
'(5) Enter the COM port number to be opened for communication after "CCOM$=" in the following line/
    CCOM$="COM2:"                                    'Set the number of the port to be opened
'(6) Enter the vision program name after "CPRG$=" in the following line/
    CPRG$="TRK.JOB"                                  'Set the vision program name
'(7) Place workpieces to be tracked in locations recognizable by the vision sensor/
'(8) Place the vision sensor in the "online" status/
'(9) When the program stops, open program C1 with T/B/
    MWKNO=PRM1.X                                     'Acquire the model number
    MENCNO=PRM1.Y                                    'Acquire the encoder number
'Establish a communication line with the vision sensor via the opened port
    NVClose                                          'Close communication line
    NVOpen CCOM$ As #1                               'Open communication line and log on
    Wait M_NvOpen(1)=1                               'Wait to log on to the vision sensor
    EBRead #1,"",MNUM,PVS1,PVS2,PVS3,PVS4    'Acquire data of one recognized workpiece
    P_101(MWKNO)=PVS1                                'Acquire data of the first recognized workpiece
    ME1#=M_Enc(MENCNO)                               'Acquire encoder data 1
    NVClose #1
Hlt
'(10) Move a workpiece on the conveyer until it gets within the robot operation area/
'(11) Move the robot to the suction position/
    ME2#=M_Enc(MENCNO)                               'Acquire encoder data 2
    P_100(MWKNO)=P_Fbc(1)                            'Acquire position 1
'(12) Perform step operation until END/
    MED#=ME2#-ME1#                                   'Calculate the amount of encoder movement
    If MED# > 800000000.0# Then MED# = MED#-1000000000.0#
    If MED# < -800000000.0# Then MED# = MED#+1000000000.0#
    M_101#(MWKNO)=MED#                               'Amount of encoder movement
    P_102(MWKNO)=PRM1                                'Encoder number
    P_103(MWKNO)=PRM2                                'Image size and workpiece size
    C_100$(MWKNO)=CCOM$                              'COM port number
    C_101$(MWKNO)=CPRG$                              'Vision program name
End
'
'This program is "the relation between the workpiece position recognized by the network vision sensor and
' the position at which the robot suctions workpieces.
PRM1=(+1.00,+1.00,+0.00,+0.00,+0.00,+0.00,+0.00,+0.00)(0,0)
PVS1=(+0.00,+0.00,+0.00,+0.00,+0.00,+0.00,+0.00,+0.00)(0,0)
PVS2=(+0.00,+0.00,+0.00,+0.00,+0.00,+0.00,+0.00,+0.00)(0,0)
PVS3=(+0.00,+0.00,+0.00,+0.00,+0.00,+0.00,+0.00,+0.00)(0,0)
PVS4=(+0.00,+0.00,+0.00,+0.00,+0.00,+0.00,+0.00,+0.00)(0,0)
PRM2=(+170.00,+30.00,+0.00,+0.00,+0.00,+0.00,+0.00,+0.00)(0,0)
```

**(4) 1.Prg**

The same program as the conveyer tracking.

**(5) CM1.Prg**

```
'### Ver.A3 ####################################
'# Conveyer tracking, communication processing between robot and vision sensor
'# Program type          : VS communication program
'# Date of creation/version : 2012.07.31 A3
'# COPYRIGHT : MITSUBISHI ELECTRIC CORPORATION.
'################################################
    Dim MX(4),MY(4),MT(4),PVS(4)                    'X/Y/C/buffer
'
'##### Main processing #####
*S00MAIN
    GoSub *S10DTGET                              'Data acquisition processing
    GoSub *S20VSINI                             'VS initialization processing
    GoSub *S30CONST                              'Condition setting
'
    MEP# = M_Enc(MENCNO)+MEI#+100
    GoSub *S70VOPEN                              'Vision sensor line open + vision program load processing
*L00_00
    GoSub *S40CHKS                               'VS recognition check processing
    GoTo *L00_00
End
'
'##### Data acquisition processing #####
*S10DTGET
    MWKNO=M_09#                                  'Model number
    MENCNO=P_102(MWKNO).Y                          'Encoder number
    MVSL=P_103(MWKNO).X                           'VS screen size longitudinal distance
    MWKL=P_103(MWKNO).Y                            'Workpiece size longitudinal distance
'
    PTEACH=P_100(MWKNO)                            'Position taught to the robot
    PVSWRK=P_101(MWKNO)                             'Position recognized by VS
    CCOM$=C_100$(MWKNO)                             'COM port number
    CPRG$=C_101$(MWKNO)                            'Vision program name
Return
'
'##### Opening communication line #####
*S70VOPEN
    NVClose                                     'Close communication line
    NVOpen CCOM$ As #1                             'Open communication line and log on
    Wait M_NvOpen(1)=1                          'Wait for line connection
    NVLoad #1,CPRG$                               'Load the vision program
Return
'
'##### VS initialization processing #####
*S20VSINI
'Move from the robot coordinate axis (P_ZERO position) to the robot origin when the vision sensor recognizes workpieces
    MED1#=M_100#(MENCNO)                           'Amount of conveyer movement at calibration between vision sensor and robot
    PRBORG=P_EncDlt(MENCNO)*MED1#                  'Robot origin when the vision sensor recognizes workpieces
'Return a workpiece recognized by the vision sensor to the position taught to the robot
    MED2#=M_101#(MWKNO)                              'Amount of conveyer movement from vision sensor recognition to workpiece teaching
    PBACK=P_EncDlt(MENCNO)*MED2#
'Calculate the position of the workpiece that the vision sensor in the robot area recognized.
    PWKPOS=PRBORG+PVSWRK+PBACK                     'Workpiece position recognized by the vision
```

```
sensor into the robot area
53    PVTR=(P_Zero/PWKPOS)*PTEACH                'Vectors specifying the center of gravity of the vision sensor and grabbing position
54    If PVTR.X<-PCHK.X Or PVTR.X>PCHK.X Then Error 9110   'The calculation result is greatly different from the theory value.
55    If PVTR.Y<-PCHK.Y Or PVTR.Y>PCHK.Y Then Error 9110
56 Return
57 '
58 '##### Condition setting #####
59 *S30CONST
60    MDX = P_EncDlt(MENCNO).X                   'Amount of movement per pulse (X)
61    MDY = P_EncDlt(MENCNO).Y                   'Amount of movement per pulse (Y)
62    MDZ = P_EncDlt(MENCNO).Z                   'Amount of movement per pulse (Z)
63    MD = Sqr(MDX^2+MDY^2+MDZ^2)                'Calculation of the amount of movement per pulse
64    MEI#=Abs((MVSL-MWKL)/MD)                   'Calculation of imaging start setting value
65 Return
66 '
67 '##### VS recognition check processing #####
68 *S40CHKS
69 *LVSCMD
70 *LWAIT
71    MEC# = M_Enc(MENCNO)
72    MEM#=MEC#-MEP#                             'Subtract the previous encoder pulse value from the current position of the encoder
73    If MEM# > 800000000.0# Then MEM# = MEM#-1000000000.0#
74    If MEM# < -800000000.0# Then MEM# = MEM#+1000000000.0#
75    If Abs(MEM#) > MEI# GoTo *LVSTRG           'Comparison between the amount of encoder movement and the camera startup setting value
76    Dly 0.01
77    GoTo *LWAIT
78 *LVSTRG
79    MEP#=MEC#                                  'Set the encoder pulse current position to the previous value
80    NVTrg #1, 5, MTR1#,MTR2#,MTR3#,MTR4#,MTR5#,MTR6#,MTR7#,MTR8#   'Imaging request + encoder value acquisition
81 'Acquisition of recognition data
82    If M_NvOpen(1)<>1   Then Error 9100        'Communication error
83    EBRead #1,"",MNUM,PVS(1),PVS(2),PVS(3),PVS(4)   'Imaging request
84    If MNUM=0 Then GoTo *LVSCMD                'If no workpieces are recognized
85    If MNUM>4 Then MNUM=4                      'Set the maximum number (4)
86    For M1=1 To MNUM                           'Repeat for the number of workpieces recognized
87      MX(M1)=PVS(M1).X                         'Data acquisition
88      MY(M1)=PVS(M1).Y
89      MT(M1)=PVS(M1).C
90    Next M1
91    GoSub *S60WRDAT                            'Tracking data storage processing
92 Return
93 '
94 '##### Tracking data storage processing #####
95 *S60WRDAT
96    For M1=1 To MNUM                           'Perform processing for the number of workpieces recognized
97      PSW=P_Zero
98      PSW=PRBORG                               'Virtually move the robot close to the vision sensor
99      PSW.X=PSW.X+MX(M1)                       'Create the grabbing position
100     PSW.Y=PSW.Y+MY(M1)
101     PSW.C=PSW.C+MT(M1)
102     PRW=P_Zero
103     PRW=PSW*PVTR                             'Compensate for the error in the calculation value
104     PRW.FL1=P_100(MWKNO).FL1
105     PRW.FL2=P_100(MWKNO).FL2
106     Select MENCNO
107     Case 1
108       TrWrt PRW, MTR1#, MWKNO,1,MENCNO   'Position, encoder value, model number, buffer number, encoder number
```

```
109          Break
110        Case 2
111          TrWrt PRW, MTR2#, MWKNO,1,MENCNO   'Position, encoder value, model number, buffer number, encoder number
112          Break
113        Case 3
114          TrWrt PRW, MTR3#, MWKNO,1,MENCNO   'Position, encoder value, model number, buffer number, encoder number
115          Break
116        Case 4
117          TrWrt PRW, MTR4#, MWKNO,1,MENCNO   'Position, encoder value, model number, buffer number, encoder number
118          Break
119        Case 5
120          TrWrt PRW, MTR5#, MWKNO,1,MENCNO   'Position, encoder value, model number, buffer number, encoder number
121          Break
122        Case 6
123          TrWrt PRW, MTR6#, MWKNO,1,MENCNO   'Position, encoder value, model number, buffer number, encoder number
124          Break
125        Case 7
126          TrWrt PRW, MTR7#, MWKNO,1,MENCNO   'Position, encoder value, model number, buffer number, encoder number
127          Break
128        Case 8
129          TrWrt PRW, MTR8#, MWKNO,1,MENCNO   'Position, encoder value, model number, buffer number, encoder number
130          Break
131        End Select
132      Next M1
133      Return
PVS(1)=(+0.00,+0.00,+0.00,+0.00,+0.00,+0.00,+0.00,+0.00)(0,0)
PVS(2)=(+0.00,+0.00,+0.00,+0.00,+0.00,+0.00,+0.00,+0.00)(0,0)
PVS(3)=(+0.00,+0.00,+0.00,+0.00,+0.00,+0.00,+0.00,+0.00)(0,0)
PVS(4)=(+0.00,+0.00,+0.00,+0.00,+0.00,+0.00,+0.00,+0.00)(0,0)
PTEACH=(+0.00,+0.00,+0.00,+0.00,+0.00,+0.00,+0.00,+0.00)(0,0)
PVSWRK=(+0.00,+0.00,+0.00,+0.00,+0.00,+0.00,+0.00,+0.00)(0,0)
PRBORG=(+0.00,+0.00,+0.00,+0.00,+0.00,+0.00,+0.00,+0.00)(0,0)
PBACK=(+0.00,+0.00,+0.00,+0.00,+0.00,+0.00,+0.00,+0.00)(0,0)
PWKPOS=(+0.00,+0.00,+0.00,+0.00,+0.00,+0.00,+0.00,+0.00)(0,0)
PVTR=(+0.00,+0.00,+0.00,+0.00,+0.00,+0.00,+0.00,+0.00)(0,0)
PCHK=(+100.00,+100.00,+0.00,+0.00,+0.00,+0.00,+0.00,+0.00)(0,0)
PSW=(+0.00,+0.00,+0.00,+0.00,+0.00,+0.00,+0.00,+0.00)(0,0)
PRW=(+0.00,+0.00,+0.00,+0.00,+0.00,+0.00,+0.00,+0.00)(0,0)
```

### 21.5.3. For RH-3S*HR

#### (1) 1.Prg

```
'### Ver.A3 ##################################
'# Conveyer tracking, robot operation program(for RH-3SDHR)
'# Program type : 1.prg
'# Date of creation/version : 2012.07.31 A3
'# MITSUBISHI ELECTRIC CORPORATION.
'###########################################
'
'### Main processing ###
*S00MAIN
    GoSub *S90HOME                             'Origin return processing
    GoSub *S10INIT                             'Initialization processing
*LOOP
    GoSub *S20TRGET                            'Tracked workpiece takeout processing
    GoSub *S30WKPUT                            'Workpiece placing processing
    GoTo *LOOP
End
'
'###   Initialization processing ###
*S10INIT
'/// Speed related ///
    Accel 100,100                              'Acceleration/deceleration setting
    Ovrd 100                                   'Speed setting
    Loadset 1,1                                'Optimal acceleration/deceleration specification
    OAdl On                                    'Turning optimal acceleration/deceleration ON
    Cnt 0
    Clr 1
    HOpen 1
'/// Initial value setting ///
    TrClr 1                                    'Clear tracking buffer 1
    MWAIT1=0                                   'Clear workpiece wait flag 1
'/// The processing to singular point of   RH-3S*HR ///
    MTUPPOS=P3HR.X                  'Move time to midair position(measurement time that the slowest J1 axis rotated from -225 to 225 degrees)
    MTWKPOS=1000 * PUP1.Y / P3HR.Y 'Move time to suction position(calculation from speed and move amount of J3)
    MTWKUP=1000 * PUP1.Z / P3HR.Y   'Move time to midair position(calculation from speed and move amount   of J3)
    MTRSTT=MTUPPOS                  'Move time to midair position
    MTREND=MTUPPOS + MTWKPOS + (PDLY1.X * 1000) + MTWKUP 'Necessary time for tracking before it passes over singular point
'/// The processing to singular point of   RH-3S*HR ///
'/// Multitask startup ///
    M_09#=PWK.X                                'Model number specification
    If M_Run(2)=0 Then                         'Confirmation of conveyer 1 multitasking
      XRun 2,"CM1",1                           'Multitasking setting
      Wait M_Run(2)=1
    EndIf
    Priority PRI.X,1
    Priority PRI.Y,2
Return
'
'### Tracked workpiece takeout processing ###
*S20TRGET
'/// Tracking buffer check ///
*LBFCHK
    If M_Trbfct(1)>=1 Then GoTo *LREAD         'If a workpiece exists
    Mov P1                                     'Move to the pull-off location
    MWAIT1=0
```

```
55     GoTo *LBFCHK
56 '/// Workpiece data acquisition ///
57 *LREAD
58     TrRd PBPOS,MBENC#,MBWK%,1,MBENCNO%              'Read data from the tracking buffer
59     GoSub *S40DTSET                                 'Transportation data setting
60 '/// Workpiece position confirmation ///
61 *LNEXT
62     PX50CUR=TrWcur(MBENCNO%,PBPOS,MBENC#)           'Acquire the current workpiece position
63     MX50ST=PRNG.X                                   'Start distance of the range where the robot can follow a workpiece
64     MX50ED=PRNG.Y                                   'End distance of the range where the robot can follow a workpiece
65     MX50PAT=PTN.X                                   'Conveyer position pattern number
66     GoSub *S50WKPOS                                 'Workpiece position confirmation processing
67     If MY50STS=3 Then GoTo *LBFCHK                  'Already passed. Go to the next workpiece
68     If MY50STS=2 Then GoTo *LTRST                   'Operable: start tracking
69     If MWAIT=1 Then GoTo *LNEXT                     'Wait for incoming workpieces
70 '/// To standby position ///
71     PWAIT=P1                                        'Change to workpiece wait posture
72     Select PTN.X                                    'Conveyer position pattern number
73     Case 1 To 2                                     'When the conveyer is the front of the robot
74         PWAIT.X=PX50CUR.X                           'X coordinates of the robot are matched to workpiece.
75     Case 3 To 6
76         PWAIT.Y=PX50CUR.Y                           'Y coordinates of the robot are matched to workpiece.
77     End Select
78     PWAIT.Z=PX50CUR.Z+PUP1.X
79     PWAIT.C=PX50CUR.C
80     Mov PWAIT                                       'Move to workpiece wait posture PWAIT
81     MWAIT1=1                                        'Set workpiece wait flag
82     GoTo *LNEXT
83 '/// Start tracking operation ///
84 *LTRST
85     Accel PAC1.X,PAC1.Y
86     Cnt 1,0,0
87     Act 1=1                                         'Monitor the robot following workpieces too far
88     Trk On,PBPOS,MBENC#,PTBASE,MBENCNO%             'Tracking operation start setting
89     Mov PGT,PUP1.Y Type 0,0                         'Move to tracking midair position
90     Accel PAC2.X,PAC2.Y
91     Mov PGT Type 0,0                                'Move to a suction position
92     GoSub *S85CLOSE                                 'Turn suction ON
93     MX80ENA=PHND.X                                  'Check instruction
94     MX80SIG=PHND.Y                                  'Check signal number
95     MX80SEC=PDLY1.X                                 'Check second number(s)
96     GoSub *S80CWON                                  'adsorbtion confirmation
97     Cnt 1
98     Accel PAC3.X,PAC3.Y
99     Mov PGT,PUP1.Z Type 0,0                         'Move to tracking midair position
100     Trk Off                                        'Tracking operation end setting
101     Act 1=0
102     Accel 100,100
103     MWAIT = 0
104 Return
105 '
106 '### Workpiece placing processing ###
107 *S30WKPUT
108     Accel PAC11.X,PAC11.Y
109     Mov PPT,PUP2.Y                                 'Move to over the placement position
110     Accel PAC12.X,PAC12.Y
111     Cnt 1,0,0
112     Mov PPT Type 0,0                               'Move to the placement position
113     GoSub *S86OPEN                                 'Turn suction OFF
114     MX81ENA=PHND.X                                 'Check instruction
115     MX81SIG=PHND.Z                                 'Check signal number
```

```
    MX81SEC=PDLY2.X                                 'Check second number(s)
    GoSub *S81CWOFF                                 'Release confirmation
    Cnt 1
    Accel PAC13.X,PAC13.Y
    Mov PPT,PUP2.Z Type 0,0                         'Move to over the placement position
    Accel 100,100
Return
'
'### Transportation data setting processing ###
*S40DTSET
    PTBASE=P_100(PWK.X)                             'Create reference position
    TrBase PTBASE,MBENCNO%                          'Tracking base setting
    PGT=PTBASE*POFSET                               'Suction position setting
    GoSub *S46ACSET                                 'Interrupt definition
Return
'
'### Interrupt definition processing 1 ###
*S46ACSET
    Select PTN.X                                    'Conveyer position pattern number
    Case 1 'Front right -> left
      MSTP1=PRNG.Z                                  'Following stop distance
      Def Act 1,P_Fbc(1).Y>MSTP1 GoTo *S91STOP      'To *S91STOP if followed far long
      Break
    Case 2 'Front left -> right
      MSTP1=-PRNG.Z
      Def Act 1,P_Fbc(1).Y<MSTP1 GoTo *S91STOP
      Break
    Case 3 'Left side rear -> front
    Case 5 'Right side rear -> front
      MSTP1=PRNG.Z
      Def Act 1,P_Fbc(1).X>MSTP1 GoTo *S91STOP
      Break
    Case 4 'Left side front -> rear
    Case 6 'Right side front -> rear
      MSTP1=-PRNG.Z
      Def Act 1,P_Fbc(1).X<MSTP1 GoTo *S91STOP
      Break
    End Select
Return
'
'### Workpiece position confirmation processing ###
     'PX50CUR:Current workpiece position
     'MX50ST:Tracking start range
     'MX50ED:Tracking end range
     'MX50PAT:Conveyer position pattern number
     'MY50STS:Result (1: Wait/2: Start tracking/3: Next workpiece)
*S50WKPOS
    MY50STS=0                                       'Clear return value
'/// The processing to singular point of   RH-3S*HR ///
    P50FWCUR=PX50CUR * Inv(P_Tool) 'Position of workpiece in flange
    PTRST=P_Zero
    PTRED=P_Zero
'/// The processing to singular point of   RH-3S*HR ///
    Select MX50PAT                                  'Conveyer pattern
    Case 1 'Front right -> left
      M50STT=-MX50ST                                'The start side has a negative value
      M50END=MX50ED
      If PosCq(PX50CUR)=1 And PX50CUR.Y>=M50STT And PX50CUR.Y<=M50END Then
        MY50STS=2                                   'Tracking possible
'/// The processing to singular point of   RH-3S*HR ///
        PTRST.Y = P_CvSpd(MBENCNO%).Y * MTRSTT / 1000
        PTRST = PTRST + P50FWCUR                    'Position when beginning to follow as for
workpiece.
        PTRED.Y = P_CvSpd(MBENCNO%).Y * MTREND / 1000
        PTRED = PTRED + P50FWCUR                    'Position when having finished following
```

```
as for workpiece.
180         If (PTRST.X > -P3HR.Z And PTRST.X < P3HR.Z) Then    'case the singular point area
181           If (PTRST.Y < -P3HR.Z And PTRED.Y < -P3HR.Z) Then MY50STS=2 'The position of the work peace is OK from the singular point if previous.
182           If (PTRED.Y > -P3HR.Z And PTRED.Y < P3HR.Z) Then MY50STS=3  'If the tracking end position is singular point neighborhood, it is NG.
183           If (PTRST.Y > -P3HR.Z And PTRST.Y < P3HR.Z) Then MY50STS=3  'If the tracking start position is singular point neighborhood, it is NG.
184           If (PTRST.Y > P3HR.Z And PTRED.Y > P3HR.Z) Then MY50STS=3    'It is NG if passing over the singular point.
185         EndIf
186 '/// The processing to singular point of   RH-3S*HR ///
187       Else 'If tracking not possible
188         If PX50CUR.Y<0 Then MY50STS=1                  'Wait
189         If PX50CUR.Y>M50END Then MY50STS=3            'Move onto the next workpiece
190             If PosCq(PX50CUR)=0 And PX50CUR.Y>=M50STT And PX50CUR.Y<=M50END Then MY50STS=3 'Outside the movement range
191       EndIf
192       Break
193     Case 2 'Front left -> right
194       M50STT=MX50ST
195       M50END=-MX50ED                                  'The end side has a negative value
196       If PosCq(PX50CUR)=1 And PX50CUR.Y<=M50STT And PX50CUR.Y>=M50END Then
197         MY50STS=2                                     'Tracking possible
198 '/// The processing to singular point of   RH-3S*HR ///
199         PTRST.Y = P_CvSpd(MBENCNO%).Y * MTRSTT / 1000
200         PTRST = PTRST + P50FWCUR                        'Position when beginning to follow as for workpiece.
201         PTRED.Y = P_CvSpd(MBENCNO%).Y * MTREND / 1000
202         PTRED = PTRED + P50FWCUR                        'Position when having finished following as for workpiece.
203         If (PTRST.X > -P3HR.Z And PTRST.X < P3HR.Z) Then    'case the singular point area
204           If (PTRST.Y > P3HR.Z And PTRED.Y > P3HR.Z) Then MY50STS=2 'The position of the work peace is OK from the singular point if previous.
205           If (PTRED.Y > -P3HR.Z And PTRED.Y < P3HR.Z) Then MY50STS=3  'If the tracking end position is singular point neighborhood, it is NG.
206           If (PTRST.Y > -P3HR.Z And PTRST.Y < P3HR.Z) Then MY50STS=3  'If the tracking start position is singular point neighborhood, it is NG.
207           If (PTRST.Y < -P3HR.Z And PTRED.Y < -P3HR.Z) Then MY50STS=3     'It is NG if passing over the singular point.
208         EndIf
209 '/// The processing to singular point of   RH-3S*HR ///
210       Else 'If tracking not possible
211         If PX50CUR.Y>0 Then MY50STS=1                  'Wait
212         If PX50CUR.Y<0 Then MY50STS=3                  'Move onto the next workpiece
213             If PosCq(PX50CUR)=0 And PX50CUR.Y<=M50STT And PX50CUR.Y>=M50END Then MY50STS=3 'Outside the movement range
214       EndIf
215       Break
216     Case 3 'Left side rear -> front
217     Case 5 'Right side rear -> front
218       M50STT=-MX50ST                                  'The start side has a negative value
219       M50END=MX50ED
220       If PosCq(PX50CUR)=1 And PX50CUR.X>=M50STT And PX50CUR.X<=M50END Then
221         MY50STS=2                                     'Tracking possible
222 '/// The processing to singular point of   RH-3S*HR ///
223         PTRST.X = P_CvSpd(MBENCNO%).X * MTRSTT / 1000
224         PTRST = PTRST + P50FWCUR                        'Position when beginning to follow as for workpiece.
225         PTRED.X = P_CvSpd(MBENCNO%).X * MTREND / 1000
226         PTRED = PTRED + P50FWCUR                        'Position when having finished following as for workpiece.
227         If (PTRST.Y > -P3HR.Z And PTRST.Y < P3HR.Z) Then    'case the singular point area
228           If (PTRST.X < -P3HR.Z And PTRED.X < -P3HR.Z) Then MY50STS=2 'The position of the work peace is OK from the singular point if previous.
```

```
            If (PTRED.X > -P3HR.Z And PTRED.X < P3HR.Z) Then MY50STS=3  'If the tracking end position is singular point neighborhood, it is NG.
            If (PTRST.X > -P3HR.Z And PTRST.X < P3HR.Z) Then MY50STS=3  'If the tracking start position is singular point neighborhood, it is NG.
            If (PTRST.X > P3HR.Z And PTRED.X > P3HR.Z) Then MY50STS=3    'It is NG if passing over the singular point.
        EndIf
'/// The processing to singular point of   RH-3S*HR ///
    Else 'If tracking not possible
        If PX50CUR.X<0 Then MY50STS=1                'Wait
        If PX50CUR.X>0 Then MY50STS=3                'Move onto the next workpiece
        If PosCq(PX50CUR)=0 And PX50CUR.X>=M50STT And PX50CUR.X<=M50END Then MY50STS=3 'Outside the movement range
    EndIf
    Break
  Case 4 'Left side front -> rear
  Case 6 'Right side front -> rear
    M50STT=MX50ST
    M50END=-MX50ED                                   'The end side has a negative value
    If PosCq(PX50CUR)=1 And PX50CUR.X<=M50STT And PX50CUR.X>=M50END Then
        MY50STS=2                                    'Tracking possible
'/// The processing to singular point of   RH-3S*HR ///
        PTRST.X = P_CvSpd(MBENCNO%).X * MTRSTT / 1000
        PTRST = PTRST + P50FWCUR                     'Position when beginning to follow as for workpiece.
        PTRED.X = P_CvSpd(MBENCNO%).X * MTREND / 1000
        PTRED = PTRED + P50FWCUR                     'Position when having finished following as for workpiece.
        If (PTRST.Y > -P3HR.Z And PTRST.Y < P3HR.Z) Then   'case the singular point area
            If (PTRST.X > P3HR.Z And PTRED.X > P3HR.Z) Then MY50STS=2 'The position of the work peace is OK from the singular point if previous.
            If (PTRED.X > -P3HR.Z And PTRED.X < P3HR.Z) Then MY50STS=3  'If the tracking end position is singular point neighborhood, it is NG.
            If (PTRST.X > -P3HR.Z And PTRST.X < P3HR.Z) Then MY50STS=3  'If the tracking start position is singular point neighborhood, it is NG.
            If (PTRST.X < -P3HR.Z And PTRED.X < -P3HR.Z) Then MY50STS=3    'It is NG if passing over the singular point.
        EndIf
'/// The processing to singular point of   RH-3S*HR ///
    Else 'If tracking not possible
        If PX50CUR.X>0 Then MY50STS=1                'Wait
        If PX50CUR.X<0 Then MY50STS=3                'Move onto the next workpiece
        If PosCq(PX50CUR)=0 And PX50CUR.X<=M50STT And PX50CUR.X>=M50END Then MY50STS=3 'Outside the movement range
    EndIf
    Break
  End Select
  P50TRST=PTRST '/// The processing to singular point of   RH-3S*HR ///
  P50TRED=PTRED '/// The processing to singular point of   RH-3S*HR ///
  If MY50STS=0 Then Error 9199                       'Program modification required
Return
'
'### Origin return processing ###
*S90HOME
  Servo On                                           'Servo ON
  P90CURR=P_Fbc(1)                                   'Acquire the current position
  If P90CURR.Z<P1.Z Then                             'If the current height is below the origin
    Ovrd 10
    P90ESC=P90CURR                                   'Create an escape position
    P90ESC.Z=P1.Z
    Mvs P90ESC                                       'Move to the escape position
    Ovrd 100
  EndIf
  Mov P1                                             'Move to the origin
Return
```

```
283 '
284 '### Tracking interruption processing ###
285 *S91STOP
286     Act 1=0
287     Trk Off
288     GoSub *S86OPEN                                      'Release suction
289     P91P=P_Fbc(1)                                      'Acquire the current position
290     P91P.Z=P1.Z
291     Mvs P91P Type 0,0                                  'Raise
292     Mov P1                                             'Return to the origin once
293     GoTo *LBFCHK
294 '
295 '##### Suction of substrates #####
296 *S85CLOSE
297     HClose 1                                           'Turn suction ON
298 Return
299 '##### Suction/release of substrates #####
300 *S86OPEN
301     HOpen 1                                             'Turn suction OFF
302 Return
303 '
304 '##### Turning on the signal is waited for #####
305     'MX80ENA:ENABLE/DISABLE of check(1/0)
306     'MX80SIG:Check signal number
307     'MX80SEC:Check second number(S)
308     'MY80SKP:OK/TIMEOUT(1/0)
309 *S80CWON
310     If MX80ENA=1 Then                                  'If the signal check is ENABLE
311        M_Timer(1)=0
312        MY80SKP=0
313        MX80SEC=MX80SEC * 1000                              'Second -> Millisecond
314 *L80LOP
315        If (M_Timer(1)>MX80SEC) Or (MY80SKP<>0) Then *L80END
316           If M_In(MX80SIG)=1 Then MY80SKP=1             'If the signal specified is turned on
317        GoTo *L80LOP
318     Else                                               'If the signal check is DISABLE
319        Dly MX80SEC                                        'Wait at the specified check time
320        MY80SKP=1                                          'OK
321     EndIf
322 *L80END
323 Return
324 '
325 '##### Turning off the signal is waited for #####
326     'MX81ENA:ENABLE/DISABLE of check(1/0)
327     'MX81SIG:Check signal number
328     'MX81SEC:Check second number(S)
329     'MY81SKP:OK/TIMEOUT(1/0)
330 *S81CWOFF
331     If MX81ENA=1 Then                                  'If the signal check is ENABLE
332        M_Timer(1)=0
333        MY81SKP=0
334        MX81SEC=MX81SEC * 1000                              'Second -> Millisecond
335 *L81LOP
336        If (M_Timer(1)>MX81SEC) Or (MY81SKP<>0) Then *L81END
337           If M_In(MX81SIG)=0 Then MY81SKP=1             'If the signal specified is turned off
338        GoTo *L81LOP
339     Else                                               'If the signal check is DISABLE
340        Dly MX80SEC                                        'Wait at the specified check time
341        MY81SKP=1 'OK
342     EndIf
343 *L81END
344 Return
P3HR=(+800.000,+1500.000,+60.000,+0.000,+0.000,+0.000,+0.000,+0.000)(0,0)
PUP1=(+50.000,+0.000,+0.000,+0.000,+0.000,+0.000,+0.000,+0.000)(0,0)
PDLY1=(+1.000,+0.000,+0.000,+0.000,+0.000,+0.000,+0.000,+0.000)(0,0)
```

```
PWK=(+1.000,+0.000,+0.000,+0.000,+0.000,+0.000,+0.000,+0.000)(0,0)
PRI=(+1.000,+1.000,+0.000,+0.000,+0.000,+0.000,+0.000,+0.000)(0,0)
P1=(+0.000,+0.000,+0.000,+0.000,+0.000,+0.000,+0.000,+0.000)(0,0)
PBPOS=(+0.000,+0.000,+0.000,+0.000,+0.000,+0.000,+0.000,+0.000)(0,0)
PX50CUR=(+0.000,+0.000,+0.000,+0.000,+0.000,+0.000,+0.000,+0.000)(0,0)
PRNG=(+300.000,+200.000,+400.000,+0.000,+0.000,+0.000,+0.000,+0.000)(0,0)
PTN=(+1.000,+0.000,+0.000,+0.000,+0.000,+0.000,+0.000,+0.000)(0,0)
PWAIT=(+0.000,+0.000,+0.000,+0.000,+0.000,+0.000,+0.000,+0.000)(0,0)
PAC1=(+100.000,+100.000,+0.000,+0.000,+0.000,+0.000,+0.000,+0.000)(0,0)
PTBASE=(+0.000,+0.000,+0.000,+0.000,+0.000,+0.000,+0.000,+0.000)(0,0)
PGT=(+0.000,+0.000,+0.000,+0.000,+0.000,+0.000,+0.000,+0.000)(0,0)
PAC2=(+100.000,+100.000,+0.000,+0.000,+0.000,+0.000,+0.000,+0.000)(0,0)
PHND=(+0.000,+900.000,+900.000,+0.000,+0.000,+0.000,+0.000,+0.000)(0,0)
PAC3=(+100.000,+100.000,+0.000,+0.000,+0.000,+0.000,+0.000,+0.000)(0,0)
PAC11=(+100.000,+100.000,+0.000,+0.000,+0.000,+0.000,+0.000,+0.000)(0,0)
PPT=(+0.000,+0.000,+0.000,+0.000,+0.000,+0.000,+0.000,+0.000)(0,0)
PUP2=(+0.000,+0.000,+0.000,+0.000,+0.000,+0.000,+0.000,+0.000)(0,0)
PAC12=(+100.000,+100.000,+0.000,+0.000,+0.000,+0.000,+0.000,+0.000)(0,0)
PDLY2=(+1.000,+0.000,+0.000,+0.000,+0.000,+0.000,+0.000,+0.000)(0,0)
PAC13=(+100.000,+100.000,+0.000,+0.000,+0.000,+0.000,+0.000,+0.000)(0,0)
POFSET=(+0.000,+0.000,+0.000,+0.000,+0.000,+0.000,+0.000,+0.000)(0,0)
P50FWCUR=(+0.000,+0.000,+0.000,+0.000,+0.000,+0.000,+0.000,+0.000)(0,0)
PTRST=(+0.000,+0.000,+0.000,+0.000,+0.000,+0.000,+0.000,+0.000)(0,0)
PTRED=(+0.000,+0.000,+0.000,+0.000,+0.000,+0.000,+0.000,+0.000)(0,0)
P50TRST=(0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000)(0,0)
P50TRED=(0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000)(0,0)
P90CURR=(+0.000,+0.000,+0.000,+0.000,+0.000,+0.000,+0.000,+0.000)(0,0)
P90ESC=(+0.000,+0.000,+0.000,+0.000,+0.000,+0.000,+0.000,+0.000)(0,0)
P91P=(+0.000,+0.000,+0.000,+0.000,+0.000,+0.000,+0.000,+0.000)(0,0)
```

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

Authorised representative:
MITSUBISHI ELECTRIC EUROPE B.V. GERMANY
Gothaer Str. 8, 40880 Ratingen / P.O. Box 1548, 40835 Ratingen, Germany