MITSUBISHI

Mitsubishi Industrial Robot

CRn-500 Series INSTRUCTION MANUAL
Expansion Serial Interface
Tracking Function Manual

MELFA

BFP-A8524-A

⚠ Safety Precautions

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

⚠CAUTION

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

⚠CAUTION

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

⚠WARNING

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

⚠CAUTION

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

⚠WARNING

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

⚠CAUTION

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

⚠CAUTION

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

⚠CAUTION

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

⚠CAUTION

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

⚠CAUTION

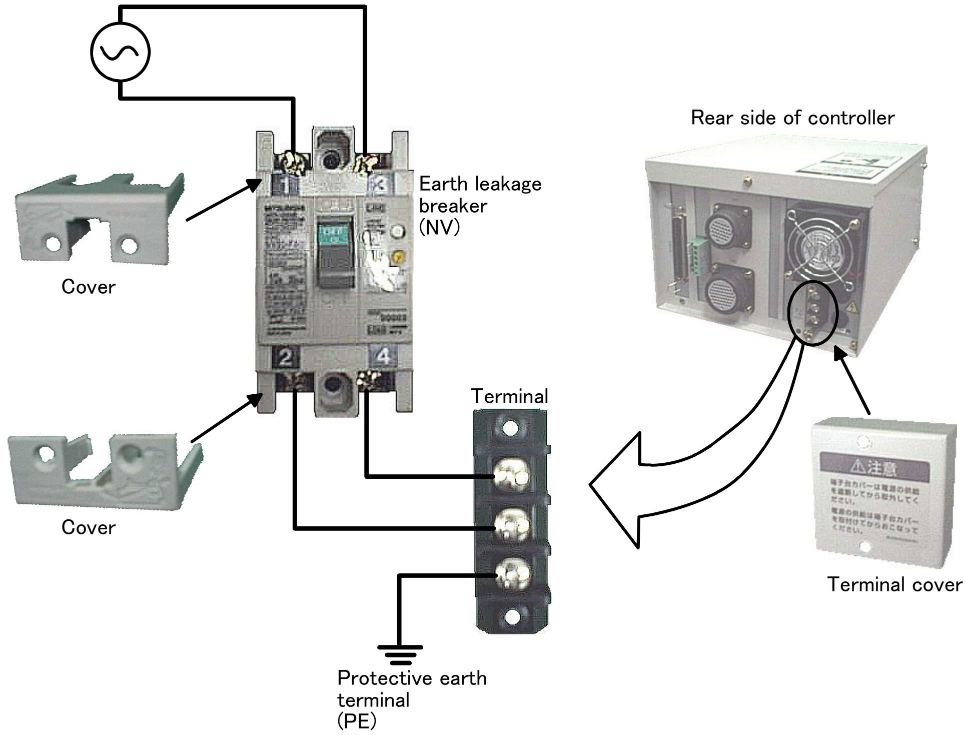
Do not turn off the main power to the robot controller while rewriting the internal information of the robot controller such as the program or parameters.
If the main power to the robot controller is turned off while in automatic operation or rewriting the program or parameters, the internal information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

Precautions for the basic configuration are shown below.(When CR1-571/CR1B-571 is used for the controller.)

⚠CAUTION

Provide an earth leakage breaker that packed together on the primary power supply of the controller as protection against electric leakage. Confirm the setting connector of the input power supply voltage of the controller, if the type which more than one power supply voltage can be used. Then connect the power supply.
Failure to do so could lead to electric shock accidents.

Power supply *RV-1A/2AJ series and RP-1AH/3AH/5AH series: Single phase 90-132VAC, 180-253VAC.
*Except the above: Single phase 180-253VAC.



⚠WARNING

For using RH-5AH/10AH/15AH series or RH-6SH/12SH/18SH series.
While pressing the brake releasing switch on the robot arm, beware of the arm which may drop with its own weight.
Dropping of the hand could lead to a collision with the peripheral equipment or catch the hands or fingers.

■ Revision history

Date of print	Specifications No.	Details of revisions
2007-01-19	BFP-A8524-*	First print.
2009-09-30	BFP-A8524-A	The EC Declaration of Conformity was changed. (Correspond to the EMC directive; 2006/42/EC)

■Preface

Thank you very much for purchasing Mitsubishi Electric Industrial Robot CRn-500 series.
The expansion serial interface is an option to add the tracking function to the robot controller in combination with a CRn-500 series controller. 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.

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

Robot controller: Version K8 or later

[Contents]

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 photoelectronic 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 IV.
(5) System construction is made easy by use of sample programs.

However, there is a restriction matter to use the tracking function. Please refer to "3.1 Tracking Specifications" for the restriction matter.

1.2. Applications

Tracking is primarily intended for applications such as the following.

(1) Transfer of processed food pallets

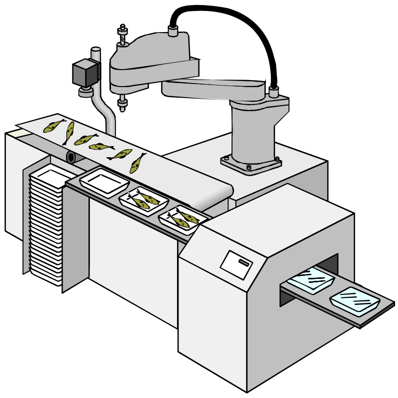


Figure 1–1 Example of Processed Food Pallet Transfer

(2) Lining up parts

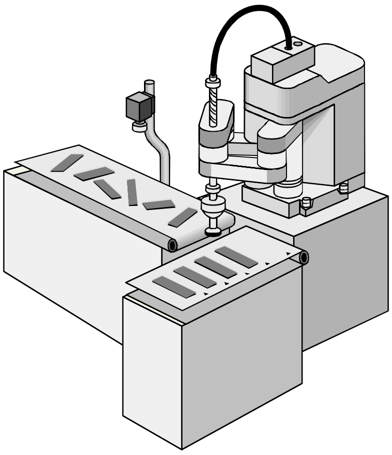


Figure 1–2 Example of Parts Lineup

(3) Assembly of small electrical products

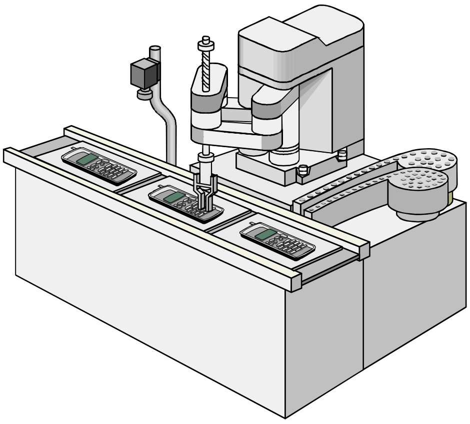


Figure 1–3 Example of Small Electrical Products Assembly

2. System Configuration

2.1. Components

2.1.1. Configuration of the Expansion serial interface and Included Items

“Table 2–1 List of Configuration in the Expansion Serial Interface (2A-RZ581)” lists the configuration of the Expansion serial interface you have purchased.

Table 2–1 List of Configuration in the Expansion Serial Interface (2A-RZ581)

Product name	Model name	Remark
Expansion serial interface card	2A-RZ581	
Tracking function Manual	BFP-A8524	This manual
Expansion serial interface instruction manual	BFP-A8106	Separate volume
Ferrite core	ZCAT3035 or equivalent	For encoder cable
Sample program	–	Please refer to "7 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	R28TB	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		Provide as necessary.
Expansion option box	CR1-EB3		
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(Recommended product): Omron encoder (E6B2-CWZ1X-1000 or -2000) Recommended connector for encoder input terminal: 10120-30000VE plug made by 3M 10320-52F0-008 shell made by 3M
Photoelectronic sensor	–		Used to synchronize tracking
5-V power supply	–		+5 VDC (±10%) : For the encorder
24-V power supply	–		+24 VDC (±10%) : For the Photoelectronic sensor
Personal computer part			
Personal computer	–	1	Please refer to the instruction manual of RT ToolBox for the details of the personal computer specifications.
RT ToolBox (Personal computer support software)	3A-01C-WINJ or 3A-02C-WINJ		

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	R28TB	1	
Hand	–		
Ethernet interface card	2A-HR533		
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		Provide as necessary.
Expansion option box	CR1-EB3		
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(Recommended product): Omron encoder (E6B2-CWZ1X-1000 or -2000) Recommended connector for encoder input terminal: 10120-30000VE plug made by 3M 10320-52F0-008 shell made by 3M
Photoelectronic sensor	–		Used to synchronize tracking
5-V power supply	–		+5 VDC (±10%) : For the encorder
24-V power supply	–		+24 VDC (±10%) : For the Photoelectronic sensor and Vision sensor
Vision sensor part			
Basic network vision sensor set	4D-2CG5xxxx-PKG	1	See the instruction manual of the network vision sensor for details
Lens	–		C-mount lens
Lighting installation	–	(1)	Provide as necessary.
Connection part			
Hub	–	1	
Ethernet cable (straight)	–	2	Between Robot controller and Hub Between Personal computer and Hub
Personal computer part			
Personal computer	–	1	Please refer to the instruction manual of RT ToolBox or the instruction of the network vision sensor for details of the personal computer specifications.
RT ToolBox (Personal computer support software)	3A-01C-WINJ or 3A-02C-WINJ		Recommended version is versions F3 or later. Versions F2 and earlier do not support MELFA-BASIC IV special commands for vision sensors, so errors occur in syntax checks, so use these versions without syntax checking.

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 photoelectronic sensor and follows the workpieces.

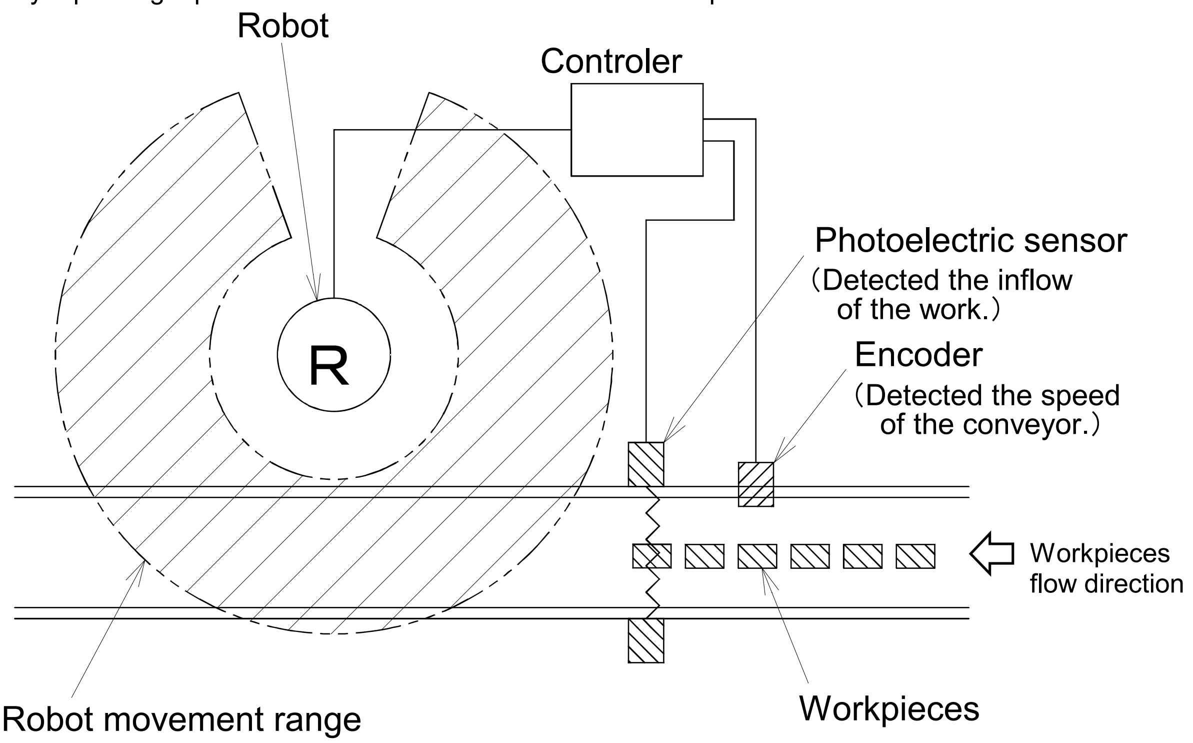

Robot
Controler
Photoelectric sensor
（Detected the inflow of the work.）
Encoder
（Detected the speed of the conveyor.）
R
Workpieces flow direction
Robot movement range
Workpieces

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

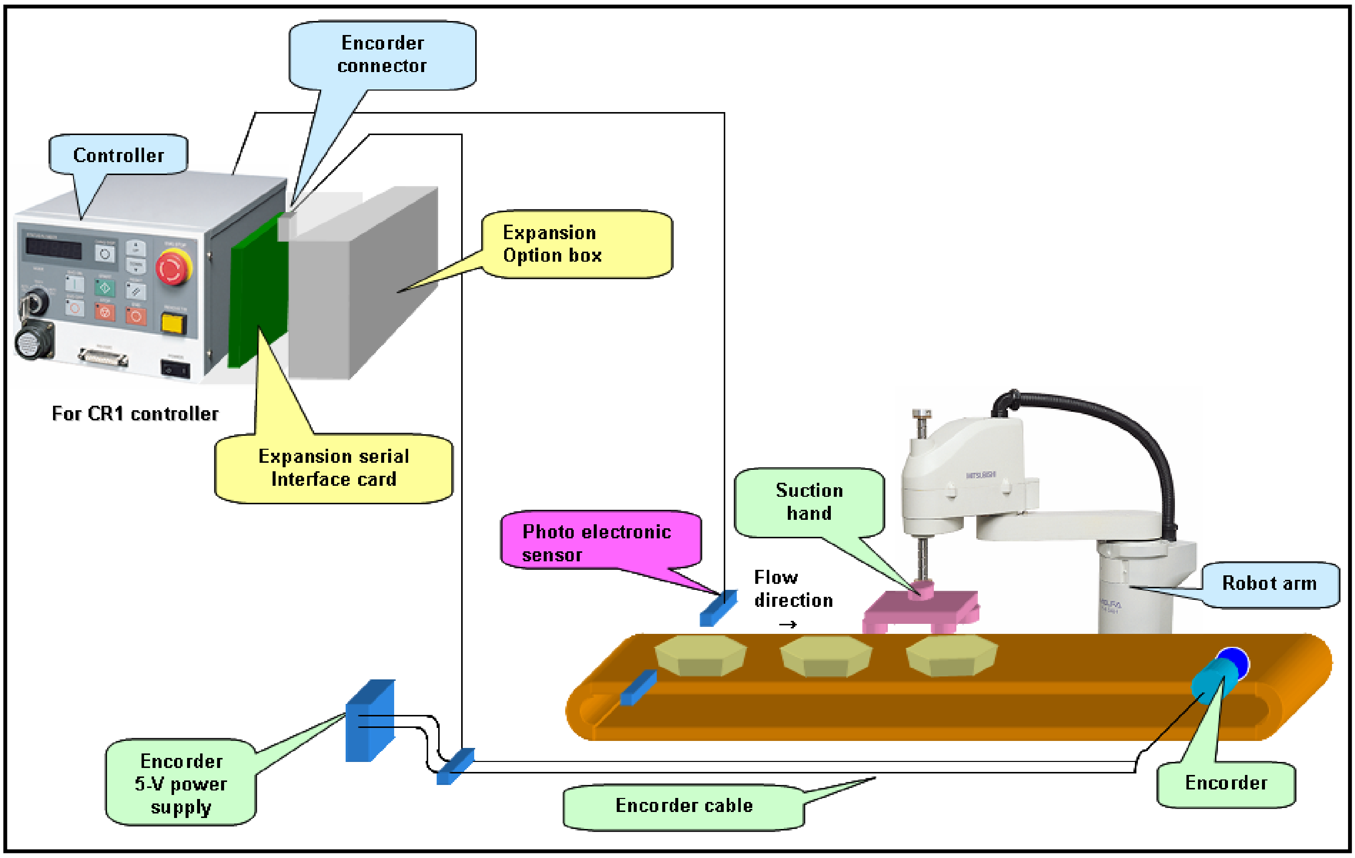


Figure 2–2 Configuration Example of Conveyer Tracking

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.

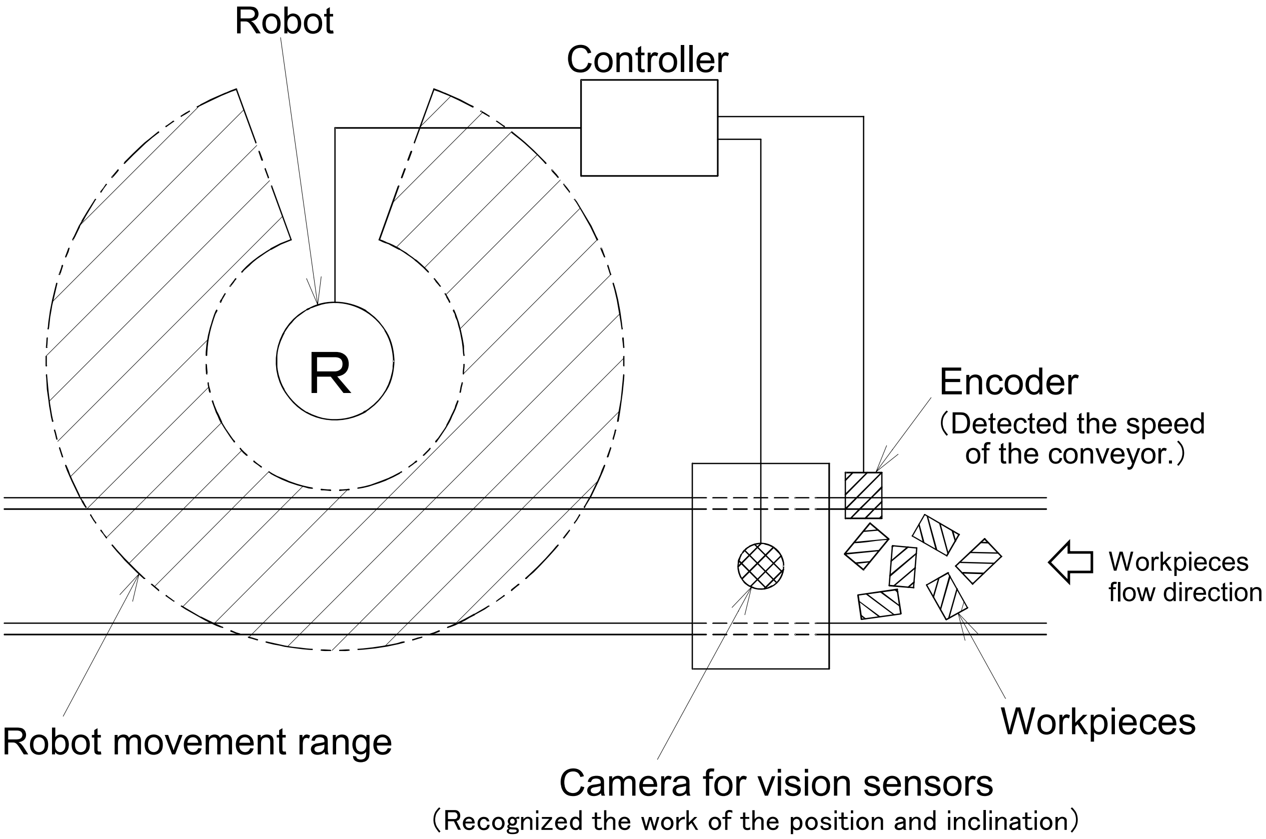


Figure 2–3 Configuration Example of Vision Tracking (Top View)

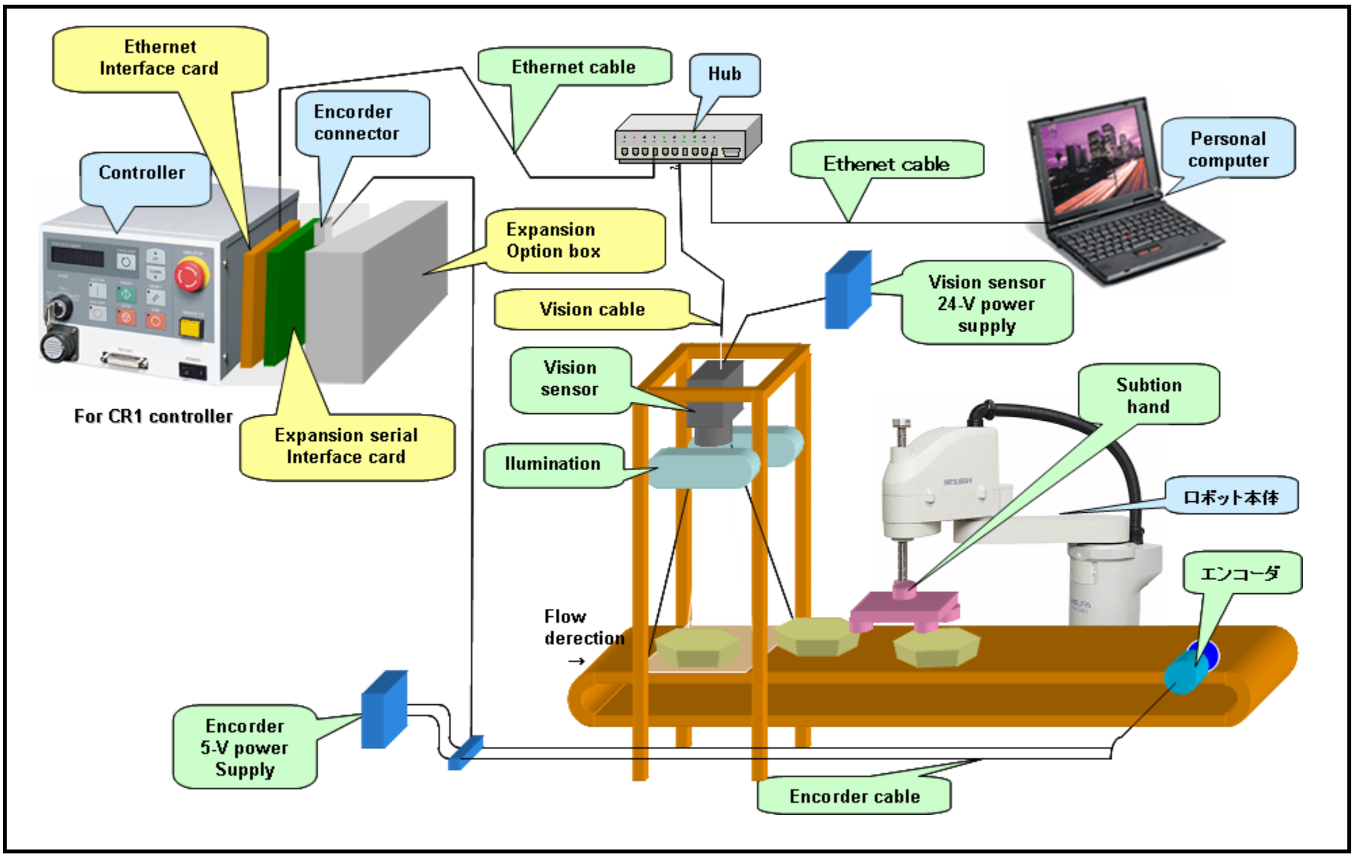


Figure 2–4 Configuration Example of Vision Tracking

3. Specification

3.1. Tracking Specifications and Restriction matter

”Table 3–1 Tracking Function Specifications” and “Table 3–2 Expansion serial interface Card 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 Tracking Function Specifications

Item		Specification and Restriction matter
Supported robots		RP-1AH/3AH/5AH series RV-1A/2AJ RV-3S/3SJ/6S/12S series RH-6SH/12SH/18SH series
Applicable robot controller		CR1/CR1B/CR2A/CR2B/CR3 contoller
Option (*1)		Expansion serial interface card (2A-HR581) Ethernet interface card(2A-HR533) is required in the case of vision tracking.
Software		In the case of conveyer tracking : Robot controller Ver. K8 or later In the case of vision tracking : Robot controller Ver. K8 or later
Robot program language		Load commands dedicated for the tracking function
Conveyer	Movement speed (*2)	Possible to support up to 300 mm/s (When the robot always transport the workpieces) Possible to support up to 500 mm/s when the interval of workpiece is wide. Possible to support two conveyers by one Expansion serial interface card.
	Encoder	Output aspect : $A,\overline{A},B,\overline{B},Z,\overline{Z}$ Output form : line driver output (*3) Highest response frequency : 100 kHz Resolution(pulse/rotation) : Up to 2000 (4000 and 8000 uncorrespond) Recommended encoder : Omron E6B2-CWZ1X-1000 E6B2-CWZ1X-2000
	Encoder cable	Shielded twisted-pair cable Conductor size: 24AWG (0.2 mm$^{2)}$ Cable length: Up to 25 m
Photoelectronic sensor (*4)		Used to detect workpieces positions in conveyer tracking.
Vision sensor (*5)		Mitsubishi’s network vision sensor
Precision at handling position (*6)		Approximately ±2 mm (when the conveyer speed is approximately 300 mm/s) (Photoelectronic sensor recognition error, vision sensor recognition error, robot repeatability error and so on)

（*1） Please refer to “5.2.1Installation of Option Cards (Expansion serial interface/Ethernet Interface)” for information about the expansion slots in which option cards can be mounted. And please refer to “Table 3–4 List of Combinations of Option Cards to be Mounted” for combinations of option cards to be installed. In the case of CR1/CR1B controllers, the expansion option box (CR1-EB3) is required.

（*2） 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.

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

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

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

（*6） 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 error and other factors. The value above should only be used as a guideline.
Please refer to “Figure 3–1 Average number of errors due to conveyer speed” for average of number errors due to conveyer speed.

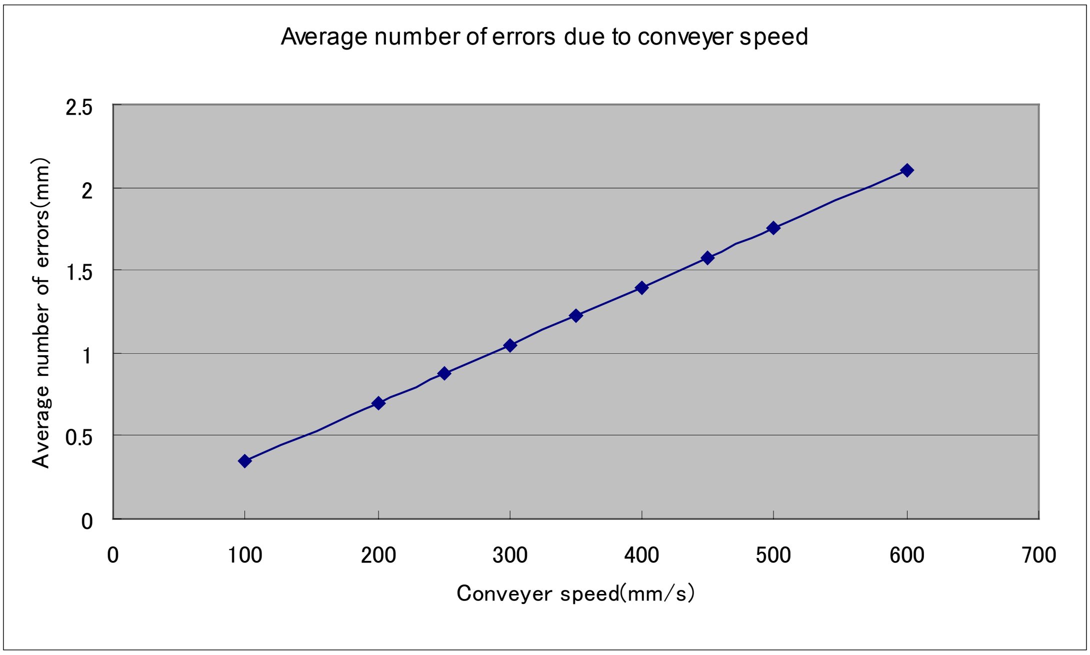


Figure 3–1 Average number of errors due to conveyer speed

Table 3–2 Expansion serial interface Card Specifications

Item	Specification of interface card	
Card name(*1)	RZ581B	RZ581A
Function(*2)	Tracking function Expansion Serial function(*4)	Expansion Serial function
Slot(*3)	Slot1/Slot2/Slot3	Slot1/Slot2
Channel	6 channels or less (2 channnels / card)	4 channels or less (2 channnels / card)
Encoder	Recommended encoder manufacturer: Omron Model: E6B2-CWZ1X 1000 E6B2-CWZ1X 2000	Recommended encoder manufacturer: Omron Model: E6B2-CWZ1X 1000 E6B2-CWZ1X 2000
Encoder cable	Shielded twisted-pair cable: Conductor size : 24AWG (0.2 mm^2) Length of cable: 25m	Shielded twisted-pair cable: Conductor size : 24AWG (0.2 mm^2) Length of cable: 25m

(*1)Please refer to “Figure 3–2 Position in which card name is confirmed" for the method of confirming the card name.

"Card name" has been described at this position of the Expansion serial interface card.
Card name is “RZ581A" or “RZ581B”.

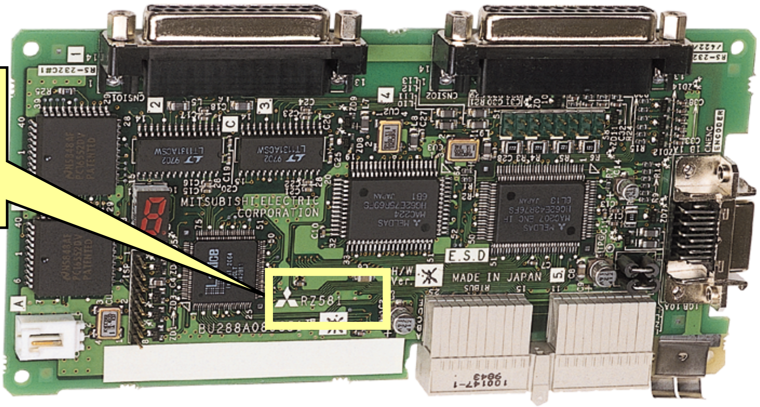


Figure 3–2 Position in which card name is confirmed

(*2)The tracking function and the expansion serial function integrated from version B of interface card (card name : RZ581B). However, robot controller's software version should be since the K8.

(*3)The installation of interface card to expansion slot 3 became possible from version B (card name : RZ581B). However, robot controller's software version should be since the K7.
Please confirm “Table 3–3 Combination of Expansion serial interface Card and robot controller’s software version” about details.

Table 3–3 Combination of Expansion serial interface Card and robot controller’s software version

	Card name	Software’s version	Combination result
1	RZ581A	Version K6 former	Error 7930(EX-SIO card is set in SLOT3) occurs if the interface card is set in slot 3.
2	RZ581B	Version K6 former	Error 7930(EX-SIO card is set in SLOT3) occurs if the interface card is set in slot 3. Because, this card is considered to be RZ581A.
3	RZ581A	Version K7 or later	Error 7930(EX-SIO card is set in SLOT3) occurs if the interface card is set in slot 3. Because, this card is considered to be RZ581A.
4	RZ581B	Version K7 or later	This card can be set in slot 3. Because, this card is considered to be RZ581B.

(*4) Please refer to "CRn-500 series INSTRUCTION MANUAL Expansion Serial Interface (BFP-A8106)" for the Expansion Serial function and the specification.

Table 3–4 List of Combinations of Option Cards to be Mounted

<Sign explanation>	<Card name>	<Installation specification of card>
ETH : Ethernet	HR533	Only one card can be mounted. Be sure to use SLOT1.
SIO(TRK) : Expansion serial	RZ581B	Up to 3 cards can be mounted.
CC : CC-Link	HR575	Only one card can be mounted. Be sure to use SLOT2.
AX : Additional axis	RZ541	Only one card can be mounted. There is no restriction on slot usage.

For CR1/CR1B/CR2A/CR2B

Number of sheets	SLOT1	SLOT2	SLOT3
1	ETH	－	－
	SIO（TRK）	－	－
	－	SIO（TRK）	－
	－	－	SIO（TRK）
	AX	－	－
	－	AX	－
	－	－	AX
	－	CC	－
2	ETH	SIO（TRK）	－
	ETH	－	SIO（TRK）
	ETH	AX	－
	ETH	－	AX
	ETH	CC	－
	SIO（TRK）	SIO（TRK）	－
	SIO（TRK）	－	SIO（TRK）
	SIO（TRK）	AX	－
	SIO（TRK）	－	AX
	SIO（TRK）	CC	－
	AX	SIO（TRK）	－
	AX	－	SIO（TRK）
	AX	CC	－
	－	SIO（TRK）	SIO（TRK）
	－	SIO（TRK）	AX
	－	AX	SIO（TRK）
	－	CC	SIO（TRK）
	－	CC	AX
3	ETH	SIO（TRK）	SIO（TRK）
	ETH	SIO（TRK）	AX
	ETH	AX	SIO（TRK）
	ETH	CC	SIO（TRK）
	ETH	CC	AX
	SIO（TRK）	SIO（TRK）	SIO（TRK）
	SIO（TRK）	SIO（TRK）	AX
	SIO（TRK）	AX	SIO（TRK）
	SIO（TRK）	CC	SIO（TRK）
	SIO（TRK）	CC	AX
	AX	SIO（TRK）	SIO（TRK）
	AX	CC	SIO（TRK）

For CR4/CR7/CR8 or more

Number of sheets	SLOT1	SLOT2
1	ETH	－
	SIO（TRK）	－
	－	SIO（TRK）
	AX	－
	－	AX
	－	CC
2	ETH	SIO（TRK）
	ETH	AX
	ETH	CC
	SIO（TRK）	SIO（TRK）
	SIO（TRK）	AX
	SIO（TRK）	CC
	AX	SIO（TRK）
	AX	CC

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.

1. Start of operation

⬇

2. Preparation and Connection of Equipment ··························· Refer to “Chapter 5.”
Chapter 5 explains installation of option cards and connection of an 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 7.”
Chapter 7 explains functions related to supplemental sample programs.

⬇

5. Calibration of Conveyer and Robot Coordinate Systems (“A” program) ········· Refer to “Chapter 8.”
Chapter 8 explains how to calculate the amount of robot movement per encoder pulse.

⬇

6. Calibration of Vision Coordinate and Robot Coordinate Systems (“B” program) ····· Refer to “Chapter 9.”
Chapter 9 explains how to display the position of a workpiece recognized by the vision sensor in the robot coordinate system.

⬇

7. Workpiece Recognition and Teaching (“C” program) ··························· Refer to “Chapter 10.”
Chapter 10 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 11.”
Chapter 11 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 13.”
In automatic operation, the robot operates via commands from the conveyer control.

⬇

End of operation

10. Maintenance ··························· Refer to “Chapter 14.”

11. Troubleshooting ··························· Refer to “Chapter 15.”

5. Preparation and Connection of Equipment

This chapter explains preparation and connection of equipment required for a conveyer tracking system and a vision tracking system.

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.

5.2. Connection of Equipment

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

5.2.1. Installation of Option Cards (Expansion serial interface/Ethernet Interface)

This section explains how to install option cards using a Expansion serial interface card as an example. Note that this may vary depending on the controller model used. If the controller model is CR1-571, refer to “Figure 5–1 Installation of Expansion serial interface Card (CR1-571 / CR1-571B Controller).” If the controller model is CR2A-572, refer to “Figure 5–2 Installation of Expansion serial interface Card (CR2-572 / CR2B-574 Controller).” If the controller model is CR3-535M, refer to “Figure 5–3 Installation of Expansion serial interface Card (CR3-535M Controller).”
If you use a network vision sensor, an Ethernet interface card must also be connected. Install other option cards and connect required equipment by referring to the instruction manuals of corresponding equipment.

(1) In the case of CR1-571/CR1-571B controller

A Expansion serial interface card should be mounted in either one of option slots 1 to 3 in the expansion option box.

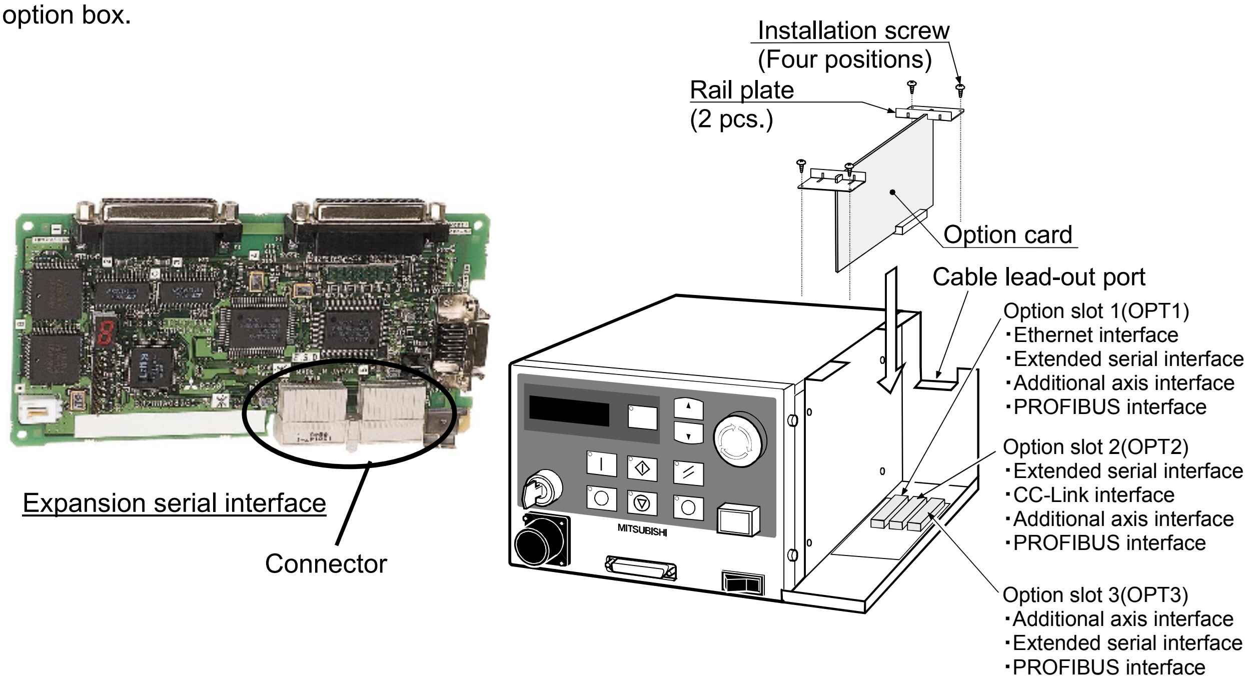


Figure 5–1 Installation of Expansion serial interface Card (CR1-571 / CR1-571B Controller)

(2) In the case of CR2A-572/CR2B-574 controller

Install the Expansion serial interface card in either one of option slots 1 to 3 in the same way as for the CR1-571 controller.

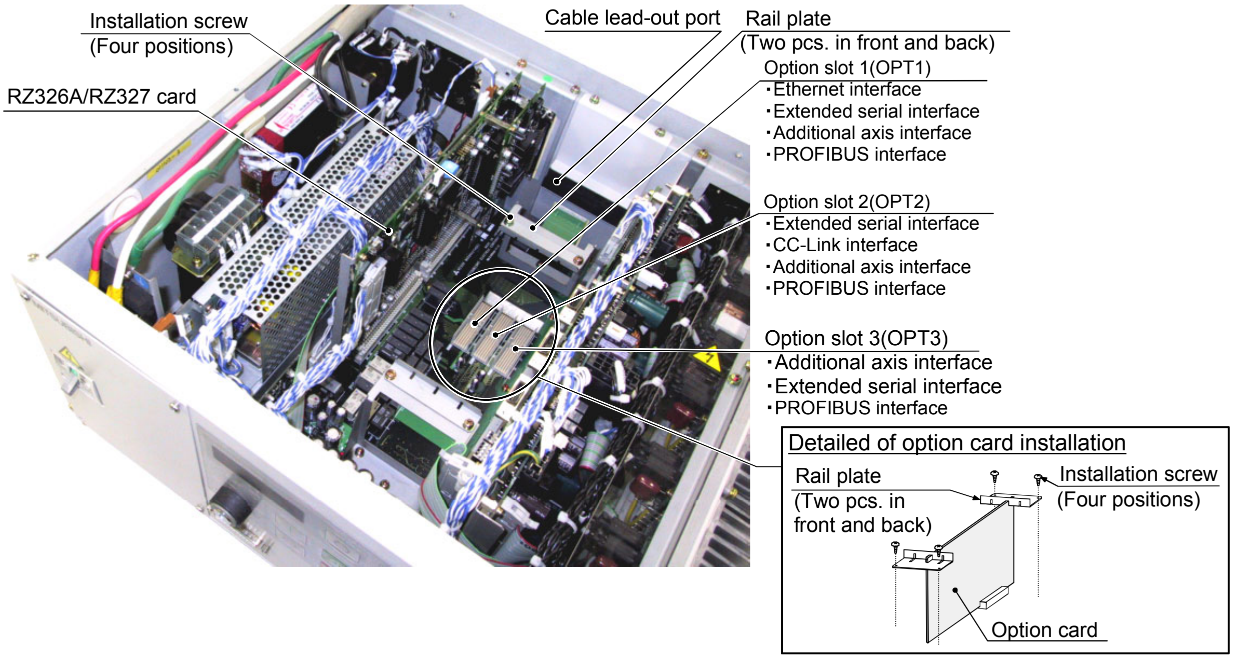


Figure 5–2 Installation of Expansion serial interface Card (CR2-572 / CR2B-574 Controller)

(3) In the case of CR3-535M controller

Install the Expansion serial interface card in either one of option slots 1 to 2 in the same way as for the CR1-571 controller.

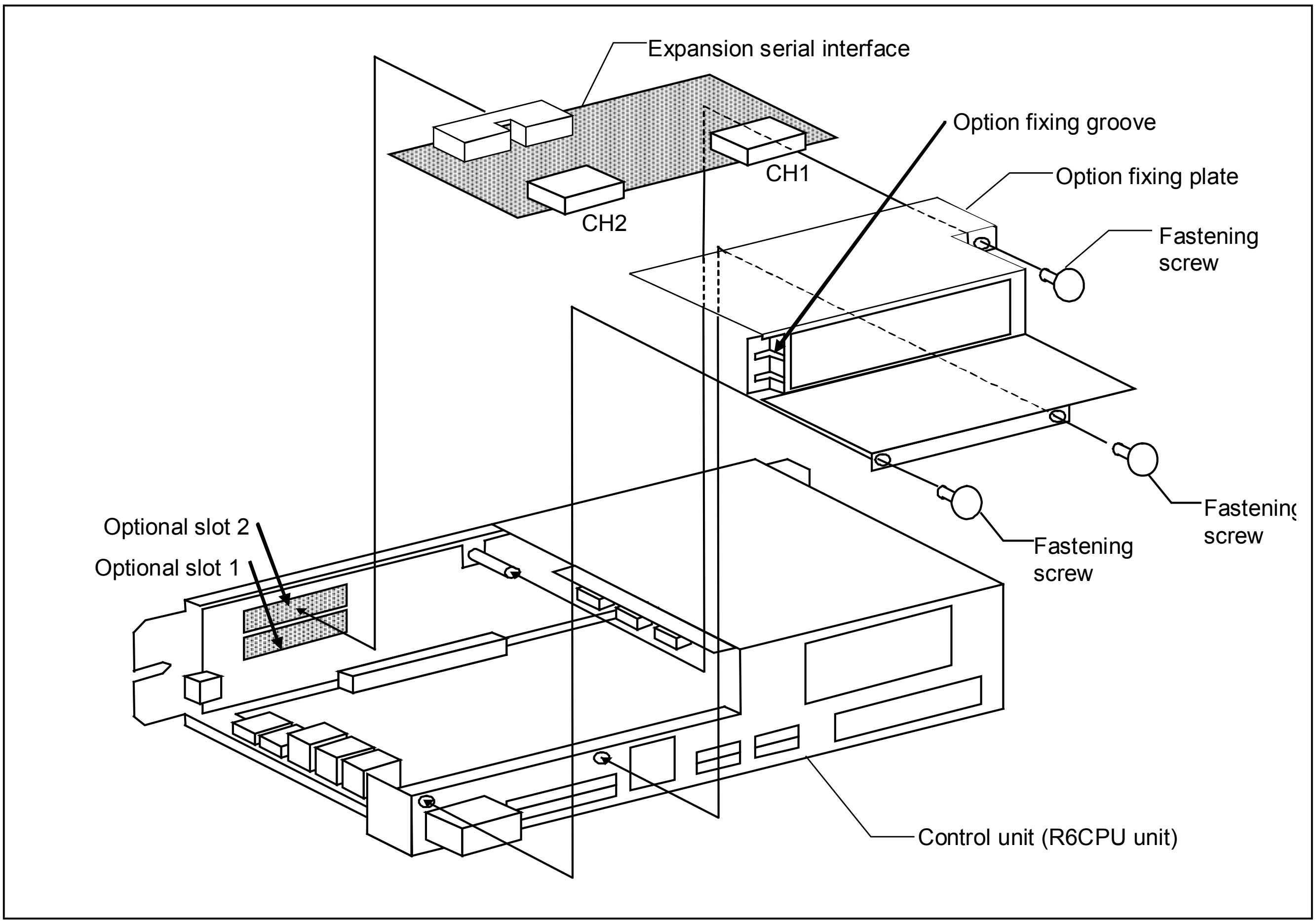


Figure 5–3 Installation of Expansion serial interface Card (CR3-535M Controller)

5.2.2. Connection of Conveyer Encoder

“Figure 5–4 Connector Connection of Expansion serial interface Card” shows the connection between a Expansion serial interface card connector and an encoder.
The figure shows a case where E6B2-CWZ1X (made by Omron) is used for the encoders and two encoders are connected to the Expansion serial interface card.
Up to 2 encoders can be connected to one Expansion serial interface card. 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 close to the expansion serial interface card. If ferrite cores are not mounted, the robot may malfunction due to the influence of noise.

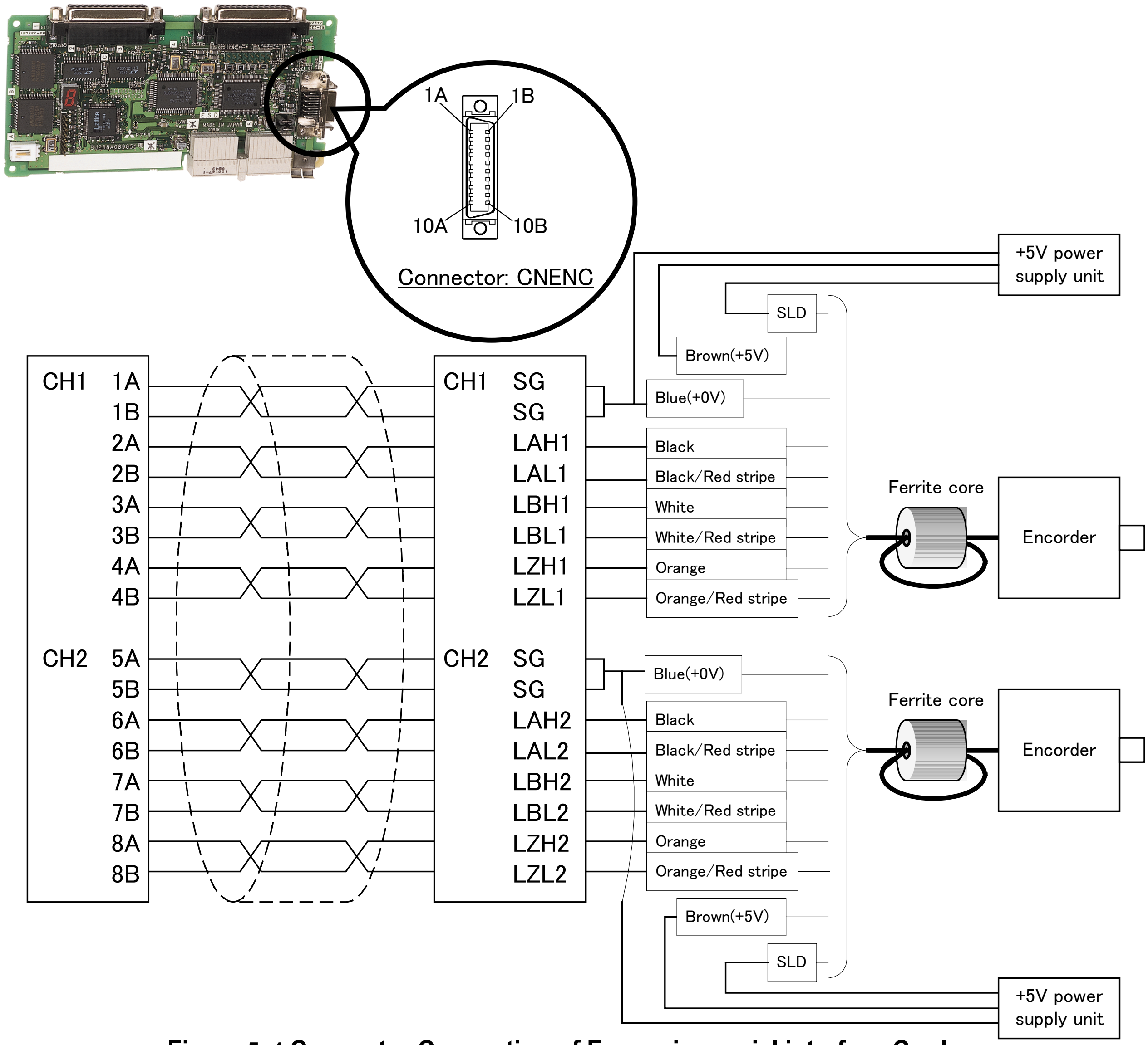


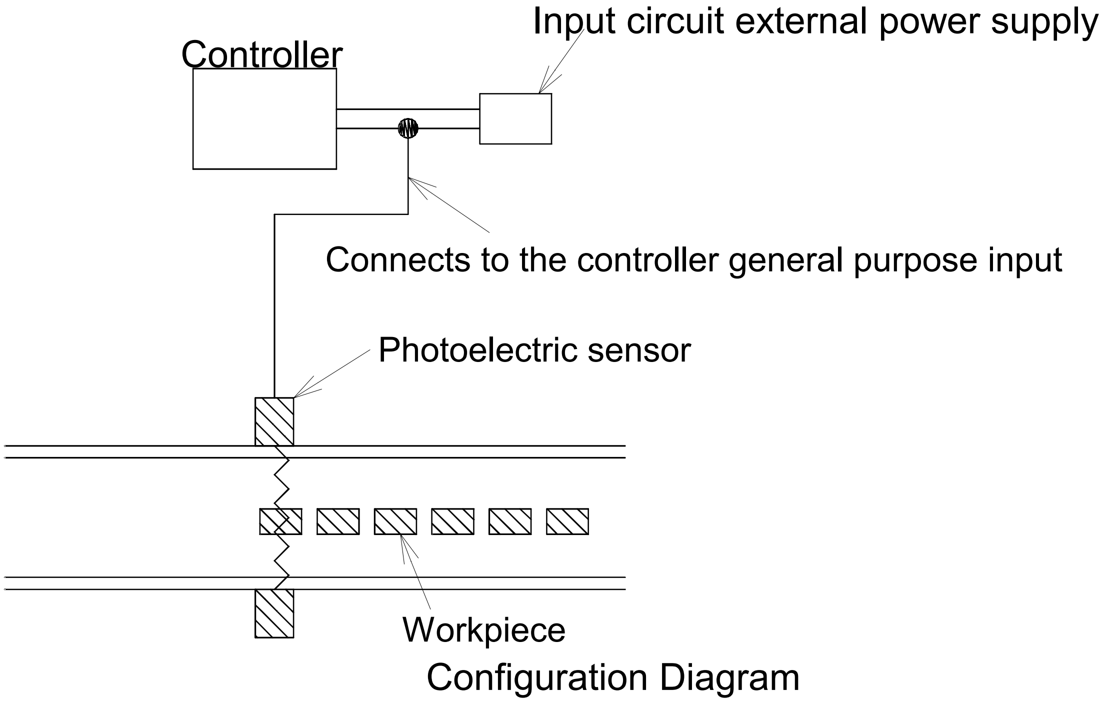
Figure 5–4 Connector Connection of Expansion serial interface Card

Refer to "Table 16–2 Connectors: CNENC Pin Assignment" with pin assignment of connector (CNENC).

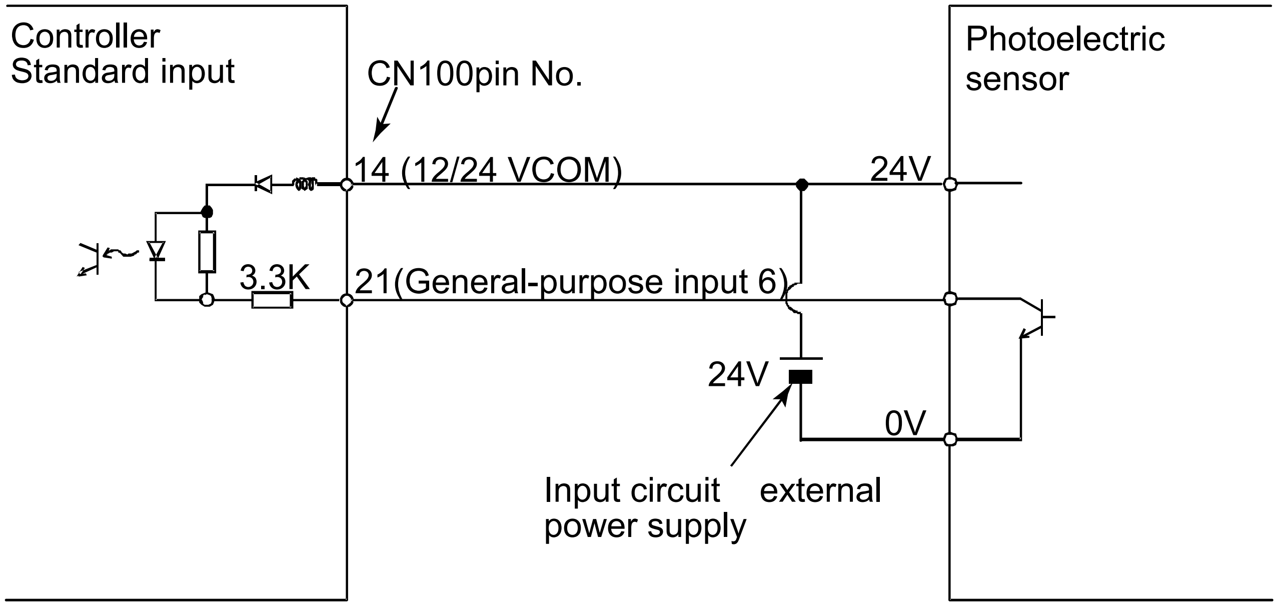
5.2.3. Connection of Photoelectronic Sensor

If a photoelectronic sensor is used for detection of workpieces, connect the output signal of the photoelectronic sensor to a 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 5–5 Photoelectronic Sensor Connection Example (6th General Input Signal is Used)."


Input circuit external power supply
Controller
Connects to the controller general purpose input
Photoelectric sensor
Workpiece

Configuration Diagram



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

Connection Diagram

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

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 (BFP-A5992)” for how to set the parameters.

6.1. Dedicated Input/Output Parameters

“Table 6–1 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 Output: Output program standby status	**0 , -1**
Servo OFF/servo ON disabled (**SRVOFF**)	Input: Turn the servo off Output: Output servo ON disabled status	**1 , -1**
Error reset/error occurring (**ERRRESET**)	Input: Cancel error status Output: Output error status	**2 , -1**
Start/operating (**START**)	Input: Start automatic operation Output: Output program running status	**3 , 1**
Servo ON/turning servo ON (**SRVON**)	Input: Turn the servo on Output: Output servo on status	**4 , 0**
Operation right/operation right enabled (**IOENA**)	Input: Enable/disable operation right of external signal control Output: Output external signal control operation enabled status	**5 , -1**
Program reset/program selectable (**SLOTINIT**)	Input: Initiate a program. The program execution returns to the first step. Output: Output a status where program No. can be changed	**10 , -1**
General output signal reset (**OUTRESET**)	Input: Reset a general output signal	**11 , -1**
User specification area 1 (**USRAREA**)	Output an indication that the robot is in an area specified by a user Set the start number and end number	**8 , 8**

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

6.2. Operation Parameters

“Table 6–2 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. For example, if the hand weighs 3 kg, changing the weight setting value from 10 kg to 3 kg makes the robot movement faster. (Hand weight (kg), size (mm) X, Y, Z, gravity (mm) X, Y, Z)	(3,0,0,0,0,0,0) The setting values are different for each robot model. **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. 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. (Workpiece weight (kg), size (mm) X, Y, Z, gravity (mm) X, Y, Z)	(1,0,0,0,0,0,0) The setting values are different for each robot model. **Use these values as reference only.**

6.3. Tracking Parameter Setting

Specify to which channel of a Expansion serial interface card an encoder of a conveyer is connected.
"Table 6–3 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 6–3 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

0: Disable/1: Enable</td><td>0</td></tr>
<tr><td>External encoder</td><td>EXTENC</td><td>8 integers</td><td>Set connection destinations on the Expansion serial interface card for encoder numbers 1 to 8.
Parameter elements correspond to encoder number 1, encoder number 2 … encoder number 8 from the left.

To set this parameter, enter a value from 1 to 6 ("Parameter value" column) to a parameter value.
<table>
<tr><th>Slot number</th><th>Encoder channel</th><th>Parameter value</th></tr>
<tr><td rowspan="2">1</td><td>1</td><td>1</td></tr>
<tr><td>2</td><td>2</td></tr>
<tr><td rowspan="2">2</td><td>1</td><td>3</td></tr>
<tr><td>2</td><td>4</td></tr>
<tr><td rowspan="2">3</td><td>1</td><td>5</td></tr>
<tr><td>2</td><td>6</td></tr>
</table>

<Example>
If an encoder with encoder number 1 is connected to channel 1 (CH1) of a Expansion serial interface card mounted in option slot 2
EXTENC=(3,0,0,0,0,0,0,0)
With this setting, it is possible to check the encoder value of encoder number 1 with the status variable "M_ENC(1)." In this case, encoder number 1 is connected to CH1 of a Expansion serial interface card mounted in option slot 2.

It is convenient to check the status variable "M_ENC" when determining the setting value of the "EXTENC" parameter.
Please refer to "14.1.2 List of Robot Status Variables" for the explanation of state variable "M_ENC".
Please refer to "Detailed Explanations of Functions and Operations (BFP-A5992)" 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)

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.
The sensor between values [mm] set to this parameter does not react after turning on the sensor.</td><td>5.00</td></tr>
</table>

7. 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 7−1 List of Sample Robot Programs (Conveyer Tracking)” and “Table 7−2 List of Sample Robot Programs (Vision Tracking)” respectively.
Please download the sample program from the **MELFANS Website(http://Global.MitsubishiElectric.com/)**.
Refer to “RT ToolBox Robot Total Engineering Support Software Instruction Manual (BFP-A8090)” for how to install programs to the robot controller.

Table 7−1 List of Sample Robot Programs (Conveyer Tracking)

Program name	Description	Explanation
A	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.
C	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. (1) Movement to the robot origin (2) Workpiece suction and transportation operation while following movement
CM1	Workpiece coordinate monitor program	This program monitors encoder values and stores workpiece coordinates.

Table 7−2 List of Sample Robot Programs (Vision Tracking)

Program name	Description	Explanation
A	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.
B	Vision coordinate system – robot coordinate system calibration program	This program matches the vision coordinate system and the robot coordinate system.
C	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. (1) Movement to the robot origin (2) Workpiece suction and transportation operation while following movement
CM1	Workpiece coordinate monitor program	This program monitors encoder values and stores workpiece coordinates.

8. Calibration of Conveyer and Robot Coordinate Systems (“A” program)

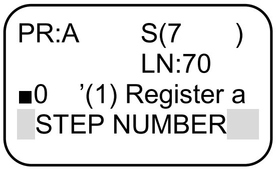
This chapter explains the tasks carried out by using “A” program.
*** “A” program contains operations required for both conveyer tracking and vision tracking.**
Calibration of a conveyer refers to determining the movement direction of the conveyer in the robot coordinate system and the amount of movement of the robot per encoder pulse. This amount of movement is stored in the robot’s status variable “P_ENCDLT.”
“A” 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 “A” program and items to be confirmed after the operations are explained below.
Please refer to “Detailed Explanations of Functions and Operations (BFP-A5992)” for the steps involved in each operation.

(1) Operation procedure

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

② Open “A” program using the teaching pendant (hereinafter referred to as T/B).

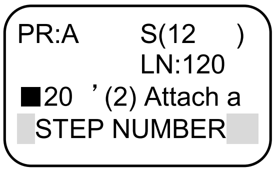
③ As step operation begins, “ '(1) …” will be displayed. Check the contents.



```
PR:A      S(7     )
          LN:70
■0   ’(1) Register a
 STEP NUMBER
```

④ Perform operations according to the instructions (the tasks are explained in **(2) Tasks**).

⑤ Step forward until the next “ '(2) …” is reached.



```
PR:A      S(12    )
          LN:120
■20  ’(2) Attach a
 STEP NUMBER
```

⑥ Repeat steps ③ and ④and perform step operation until “END.”

⑦ Press [INP] while holding down [ERROR RESET] to return to the start of the program.

⑧ Press the [MENU] key to end the program.

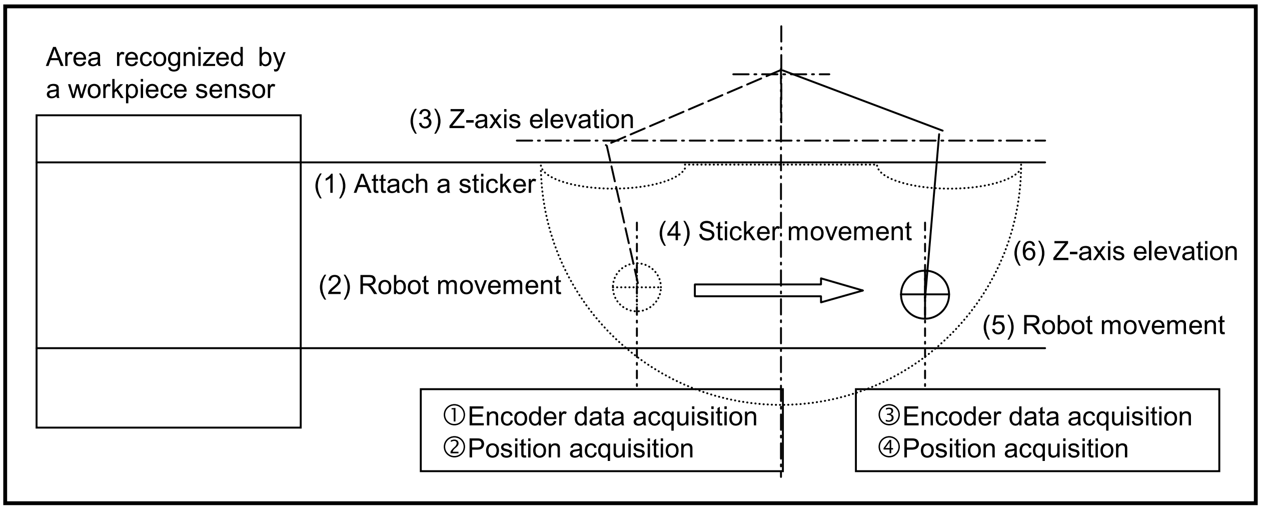


Figure 8–1 Conveyer and Robot Calibration Operation Diagram

(2) Tasks

(1) Enter the encoder number in the X coordinate of the position variable “PE.”

(a) Open the [Position data Edit] screen.

(b) Display ”PE” at the position name.

```
MO.POS(PE      )
     X :       0.00
     Y :       0.00
     Z :       0.00
```

(c) Enter the encoder number in the X coordinate.

```
MO.POS(PE      )
     X :      +1.00
     Y :      +0.00
     Z :      +0.00
```

(d) Return to the [Program instruction 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.

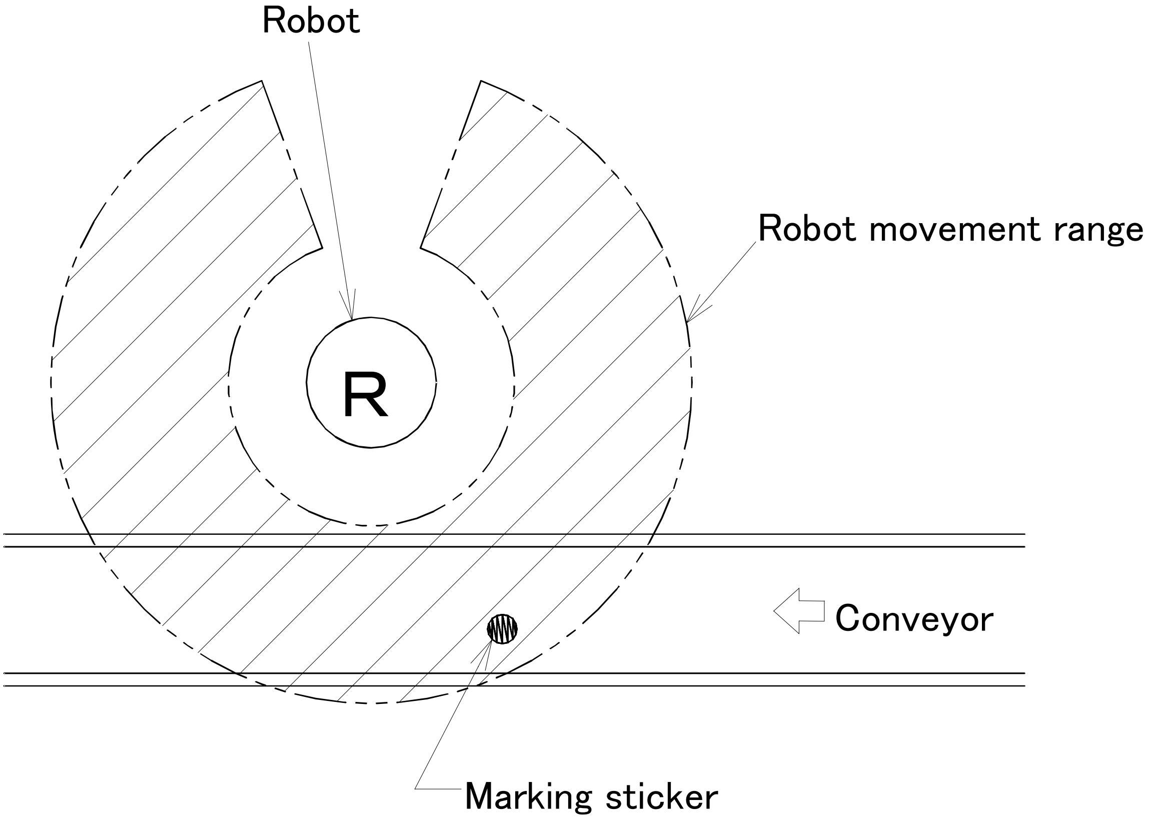


Figure 8–2 Position of Marking Sticker on Conveyer

(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.**

(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.

(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 ToolBox when confirming the data after operation.

⚠ CAUTION

It is not possible to check the value of P_ENCDLT(2) using T/B(R28TB).

With T/B(R28TB), no more than 8 characters can be entered in the variable name input column. Use the program monitor tool of RT Tool Box to check the value of P_ENCDLT(2).

Refer to “RT ToolBox Robot Total Engineering Support Software Instruction Manual (BFP-A8090)” for how to check variable values using RT ToolBox.

9. Calibration of Vision Coordinate and Robot Coordinate Systems (“B” program)

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

*** “B” 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 “calibration data creation” function of MELFA-Vision (network vision support software). Refer to “Network Vision Sensor Instruction Manual (BFP-A8520)” for the details of this function.

“B” 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).

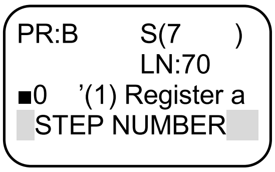
The procedures of operations specified by “B” program and items to be confirmed after the operations are explained below.

Please refer to “Detailed Explanations of Functions and Operations (BFP-A5992)” for the steps involved in each operation.

(1) Operation procedure

① Open “B” program using T/B.

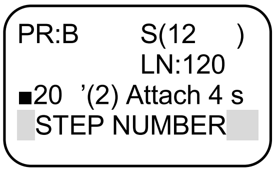
② As step operation begins, “ '(1) …” will be displayed. Check the contents.



```
PR:B       S(7      )
           LN:70
■0   ’(1) Register a
 STEP NUMBER
```

③ Perform operations according to the instructions (the tasks are explained in **(2) Tasks**).

④ Step forward until the next “ '(2) …” is reached.



```
PR:B       S(12     )
           LN:120
■20  ’(2) Attach 4 s
 STEP NUMBER
```

⑤ Repeat steps ③ and ④ and perform step operation until “END.”

⑥ Press [INP] while holding down [ERROR RESET] to return to the start of the program.

⑦ Press the [MENU] key to end the program.

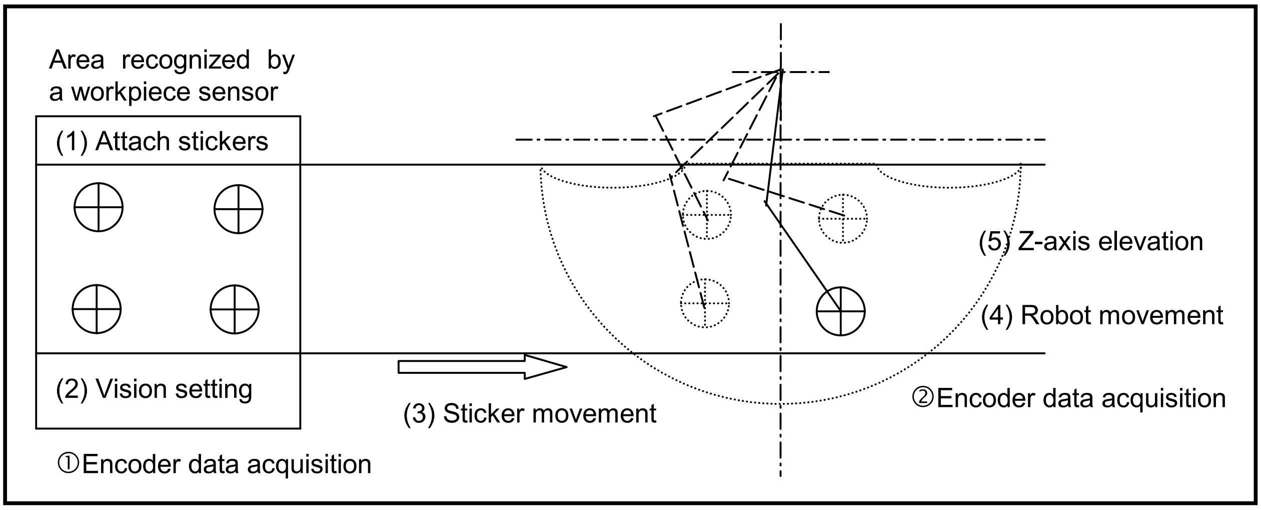


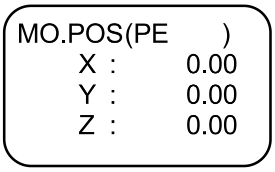
Figure 9–1 Vision Sensor and Robot Calibration Operation Procedure Diagram

(2) Tasks

(1) Enter the encoder number in the X coordinate of the position variable “PE.”

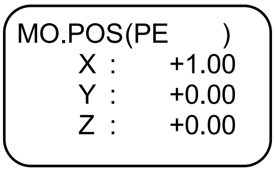
(a) Open the [Position data Edit] screen.

(b) Display “PE” at the position name.



```
MO.POS(PE      )
      X :       0.00
      Y :       0.00
      Z :       0.00
```

(c) Enter the encoder number in the X coordinate.



```
MO.POS(PE      )
      X :     +1.00
      Y :     +0.00
      Z :     +0.00
```

(d) Return to the [Program instruction Edit] screen.

(2) Start MELFA-Vision and place the vision sensor in the offline status. Select [Live Mode] in MELFA-Vision to display images taken by the vision sensor in real-time.

Please refer to “Network Vision Sensor Instruction Manual (BFP-A8520)” for the operation of MELFA-Vision.

(3) Paste four marking stickers on the conveyer (stickers with X marks are the best choice for the marking stickers).

Attach these marking stickers within the field of vision of the vision sensor while checking the live images of MELFA-Vision.

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

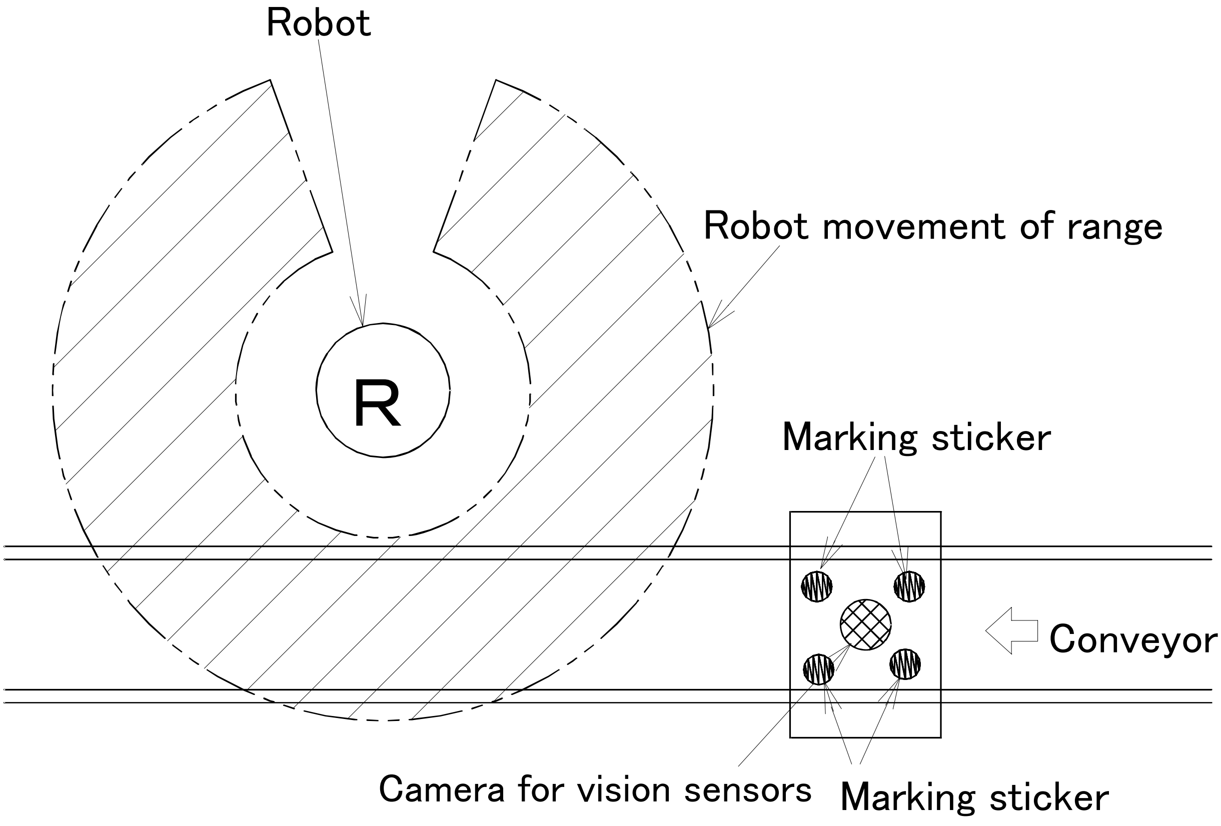


Figure 9–2 Pasting Marking Stickers

(4) Close [Live Mode] of MELFA-Vision and open the Create Calibration Data screen. Click the [Image] button in the [Sensor Reference Point] column and register the positions of the four stickers on [Camera Image].

(5) Move the marking stickers within the robot movement range.

(6) Move the robot to the position right above the first marking sticker on the conveyer.

(7) Click the [Position] button in the Create Calibration Data screen of MELFA-Vision and acquire the current robot position.

(8) Move the robot to the position right at the second marking sticker.

(9) Click the [Position] button in the Create Calibration Data screen and acquire the current robot position.
(10) Move the robot to the position right at the third marking sticker.
(11) Click the [Position] button in the Create Calibration Data screen and acquire the current robot position.
(12) Move the robot to the position right at the fourth marking sticker.
(13) Click the [Position] button in the Create Calibration Data screen and acquire the current robot position.
(14) Raise the robot.
*** With this operation, encoder data is acquired.**
(15) Use MELFA-Vision to save the calibration data.

* **With this operation, the workpiece coordinates recognized by the vision sensor can be displayed in the robot coordinate system.**

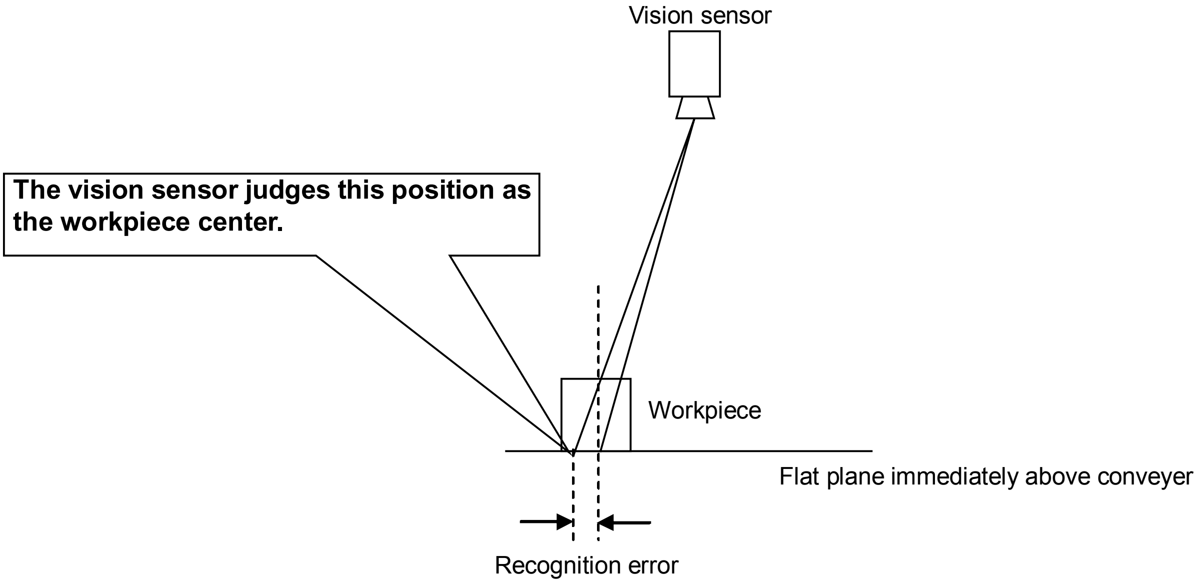
(3) Confirmation after operation

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.


Vision sensor
The vision sensor judges this position as the workpiece center.
Workpiece
Flat plane immediately above conveyer
Recognition error

10. Workpiece Recognition and Teaching (“C” program)

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

* **“C” program contains operations required for both conveyer tracking and vision tracking, but different operations are performed. Refers to “10.1Program for Conveyer Tracking” for operations in the case of conveyer tracking and “10.2Program for Vision Tracking” for operations in the case of vision tracking.**

Please refer to “Detailed Explanations of Functions and Operations (BFP-A5992)” for the steps involved in each operation.

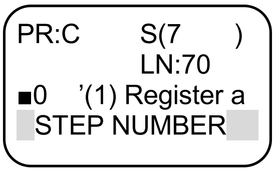
10.1. Program for Conveyer Tracking

In “C” 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 “C” program for conveyer tracking are explained below.

(1) Operation procedure

① Open “C” program using T/B.

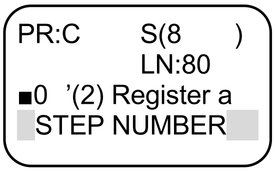
② As step operation begins, “ '(1) …” will be displayed. Check the contents.



```
PR:C       S(7      )
           LN:70
■0   ’(1) Register a
 STEP NUMBER
```

③ Perform operations according to the instructions (<u>the tasks are explained in **(2) Tasks**</u>).

④ Step forward until the next “ '(2) …” is reached.

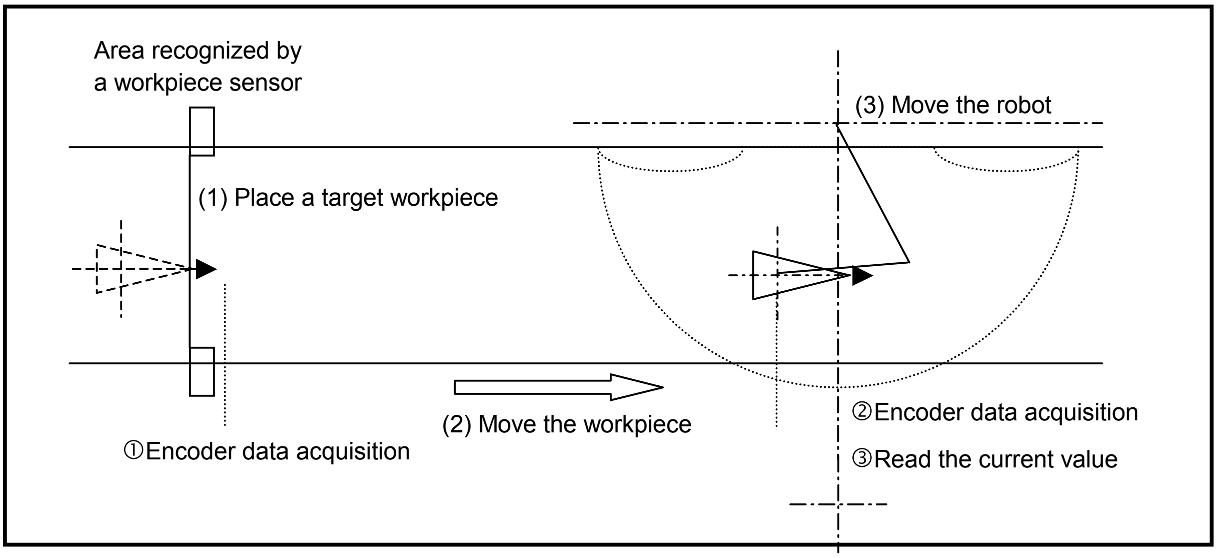


```
PR:C       S(8      )
           LN:80
■0  ’(2) Register a
 STEP NUMBER
```

⑤ Repeat steps ③ and ④ and perform step operation until “END.”

⑥ Press [INP] while holding down [ERROR RESET] to return to the start of the program.

⑦ Press the [MENU] key to end the program.



Area recognized by a workpiece sensor
(3) Move the robot
(1) Place a target workpiece
(2) Move the workpiece
①Encoder data acquisition
②Encoder data acquisition
③Read the current value

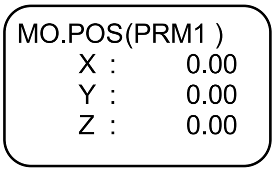
Figure 10–1 Operation for Matching Workpiece Coordinates and Robot Coordinates

(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.

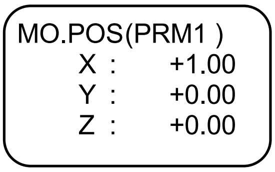
(a) Open the [Position data Edit] screen.

(b) Display “PRM1” at the position name.



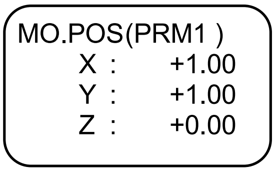
```
MO.POS(PRM1 )
     X :       0.00
     Y :       0.00
     Z :       0.00
```

(c) Enter the model number in the X coordinate.



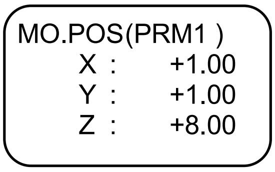
```
MO.POS(PRM1 )
     X :     +1.00
     Y :     +0.00
     Z :     +0.00
```

(d) Enter the encoder number in the Y coordinate.



```
MO.POS(PRM1 )
     X :     +1.00
     Y :     +1.00
     Z :     +0.00
```

(e) Enter the number of the sensor that monitors the workpieces in the Z coordinate.



```
MO.POS(PRM1 )
     X :     +1.00
     Y :     +1.00
     Z :     +8.00
```

(f) Return to the [Program instruction 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.

10.2. Program for Vision Tracking

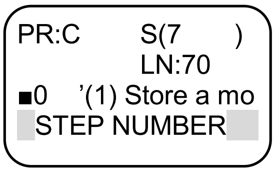
Vision tracking “C” program acquires encoder data at the position where the vision sensor recognizes workpieces and where the robot suctions workpieces such that the robot can recognize the work coordinates recognized by the vision sensor. The following explains the operation procedure and items to confirm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

(1) Operation procedure

① Register workpieces to be recognized by a vision sensor and create a vision program.

② Open “C” program using T/B.

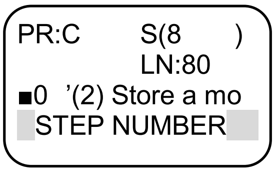
③ As step operation begins, “ '(1) …” will be displayed. Check the contents.



```
PR:C       S(7      )
           LN:70
■0   ’(1) Store a mo
 STEP NUMBER
```

④ Perform operations according to the instructions (the tasks are explained in **(2) Tasks**).

⑤ Step forward until the next “ '(2) …” is reached.



```
PR:C       S(8      )
           LN:80
■0  ’(2) Store a mo
 STEP NUMBER
```

⑥ Repeat steps ④ and ⑤ and perform step operation until “END.”

⑦ Press [INP] while holding down [ERROR RESET] to return to the start of the program.

⑧ Press the [MENU] key to end the program.

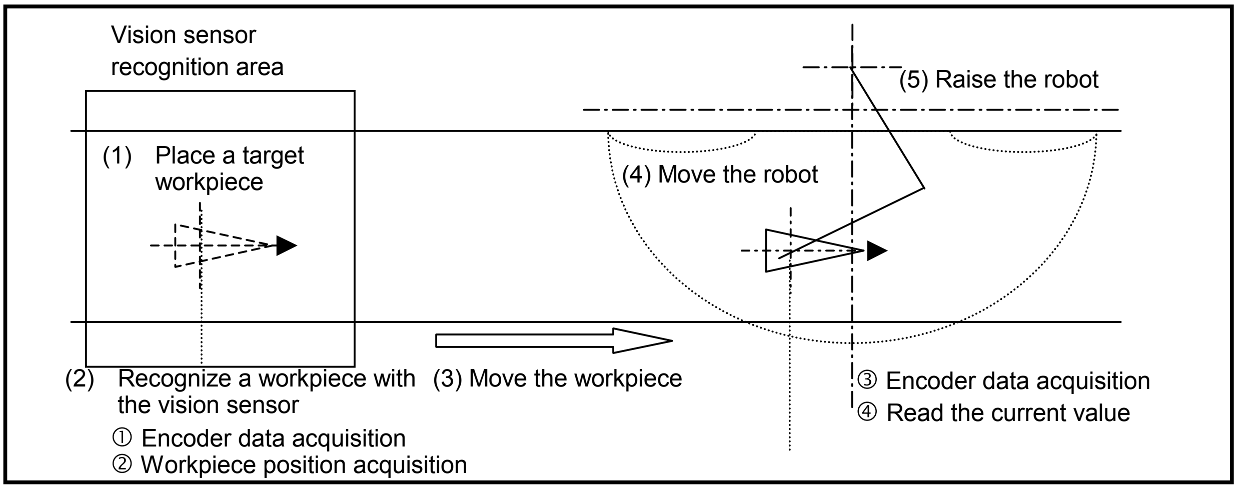
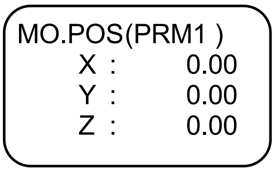

Vision sensor recognition area
(1) Place a target workpiece
(5) Raise the robot
(4) Move the robot
(2) Recognize a workpiece with the vision sensor
① Encoder data acquisition
② Workpiece position acquisition
(3) Move the workpiece
③ Encoder data acquisition
④ Read the current value

Figure 10‒2 Operation for Matching Workpiece Coordinates and Robot Coordinates

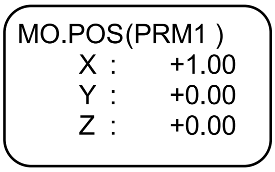
(2) Tasks

(1) Enter the model number and encoder number in the X and Y and Z coordinates of the position variable “**PRM1**” in the program.
(a) Open the [Position data Edit] screen.
(b) Display “PRM1” at the position name.



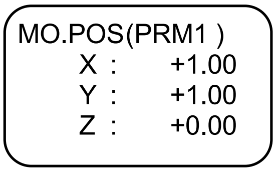
```
MO.POS(PRM1 )
    X :      0.00
    Y :      0.00
    Z :      0.00
```

(c) Enter the model number in the X coordinate.



```
MO.POS(PRM1 )
    X :     +1.00
    Y :     +0.00
    Z :     +0.00
```

(d) Enter the encoder number in the Y coordinate.

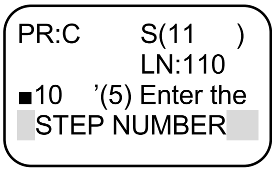


```
MO.POS(PRM1 )
    X :     +1.00
    Y :     +1.00
    Z :     +0.00
```

(d) Return to the [Program instruction Edit] screen.

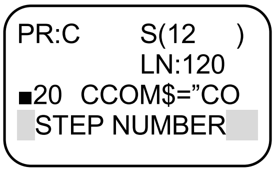
(3) Start MELFA-Vision and place the vision sensor in the offline status. Select [Live Mode] in MELFA-Vision to display images taken by the vision sensor in real-time. Check the images and set the field of vision in the moving direction of the conveyer (mm) and the length of workpieces detected by the 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 [Program instruction Edit] screen.

(5) Specify a communication line to be connected with the vision sensor.
(a) Open the [Program instruction Edit] screen.



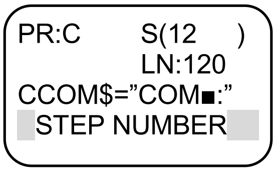
```
PR:C       S(11    )
           LN:110
■10   '(5) Enter the
 STEP NUMBER
```

(b) Display the instruction in the next line.

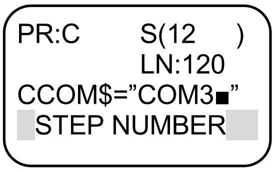


```
PR:C       S(12    )
           LN:120
■20  CCOM$="CO
 STEP NUMBER
```

(c) Edit the program and specify a communication line to be opened by the robot controller in order to connect with the vision sensor in the variable “CCOM$.”
Example) If COM3 is opened:



```
PR:C       S(12    )
           LN:120
CCOM$="COM■:"
 STEP NUMBER
```



```
PR:C       S(12    )
           LN:120
CCOM$="COM3■"
 STEP NUMBER
```

(6) Specify a vision program to be started.
In the same way as in step (5), change the vision program name entered after “CPRG$=” in the program.

(7) Specify the cell in the program in which the number of recognized workpieces is stored.

In the same way as in step (5), change the cell storing the number of recognized workpieces entered after “CKOSU$=” in the program.

(8) Specify the starting cell of the area where recognition results are stored in the vision program.
In the same way as in step (5), change the starting cell of the area storing recognition results entered after “CSTT$=” in the program.

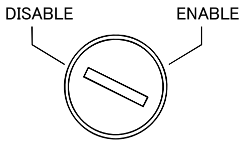
(9) Specify the ending cell of the area where recognition results are stored in the vision program.
In the same way as in step (5), change the ending cell of the area storing recognition results entered after “CEND$=” in the program.

(10) Place a workpiece to be recognized within the area that the vision sensor can recognize.

(11) Using MELFA-Vision, place the vision sensor in the online status.

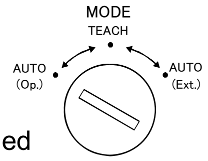
(12) Using T/B, close the opened “C” program once and then run the modified “C” program automatically with the robot controller.

T/B disabled

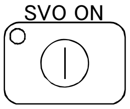


Set the [ENABLE / DISABLE] switch of T/B to “DISABLE” and the [MODE] switch of the controller to “AUTO(Op.)”.

Controller enabled



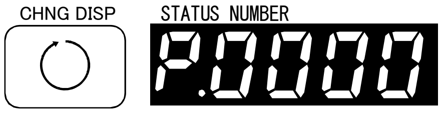
Servo on



The servo is turned off once. Turn the servo on by pressing the [SVO.ON] button of the controller.

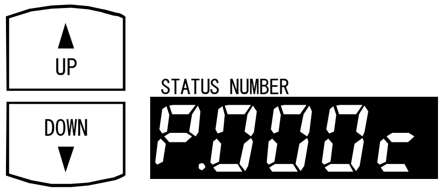
Selection of a program number

Display of
a program number



Press the [CHANG DISP] switch of the controller to display the “program number” on the STATUS NUMBER display panel.

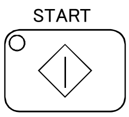
Selection of
a program number



Press the [UP] or [DOWN] switch of the controller to display the program number “C” of the sample program.

Start of automatic operation

Start



Press the [START] button of the controller.

After automatic operation, “C” program automatically stops and the LED of the [STOP] button is turned on. Open “C” program again with T/B. Press the [+/FORWD] button to display the subsequent operation messages.

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

(13) Rotate the conveyer forward and move a workpiece within the vision sensor recognition area into the

robot movement range.

(14) Move the robot to the position where it is able to suction the workpiece.

*** With this operation, encoder data and robot position are acquired.**

(15) 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
- Value of “C_102$()”: Recognized quantity cell
- Value of “C_103$()”: Recognition area start cell
- Value of “C_104$()”: Recognition area end cell

Confirm that each of the above values is entered.

11. 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.

(1) Teaching

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.

(2) Setting of adjustment variables in the program

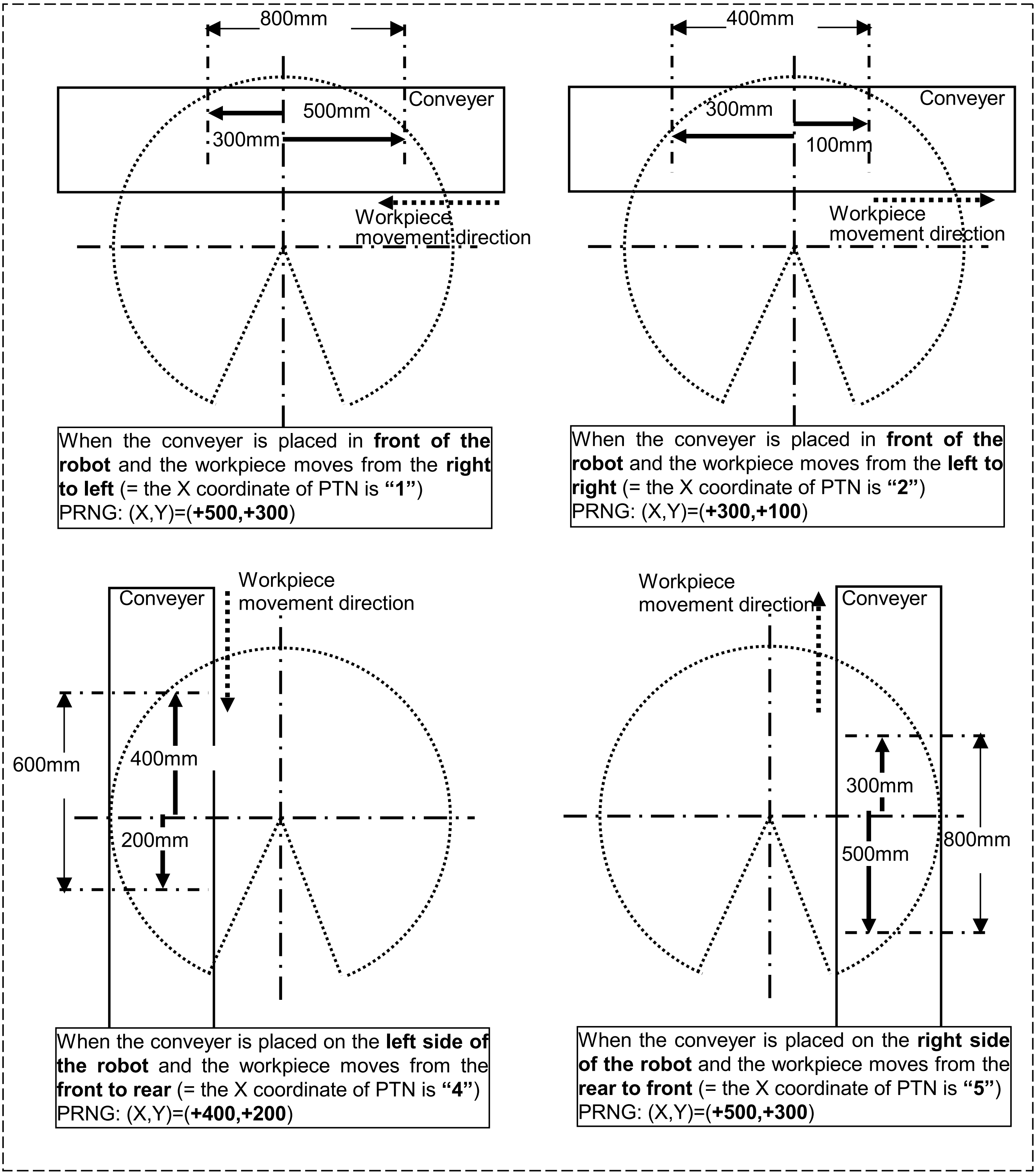
The following section explains how to set adjustment variables, which are required at transportation, and details about their setting.
Please refer to “Detailed Explanations of Functions and Operations (BFP-A5992)” for how to set adjustment variables.

Table 11–1 List of Adjustment Variables in Programs

Variable name	Explanation	Setting example
PWK	Set the model number. X = model number(1 to 10)	When you set 1 to the model number: (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. It is possible to specify which program is processed with a higher priority, rather than performing the same amount of processing at the same time. X = Set the line numbers of “1” program to be performed (1 to 31). Y = Set the line numbers of “CM1” program to be performed (1 to 31).	When you set to run “1” program by one line and run “CM1” program by 10 lines: (X, Y, Z, A, B, C) = (+1,+10,+0,+0,+0,+0)
PUP1	When operating by the adsorption of workpiece, set the height that the robot works. Height sets the amount of elevation (mm) from the position where workpiece is adsorbed. X = Amount of elevation of the position where a robot waits until a workpiece arrives. (mm) Y = Amount of elevation from the workpiece suction position (before suctioning) (mm) Z = Amount of elevation from the workpiece suction position (after suctioning) (mm) *** 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: Amount of elevation of the position where a robot waits until a workpiece arrives : 50 mm Amount of elevation from the workpiece suction position (before suctioning) : -50 mm Amount of elevation from the workpiece suction position (after suctioning) : -50 mm (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. Height sets the amount of elevation (mm) from the position where workpiece is adsorbed. Y = Amount of elevation from the workpiece release position (before release). (mm) Z = Amount of elevation from the workpiece release position (after release). (mm) *** 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: Amount of elevation from the workpiece release position (before release) : -50 mm Amount of elevation from the workpiece release position (after release) : -50 mm (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. X = The acceleration until moving to the position on the workpiece. (1 to 100) (%) Y = The deceleration until moving to the position on the workpiece. (1 to 100) (%) *** The value set by X coordinates and Y coordinates of “PAC*” is used for <acceleration ratio(%)> of the ACCEL instruction and <deceleration ratio(%)>. The value is reduced when the speed of time when the robot vibrates and the robot is fast.**	When the following values are set: Acceleration until moving to the position on the workpiece. : 100% Deceleration until moving to the position on the workpiece. : 100% (X, Y, Z, A, B, C) = (+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. X = The acceleration until moving to the workpiece suction position. (1 to 100) (%) Y = The deceleration until moving to the workpiece suction position. (1 to 100) (%)	When the following values are set: Acceleration until moving to the workpiece suction position. : 10% Deceleration until moving to the workpiece suction position. : 20% (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. X = The acceleration until moving to the position on the workpiece. (1 to 100) (%) Y = The deceleration until moving to the position on the workpiece. (1 to 100) (%)	When the following values are set: Acceleration until moving to the position on the workpiece. : 50% Deceleration until moving to the position on the workpiece. : 80% (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. X = The acceleration until moving to the position release position. (1 to 100) (%) Y = The deceleration until moving to the position release position. (1 to 100) (%)	When the following values are set: Acceleration until moving to the position on the workpiece : 80% Deceleration until moving to the position on the workpiece : 70% (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. X = The acceleration until moving to the workpiece release position. (1 to 100)(%) Y = The deceleration until moving to the workpiece release position. (1 to 100)(%)	When the following values are set: Acceleration until moving to the workpiece release position. : 5% Deceleration until moving to the workpiece release position. : 10% (X, Y, Z, A, B, C) = (+5,+10,+0,+0,+0,+0)

<table>
<tr><td>PAC13</td><td>When operating by the release of workpiece, the acceleration and the deceleration when moving toward the position on the workpiece are set.
X = The acceleration until moving to the position on the workpiece. (1 to 100) (%)
Y = The deceleration until moving to the position on the workpiece. (1 to 100) (%)</td><td>When the following values are set:
Acceleration until moving to the position on the workpiece. : 100%
Deceleration until moving to the position on the workpiece. : 100%
(X, Y, Z, A, B, C) =
(+100,+100,+0,+0,+0,+0)</td></tr>
<tr><td>PHND</td><td>Set the Enable or Disable of the process and the signal which confirms adsorption.
X = Enable or Disable of the process.
(0: Disable / 1: Enable)
Y = The signal number which confirms adsorption.
(When the signal is turned on, the robot has adsorbed workpieces.)
Z = The signal number which confirms release.
(When the signal is turned on, the robot is not adsorbing workpieces.)</td><td>When the following values are set:
Enable or Disable of the process :Enable
The adsorption signal number : 900
The release signal number : 901
(X, Y, Z, A, B, C) =
(+1,+900,+901,+0,+0,+0)</td></tr>
<tr><td>PDLY1</td><td>Set the suction time.
X: Suction time (s).</td><td>When setting 0.5 second for the sucking time:
(X, Y, Z, A, B, C) = (+0.5,+0,+0,+0,+0,+0)</td></tr>
<tr><td>PDLY2</td><td>Set the release time.
X: Release time (s).</td><td>When setting 0.3 second for the release time:
(X, Y, Z, A, B, C) = (+0.3,+0,+0,+0,+0,+0)</td></tr>
<tr><td>POFSET</td><td>When the adsorption position shifts, the gap can be corrected. Set the correction value.
<b>* 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.</b></td><td></td></tr>
<tr><td>PTN</td><td>Set the position of the robot and conveyer, and the direction where the workpiece moves.
X = The following values. (1 to 6)
<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>Leftv</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)
(X, Y, Z, A, B, C) = (+1,+0,+0,+0,+0,+0)
<b>The relationship between PRNG and PTN is shown in “Figure 11–1 Diagram of Relationship between Adjustment Variables “PRNG” and “PTN” in the Program”.</b></td></tr>
<tr><td>PRNG</td><td>Set range of motion where the robot judges workpiece to be able to follow.
X = The start distance of the range in which the robot can follow a workpiece :(mm)
Y = The end distance of the range in which the robot can follow a workpiece :(mm)
Z = The distance in which follow is canceled :(mm)</td><td><b>The relationship between PRNG and PTN is shown in “Figure 11–1 Diagram of Relationship between Adjustment Variables “PRNG” and “PTN” in the Program”.</b></td></tr>
</table>


800mm
Conveyer
500mm
300mm
Workpiece movement direction
400mm
Conveyer
300mm
100mm
Workpiece movement direction
Conveyer
Workpiece movement direction
600mm
400mm
200mm
Workpiece movement direction
Conveyer
300mm
500mm
800mm

When the conveyer is placed in **front of the robot** and the workpiece moves from the **right to left** (= the X coordinate of PTN is **"1"**)
PRNG: (X,Y)=(**+500,+300**)

When the conveyer is placed in **front of the robot** and the workpiece moves from the **left to right** (= the X coordinate of PTN is **"2"**)
PRNG: (X,Y)=(**+300,+100**)

When the conveyer is placed on the **left side of the robot** and the workpiece moves from the **front to rear** (= the X coordinate of PTN is **"4"**)
PRNG: (X,Y)=(**+400,+200**)

When the conveyer is placed on the **right side of the robot** and the workpiece moves from the **rear to front** (= the X coordinate of PTN is **"5"**)
PRNG: (X,Y)=(**+500,+300**)

Figure 11–1 Diagram of Relationship between Adjustment Variables "PRNG" and "PTN" in the Program

12. 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.

12.1. 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 “A” program and “C” 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

12.2. 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 “A” program, “B” program and “C” 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 “C” program.
<Data specified in “C” 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.

13. Automatic Operation

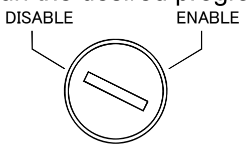
This chapter explains how to prepare the robot before starting the system.

(1) Preparation

① Check that there is no interfering object within the robot movement range.

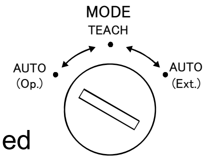
② Prepare to run the desired program.

T/B disabled

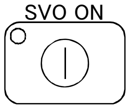


Set the [ENABLE / DISABLE] switch of T/B to “DISABLE” and the [MODE] switch of the controller to “AUTO(Op.)”.

Controller enabled



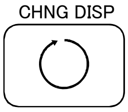
Servo on

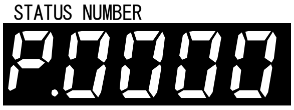


The servo is turned off once. Turn the servo on by pressing the [SVO.ON] button of the controller.

Selection of a program number

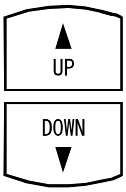
Display of
a program number

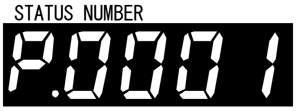




Press the [CHANG DISP] switch of the controller to display the “program number” on the STATUS NUMBER display panel.

Selection of
a program number





Press the [UP] or [DOWN] switch of the controller to display the program number “C” of the sample program.

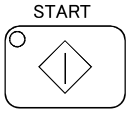
(2) Execution

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.

Start of automatic operation

Start



Press the [START] button of the controller.

(3) At error occurrence

If the robot moves erroneously, refer to “CR1/CR2/CR4/CR7/CR8 Controller Instruction Manual Troubleshooting” (BFP-A5993).

(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.

14. Maintenance

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

14.1. MELFA-BASIC IV Instructions

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

14.1.1. List of Instructions

Table 14–1 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.

14.1.2. List of Robot Status Variables

Table 14–2 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 External encoder data can be rewritten. **If this state variable does not set parameter “TRMODE" to “1", the value becomes like “0".**	R/W	Double-precision real number
P_ENCDLT	number of encoders 1 to 8	Amount of robot movement per encoder pulse ***This state variable is made by sample “A” 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 1: Tracking 0: Not tracking	R	Integer

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

14.1.3. List of Functions

Table 14–3 List of Functions

Function name	Function	Result
POSCQ(<position>)	Check whether the specified position is within the movement range. 1: Within the movement range 0: Outside the movement range	Integer
TRWCUR(<encoder number>,<position>,<encoder value>)	Obtain the current position of a workpiece. <number of encoders> 1 to 8	Position
TRPOS(<position>)	Acquire the coordinate position of a workpiece being tracked. TRK ON P0,P1,1,M1# PC2=TRPOS(P2) PC2 above is obtained in the following manner. PC1=P1+P_ENCDLT*(M_ENC-M1#) ‘ The current position of P1 PC2=PC1*(P_ZERO/P0*P2)	Position

14.1.4. 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 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]

```
10 TRBASE P0          ' Specify the workpiece coordinate origin at the teaching position.
20 TRRD P1,M1,MKIND   ' Read the workpiece position data from the data buffer.
30 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.
40 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.
50 HCLOSE 1           ' Close hand 1.
60 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.

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]

```
10 TRCLR 1                          ' Clear tracking data buffer No. 1.
20 *LOOP
30 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.
40 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#.
50 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]

```
10 TRBASE P0          ' Specify the workpiece coordinate origin at the teaching position.
20 TRRD P1,M1,MKIND ' Read the workpiece position data from the data buffer.
30 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.
40 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).
50 HCLOSE 1           ' Close hand 1.
60 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.
- Lines 30 and 40 of the example above can also be written as follows.
  30 TRK ON,P1,M1,P0
  40 MVS P2
  In this example, P2 in line 40 is regarded as a position relative to P0.
- “P_ZERO/P0” in ”P1c*P_ZERO/P0*P2” in [Example] can be replaced with INV(P0).

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]

```
10 IF M_IN(10) <> 1 GOTO 10    ' Check whether a photoelectronic sensor is activated.
20 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.
30 IF M_IN(21) <> 1 GOTO 30    ‘ Wait until the signal (general-purpose input No.21) which shows acquiring
                                 image from the vision sensor is turned on.
40 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

```
10 TRBASE P0        ' Specify the workpiece coordinate origin at the teaching position.
20 TRRD P1,M1,MK    ' Read the workpiece position data from the data buffer.
30 TRK ON,P1,M1     ' Start tracking of a workpiece whose measured position is P1 and encoder value
                      at the time of measurement is M1.
40 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.
50 HCLOSE 1         ' Close hand 1.
60 TRK OFF          ' End the tracking operation.
```

(2) Sensor data reception program

```
10 *LOOP
20 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.
30 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#.
40 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

```
10 TRBASE P0          ' Specify the workpiece coordinate origin at the teaching position.
20 TRRD P1,M1,MKIND   ' Read the workpiece position data from the data buffer.
30 TRK ON,P1,M1       ' Start tracking of a workpiece whose measured position is P1 and encoder
                        value at the time of measurement is M1.
40 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.
50 HCLOSE 1           ' Close hand 1.
60 TRK OFF            ' End the tracking operation.
```

(2) Sensor data reception program

```
10 *LOOP
20 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.
30 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#.
40 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.
```

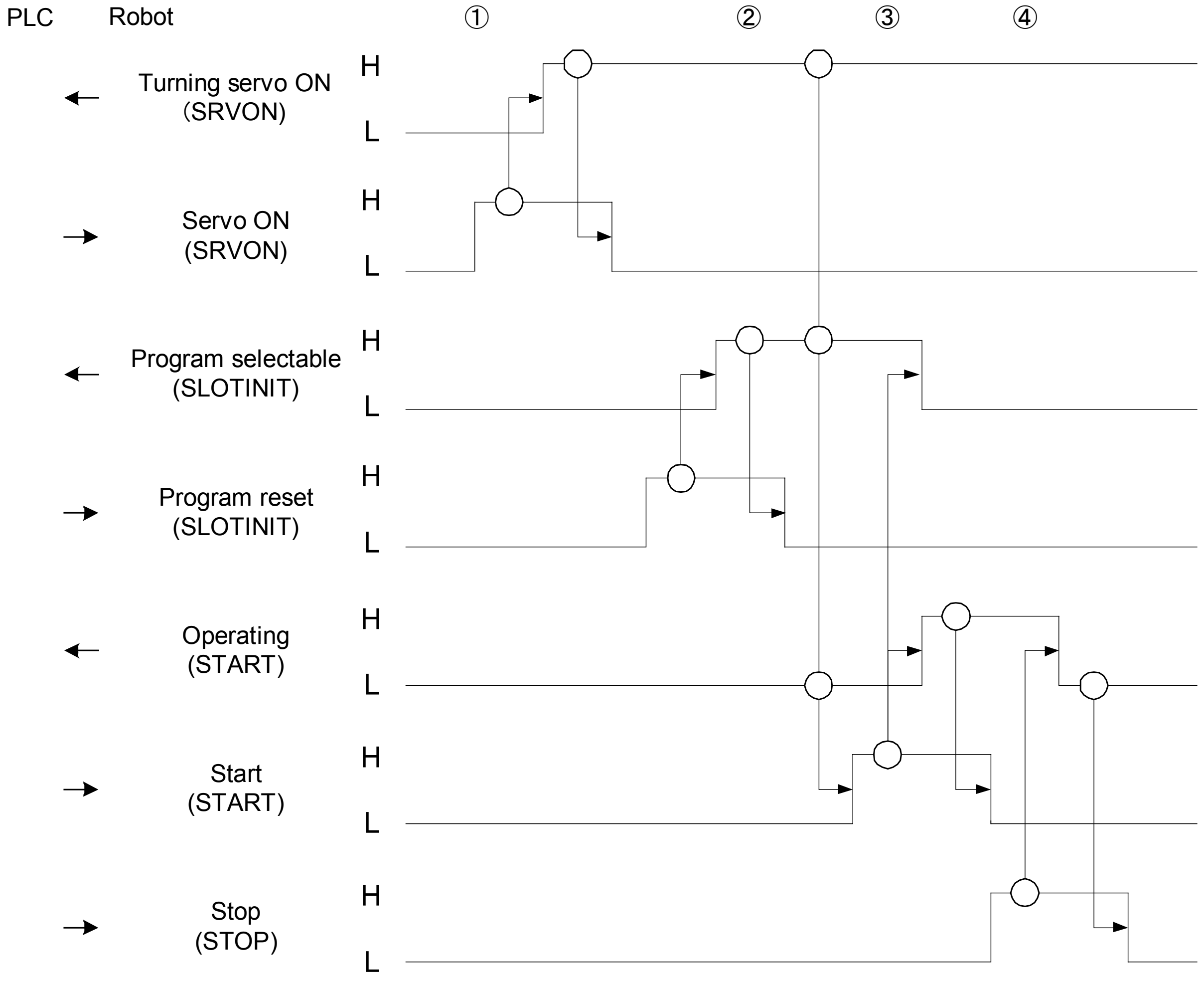
[Explanation]

- 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.

14.2. Timing Diagram of Dedicated Input/Output Signals

14.2.1. 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.”

15. Troubleshooting

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

15.1. 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 15–1 List of Errors in Sample Programs

Error number	Error description	Causes and actions
9100	Communication error	[Causes] The network vision sensor and the robot cannot be connected by the “C” program or the robot cannot log on the vision sensor. [Actions] ① Check the Ethernet cable which connects the robot with the network vision sensor.
9101	Encoder number out of range	[Causes] The encoder number specified in “A” program to “C” program is ”0” or ”9” or larger. [Actions] ① Check the X coordinate of the position variable ”PE” in the programs.
9102	Model number out of range	[Causes] The model number specified in “C” program is ”0” or ”10” or larger. [Actions] ① Check the X coordinate of the position variable “PRM1” in “C” program. ② If there are more than 11 models, change the following line in “C” program. 110 MWKMAX=10
9110	Position accuracy out of range	[Causes] The workpiece position calculated by operations in “A” program to “C” program is very different from the theoretical value. [Actions] ① Check the X and Y coordinates of the position variable ”PVTR” in “CM1” program. These values represent the difference from the theoretical value. ② If the difference stored in “PVTR” is large, run “A” program to “C” program again. 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.
9199	Program error	[Causes] A return value cannot be created by the *S50WKPOS function of “1” program. [Actions] ① Check the reason why “MY50STS” of the *S50WKPOS function in “1” program does not change from”0”.

15.2. Occurrence of Other Errors

Table 15–2 List of Tracking relation Errors

Error number	Error description	Causes and actions
2500	Tracking encoder data error	[Causes] The data of the tracking encoder is abnormal. [Actions] ① Check whether the conveyer rotates at a constant speed. ② Check the wiring for the encoder. ③ Check whether the ground lead is connected.
2510	Tracking parameter reverses	[Causes] Tracking parameter[EXCRGMN] and [EXCRGMX] Setting value reverses [Actions] ① Check the value of [ENCRGMX] and [ENCRGMN] parameters.
2520	Tracking parameter is range over	[Causes] The set value is outside the range parameter [TRBUF]. The first argument is 1 to 8, and the second argument is 1 to 64. [Actions] ① Check the value of [TRBUF] parameter.
2530	There is no area where data is written	[Causes] 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. [Actions] ① Check the execution count of the TRWRT command is correct. ② Check the value of the second argument of parameter [TRBUF] is correct. 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.
2540	There is no read data	[Causes] The workpiece position calculated by operations in “A” program to “C” program is very different from the theoretical value. [Actions] ① Check the X and Y coordinates of the position variable ”PVTR” in “CM1” program. These values represent the difference from the theoretical value. ② If the difference stored in “PVTR” is large, run “A” program to “C” program again. ③ Check the TRRD command executes.
2560	Ilegal parameter of Tracking	[Causes] The TRRD command was executed when there was no data in the tracking buffer. [Actions] ① Execute the TRRD command after Checking whether data in the buffer state variable [M_TRBFCT]. ② Check the buffer number specified by the TRWRT command is corresponding to the buffer number specified by the TRRD command.
3982	Cannot be used (singular point)	[Causes] The robot tried to pass the significant point while doing the tracking. [Actions] ① Keep away the position of the workpiece which flows on the conveyer from the robot. ② Expand the interval between the robot and the conveyer.

Please refer to “CR1/CR2/CR4/CR7/CR8 Controller Instruction Manual Troubleshooting” (BFP-A5993).

16. 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.

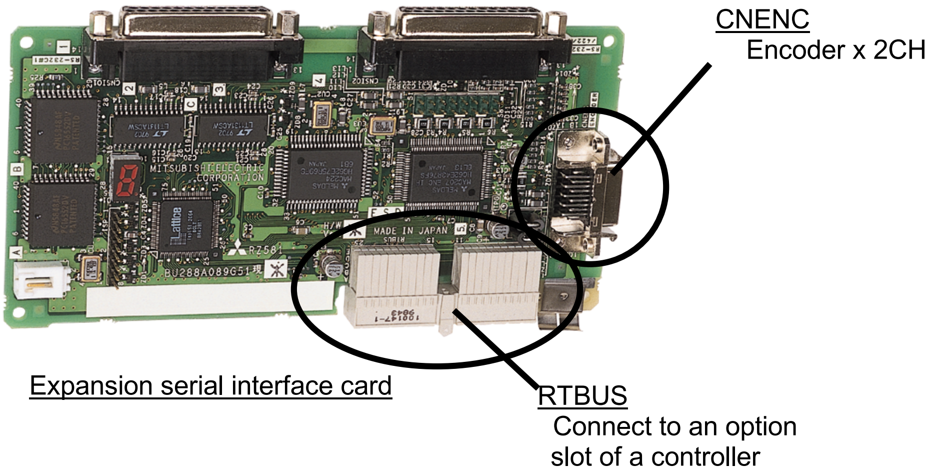
16.1. List of Parameters Related to Tracking

Table 16–1 List of Parameters Related to Tracking

Parameter	Parameter name	Number of elements	Description	Setting value at factory shipment
Minimum external encoder value	ENCRGMN	8 integers	The minimum external encoder data value (pulse) The range of the encoder value which can be acquired in state variable “M_ENC" (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) The range of the encoder value which can be acquired in state variable “M_ENC" (maximum value side)	100000000, 100000000, 100000000, 100000000, 100000000, 100000000, 100000000, 100000000
Tracking buffer	TRBUF	2 integers	Number of tracking buffers and their sizes (KB) <Buffer number> Specify the number of buffers where the tracking data is stored. Setting range: 1 to 8 <Buffer size> Specify the size in which the tracking data is preserved. Setting range: 1 to 64	4 , 64
Tracking adjustment coefficient 1	TRADJ1	8 real numbers	Tracking adjustment coefficient 1 Set the amount of delay converted to the conveyer speed. Convert to 100 mm/s. Example) • If the delay is 2 mm when the conveyer speed is 50 mm/s: Setting value = 4.0 (2 / 50 * 100) • If the advance is 1 mm when the conveyer speed is 50 mm/s: Setting value = -2.0 (-1 / 50 * 100)	0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00
Tracking adjustment coefficient 2	TRADJ2	8 real numbers	Tracking adjustment coefficient 2 Modify the conveyer speed to Vc + TRADJ2 * (Vc - Vp). Vc = Conveyer speed at the current sampling Vp = Conveyer speed at the previous sampling	0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00

16.2. Expansion serial interface Connector Pin Assignment

“Figure 16–1 Connector Arrangement” shows the connector arrangement of Expansion serial interfaces and “Table 16–2 Connectors: CNENC Pin Assignment” shows pin assignment of each connector.


CNENC
Encoder x 2CH
Expansion serial interface card
RTBUS
Connect to an option slot of a controller

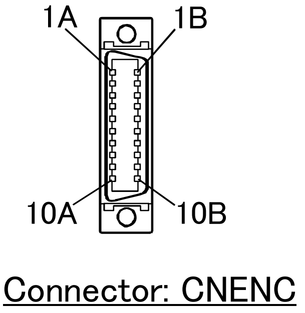


Figure 16–1 Connector Arrangement

Table 16–2 Connectors: CNENC Pin Assignment

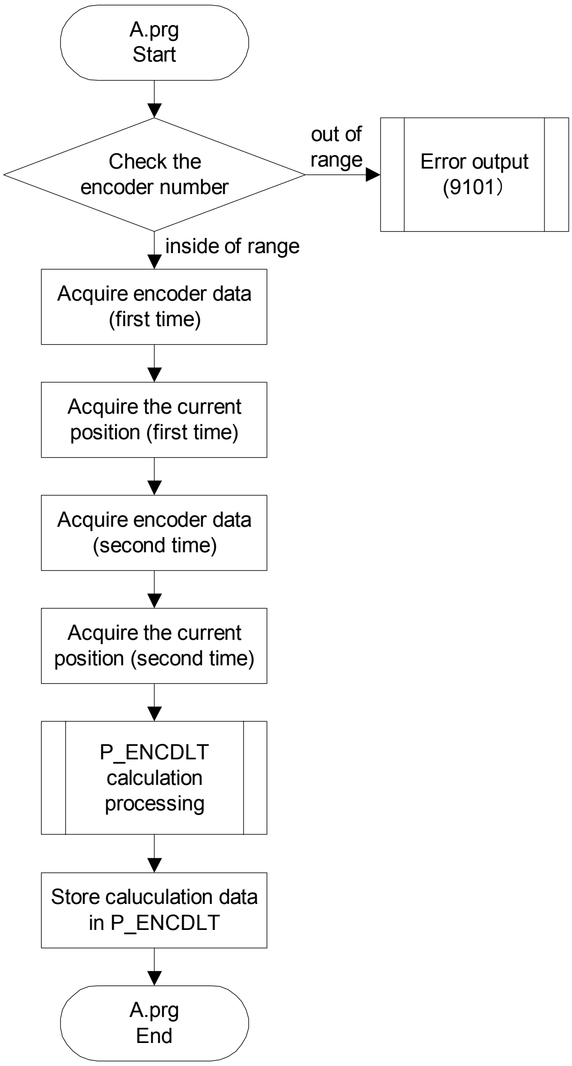
Pin NO.	Signal name	Explanation	Input/output	Remark
1A	SG	Control power supply 0 V	GND	
2A	LAH1	+ terminal of differential encoder A-phase signal	Input	CH1
3A	LBH1	+ terminal of differential encoder B-phase signal	Input	
4A	LZH1	+ terminal of differential encoder Z-phase signal	Input	
5A	SG	Control power supply 0 V	GND	
6A	LAH2	+ terminal of differential encoder A-phase signal	Input	CH2
7A	LBH2	+ terminal of differential encoder B-phase signal	Input	
8A	LAH2	+ terminal of differential encoder Z-phase signal	Input	
9A	-	Empty	–	
10A	-	Empty	–	
1B	SG	Control power supply 0 V	GND	
2B	LAL1	- terminal of differential encoder A-phase signal	Input	CH1
3B	LBL1	- terminal of differential encoder B-phase signal	Input	
4B	LZL1	- terminal of differential encoder Z-phase signal	Input	
5B	SG	Control power supply 0 V	GND	
6B	LAL2	- terminal of differential encoder A-phase signal	Input	CH2
7B	LBL2	- terminal of differential encoder B-phase signal	Input	
8B	LZL2	- terminal of differential encoder Z-phase signal	Input	
9B	-	Empty	–	
10B	-	Empty	–	

16.3. Chart of sample program

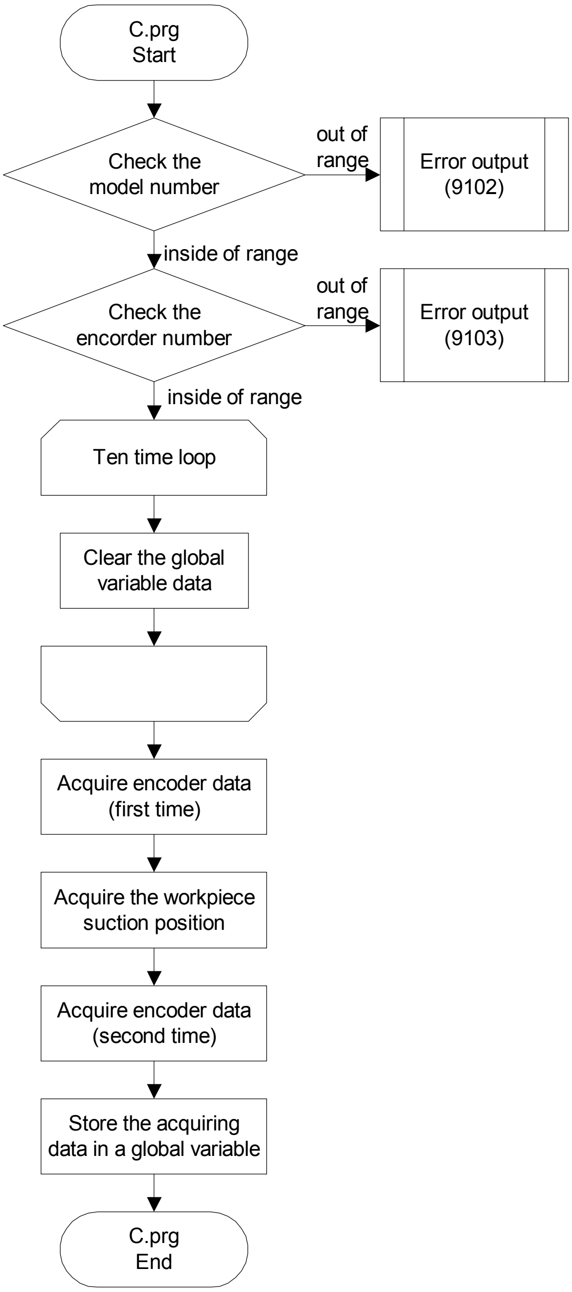
The chart of the sample program is shown below.

16.3.1. Conveyer tracking

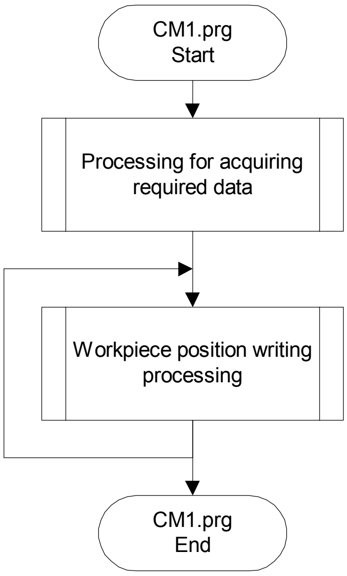
(1) A.prg

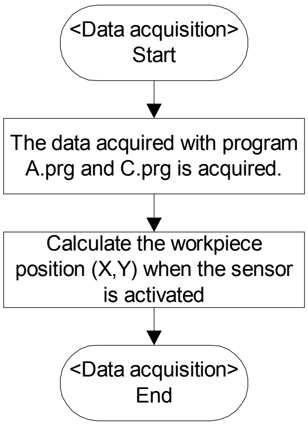

A.prg
Start
Check the encoder number
out of range
Error output (9101）
inside of range
Acquire encoder data (first time)
Acquire the current position (first time)
Acquire encoder data (second time)
Acquire the current position (second time)
P_ENCDLT calculation processing
Store caluculation data in P_ENCDLT
A.prg
End

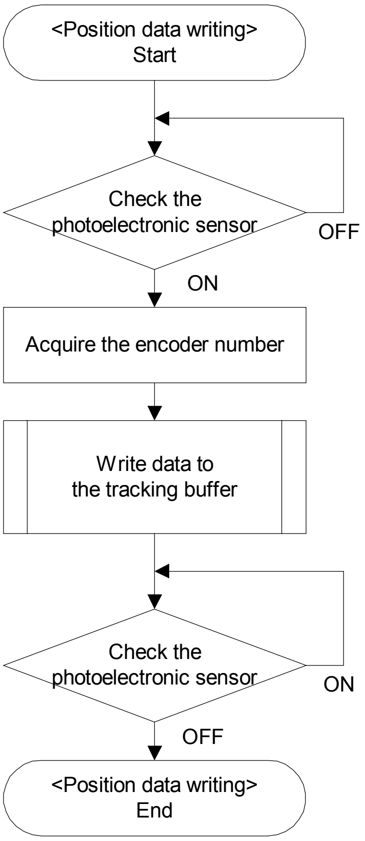
(2) C.prg


C.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
C.prg
End

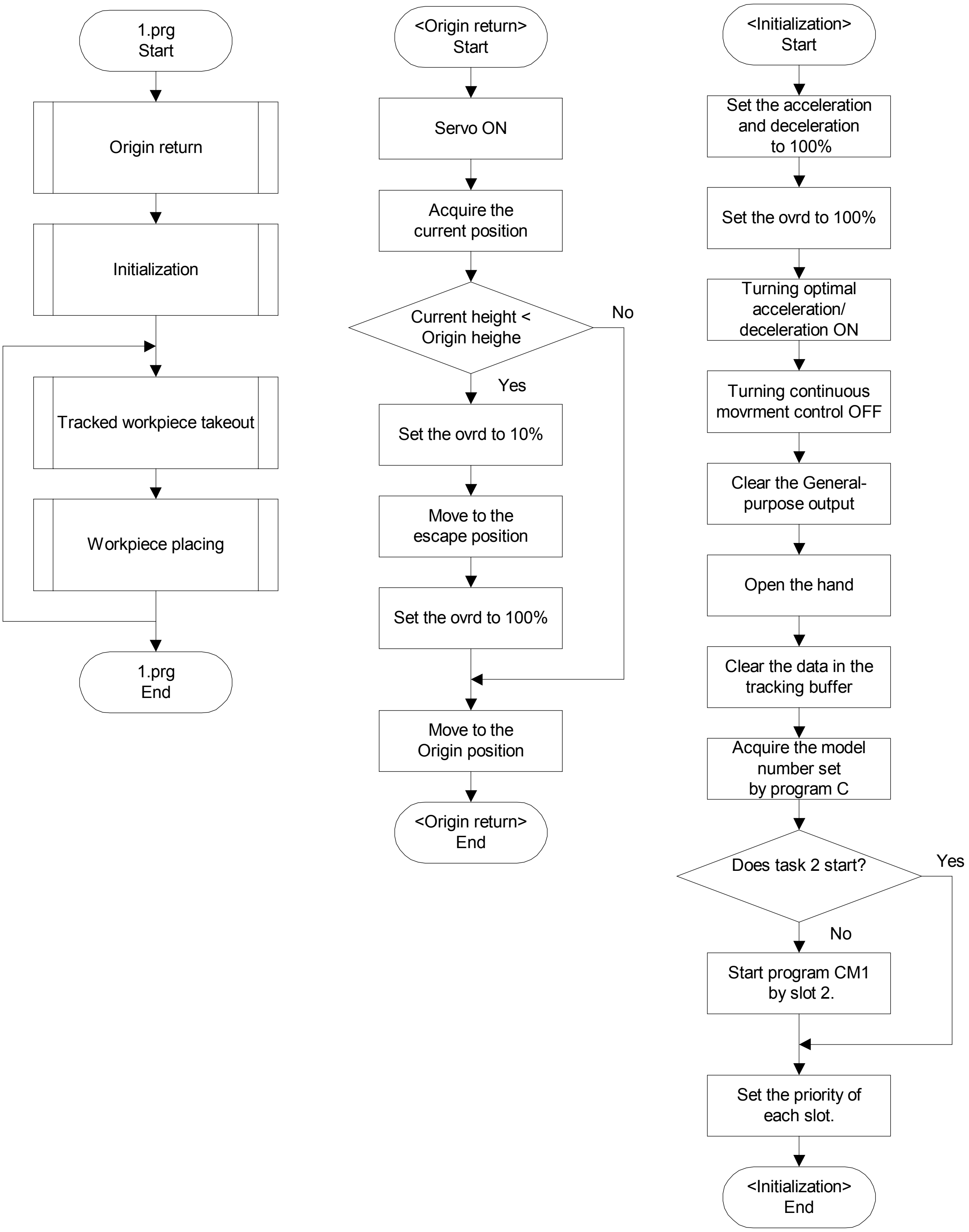
(3) CM1.prg

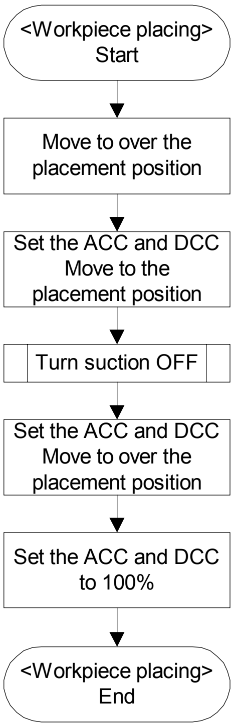

CM1.prg
Start
Processing for acquiring required data
Workpiece position writing processing
CM1.prg
End

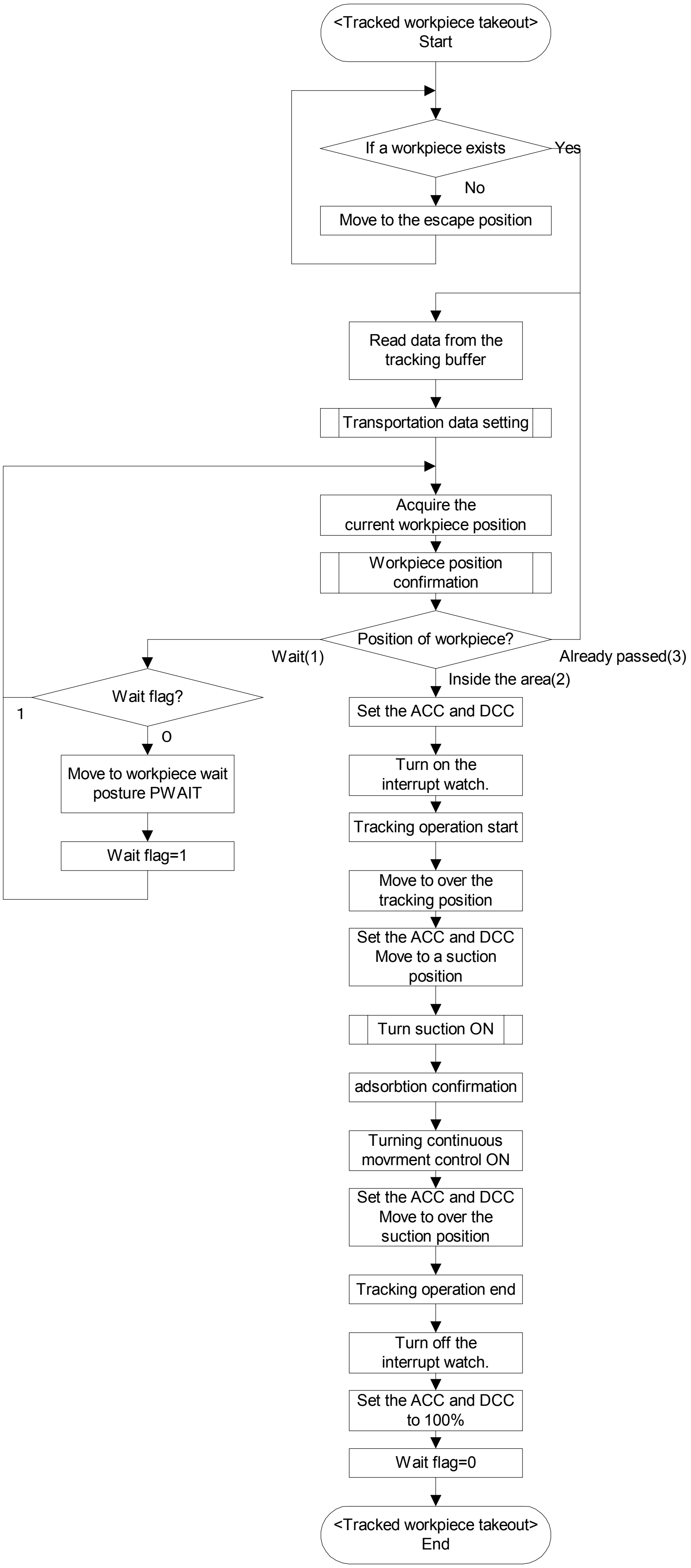

<Data acquisition>
Start
The data acquired with program A.prg and C.prg is acquired.
Calculate the workpiece position (X,Y) when the sensor is activated
<Data acquisition>
End

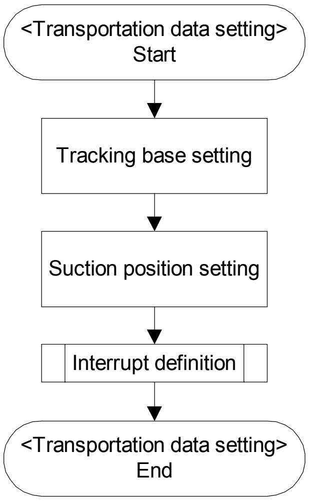

<Position data writing>
Start
Check the photoelectronic sensor
OFF
ON
Acquire the encoder number
Write data to the tracking buffer
Check the photoelectronic sensor
ON
OFF
<Position data writing>
End

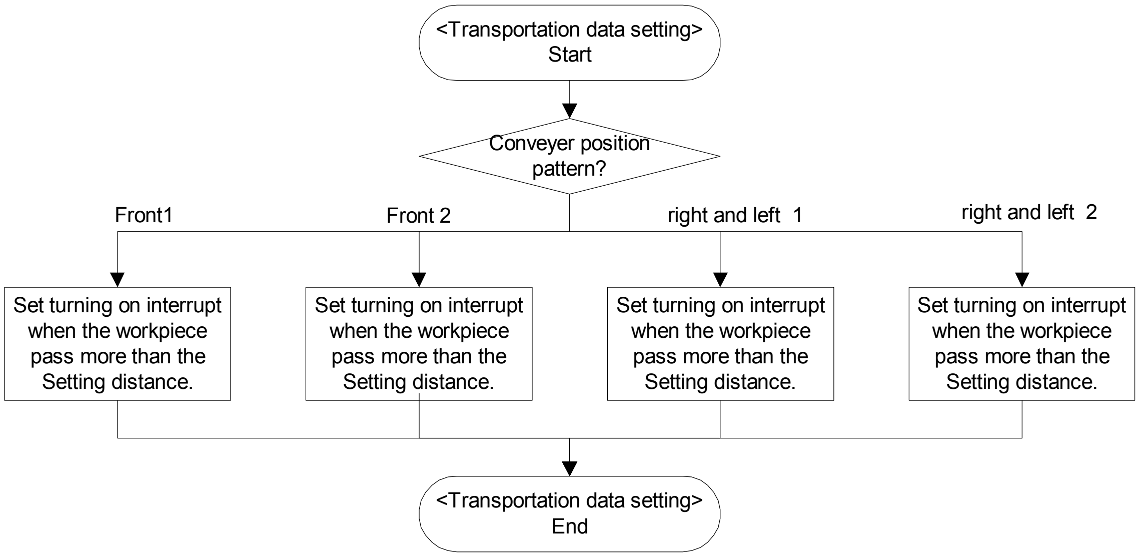
(4) 1.prg

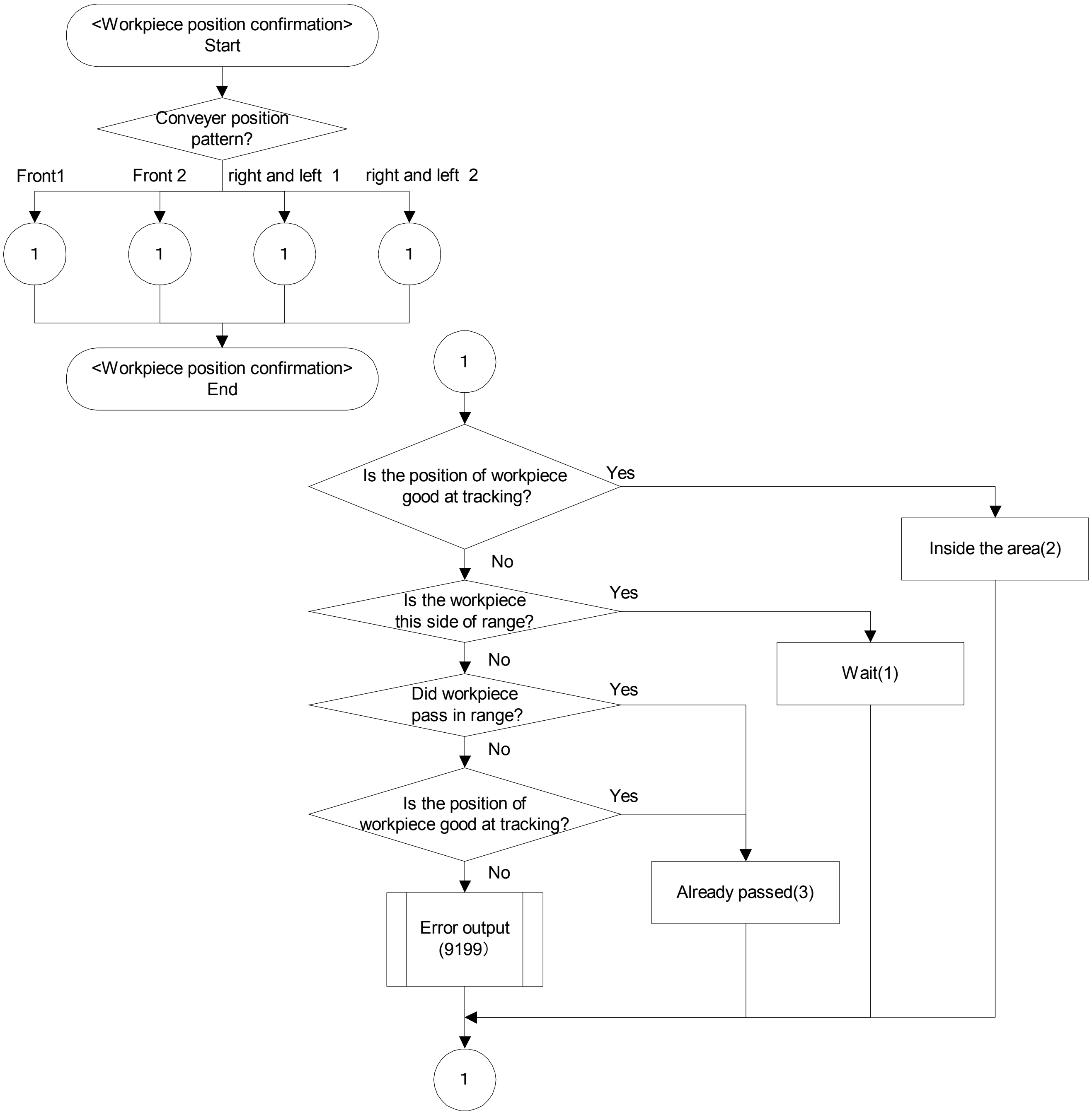

1.prg
Start
Origin return
Initialization
Tracked workpiece takeout
Workpiece placing
1.prg
End
<Origin return>
Start
Servo ON
Acquire the current position
Current height < Origin heighe
No
Yes
Set the ovrd to 10%
Move to the escape position
Set the ovrd to 100%
Move to the Origin position
<Origin return>
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

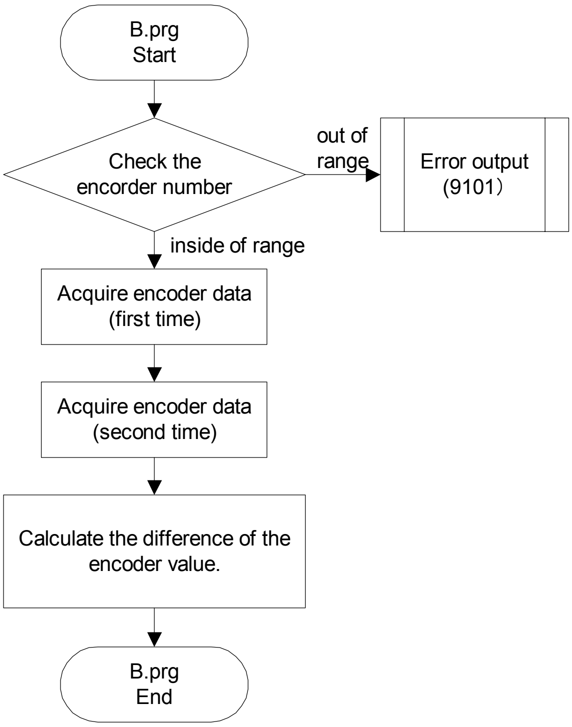

<Workpiece position confirmation>
Start
Conveyer position pattern?
Front1
Front 2
right and left 1
right and left 2
1
1
1
1
<Workpiece position confirmation>
End
1
Is the position of workpiece good at tracking?
Yes
Inside the area(2)
No
Is the workpiece this side of range?
Yes
Wait(1)
No
Did workpiece pass in range?
Yes
No
Is the position of workpiece good at tracking?
Yes
Already passed(3)
No
Error output (9199)
1

16.3.2. Vision Tracking

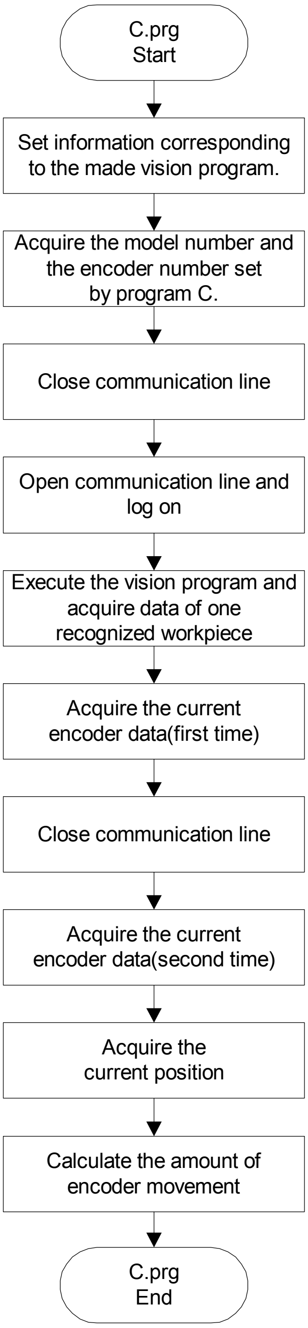
(1) A.prg

The same program as the conveyer tracking.

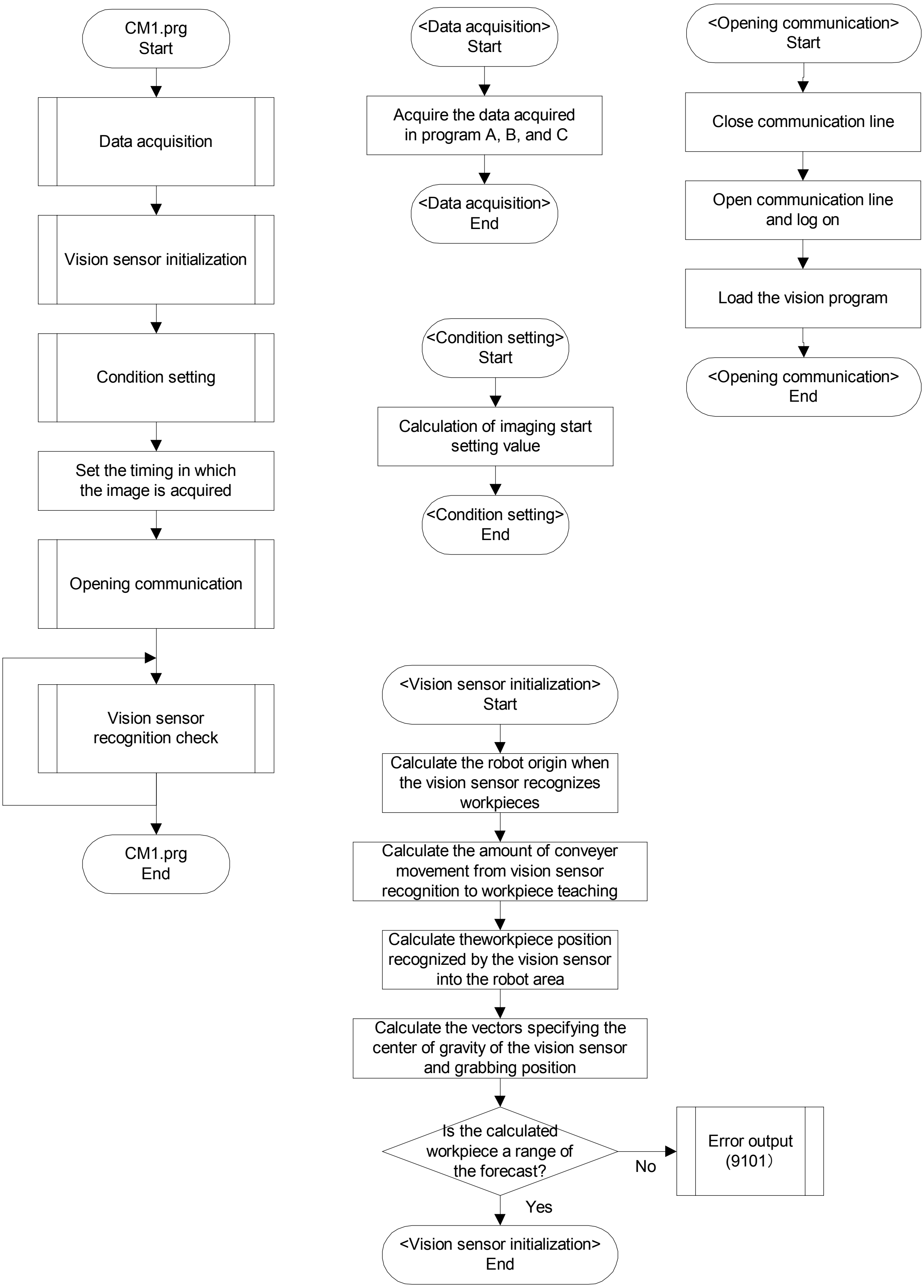
(2) B.prg

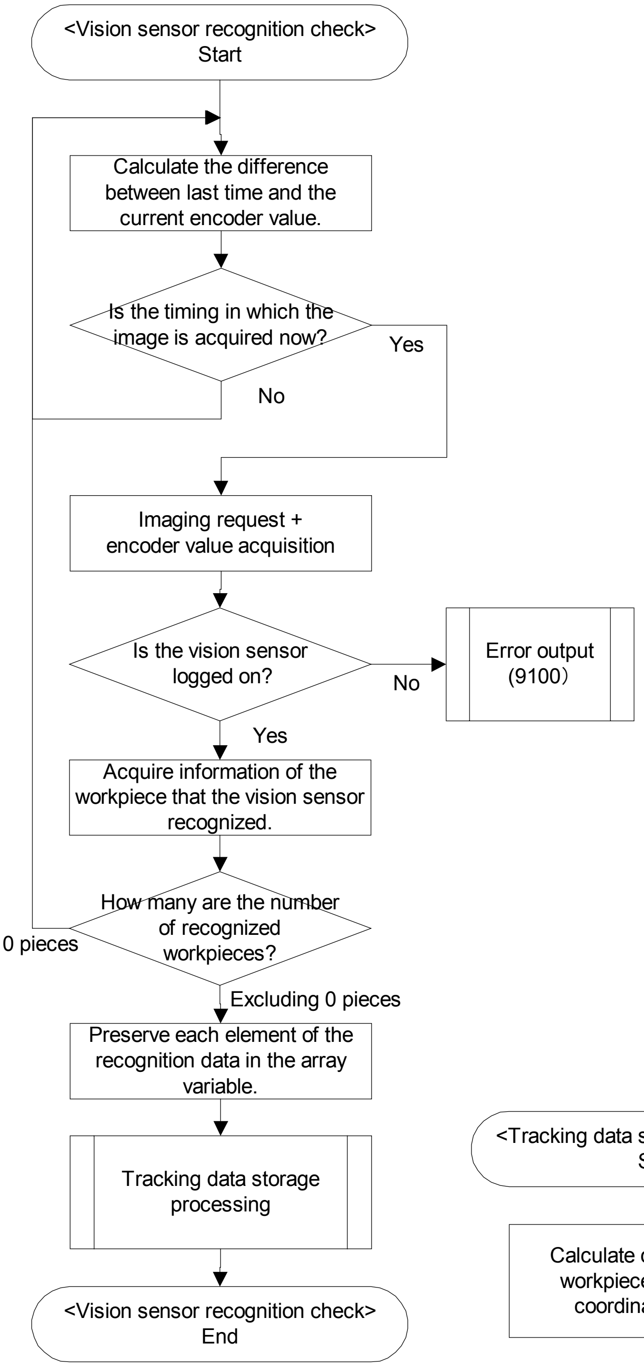


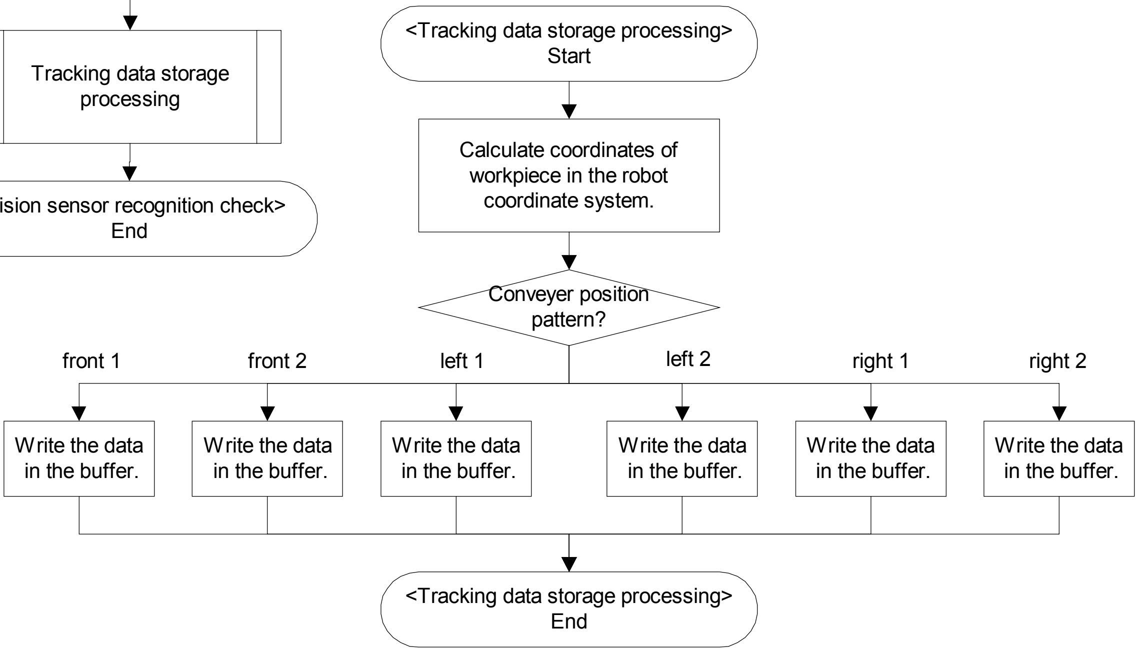
(3) C.prg


C.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
C.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


<Vision sensor recognition check>
Start
Calculate the difference between last time and the current encoder value.
Is the timing in which the image is acquired now?
Yes
No
Imaging request + encoder value acquisition
Is the vision sensor logged on?
No
Error output (9100)
Yes
Acquire information of the workpiece that the vision sensor recognized.
How many are the number of recognized workpieces?
0 pieces
Excluding 0 pieces
Preserve each element of the recognition data in the array variable.
Tracking data storage processing
<Vision sensor recognition check>
End


<Tracking data storage processing>
Start
Calculate coordinates of workpiece in the robot coordinate system.
Conveyer position pattern?
front 1
front 2
left 1
left 2
right 1
right 2
Write the data in the buffer.
Write the data in the buffer.
Write the data in the buffer.
Write the data in the buffer.
Write the data in the buffer.
Write the data in the buffer.
<Tracking data storage processing>
End

(5) 1.prg

The same program as the conveyer tracking.

16.4. Sample Programs

16.4.1. Conveyer Tracking

(1) A.Prg

```
10 '## Ver.A1 ########################################
20 '# Program for calibration between tracking robot and conveyer
30 '# Program type : A.prg
40 '# Date of creation/version : 2006.04.21 A1a
50 '# COPYRIGHT : MITSUBISHI ELECTRIC CORPORATION.
60 '###############################################
70 '(1) Register an encoder number to the X coordinate of the "PE" variable/
80 'Check the setting value
90    MECMAX=8                              'The maximum encoder number value (for checking)
100    IF PE.X<1 OR PE.X>MECMAX THEN ERROR 9101  'Encoder number out of range
110    MENCNO=PE.X                          'Acquire the encoder number
120 '(2) Attach a marking sticker on the conveyer upstream side/
130 '(3) Move the robot to the position right at the center of the attached sticker/
140    MX10EC1#=M_ENC(MENCNO)               'Acquire encoder data (first time)
150    PX10PS1=P_ZERO                        'Set all elements to ZERO
160    PX10PS1=P_FBC(1)                     'Acquire the current position (first time)
170 '(4) Raise the robot/
180 '(5) Move the sticker in the forward direction of the conveyer/
190 '(6) Move the robot to the position right at the center of the moved sticker/
200    MX10EC2#=M_ENC(MENCNO)               'Acquire encoder data (second time)
210    PX10PS2=P_ZERO                        'Set all elements to ZERO
220    PX10PS2=P_FBC(1)                     'Acquire the current position (second time)
230 '(7) Raise the robot/
240 '(8) Perform step operation until END/
250    GOSUB *S10ENC                        'P_ENCDLT calculation processing
260    P_ENCDLT(MENCNO)=PY10ENC              'Store data in P_ENCDLT
270 END
280 '
290 '##### Processing for obtaining P_ENCDLT #####
300     'MX10EC1: Encoder data 1
310     'MX10EC2: Encoder data 2
320     'PX10PS1: Position 1
330     'PX10PS2: Position 2
340     'PY10ENC: P_ENCDLT value
350 *S10ENC
360    M10ED#=MX10EC2#-MX10EC1#
370    IF M10ED#>800000000.0 THEN M10ED#=M10ED#-1000000000.0
380    IF M10ED#<-800000000.0 THEN M10ED#=M10ED#+1000000000.0
390    PY10ENC.X=(PX10PS2.X-PX10PS1.X)/M10ED#
400    PY10ENC.Y=(PX10PS2.Y-PX10PS1.Y)/M10ED#
410    PY10ENC.Z=(PX10PS2.Z-PX10PS1.Z)/M10ED#
420    PY10ENC.A=(PX10PS2.A-PX10PS1.A)/M10ED#
430    PY10ENC.B=(PX10PS2.B-PX10PS1.B)/M10ED#
440    PY10ENC.C=(PX10PS2.C-PX10PS1.C)/M10ED#
450    PY10ENC.L1=(PX10PS2.L1-PX10PS1.L1)/M10ED#
460    PY10ENC.L2=(PX10PS2.L2-PX10PS1.L2)/M10ED#
470 RETURN
480 '
490 'This program “computes how much a robot moves per 1 pulse and stores the result in P_ENCDLT."
PE=(1.000,0.000,0.000,0.000,0.000,0.000)
PX10PS1=(447.500,-383.610,328.600,-180.000,0.000,180.000,0.000,0.000)(7,0)
```

```
PX10PS2=(445.150,244.300,328.470,180.000,0.000,180.000,0.000,0.000)(7,0)
PY10ENC=(0.000,-0.020,0.000,-0.010,0.000,0.000,0.000,0.000)(0,0)
```

(2) C.Prg

```
10 '## Ver.A1 ##########################################
20 '# Conveyer tracking, workpiece suction position registration program
30 '# Program type : C.prg
40 '# Date of creation/version : 2006.04.21 A1a
50 '# COPYRIGHT : MITSUBISHI ELECTRIC CORPORATION.
60 '####################################################
70 '(1) Register a model number in the X coordinate of the "PRM1" variable/
80 '(2) Register an encoder number in the Y coordinate of the "PRM1" variable/
90 '(3) Register the number of the sensor that monitors workpieces in the Z coordinate of the "PRM1" variable /
100 'Check the conditions set in the "PRM1" variable
110     MWKMAX=10                              'The maximum model number value (for checking)
120     MECMAX=8                               'The maximum encoder number value (for checking)
130     MWKNO=PRM1.X                             'Acquire a model number
140     MENCNO=PRM1.Y                          'Acquire an encoder number
150     IF MWKNO<1 OR MWKNO>MWKMAX THEN ERROR 9102       'Model number out of range
160     IF MENCNO<1 OR MENCNO>MECMAX THEN ERROR 9103     'Encoder number out of range
170     FOR M1=1 TO 10                         'Clear the information
180        P_100(M1)=P_ZERO                    'A variable that stores workpiece positions
190        P_102(M1)=P_ZERO                    'A variable that stores operation conditions
200        M_101#(M1)=0                        'A variable that stores encoder value differences
210     NEXT M1
220 '(4) Move a workpiece to the position where the photoelectronic sensor is activated/
230     ME1#=M_ENC(MENCNO)                       'Acquire encoder data (first time)
240 '(5) Move a workpiece on the conveyer into the robot operation area/
250 '(6) Move the robot to the suction position/
260     ME2#=M_ENC(MENCNO)                       'Acquire encoder data (second time)
270     P_100(MWKNO)=P_FBC(1)                  'Acquire the workpiece suction position(current position)
280 '(7) Perform step operation until END/
290     MED#=ME2#-ME1#                         'Calculate the difference of the encoder value.
300     IF MED# > 800000000.0 THEN MED# = MED#-1000000000.0
310     IF MED# < -800000000.0 THEN MED# = MED#+1000000000.0
320 '
330     M_101#(MWKNO)=MED#                'Store the amount of encoder movement in a global variable
340     P_102(MWKNO).X=PRM1.Y             'Store encoder numbers in a global variable
350     P_102(MWKNO).Y=PRM1.Z               'Store the sensor number in a global variable
360 END
370 '
380 'This program is "the relation between the position at which the sensor is reacted and the position at which
390 'the robot absorbs workpieces.
```

(3) 1.Prg

```
10 '### Ver.A1 ###################################
20 '# Conveyer tracking, robot operation program
30 '# Program type : 1.prg
40 '# Date of creation/version : 2006.04.21 A1a
50 '# MITSUBISHI ELECTRIC CORPORATION.
60 '###############################################
70 '
80 '### Main processing ###
90 *S00MAIN
100     GOSUB *S90HOME                             'Origin return processing
110     GOSUB *S10INIT                            'Initialization processing
120 *LOOP
130     GOSUB *S20TRGET                           'Tracked workpiece takeout processing
140     GOSUB *S30WKPUT                           'Workpiece placing processing
150     GOTO *LOOP
160 END
170 '
180 '###   Initialization processing ###
190 *S10INIT
200 '/// Speed related ///
210     ACCEL 100,100                             'Acceleration/deceleration setting
220     OVRD 100                                  'Speed setting
230     LOADSET 1,1                               'Optimal acceleration/deceleration specification
240     OADL ON                                   'Turning optimal acceleration/deceleration ON
250     CNT 0
260     CLR 1
270     HOPEN 1
280 '/// Initial value setting ///
290     TRCLR 1                                   'Clear tracking buffer 1
300     MWAIT1=0                                  'Clear workpiece wait flag 1
310 '/// Multitask startup ///
320     M_09#=PWK.X                               'Model number specification
330     IF M_RUN(2)=0 THEN                        'Confirmation of conveyer 1 multitasking
340       XRUN 2,"CM1",1                          'Multitasking setting
350       WAIT M_RUN(2)=1
360     ENDIF
370     PRIORITY PRI.X,1
380     PRIORITY PRI.Y,2
390 RETURN
400 '
410 '### Tracked workpiece takeout processing ###
420 *S20TRGET
430 '/// Tracking buffer check ///
440 *LBFCHK
450     IF M_TRBFCT(1)>=1 THEN GOTO *LREAD        'If a workpiece exists
460     MOV P1 TYPE 0,0                           'Move to the pull-off location
470     MWAIT1=0
480     GOTO *LBFCHK
490 '/// Workpiece data acquisition ///
500 *LREAD
510     TRRD PBPOS,MBENC#,MBWK%,1,MBENCNO%         'Read data from the tracking buffer
520     GOSUB *S40DTSET                           'Transportation data setting
530 '/// Workpiece position confirmation ///
540 *LNEXT
550     PX50CUR=TRWCUR(MBENCNO%,PBPOS,MBENC#)  'Acquire the current workpiece position
560     MX50ST=PRNG.X                 'Start distance of the range where the robot can follow a workpiece
570     MX50ED=PRNG.Y                 'End distance of the range where the robot can follow a workpiece
580     MX50PAT=PTN.X                 'Conveyer position pattern number
590     GOSUB *S50WKPOS             'Workpiece position confirmation processing
600     IF MY50STS=3 THEN GOTO *LBFCHK          'Already passed. Go to the next workpiece
610     IF MY50STS=2 THEN GOTO *LTRST          'Operable: start tracking
620     IF MWAIT=1 THEN GOTO *LNEXT            'Wait for incoming workpieces
630 '/// To standby position ///
```

```
640     PWAIT=P1                                    'Change to workpiece wait posture
650     SELECT PTN.X                                  'Conveyer position pattern number
660     CASE 1 TO 2                                 'When the conveyer is the front of the robot
670       PWAIT.X=PX50CUR.X                         'X coordinates of the robot are matched to workpiece.
680     CASE 3 TO 6
690       PWAIT.Y=PX50CUR.Y                         'Y coordinates of the robot are matched to workpiece.
700     END SELECT
710     PWAIT.Z=PX50CUR.Z+PUP1.X
720     PWAIT.C=PX50CUR.C
730     MOV PWAIT TYPE 0,0                          'Move to workpiece wait posture PWAIT
740     MWAIT1=1                                    'Set workpiece wait flag
750     GOTO *LNEXT
760 '/// Start tracking operation ///
770 *LTRST
780     ACCEL PAC1.X,PAC1.Y
790     CNT 1,0,0
800     ACT 1=1                                     'Monitor the robot following workpieces too far
810     TRK ON,PBPOS,MBENC#,PTBASE,MBENCNO%    'Tracking operation start setting
820     MOV PGT,PUP1.Y TYPE 0,0                     'Move to tracking midair position
830     ACCEL PAC2.X,PAC2.Y
840     MOV PGT TYPE 0,0                              'Move to a suction position
850     GOSUB *S85CLOSE                             'Turn suction ON
860     MX80ENA=PHND.X                              'Check instruction
870     MX80SIG=PHND.Y                                'Check signal number
880     MX80SEC=PDLY1.X                             'Check second number(s)
890     GOSUB *S80CWON                              'adsorbtion confirmation
900     CNT 1
910     ACCEL PAC3.X,PAC3.Y
920     MOV PGT,PUP1.Z TYPE 0,0                     'Move to tracking midair position
930     TRK OFF                                     'Tracking operation end setting
940     ACT 1=0
950     ACCEL 100,100
960     MWAIT = 0
970 RETURN
980 '
990 '### Workpiece placing processing ###
1000 *S30WKPUT
1010     ACCEL PAC11.X,PAC11.Y
1020     MOV PPT,PUP2.Y TYPE 0,0                    'Move to over the placement position
1030     ACCEL PAC12.X,PAC12.Y
1040     CNT 1,0,0
1050     MOV PPT TYPE 0,0                             'Move to the placement position
1060     GOSUB *S86OPEN                             'Turn suction OFF
1070     MX81ENA=PHND.X                             'Check instruction
1080     MX81SIG=PHND.Z                               'Check signal number
1090     MX81SEC=PDLY2.X                            'Check second number(s)
1100     GOSUB *S81CWOFF                            'Release confirmation
1110     CNT 1
1120     ACCEL PAC13.X,PAC13.Y
1130     MOV PPT,PUP2.Z TYPE 0,0                    'Move to over the placement position
1140     ACCEL 100,100
1150 RETURN
1160 '
1170 '### Transportation data setting processing ###
1180 *S40DTSET
1190     PTBASE=P_100(PWK.X)                        'Create reference position
1200     TRBASE PTBASE,MBENCNO%                       'Tracking base setting
1210     PGT=PTBASE*POFSET                          'Suction position setting
1220     GOSUB *S46ACSET                            'Interrupt definition
1230 RETURN
1240 '
1250 '### Interrupt definition processing 1 ###
1260 *S46ACSET
1270     SELECT PTN.X                                 'Conveyer position pattern number
1280     CASE 1 'Front right -> left
1290       MSTP1=PRNG.Z                                            'Following stop distance
```

```
1300      DEF ACT 1,P_FBC(1).Y>MSTP1 GOTO *S91STOP    'To *S91STOP if followed far long
1310      BREAK
1320    CASE 2 'Front left -> right
1330      MSTP1=-PRNG.Z
1340      DEF ACT 1,P_FBC(1).Y<MSTP1 GOTO *S91STOP
1350      BREAK
1360    CASE 3 'Left side rear -> front
1370    CASE 5 'Right side rear -> front
1380      MSTP1=PRNG.Z
1390      DEF ACT 1,P_FBC(1).X>MSTP1 GOTO *S91STOP
1400      BREAK
1410    CASE 4 'Left side front -> rear
1420    CASE 6 'Right side front -> rear
1430      MSTP1=-PRNG.Z
1440      DEF ACT 1,P_FBC(1).X<MSTP1 GOTO *S91STOP
1450      BREAK
1460    END SELECT
1470 RETURN
1480 '
1490 '### Workpiece position confirmation processing ###
1500     'PX50CUR:Current workpiece position
1510     'MX50ST:Tracking start range
1520     'MX50ED:Tracking end range
1530     'MX50PAT:Conveyer position pattern number
1540     'MY50STS:Result (1: Wait/2: Start tracking/3: Next workpiece)
1550 *S50WKPOS
1560    MY50STS=0                               'Clear return value
1570    SELECT MX50PAT                          'Conveyer pattern
1580    CASE 1 'Front right -> left
1590      M50STT=-MX50ST                        'The start side has a negative value
1600      M50END=MX50ED
1610      IF POSCQ(PX50CUR)=1 AND PX50CUR.Y>=M50STT AND PX50CUR.Y<=M50END THEN
1620        MY50STS=2                           'Tracking possible
1630      ELSE 'If tracking not possible
1640        IF PX50CUR.Y<0 THEN MY50STS=1       'Wait
1650        IF PX50CUR.Y>M50END THEN MY50STS=3  'Move onto the next workpiece
1660          IF  POSCQ(PX50CUR)=0  AND  PX50CUR.Y>=M50STT  AND  PX50CUR.Y<=M50END  THEN
MY50STS=3 'Outside the movement range
1670      ENDIF
1680      BREAK
1690    CASE 2 'Front left -> right
1700      M50STT=MX50ST
1710      M50END=-MX50ED                        'The end side has a negative value
1720      IF POSCQ(PX50CUR)=1 AND PX50CUR.Y<=M50STT AND PX50CUR.Y>=M50END THEN
1730        MY50STS=2                           'Tracking possible
1740      ELSE 'If tracking not possible
1750        IF PX50CUR.Y>0 THEN MY50STS=1       'Wait
1760        IF PX50CUR.Y<0 THEN MY50STS=3       'Move onto the next workpiece
1770          IF  POSCQ(PX50CUR)=0  AND  PX50CUR.Y<=M50STT  AND  PX50CUR.Y>=M50END  THEN
MY50STS=3 'Outside the movement range
1780      ENDIF
1790      BREAK
1800    CASE 3 'Left side rear -> front
1810    CASE 5 'Right side rear -> front
1820      M50STT=-MX50ST                         'The start side has a negative value
1830      M50END=MX50ED
1840      IF POSCQ(PX50CUR)=1 AND PX50CUR.X>=M50STT AND PX50CUR.X<=M50END THEN
1850        MY50STS=2                           'Tracking possible
1860      ELSE 'If tracking not possible
1870        IF PX50CUR.X<0 THEN MY50STS=1       'Wait
1880        IF PX50CUR.X>0 THEN MY50STS=3       'Move onto the next workpiece
1890          IF  POSCQ(PX50CUR)=0  AND  PX50CUR.X>=M50STT  AND  PX50CUR.X<=M50END  THEN
MY50STS=3 'Outside the movement range
1900      ENDIF
1910      BREAK
1920    CASE 4 'Left side front -> rear
```

```
1930    CASE 6 'Right side front -> rear
1940       M50STT=MX50ST
1950       M50END=-MX50ED                          'The end side has a negative value
1960       IF POSCQ(PX50CUR)=1 AND PX50CUR.X<=M50STT AND PX50CUR.X>=M50END THEN
1970          MY50STS=2                            'Tracking possible
1980       ELSE 'If tracking not possible
1990          IF PX50CUR.X>0 THEN MY50STS=1        'Wait
2000          IF PX50CUR.X<0 THEN MY50STS=3        'Move onto the next workpiece
2010             IF  POSCQ(PX50CUR)=0  AND  PX50CUR.X<=M50STT  AND  PX50CUR.X>=M50END  THEN
MY50STS=3 'Outside the movement range
2020       ENDIF
2030       BREAK
2040    END SELECT
2050    IF MY50STS=0 THEN ERROR 9199               'Program modification required
2060 RETURN
2070 '
2080 '### Origin return processing ###
2090 *S90HOME
2100    SERVO ON                                   'Servo ON
2110    P90CURR=P_FBC(1)                           'Acquire the current position
2120    IF P90CURR.Z<P1.Z THEN                     'If the current height is below the origin
2130       OVRD 10
2140       P90ESC=P90CURR                            'Create an escape position
2150       P90ESC.Z=P1.Z
2160       MVS P90ESC                              'Move to the escape position
2170       OVRD 100
2180    ENDIF
2190    MOV P1                                     'Move to the origin
2200 RETURN
2210 '
2220 '### Tracking interruption processing ###
2230 *S91STOP
2240    ACT 1=0
2250    TRK OFF
2260    GOSUB *S86OPEN                             'Release suction
2270    P91P=P_FBC(1)                              'Acquire the current position
2280    P91P.Z=P1.Z
2290    MVS P91P TYPE 0,0                          'Raise
2300    MOV P1                                     'Return to the origin once
2310    GOTO *LBFCHK
2320 '
2330 '##### Suction of substrates #####
2340 *S85CLOSE
2350    HCLOSE 1                                   'Turn suction ON
2360 RETURN
2370 '##### Suction/release of substrates #####
2380 *S86OPEN
2390    HOPEN 1                                    'Turn suction OFF
2400 RETURN
2410 '
2420 '##### Turning on the signal is waited for #####
2430    'MX80ENA:ENABLE/DISABLE of check(1/0)
2440    'MX80SIG:Check signal number
2450    'MX80SEC:Check second number(S)
2460    'MY80SKP:OK/TIMEOUT(1/0)
2470 *S80CWON
2480    IF MX80ENA=1 THEN                          'If the signal check is ENABLE
2490       M_TIMER(1)=0
2500       MY80SKP=0
2510       MX80SEC=MX80SEC * 1000                  'Second -> Millisecond
2520 *L80LOP
2530       IF (M_TIMER(1)>MX80SEC) OR (MY80SKP<>0) THEN *L80END
2540          IF M_IN(MX80SIG)=1 THEN MY80SKP=1    'If the signal specified is turned on
2550       GOTO *L80LOP
2560    ELSE                                       'If the signal check is DISABLE
2570       DLY MX80SEC                             'Wait at the specified check time
```

```
2580        MY80SKP=1                                       'OK
2590     ENDIF
2600 *L80END
2610 RETURN
2620 '
2630 '##### Turning off the signal is waited for #####
2640     'MX81ENA:ENABLE/DISABLE of check(1/0)
2650     'MX81SIG:Check signal number
2660     'MX81SEC:Check second number(S)
2670     'MY81SKP:OK/TIMEOUT(1/0)
2680 *S81CWOFF
2690     IF MX81ENA=1 THEN                                  'If the signal check is ENABLE
2700        M_TIMER(1)=0
2710        MY81SKP=0
2720        MX81SEC=MX81SEC * 1000                           'Second -> Millisecond
2730 *L81LOP
2740        IF (M_TIMER(1)>MX81SEC) OR (MY81SKP<>0) THEN *L81END
2750           IF M_IN(MX81SIG)=0 THEN MY81SKP=1            'If the signal specified is turned off
2760        GOTO *L81LOP
2770     ELSE                                               'If the signal check is DISABLE
2780        DLY MX80SEC                                     'Wait at the specified check time
2790        MY81SKP=1 'OK
2800     ENDIF
2810 *L81END
2820 RETURN
P1=(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)
PAC2=(100.000,100.000,0.000,0.000,0.000,0.000,0.000,0.000)(0,0)
PAC3=(100.000,100.000,0.000,0.000,0.000,0.000,0.000,0.000)(0,0)
PDLY1=(1.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000)(0,0)
PDLY2=(1.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000)(0,0)
PHND=(0.000,900.000,900.000,0.000,0.000,0.000,0.000,0.000)(0,0)
PPT=(0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000)(0,0)
PRI=(1.000,1.000,0.000,0.000,0.000,0.000,0.000,0.000)(0,0)
PRNG=(300.000,200.000,0.000,0.000,0.000,0.000,0.000,0.000)(0,0)
PTN=(1.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000)(0,0)
PUP1=(50.000,-50.000,-70.000,0.000,0.000,0.000,0.000,0.000)(0,0)
PUP2=(-50.000,-50.000,0.000,0.000,0.000,0.000,0.000,0.000)(0,0)
PWK=(1.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000)(0,0)
POFSET=(0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000)(0,0)
```

(4) CM1.Prg

```
10 '## Ver.A1 #####################################
20 '# Conveyer tracking, sensor monitoring program
30 '# Program type : CM1.prg
40 '# Date of creation/version : 2006.04.21 A1a
50 '# COPYRIGHT : MITSUBISHI ELECTRIC CORPORATION.
60 '################################################
70 '
80 '##### Main processing #####
90 *S00MAIN
100     GOSUB *S10DTGET                         'Processing for acquiring required data
110 *LOOP
120     GOSUB *S20WRITE                         'Workpiece position writing processing
130     GOTO *LOOP
140 END
150 '##### Data acquisition processing #####
160 *S10DTGET
170 'Acquire the suction position, amount of encoder movement and encoder number set with program C
180     MWKNO=M_09#                              'Acquire model number
190     M10ED#=M_101#(MWKNO)                     'Amount of encoder movement
200     MENCNO=P_102(MWKNO).X                   'Encoder number
210     MSNS=P_102(MWKNO).Y                     'Sensor number
220 'Calculate the workpiece position (X,Y) when the sensor is activated
230     PWPOS=P_100(MWKNO)-P_ENCDLT(MENCNO)*M10ED#
240 RETURN
250 '##### Position data writing processing #####
260 *S20WRITE
270     IF M_IN(MSNS)=0 THEN GOTO *S20WRITE    'Wait for a workpiece to activate the sensor
280     MENC#=M_ENC(MENCNO)                      'Encoder number
290     TRWRT PWPOS,MENC#,MWKNO,1,MENCNO        'Write data the tracking buffer
300 *L20WAIT
310     IF M_IN(MSNS)=1 THEN GOTO *L20WAIT
320 RETURN
```

16.4.2. Vision Tracking

(1) A.Prg

The same program as the conveyer tracking.

(2) B.Prg

```
10 '### Ver. A1 #########################################
20 '# Network vision tracking, calibration between robot and vision sensor
30 '# Program type          : B.prg
40 '# Date of creation   : 2006.04.21 A1a
50 '# COPYRIGHT : MITSUBISHI ELECTRIC CORPORATION.
60 '#####################################################
70 '(1) Register an encoder number to the X coordinate of the “PE” variable/
80 'Check the setting value
90     MECMAX=8                                    'The maximum encoder number value (for checking)
100     IF PE.X<1 OR PE.X>MECMAX THEN ERROR 9101 'Encoder number out of range
110     MENCNO=PE.X                                'Acquire the encoder number
120 '(2) Attach 4 stickers within the vision sensor recognition area/
130 '(3) Check that the sticker positions are correct by looking at vision images/
140 '(4) Make setting for 4 points in the calibration setting screen of MELFA-Vision/
150     ME1#=M_ENC(MENCNO)                         'Acquire encoder data (first time)
160 '(5) Move the stickers forward until they are within the robot operation area/
170 '(6) Move the robot hand to the position right at the center of sticker 1/
180 '(7) Acquire the robot position for the first point with MELFA-Vision/
190 '(8) Move the robot hand to the position right at the center of sticker 2/
200 '(9) Acquire the robot position for the second point with MELFA-Vision/
210 '(10) Move the robot hand to the position right at the center of sticker 3/
220 '(11) Acquire the robot position for the third point with MELFA-Vision/
230 '(12) Move the robot hand to the position right at the center of sticker 4/
240 '(13) Acquire the robot position for the fourth point with MELFA-Vision/
250 '(14) Raise the robot arm/
260     ME2#=M_ENC(MENCNO)                        'Acquire encoder data (second time)
270     MED#=ME1#-ME2#                             'Calculate the difference of the encoder value.
280     IF MED# > 800000000.0 THEN MED# = MED#-1000000000.0
290     IF MED# < -800000000.0 THEN MED# = MED#+1000000000.0
300     M_100#(MENCNO)=MED#
310 '(15) Enter a comment describing the calibration data and store it using MELFA-Vision/
320 END
PE=(1.000,0.000,0.000,0.000,0.000,0.000)
```

(3) C.Prg

```
10 '### Ver.A1 #########################################
20 '# Network vision tracking, workpiece suction position registration program
30 '# Program type                    : C.prg
40 '# Date of creation/version   : 2006.04.21 A1a
50 '# COPYRIGHT : MITSUBISHI ELECTRIC CORPORATION.
60 '####################################################
70 '(1) Store a model number in the X coordinate of the "PRM1" variable/
80 '(2) Store an encoder number in the Y coordinate of the "PRM1" variable/
90 '(3) Check live images and register the length in the movement direction to the X coordinate of the "PRM2" variable/
100 '(4) Store the workpiece length in the Y coordinate of the "PRM2" variable/
110 '(5) Enter the COM port number to be opened for communication after "CCOM$=" in the following line/
120     CCOM$="COM2:"           'Set the number of the port to be opened
130 '(6) Enter the vision program name after "CPRG$=" in the following line/
140     CPRG$="TRK.JOB"          'Set the vision program name
150 '(7) Enter the cell in which the recognized quantity is stored after "CKOSU$=" in the following line/
160     CKOSU$="E76"             'Set the cell in which the recognized quantity is stored
170 '(8) Enter the start of the area where recognition results are stored after "CSTT$=" in the following line/
180     CSTT$="J81"              'Set the start cell where recognition result data is stored
190 '(9) Enter the end cell of the area where recognition results are stored after "CEND$=" in the following line/
200     CEND$="L81"               'Set the end cell where recognition result data is stored
210 '(10) Place workpieces to be tracked in locations recognizable by the vision sensor/
220 '(11) Place the vision sensor in the "online" status/
230 '(12) Press [MENU] of T/B to automatically run program C. When the program stops, open it with T/B/
240     MWKNO=PRM1.X          'Acquire the model number
250     MENCNO=PRM1.Y        'Acquire the encoder number
260 'Establish a communication line with the vision sensor via the opened port
270     NVCLOSE                   'Close communication line
280     NVOPEN CCOM$ AS #1   'Open communication line and log on
290     WAIT M_NVOPEN(1)=1    'Wait to log on to the vision sensor
300     NVPST #1,CPRG$,CKOSU$,CSTT$,CEND$,0   'Acquire data of one recognized workpiece
310     P_101(MWKNO)=P_NVS1(1)   'Acquire data of the first recognized workpiece
320     ME1#=M_ENC(MENCNO)        'Acquire encoder data 1
330     NVCLOSE #1
340 HLT
350 '(13) Move a workpiece on the conveyer until it gets within the robot operation area/
360 '(14) Move the robot to the suction position/
370     ME2#=M_ENC(MENCNO)        'Acquire encoder data 2
380     P_100(MWKNO)=P_FBC(1)   'Acquire position 1
390 '(15) Perform step operation until END/
400     MED#=ME2#-ME1#                'Calculate the amount of encoder movement
410     IF MED# > 800000000.0 THEN MED# = MED#-1000000000.0
420     IF MED# < -800000000.0 THEN MED# = MED#+1000000000.0
430     M_101#(MWKNO)=MED#       'Amount of encoder movement
440     P_102(MWKNO)=PRM1           'Encoder number
450     P_103(MWKNO)=PRM2           'Image size and workpiece size
460     C_100$(MWKNO)=CCOM$       'COM port number
470     C_101$(MWKNO)=CPRG$       'Vision program name
480     C_102$(MWKNO)=CKOSU$     'Recognized quantity cell
490     C_103$(MWKNO)=CSTT$        'Start cell
500     C_104$(MWKNO)=CEND$       'End cell
510 END
520 '
530 'This program is "the relation between the workpiece position recognized by the network vision sensor and
540 ' the position at which the robot suctions workpieces.
PRM1=(1.000,1.000,0.000,0.000,0.000,0.000)
PRM2=(170.000,30.000,0.000,0.000,0.000,0.000)
```

(4) 1.Prg

The same program as the conveyer tracking.

(5) CM1.Prg

```
10 '### Ver.A1 #################################
20 '# Conveyer tracking, communication processing between robot and vision sensor
30 '# Program type          : VS communication program
40 '# Date of creation/version : 2006.04.21 A1a
50 '# COPYRIGHT : MITSUBISHI ELECTRIC CORPORATION.
60 '############################################
70    DIM MX(4),MY(4),MT(4)            'X/Y/C/correlation value/model/buffer
80    ""DIM MTR(8)                     'Encoder value
90 '
100 '##### Main processing #####
110 *S00MAIN
120    GOSUB *S10DTGET                 'Data acquisition processing
130    GOSUB *S20VSINI                'VS initialization processing
140    GOSUB *S30CONST                'Condition setting
150 '
160    MEP# = M_ENC(MENCNO)+MEI#+100
170    GOSUB *S70VOPEN                'Vision sensor line open + vision program load processing
180 *L00_00
190    GOSUB *S40CHKS                  'VS recognition check processing
200    GOTO *L00_00
210 END
220 '
230 '##### Data acquisition processing #####
240 *S10DTGET
250    MWKNO=M_09#                    'Model number
260    MENCNO=P_102(MWKNO).Y'Encoder number
270    MVSL=P_103(MWKNO).X             'VS screen size longitudinal distance
280    MWKL=P_103(MWKNO).Y            'Workpiece size longitudinal distance
290 '
300    PTEACH=P_100(MWKNO)             'Position taught to the robot
310    PVSWRK=P_101(MWKNO)            'Position recognized by VS
320    CCOM$=C_100$(MWKNO)             'COM port number
330    CPRG$=C_101$(MWKNO)             'Vision program name
340    CKOSU$=C_102$(MWKNO)            'Recognized quantity cell
350    CSTT$=C_103$(MWKNO)             'Start cell
360    CEND$=C_104$(MWKNO)             'End cell
370 RETURN
380 '
390 '##### Opening communication line #####
400 *S70VOPEN
410    NVCLOSE                         'Close communication line
420    NVOPEN CCOM$ AS #1             'Open communication line and log on
430    WAIT M_NVOPEN(1)=1              'Wait for line connection
440    NVLOAD #1,CPRG$                 'Load the vision program
450 RETURN
460 '
470 '##### VS initialization processing #####
480 *S20VSINI
490 'Move from the robot coordinate axis to the robot origin when the vision sensor recognizes workpieces
500    MED1#=M_100#(MENCNO)            'Amount of conveyer movement at calibration between vision sensor and
robot
510    PRBORG=P_ENCDLT(MENCNO)*MED1#    'Robot origin when the vision sensor recognizes workpieces
520 'Return a workpiece recognized by the vision sensor to the position taught to the robot
530    MED2#=M_101#(MWKNO)             'Amount of conveyer movement from vision sensor recognition to
workpiece teaching
540    PBACK=P_ENCDLT(MENCNO)*MED2#
550 'Calculate the position of the workpiece that the vision sensor in the robot area recognized.
560    PWKPOS=PRBORG+PVSWRK+PBACK      'Workpiece position recognized by the vision sensor into the
```

```
robot area
570     PVTR=(P_ZERO/PWKPOS)*PTEACH          'Vectors specifying the center of gravity of the vision sensor
and grabbing position
580      IF  PVTR.X<-PCHK.X  OR  PVTR.X>PCHK.X  THEN  ERROR  9110  'The  calculation  result  is  greatly
different from the theory value.
590     IF PVTR.Y<-PCHK.Y OR PVTR.Y>PCHK.Y THEN ERROR 9110
600 RETURN
610 '
620 '##### Condition setting #####
630 *S30CONST
640     MDX = P_ENCDLT(MENCNO).X          'Amount of movement per pulse (X)
650     MDY = P_ENCDLT(MENCNO).Y          'Amount of movement per pulse (Y)
660     MDZ = P_ENCDLT(MENCNO).Z        'Amount of movement per pulse (Z)
670     MD = SQR(MDX^2+MDY^2+MDZ^2)     'Calculation of the amount of movement per pulse
680     MEI#=ABS((MVSL-MWKL)/MD)          'Calculation of imaging start setting value
690 RETURN
700 '
710 '##### VS recognition check processing #####
720 *S40CHKS
730 *LVSCMD
740 *LWAIT
750     MEC# = M_ENC(MENCNO)
760     MEM#=MEC#-MEP#       'Subtract  the  previous  encoder  pulse  value  from  the  current  position  of  the
encoder
770     IF MEM# > 800000000.0 THEN MEM# = MEM#-1000000000.0
780     IF MEM# < -800000000.0 THEN MEM# = MEM#+1000000000.0
790     IF ABS(MEM#) > MEI# GOTO *LVSTRG  'Comparison between the amount of encoder movement and
the camera startup setting value
800     DLY 0.01
810     GOTO *LWAIT
820 *LVSTRG
830     MEP#=MEC#                          'Set the encoder pulse current position to the previous value
840      NVTRG  #1,  5,  MTR1#,MTR2#,MTR3#,MTR4#,MTR5#,MTR6#,MTR7#,MTR8#    'Imaging  request  +
encoder value acquisition
850 'Acquisition of recognition data
860     IF M_NVOPEN(1)<>1   THEN ERROR 9100  'Communication error
870     NVIN #1,"",CKOSU$,CSTT$,CEND$,0           'Imaging request
880     MNUM=M_NVNUM(1)                        'Acquire the number of workpieces recognized
890     IF MNUM=0 THEN GOTO *LVSCMD            'If no workpieces are recognized
900     IF MNUM>4 THEN MNUM=4                  'Set the maximum number (4)
910     FOR M1=1 TO MNUM                       'Repeat for the number of workpieces recognized
920        MX(M1)=P_NVS1(M1).X                 'Data acquisition
930        MY(M1)=P_NVS1(M1).Y
940        MT(M1)=P_NVS1(M1).C
950     NEXT M1
960     GOSUB *S60WRDAT                        'Tracking data storage processing
970 RETURN
980 '
990 '##### Tracking data storage processing #####
1000 *S60WRDAT
1010     FOR M1=1 TO MNUM                   'Perform processing for the number of workpieces recognized
1020        PSW=P_ZERO
1030        PSW=PRBORG                        'Virtually move the robot close to the vision sensor
1040        PSW.X=PSW.X+MX(M1)                'Create the grabbing position
1050        PSW.Y=PSW.Y+MY(M1)
1060        PSW.C=PSW.C+MT(M1)
1070        PRW=P_ZERO
1080        PRW=PSW*PVTR                      'Compensate for the error in the calculation value
1090        PRW.FL1=P_100(MWKNO).FL1
1100        PRW.FL2=P_100(MWKNO).FL2
1110        SELECT MENCNO
1120        CASE 1
1130           TRWRT  PRW,  MTR1#,  MWKNO,1,MENCNO   'Position,  encoder  value,  model  number,  buffer
number, encoder number
1140           BREAK
1150        CASE 2
```

```
1160        TRWRT PRW, MTR2#, MWKNO,1,MENCNO  'Position, encoder value, model number, buffer number, encoder number
1170        BREAK
1180      CASE 3
1190        TRWRT PRW, MTR3#, MWKNO,1,MENCNO  'Position, encoder value, model number, buffer number, encoder number
1200        BREAK
1210      CASE 4
1220        TRWRT PRW, MTR4#, MWKNO,1,MENCNO  'Position, encoder value, model number, buffer number, encoder number
1230        BREAK
1240      CASE 5
1250        TRWRT PRW, MTR5#, MWKNO,1,MENCNO  'Position, encoder value, model number, buffer number, encoder number
1260        BREAK
1270      CASE 6
1280        TRWRT PRW, MTR6#, MWKNO,1,MENCNO  'Position, encoder value, model number, buffer number, encoder number
1290        BREAK
1300      CASE 7
1310        TRWRT PRW, MTR7#, MWKNO,1,MENCNO  'Position, encoder value, model number, buffer number, encoder number
1320        BREAK
1330      CASE 8
1340        TRWRT PRW, MTR8#, MWKNO,1,MENCNO  'Position, encoder value, model number, buffer number, encoder number
1350        BREAK
1360      END SELECT
1370    NEXT M1
1380    RETURN
PBACK=(-1.369,-702.674,1.957,0.000,0.000,0.000,0.000,0.000)(0,0)
PCHK=(10.000,10.000,0.000,0.000,0.000,0.000,0.000,0.000)
PRBORG=(1.427,732.255,-2.040,0.000,0.000,0.000,0.000,0.000)(0,0)
PRW=(248.142,734.785,43.410,0.000,0.000,-2.783,0.000,0.000)(0,0)
PSW=(249.449,734.948,-2.040,0.000,0.000,0.147,0.000,0.000)(0,0)
PTEACH=(202.341,38.161,45.367,0.000,0.000,-4.612,0.000,0.000)(0,0)
PVSWRK=(203.594,8.701,0.000,0.000,0.000,-1.682,0.000,0.000)(0,0)
PVTR=(-1.307,-0.159,45.450,0.000,0.000,-2.930,0.000,0.000)(0,0)
PWKPOS=(203.652,38.282,-0.082,0.000,0.000,-1.682,0.000,0.000)(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

Oct..2009 MEE Printed in Japan on recycled paper.

Specifications are subject to change without notice.