MITSUBISHI ELECTRIC

Mitsubishi Industrial Robot

CR750/CR751 series controller

Circular Arc Tracking Function Instruction Manual

MELFA

BFP-A3380-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

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

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

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

⚠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.

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

CAUTION Use the robot within the environment given in the specifications. Failure to do so could lead to faults or a drop of reliability.
(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, 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. Also a dropped or coasted robot arm could collide with peripheral devices.

CAUTION Do not turn OFF the robot controller's main power while rewriting the robot controller's internal information, such as a program and parameter. Turning OFF the robot controller's main power during automatic operation or program/parameter writing could break the internal information of the robot controller.

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

DANGER Do not connect the Handy GOT to a programmable controller when using an iQ Platform compatible product with the CR750-Q/CR751-Q controller. Failure to observe this may result in property damage or bodily injury because the Handy GOT can automatically operate the robot regardless of whether the operation rights are enabled or not.

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

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

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

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

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

Revision history

Date of print	Specifications No.	Details of revisions
2015-03-06	BFP-A3380	First print
2017-05-10	BFP-A3380-A	Contact information of the authorized representative was updated.

■Preface

Thank you very much for purchasing Mitsubishi Electric Industrial Robot.
The circular arc tracking function allows robots to follow workpiece on a turntable and a circular arc conveyer, 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 circular arc 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.

CR750-Q/CR751-Q series : Ver. R6 or later
CR750-D/CR751-D series : Ver. S6 or later

◆When not the circular arc conveyer or turntable, but the straight conveyer is used

Please refer to “High Speed and Accuracy Tracking Function Instruction Manual” (BFP-A3382).

[Contents]

1. Overview

1.1. What is the circular arc tracking function?

The circular arc tracking function allows robots to follow workpiece on a turntable and a circular arc 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 turntable and a circular arc conveyer while working on them(conveyer tracking making use of photoelectronic sensors)
2) It is possible to follow changes of movement speed due to automatic calculation of conveyer movement speed.
3) Tracking function can be easily achieved by using Mitsubishi's robot command MELFA-BASIC V.
4) System construction is made easy by use of sample programs.

1.2. System that can achieve

With the circular arc tracking function, the example of the system that can be achieved is shown as following.

Table 1-1 Example of system that can be achieved by the circular arc tracking function

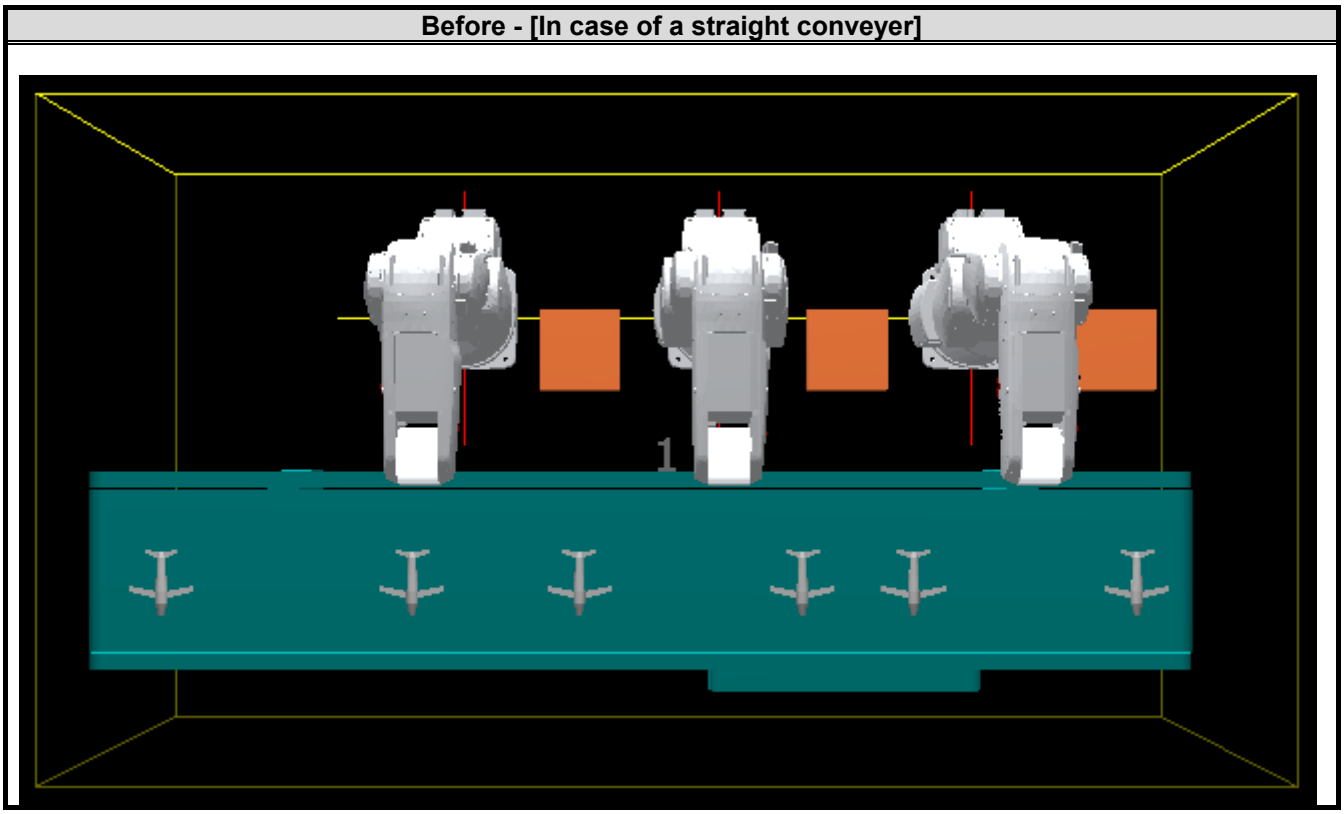
No.	CR750-Q CR751-Q	CR750-D CR751-D	Example of the system
1	●	●	A robot can catch the workpieces moving on a circular arc conveyer while tracking.
2	●	●	A robot can decorate (processing) the workpieces moving on a circular arc conveyer while tracking.
3	●	●	A robot can attach the parts (assembling) with the workpieces moving on a circular arc conveyer while tracking.
4	●	●	A robot can catch the workpieces moving on a circular arc conveyer while tracking, and a robot places the workpieces while tracking to marking on a straight line conveyer.
5	●	●	The tracking is done with an encoder of line driver (differential motion) output type.
6	●	(●) Note1)	The tracking is done with an encoder of voltage output/open collector type.
7	●	-	In case of multi CPU system, it makes possible to add max 9 pcs Q173DPX units (3 units per 1 CPU). However, in each CPU, only the two channels can be used at the 3rd set of Q173DPX units.

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

A advantage using the circular arc tracking function is shown as following.

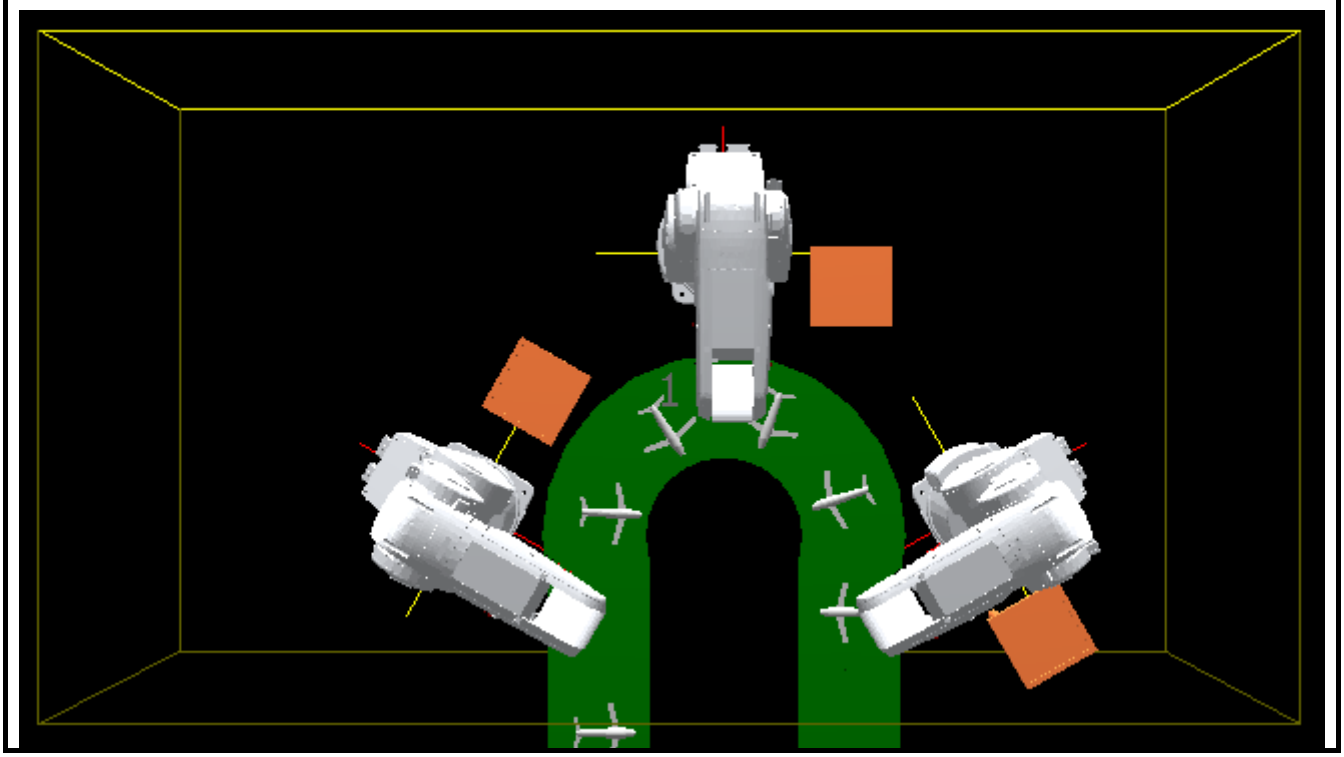
Point **The area of the system can be done small by a turntable!**

Before - [In case of a straight conveyer]



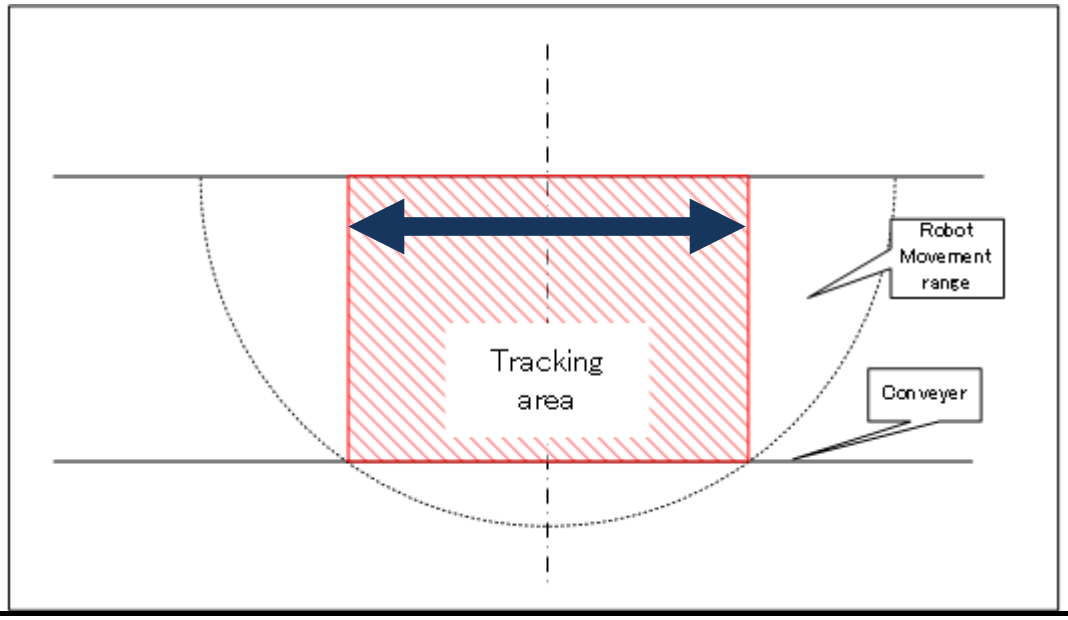
↓

Before - [In case of a circular arc conveyer]



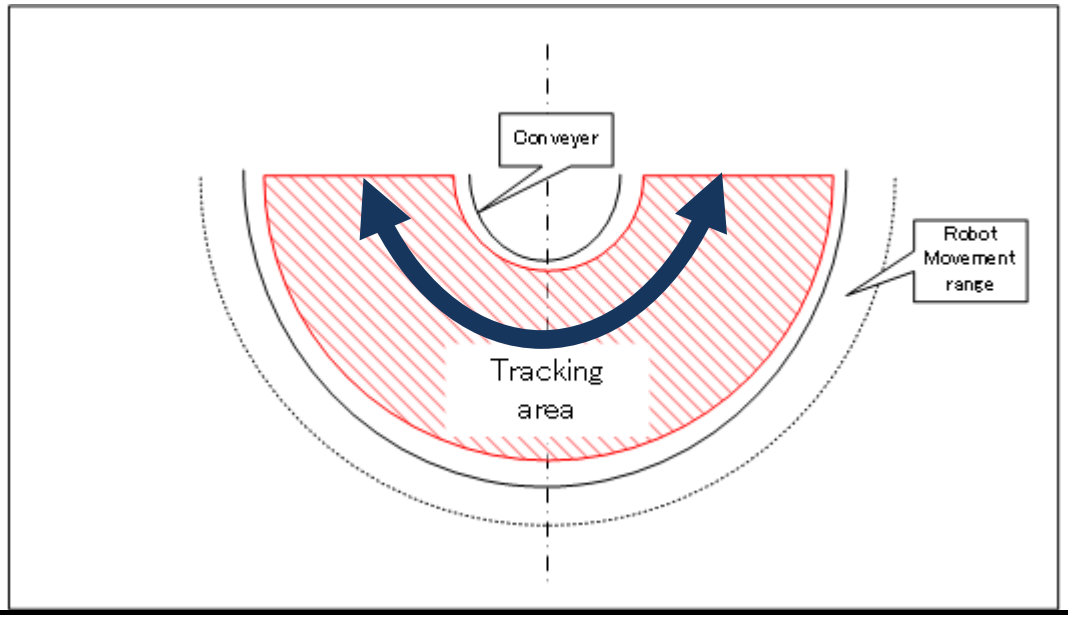
Point Tracking distance becomes long, robot can do much work!

Before - [In case of a straight conveyer]



↓

Before - [In case of a circular arc conveyer]



1.3. The terminology explanation

Table 1-2 The terminology explanation for circular arc tracking

Generic name and abbreviation	Contents
Q type	“Q type” means CR750-Q/CR751-Q series controller.
D type	“D type” means CR750-D/CR751-D series controller.
Circular arc tracking function	The conveyer tracking allows a robot to follow workpieces lining up on a turntable and a circular arc conveyer. With this function, it becomes possible to transport, process workpieces.
Conveyer tracking	The conveyer tracking allows a robot to follow workpieces lining up on a conveyer. With this function, it becomes possible to transport, process workpieces.
Vision tracking (As of February, 2015, not supported)	The vision tracking allows a robot to follow workpieces not lining up on a conveyer. With this function, it becomes possible to transport line up and process workpieces.
Q173DPX unit	Q173DRX unit is manual pulser input unit for motion controller. At Q series CPU, it is used as intelligent function unit (occupation 32 points) Each encoder figure can be got by connection with 1 pc the manual pulser machine (MR-HDP01) or 3pcs the incremental encoder.
Physical encoder number	Physical encoder numbers a number of the encoder physically allocated according to a certain rule. In the CR750-Q/CR751-Q series, the number is allocated by arranging the encoder connected with Q173DPX unit. The encoder which connected with CH1 of the Q173DPX unit specified for parameter “ENC UNIT1” is the first, the encoder which connected with CH2 is the second and with CH3 is the third. It becomes from 4 to 6 for the Q173DPX unit specified for parameter”ENCUNIT2”. It becomes from 7 to 8 for the Q173DPX unit specified for parameter”ENCUNIT3”. Note) The 3rd set of Q173DPX units can use only the two channels.
Logical encoder number	The physical encoder number change to the logical encoder number by parameter “EXTENC”. The purpose of this is to change freely number by the parameter for the encoder physically arranged. This logical encoder number is used with the instruction and the state variable of the robot program.
TREN signal	tracking enable signal

2. System Configuration

2.1. Components

2.1.1. Robot controller enclosure products

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

Table 2-1 List of Configuration in the circular arc tracking functional-related product

Product name	Model name	Remark
Circular arc tracking function INSTRUCTION MANUAL	BFP-A3380	Please download it from Web.
Sample program	-	Please refer to "7 Installation of a sample robot program" 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.

Table 2-2 List of Devices Provided by Customers

Target type Q	Target type D	Name of devices to be provided by customers	Model	Quantity	Remark
●	●	Hand	-	1	
●	●	Hand sensor	-	(1)	Used to confirm that workpieces are gripped correctly. Provide as necessary.
●	●	Solenoid valve set	Refer to Remark	(1)	Different models are used depending on the robot used. Check the robot version and provide as necessary.
●	●	Hand input cable			
●	●	Calibration jig	-	(1)	This is a jig with a sharp tip that is attached to the mechanical interface of the robot arm and used for calibration tasks. It is recommended to use the jig if high precision is required.
●		Encoder pulse unit	Q173DPX	1	Manual pulser input unit for motion controller
	●	Parallel I/O interface	2D-TZ368/ 2D-TZ378	1	Used to confirm the input of the photoelectronic sensor. [*]In the case of CR750-Q/CR751-Q, This interface and unit are unnecessary to input to the TREN signal of the Q173DPX unit.
●	●	Conveyer	-	1	
●		Encoder	[Confirmed operation product] Omron encoder E6B2-CWZ1X -1000/ E6B2-CWZ1X -2000	1	Voltage output/open collector type Line driver output
	●	Encoder	[Confirmed operation product] Omron encoder E6B2-CWZ1X -1000/ E6B2-CWZ1X -2000	1	Line driver output
●		Encoder cable	2D-CBL05/ 2D-CBL15	1	

Target type Q	Target type D	Name of devices to be provided by customers	Model	Quantity	Remark
	●	Encoder cable	-	1	Shielded twisted pair cable
	●	5V power supply	-	1	+5V DC (±10%):For Encoder [*]In the case of Q type, the Q173DPX unit supplies 5V power supply to the encoder.
●	●	Photo electronic sensor	-	1	Used to detect a workpiece position
●	●	24V power supply	-	1	+24 VDC (±10%) : For the photo electronic sensor
●	●	RT ToolBox2 (Personal computer support software)	3D-11C-WINE 3D-12C-WINE	1	Please refer to the instruction manual of RT ToolBox2 for the details of the personal computer specifications.

2.2. Example of System Configuration

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

2.2.1. Configuration Example of Q type

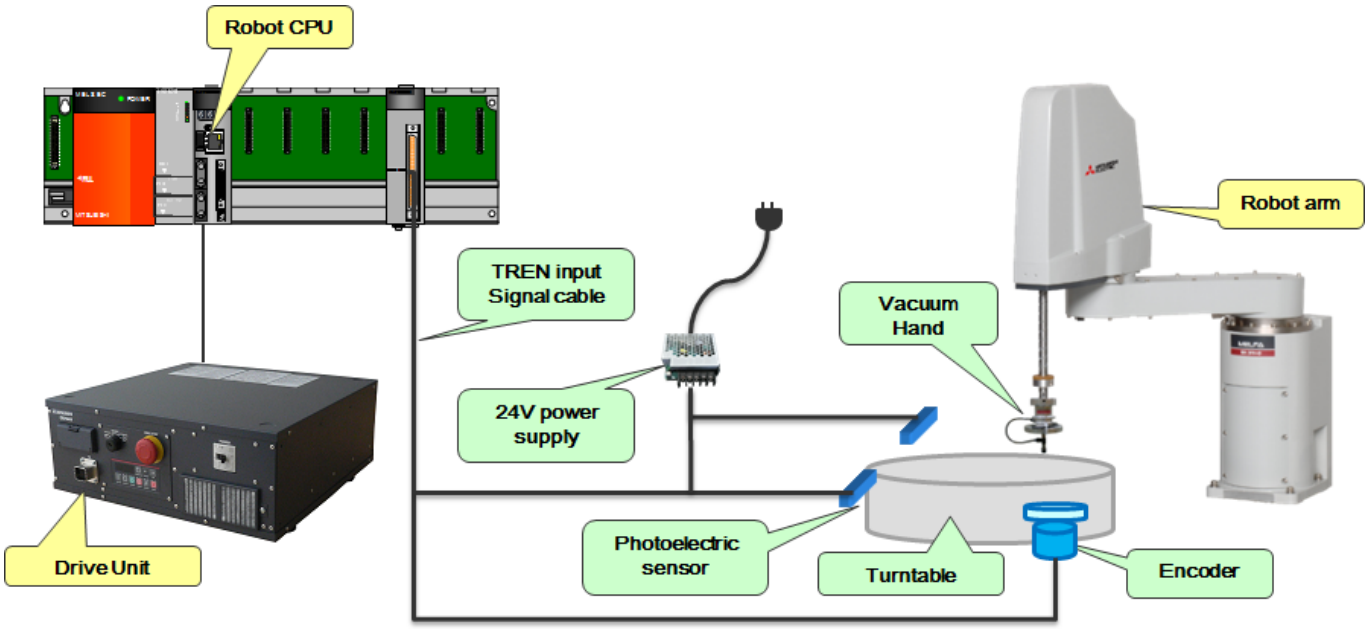


Figure 2-1 Configuration Example of Q type

2.2.2. Configuration Example of D type

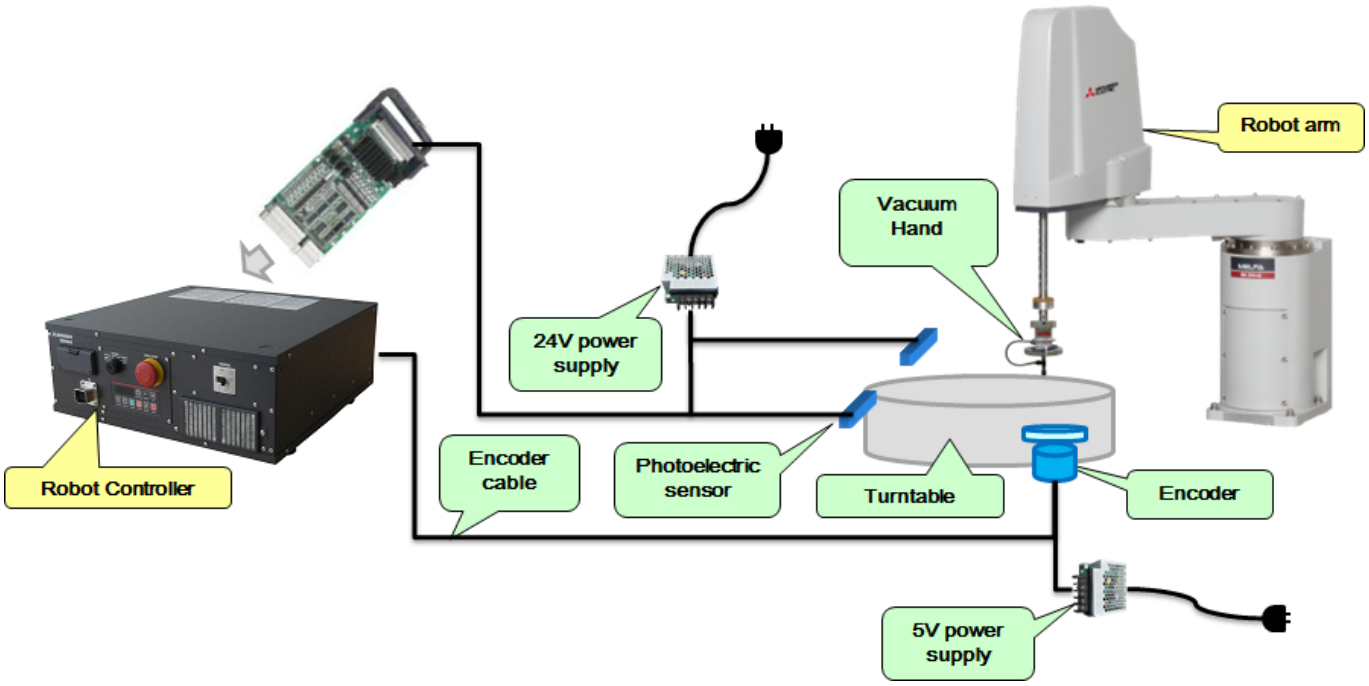


Figure 2-2 Configuration Example of D type

3. Specification

3.1. Circular arc tracking Specifications

The table below shows the circular arc tracking specifications.
Please refer to “Standard Specifications Manual” for the specifications of the robot arm and controller to be used.

3.1.1. Q type

Table 3-1 CR750-Q/CR751-Q Series Controller Circular Arc Tracking Function Specifications

Item		Specification
Supported robots (*1)		RH-FH-Q series / RV-F-Q series
Applicable robot controller		CR750-Q/CR751-Q series controller
Correspondence to Vision sensor		(As of February, 2015, not supported)
Conveyer	Number of conveyer (*2)	Max 8pcs (in case 1pc encoder connect to 1 pc conveyer) Encoder 3 pcs / Q173DPX unit 1pc Q173DPX unit 3pcs / system
	Movement Speed (*3)	Possible to support up to 300mm/s (When the robot always transport the workpieces) Possible to support up to 500mm/s when the interval of workpiece is wide.
	Encoder	Voltage output/open collector type : A、B、Z (*4) Line driver output : A、$\overline{A}$、B、$\overline{B}$、Z、$\overline{Z}$(*5) Resolution(pulse/rotation) : Up to 2000 (4000 and 8000 uncorrespond) Confirmed operation product : Omuron E6B2-CWZ1X-1000 E6B2-CWZ1X-2000
	Encoder cable	2D-CBL05(External I/O cable 5m) 2D-CBL15(External I/O cable 15m)
	Establishment	Only the X-Y plane of the robot supports (The robot cannot follow the arc direction of the Y-Z plane and the X-Z plane)
Encoder unit		Only Q173DPX unit
Photoelectronic sensor (*6)		Used to detect workpieces positions in conveyer tracking. Output signal of sensor need to be connected to TREN terminal of Q173DPX unit. (Input signal number 810 to 817) And a momentary encoder value that the input enters is preserved in state variable "M_EncL".
Precision at handling position (*3)		Approximately ±1 mm (when the conveyer speed is approximately 300 mm/s) (Photoelectronic sensor recognition accuracy, robot repeatability accuracy and so on)

(*1) The sample program doesn't correspond to the RV-5 axis robot.
(*2) The encoder connected with the third channel of the Q173DPX unit specified for parameter "ENCUNIT3" cannot be used.
(*3) The specification values in the table should only be considered guidelines. The actual values depend on the specific operation environment, robot model, hand, Sensitivity of the sensor and other factors.
(*4) Voltage output/open collector type is an output circuit with two output transistors of NPN and PNP.
(*5) The line driver output is a data transmission circuit in accordance with RS-422A. It enables the long-distance transmission.
(*6) Please connect the output signal of a photoelectric sensor with the terminal TREN of the Q173DPX unit. This input can be confirmed, by the input signal 810th-817th.

3.1.2. D type

Table 3-2 CR750-D/CR751-D Series Controller Circular Arc Tracking Function Specifications

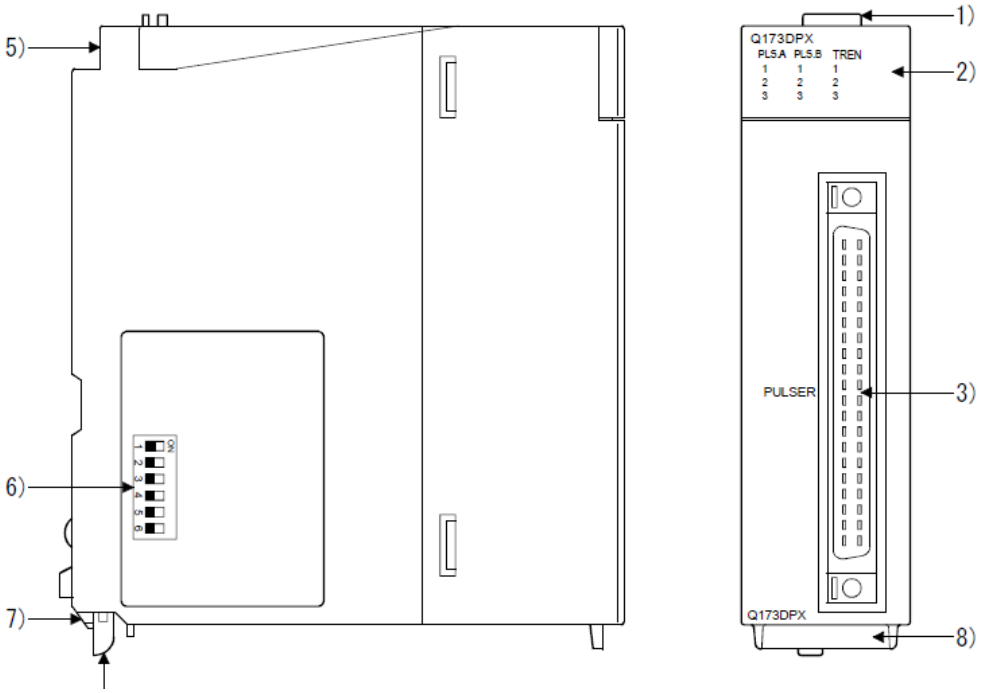
Item		Specification
Supported robots (*1)		RH-FH-D series / RV-F-D series
Applicable robot controller		CR750-D/CR751-D series controller
Correspondence to Vision sensor		(As of February, 2015, not supported)
Conveyer	Number of conveyer	Max 2pcs (in case 1pc encoder connect to 1 pc conveyer) Encoder 2 pcs / system Possible to support up to two conveyers by robot controller standard constitution
	Movement Speed (*2)	Possible to support up to 300mm/s (When the robot always transport the workpieces) Possible to support up to 500mm/s when the interval of workpiece is wide.
	Encoder	Line driver output : A、$\overline{A}$、B、$\overline{B}$、Z、$\overline{Z}$ (*3) Resolution(pulse/rotation) : Up to 2000 (4000 and 8000 uncorrespond) Confirmed operation product : Omuron E6B2-CWZ1X-1000 E6B2-CWZ1X-2000 Maximum response frequency : 100 kHz
	Encoder cable	Shielded twisted-pair cable Outside dimension : Maximum phi6mm Conductor size: 24AWG (0.2 mm^2) Cable length: Up to 25 m
	Establishment	Only the X-Y plane of the robot supports (The robot cannot follow the arc direction of the Y-Z plane and the X-Z plane)
Encoder wiring		An encoder and the robot controller are accessible with one to one Encoder Distribution Unit
Photoelectronic sensor (*4)		Used to detect workpieces positions in conveyer tracking.
Precision at handling Position(*2)		Approximately ±1 mm (when the conveyer speed is approximately 300 mm/s) (Photoelectronic sensor recognition accuracy, robot repeatability accuracy and so on)

(*1) The sample program doesn't correspond to the RV-5 axis robot.
(*2) The specification values in the table should only be considered guidelines. The actual values depend on the specific operation environment, robot model, hand, Sensitivity of the sensor 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) Please input the output signal of the photoelectric sensor into the general-purpose input signal (voluntarily) of the robot controller.

3.2. Q173DPX(manual pulser input) unit specification

Add Q173DPX unit into PLC base unit (Q3□DB) when the customer use Q type circular arc tracking function. Please refer to "Q173DCPU/Q172DCPU user's manual" about details of this unit.

(1) External and name of Q173DPX unit

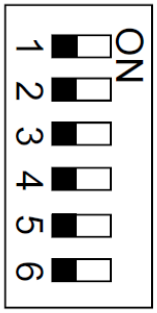


No.	Name	Application
1)	Module fixing hook	Hook used to fix the module to the base unit. (Single-motion installation)
2)	Mode judging LED	Display the input status from the external equipment. LED: PLS.A 1 to 3, PLS.B 1 to 3 — Display for input signal status of manual pulse generator/incremental synchronous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

Figure 3-1 Externals of Q173DPX unit

(2) Dip switch
By setting the dip switch, the condition of the tracking enable signal is decided.

Table 3-3 Item of dip switch

<table>
<tr><th>No.</th><th>Name</th><th colspan="2">Application</th></tr>
<tr><td rowspan="6">6)</td><td rowspan="6">Dip switches (Note-1)
 

(Factory default in OFF position)</td><td>Dip switch 1</td><td rowspan="2">Detection setting of TREN1 signal
SW1 SW2
OFF OFF
ON ON
ON OFF } TREN is detected at leading edge of TREN signal.
OFF ON TREN is detected at trailing edge of TREN signal.</td></tr>
<tr><td>Dip switch 2</td></tr>
<tr><td>Dip switch 3</td><td rowspan="2">Detection setting of TREN2 signal
SW3 SW4
OFF OFF
ON ON
ON OFF } TREN is detected at leading edge of TREN signal.
OFF ON TREN is detected at trailing edge of TREN signal.</td></tr>
<tr><td>Dip switch 4</td></tr>
<tr><td>Dip switch 5</td><td rowspan="2">Detection setting of TREN3 signal
SW5 SW6
OFF OFF
ON ON
ON OFF } TREN is detected at leading edge of TREN signal.
OFF ON TREN is detected at trailing edge of TREN signal.</td></tr>
<tr><td>Dip switch 6</td></tr>
<tr><td>7)</td><td>Module fixing projection</td><td colspan="2">Projection used to fix to the base unit.</td></tr>
<tr><td>8)</td><td>Serial number display</td><td colspan="2">Display the serial number described on the rating plate.</td></tr>
</table>

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

⚠CAUTION

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

(3) Specification of hardware

(a) Module specifications

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

(b) Tracking enable signal input

Item		Specifications
Number of input points		Tracking enable signal : 3 points
Input method		Sink/Source type
Isolation method		Photocoupler
Rated input voltage		12/24VDC
Rated input current		12VDC 2mA/24VDC 4mA
Operating voltage range		10.2 to 26.4VDC (12/24VDC +10/ -15%, ripple ratio 5% or less)
ON voltage/current		10VDC or more/2.0mA or more
OFF voltage/current		1.8VDC or less/0.18mA or less
Input resistance		Approx. 5.6kΩ
Response time	OFF to ON	0.4ms/0.6ms/1ms (CPU parameter setting, Default 0.4ms)
	ON to OFF	
Common terminal arrangement		1 point/common(Common contact: TREN.COM)
Indicates to display		ON indication(LED)

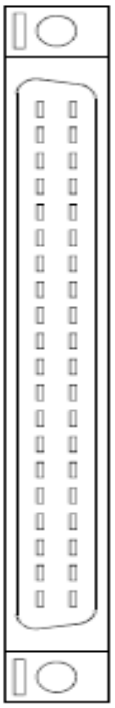
(Note): Functions are different depending on the operating system software installed.

(c) Manual pulse generator/Incremental synchronous encoder input

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

(4) Wiring

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



PULSER connector

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

Applicable connector model name

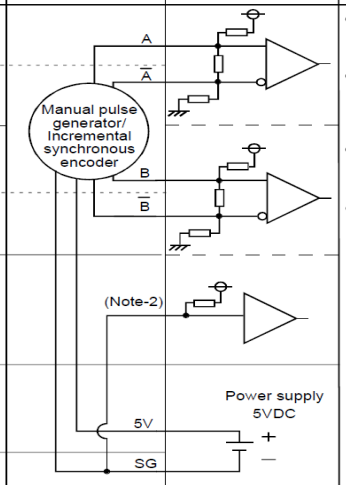
A6CON1 type soldering type connector
FCN-361J040-AU connector (FUJITSU COMPONENT LIMITED)
FCN-360C040-B connector cover
} (Attachment)

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

Figure 3-2 Pin assignment of the PULSER connector

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

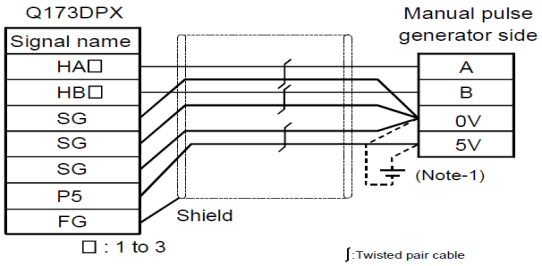
Interface between Manual pulse generator (Differential-output type)/ Incremental synchronous encoder

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

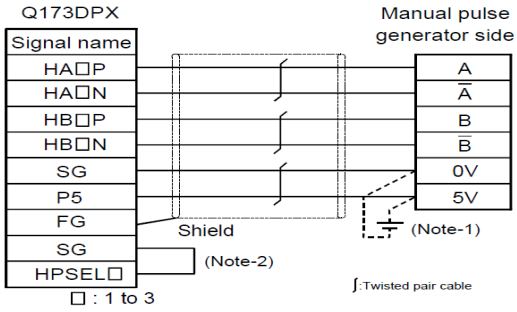
(Note-1) : The 5V(P5)DC power supply from the Q173DPX must not be connected if a separated power supply is used as the Manual pulse generator/Incremental synchronous encoder power supply. Use a 5V stabilized power supply as a separated power supply. Any other power supply may cause a failure.

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

Connection of manual pulse generator (Voltage-output/Open-collector type)



Connection of manual pulse generator (Differential-output type)



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

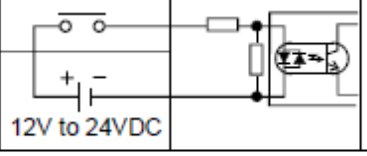
(Note-2) : Connect HPSEL□ to the SG terminal if the manual pulse generator (differential-output type)/incremental synchronous encoder is used.

Figure 3-3 Wiring connection with rotary encoder

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

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

Interface between tracking enable signal

Input or Output	Signal name		Pin No. PULSER connector 1	2	3	Wiring example	Internal circuit	Specification	Description
Input	Tracking enable	TREN□+	A4	A3	A2				Tracking enable signal input.
		TREN□–	B4	B3	B2				

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

Figure 3-4 Connected composition of tracking enable signal

⚠CAUTION

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

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

Table 3-4 Spec list of Q173DPX in robot system

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

4. Operation Procedure

This chapter explains the operation procedure for constructing a circular arc tracking system.

Start of operation

↓

1. Preparations and Connection of Equipment ····· Refer to “Chapter 5.”

[Q type]

Chapter 5 explains Q173DPX (manual pulser input) unit preparation and the connection with the encoder.

[D type]

Chapter 5 explains setting of the option card and the connection with the encoder.

↓

2. Parameter Setting ····· Refer to “Chapter 6.”

Chapter 6 explains assignment of signals for external equipment to control a robot and parameter about the tracking and parameter about movement such as the length of the tool.

↓

3. Installation of a sample robot program ····· Refer to “Chapter 7.”

Chapter 7 explains functions related to sample programs.

↓

4. Teaching Operation(“A1” Program) ····· Refer to “Chapter 8.”

Chapter 8 explains work procedure to appoint information necessary for circular arc tracking.

↓

5. Setting of an operating condition and operations check (“1“Program) ····· Refer to “Chapter 9.”

Chapter 9 explains adjustment of the conveyance route in the automatic driving and a change of the adsorption time.

↓

End of operation

6. Maintenance of robot program ····· Refer to “Chapter 10.”

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

5. Preparations and Connection of Equipment

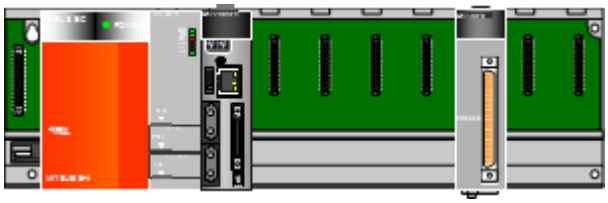
This section explains how to connect each of the prepared pieces of equipment.
Prepare equipment by referring to “Table 2-2 List of Devices Provided by Customers”.

5.1. Connection of Equipment [Q type]

The connection with each equipments is explained as follow.

5.1.1. Connection of Unit

Q173DPX unit is connected to base unit（Q3□DB）or Q6□B increase base unit.
For example, attach Q173DPX unit to I/O5 slot as follows.



5.1.2. Connection with encoder and encoder cable

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

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

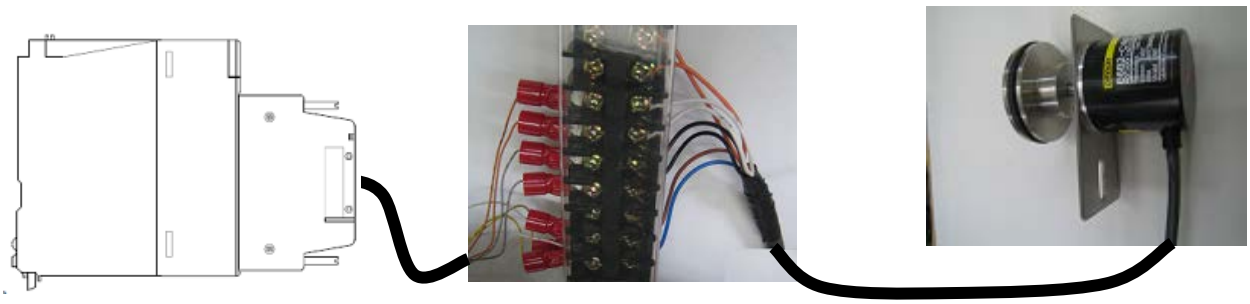


Figure 5-1 Wiring example from an encoder to a unit [Q type]

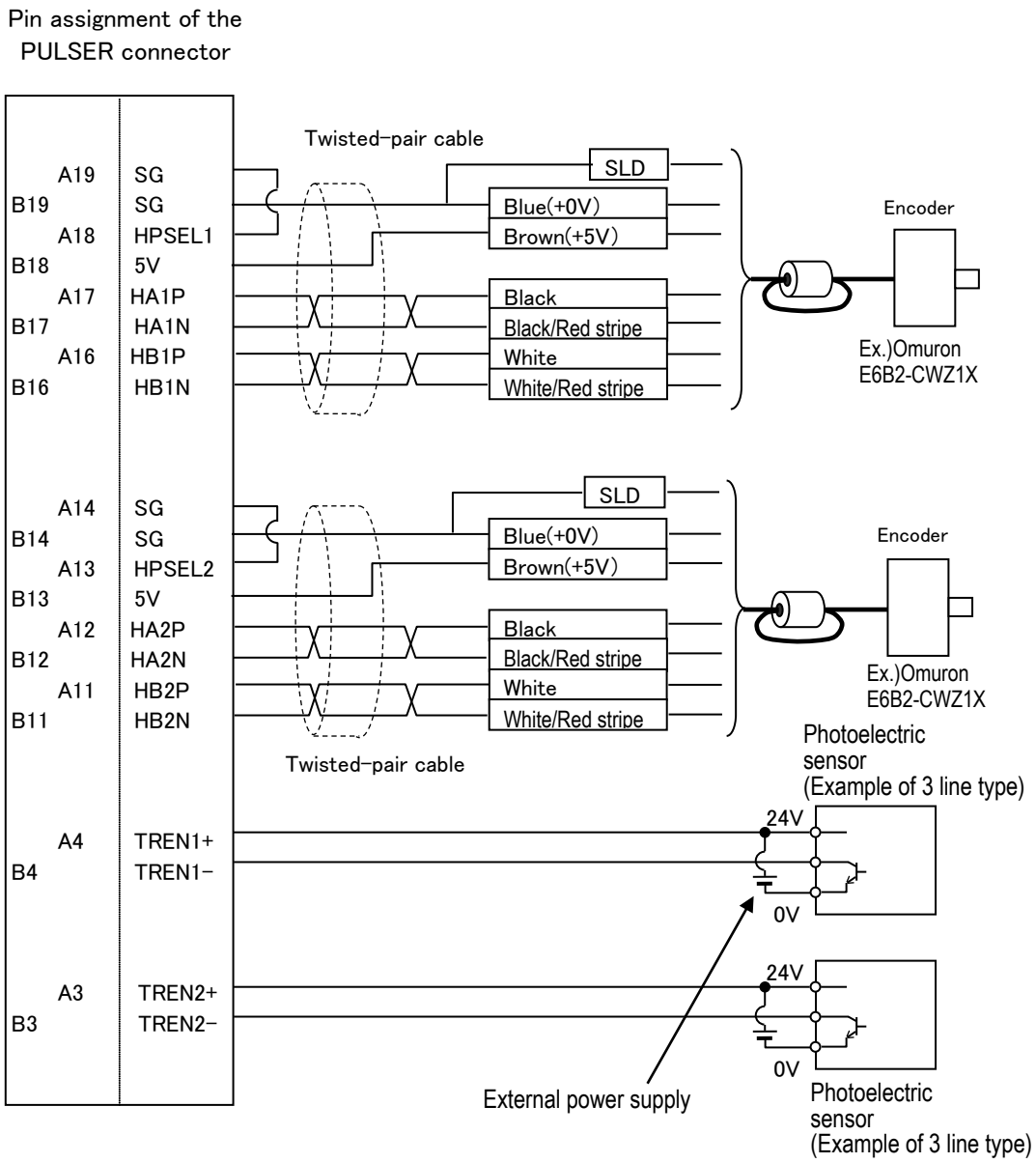


Figure 5-2 The encoder for the conveyer and the wiring diagram of the encoder cable [Q type]

※Please refer to " Figure 3-2 Pin assignment of the PULSER connector" with the pin crack of the PULSER connector that arrives at the unit.

5.1.3. Connection of photoelectronic Sensor

If a photoelectronic sensor is used for detection of workpieces, connect the output signal of the photoelectronic sensor to a tracking enable signal of the Q173DPX unit.
In this section, the connection example to 1 channel (A4, B4) is shown below.

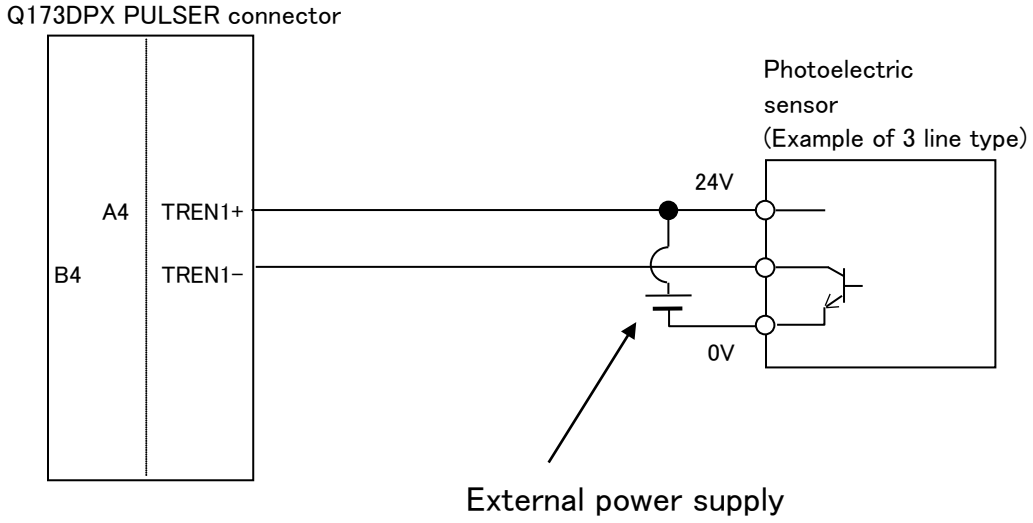


Figure 5-3 Photoelectronic Sensor Connection Example (6th General Input Signal is Used) [Q type]

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

Table 5-1 List with signal crack of tracking enable signal (TREN)

Encoder physics number	Connection channel Q type	Robot Input signal number
1	1st channel of Parameter ENCUNIT1	810
2	2nd channel	811
3	3rd channel	812
4	1st channel of Parameter ENCUNIT2	813
5	2nd channel	814
6	3rd channel	815
7	1st channel of Parameter ENCUNIT3	816
8	2nd channel	817

5.2. Connection of Equipment [D type]

The connection with each equipments is explained as follow.

5.2.1. Connection with encoder and encoder cable

E6B2-CWZ1X (made by Omron) is used, and the wiring for the encoder and the encoder cable for the conveyer is shown in " Figure 5-5 The encoder and the wiring diagram of the encoder cable (CR750-D series controller)" and " Figure 5-7 The encoder and the wiring diagram of the encoder cable (CR751-D series controller)".
The a maximum of two encoders for the conveyors are connectable as standard specification. A total of 8 signal wires are required for the connection for the power supply (+ and - terminals) and the + and - terminals of the differential encoders' A, B and Z phases. Refer to the instruction manual of the encoders to be used and connect the signal wires correctly. Note that shielded wires (SLD) should be connected to the ground of the controller and system.

CAUTION

Be sure to mount ferrite cores on all encoder cables.

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

CAUTION

There is one robot controller connectable with the one encoder.

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

(1)CR750-D series controller

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

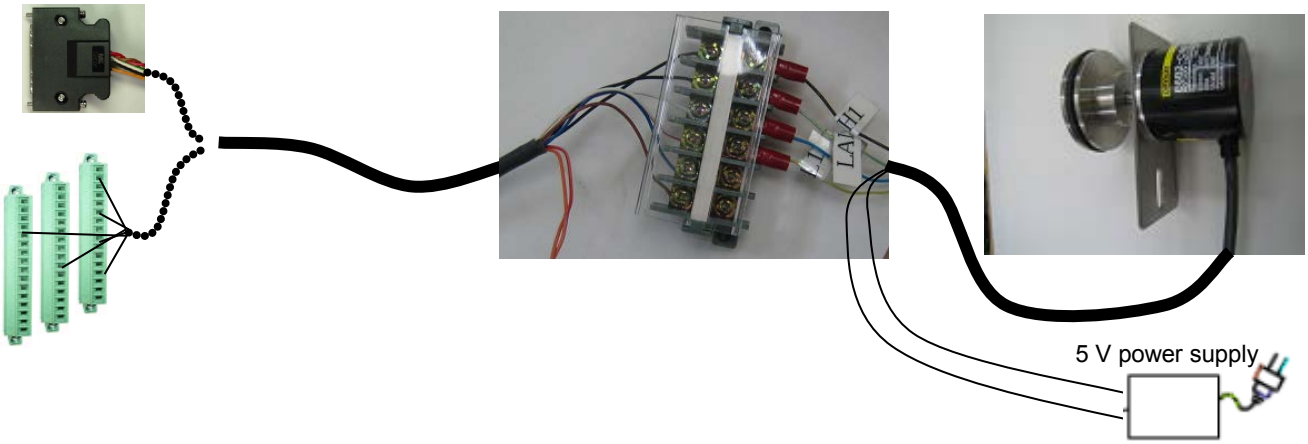


Figure 5-4 Wiring example (CR750-D series controller)

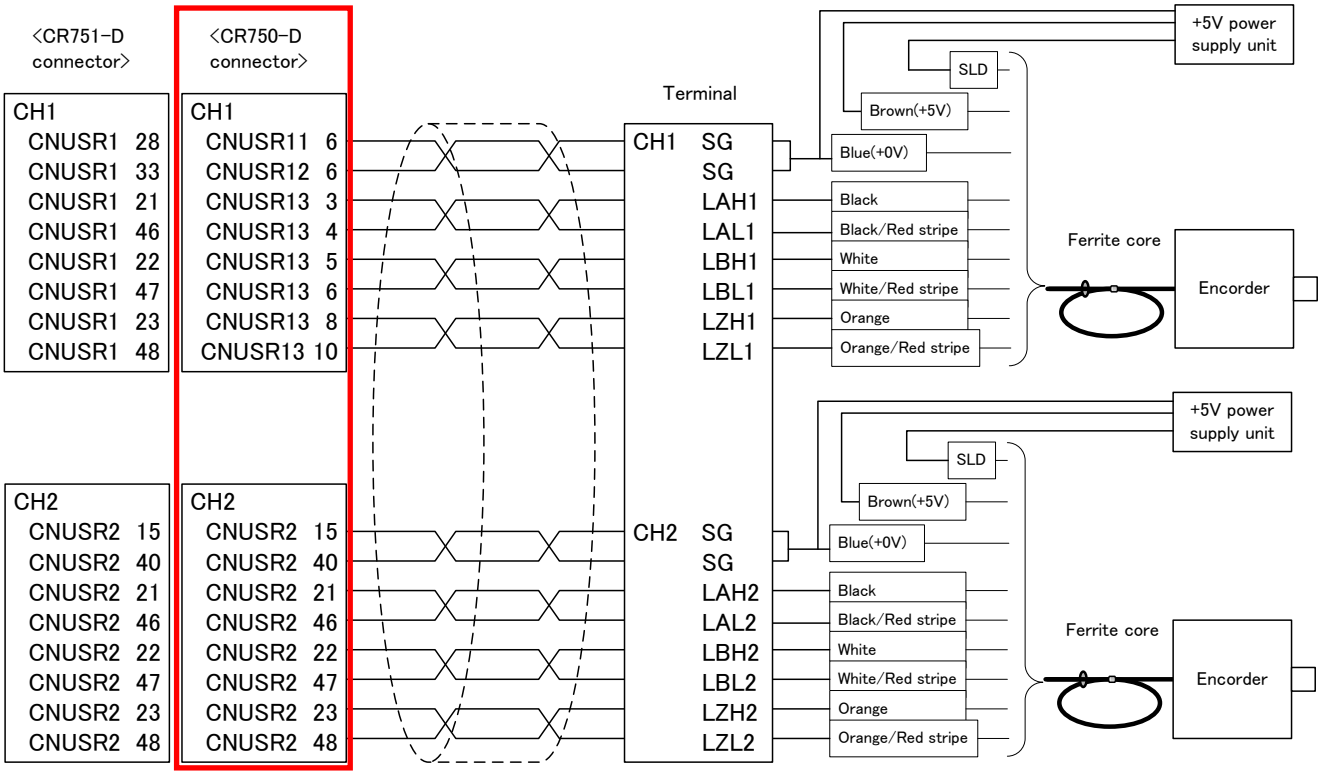


Figure 5-5 The encoder and the wiring diagram of the encoder cable (CR750-D series controller)

※Refer to "Table 12-3 Connectors: CNENC/CNUSR Pin Assignment" with pin assignment of connector CNUSR.

(2)CR751-D series controller

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

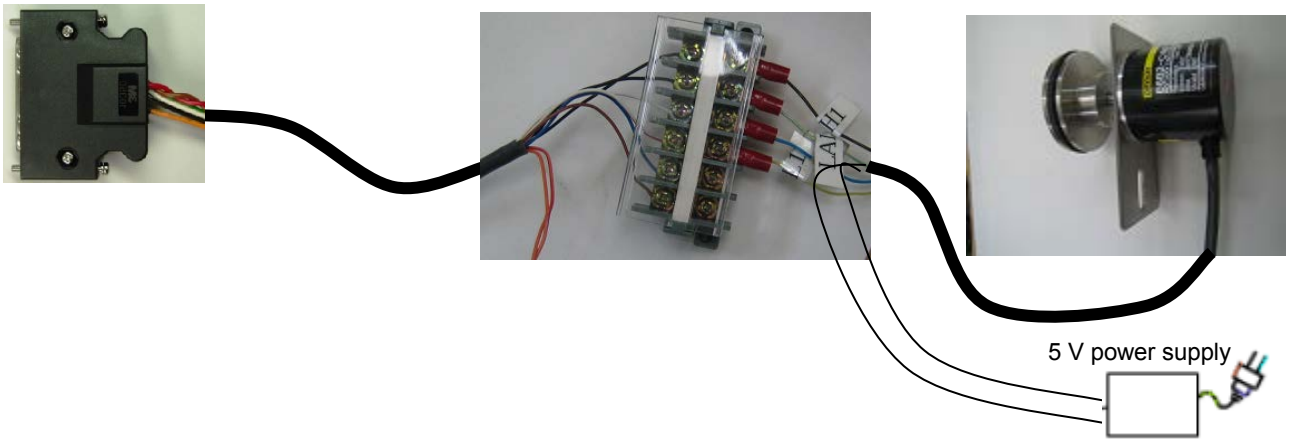


Figure 5-6 Wiring example (CR751-D series controller)

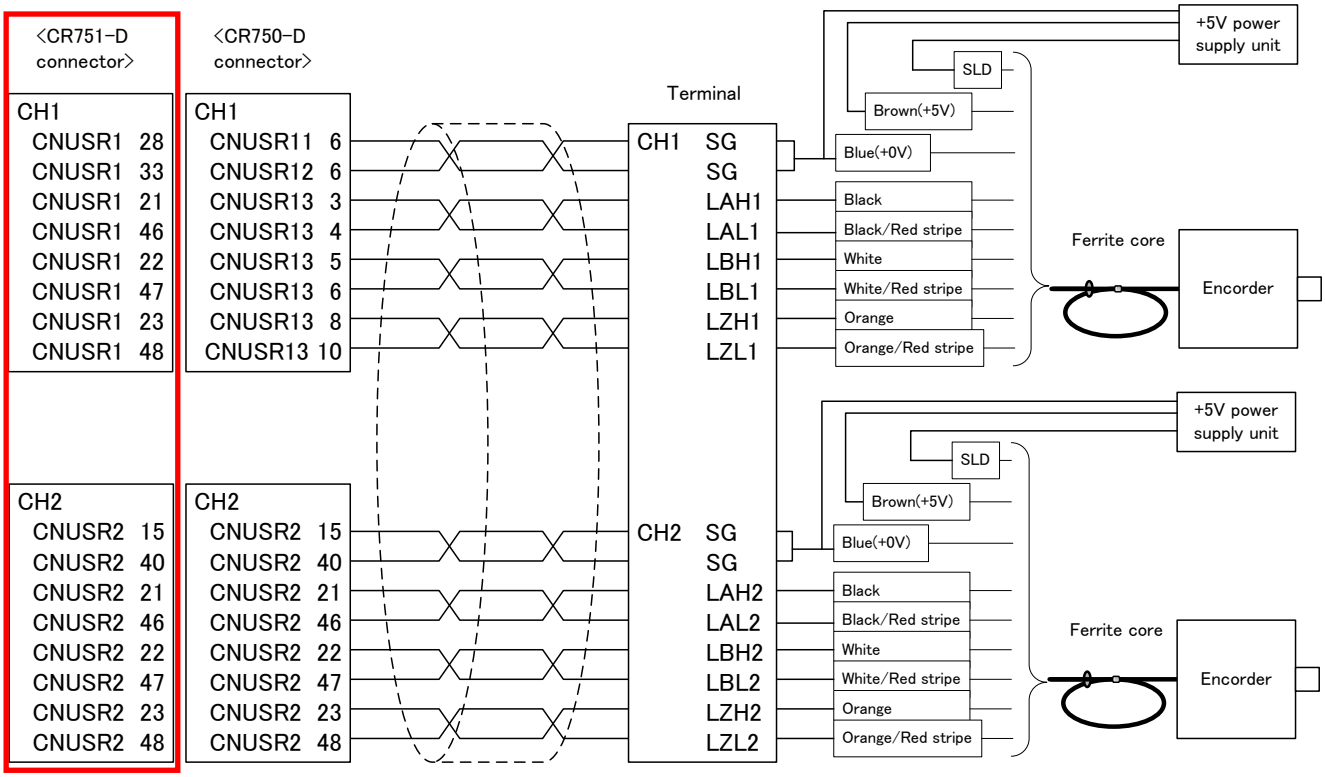


Figure 5-7 The encoder and the wiring diagram of the encoder cable (CR751-D series controller)

※Refer to "Table 12-3 Connectors: CNENC/CNUSR Pin Assignment" with pin assignment of connector CNUSR.

5.2.2. Installation of encoder cable

The installation method of the encoder cable is shown by controller to be used.
*CR750-D series: “ Figure 5-8 Installation of encoder cable (CR750-D series)”
*CR751-D series: “ Figure 5-9 Installation of encoder cable (CR751-D series)”
And, the description about the measures against the noise is shown in the figure “ Figure 5-13 Example of noise measures of tracking system”.

(1)CR750-D series controller

<CR750-D series controller (rear)>

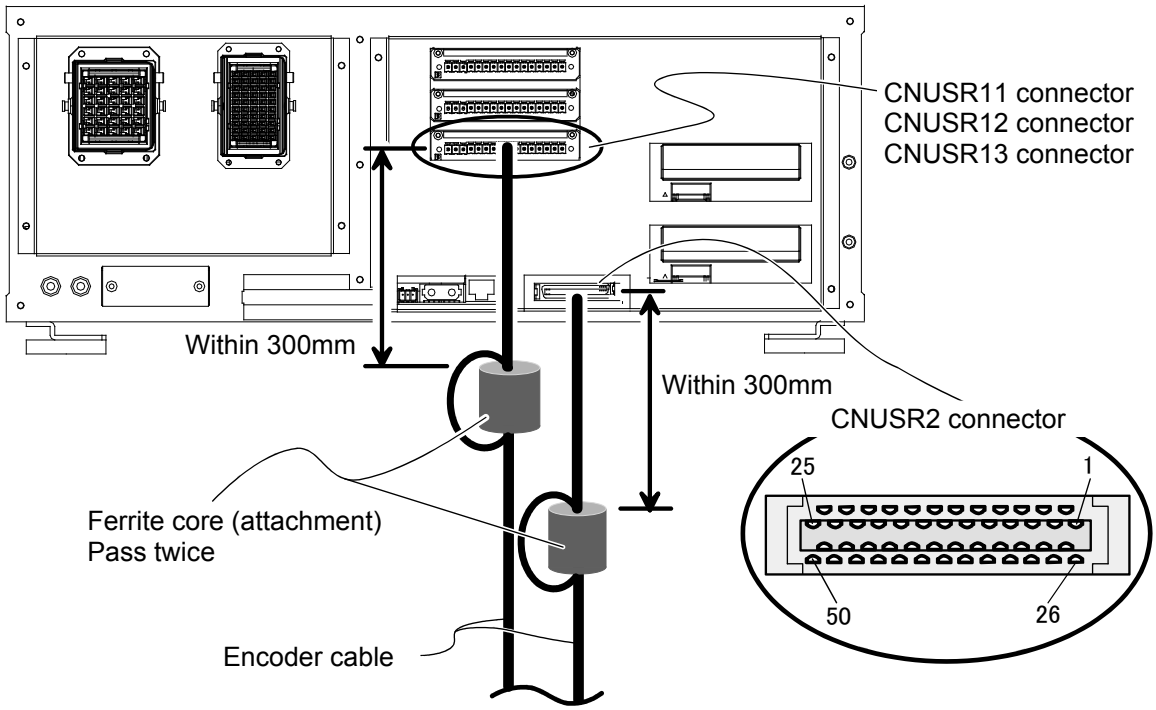


Figure 5-8 Installation of encoder cable (CR750-D series)

(2)CR751-D series controller

<CR750-D series controller (front)>

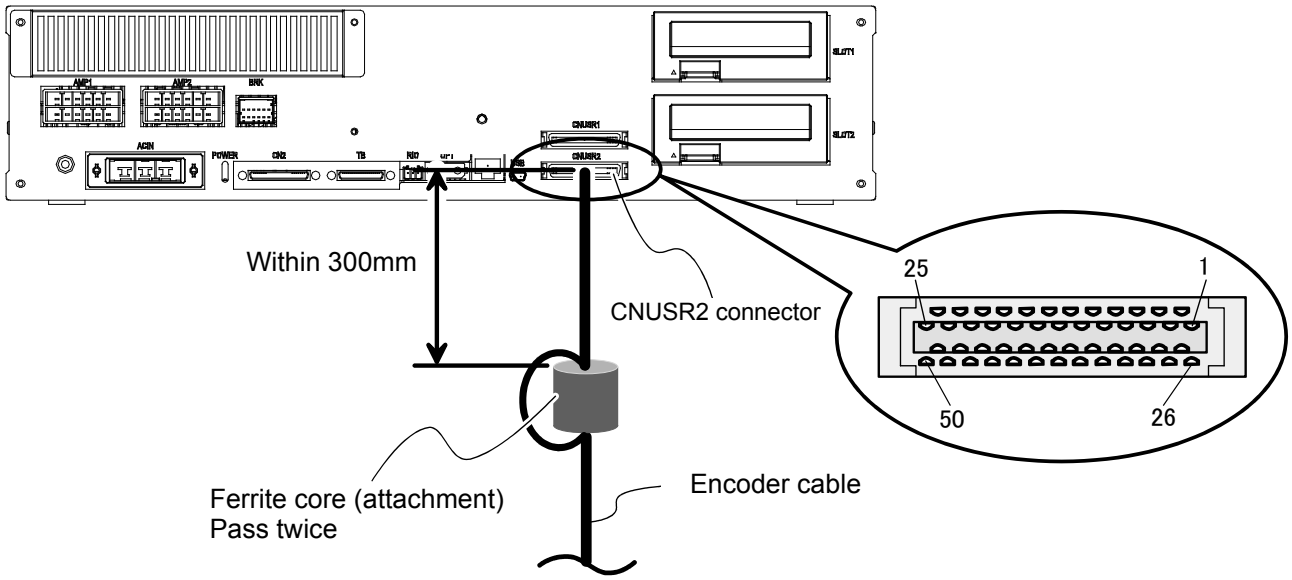
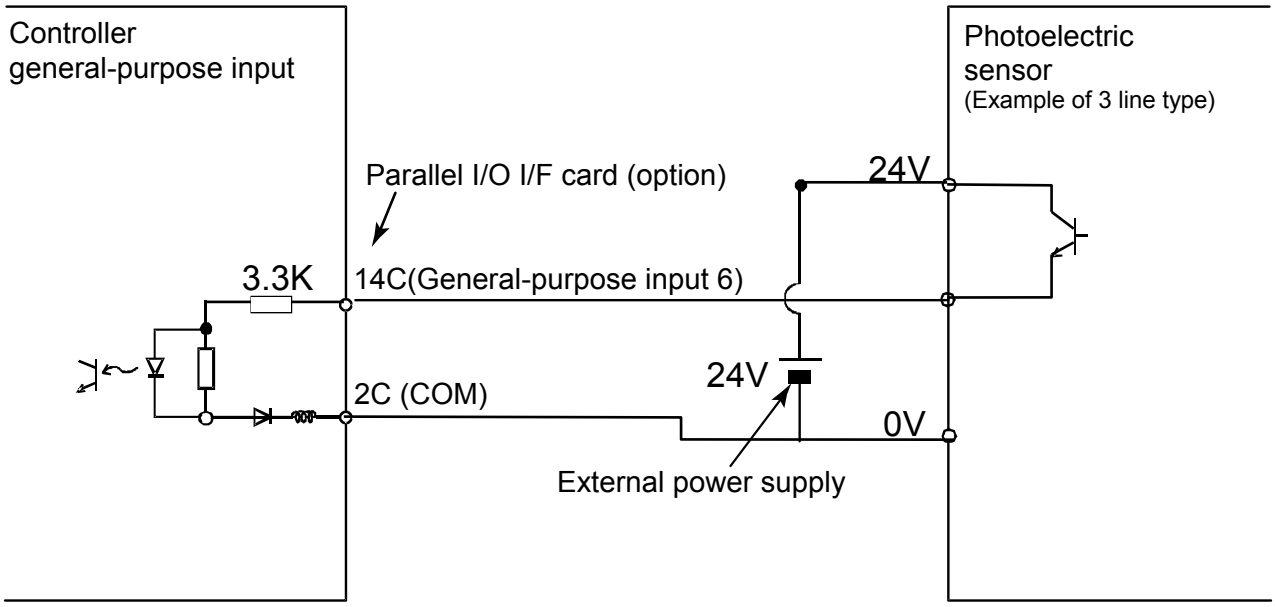


Figure 5-9 Installation of encoder cable (CR751-D series)

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 5-10 Photoelectronic Sensor Connection Example (6th General Input Signal is Used)".



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

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

Figure 5-10 Photoelectronic Sensor Connection Example (6th General Input Signal is Used)

5.3. Installation of an encoder

When installing an encoder as follows in the turntable with the short radius, there is a possibility that the tracking precision becomes bad by the case that the direction of rotation of the table and the direction of rotation of the encoder aren't identical.

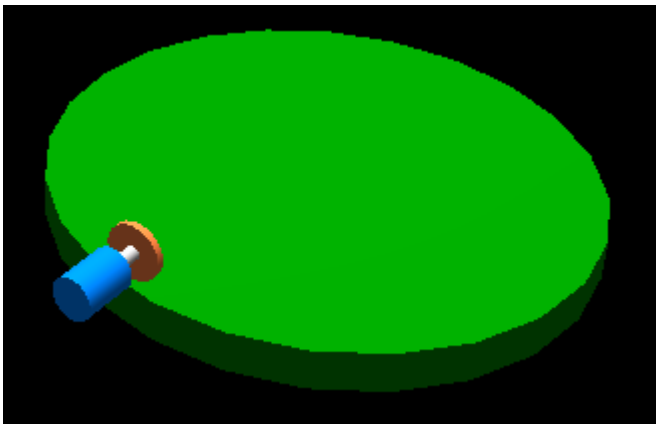


Figure 5-11 Installation example of the encoder when the tracking precision becomes bad

Please install an encoder as follows in this case.

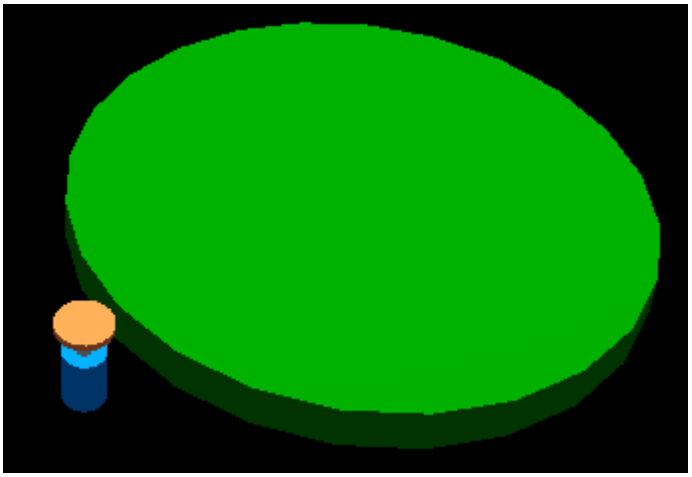


Figure 5-12 Measure example of encoder installation

5.4. Measures against the noise

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

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

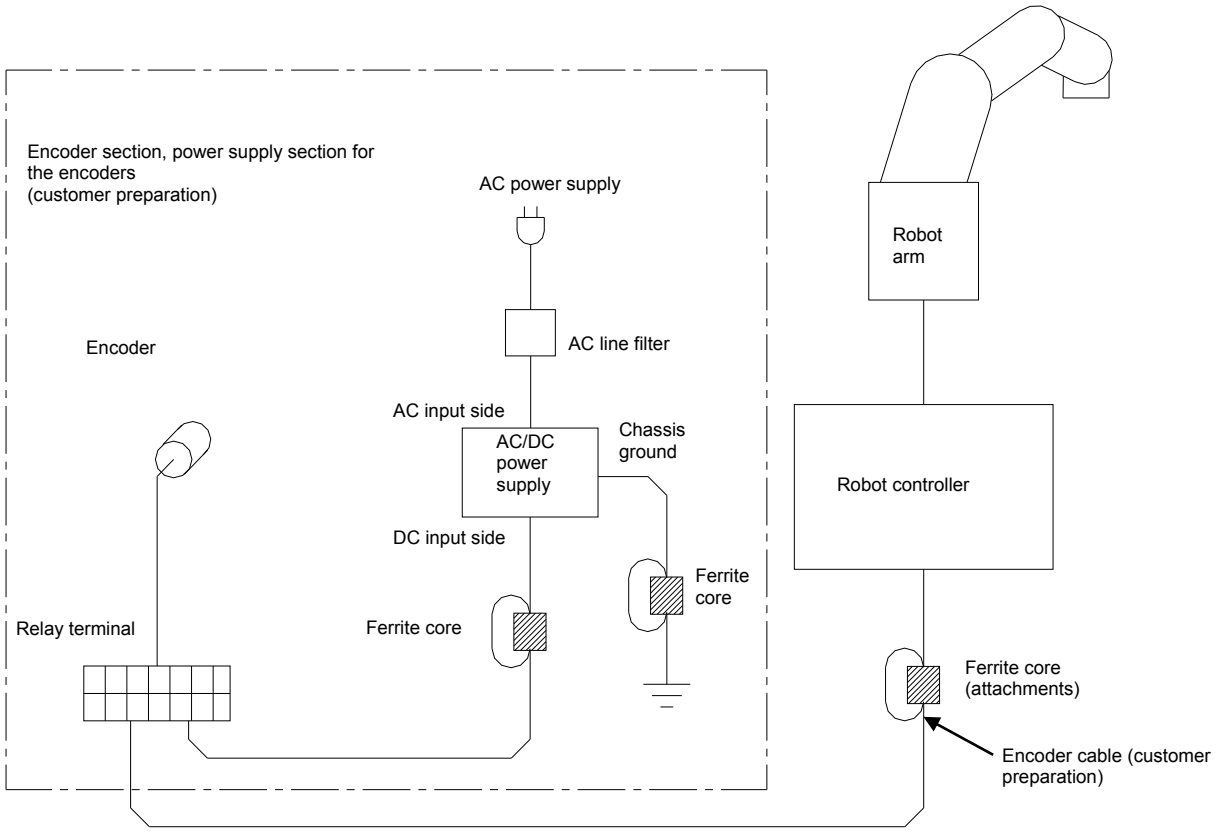


Figure 5-13 Example of noise measures of tracking system

6. Parameter Setting

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

6.1. Tracking Parameter Setting

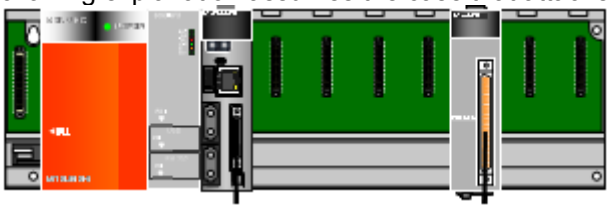
Specify to which channel of the encoder connector an encoder of conveyer is connected.
The parameter to set is shown below, make settings as required.

6.1.1. Sequencer CPU Parameter Setting [Q type]

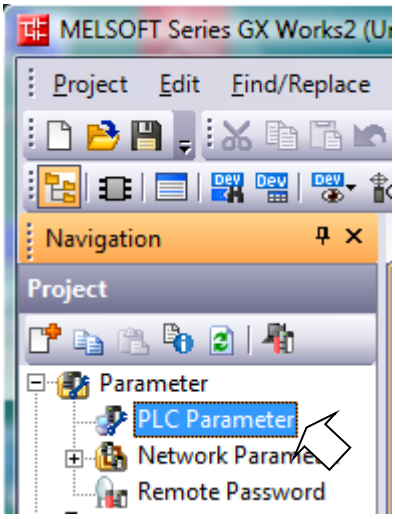
In the case of Q type, it is necessary to set multi CPU related parameters for both the sequencer CPU and robot CPU In order to use the sequencer link function.

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

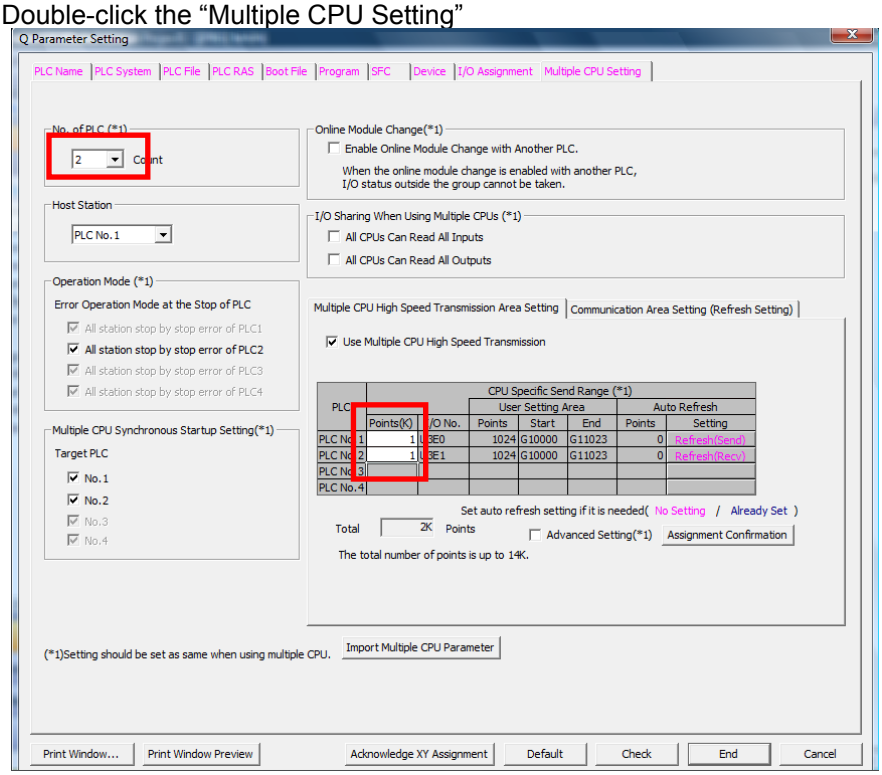
The setting procedure of the parameter is as below.
The following explanation assumes the case that attached Q173DPX unit to the fifth slot of baseboard.



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

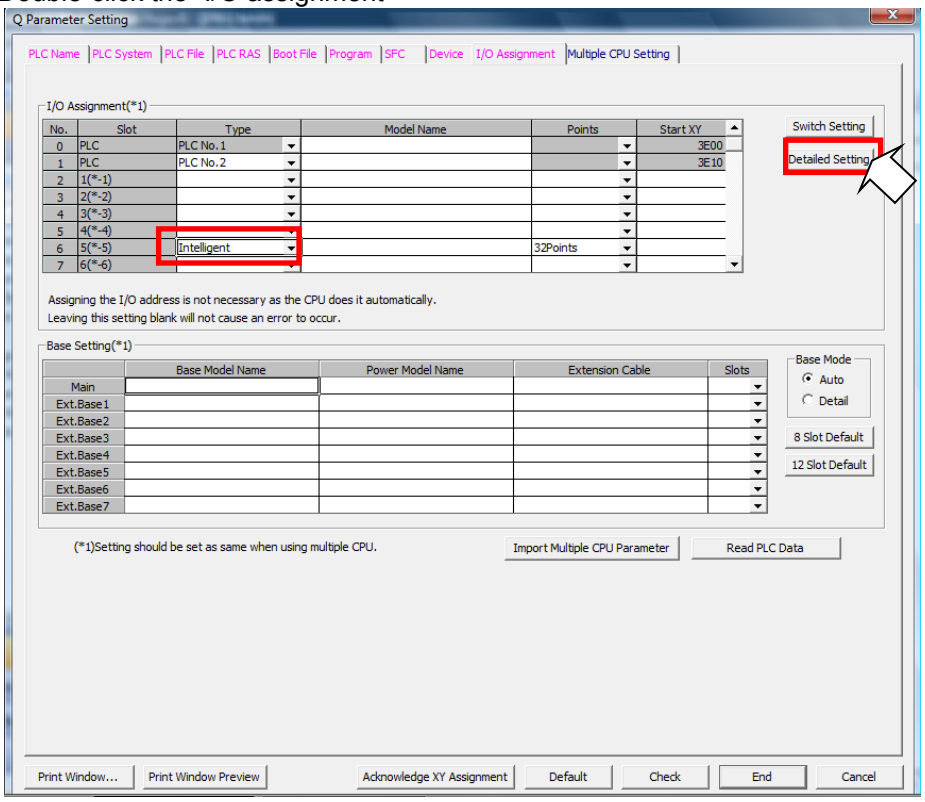


(3) Double-click the “Multiple CPU Setting”



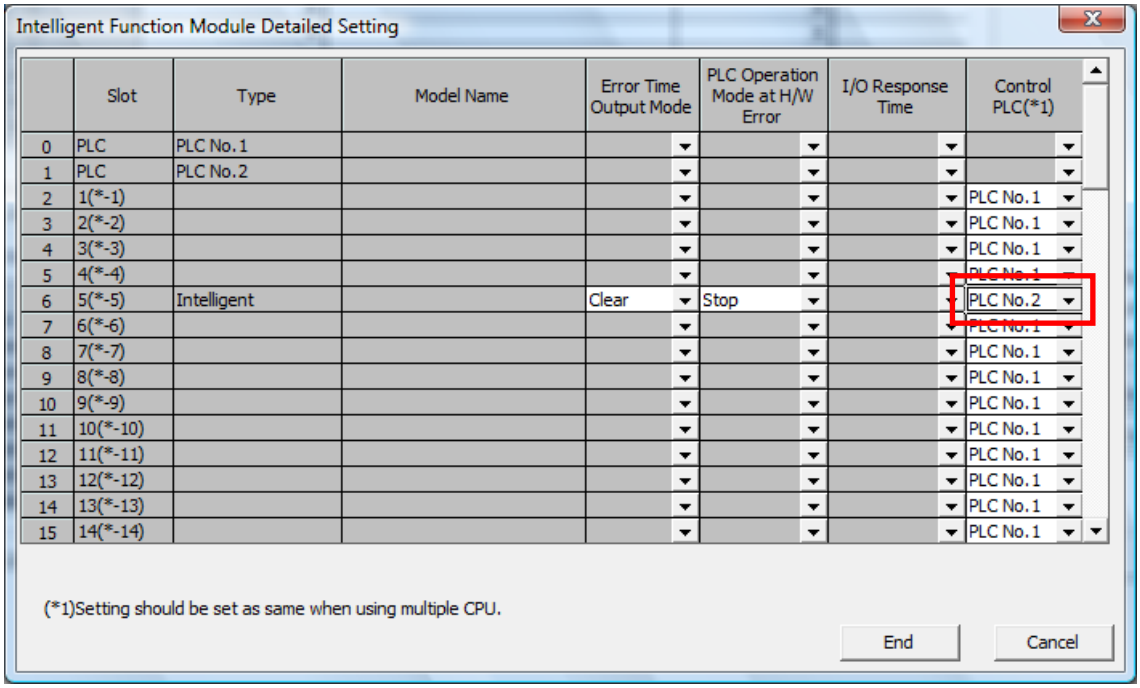
Set the number of CPU and this system area size (K Points)

(4) Double-click the “I/O assignment”



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

(5) Click the “Detailed Setting” button.



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

(6) Click the “END” button. The Parameters are memorized into the sequencer CPU.
(7) A power supply of a sequencer is reset.
(8) Close GX Works2.

6.1.2. Robot Parameter Setting

After the installation of Q173DPX module and connection with the encoder are complete, use the following steps to establish robot CPU parameters.

(1) Set a parameter TRMODE to 1, validate a function of tracking.
(2) Specify the channel to which the encoder is connected using a parameter EXTENC.
(3) In the case of Q type, Using parameter ENCUNT* (*=1 to 3), designate the slot in which Q173DPX module under the control of robot CPU is installed.
(4) Reset a power supply and reflect a parameter.

Table 6-1 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
Please set it to “1" when you use the tracking function.
0: Disable/1: Enable</td><td>0 → 1</td></tr>
<tr><td>Encoder number allocation

(*1)</td><td>EXTENC</td><td>8 integers</td><td>Set connection destinations on the connector for encoder numbers 1 to 8.
Parameter elements correspond to encoder number 1, encoder number 2 … encoder number 8 of a state variable "M_Enc" from the left.
Setting value is input encoder physics number from below list.
<table>
<tr><th rowspan="2">Encoder physics number</th><th colspan="2">Connection channel</th></tr>
<tr><th>Q</th><th>D</th></tr>
<tr><td>1</td><td>1st channel of Parameter ENCUNIT1</td><td>Standard CH1</td></tr>
<tr><td>2</td><td>2nd channel</td><td>Standard CH2</td></tr>
<tr><td>3</td><td>3rd channel</td><td>-</td></tr>
<tr><td>4</td><td>1st channel of Parameter ENCUNIT2</td><td>-</td></tr>
<tr><td>5</td><td>2nd channel</td><td>-</td></tr>
<tr><td>6</td><td>3rd channel</td><td>-</td></tr>
<tr><td>7</td><td>1st channel of Parameter ENCUNIT3</td><td>-</td></tr>
<tr><td>8</td><td>2nd channel</td><td>-</td></tr>
</table>
In the case of Q type, it is convenient to check the status variable “M_Enc” when determining the setting value of the “EXTENC” parameter.
In the case of D type, The value of the encoder which wired the channel 1 in case of the standard encoder input connector [CNENC] for the robot controller is equipped with the encoder cable with initial setting, The value of the encoder which wired the channel 2 by the status variable "M_Enc (1)", "M_Enc (3)", "M_Enc (5)", and "M_Enc (7)",It can confirm by the status variable "M_Enc (2)", "M_Enc (4)", "M_Enc (6)", and "M_Enc (8)."

Please refer to "10.1.2 List of Robot Status Variables” for the explanation of state variable “M_Enc".
Please refer to “Detailed Explanations of Functions and Operations” for how to check the status variable.</td><td>[Q type]
1,2,3,4,5,6,7,8

[D type]
1,2,1,2,1,2,1,2

↓

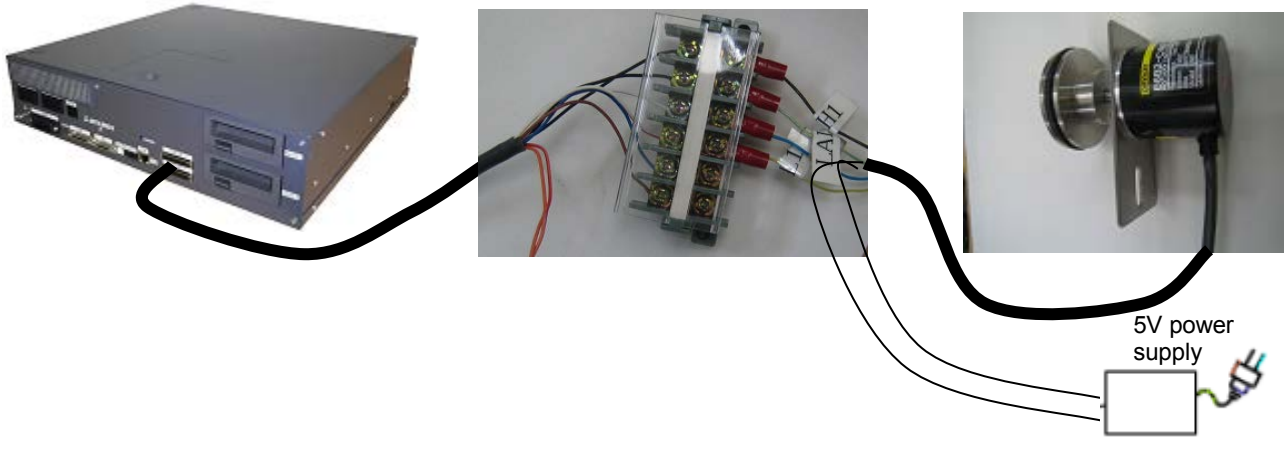
Change the set value according to the situation.</td></tr>
</table>

Parameter	Parameter name	Number of elements	Explanation	Value set at factory shipping
Tracking Workpiece judgment distance	TRCWDST	1 integer	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.	5.00 ↓ Size of the workpiece
first Q173DPX	ENCUNIT1	2 integers	The base unit-number of the first Q173DPX unit (element 1) that robot CPU uses and slot number (element 2) are set. [Element 1] -1 : No connection 0 : Basic base unit 1 - 7 : Increase base unit [Element 2] 0 - 11 : I/O Slot number	[Q type] -1,0 ↓ Installation place of Q173DPX
Second Q173DPX	ENCUNIT2	2 integers	The base unit-number of the second Q173DPX unit (element 1) that robot CPU uses and slot number (element 2) are set. [Element 1] -1 : No connection 0 : Basic base unit 1 - 7 : Increase base unit [Element 2] 0 - 11 : I/O Slot number	[Q type] -1,0
third Q173DPX	ENCUNIT3	2 integers	The base unit-number of the third Q173DPX unit (element 1) that robot CPU uses and slot number (element 2) are set. [Element 1] -1 : No connection 0 : Basic base unit 1 - 7 : Increase base unit [Element 2] 0 - 11 : I/O Slot number	[Q type] -1,0

(*1) The example of a setting of a parameter EXTENC is shown as follow.

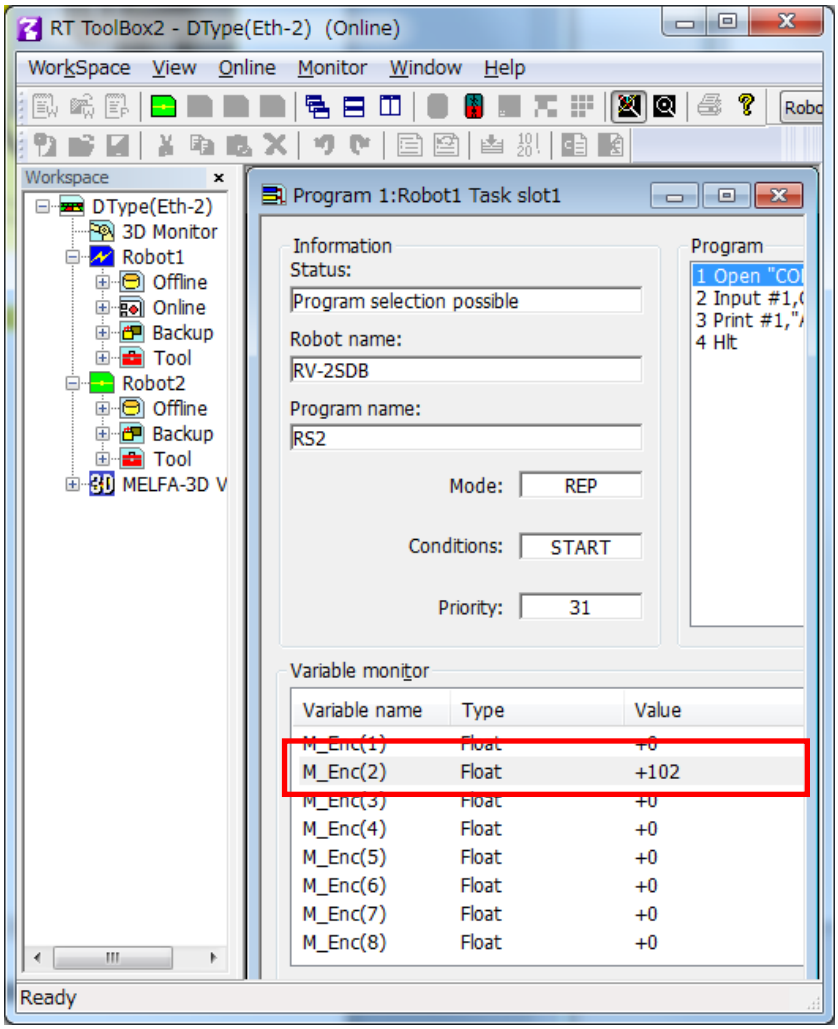
Hardware configuration

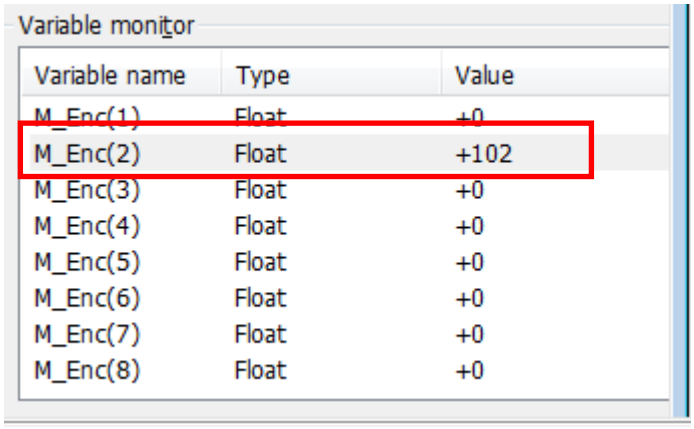
In CR750-D and a CR751-D controller, when using a common encoder cable, it is convenient to use CNUSR2 connector.
In this case, in order to connect with the channel 2 of an encoder, an encoder value will be checked using a state variable "M_Enc (2)."



Monitoring the encoder value

When the encoder value is showed by variable monitor of “Program monitor”, the encoder value changes as follows.



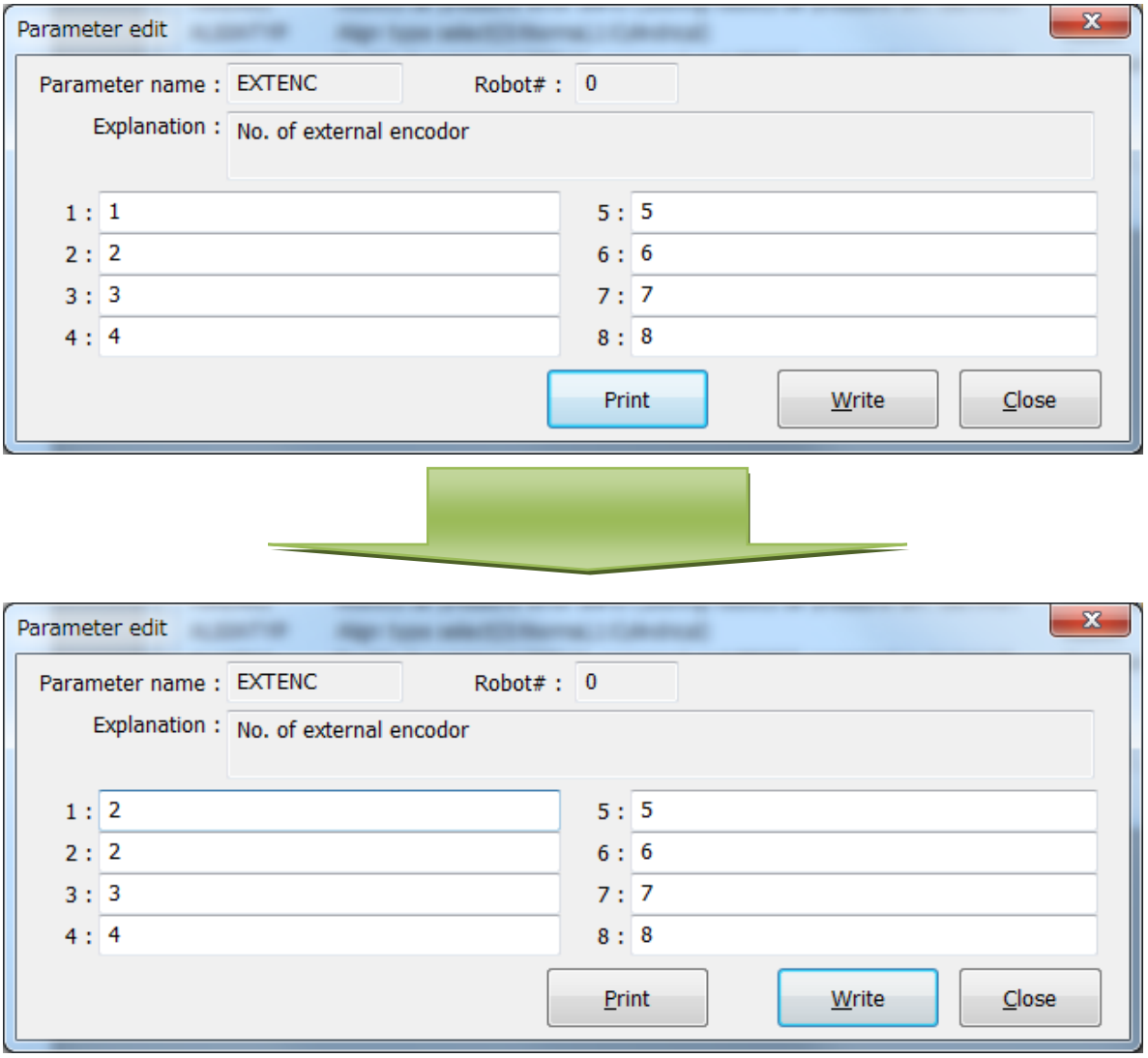


In this way, in the case of connection to channel 2, the encoder data is stored in “M_Enc(2)”.

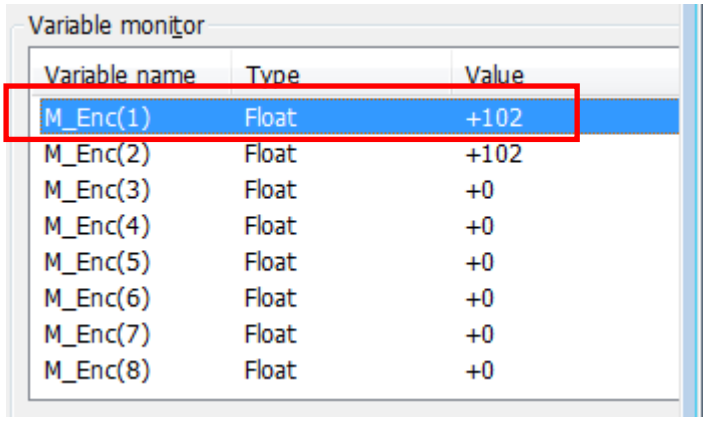
It is useful to change parameter EXTENC when confirming the encoder value by using “M_Enc(1)”and encoder value 1.

Common control to "M_Enc(1)" by parameter EXTENC

Change the first element of a parameter EXTENC into "2" from "1".



If you reset a power supply and reflect the parameter value, the encoder value is displayed in M_Enc(1)" as follows.

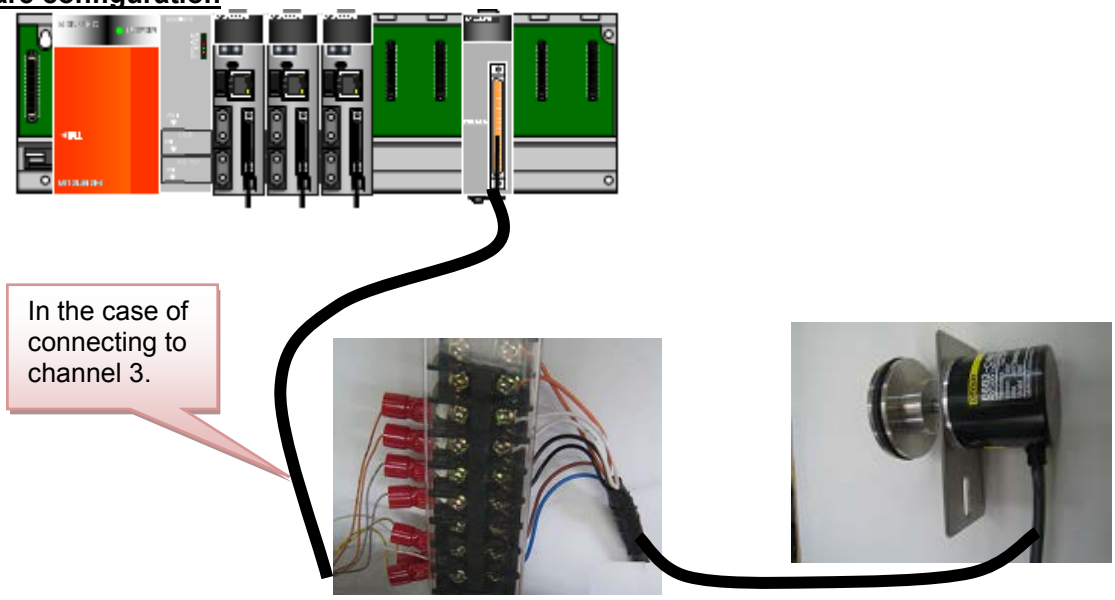


6.1.3. Example of three robots' CPU sharing one Q173DPX [Q type]

For example, the setting of one Q173DPX ,three robots CPU, and one encoder is shown as follows.

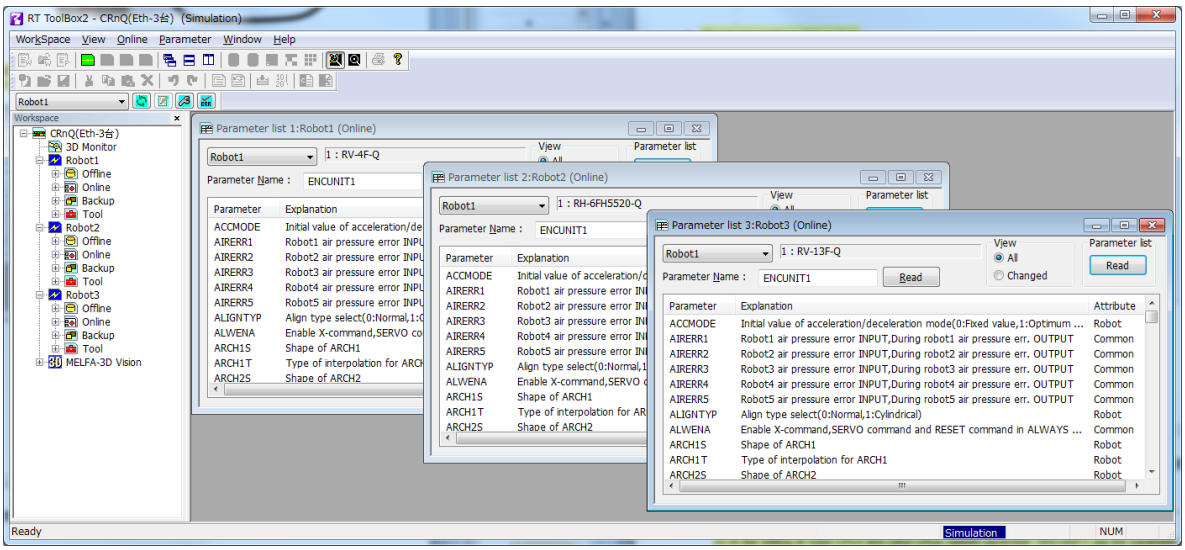
You will be able to understand some parameters ENCUNIT* and EXTENC.
[Conditions]
- An encoder is connected to the channel 3.
- Robot CPU1 and 2 use the parameter ENCUNIT1 and robot CPU3 uses the parameter ENCUNIT2.

Hardware configuration

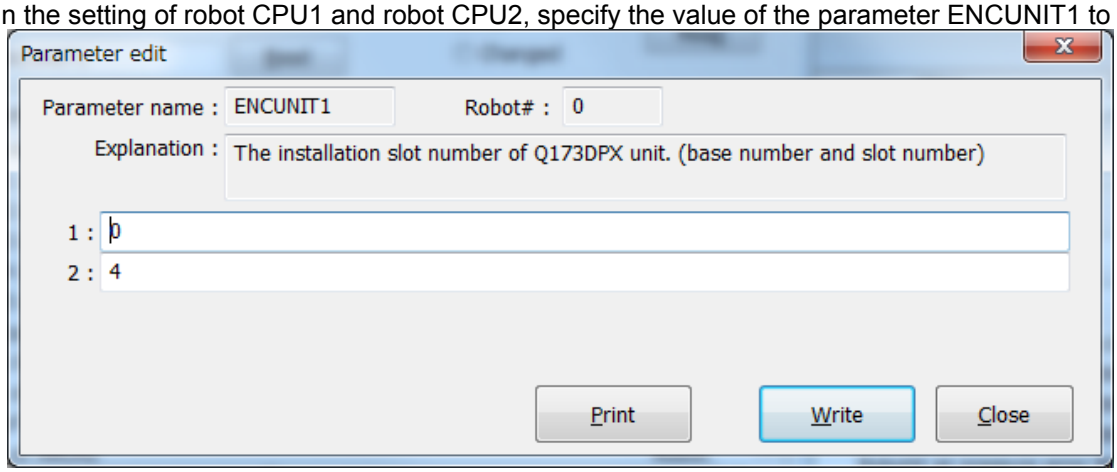


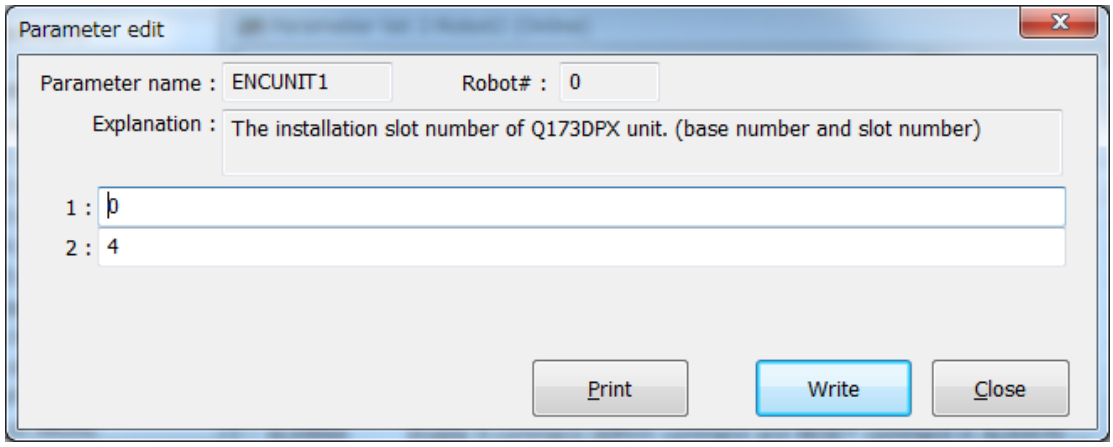
Parameter setting of the robot

(1) Display the list of parameters of three robots CPU.

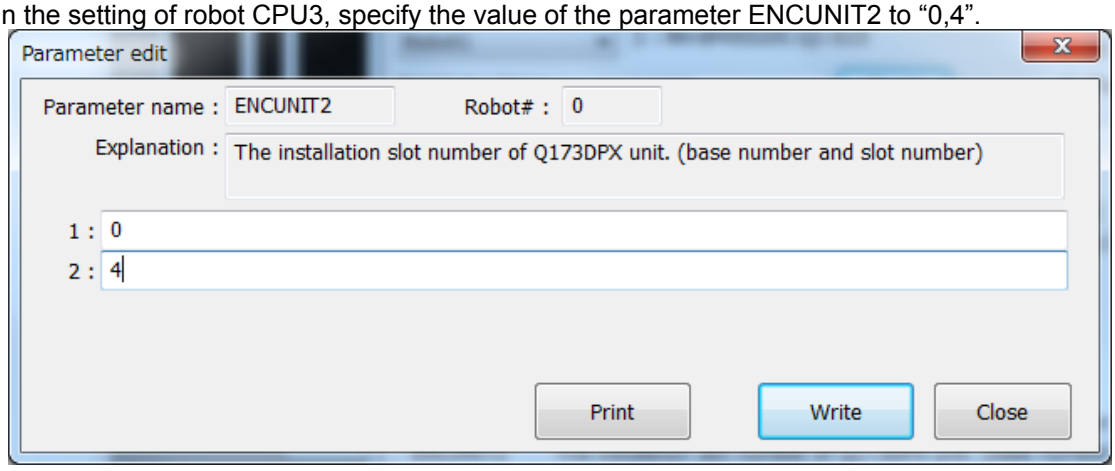


(2) In the setting of robot CPU1 and robot CPU2, specify the value of the parameter ENCUNIT1 to “0,4”.

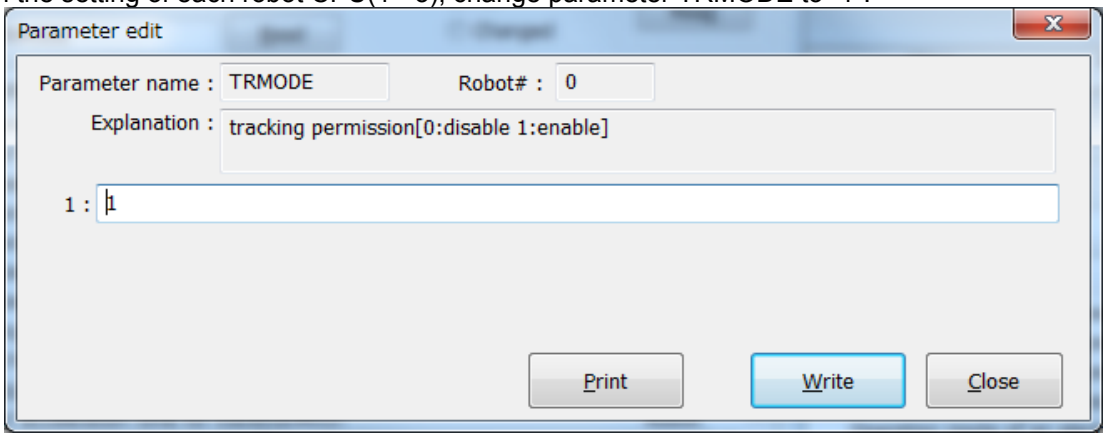




(3) In the setting of robot CPU3, specify the value of the parameter ENCUNIT2 to “0,4”.

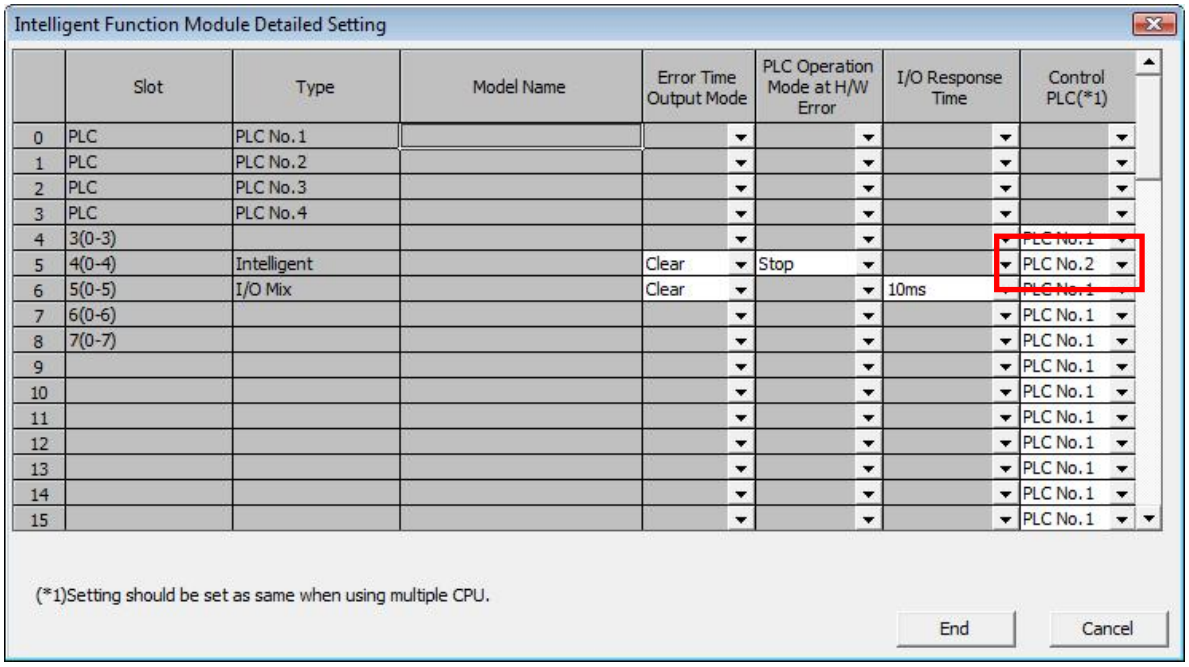


(4) In the setting of each robot CPU(1 - 3), change parameter TRMODE to “1”.



Parameter setting of GX Works

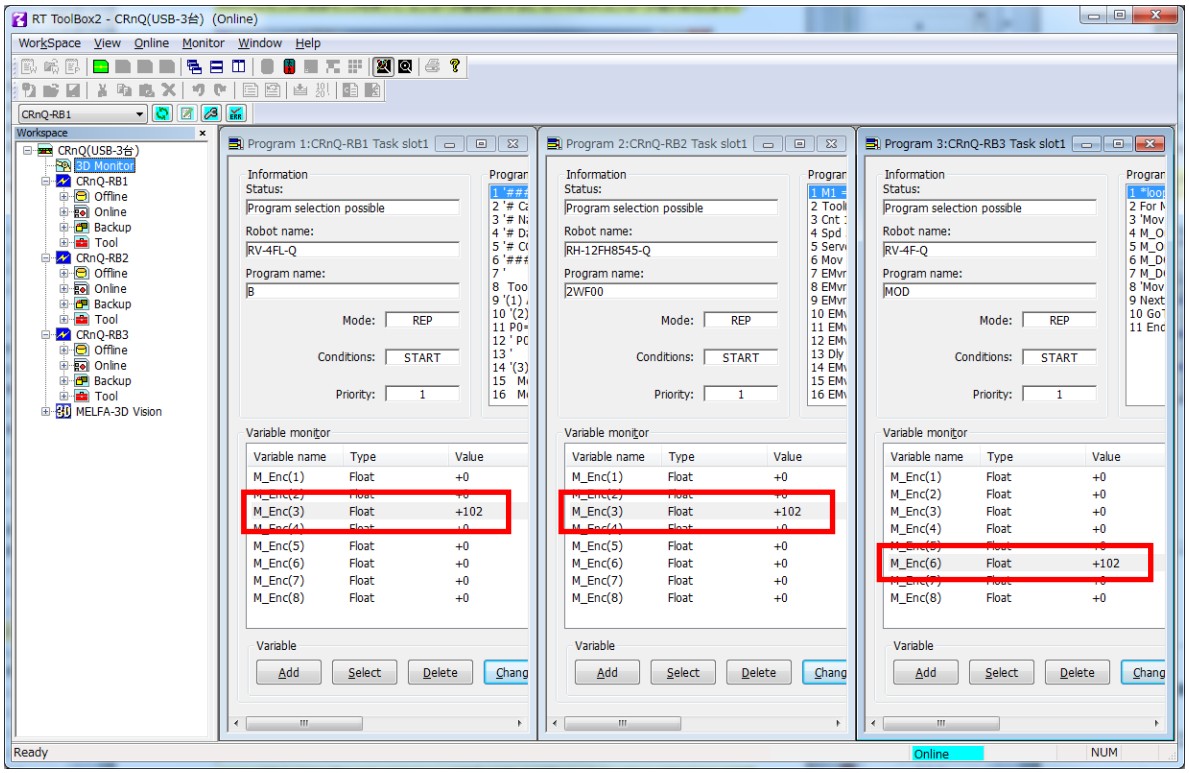
The example of the second unit (robot CPU1) controlling Q173DPX unit.

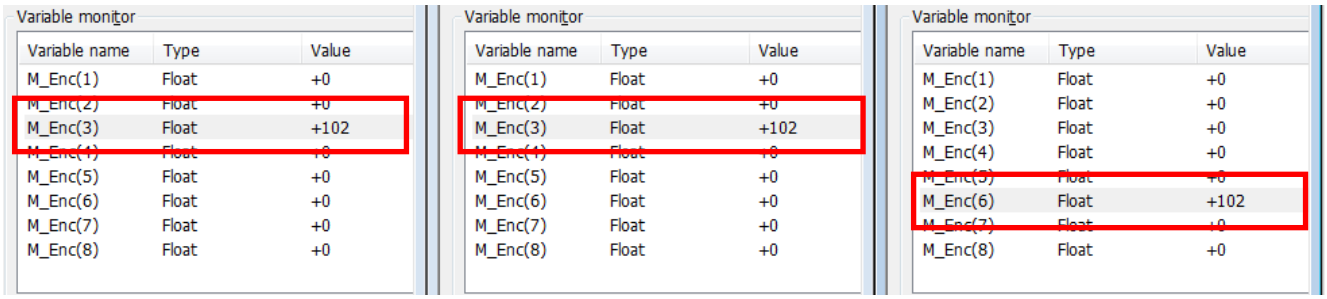


Change “Control PLC” columns to “PLC No.2” in slot 4(0-4) rows of No.5.
Reset the power supply of sequencer and the robot controller after the setting was changed.

Monitoring the encoder value

When the encoder value is showed by variable monitor of “Program monitor”, the encoder value changes as follows.



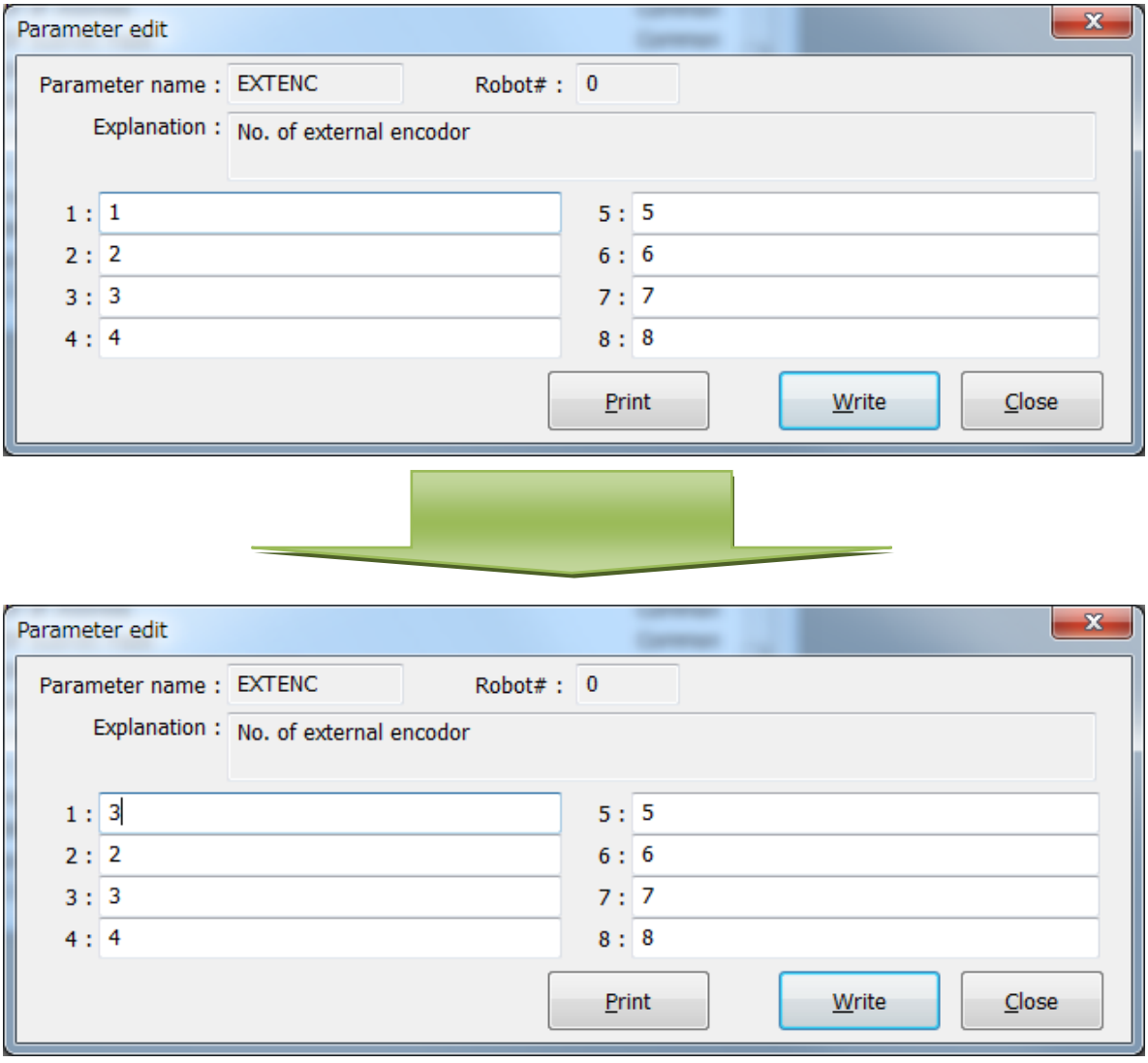


In this way, in the case of connection to channel 3, the data of robot CPU1 and robot CPU2 is stored in "M_Enc(3)".
The data of robot CPU3 is stored in"M_Enc(6)" because parameter ENCUNIT2 is specified.

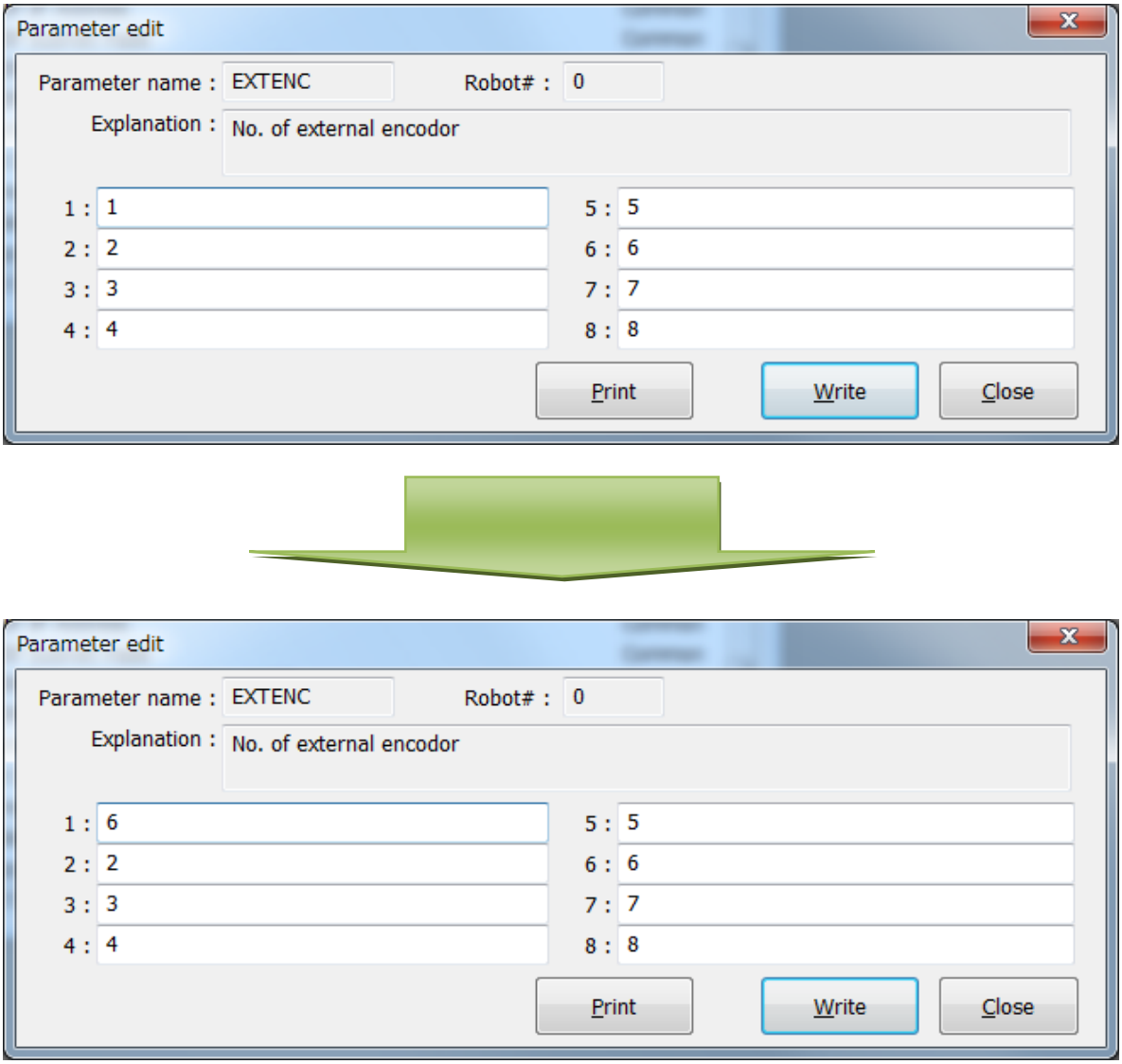
It is useful to change parameter EXTENC when confirming the encoder value by using "M_Enc(1)"and encoder value 1.

Common control to "M_Enc(1)" by parameter EXTENC

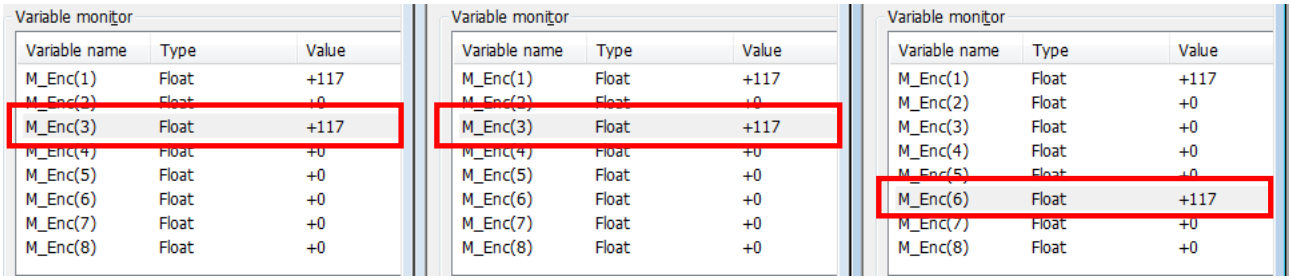
In the setting of the robot CPU1 and CPU2, change the first element of a parameter EXTENC into "3" from "1".



In the setting of the robot CPU3, changes the first element of a parameter EXTENC into “6” from “1”.



If you reset a power supply and reflect the parameter value, the encoder value is displayed in M_Enc(1)” as follows.



6.2. Operation Parameters

The following list 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
Tool coordinate system (**MEXTL**) (*1)	A parameter "MEXTL" designates a coordinate system of a tool installed in the mechanical interface side of the robot (hand). For example it's possible to move and revolve based on a tip of a hand.	Defaults: (0,0,0,0,0,0,0) For example: (0,0,+80,0,0,0,0)
Tool data 1 - 16 (**MEXTL1 - 16**) (*1)	I can point out the tool data for 16 as needed. For example when changing a hand by a multi-hand and a hand changer, it's possible to establish and change the respective tool coordinate systems.	Defaults: (0,0,0,0,0,0,0) For example: (0,0,+80,0,0,0,0)
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.**

(*1) Refer to “8.1.1 Setting of tool length” about setting of a tool length.

6.3. Dedicated Input/Output Parameters

The following list 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 6-3 List of Dedicated Input/Output Parameters

Input name/output name (parameter name)	Explanation	Setting Example (*1)	
		Q	D
Stop/pausing (**STOP**) or (**STOP2**)	Input: Stop a program Output: Output program standby status	**10000, -1**	**0 , -1**
Servo OFF/servo ON disabled (**SRVOFF**)	Input: Turn the servo off Output: Output servo ON disabled status	**10011, -1**	**1 , -1**
Error reset/error occurring (**ERRRESET**)	Input: Cancel error status Output: Output error status	**10009, -1**	**2 , -1**
Start/operating (**START**)	Input: Start automatic operation Output: Output program running status	**10006, 1**	**3 , 1**
Servo ON/turning servo ON (**SRVON**)	Input: Turn the servo on Output: Output servo on status	**10010, 0**	**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	**10005, -1**	**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	**10008, -1**	**10 , -1**
General output signal reset (**OUTRESET**)	Input: Reset a general output signal	**10015, -1**	**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	**10064, 10071**	**8 , 8**

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

7. Installation of a sample robot program

This chapter explains the structure of the sample robot programs.

Please inquire about an offer of a sample program.

Refer to “RT ToolBox2 Robot Total Engineering Support Software Instruction Manual” for how to install programs to the robot controller.

Table 7-1 List of Sample Robot Programs

Program name	Description	Explanation
A1	Setting program	This program synchronizes the coordinate system of the robot with the conveyer on the arc such as turntable and teaches the location necessary to conveyance.
1	Operation program	The recognized workpiece is followed and transported. (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. Teaching Operation("A1" Program)

This chapter explains the tasks carried out by using "A1" program.
You can just execute "1" program and do now circular arc tracking by putting this work into effect.
Please refer to "Detailed Explanations of Functions and Operations" for the steps involved in each operation.

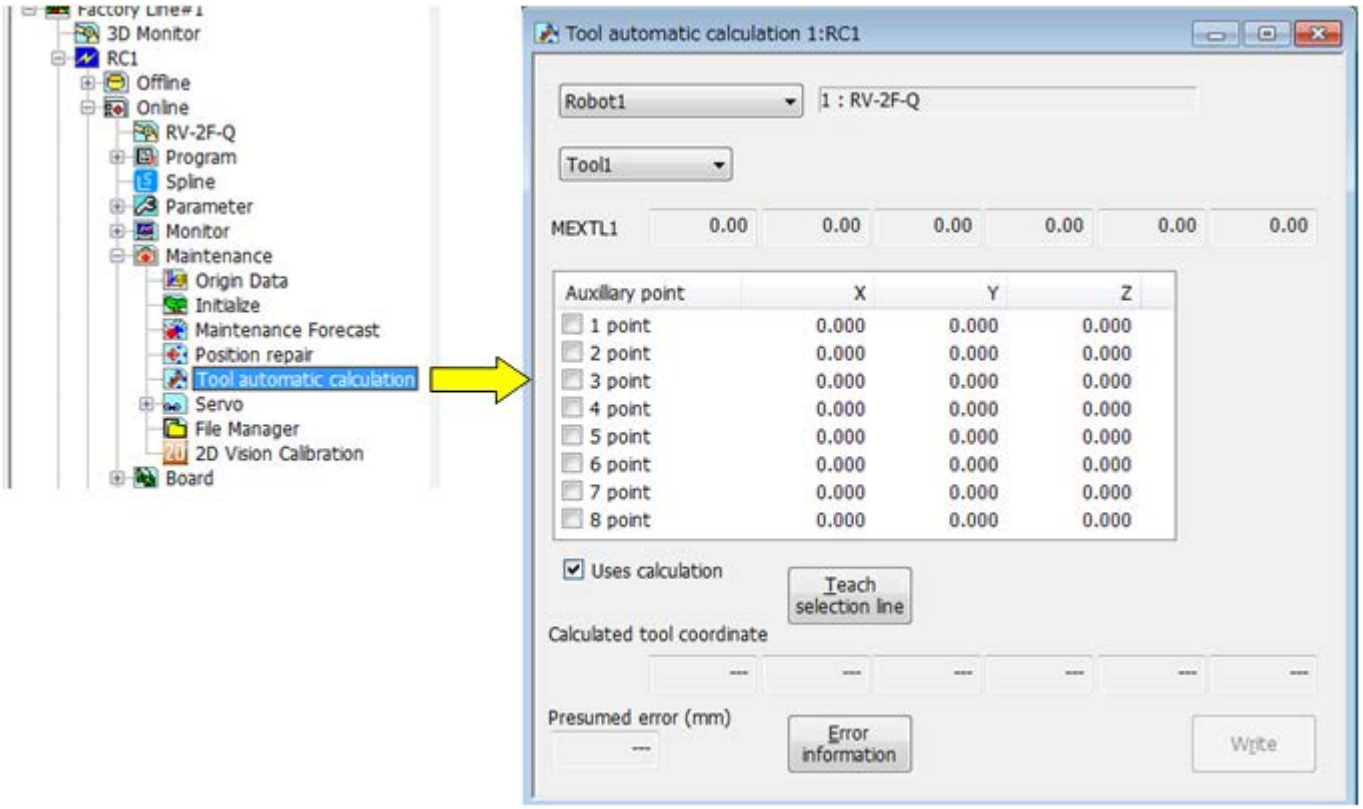
8.1. Preliminary preparations

This chapter explains the knowledge about confirmation and operation necessary to a minimum before beginning work.
The contents which should be checked are "Tool length" and "change in the encoder value".

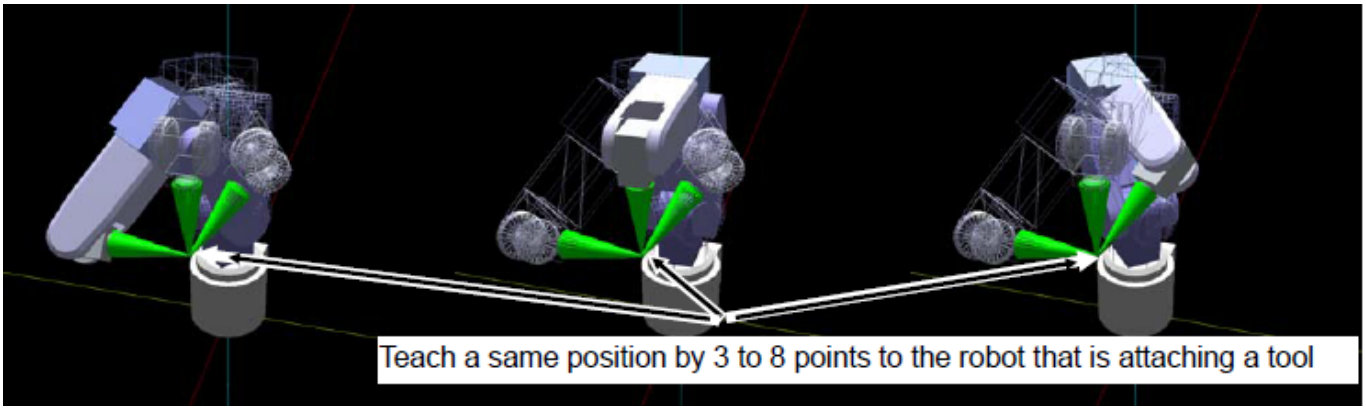
8.1.1. Setting of tool length

When you'd like to change the angle at the place which isn't a flange part of a robot(For example, tip of a hand), you have to set tool length.
The "tool length automatic measuring system" function of RT ToolBox2 is useful when setting tool length.
Refer to "RT ToolBox2 Robot Total Engineering Support Software Instruction Manual" about operational details.

When the robot model and robot controller which have connected, correspond to this function, a [Tool automatic calculation] is displayed under [Maintenance] in the project tree. Double-click [Online] ->[Maintenance] -> [Tool automatic calculation] in the project tree.



Tool length is calculated automatically by instructing in the location of 3-8 points as follows in the screen mentioned above.



CAUTION

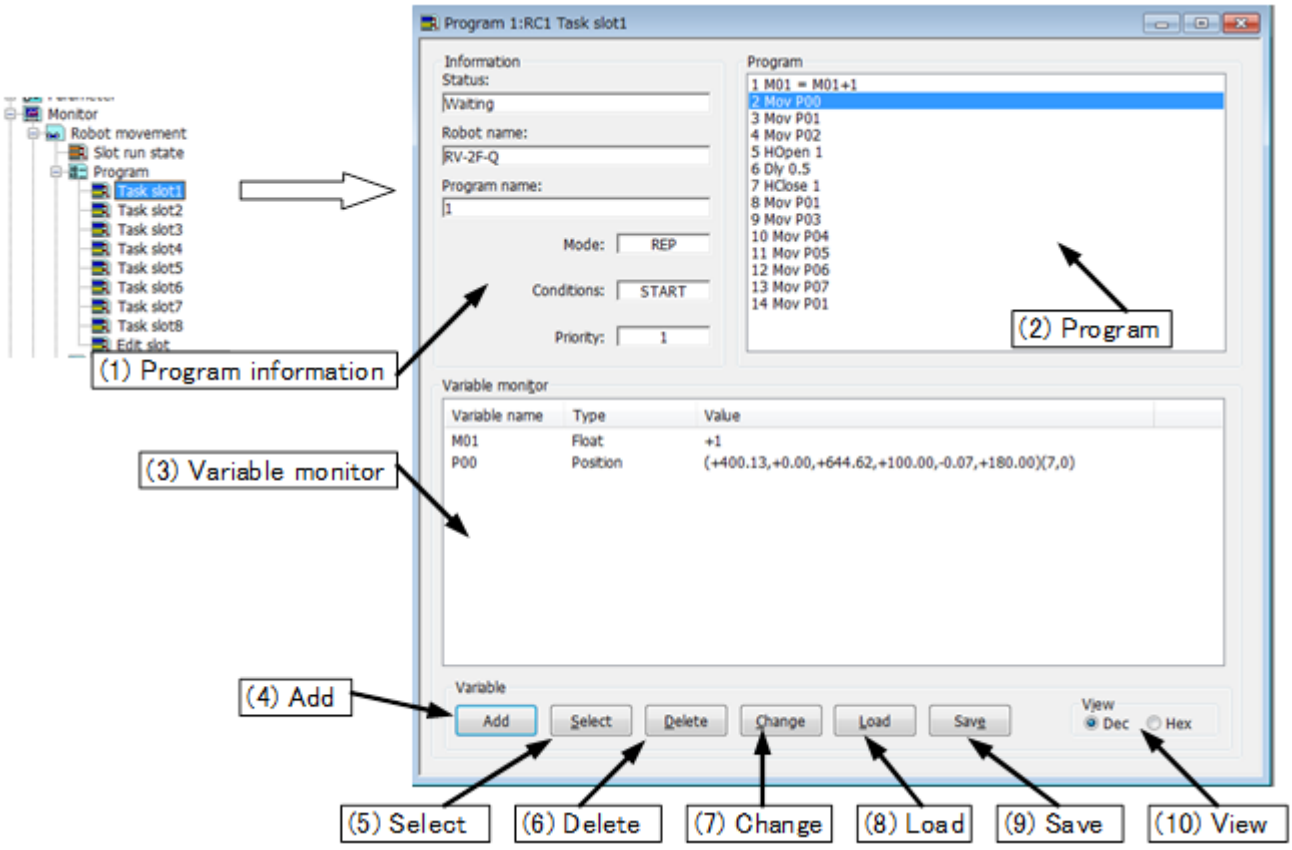
Move a robot arm to the correct location

Specify the correct location of 5-8 points as the "length" made to this work by the one of the precision of the tracking function.

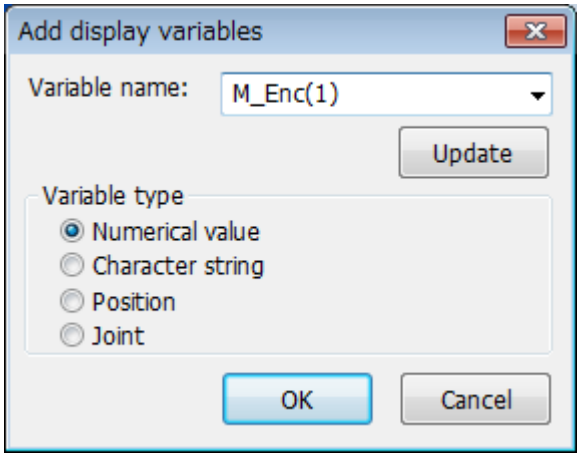
8.1.2. Confirm the encoder value

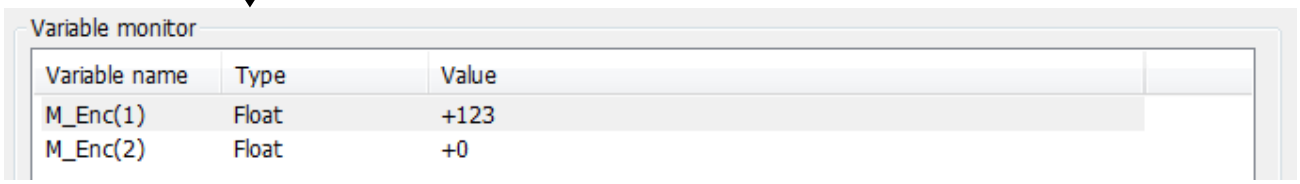
An important one is a change in the encoder value in this work. Confirm whether a robot controller grasps the turn of the encoder.

From the project tree, click the target project [Online] -> [Monitor] -> [Movement Monitor] -> [Program Monitor], then double click the "Task slot" to monitor.

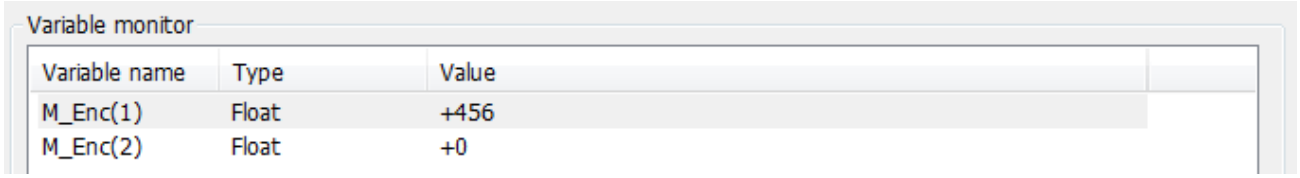


Click a [Add] button and open a "Add display variables" screen. Input "M_Enc (1)" to a space "variable name", and click a [OK] button. also input "M_Enc (2)"-"M_Enc (8)" equally, and click a [OK] button.





Confirm that the value of "M_Enc" changes by a revolution of a turntable.



When the encoder value doesn't change, confirm the parameter setting and the wiring of "6.1.2 Robot Parameter Setting".

8.1.3. Knowledge about work

This chapter explains below about the contents it's necessary to know before this work.

On the turntable, decide the area where the robot starting tracking (Tracking starting possible area) and the area where a robot can continue tracking a workpiece (Tracking area).

[Tracking starting possible area]

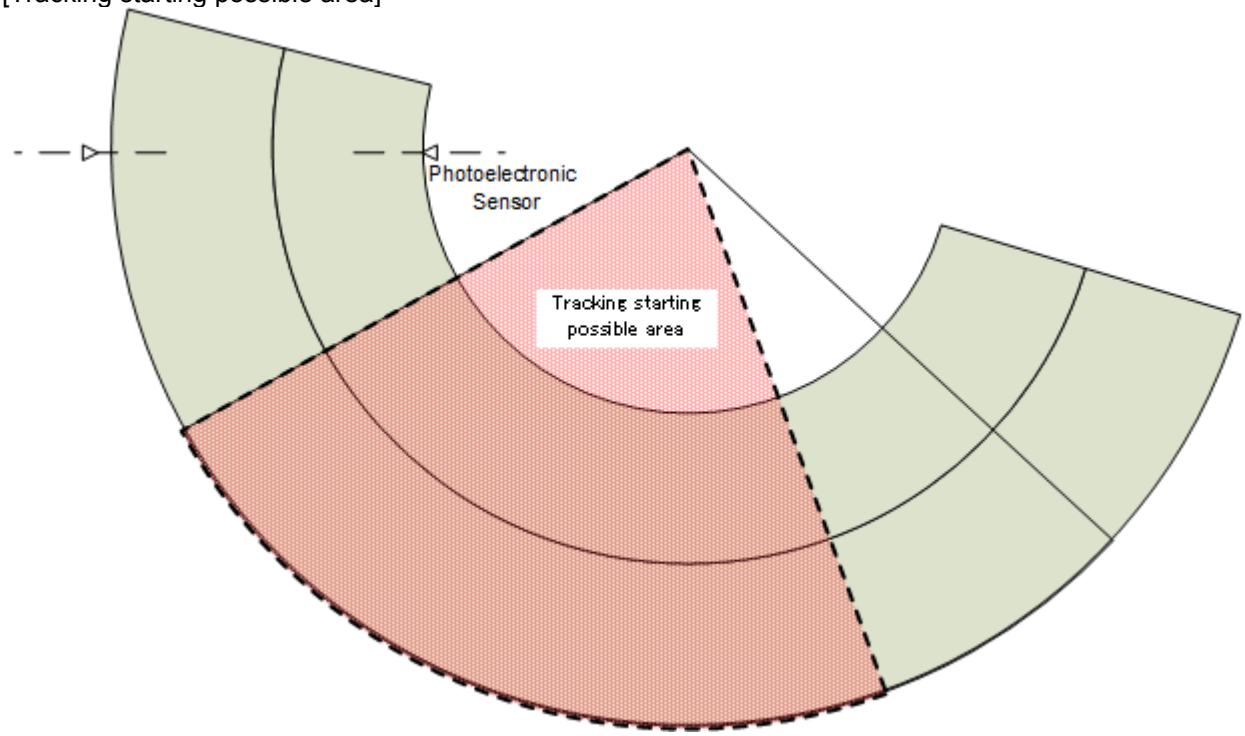


Figure 8-1 Tracking starting possible area

[Tracking area]

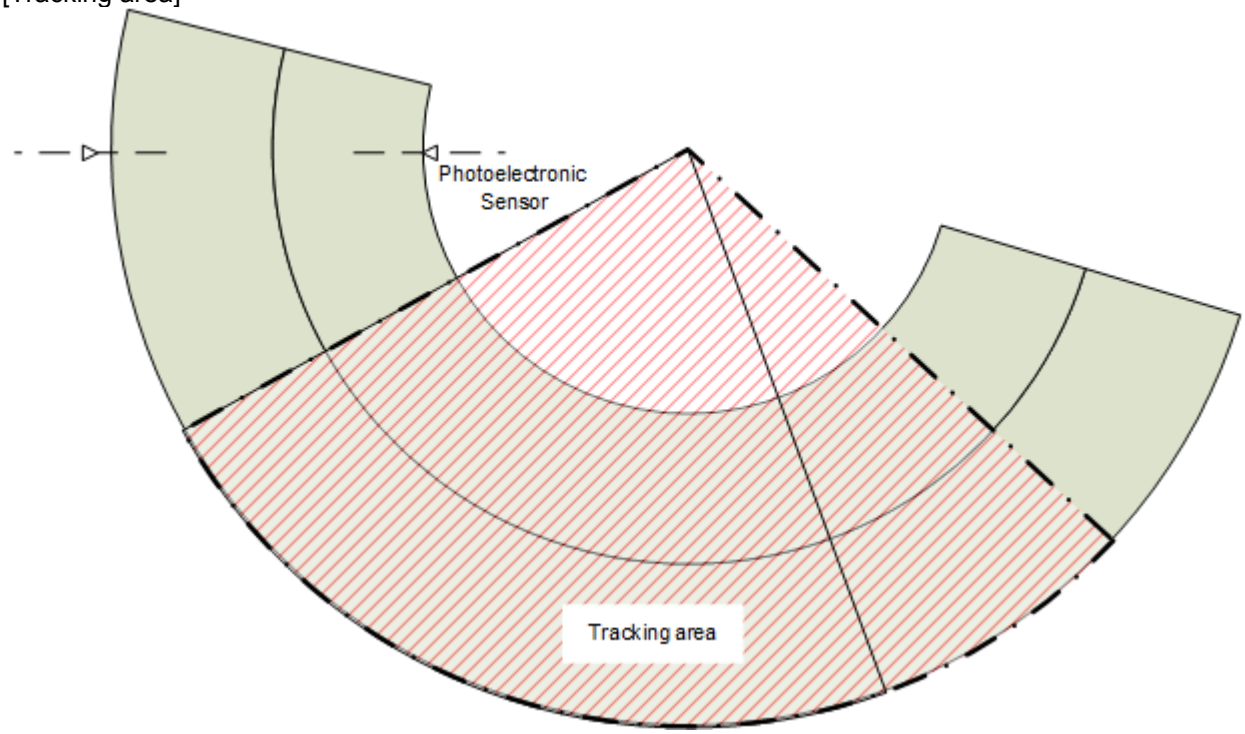


Figure 8-2 Tracking area

8.2. Operation procedure

Using "A1" program, operate in the following procedures.

(1) Exchange it for a use hand from a hand for tool setting.
For example change it to the following hand.

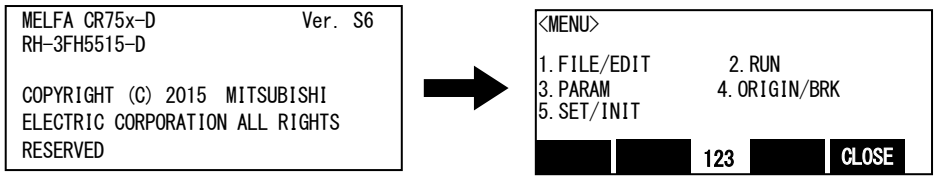
(2) Set the controller mode to "MANUAL". Set the T/B to "ENABLE".

O/P T/B

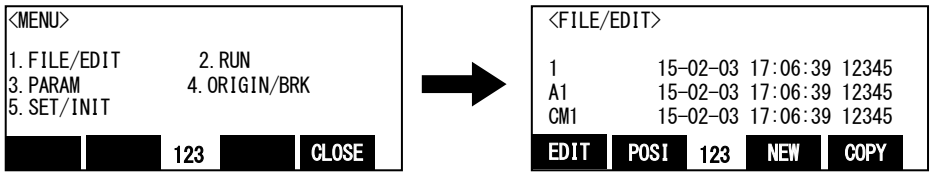
↑上:DISABLE
↓下:ENABLE
*Lamp lighting

T/B rear

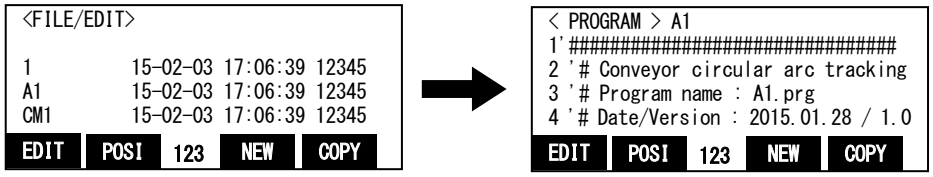
(3) Press one of the keys (example, [EXE] key) while the <TITLE> screen is displayed. The <MENU> screen will appear.



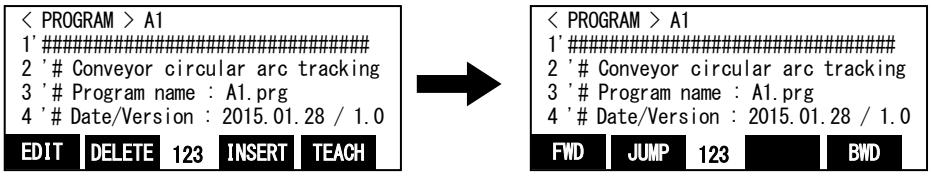
(4) Select "1. FILE /EDIT" screen on the <MENU> screen.



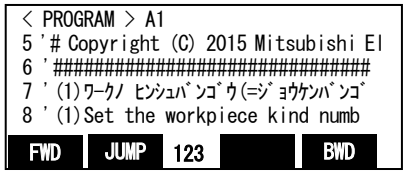
(5) Press the arrow key, combine the cursor with the program name "A1" and press the [EXE] key. Display the <program edit> screen.



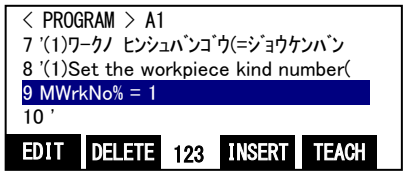
(6) Press the [FUNCTION] key, and change the function display



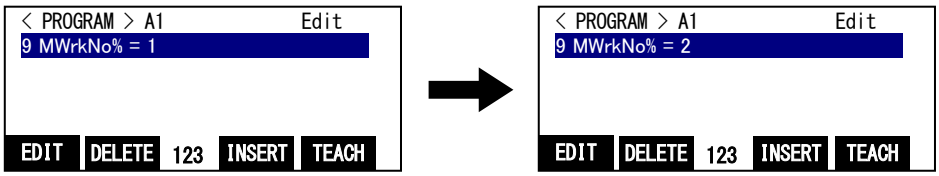
(7) Press the [F1] (FWD) key and execute step feed. "(1)Set the workpiece kind number(= condition number) to the variable "MWrkNo""is displayed. Execute work according to the comment in the robot program.



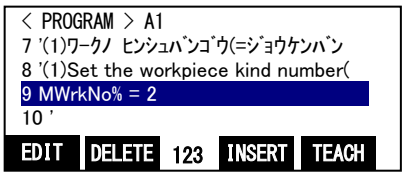
Here, specify the condition number.
If you want to change the condition number, please edit the program as follows.
(a) Display the following Statement.



(b) Press the [F1] (EDIT) key, and then specify the condition number in the variable "MWrkNo%".
Example) specify a "2"



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

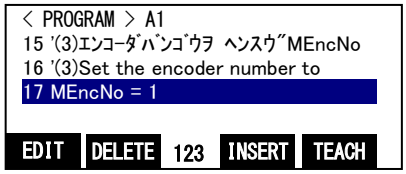


(8) Press the [F1] (FWD) key and execute step feed. "(2)Confirm that Tool is set"is displayed.

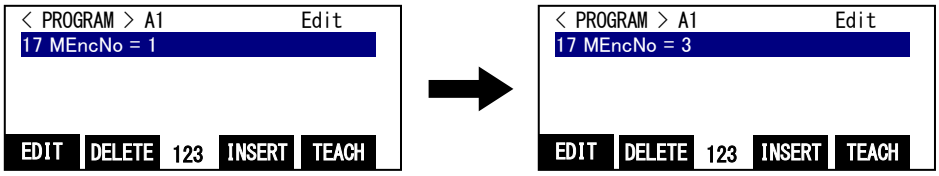
If you are able to set the tool length, or if the setting is not required in "6.2 Operation Parameters", go to the next.
If you need to set, set the tool length, refer to "6.2 Operation Parameters".

(9) Press the [F1] (FWD) key and execute step feed. "(3)Set the encoder number to the variable "MEncNo""is displayed.
Here, specify the encoder number.
If you want to change the encoder number, please edit the program as follows.

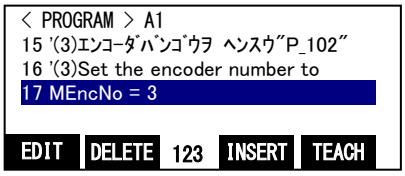
(a) Display the following Statement.



(b) Press the [F1] (EDIT) key, and then specify the encoder number in the variable "MEncNo".
Example) specify a "3"



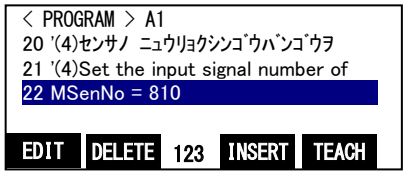
(c) Press the [EXE] key, accept the changes.



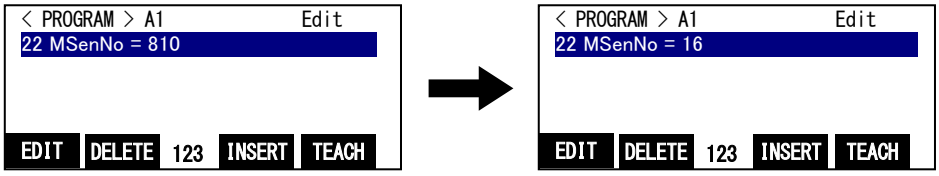
(10) Press the [F1] (FWD) key and execute step feed. "(4)Set the input signal number of sensor to the variable "MSenNo""is displayed.

Here, specify the signal number of photoelectronic sensor.
If you want to change the signal number, please edit the program as follows.

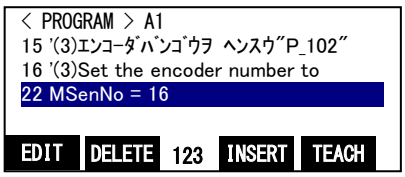
(a) Display the following Statement.



(b) Press the [F1] (EDIT) key, and then specify the encoder number in the variable "MSenNo".
Example) specify a "16"

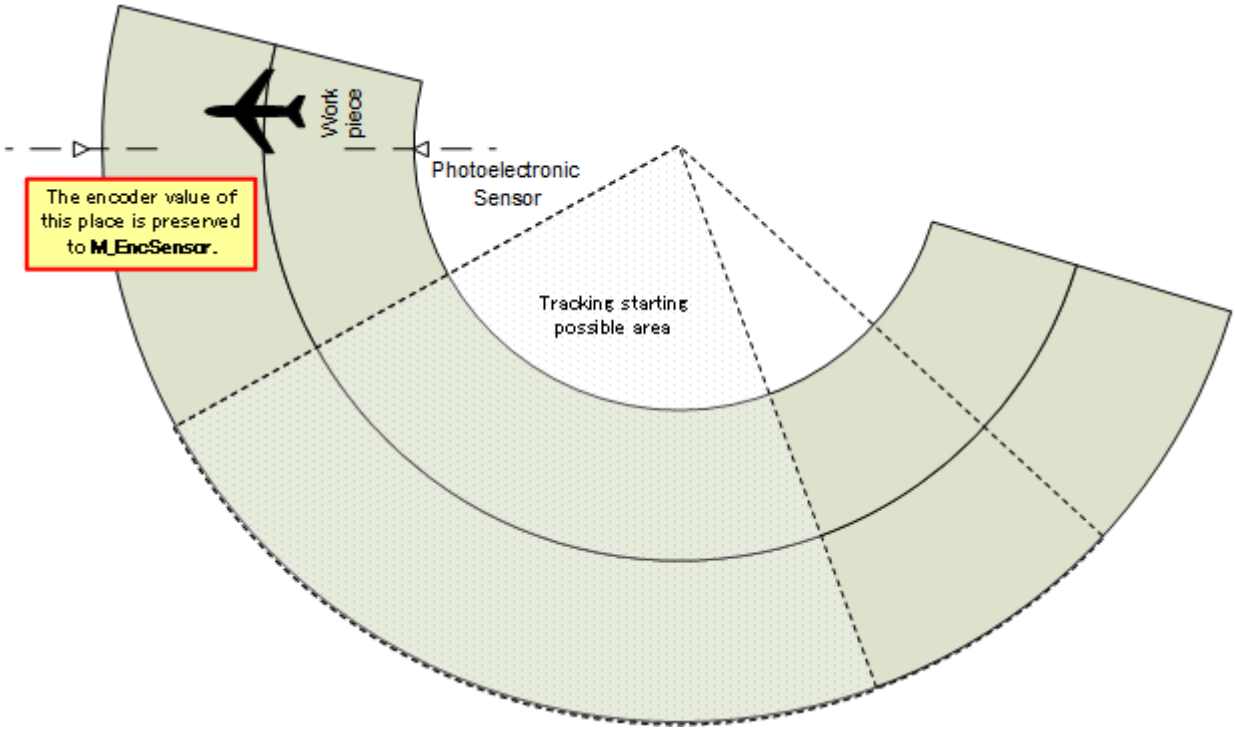


(c) Press the [EXE] key, accept the changes.



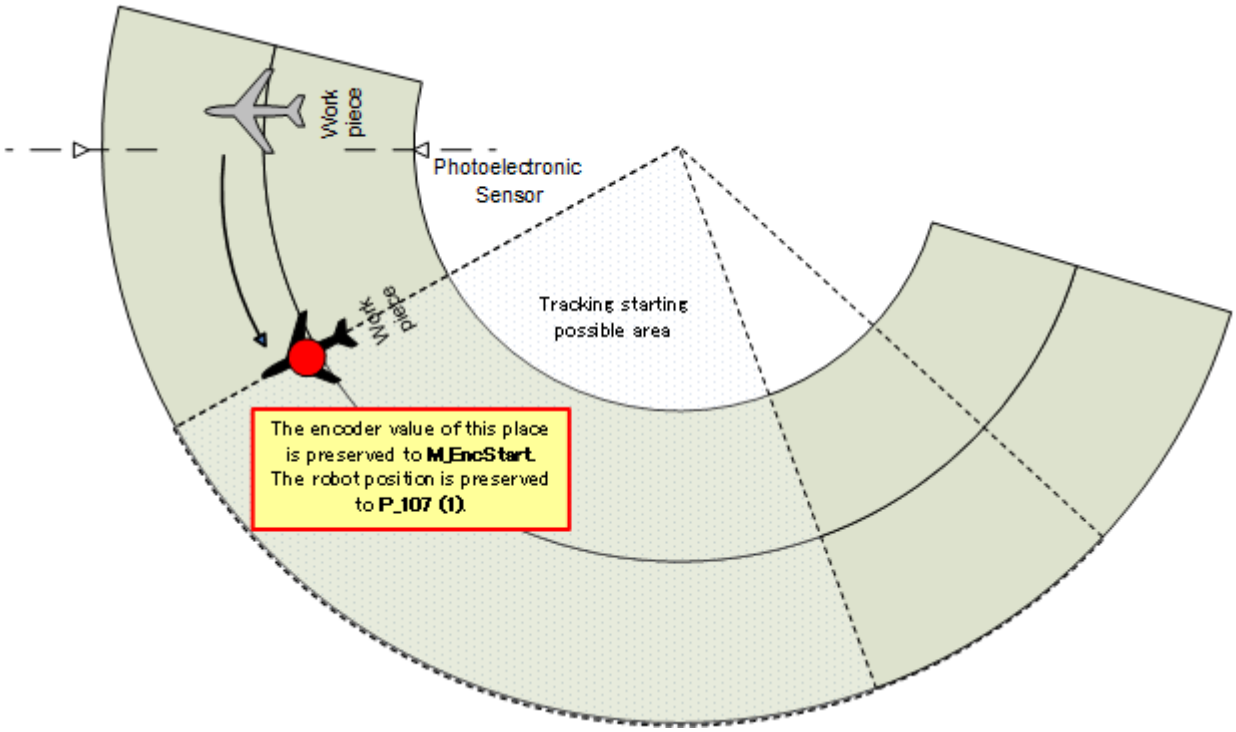
(11) Press the [F1] (FWD) key and execute step feed. "(5)Put workpiece on the sensor position of the conveyor"is displayed.

Move the turntable, and place the workpiece at a position where photoelectronic sensor reacts.



(12) Press the [F1] (FWD) key and execute step feed. "(6)Move workpiece to the tracking area start position by conveyor"is displayed.

Move the turntable, and place the workpiece in position to enter the area where the robot starting tracking (start position of the Tracking starting possible area).

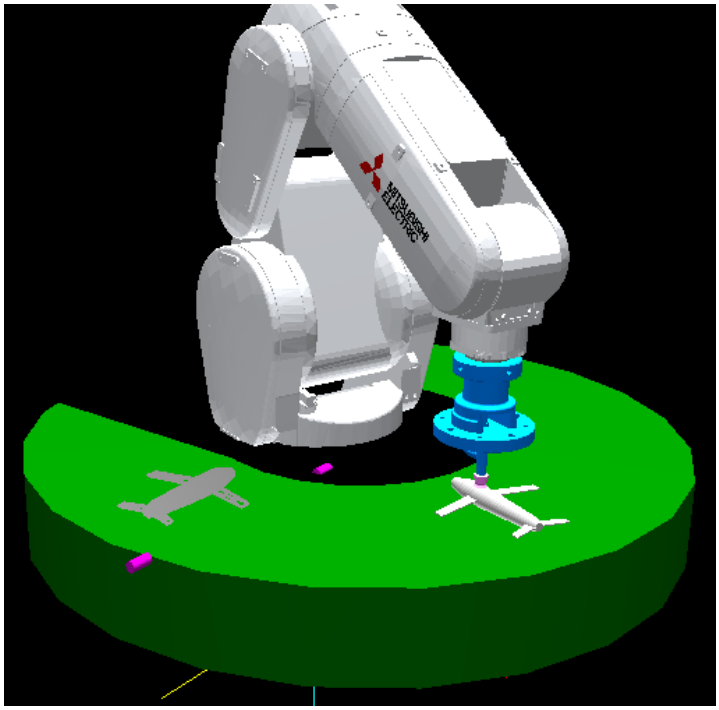


(13) Press the [F1] (FWD) key and execute step feed. "(7)Move the robot to the adsorption point of workpiece"is displayed.

Move the robot arm to adsorption position (or initial position to be processed) of the workpiece in the Tracking starting possible area.

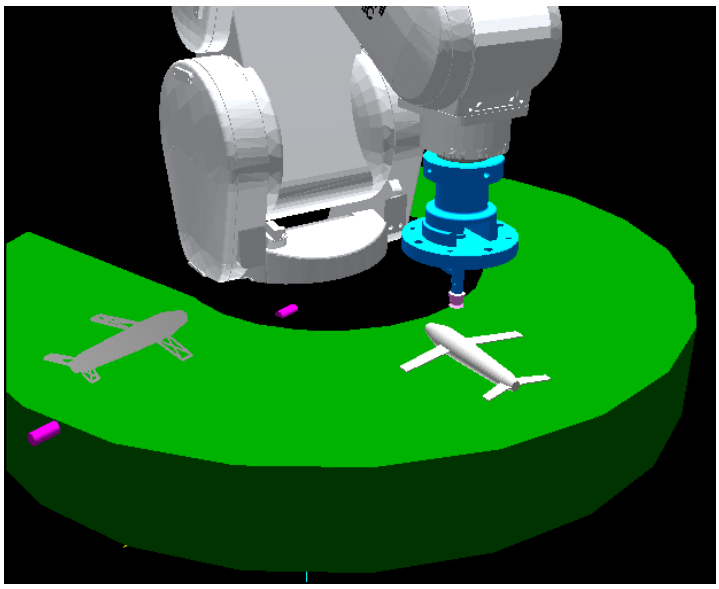
POINT

Please work so that it is not to move the workpiece.
In future work, The workpiece on the turntable is used.
If the workpiece are shifted, you need to redo the work again.



This position will be taught in **P_107 (1)**.

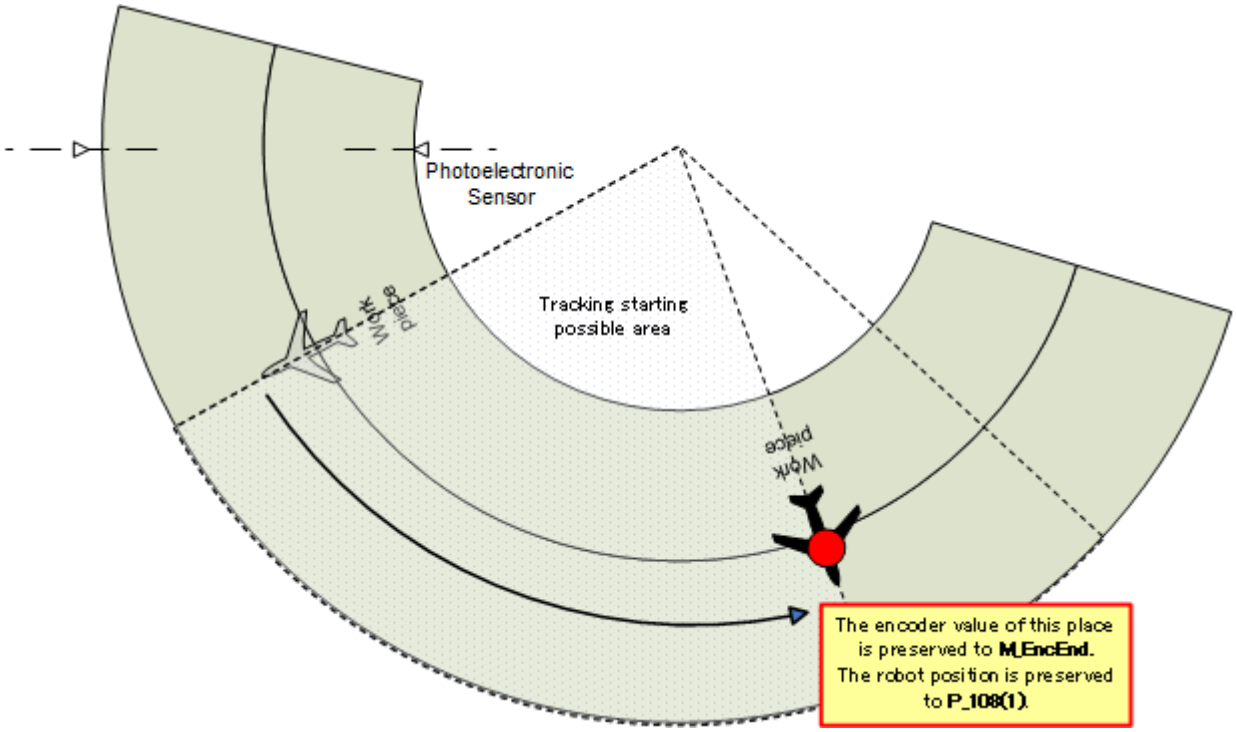
(14) Press the [F1] (FWD) key and execute step feed. "(8)Move the robot to the waiting position(= home position)"is displayed.
Raise the robot arm to work without adsorption, specify the retracted position of the automatic operation start(Start position) and the standby position to wait for workpiece.
Please decide the amount of increase, depending on the system.



These positions will be taught in **P_103(1)** and **P_104(1)**

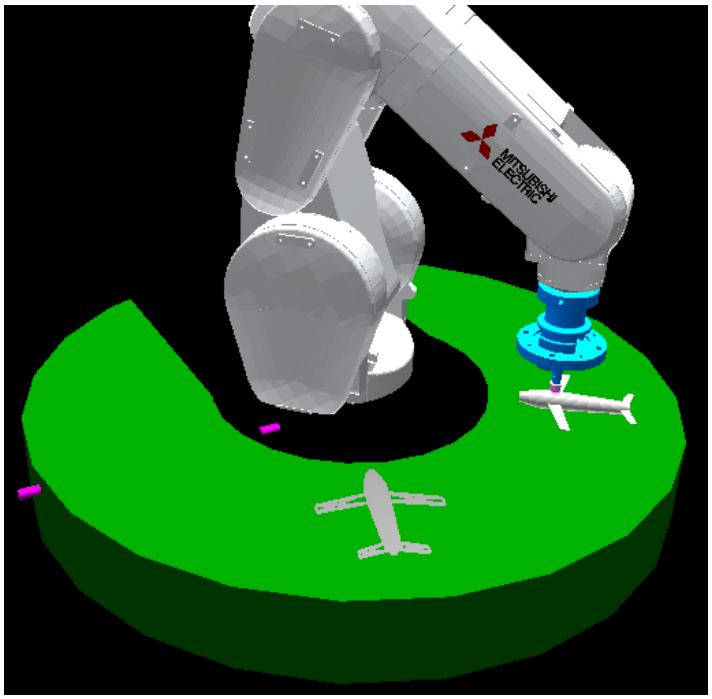
(15) Press the [F1] (FWD) key and execute step feed. "(9)Move workpiece to the tracking area end position by conveyor"is displayed.

Move the turntable, and place the workpiece at the end position of the Tracking starting possible area.



(16) Press the [F1] (FWD) key and execute step feed. "(10)Move the robot to the adsorption point of workpiece"is displayed.

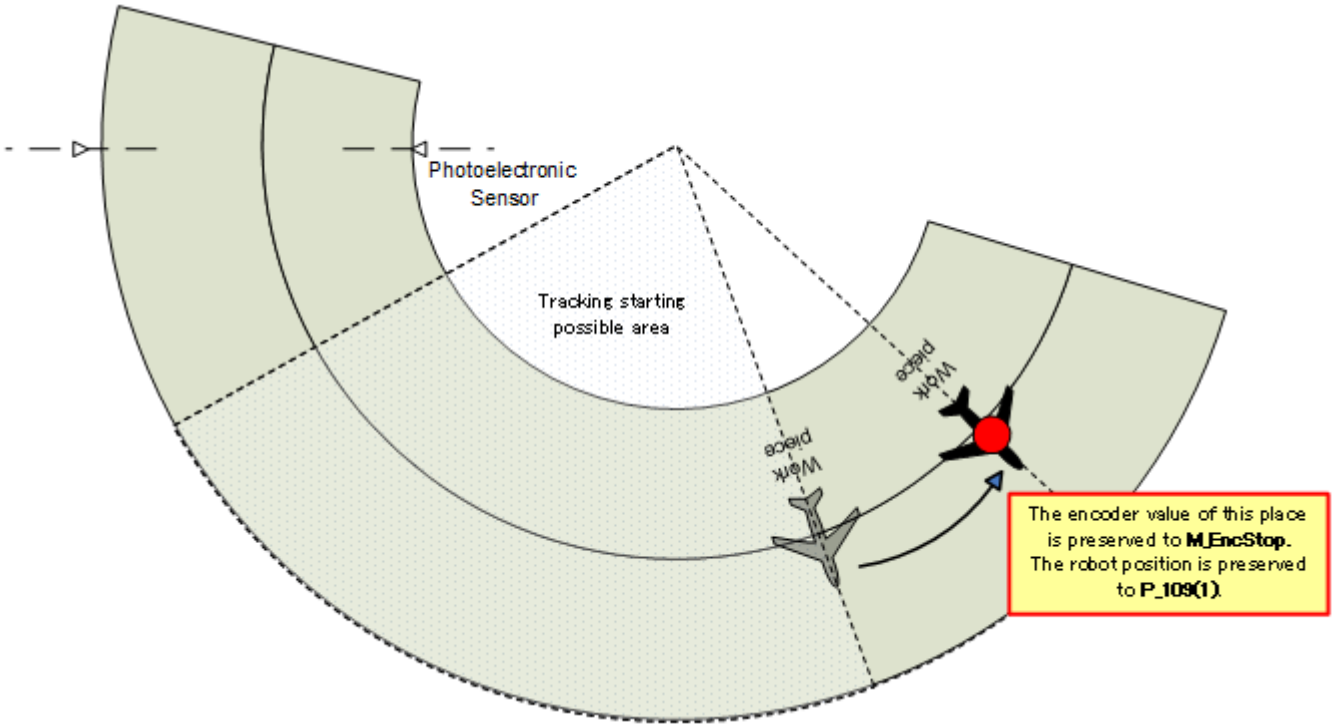
Against the workpiece at the end position of the Tracking starting possible area, and then move the robot arm to the position of the same conditions as when it was taught in the start position of the Tracking starting possible area.



This position will be taught in P_108(1)

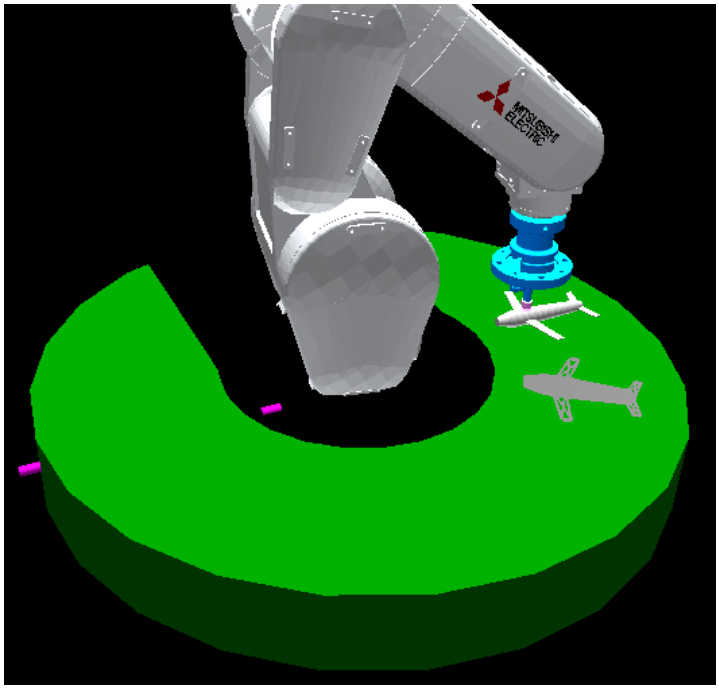
(17) Press the [F1] (FWD) key and execute step feed. "(11)Move workpiece to the tracking cancellation position by conveyor"is displayed.

Move the turntable, and place the workpiece at the position to forcibly terminate the tracking.



(18) Press the [F1] (FWD) key and execute step feed. "(12)Move the robot to the adsorption point of workpiece"is displayed.

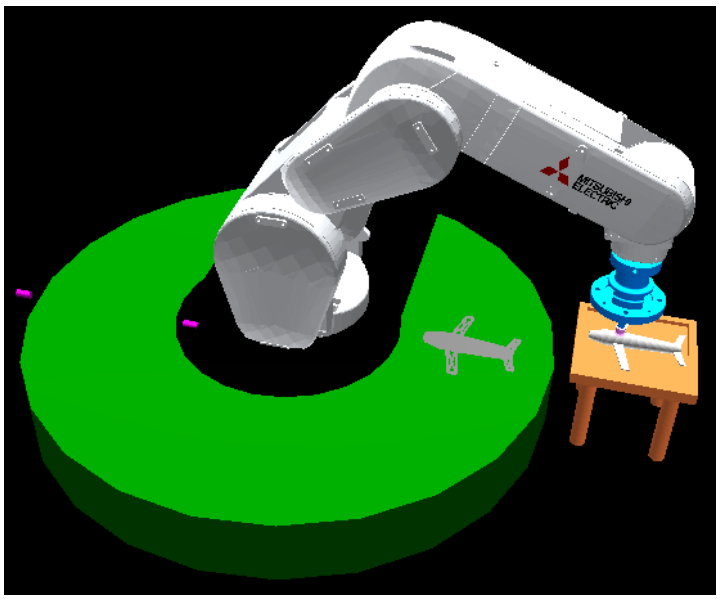
Against the workpiece at the tracking cancellation position, and then move the robot arm to the position of the same conditions as when it was taught in the start position of the Tracking starting possible area.



This position will be taught in P_109(1)

(19) Press the [F1] (FWD) key and execute step feed. "(13)Absorb a workpiece. And move to the transportation position."is displayed.

Move the robot arm to a position to transport the adsorbed workpiece from the turntable (Transport destination).



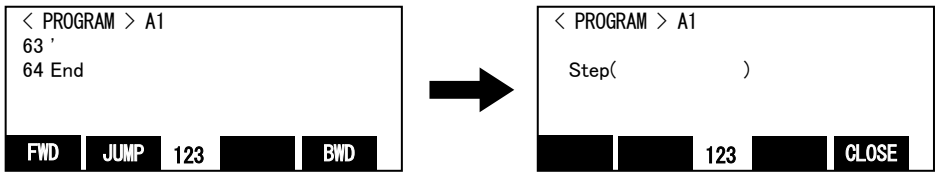
This position will be taught in **P_105(1)**

Press the [F1] (FWD) key and execute step feed. "End"is displayed.

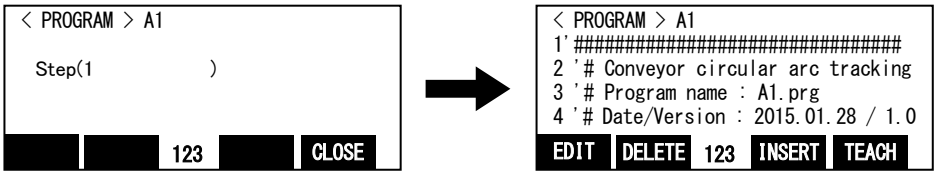
(20) Press the [F1] (FWD) key and execute step feed. "End"is displayed. Work is now completed, but in case you want to perform the work by the side "A1" program, and then save the program in the state in which to display the first line.

Return a program to the first line and save it as follows.

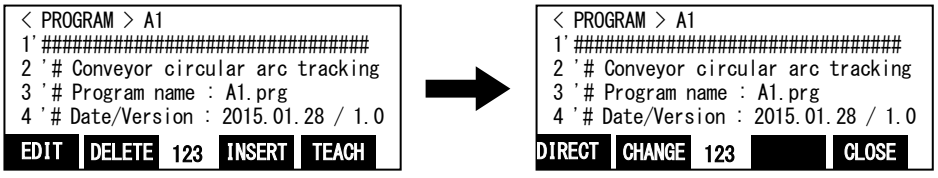
(a) Press the [F2] (JUMP) key



(b) input the step number. Press the [EXE] key. Then returns to first step



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



8.3. What to confirm

Confirm that the following data is remembered after work.

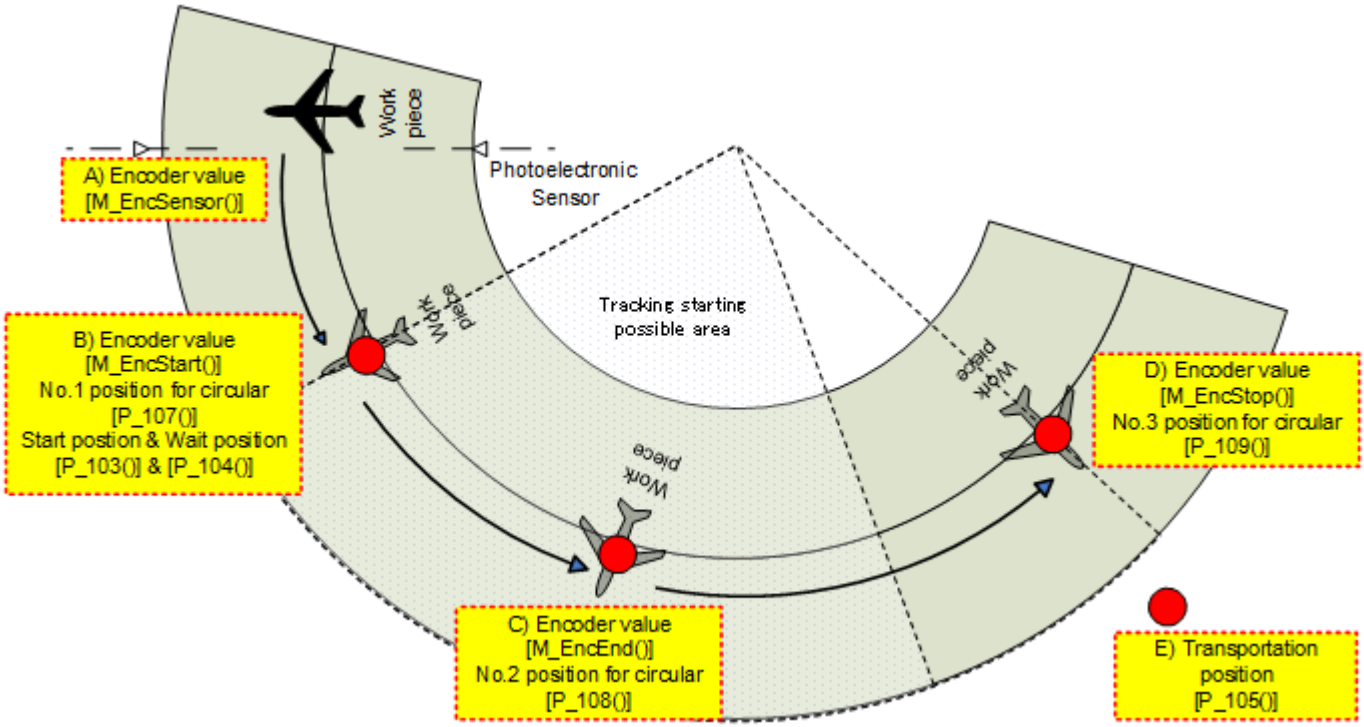


Table8-1 Overall picture of the teachings

Confirm that the following variable includes the price using the variable monitor of RT ToolBox2 in confirmation of data.

Variable monitor

Variable name	Type	Value
M_EncEnd(1)	Float	11158
M_EncSensor(1)	Float	1842
M_EncStart(1)	Float	6994
M_EncStop(1)	Float	13902
P_103(1)	Position	(+337.00,+133.15,+469.93,-179.96,-0.06,+125.48,+0.00,+0.00)(7...
P_104(1)	Position	(+337.00,+133.15,+469.93,-179.96,-0.06,+125.48,+0.00,+0.00)(7...
P_105(1)	Position	(+148.66,+135.64,+433.49,-180.00,-0.00,+180.00,+0.00,+0.00)(7...
P_107(1)	Position	(+337.00,+133.15,+439.93,-179.96,-0.06,+125.48,+0.00,+0.00)(7...
P_108(1)	Position	(+317.79,+20.76,+439.93,-179.96,-0.06,+125.48,+0.00,+0.00)(7,0)
P_109(1)	Position	(+390.03,-12.72,+439.93,-179.96,-0.06,+125.48,+0.00,+0.00)(7,0)

8.4. When multiple conveyers and turntables 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 3 (encoder number "3"), kind number "2", signal number of photoelectronic sensor "16":

(a) Copy the "A1" program, please create a "A2" program.
(b) If you want to change the tool length, please change the tool length in advance.
(c) Please change the kind number for variable "MWrkNo" in the "A2" program to "2".
(d) Please change the encoder number for variable "MWrkNo" in the "A2" program to "3".
(e) Please change the signal number for variable "MWrkNo" in the "A2" program to "16".

9. Setting of an operating condition and operations check ("1"Program)

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

9.1. Variable for operating conditions

The variable indicated below is used for designation of an operating condition of a robot.
Please refer to "Detailed Explanations of Functions and Operations" for how to set the variable.

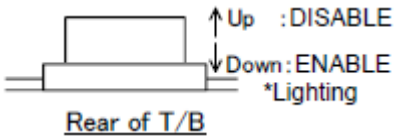
Table 9-1 List of variable for operating conditions

Variable name	Explanation	Setting Example
PUp1	In movement to adsorb workpiece, appoint quantity of offset to the sky position. ***For this variable to be calculated relatively to the adhesion location, it's necessary to pay attention to a sign according to the model of the robot.**	If you want to rise 30mm from position to adsorb the work (Example)RV series (X,Y,Z,A,B,C)= (+0, +0,-30,+0,+0,+0) (Example) Other than RV series (X,Y,Z,A,B,C)= (+0,+0,+30,+0,+0,+0)
PUp2	In movement to desorb workpiece, appoint quantity of offset to the sky position. ***For this variable to be calculated relatively to the adhesion location, it's necessary to pay attention to a sign according to the model of the robot.**	If you want to rise 50mm from position to desorb the work (Example)RV series (X,Y,Z,A,B,C)= (+0, +0,-50,+0,+0,+0) (Example) Other than RV series (X,Y,Z,A,B,C)= (+0,+0,+50,+0,+0,+0)
PDly1	Specify the adsorption time (s).	Specify the adsorption time to 1 second: (X,Y,Z,A,B,C)= (+1,+0,+0,+0,+0,+0)
PDly2	Specify the desorption time (s).	Specify the desorption time to 1 second: (X,Y,Z,A,B,C)= (+0.5,+0,+0,+0,+0,+0)
PPri	When you start the "1" program, in slot 2 of a multi-task, "CM1" program is executed. "1" monitors the operation of the robot, "CM1" monitors the sensor. You can specify whether to prioritize either of processing. X coordinate ＝ The number of execution line of the "1"program(1 - 31) Y coordinate ＝ The number of execution line of the "CM1" program (1 - 31)	While execute 1 line of "CM1", execute 3 lines of "1". (X,Y,Z,A,B,C)= (+3,+1,+0,+0,+0,+0)

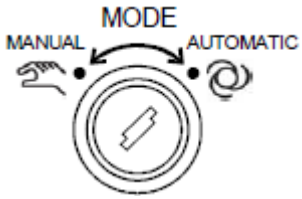
9.2. Automatic operation

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

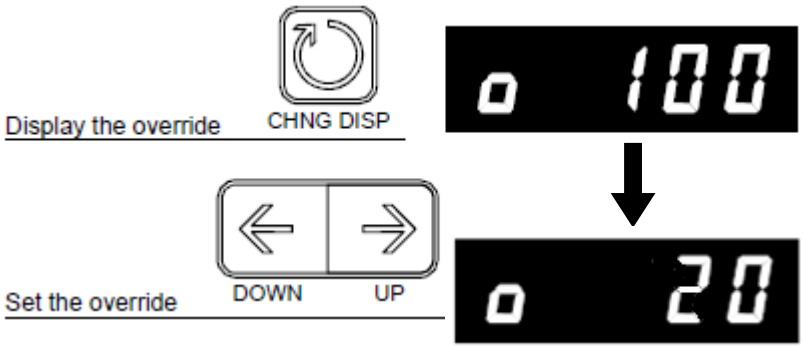
(1) Confirm that there isn't an intervention thing in the robot movement area.
(2) Set the T/B [ENABLE] switch to "DISABLE"



(3) Set the controller mode to "AUTOMATIC".

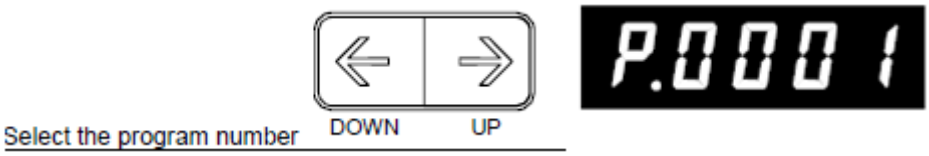


(4) Press the controller [CHNG DISP] button twice, and display the "OVERRIDE" on the STATUS NUMBER display panel, and specify the override to 20% - 30%.



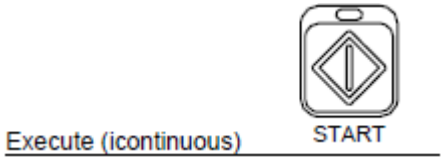
(5) Press the [CHNG DISP] button and display "PROGRAM NO." on the STATUS NUMBER display. Then press the [RESET] button to reset program.

(6) Press the [UP] key or the [DOWN] button and display "program 1" to the STATUS NUMBER display.

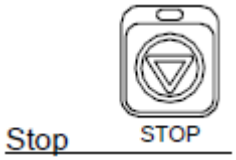


(7) Automatic operation will start when the controller [START] button is pressed.

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



(8) When the robot moves to the specified retracted position, to drive the turntable and place the workpiece.
(9) Confirm to be a work that is unloaded to the transport destination after following the workpiece.
(10)If you check the operation, press the [STOP] button and stop the robot.



POINT

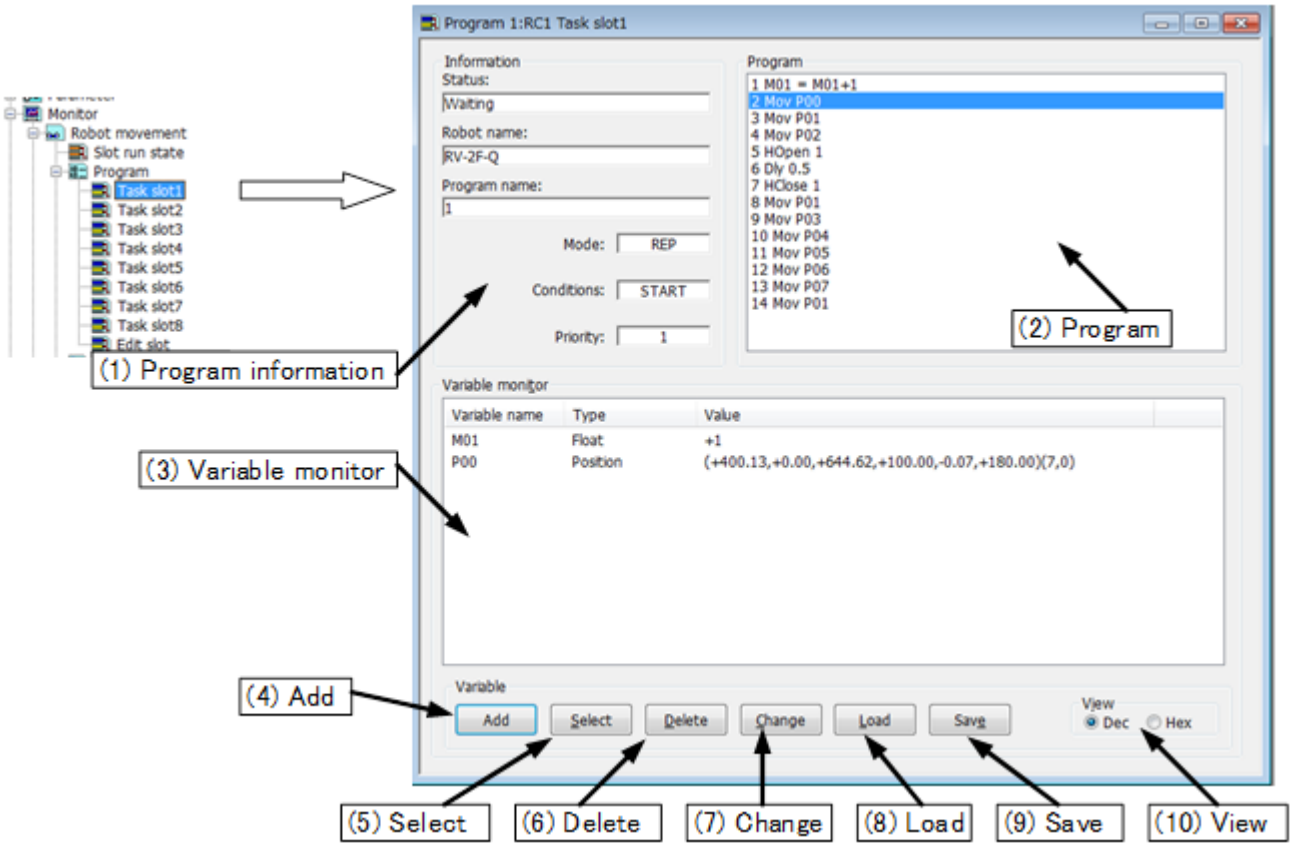
T / B software in a specific version or later, you can be the automatic operation from T / B.

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

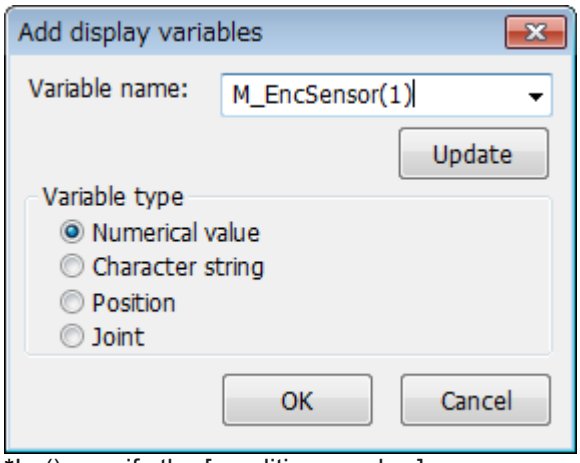
9.3. Adjustment of the follow position

When driving a turntable, the position where photoelectronic sensor reacts to a workpiece is different from the set position in "A1" program.
Therefore, after determining the rotation speed of the turntable, you have to adjust the position with the following procedure.

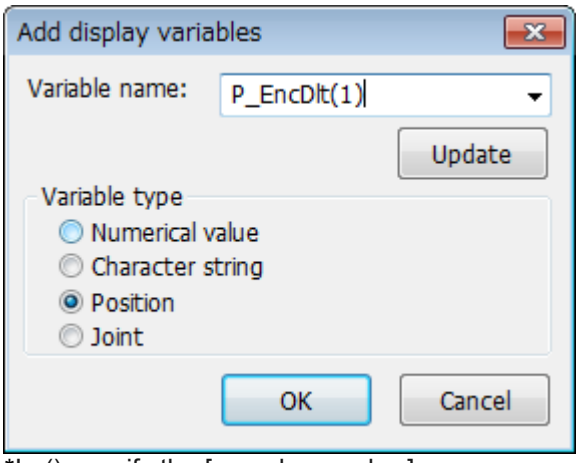
(1) Start the "Program monitor" of RT ToolBox2.



(2) Click a [Add] button and open a "Add display variables" screen. Input "M_EncSensor(1)" to a space "variable name", and click a [OK] button. also input " P_EncDlt(1)" equally, and click a [OK] button.

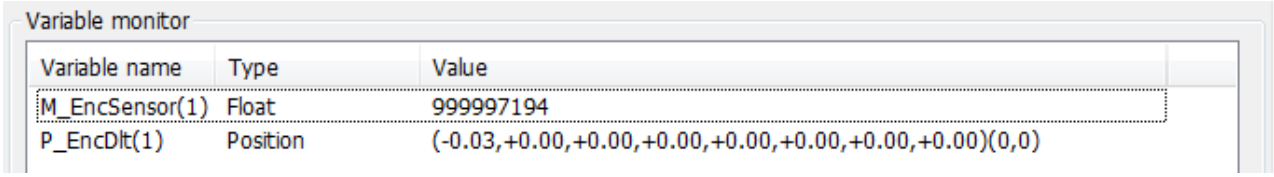


*In (),specify the [condition number].



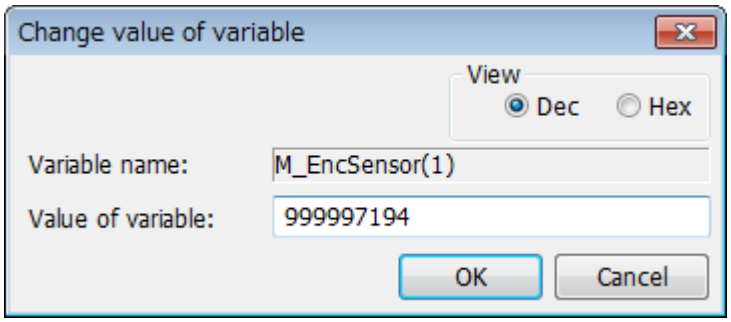
*In (),specify the [encoder number].

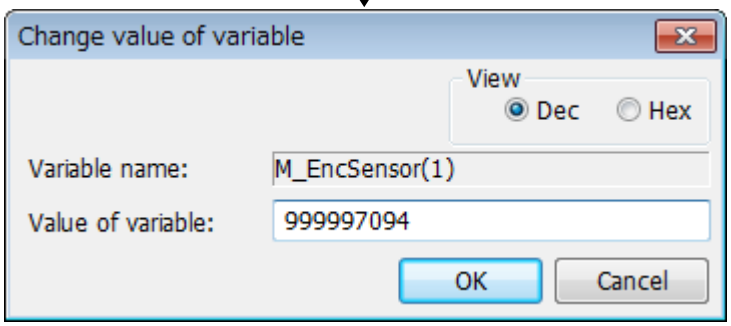
(3) Confirm that the value of the specified variable is displayed in the "Variable monitor".
Displayed "M_EncSensor (1)" is the encoder value when the photoelectronic sensor has reacted to the workpiece.
Displayed "P_EncDlt (1)" indicates the distance from which a workpiece moves on the circumference every 1 pulse.



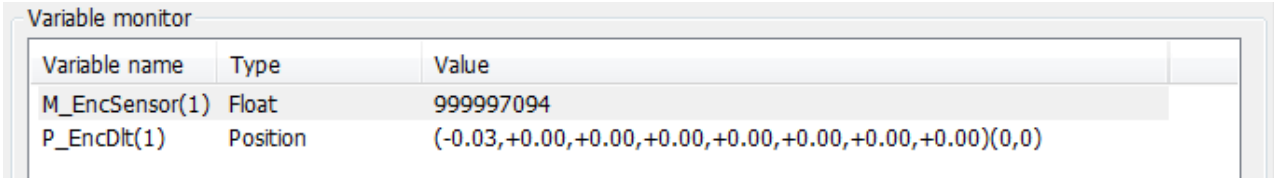
The direction of rotation of the turntable is assumed "plus direction".
If you want to correct 3mm of robot arm in the plus direction, you reduce the value of "M_EncSensor (1)" 100 pulses.

(4) Double-click the "M_EncSensor (1)", and change the number of displayed "value of variable" column.
*In (), specify the [condition number].





(5) Click [OK] button, and confirm that the value of "M_EncSensor (1)" displayed in the "Variable Monitor" has been changed.



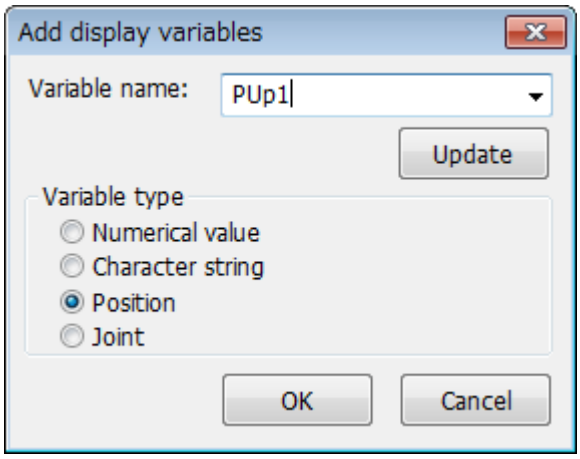
(6) Return to the "9.2 Automatic operation", and then check to see whether the can be corrected by implementing the automatic operation.

9.4. Adjustment of operating conditions

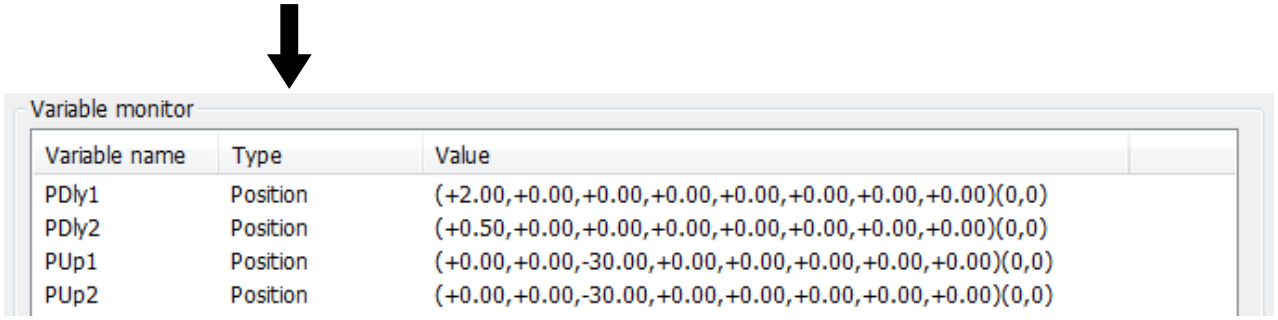
In automatic operation, if you want to adjust the vertical movement and adsorption time of the robot arm that was described in "9.1 Variable for operating conditions" should be changed in the following procedure.

(1) Start the "Program monitor" of RT ToolBox2.

(2) Click the [Add] button and open the "Add display variables" screen. Enter the variables listed in the "Table 9-1 List of variable for operating conditions", and then click the [OK] button.

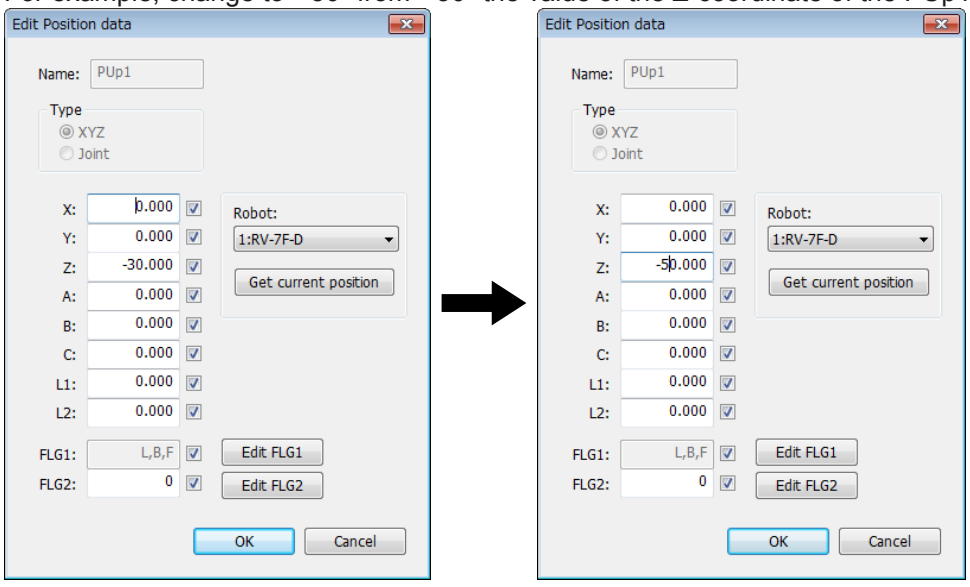


Others, "PUp2", "PDly1", "PDly2" etc.

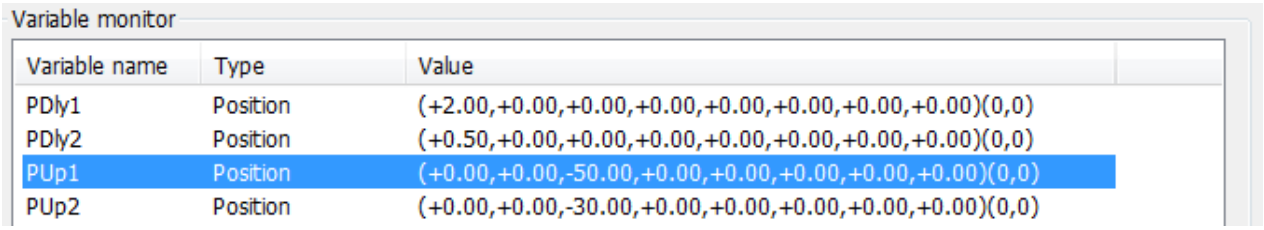


(3) Double-click the variable you want to change, and change the appropriate value for displayed in the "Edit Position data".

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



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


Variable monitor

Variable name	Type	Value
PDly1	Position	(+2.00,+0.00,+0.00,+0.00,+0.00,+0.00,+0.00,+0.00)(0,0)
PDly2	Position	(+0.50,+0.00,+0.00,+0.00,+0.00,+0.00,+0.00,+0.00)(0,0)
PUp1	Position	(+0.00,+0.00,-50.00,+0.00,+0.00,+0.00,+0.00,+0.00)(0,0)
PUp2	Position	(+0.00,+0.00,-30.00,+0.00,+0.00,+0.00,+0.00,+0.00)(0,0)

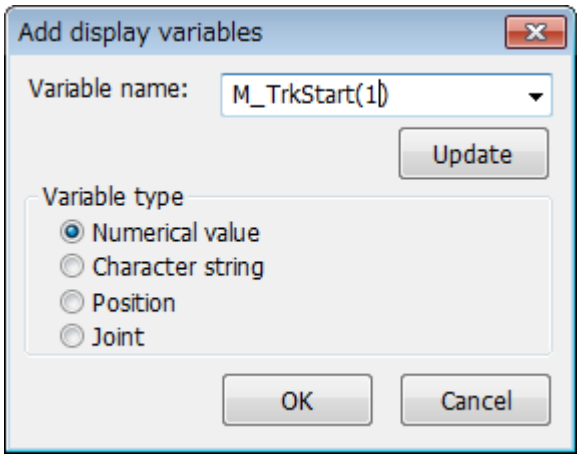
(5) Return to the "9.2 Automatic operation", and then check to see whether the can be corrected by implementing the automatic operation.

9.5. Adjustment of Tracking starting possible area

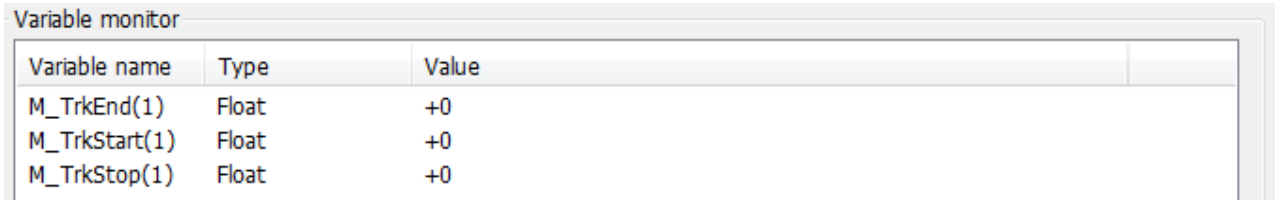
In automatic operation, if you want to adjust the Tracking starting possible area that was taught in the "8 Teaching Operation(“A1” Program)", change the following procedure.

(1) Start the "Program monitor" of RT ToolBox2.

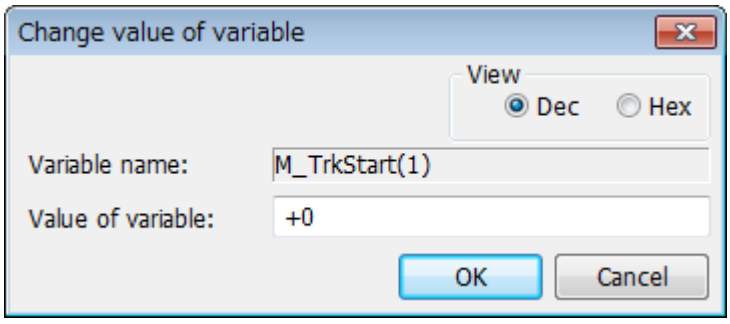
(2) Click the [Add] button and open the "Add display variables" screen. Enter the following three state variables, and then click the [OK] button.



Others, “M_TrkEnd(1)”, “M_TrkStop(2)” *In (), specify the [condition number].

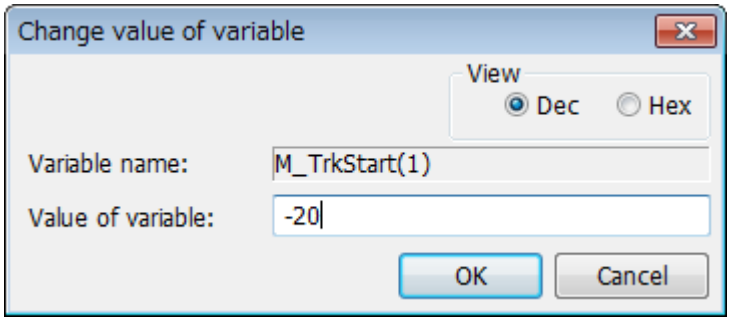


(3) Double-click the variable you want to change, and change the value in the displayed "Changing Values" screen.



Assume that the traveling direction of the turntable "plus", enter the distance you want to correct, and then click [OK].

For example, if you want the tracking started early 20mm:



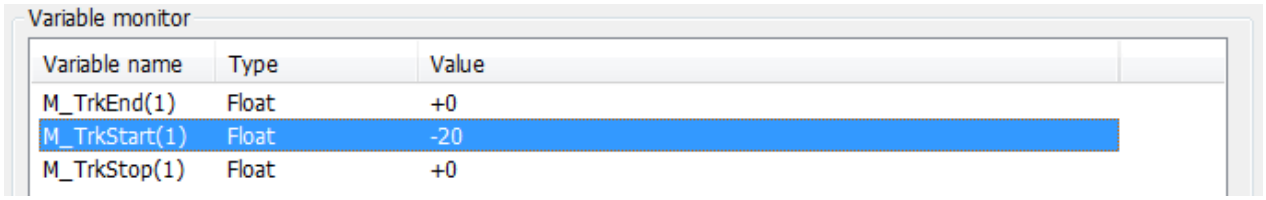
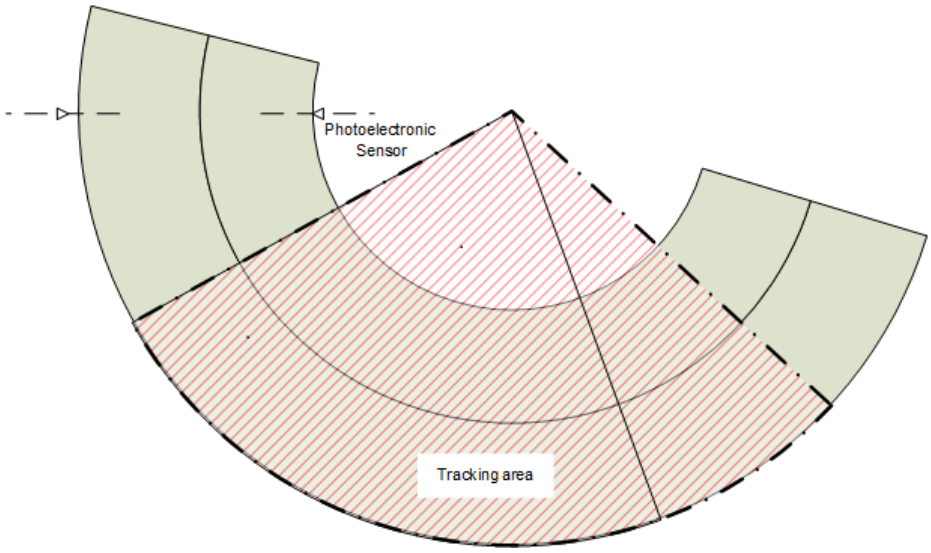
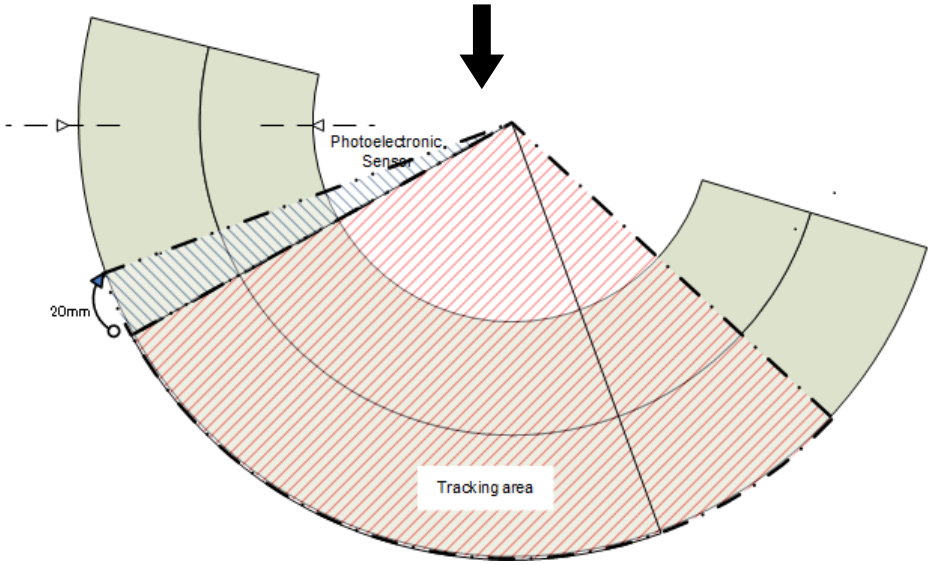


Image of the tracking area is as follows.





(4) Similarly, please adjust using the "M_TrkEnd" for the end position of the tracking starting possible area. Also, please adjust using the "M_TrkStop" for the position to be forcibly terminated.

9.6. Occurrence of error

When an error occurred, please confirm the "11 Troubleshooting".

10. Maintenance of robot program

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

10.1. MELFA-BASIC V Instructions

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

10.1.1. List of Instructions

Table **10-1** **List of Instructions**

Instruction name	Function
TrClr	Clear the tracking data buffer.
TrWrt	Write workpiece data in the tracking data buffer.
TrkArc	Set the information of the arc conveyer.
TrkChk	Execute the processing depending on the state of workpiece corresponding to <Condition number> specified.
TrkWait	Wait until workpiece corresponding to <Condition number> specified enters to the tracking area.
TrkMv	Execute the next processing. Validate specified interruption, Start tracking, Move to the tracking upper position by Joint interpolation movement.

10.1.2. List of Robot Status Variables

Table **10-2** **List of Robot Status Variables**

Variable name	Number of arrays	Function	Attribute (*1)	Data type
M_Enc	Number of encoder 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 “0".**	Read/Write	Double-precision real number
M_EncL	Number of encoder 1 to 8	The stored encoder data ※ **0 always returns in D-type.**	Read/Write	Double-precision real number
P_EncDlt	Number of encoder 1 to 8	Amount of robot movement per encoder pulse	Read/Write	Position
P_TrkSensor	Condition Number 1 to 8.	The location of the workpiece when a sensor reacted	Read Only	Position
M_EncSensor	Condition Number 1 to 8.	The encoder data at the position in which the sensor reacts to workpiece. Possible to change the value to adjust it.	Read/Write	Long-precision real number
M_EncStart	Condition Number 1 to 8.	The encoder data at start position of Tracking starting possible area ***It's changed by a program "A1" automatically. Don't change this variable manually. When it's changed, arc information becomes abnormal.**	Read/Write	Long-precision real number
M_EncEnd	Condition Number 1 to 8.	The encoder value in the end position of Tracking starting possible area	Read/Write	Long-precision real number
M_EncStop	Condition Number 1 to 8.	The encoder value in the location where a tracking is ended compulsorily ***It's changed by a program "A1" automatically. Don't change this variable manually. When it's changed, arc information becomes abnormal.**	Read/Write	Long-precision real number

Variable name	Number of arrays	Function	Attribute (*1)	Data type
P_TrkPAcl	Condition Number 1 to 8.	Parameter [TRPACL] value	Read/Write	Position
P_TrkPDcl	Condition Number 1 to 8.	Parameter [TRPDCL] value	Read/Write	Position
M_TrkBuf	Condition Number 1 to 8.	Buffer Number	Read/Write	Integer
M_TrkStart	Condition Number 1 to 8.	Tracking Starting Distance	Read/Write	Single-precision real number
M_TrkEnd	Condition Number 1 to 8.	Tracking Ending Distance	Read/Write	Single-precision real number
M_TrkStop	Condition Number 1 to 8.	Tracking Forced Ending Distance	Read/Write	Single-precision real number
M_TrkTime	Condition Number 1 to 8.	Timeout period of TrkWait command	Read/Write	Single-precision real number
P_TrkBase	Condition Number 1 to 8.	Tracking Base coordinates	Read/Write	Position
M_TrkArcEnc	Condition Number 1 to 8.	The encoder value towards which the workpiece advanced on the arc after a sensor reacted	Read Only	Long-precision real number
M_TrkChk	Condition Number 1 to 8.	TrkChk result	Read Only	Integer
P_TrkWork	Condition Number 1 to 8.	Workpiece position when the sensor taken out from the tracking buffer reacts.	Read Only	Position
M_TrkEnc	Condition Number 1 to 8.	Workpiece Encoder when the sensor taken out from the tracking buffer reacts.	Read Only	Long-precision real number
M_TrkKind	Condition Number 1 to 8.	Model number of the workpiece taken out from the tracking buffer.	Read Only	Integer
M_TrkEncNo	Condition Number 1 to 8.	Encoder number taken out from the tracking buffer.	Read Only	Integer
P_TrkTarget	-	The workpiece coordinate where the robot is following	Read Only	Position
M_Trbfct	buffer No. 1 to The first argument of parameter [TRBUF]	Number of data items stored in the tracking buffer	Read Only	Integer
P_CvSpd	number of encoders 1 to 8	Conveyer speed (mm, rad/sec)	Read Only	Position
M_Hnd	Hand Number 1 to 8	Hand open/close instruction and Hand open/close states. ※Used when you open or close the hand during “WthIf”.	Read/Write	Integer
M_TrkType	Condition Number 1 to 8.	The type of the tracking function 0 – Straight line tracking 1 – Circular arc tracking	Read Only	Integer

10.1.3. 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.

TrClr (tracking data clear)

[Function]

Clear the tracking data buffer.

[Format]

TrClr □ [<Buffer number>]

[Terminology]

<Buffer number [integer]**>** (can be omitted):

Specify the number of a general-purpose output to be output.
Setting range:1 to The first argument of parameter "TRBUF"

[Reference program]

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

[Explanation]

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

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>]]]]

[Terminology]

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

<Encoder data [double-precision real number]**>** (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 [integer]**>** (can be omitted):
Specify the model number of workpieces.
Setting range: 1 to 65535

<Buffer number [integer]**>** (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 [integer]**>** (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

<Pixel data [position]**>** (cannot be omitted):
Specify the workpiece pixel position measured by a sensor.

[Reference program]

(1) Tracking operation program

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

(2) Sensor data reception program

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

[Explanation]

(1) 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.

(2) 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.

(3) 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.

TrkArc (Setting of arc information)

[Function]

Conveyer information for a circular arc tracking is set.

[Format]

TrkArc□<Condition number>, <Encoder number>, <Circular arc position 1>, <Circular arc position 2>, <Circular arc position 3>

[Terminology]

<Condition number [integer]**>**

Specify the tracking condition number.
Setting range: 1 to 8

<Encoder number [integer]**>**

Specify a logic number indicating the external encoderthat performs tracking operation.
Setting range: 1 to 8

<Circular arc position 1 [position]**>**

Specify tracking area starting position.

<Circular arc position 2 [position]**>**

Specify tracking area ending position.

<Circular arc position 3 [position]**>**

Specify tracking cancellation position.

[Reference Program]

```
1 TrkArc 1, 1, P_107(1), P_108(1), P_109(1)  'Circular arc tracking conveyer information is set
```

[Explanation]

(1) Conveyer information for a circular arc pursuit is calculated from "position data which were specified with an argument"(<Circular arc position 1>, <Circular arc position 2>, <Circular arc position 3>) and "encoder data which were set in robot status variable"(M_EncSensor, M_EncStart, M_EncEnd, M_EncStop).
(2) < Circular arc position 1>< Circular arc position 2>< Circular arc position 3> means < starting position >< ending position >< cancellation position > of an arc.
(3) Execute TrkArc before beginning a circular arc tracking.
(4) When this command is executed, the amount of robot movement per encoder pulse is set to robot status variable P_EncDlt.
(5) When this command is executed, the position in which the sensor reacts to workpiece is set to robot status variable P_TrkSensor.
(6) When this command is executed, the value of “Circular arc tracking(1)”is set to robot status variable M_TrkType.
(7) Error L.3110 (value of the argument outside of the range) occurs when <Condition number> is outside a set range.
(8) Error L.3110 (value of the argument outside of the range) occurs when <Encoder number> is outside a set range.
(9) Error occurs when there is the same position in three specified points or when three points are being on the straight line.

TrkChk (Tracking check function)

[Function]

Execute the processing depending on the state of workpiece corresponding to <Condition number> specified.

[Format]

TrkChk □ <Condition number> , <Starting position> , [<Waiting position>] , <Branch destination>

[Terminology]

<Condition number [Integer]**>**

Specify the condition number correspond to tracking.
Setting range: 1 to 8

<Starting position [Position]**>**

When there is no workpiece in tracking buffer(no workpiece on the conveyor), specify the starting position to which robot moves at the beginning of the system. Mainly, specify the starting position as the system to which robot moves at the beginning of the system.

<Waiting position [Position]**> :** (can be omitted.)

Specify the waiting position until workpiece enters a tracking possible area.
In case of the high speed tracking, X or Y coordinates of <Waiting position> are changed to coordinates of workpiece.
In case of the circular arc tracking, move to the designated location without changing it.

<Branch destination [label]**>**

Specify the label name that jumps when specified workpiece can be followed.

[Reference program]

```
*LBFCHK
.....
 TrkChk 1, P1, PWAIT, *LTRST              'No workpiece->P1/ Wait for the workpiece->PWAIT/
                                                          Tracking possible->Jump to "LTRST".
 If M_TrkChk(1) <= 1 Then GoTo *LBFCHK    '0:No workpiece   /   1: Workpiece passed over ->"LBFCHK".
 TrkWait *LBFCHK                          'Wait for the workpiece / Jump to "LBFCHK" at the timeout.
```

[Explanation]

(1) Workpiece information is taken out of the tracking buffer of state variable "M_TrkBuf" corresponding to <condition number >.The position of the workpiece is checked by using the range specified for robot state variable "M_EncSensor","M_EncStart","M_EncEnd","M_EncStop","M_TrkStart","M_TrkEnd","M_TrkStop" The checked result is stored in robot state variable "M_TrkChk".

(2) Workpiece information which is taken out of the specified tracking buffer is in state variable "P_TrkWork", "M_TrkEnc", "M_TrkKind" and "M_TrkEncNo" when "TrkChk" is executed.

(3) If state variable "M_TrkBuf" is not specified when "TrkChk" is executed, buffer number is assumed to be "1".

(4) Execute the following processings according to the execution result of this command.

M_TrkChk value	Execution result	Processing	Robot operation
0	No workpiece in the tracking buffer.	Execute the process that move to specified <Starting position>.	Robot move from current position to <Starting position>.
1	There is workpiece information in the tracking buffer. And the workpiece has passed the tracking starting possible area.	No processing.	Robot does not move.
2	There is workpiece information in the tracking buffer. And the workpiece exists in front of the tracking starting possible area.	Confirm the workpiece position. Change the position data of specified <Waiting position>. Move to the position.	Robot moves from the current position to the position to which the workpiece flows.

M_TrkChk value	Execution result	Processing	Robot operation
3	There is workpiece information in the tracking buffer. And the workpiece exists in the tracking starting possible area.	Jump to the specified <Branch destination>.	Robot does not move.

(5) Number which you can enter to specify <Condition number> is an integer in the range of "1" to "8." Entering anything else causes L3110_99000 (Argument value range over) error to occur.

(6) If you appoint the label which does not exist as “Branch destination”, error L3600_00000 (Jump destination does not exist) occurs.

TrkWait (Tracking wait function)

[Function]

Wait until workpiece correspond to appointed <Condition number> enters to the tracking area.

[Format]

TrkWait □ < Branch destination >

[Terminology]

<Branch destination [label]**>** :(can be omitted.)

Even if the time specified as the state variable "M_TrkTime" passes, when the specified work piece does not go into tracking area, specify the label name to jump.

If < Branch destination > is omitted, the timeout does not occur, and workpiece information is written into the tracking buffer by “TrWrt”, waits until the workpiece enters to the tracking possible area.

[Reference program]

```
M_TrkTime(1) = 60                    ' The timeout period is 60 seconds.
.....
'/// Tracking buffer check ///
*LBFCHK
TrkChk 1, PSave, PWait, *LTRST       'No workpiece->PSave/ Wait for the workpiece->PWait/
                                                Tracking possible->Jump to “LTRST”.

If M_TrkChk(1) <= 1 Then GoTo *LBFCHK   ' 0:No workpiece  /  1: Workpiece passed over->“LBFCHK”.
TrkWait *LBFCHK                      ' Wait for the workpiece / Jump to “LBFCHK” at the timeout.
```

[Explanation]

(1) Take workpiece information out of “TrkChk”, wait until the workpiece enters to the range specified for state variable “M_TrkStart” and “M_TrkEnd”.

(2) When work piece passes away by discontinuation etc., the following work piece information is taken out from a tracking buffer, and it waits until the work piece goes into the range specified as a state variable "M_TrkStart" and "M_TrkEnd."

(3) If specified workpiece does not enter to the tracking area when the time specified for state variable "M_TrkTime" is exceeded at waiting time, jump to <Branch destination>.

(4) When robot state variable "M_TrkBuf" is not executed, the buffer number is assumed to be “1".

(5) If <Branch destination> is omitted or state “M_TrkTime” is “0.00”, the timeout does not occur, and workpiece information is written in into the tracking buffer by “TrWrt”, waits until the workpiece enters to the tracking possible area.

(6) If you appoint the label which does not exist as <Branch destination>, error 3600_00000 (Jump destination does not exist) occurs.

TrkMv (Tracking movement function)

[Function]

Execute the next processing. Validate specified interruption, Start tracking, Move to the tracking upper position by Joint interpolation movement.

[Format]

```
TrkMv □ On , <Tracking upper position> [, <Interrupt number> , <Branch destination>]
TrkMv □ Off
```

[Terminology]

<Tracking upper position [position]**>**

Specify the tracking upper position to follow. (Example : PGT * PGUP1)

<Interrupt number [Integer]**>** : (can be omitted.)

Specify the interrupt number checks the following.

・When tracking, does the workpiece reach <Forced Ending Distance > specified for robot state variable "M_TrkStop()"?

Setting range: 1 to 8

<Branch Destination [Label]**>** :(can be omitted.)

Specify the jumping label name when specified workpiece reach <Forced Ending Distance >.

[Reference program]

```
M_TrkBuf(1) = 1                          ' <Buffer number> is "1".
P_TrkBase(1) = PTBASE                    ' P_TrkBase(1) variable is PTBASE variable.
・・・・・
'/// Tracking buffer check ///
*LBFCHK
TrkChk 1, PSave, PWait, *LTRST           'No workpiece->PSave/ Wait for the workpiece->PWait/
                                         Tracking possible->Jump to "LTRST".
If M_TrkChk(1) <= 1 Then GoTo *LBFCHK    ' 0:No workpiece  /  1: Workpiece passed over->"LBFCHK".
TrkWait *LBFCHK                          ' Wait for the workpiece  / Jump to "LBFCHK" at the timeout.
・・・・・
'/// Start tracking operation ///
*LTRST
TrkMv On, PGTUP, 1, *S91STOP             'Start the interrupt check->Trk On->Move to the tracking upper
                                         position / In the case of exceeding the distance specified by
                                         "M_TrkStop"-Trk Off→Jump to "S91STOP"
・・・・・ adsorption / Release / assembly etc. ・・・・・
TrkMv Off                                'Stop the interrupt check -> Trk Off
```

[Explanation]

(1) In the case of "TrkMv On", if the workpiece position exceed the distance specified by "M_TrkStop", execute the interrupt processing that jump to label specified for <Branch destination> by using <Interrupt number>.

(2) After the starting of the above interrupt monitoring, start tracking on upper position.

(3) In the case of "TrkMv Off", stop the interrupt monitoring specified in "TrkMv On", stop tracking.

(4) <Position data>, <Encoder data>, <Reference position data>, <Encoder number> which is necessary for conventional "Trk On" uses the data in the tracking buffer correspond to <Condition number> specified by "TrkChk" (Buffer number specified by state variable "M_Trkbuf") and the data specified by state variable "P_TrkBase".

(5) The data in the tracking buffer is confirmed by state variable "P_TrkWork","M_TrkEnc","M_TrkKind" and "M_TrkEncNo".

(6) When there is no work piece in back from the starting position of tracking area and this command is executed, L2580 (Workpiece isn't in tracking area) error occurs.

(7) If you omit <Interrupt number> and <Branch destination>, the interrupt processing does not become effective. But you can specify another interrupt processing by using "Def MoTrg" and "Def Act".

(8) If you appoint the label which does not exist as "Branch destination", error L3600_00000(Jump destination does not exist) occurs.

M_Enc (Encoder value)

[Function]

Read the encoder value of the designated logic encoder number. It can be changed to the optional value.

[Format]

[Write]
M_Enc(<logic encoder number>) ＝ <Fixed value>
[Read]
<Numeric value> ＝ M_Enc(<logic encoder number>)

[Terminology]

< logic encoder number [integer]**>**:(can be omitted.)
Specify the logic encoder number which acquires the encoder value.
Setting range: 1 to 8
If the argument is omitted, 1 is set as the default value.

< Fixed value [double-precision real number]**>**
Specify the numerical value.

< Numeric value [double-precision real number]**>**
Specify the numeric variable in which the value.

[Reference program]

MENC1#=M_Enc(1)	'Stocks the logic encoder number encoder of 1 value in MENC1# variable.
MENC2#=M_Enc(M1%)	'Stocks the encoder value of the logic encoder number designated by M1% variable in MENC2# variable.
TrWrt P1, M_Enc(1), MK	' This variable writes in buffer 1 that the location of the workpiece which was kind number MK is P1 at the present encoder value M_Enc (1).
M_Enc(1)=0	'Changes the encoder value of the logic encoder No.1 to "0".

[Explanation]

(1) Acquire the encoder value of the designated <logic encoder number>.
The acquired encoder value is written in a tracking buffer using a TrWrt command to tracking movement.
(2) The encoder value is the double-precision real number value, so please specify a variable of double-precision real number type as<Numeric value>.
(3) It's possible to change the encoder value of the number specified as<logic encoder number> to the value specified as<Fixed value>.
(4) You can omit the step to specify <logic encoder number>.When it is omitted, logic encoder number will be treated as "1."
(5) Error L.3110 (value of the argument outside of the range) occurs when <Condition number> is outside a set range.
*When inputting the numerical value including the decimal point, its value is rounded up.

M_EncL (Latched Encoder data)

[Function]

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

[Format]

[Write]
M_EncL(<logic encoder number>) = <Fixed value>
[Read]
<Numeric value> = M_EncL(<logic encoder number>)

[Terminology]

<Logic encoder number [Integer]**> :**(can be omitted)
Specify the value of logic encoder number

< Fixed value [double-precision real number]**>**
Specify the stored encoder data to initial value(zero or other).

<Numeric variable [double-precision real number]**>**
Specify the numerical variable to substitute.

[Reference program]

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

[Explanation]

(1) Stored encoder value corresponding to the encoder number specified for <logical encoder number> is acquired.
Encoder value is stored in memory at a low-to-high or high-to-low transition of TREN signal which has been specified with a DIP switch on Q173DPX module.
Encoder value thus acquired is written onto the buffer for tracking by using a TrWrk command so as to perform tracking operations.

(2) As encoder value is in double-precision real number, specify <Numerical variable> with a variable which is of double-precision real-number type.

(3) You can omit the step to specify <Logic encoder number>. When it is omitted, logic encoder number will be treated as "1."

(4) Number which you can enter to specify <Logic encoder number> is an integer in the range of "1" to "8." Entering anything else causes L3110 (Argument value range over) error to occur.
* If a number having a decimal part is entered, the fraction of 0.5 or over will be counted as one and the rest will be cut away.

(5) As latched encoder data represents a value present at a low-to-high or high-to-low transition of TREN signal, you should check input corresponding to input number in 810 to 817 range which has been assigned to TREN signal when reading it out.

(6) You can clear the encoder to zero by typing "0" after having used latched encoder data. This step may be performed as a precaution against using previously latched data.

P_EncDlt(The encoder amount of movement)

[Function]

Set the amount of robot movement per encoder pulse.
Or, the amount of robot movement per encoder pulse is returned.
The amount of robot movement :
Straight line tracking :(X, Y, Z, 0, 0, 0, L1, L2)
Circular arc tracking :(Arc length, 0, 0, 0, 0, 0, 0, 0)

[Format]

[Write]
P_EncDlt(<Encoder number>) = <Position Data>
[Read]
<Position Variables> = P_EncDlt(<Encoder number>)

[Terminology]

<Encoder number [Integer]**>**: (can be omitted.)
Specify a logic number indicating the external encoder that performs tracking operation.
Setting range: 1 to 8
If the argument is omitted, 1 is set as the default value.

<Position Data [Position]**>**
Specify the amount of robot movement per encoder pulse.

<Position Variables [Position]**>**
Specify a position variable that stores amount of robot movement per encoder pulse.

[Reference Program]

```
P_EncDlt(1) = P1        'Amount of robot movement per encoder pulse of encoder number 1 is set.
P2 = P_EncDlt(2)        'Amount of robot movement per encoder pulsee of encoder number 2 is stored in positional variable.
```

[Explanation]

(1) The amount of robot movement per encoder pulse of specified <Encoder number> is set. Or, the amount of robot movement per encoder pulse is returned.
(2) If tracking type is a circular arc tracking, it is set by the TrkArc command, the meaning of each element is as follows.
X : Amount of robot movement on circular arc per encoder pulse (Unit:[mm])
(3) You can omit the step to specify <logic encoder number>.When it is omitted, logic encoder number will be treated as "1."
(4) Error L.3110 (value of the argument outside of the range) occurs when <Encoder number> is outside a set range.

P_TrkSensor

[Function]

The position of workpiece to which the sensor reacted is returned.

[Format]

```
[Read]
    <Position Variables> = P_TrkSensor(<Condition number>)
```

[Terminology]

<Condition number [Integer]**>**: (can be omitted.)

Specify the tracking condition number.
Setting range: 1 to 8
If the argument is omitted, 1 is set as the default value.

<Position Variables [Position]**>**

Specify a position variable that stores the position of workpiece to which the sensor reacts.

[Reference Program]

```
PWrk = P_TrkSensor(1)                          'Workpiece position is stored in positional variable.
TrWrt PWrk, MEncData#, MWrkNo, 1, MEncNo ' Workpiece information is written in a tracking buffer.
```

[Explanation]

(1) The position of workpiece to which the sensor of specified <Condition number> reacted is returned.
(2) When the "TrkArc" and "TrkChk" command isn't executed, the value of all zero returns.
(3) You can omit the step to specify <Condition number>.When it is omitted, condition number will be treated as "1."
(4) Error L.3110 (value of the argument outside of the range) occurs when <Condition number> is outside a set range.
(5) This variable is read only.

M_EncSensor

[Function]

Set the encoder data at the position in which the sensor reacts to workpiece.
Or, the encoder data at the position in which the sensor reacts to workpiece is returned.
The set value is set by the 1st element of a parameter "TRKENC*" (*= condition number 1-8).

[Format]

```
[Write]
    M_EncSensor(<Condition number>) = <Numeric value>
[Read]
    <Numeric Variable> = M_EncSensor(<Condition number>)
```

[Terminology]

<Condition number [Integet]**>**: (can be omitted.)
Specify the tracking condition number.
Setting range: 1 to 8
If the argument is omitted, 1 is set as the default value.

<Numeric value [Long-precision real number]**>**
Specify the encoder data at the position in which the sensor reacts to workpiece.
Setting range: Parameter ”ENCRGMN” to “ENCRGMX”

<Numeric Variable [Long-precision real number]**>**
Specify a numeric variable that current the encoder data being set now.

[Reference Program]

```
M_EncSensor(1) = M_Enc(1)       ' Encoder data at the position in which the sensor reacts to workpiece is
                                  gotten.
...
M_EncStart(1) = M_Enc(1)        ' Encoder data at tracking area starting position is gotten.
P_107(1) = P_Fbc(1)             ' Robot current position is gotten.
...
M_EncEnd(1) = M_Enc(1)          ' Encoder data at tracking area ending position is gotten
P_108(1) = P_Fbc(1)             ' Robot current position is gotten.
...
M_EncStop(1) = M_Enc(1)         ' Encoder data at tracking cancellation position is gotten
P_109(1) = P_Fbc(1)             ' Robot current position is gotten.
```

[Explanation]

(1) Set the Encoder data at the position in which the sensor reacts to workpiece.
(2) The set value is set by the 1st element of a parameter "TRKENC*" (*= condition number 1-8).
(3) Read this value, the Encoder data at the position in which the sensor reacts to workpiece is returned.
(4) You can omit the step to specify <Condition number>.When it is omitted, condition number will be treated as "1."
(5) Error L.3110 (value of the argument outside of the range) occurs when <Condition number> is outside a set range.
(6) When the value designated as <Numeric value> is outside the setting range, Error L2560 (tracking parameter abnormality) occurs at the time of TrkArc command execution.

M_EncStart

[Function]

Set the encoder data at tracking area starting position.
Or, the encoder data at tracking area starting position is returned.
The set value is set by the 2nd element of a parameter "TRKENC*" (*= condition number 1-8).

[Format]

```
[Write]
    M_EncStart(<Condition number>) = <Numeric value>
[Read]
    <Numeric Variable> = M_EncStart(<Condition number>)
```

[Terminology]

<Condition number [Integet]**>:** (can be omitted.)
Specify the tracking condition number.
Setting range: 1 to 8
If the argument is omitted, 1 is set as the default value.

<Numeric value [Long-precision real number]**>**
Specify the encoder data at the position in which the sensor reacts to workpiece.
Setting range: Parameter ”ENCRGMN” to “ENCRGMX”

<Numeric Variable [Long-precision real number]**>**
Specify a numeric variable that current the encoder data being set now.

[Reference Program]

```
M_EncSensor(1) = M_Enc(1)       ' Encoder data at the position in which the sensor reacts to workpiece is gotten.
...
M_EncStart(1) = M_Enc(1)        ' Encoder data at tracking area starting position is gotten.
P_107(1) = P_Fbc(1)             ' Robot current position is gotten.
...
M_EncEnd(1) = M_Enc(1)          ' Encoder data at tracking area ending position is gotten
P_108(1) = P_Fbc(1)             ' Robot current position is gotten.
...
M_EncStop(1) = M_Enc(1)         ' Encoder data at tracking cancellation position is gotten
P_109(1) = P_Fbc(1)             ' Robot current position is gotten.
```

[Explanation]

(1) Set the encoder data at tracking area starting position.
(2) The set value is set by the 2nd element of a parameter "TRKENC*" (*= condition number 1-8).
(3) Read this value, the encoder data at tracking area starting position is returned.
(4) You can omit the step to specify <Condition number>.When it is omitted, condition number will be treated as "1."
(5) Error L.3110 (value of the argument outside of the range) occurs when <Condition number> is outside a set range.
(6) When the value designated as <Numeric value> is outside the setting range, Error L2560 (tracking parameter abnormality) occurs at the time of TrkArc command execution.

M_EncEnd

[Function]

Set the encoder data at tracking area ending position.
Or, the encoder data at tracking area ending position is returned.
The set value is set by the 3rd element of a parameter "TRKENC*" (*= condition number 1-8).

[Format]

```
[Write]
    M_EncEnd(<Condition number>) = <Numeric value>
[Read]
    <Numeric Variable> = M_EncEnd(<Condition number>)
```

[Terminology]

<Condition number [Integet]**>:** (can be omitted.)
Specify the tracking condition number.
Setting range: 1 to 8
If the argument is omitted, 1 is set as the default value.

<Numeric value [Long-precision real number]**>**
Specify the encoder data at tracking area ending position.
Setting range: Parameter ”ENCRGMN” to “ENCRGMX”

<Numeric Variable [Long-precision real number]**>**
Specify a numeric variable that current the encoder data being set now.

[Reference Program]

```
M_EncSensor(1) = M_Enc(1)       ' Encoder data at the position in which the sensor reacts to workpiece is gotten.
...
M_EncStart(1) = M_Enc(1)        ' Encoder data at tracking area starting position is gotten.
P_107(1) = P_Fbc(1)             ' Robot current position is gotten.
...
M_EncEnd(1) = M_Enc(1)          ' Encoder data at tracking area ending position is gotten
P_108(1) = P_Fbc(1)             ' Robot current position is gotten.
...
M_EncStop(1) = M_Enc(1)         ' Encoder data at tracking cancellation position is gotten
P_109(1) = P_Fbc(1)             ' Robot current position is gotten.
```

[Explanation]

(1) Set the encoder data at tracking area ending position.
(2) The set value is set by the 3rd element of a parameter "TRKENC*" (*= condition number 1-8).
(3) Read this value, the encoder data at tracking area ending position is returned.
(4) You can omit the step to specify <Condition number>.When it is omitted, condition number will be treated as "1."
(5) Error L.3110 (value of the argument outside of the range) occurs when <Condition number> is outside a set range.
(6) When the value designated as <Numeric value> is outside the setting range, Error L2560 (tracking parameter abnormality) occurs at the time of TrkArc command execution.

M_EncStop

[Function]

Set the encoder data at tracking cancellation position.
Or, the encoder data at tracking cancellation position is returned.
The set value is set by the 4th element of a parameter "TRKENC*" (*= condition number 1-8).

[Format]

```
[Write]
    M_EncStop(<Condition number>) = <Numeric value>
[Read]
    <Numeric Variable> = M_EncStop(<Condition number>)
```

[Terminology]

<Condition number [Integet]**>:** (can be omitted.)
Specify the tracking condition number.
Setting range: 1 to 8
If the argument is omitted, 1 is set as the default value.

<Numeric value [Long-precision real number]**>**
Specify the encoder data at tracking cancellation position.
Setting range: Parameter ”ENCRGMN” to “ENCRGMX”

<Numeric Variable [Long-precision real number]**>**
Specify a numeric variable that current the encoder data being set now.

[Reference Program]

```
M_EncSensor(1) = M_Enc(1)       ' Encoder data at the position in which the sensor reacts to workpiece is
                                  gotten.
...
M_EncStart(1) = M_Enc(1)        ' Encoder data at tracking area starting position is gotten.
P_107(1) = P_Fbc(1)             ' Robot current position is gotten.
...
M_EncEnd(1) = M_Enc(1)          ' Encoder data at tracking area ending position is gotten
P_108(1) = P_Fbc(1)             ' Robot current position is gotten.
...
M_EncStop(1) = M_Enc(1)         ' Encoder data at tracking cancellation position is gotten
P_109(1) = P_Fbc(1)             ' Robot current position is gotten.
```

[Explanation]

(1) Set the encoder data at tracking cancellation position.
(2) The set value is set by the 4th element of a parameter "TRKENC*" (*= condition number 1-8).
(3) Read this value, the encoder data at tracking cancellation position is returned.
(4) You can omit the step to specify <Condition number>.When it is omitted, condition number will be treated as "1."
(5) Error L.3110 (value of the argument outside of the range) occurs when <Condition number> is outside a set range.
(6) When the value designated as <Numeric value> is outside the setting range, Error L2560 (tracking parameter abnormality) occurs at the time of TrkArc command execution.

P_TrkPAcl

[Function]

Change the tracking acceleration coefficient of the parameter “TRPACL” temporarily.

[Format]

[Writing]
P_TrkPAcl(<Condition number>) = <Position data>
[Referencing]
<Position variable> = P_TrkPAcl(<Condition number>)

[Terminology]

< Condition number [Integer]**>**

Specify the condition number corresponding to the tracking.
Setting range: 1 to 8

<Position data [Position]**>**

Specify the tracking acceleration coefficient.
Setting area: For each component, 0.10 to 10.0

<Position variable [Position]**>**

Return the specified tracking acceleration coefficient.

[Reference program]

```
P_TrkPAcl(1) = (0.2, 0.2, 0.2, 1.0, 1.0, 1.0, 1.0, 1.0)  'Specify the tracking acceleration coefficient.
P_TrkPDcl(1) = (0.2, 0.2, 0.2, 1.0, 1.0, 1.0, 1.0, 1.0)  ' Specify the tracking deceleration coefficient.
  .....
*LTRST
TrkMv On, PGTUP, 1, *S91STOP  'Start the interrupt processing->Trk On-> Move to the tracking upper position
```

[Explanation]

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

P_TrkPDcl

[Function]

Change the tracking deceleration coefficient of the parameter “TRPDCL” temporarily.

[Format]

[Writing]
P_TrkPDcl(<Condition number>) = <Position data>
[Referencing]
<Position variable> = P_TrkPDcl(<Condition number>)

[Terminology]

< Condition number [Integer]**>**

Specify the condition number corresponding to the tracking.
Setting area: 1 to 8

<Position data [Position]**>**

Specify the tracking deceleration coefficient.
Setting area: For each component, 0.1 to 10.0

<Position variable [Position]**>**

Return the specified tracking deceleration coefficient

[Reference program]

```
P_TrkPAcl(1) = (0.2, 0.2, 0.2, 1.0, 1.0, 1.0, 1.0, 1.0)  ' Specify the tracking acceleration coefficient.
P_TrkPDcl(1) = (0.2, 0.2, 0.2, 1.0, 1.0, 1.0, 1.0, 1.0)  ' Specify the tracking deceleration coefficient.
  .....
*LTRST
TrkMv On, PGTUP, 1, *S91STOP  'Start the interrupt processing->Trk On-> Move to the tracking upper position
```

[Explanation]

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

M_TrkBuf

[Function]

Specify and refer to the tracking buffer number to use.

[Format]

[Writing]
M_TrkBuf(<Condition number>) = <Value>
[Referencing]
<Numeric variable> = M_TrkBuf(<Condition number>)

[Terminology]

<Condition number [Integer]**>**
Specify the condition number corresponding to the tracking.
Setting range: 1 to 8

<Value [Integer]**>**
Specify the tracking buffer number.
Setting range: 1 to the first argument of parameter “TRBUF”.
The initial value of “TRBUF” is 2, the maximum value of “TRBUF” is 8.

<Numeric variable [Integer]**>**
Return the specified tracking buffer number.

[Reference program]

```
M_TrkBuf(1) = 1              'The tracking buffer corresponding to the condition number 1 uses number 1.
.....
TrkChk 1, P1, PWAIT, *LTRST  'Check the workpiece in the tracking buffer which is specified.
```

[Explanation]

(1) Specify the tracking buffer number used in tracking command “TrkChk”,”TrkWait”,”TrkMv”.
(2) You can confirm the specified tracking buffer number by referencing “M_TrkBuf”.
(3) If the tracking buffer number is not specified by using “M_TrkBuf”before executing “TrkChk” command, tracking number will be treated as “1”.
(4) You can omit the step to specify <Condition number>.When it is omitted, condition number will be treated as "1."
(5) Number which you can enter to specify <Condition number> is an integer in the range of "1" to "8." Entering anything else causes L3110 (Argument value range over) error to occur.
(6) Number which you can enter to specify <Value> is an integer in the range of "1" to " the first argument of parameter “TRBUF” ". Entering anything else causes L3110 (Argument value range over) error to occur.

M_TrkStart

[Function]

Specify and refer to the starting position of range in which it is possible to execute the tracking.
In case of the high speed tracking, designate a coordinate from a reference mark in world coordinate system (the coordinate value "0.00").
In case of the circular arc tracking, designate the correction amount from the tracking starting possible area.

[Format]

[Writing]
M_TrkStart(<Condition number>) = <Value>
[Referencing]
<Numeric variable> = M_TrkStart(<Condition number>)

[Terminology]

< Condition number [Integer] **>**
Specify the condition number corresponding to the tracking.
Setting range ： 1 to 8

< Value [Single-precision real number]**>**
Specify the starting position (mm) of range in which it is possible to execute the tracking.
In case of the high speed tracking, designate a coordinate from a reference mark in world coordinate system (the coordinate value "0.00").
In case of the circular arc tracking, designate the correction amount from the tracking starting possible area.
Setting range: 0.00 to （Robot operation range）
Unit: mm

< Numeric variable [Single-precision real number] **>**
Return the starting position of range in which it is possible to execute the tracking..

[Reference program]

```
M_TrkBuf(1) = 1              ' Tracking buffer corresponding to the condition number 1 uses number 1.
M_TrkStart(1) = 30           ' Starting position of range in which it is possible to execute the tracking
                             corresponding to condition number 1 is 30mm.
.....
TrkChk 1, P1, PWAIT, *LTRST  ' Check the workpiece of the specified tracking buffer.
```

[Explanation]

(1) Specify the starting position of range in which it is possible to execute the tracking used in tracking command “TrkChk””TrkWait”.
(2) You can confirm the specified starting position of range in which it is possible to execute the tracking by referencing “M_TrkStart”.
(3) You can omit the step to specify <Condition number>.When it is omitted, condition number will be treated as "1."
(4) Number which you can enter to specify <Condition number> is an integer in the range of "1" to "8." Entering anything else causes L3110 (Argument value range over) error to occur.

M_TrkEnd

[Function]

Specify and refer to the ending position of range in which it is possible to execute the tracking..
In case of the high speed tracking, designate a coordinate from a reference mark in world coordinate system (the coordinate value "0.00").
In case of the circular arc tracking, designate the correction amount from the tracking starting possible area.

[Format]

[Writing]
M_TrkEnd(<Condition number>) = <Value>
[Referencing]
<Numeric variable> = M_TrkEnd(<Condition number>)

[Terminology]

< Condition Number [Integer]**>**
Specify the condition number corresponding to tracking.
Setting range ： 1 to 8

<Value [Single-precision real number]**>**
Specify the ending position (mm) of range in which it is possible to execute the tracking..
In case of the high speed tracking, designate a coordinate from a reference mark in world coordinate system (the coordinate value "0.00").
In case of the circular arc tracking, designate the correction amount from the tracking starting possible area.
Setting range: 0.00 to （Robot operation range）
Unit: mm

< Numeric Variable [Single-precision real number]**>**
Return end position of range in which it is possible to execute the tracking..

[Reference program]

```
M_TrkBuf(1) = 1              ' Tracking buffer corresponding to the condition number 1 uses number 1.
M_TrkStart(1) = 30           ' Starting position of range in which it is possible to execute the tracking
                             corresponding to the condition number 1 is 30mm.
M_TrkEnd(1) = -10            ' End position of range in which it is possible to execute the tracking
                             corresponding to the condition number 1 is -10mm.
.....
TrkChk 1, P1, PWAIT, *LTRST  ' Check the workpiece of the specified tracking buffer
```

[Explanation]

(1) Specify the ending position of range in which it is possible to execute the tracking used in tracking command “TrkChk””TrkWait”.
(2) You can confirm the specified ending position of range in which it is possible to execute the tracking by referencing “M_TrkEnd”.
(3) You can omit the step to specify <Condition number>.When it is omitted, condition number will be treated as "1."
(4) Number which you can enter to specify <Condition number> is an integer in the range of "1" to "8." Entering anything else causes L3110 (Argument value range over) error to occur.

M_TrkStop

[Function]

Specify and refer to forced ending position of range in which it is possible to execute the tracking..
In case of the high speed tracking, designate a coordinate from a reference mark in world coordinate system (the coordinate value "0.00").
In case of the circular arc tracking, designate the correction amount from the tracking starting possible area.

[Format]

[Writing]
M_TrkStop(<Condition number>) = <Value>
[Referencing]
<Numeric variable> = M_TrkStop(<Condition number>)

[Terminology]

< Condition Number [Integer]**>**
Specify the condition number corresponding to tracking.
Setting range: 1 to 8

<Value [Single-precision real number]**>**
Specify the forced ending position(mm) of range in which it is possible to execute the tracking..
In case of the high speed tracking, designate a coordinate from a reference mark in world coordinate system (the coordinate value "0.00").
In case of the circular arc tracking, designate the correction amount from the tracking starting possible area.
Setting range: 0.00 to （Robot operation range）
Unit: mm

< Numeric Variable [Single-precision real number]**>**
Return forced ending position of range in which it is possible to execute the tracking..

[Reference program]

```
M_TrkBuf(1) = 1          ' Tracking buffer corresponding to the condition number 1 uses number 1.
M_TrkStart(1) = 30       ' Starting position of range in which it is possible to execute the tracking
                           corresponding to condition number 1 is 30mm.
M_TrkEnd(1) = -10        ' End position of range in which it is possible to execute the tracking
                           corresponding to condition number 1 is -10mm.
M_TrkStop(1) = -20       ' Forced ending position of range in which it is possible to execute the tracking
                           corresponding to condition number 1 is -20mm.
.....
TrkChk 1, P1, PWAIT, *LTRST  ' Check the work of the specified tracking buffer
```

[Explanation]

(1) Specify the forced ending position of range in which it is possible to execute the tracking used in tracking command “TrkChk””TrkWait”.
(2) You can confirm the specified forced ending position of range in which it is possible to execute the tracking by referencing “M_TrkStop”.
(3) You can omit the step to specify <Condition number>.When it is omitted, condition number will be treated as "1."
(4) Number which you can enter to specify <Condition number> is an integer in the range of "1" to "8." Entering anything else causes L3110 (Argument value range over) error to occur.

M_TrkTime

[Function]

Specify and refer to the timeout value for “TrkWait” command.

[Format]

[Writing]
M_TrkTime(<Condition number>) = <Value>
[Referencing]
<Numeric variable> = M_TrkTime(<Condition number>)

[Terminology]

< Condition number [Integer]**>**

Specify the condition number corresponding to the tracking.
Setting range: 1 to 8

<Value [Single-precision real number]**>**

Specify the timeout time waits until the workpiece enters to range in which it is possible to execute the tracking..
Setting range: 0.00 to
Unit: second

< Numeric Variable [Single-precision real number]**>**

Return specified tracking buffer number.

[Reference program]

```
M_TrkTime(1) = 60                    'Set the timeout time to 60 second.
.....
TrkChk 1, PSave, PWait, *LTRST   ' No workpiece->PSave/ Waits for the workpiece->PWait/Workpiece can
                                     be followed by tracking->*LTRST
If M_TrkChk(1) <= 1 Then GoTo *LBFCHK   '0:No workpiece/1:Workpiece passed over->Jump to *LBFCHK.
TrkWait *LBFCHK                      'Waits until workpiece enters to the tracking area
```

[Explanation]

(1) Specify the timeout time used in tracking command “TrkWait”.
(2) You can confirm the specified timeout time by referencing “M_TrkStop”.
(3) You can omit the step to specify <Condition number>.When it is omitted, condition number will be treated as "1."
(4) Number which you can enter to specify <Condition number> is an integer in the range of "1" to "8." Entering anything else causes L3110 (Argument value range over) error to occur.

P_TrkBase

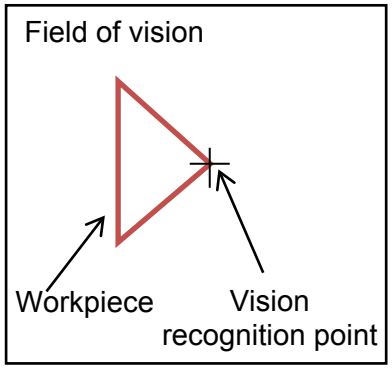
[Function]

Specify and refer to the origin (For example, the position which a vision sensor outputs) of the workpiece to be followed.

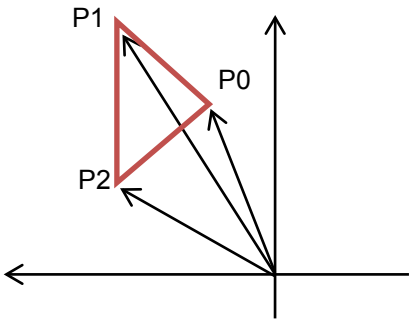
Specify the position data (For example, the position which a vision sensor outputs) used as the reference point when you teach the movement path on the workpiece, as described below

The robot moves to the relative position correspond to this reference point by the movement instruction during the tracking.

[Vision recognition position]



[Teaching position]



Teach three position P0,P1,P2

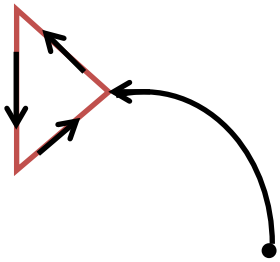
[Robot program]

```
P_TrkBase(1)=P0
  ......
    Mvs    P1
    Mvs    P2
    Mvs    P0

    TrkMv  Off
```

[Robot operation]

While following…

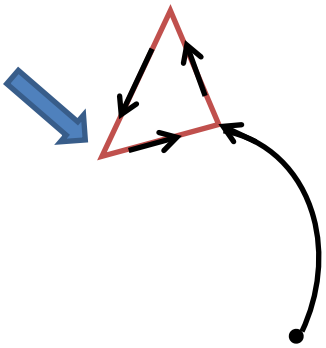


[Structure]

Regard the position outputted by vision as P0 by “P_TrkBase(1)=P”, “TrkBase P0”

Follow “Mvs P1”, “Mvs P2” as the reference position from P0.

If the workpiece Declines, P0 Inclines too, and P1,P2 declines correspond to P0.



For example, when you only absorb the workpiece (do not operate along the external form of the workpiece), you may appoint the position specified when you teach the position which absorb the workpiece (for example, “PTeach”)as “P_TrkBase”,and appoint the above “PTeach” as movement instruction that moves during “TrkOn ～TrkMv Off”(Mov PTeach).

[Format]

[Writing]
P_TrkBase(<Condition number>) = <Position data>
[Referencing]
<Position variable> = P_TrkBase(<Condition number>)

[Terminology]

< Condition number [Integer]**>**
Specify the condition number corresponding to the tracking.
Setting range: 1 to 8

<Position data [Position]**>**
Specify the base position of the tracking.

<Position variable [Position]**>**
Return the base coordinates of the specified tracking.

[Reference program]

```
P_TrkBase(1) = PTBASE     'Specify the tracking base.
  .....
*LTRST
TrkMv On, PGTUP, 1, *S91STOP  'Start the interrupt processing->Trk On->Move to the tracking upper position
```

[Explanation]

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

M_TrkArcEnc

[Function]

Refer to the encoder value which accumulated after a sensor reacts to a workpiece.

[Format]

```
[Referencing]
    <Numeric value> = M_TrkArcEnc(<Condition number>)
```

[Terminology]

<Condition number [Integer]**>:** (can be omitted.)

Specify the tracking condition number.
Setting range: 1 to 8
If the argument is omitted, 1 is set as the default value.

< Numeric value [Long-precision real number]**>**

Return the encoder value which accumulated after a sensor reacts to a workpiece.

[Reference Program]

```
MLimit = M_EncStop(1) – M_EncSensor(1)          ' the encoder value which accumulated after a sensor reacts to a workpiece is calculated
Def Act 1, M_TrkArcEnc(1) > MLimit Goto *S91STOP  ' The definition which interrupts if the termination location is exceeded
.....
Act 1 = 1                                       'Interrupt enable
```

[Explanation]

(1) You can check the encoder value which accumulated after a sensor reacts to a workpiece.
(2) You can omit the step to specify <Condition number>.When it is omitted, condition number will be treated as "1."
(3) Error L.3110 (value of the argument outside of the range) occurs when <Condition number> is outside a set range.

M_TrkChk

[Function]

Refer to the workpiece state read from the tracking buffer when “TrkChk”, “TrkWait” command is executed.

[Format]

[Referencing]
<Numeric variable> = M_TrkChk(<Condition number>)

[Terminology]

< Condition number [Integer]**>**

Specify the condition number corresponding to the tracking.
Setting range: 1 to 8

< Numeric variable [Integer]**>**

Return the workpiece state read from the tracking buffer when “TrkChk”, “TrkWait” command is executed.

0 : No workpiece in the buffer.
1 : The specified workpiece passed over.
2 : Wait for the specified workpiece.
3 : The specified workpiece can be followed by tracking.

[Reference program]

```
M_TrkBuf(1) = 1                ' Tracking buffer corresponding to the condition number 1 uses number 1.
.....
*LBFCHK
TrkChk 1, PSave, PWait, *LTRST ' Check the workpiece of the specified tracking buffer.
If M_TrkChk(1) <= 1 Then GoTo *LBFCHK   '0:No Workpiece/ 1: Workpiece passed over->Jump to
                                        “LBFCHK”.
```

[Explanation]

(1) You can confirm the workpiece state read from the tracking buffer when “TrkChk”, “TrkWait” command is executed..
(2) You can omit the step to specify <Condition number>.When it is omitted, condition number will be treated as "1."
(3) Number which you can enter to specify <Condition number> is an integer in the range of "1" to "8." Entering anything else causes L3110 (Argument value range over) error to occur.
(4) When you execute the writing to “M_TrkChk”, L3210 (This variable is write protected) error occurs.

P_TrkWork

[Function]

Refer to the workpiece position read from the tracking buffer when “TrkChk”, “TrkWait” command is executed.

[Format]

```
[Referencing]
    <Position type variable> = P_TrkWork(<Condition number>)
```

[Terminology]

< Condition Number [Integer]**>**

Specify the condition number corresponding to the tracking.
Setting range: 1 to 8

<Position variable [Position]**>**

Return the workpiece position read from the tracking buffer corresponding to the specified condition number.

[Reference program]

```
M_TrkBuf(1) = 1                 ' Tracking buffer corresponding to the condition number 1 uses number 1.
.....
TrkChk 1, PSave, PWait, *LTRST 'Check the workpiece of the specified tracking buffer.
.....
PWrk = P_TrkWork(1)             'Substitute the workpiece position read from the tracking buffer 1.
```

[Explanation]

(1) You can confirm the workpiece position read from the tracking buffer when “TrkChk”, “TrkWait” command is executed.
(2) If there is no data in the tracking buffer, the data will be cleared.
(3) You can omit the step to specify <Condition number>.When it is omitted, condition number will be treated as "1."
(4) Number which you can enter to specify <Condition number> is an integer in the range of "1" to "8." Entering anything else causes L3110 (Argument value range over) error to occur.
(5) If you execute the writing to “P_TrkWork”, L3210 (This variable is write protected) error occurs.

M_TrkEnc

[Function]

Refer to the encoder value read from the tracking buffer when the “TrkChk”, “TrkWait” command is executed.

[Format]

[Referencing]
<Numeric variable> = M_TrkEnc(<Condition number>)

[Terminology]

< Condition number [Integer]**>**
Specify the condition number corresponding to the tracking.
Setting range: 1 to 8

< Numeric variable [Long-precision real number]**>**
Return the encoder value (pulse) read from the tracking buffer correspond to the specified condition number.

[Reference program]

```
M_TrkBuf(1) = 1                 ' Tracking buffer corresponding to the condition number 1 uses number 1.
.....
TrkChk 1, P1, PWAIT, *LTRST  'Check the workpiece of the specified tracking buffer.
.....
MEnc& = M_TrkEnc(1)             ' Substitute the workpiece position read from the tracking buffer 1.
```

[Explanation]

(1) You can confirm the encoder value read from the tracking buffer when the “TrkChk”, “TrkWait” command is executed.
(2) If there is no data in the tracking buffer, the data will be cleared.
(3) You can omit the step to specify <Condition number>.When it is omitted, condition number will be treated as "1."
(4) Number which you can enter to specify <Condition number> is an integer in the range of "1" to "8." Entering anything else causes L3110 (Argument value range over) error to occur.
(5) If you execute the writing to “M_TrkEnc”, L3210 (This variable is write protected) error occurs.

M_TrkKind

[Function]

Refer to the model number read from the tracking buffer when “TrkChk”, “TrkWait” command is executed.

[Format]

[Referencing]
<Numeric variable> = M_TrkKind(<Condition number>)

[Terminology]

< Condition number [Integer]**>**
Specify the condition number corresponding to the tracking.
Setting range: 1 to 8

< Numeric variable [Long-precision real number]**>**
Return the model number read from the tracking buffer correspond to the specified condition number.

[Reference program]

```
M_TrkBuf(1) = 1                 ' Tracking buffer corresponding to the condition number 1 uses number 1.
.....
TrkChk 1, P1, PWAIT, *LTRST  ' Check the workpiece of the specified tracking buffer.
.....
MKind = M_TrkKind(1)            ' Substitute the model number read from the tracking buffer 1.
```

[Explanation]

(1) You can confirm the model number read from the tracking buffer when “TrkChk”, “TrkWait” command is executed.
(2) If there is no data in the tracking buffer, the data will be cleared.
(3) You can omit the step to specify <Condition number>.When it is omitted, condition number will be treated as "1."
(4) Number which you can enter to specify <Condition number> is an integer in the range of "1" to "8." Entering anything else causes L3110 (Argument value range over) error to occur.
(5) If you execute the writing to “M_TrkKind”, L3210 (This variable is write protected) error occurs.

M_TrkEncNo

[Function]

Refer to the encoder number read from the tracking buffer when “TrkChk”, “TrkWait” command is executed.

[Format]

[Referencing]
<Numeric variable> = M_TrkEncNo(<Condition number>)

[Terminology]

< Condition number [Integer]**>**
Specify the condition number corresponding to the tracking.
Setting range: 1 to 8

< Numeric variable [Long-precision real number]**>**
Return the encoder number read from the tracking buffer correspond to the specified condition number.

[Reference program]

```
M_TrkBuf(1) = 1                 ' Tracking buffer corresponding to the condition number 1 uses number 1.
.....
TrkChk 1, P1, PWAIT, *LTRST  ' Check the workpiece of the specified tracking buffer.
.....
MEncNo = M_TrkEncNo(1)       ' Substitute the encoder number read from the tracking buffer 1.
```

[Explanation]

(1) You can confirm the encoder number read from the tracking buffer when “TrkChk”, “TrkWait” command is executed.
(2) If there is no data in the tracking buffer, the data will be cleared.
(3) You can omit the step to specify <Condition number>.When it is omitted, condition number will be treated as "1."
(4) Number which you can enter to specify <Condition number> is an integer in the range of "1" to "8." Entering anything else causes L3110 (Argument value range over) error to occur.
(5) If you execute the writing to “M_TrkEncNo”, L3210 (This variable is write protected) error occurs.

P_TrkTarget

[Function]

Refer to the information (“P_TrkWork”, “M_TrkEnc”) read from the tracking buffer when “TrkChk”, “TrkWait” command is executed, and the current workpiece position calculated by the state variable “P_EncDlt”.

[Format]

[Referencing]
<Position variable> = P_TrkTarget

[Terminology]

<Position variable>

Return the information (P_TrkWork, M_TrkEnc) read from the tracking buffer when “TrkChk”, “TrkWait” command is executed, and the current workpiece position calculated from the state variable “P_EncDlt”.

[Reference program]

```
M_TrkBuf(1) = 1                 ' Tracking buffer corresponding to the condition number 1 uses number 1.
.....
TrkChk 1, P1, PWAIT, *LTRST  ' Check the workpiece of the specified tracking buffer.
.....
PWrkNow = P_TrkTarget        ' Substitute the current workpiece position.
```

[Explanation]

(1) You can confirm the current workpiece position by referencing the information read from the tracking buffer when “TrkChk”, “TrkWait” command is executed.
(2) If you execute the writing to “M_TrkTarget”, L3210 (This variable is write protected) error occurs.

M_Trbfct

[Function]

Refer to the number of workpieces which exists in a designated buffer.

[Format]

[Referencing]
< Numeric value > = M_Trbfct(<Buffer number>)

[Terminology]

<Buffer number [integer]**>** : (can be omitted.)

Specify the tracking buffer number.
Setting range : 1 to the 1st argument of a parameter "TRBUF"
If the argument is omitted, 1 is set as the default value

< Numeric value [integer]**>**

The number of workpieces in the designated buffer is returned to< Buffer number>.

[Reference program]

```
MWrk = M_Trbfct(1)          'The number of works in number 1 of tracking buffer is stocked in variable MWrk.
```

[Explanation]

(1) You can confirm the number of works in the designated buffer.
(2) You can omit the step to specify <Buffer number>.When it is omitted, buffer number will be treated as "1."
(3) Error L.3110 (value of the argument outside of the range) occurs when <Buffer number> is outside a set range.

P_CvSpd

[Function]

Return the conveyer speed.

[Format]

[Referencing]
< Position variable > = P_CvSpd(<Logic encoder number >)

[Terminology]

<Logic encoder number [integer]**>** : (can be omitted.)

Specify the number of logic encoders which do a chase movement.
Setting range: 1 to 8
If the argument is omitted, 1 is set as the default value

<Position variable [position]**>**

Return the conveyer speed.
In case of the high-speed tracking function, returns the rate in each coordinate of (X, Y, Z, 0, 0, 0, L1, L2).
（When a conveyor is arranged slantingly, the value enters X,Y,Z.）
In case of the circular arc tracking function, returns tool-up speed on the arc to an X element.

[Reference program]

```
PCvSpd = P_CvSpd(1)                ' Stocks the speed of logic encoder No 1 in a PCvSpd variable
```

[Explanation]

(1) Refer to speed of the conveyer and the turntable.
(2) In case of the circular arc tracking, when do not execute the command "TrkArc", returns the value of all zero.
(3) You can omit the step to specify <Logic encoder number>.When it is omitted, logic encoder number will be treated as "1."
(4) Error L.3110 (value of the argument outside of the range) occurs when <Logic encoder number> is outside a set range.
(5) This variable is read only.

M_Hnd

[Function]

Set and refer to the hand open/close states corresponding to the specified <Hand number>.
The contents of processing of this variable are same as HOpen and HClose, but it's used for a<processing> part of Wth / WthIf join mainly.

[Format]

[Writing]
M_Hnd(<Hand number>) = <Value>
[Referencing]
<Numeric variable> = M_Hnd(<Hand number>)

[Terminology]

< Hand number [Integer]**>**
Specify the hand number: (cannot be omitted).
Setting area: 1 to 8

<Value [Integer]**>**
Describe the hand open/close states by numeric variable, constants, or numeric operation expression.
0 : Hand close
1 : Hand open

< Numeric Variable [Integer]**>**
Specify the numeric variable which stores the hand open/close states.
-1 : Undefined hand
0 : Hand close
1 : Hand open

[Reference program]

```
1 Mov P1, 50      ' Move 50mm to Z direction in the tool coordinates system of P1 by Joint interpolation movement.
2 Mvs P1 WthIf M_Ratio > 50, M_Hnd(1) = 0   ' Close the hand of the hand number 1 if it comes to 50% of distance of the purpose position during the movement to P1.
3 *Label : If M_Hnd(1) = 1 Then GoTo *Label  ' Wait until the hand of the hand number 1 closes.
```

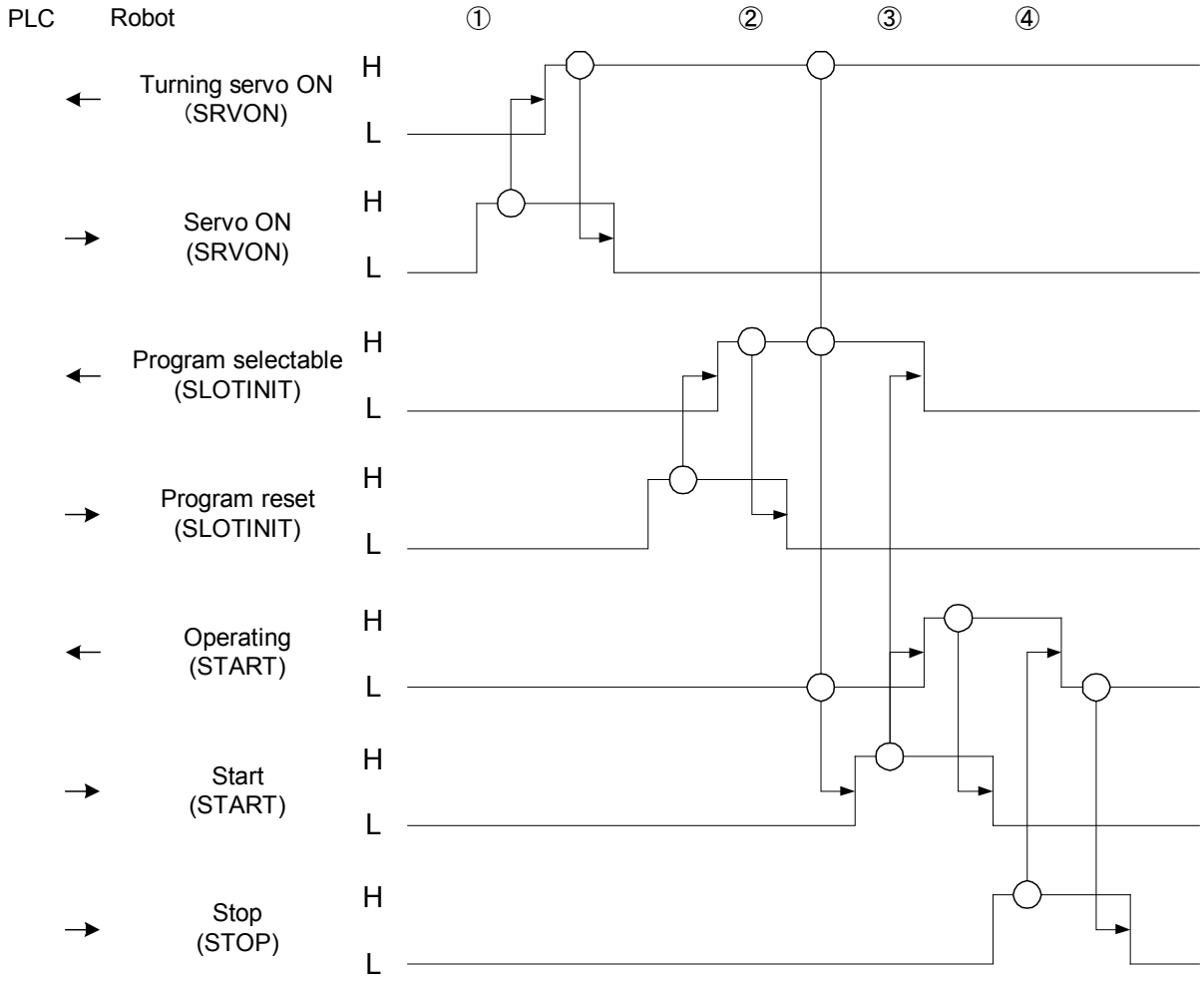
[Explanation]

(1) Change and refer to the hand open/close states.
(2) Writing to “M_Hnd” is treated as the processing equal to the HOpen instruction /HClose instruction.
(3) You can make a statement on <Additional condition>/<Processing> of accompanying instruction to the operation instruction.
(4) Initial value just after the power supply obeys the setting value of the parameter “HANDTYPE” or “HANDINIT” (Output signal number 900 to 907),or “ORS***”(General-purpose output signal).
(5) If you appoint the hand number which is not specified by the parameter “HANDTYPE”, it becomes no processing at the time of writing, and -1 (Undefined hand) returns at the time of reading.
(6) If the hand of specified < hand number> is Double solenoid (D) setting, and output signal state is neither hand open(&B01) nor hand close($B10), return 1(hand open).
(7) You can omit the step to specify <Hand number>.When it is omitted, L3110 (Argument value range over) error occurs.
(8) Number which you can enter to specify <Hand number> is an integer in the range of "1" to "8." Entering anything else causes L3110 (Argument value range over) error to occur.
(9) Number which you can enter to specify <Value> is an integer “0” or “1”. Entering anything else causes L3110 (Argument value range over) error to occur.
(10) If you write “M_Hnd” by using the task slot which does not acquire a machine control rights, error L3280 (Cannot execute without GETM) occurs.
(11) If you read “M_Hnd” by using the task slot which does not acquire a machine control rights, return the robot hand open/close states of machine number 1.
(12) It is impossible to use for the electric hand with many functions made in TAIYO company.
Please refer to the explanation of “Usage of the electric hand with many functions”.
(13) “M_Hnd” does not correspond to the hand macro.

10.2. Timing Diagram of Dedicated Input/Output Signals

10.2.1. 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.”

11. Troubleshooting

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

11.1. Occurrence of Other Errors

Table 11-1 List of Tracking relation Errors

Error number	Error description	Causes and actions
L2500	Tracking encoder data error	[Causes] The data of the tracking encoder is abnormal. (The amount of the change is 1.9 times or more.) [Actions] 1) Check the conveyor rotates at the fixed velocity. 2) Check the connection of the encoder. 3) Check the earth of the earth wire.
L2510	Tracking parameter reverses	[Causes] Tracking parameter[EXCRGMN] and [EXCRGMX] Setting value reverses [Actions] 1) Check the value of [ENCRGMX] and [ENCRGMN] parameters.
L2520	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.
L2530	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.
L2540	There is no read data	[Causes] The TrRd command was executed in state the data is not written in tracking buffer. [Actions] 1) Execute the TrRd command after confirming whether the buffer has the data with the state variable [M_Trbfct]. 2) Confirm whether the buffer number specified by the buffer number specified in TrWrt command and the TrRd command is in agreement.
L2560	Illegal parameter of Tracking	[Causes] The value set as the parameter [EXTENC] is outside the range. The ranges are 1-8. [Actions] Please confirm the value set to Parameter [EXTENC]. Please confirm whether the Q173DPX unit is installed in the slot specified for parameter "ENCUNITn" (n=1-3). Please confirm whether slot 0-2 of a basic base is not specified by setting the parameter. Please confirm whether the setting of "Management CPU" that exists in "I/O unit and intelligent function unit details setting" of the parameter of the sequencer and specification of parameter "ENCUNITn" (n=1-3) are corresponding. There is a possibility Q173DPX is not robot CPU management.

Error number	Error description	Causes and actions
L2570	Installation slot error.	[Causes] Q173DPX is installed in slot 0-2 of a basic base. [Actions] Slot 0-2 of the basic base is basically only for CPU. Please install Q173DPX since slot3.
L2580	No workpiece in the tracking area.	[Causes] There is no workpiece in the tracking buffer or “TrkMv On” command is executed Before the workpiece enters to the tracking area. [Actions] Execute “TrkMv On” command when the workpiece is in the tracking area.
L3982	Cannot be used (singular point)	[Causes] 1) This robot does not correspond to the singular point function 2) Cmp command is executed 3) A synchronous addition axis control is effective 4) Tracking mode is effective 5) Pre-fetch execution is effective 6) This robot is a setting of the multi mechanism 7) ColChk On command is executed [Actions] 1) Check the argument of Type specification 2) Invalidate a compliance mode (execute Cmp Off) 3) Invalidate a synchronous addition axis control 4) Invalidate a tracking mode (execute Trk Off) 5) Invalidate a pre-fetch execution 6) Do not use the function of passage singular point 7) Invalidate a collision detection (execute ColChk Off)
L6632	Input TREN signal cannot be written	[Causes] During the actual signal input mode, external output signal 810 to 817 (TREN signal) cannot be written. [Actions] 1) Use an real input signal (TREN signal)

Please refer to separate manual “Troubleshooting”.

11.2. In such a case (improvement example)

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

11.2.1. The adsorption position shifts.

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

(1) Please confirm turntable and the encoder is not slipping.
(2) Please confirm whether a difference in the adhesion locations is fixed or different depending on the timing from which a workpiece is taken.

It's to (3) in case of a fixed difference. When being different depending on timing, it's to (4).

(3) Please change the on timing of a sensor by making reference to "9.5 Adjustment of Tracking starting possible area".

When it can't be settled, it's to the next.

(4) Please confirm whether the timing a stock sensor turns on is right.

When being not right, please do a sensitivity adjustment of a sensor. When being right, it's to the next.

(5) Please confirm the state variable "P_CvSpd (<Logic encoder number>)" using a variable monitor of RT ToolBox2, and confirm whether conveyor speed isn't changing extremely.

When there is an extreme change, please confirm whether there isn't influence of noise.
When noise doesn't influence, it's to the next.

(6) There is a possibility that the encoder value was abnormal or a possibility that the price of the state variable "M_EncStart" and "M_EncStop" was changed after instruction work in instruction work by a program "A1". Please put "8. Teaching Operation(“A1” Program) " into effect again.

11.2.2. Make adsorption and release of the work speedy.

Adjust the adjustment variable "PDly1", and the value of X coordinates of "PDly2" of the program 1. Refer to "Table 9-1 List of variable for operating conditions" for the adjustment method.

11.2.3. Make movement of the robot speedy.

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

(1) Adjustment of the optimal acceleration-and-deceleration setting
Set mass, size, and center of gravity of the hand installed in the robot as the parameter "HNDDAT1." And, set mass, size, and center of gravity of the work as the parameter "WRKDAT1."
By this setting, the robot can move with the optimal acceleration and deceleration and speed.
Refer to "Table 6-2 List of Operation Parameter" for setting method.

(2) Adjustment of carrying height
By making low distance at adsorption and release of robot, the moving distance decreases and motion time can be shortened as a result. Refer to the adjustment variable of "Pup1"and "Pup2" in the "Table 9-1 List of variable for operating conditions" for change of rise distance.

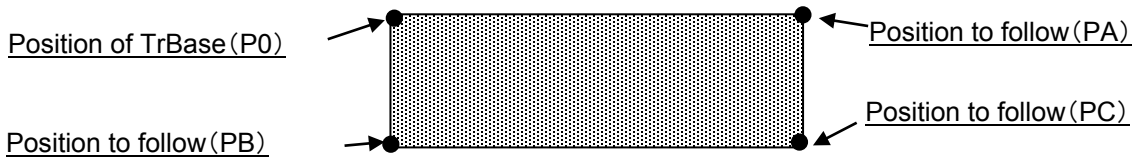
11.2.4. Circle movement in tracking.

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

Before sample program change	After sample program change
39 Mvs PGet 40 Dly PDly1.X 41 Mvs PGetUp ..... 70 P_TrkBase(MWrkNo) = P_107(MWrkNo) 71 PGet = P_TrkBase(MWrkNo)	Mvs PGet Dly PDly1.X **'<Add>->** **Mvc PS1,PS2,PS3** **Mvs PGet** **Dly PDly1.X** **'<-<Add>** Mvs PGetUp ..... P_TrkBase(MWrkNo) = P_107(MWrkNo) PGet = P_TrkBase(MWrkNo) **'<Add>->** **PS1 = PGet * (+5.00,+5.00,+0.00,+0.00,+0.00,+0.00,+0.00)** **PS2 = PGet * (+5.00,-5.00,+0.00,+0.00,+0.00,+0.00,+0.00)** **PS3 = PGet * (-5.00,-5.00,+0.00,+0.00,+0.00,+0.00,+0.00)** **'<-<Add>**

11.2.5. Draw the square while doing the tracking.

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



Before sample program change	After sample program change
39 Mvs PGet 40 Dly PDly1.X 41 Mvs PGetUp ..... 70 P_TrkBase(MWrkNo) = P_107(MWrkNo) 71 PGet = P_TrkBase(MWrkNo)	Mvs PGet Dly PDly1.X **'<Add>->** **Mvs PA** **Mvs PC** **Mvs PB** **Mvs PGet** **Dly PDly1.X 'Adsorption confirmation** **'<-<Add>** Mvs PGetUp ..... P_TrkBase(MWrkNo) = P_107(MWrkNo) PGet = P_TrkBase(MWrkNo) **'<Add>->** **PA = PGet * (+0.00,-50.00,+0.00,+0.00,+0.00,+0.00,+0.00)** **PC = PGet * (-20.00,-50.00,+0.00,+0.00,+0.00,+0.00,+0.00)** **PB = PGet * (-20.00,+0.00,+0.00,+0.00,+0.00,+0.00,+0.00)** **'<-<Add>**

12. 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.

12.1. List of Parameters Related to Tracking

Table 12-1 List of Parameters Related to Tracking

Parameter	Parameter name	Number of elements	Description	Setting value at factory shipment
Encoder information for trackings	TRKENC* (*=1 to 8)	4 Long-precision real numbers	Stocks the encoder value of the state variable "M_EncSensor" "M_EncStart" "M_EncEnd" "M_EncStop". 1st: Encoder data at the position in which the sensor reacts to workpiece (M_EncSensor) 2nd: Encoder data at tracking area starting position (M_EncStart) 3rd: Encoder data at tracking area ending position (M_EncEnd) 4th: Encoder data at tracking cancellation position (M_EncStop)	0, 0, 0, 0
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. Mainly the tracking data for each conveyors is saved at the buffer. Change the set value, when the conveyor for tracking is increased. However, if the value is enlarged, the memory area where the tracking data is saved will be secured. Be careful because the program number which can be saved decreases. Setting range: 1 to 8 <Buffer size> Specify the size in which the tracking data is preserved. Change this element when there is larger tracking data saved by TrWrt command than reading by TrRd command. Be careful because the memory is secured like the above-mentioned [Buffer number]. Setting range: 1 to 200	2 , 64
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)	1000000000, 1000000000, 1000000000, 1000000000, 1000000000, 1000000000, 1000000000, 1000000000

Parameter	Parameter name	Number of elements	Description	Setting value at factory shipment
Tracking adjustment coefficient 1	TRADJ1	8 real numbers (X,Y,Z, A,B,C, L1,L2)	Tracking adjustment coefficient 1 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 acceleration	TRPACL	8 real numbers (X,Y,Z, A,B,C, L1,L2)	Tracking acceleration. Acceleration during execution of tracking movement.	1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0
Tracking deceleration	TRPDCL	8 real numbers (X,Y,Z, A,B,C, L1,L2)	Tracking deceleration. Deceleration during execution of tracking movement.	1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0

12.2. Scene of changing parameter

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

Table 12-2 List of the user scene of changing parameter

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

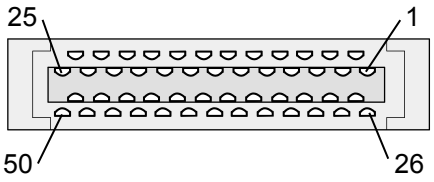
No.	Operation phase	Model		Parameter name	Example	Explanation
		CR750-Q CR751-Q	CR750-D CR751-D			
5	In case of system debag	●	●	TRCWDST	20.0	In case of vision tracking, if there is a workpiece not recognized well by vision sensor, it might reply over one recognition results to one workpiece. In this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

No.	Operation phase	Model		Parameter name	Example	Explanation
		CR750-Q CR751-Q	CR750-D CR751-D			
8	Others	●	●	ENCRGMN	0,0,0,0, 0,0,0,0	This parameter is a parameter that sets the range of the value of state variable M_Enc. M_Enc becomes the range of 0-1000000000, and next to 1000000000, it becomes 0 encoder rotates in case of an initial value. Though this range is changed by this parameter, tracking sample program is made on the assumption that it is used within this range, so do not change this parameter.
9		●	●	ENCRGMX	1000000000, 1000000000, 1000000000, 1000000000, 1000000000, 1000000000, 1000000000, 1000000000	

12.3. Expansion serial interface Connector Pin Assignment (CR750/CR751 series controller)

"Figure 12-1 Connector Arrangement" shows the connector arrangement and "Table 12-3 Connectors: CNENC/CNUSR Pin Assignment" shows pin assignment of each connector.

CNUSR2(CR750-D/CR751-D)
Encoder



Connector: CNUSR2

CNUSR11/12/13(CR750-D)
Encoder

Connector: CNUSR11/12/13

Figure 12-1 Connector Arrangement

Table 12-3 Connectors: CNENC/CNUSR Pin Assignment

Pin NO. Connector name – Pin name CR751-D controller	Pin NO. Connector name – Pin name CR751-D controller	Signal name	Explanation	Input/output	Remark
CNUSR1-28	CNUSR1-28	SG	Control power supply 0 V	GND	CH1
CNUSR1-21	CNUSR1-21	LAH1	+ terminal of differential encoder A-phase signal	Input	
CNUSR1-22	CNUSR1-22	LBH1	+ terminal of differential encoder B-phase signal	Input	
CNUSR1-23	CNUSR1-23	LZH1	+ terminal of differential encoder Z-phase signal	Input	
CNUSR1-33	CNUSR1-33	SG	Control power supply 0 V	GND	CH2
CNUSR2-21	CNUSR2-21	LAH2	+ terminal of differential encoder A-phase signal	Input	
CNUSR2-22	CNUSR2-22	LBH2	+ terminal of differential encoder B-phase signal	Input	
CNUSR2-23	CNUSR2-23	LZH2	+ terminal of differential encoder Z-phase signal	Input	
-	-	-	Empty	-	
-	-	-	Empty	-	
CNUSR2-15	CNUSR2-15	SG	Control power supply 0 V	GND	CH1
CNUSR1-46	CNUSR1-46	LAL1	- terminal of differential encoder A-phase signal	Input	
CNUSR1-47	CNUSR1-47	LBL1	- terminal of differential encoder B-phase signal	Input	
CNUSR1-48	CNUSR1-48	LZL1	- terminal of differential encoder Z-phase signal	Input	
CNUSR2-40	CNUSR2-40	SG	Control power supply 0 V	GND	CH2
CNUSR2-46	CNUSR2-46	LAL2	- terminal of differential encoder A-phase signal	Input	
CNUSR2-47	CNUSR2-47	LBL2	- terminal of differential encoder B-phase signal	Input	
CNUSR2-48	CNUSR2-48	LZL2	- terminal of differential encoder Z-phase signal	Input	
-	-	-	Empty	-	
-	-	-	Empty	-	

MITSUBISHI ELECTRIC CORPORATION

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

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

May., 2017 MEE Printed in Japan on recycled paper.