

DENSO ROBOT

RC7M CONTROLLER

**OPTIONS MANUAL
(SUPPLEMENT)**

CONVEYER TRACKING BOARD

Copyright © DENSO WAVE INCORPORATED, 2005-2011

All rights reserved. No part of this publication may be reproduced in any form or by any means without permission in writing from the publisher.

Specifications are subject to change without prior notice.

All products and company names mentioned are trademarks or registered trademarks of their respective holders.

Preface

Thank you for purchasing optional devices designed for DENSO robots.

This manual covers the specifications, installation, and use of the optional conveyer tracking board to be configured in the **G series robot system together with the RC7M controller.

Before use, read this manual carefully to safely get the maximum benefit from your robot and options in your assembling operations.

Refer to “Chapter 14, Mounting I/O Extension Boards”, in the “RC7M OPTIONS MANUAL” for installation method of extension board.

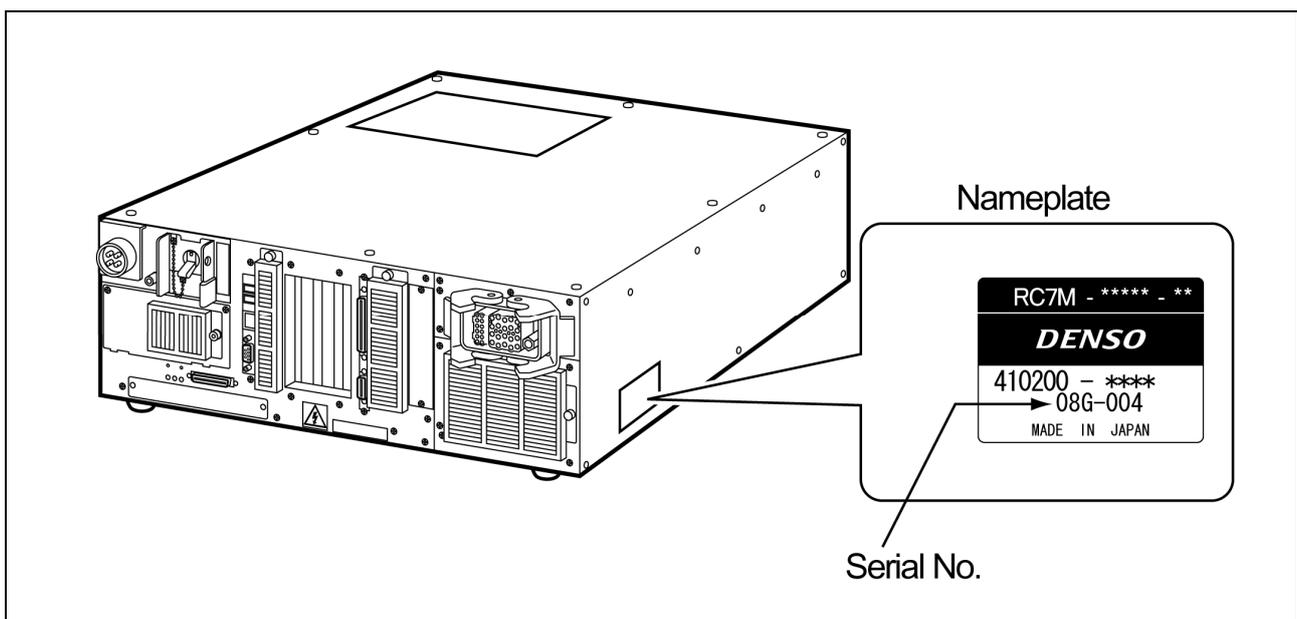
Optional board covered by this manual

Optional extension board “Conveyer Tracking board” for RC7M type controller

To the customer who purchased an extension board as an additional component

- (1) If you order the extension board together with robot, the extension board will be installed to the robot controller, and the controller will be set up with the extension function enabled for shipment.
- (2) If you purchase the extension board separately, inform us of the Serial No. of your robot: Extension board will be delivered with the Customer ID (Password) packed together.
- (3) When this product is first used, be sure to enable the extension function of the extension board using the Customer ID (Password): Refer to the following pages for the procedure to enable function.

NOTE: If your extension board is installed to any robot controller other than the one whose serial number you informed us of at the time of purchase, the extension function cannot be enabled



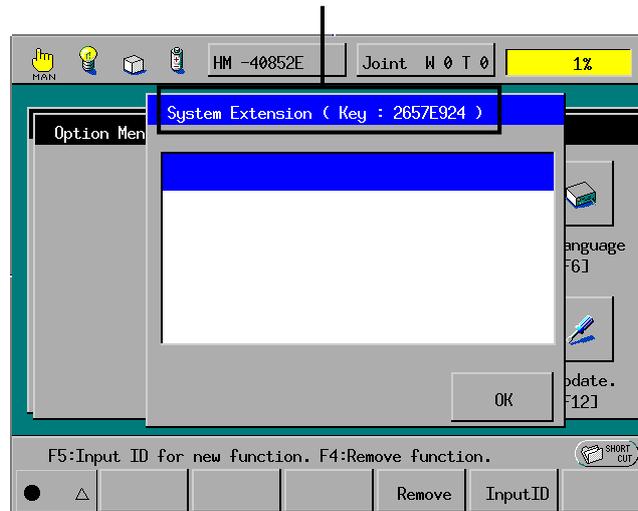
Enabling extension functions by the teach pendant

Access: [F6 Set]—[F7 Options.]—[F8 Extnsion]—
[F5 Input ID]

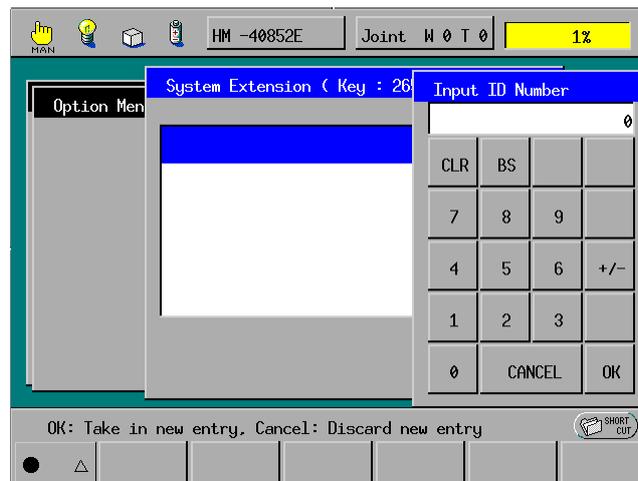
Enables the extension function. Once enabled, the setting will be retained even if the controller power is turned off and on.

- (1) Press [F8 Extnsion] in the Option Menu, and the System Extension window will appear as shown below.

The serial number appears.



- (2) Press [F5 Input ID] on the System Extension window, and the numeric keypad will appear.
- (3) Enter the password and press [OK].
The name of the newly added function will be displayed.

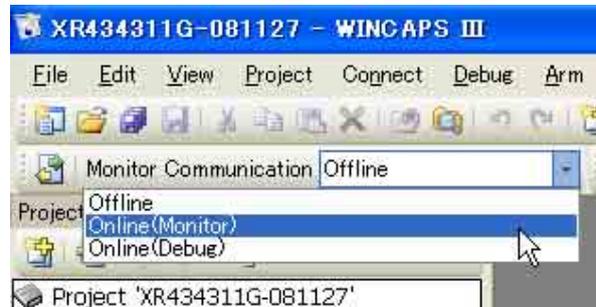


- (4) Restart the controller to make the extension function go into effect.

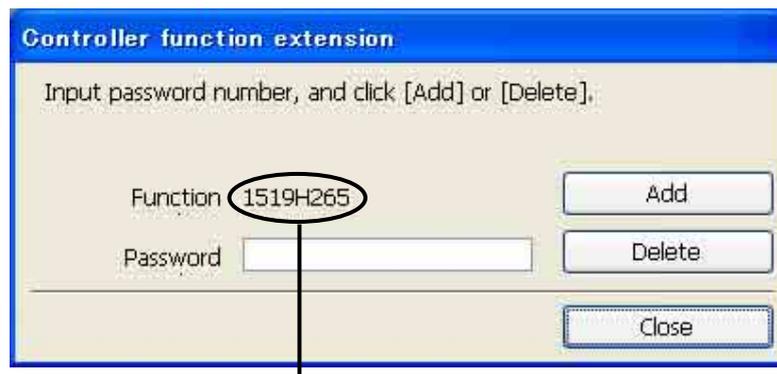
Enabling extension functions in WINCAPS III

Enables the extension function. Once enabled, the setting will be retained even if the controller power is turned off and on.

- (1) Choose Connect | Monitor Communication | Monitoring (online). Then choose Tool | Controller extension.



- (2) The Controller function extension window appears. Enter the password and press Add.



The serial number appears.

- (3) Restart the controller to make the extension function go into effect.

Contents

Preface	
To the customer who purchased an extension board as an additional component.....	i
Enabling extension functions by the teach pendant	ii
Enabling extension functions in WINCAPSIII	iii
Chapter 1 Hardware Setting for Conveyer Tracking.....	1
1.1 Optional Components	1
1.2 Conveyer Tracking Board Specifications	1
1.2.1 Location of Conveyer Tracking Board	1
1.2.2 Specifications	1
1.2.3 Pin Assignment of Conveyer Tracking Board.....	2
1.2.4 Conveyer Tracking Board Settings (Factory-default Settings).....	2
1.3 System Configuration of Conveyer Tracking	3
1.3.1 Encoder Configuration	3
1.3.2 Configuration of Sensors and Cameras	3
1.4 Specifications of Components.....	4
1.4.1 Encoder.....	4
(1) Specifications	4
(2) Outline drawing	4
1.4.2 DC/DC Converter.....	5
(1) Specifications	5
(2) Outer dimensions.....	5
(3) Wiring for DC/DC Converter	6
1.4.3 Sensor Signal Input Pin Layout (NPN Type I/O)	7
1.4.4 Wiring Diagram of Optional Cables and Precautions in Wiring.....	8
(1) Encoder cable	8
(2) Wire connections of encoder (E6B2-CWZ1X) cables	8
(3) Wiring diagram of 5V power supply cable	9
1.5 Hardware Operation Confirming Procedure	10
1.6 Configuring Cameras for the Use of μ Vision Board.....	11
1.6.1 Setting on the Rear Panel.....	11
1.6.2 Setting Inside a Camera	12
[1] Mode change switch (DIP switch)	12
[2] Electronic shutter speed switch (Rotary switch).....	13
[3] Notes.....	13
Chapter 2 Conveyer Tracking Adjustment	14
2.1 Setting Conveyer Tracking Parameters	14
2.1.1 How to Set Conveyer Tracking Parameters	14
2.1.2 Detailed Explanation of Parameters	19
2.2 Conveyer Calibration.....	23
2.2.1 How to Execute Conveyer Calibration.....	23
2.2.2 Conveyer Calibration Execution Procedure	23
2.3 Camera Calibration	28
2.3.1 General	28
2.3.2 Execution Procedure for Camera Calibration Library (CALCCAMCALPOS).....	29
2.4 Teaching of Workpiece Position when Hand I/O Input Interrupt is in Use.....	34
2.4.1 General	34
2.4.2 Execution Procedure for the Workpiece Position Teaching Library (CALCIOTEACHPOS).....	35

Chapter 3 Conveyer Tracking Program	38
3.1 General.....	38
3.2 Program Examples.....	39
3.2.1 Workpiece Detection Program.....	39
(1) When using μ Vision	39
(2) When using a Hand I/O input interrupt.....	39
(3) When using an external visual equipment	40
3.2.2 Tracking Operation Program	41
3.3 Program Confirmation Procedure	42
3.4 Conveyer Tracking Accuracy Checking Procedures.....	44
3.4.1 Encoder Installation Check.....	44
3.4.2 When Using Visual Equipment (External visual equipment such as μ Vision)	44
3.4.3 When Position is Recognized by a Sensor (Hand I/O Interrupt).....	46
3.5 Tact time Decreasing Operation.....	48
3.6 Program Examples.....	49
3.7 List of Commands Related to the Conveyer Tracking Function	52
TRACKDATAINITIALIZE (statement).....	52
TRACKDATASET (statement).....	53
TRACKDATAGET (statement)	54
TRACKDATAINFO (statement).....	55
TRACKDATANUM (statement)	55
CURTRACKPOS (statement)	56
CURTRACKSPD (statement).....	57
WAITTRACKMOVE (statement)	58
CALCWORKPOS (statement).....	59
CURTRACKPOSEX (statement).....	60
WAITTRACKMOVEEX (statement).....	61
CAMIN (Statement).....	62
CAMMODE (Statement).....	63
3.8 List of Libraries Related to the Conveyer Tracking Function.....	64
SETTRACKMOVE (library)	64
RESETTRACKMOVE (library)	65
CONVCAL (library).....	65
CALCCAMCALPOS (library).....	66
CALCIOTEACHPOS (library).....	66
SetTrackStartArea (library).....	67
CalcConvPos (library)	68
SetConvLowVelErr (library).....	68
CalcConvVec (library)	69
SortTrackData (library)	69
SortTrackAllData (library).....	70

Chapter 1 Hardware Setting for Conveyer Tracking

1.1 Optional Components

Table below shows the optional components of the conveyer tracking system.

Optional Components

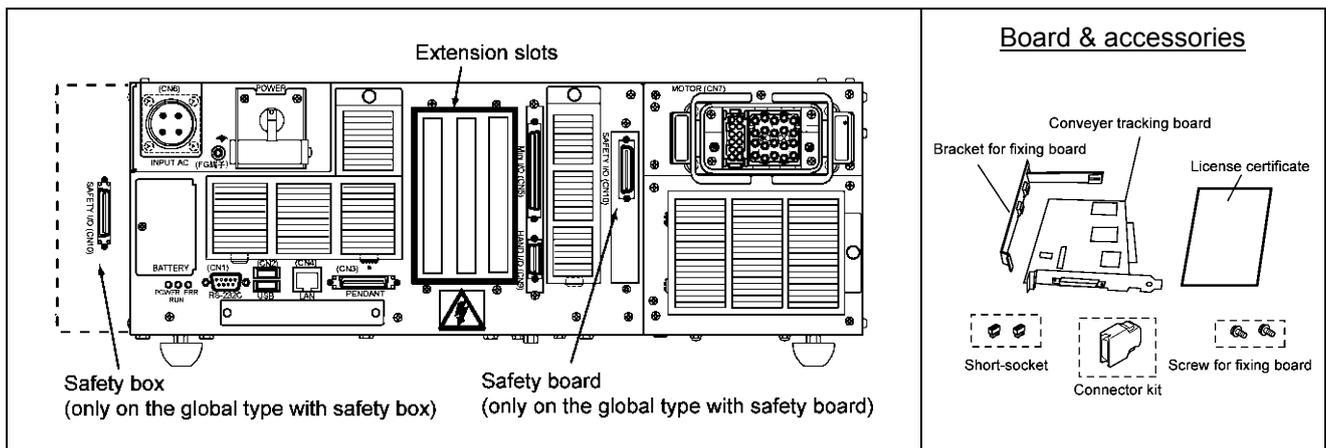
No.	Part name	Remarks	Part No.
1	Conveyer tracking board	Delivered with the board installed on the controller	410010-3460
		Delivered separately	410010-3470
2	Encoder	OMRON: E6B2-CWZ1X1000P/R	410010-1690
3	Encode cable	(3m)	410141-2440
4	Encode cable	(5m)	410141-2430
5	Encode cable	(15m)	410141-2420
6	5V power supply cable	(2m) Encoder to DC/DC converter	410141-2450
7	DC/DC converter	(24V-5V) OMRON: S82S-7305	410010-1680

Note: If delivered with the board installed on the controller, the bracket, screws and short-sockets are also installed on the controller. The JP12 and JP13 of the Mini I/O board are short-circuited with short-sockets. See "1.4.2 (3) Wiring for DC-DC converter".

1.2 Conveyer Tracking Board Specifications

1.2.1 Location of Conveyer Tracking Board

The conveyer tracking board may be inserted into extension slot of the robot controller. For mounting the board to the controller, refer to "Chapter 14, Mounting I/O Extension Boards in the OPTIONS MANUAL".

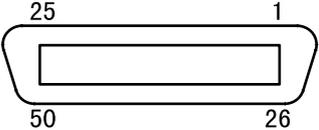


1.2.2 Specifications

Items	Specifications
Input count configuration	Line receiver input, Non isolation (Note)
Input signal	A、 \bar{A} 、B、 \bar{B} 、Z、 \bar{Z}
Maximum response frequency	1.5MHz
No. of channels	2(32Bit)
Supply voltage	5VDC ($\pm 5\%$)

Note: The Input count configuration is non-isolation. Encoder-power-supply-0V must be common to 0V of the controller.

1.2.3 Pin Assignment of Conveyor Tracking Board



View from the cable side

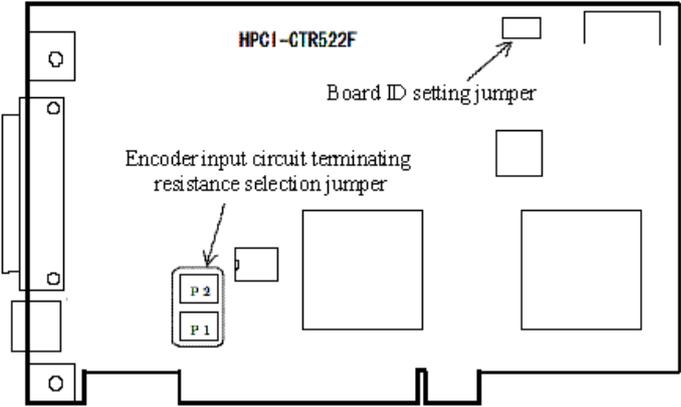
Connector model: 10250-52A2JL (SUMITOMO 3M Limited)

Terminal number	Input signal	Terminal number	Input signal
1	XA	9	YA
2	X\bar{A}	10	Y\bar{A}
3	XB	11	YB
4	X\bar{B}	12	Y\bar{B}
5	XZ	13	ZB
6	X\bar{Z}	14	Z\bar{Z}
7	GND	15	GND

1.2.4 Conveyor Tracking Board Settings (Factory-default Settings)

The conveyor tracking board is shipped with the following factory-default settings. Check the settings before installing the board into the controller after purchase.

The settings are made for (1) Board setting jumper and (2) Encoder input circuit terminating resistance selection jumper.



HPC1-CTR522F

Board ID setting jumper

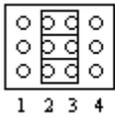
Encoder input circuit terminating resistance selection jumper

P2
P1

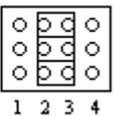
(1) Board ID setting jumper
Set Board ID setting jumper as shown below.



(2) Encoder input circuit terminating resistance selection jumper
Set jumper P1 and P2 to short-circuit between Pin 2 and Pin 3 for all A, B and Z.



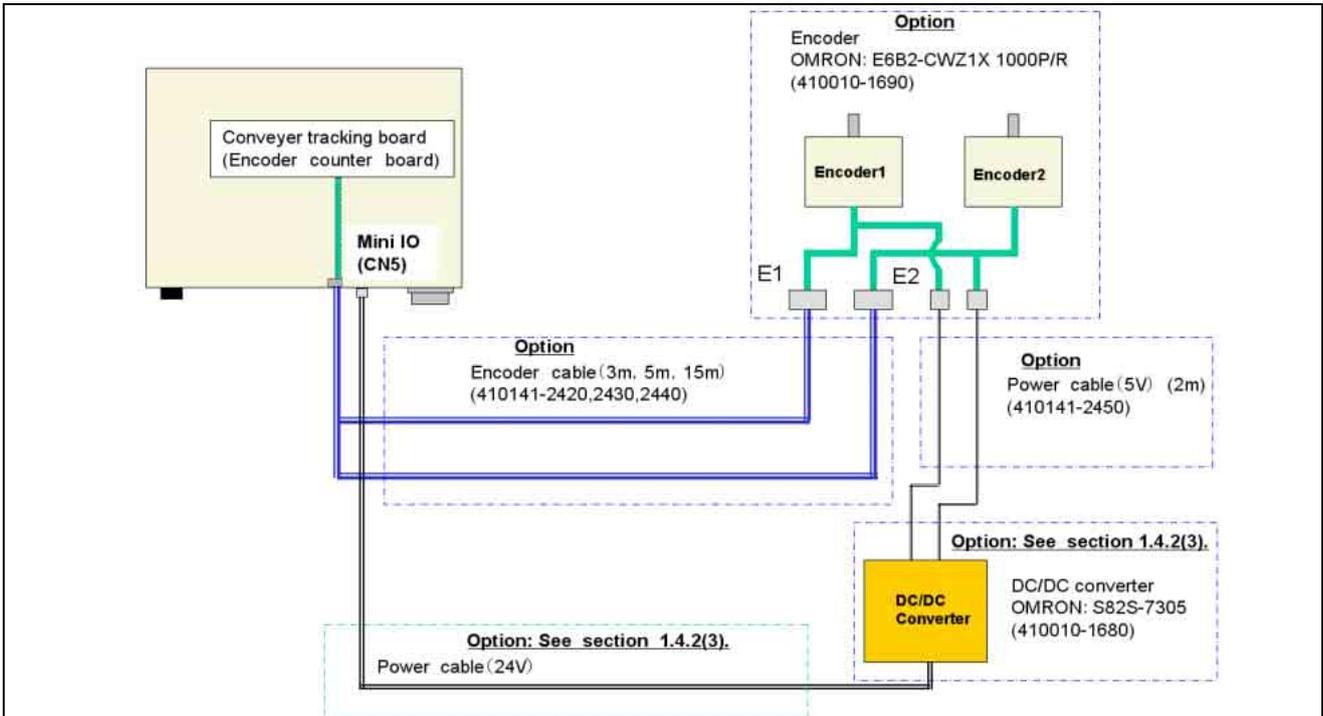
P1



P2

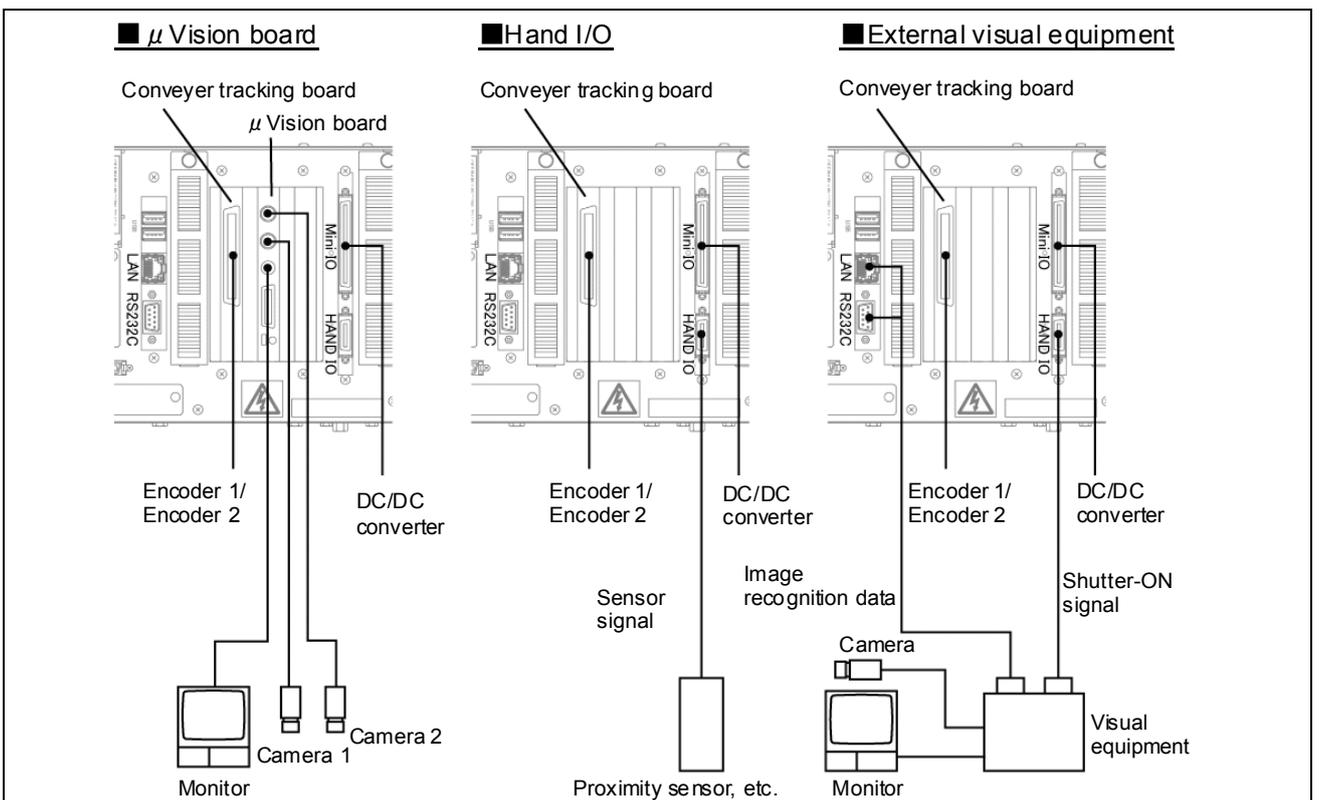
1.3 System Configuration of Conveyer Tracking

1.3.1 Encoder Configuration



Caution: Route the cables properly so that connectors are not to be pulled by a force of 1 kgf or larger. When I/O converter box is used, note that power supply terminals (30, 64) exclusive for tracking encoder on the Mini I/O cannot be used. Set Mini I/O to internal power supply and use 24V output terminal of I/O POWER connector in the I/O converter box, or prepare 24V power supply separately.

1.3.2 Configuration of Sensors and Cameras



1.4 Specifications of Components

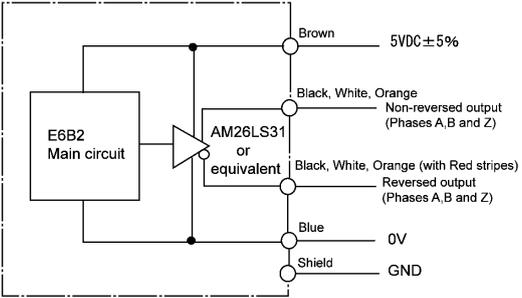
1.4.1 Encoder

(1) Specifications

Be sure to read the INSTRUCTION Sheet enclosed in the package.

Item	Specifications
Part No.	410010-1690
Model	E6B2-CWZ1X (OMRON)
Type	Incremental
Supply voltage	5VDC ($\pm 5\%$)
Current consumption	160 mA max.
Output circuit configuration	Line driver output
Output phase	A, \bar{A}, B, \bar{B}, Z, \bar{Z}
Maximum response frequency	100 kHz
Slewing speed	6000 r/min
Resolution	1000 pulses/revolution (Note 1)
Starting torque	980 μ Nm max.
Moment of inertia	1×10^{-6} kgm ² max.
Shaft loading	Radial: 30 N, Axial: 20 N
Dust & splash proof	IP50
Ambient temperature	-10°C to +70°C
Ambient humidity	35% to 85%RH

OUTPUT CIRCUIT DIAGRAM



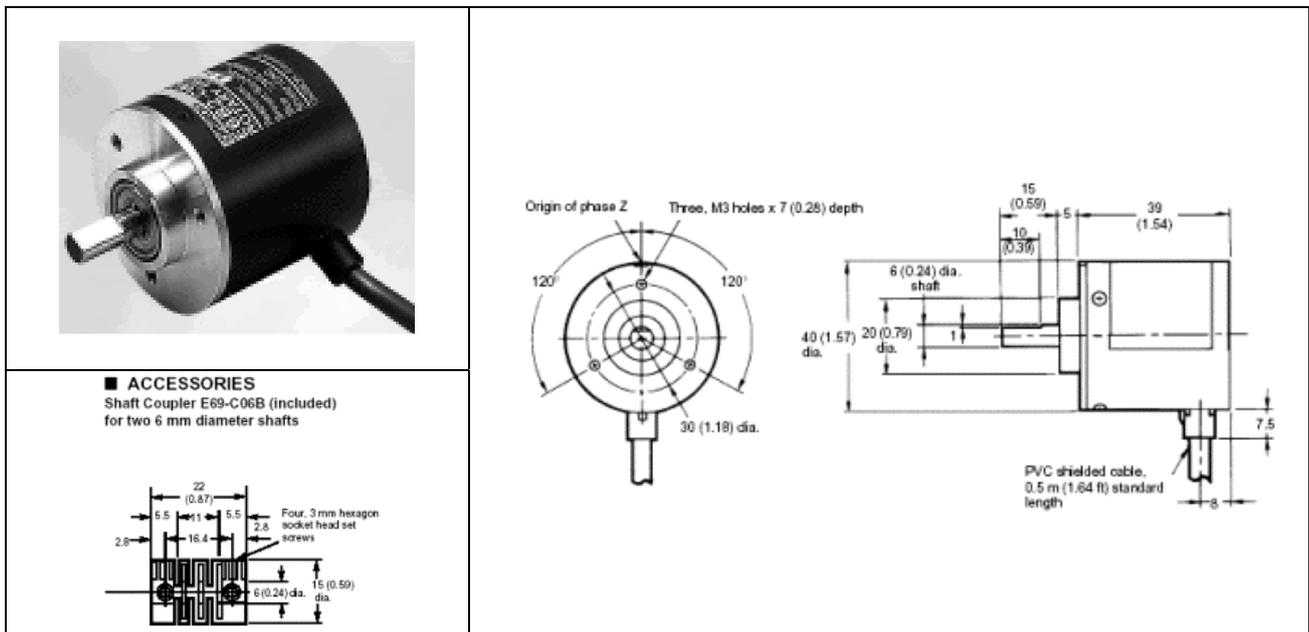
Terminal	Wire color
DC5V	Brown
0V	Blue
A	Black
B	White
Z	Orange
\bar{A}	Black/Red
\bar{B}	White/Red
\bar{Z}	Orange/Red

Note 1: The number of pulses is multiplied by 4, so the count value is 4000 pulses/revolution.

Note 2: Since this equipment is neither dust-proof nor drip-proof, protect it with a cover or something if used in an adverse environment.

Note 3: Protect the encoder connectors by putting covers on them after installation.

(2) Outline drawing



1.4.2 DC/DC Converter

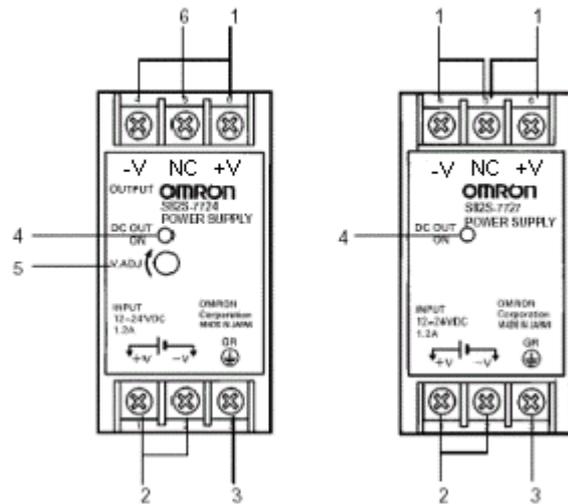
Be sure to read the INSTRUCTION Sheet enclosed in the package.

(1) Specifications

Model	S82S-7305
	DIN Rail Mount
Input voltage	10.2 to 27.5VDC
Output voltage	5V
Poer ratings	3W
Output current	0.6A
Voltage adjustment range	±5%
Ambient temperature	0°C to +50°C
Ambient humidity	25 to 85%RH



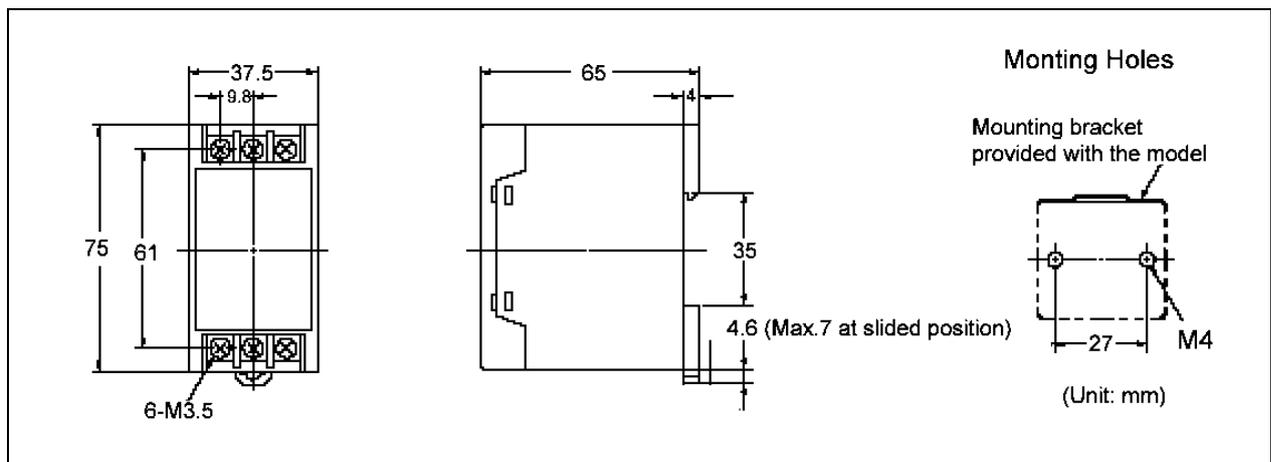
Front Panel and Its Functions



1. DC Output Terminals: Connect load wiring.
2. Input Terminals: Connect input wiring.
3. Ground Terminals: Connect ground wiring.
4. Output LED Indicator: Lights when DC current is being output.
5. V. ADJ Adjuster: Use to adjust the output voltage.
6. NC Terminals: Vacant terminals.

Note: Since this equipment is neither dust-proof nor drip-proof, protect it with a cover or something when used in an adverse environment.

(2) Outer dimensions

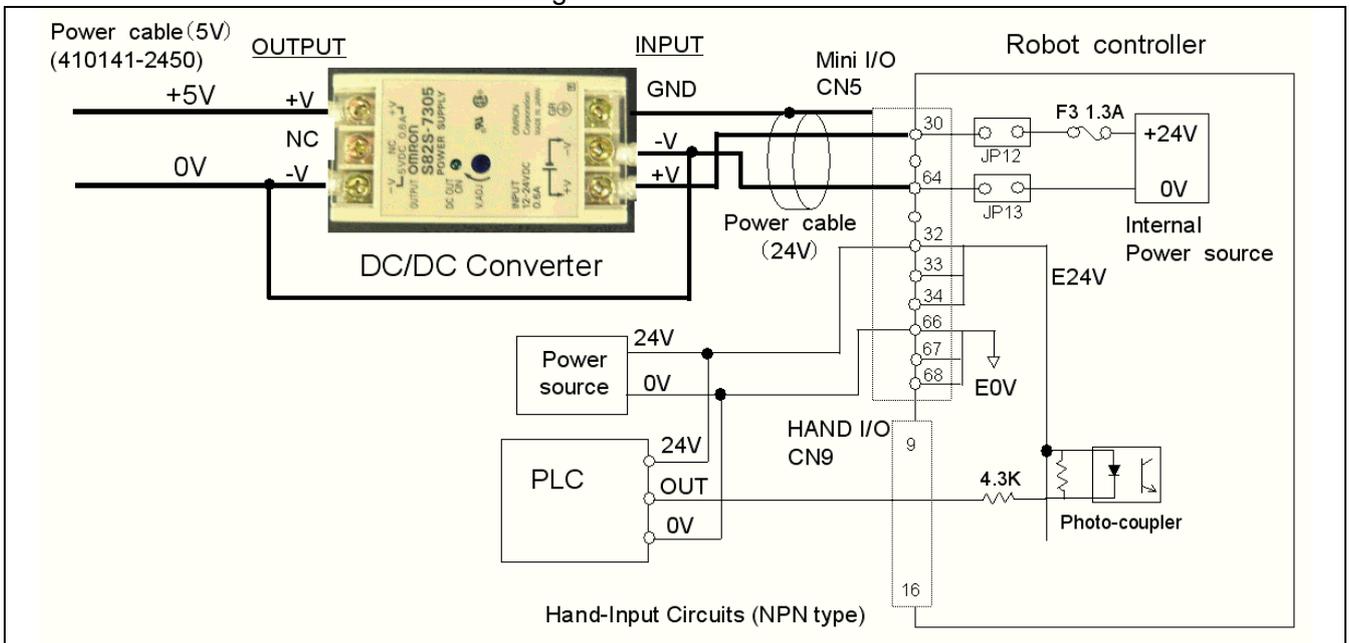


(3) Wiring for DC/DC Converter

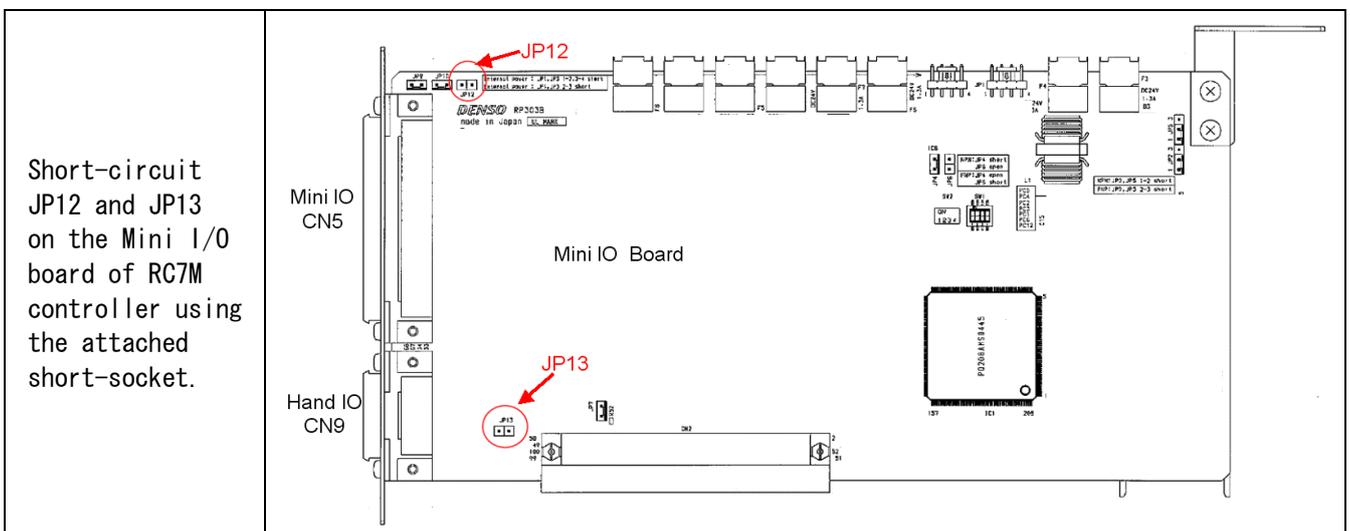
A wiring example when hand input (NPN type) is used is shown in the diagram below. Refer to “RC7M CONTROLLER INTERFACE MANUAL” for details.

Notes:

- (1) Use a twisted pair shielded cable as the power cable (24 V).
- (2) Connect the shield with CN5 shell and the GND terminal of the DC/DC converter.
- (3) The capacity of current for the internal power supply is 1.3 A. When using PLC or a proximity switch, use external power supplies.
- (4) When board is purchased separately, JP12 and JP13 will be open. Short-circuit JP12 and JP13 on mini I/O board using short-socket.
- (5) Fuse is not installed in internal 0V power supply. If external 24V power supply is connected, Mini I/O board will break: Take care not to make the incorrect wiring.



Wiring for DC/DC Converter



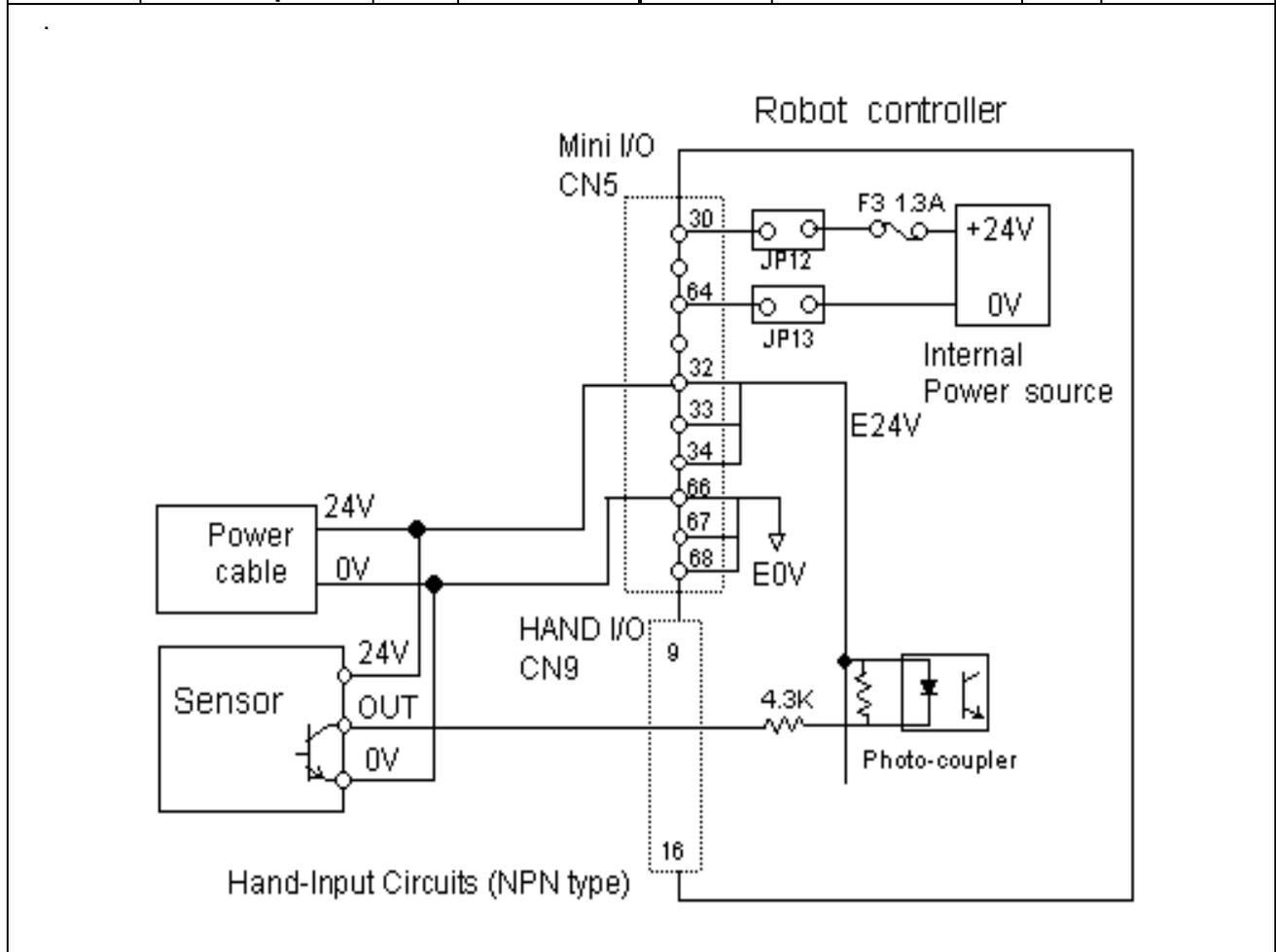
Short-circuit JP12 and JP13 on the Mini I/O board of RC7M controller using the attached short-socket.

Jumper for setting internal power output exclusive for conveyer tracking board (on the Mini I/O board)

1.4.3 Sensor Signal Input Pin Layout (NPN Type I/O)

Table below shows sensor signal input pin layout. Refer to “RC7M CONTROLLER INTERFACE MANUAL” for details.

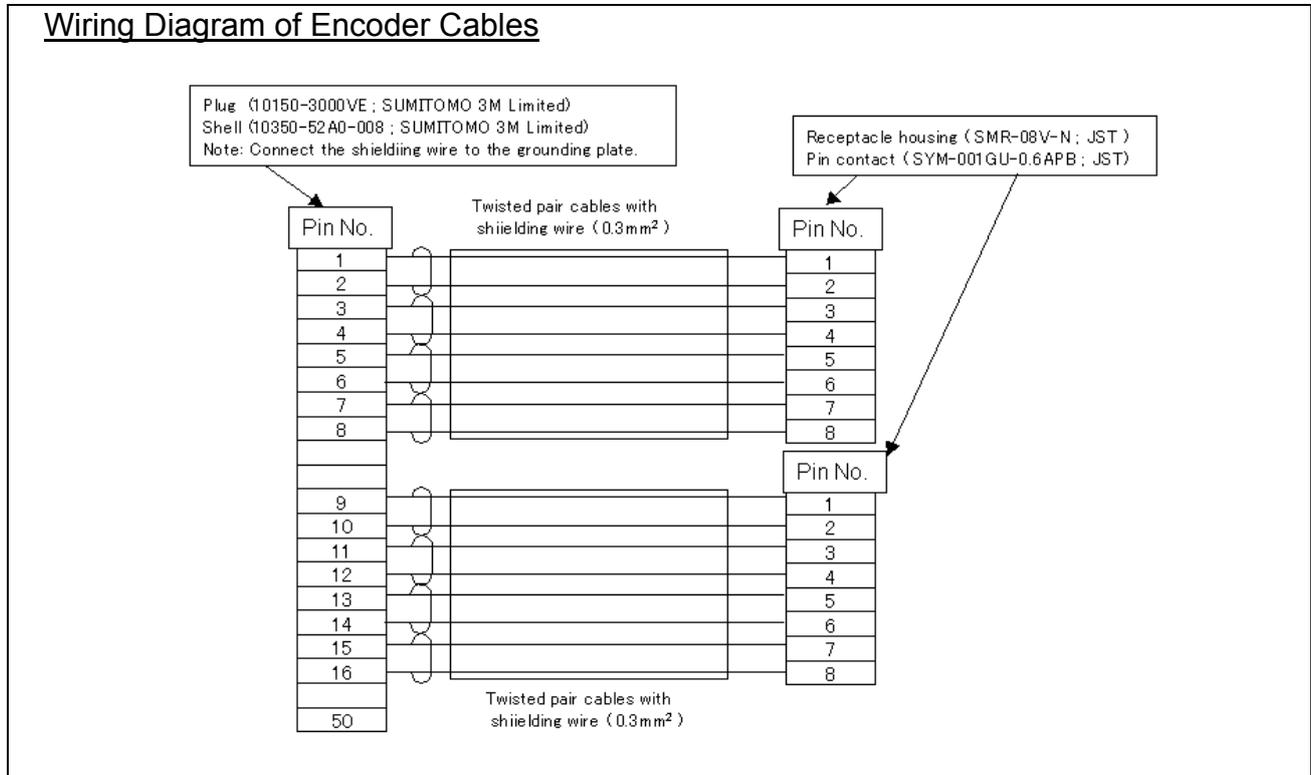
HAND I/O (CN9)							
View from the cable aside							
Terminal No.	Name	Port No.	I/O interrupt No.	Terminal No.	Name	Port No.	I/O interrupt No.
1	Hand output 1	64	-	11	Hand input 3	50	3
2	Hand output 2	65	-	12	Hand input 4	51	4
3	Hand output 3	66	-	13	Hand input 5	52	5
4	Hand output 4	67	-	14	Hand input 6	53	6
5	Hand output 5	68	-	15	Hand input 7	54	7
6	Hand output 6	69	-	16	Hand input 8	55	8
7	Hand output 7	70	-	17	Power E24V for Hand	-	-
8	Hand output 8	71	-	18	Power E0V for Hand	-	-
9	Hand input 1	48	1	19	Not connected	-	-
10	Hand input 2	49	2	20	Not connected	-	-



1.4.4 Wiring Diagram of Optional Cables and Precautions in Wiring

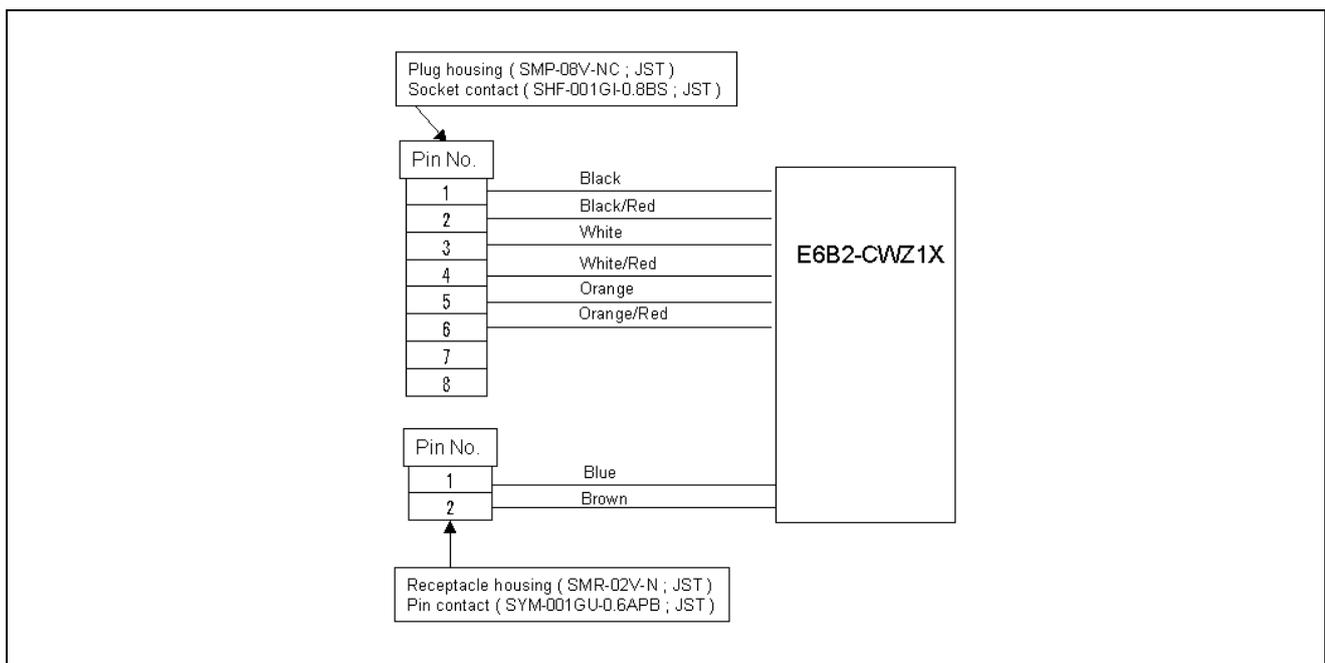
(1) Encoder cable

The following shows a wiring diagram of encoder cables.



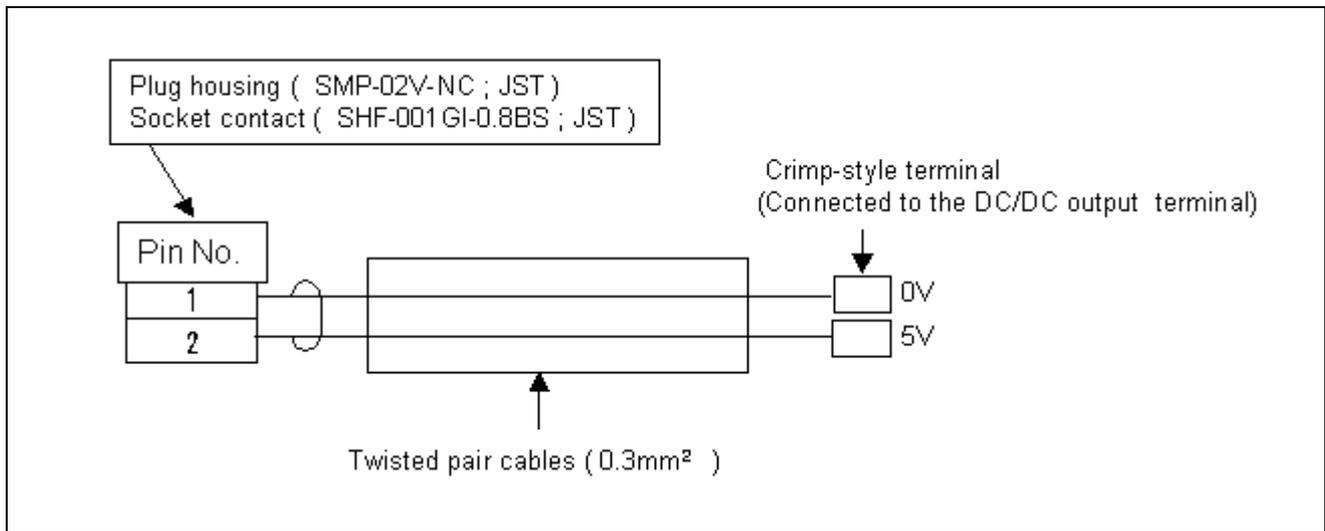
(2) Wire connections of encoder (E6B2-CWZ1X) cables

Connect the encoder cable wires to the following connector pins.



(3) Wiring diagram of 5V power supply cable

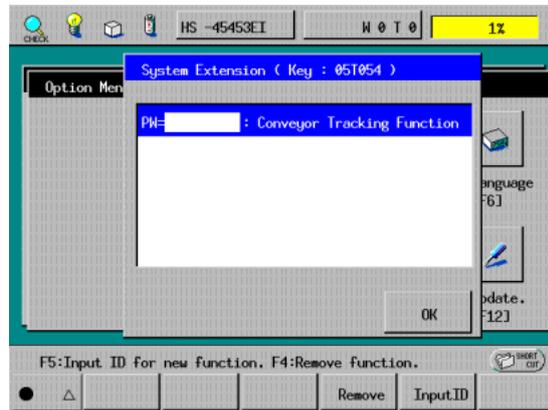
The following shows a wiring diagram of the 5V power supply cable.



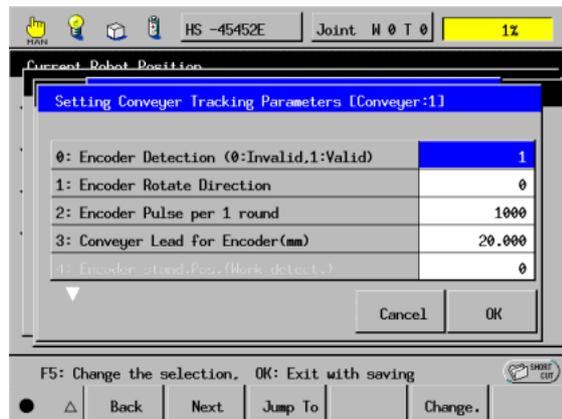
1.5 Hardware Operation Confirming Procedure

- (1) Confirm that the Conveyor tracking function is effective using the teach pendant.

Access: [Top Screen]-[F6 Set]-[F7 Options.]-[F8 Extension]



- (2) Set Conveyor tracking parameters.
Refer to “2.1 Setting Conveyor Tracking Parameters”.



- (3) Confirm the encoder count value.
Refer to “2.1 Setting Conveyor Tracking Parameters”.

Operate the Conveyor and confirm “Encoder current position”.
Confirm that the value changes according to the conveyor operation.

- (4) Confirm the encoder count value when a workpiece is detected.
Refer to “2.1 Setting Conveyor Tracking Parameters”.

Operate the conveyor and confirm “Encoder stand. Pos. (Work detect.)”.
Confirm that the value changes every time a workpiece is detected.

1.6 Configuring Cameras for the Use of μ Vision Board

To use the μ Vision board as a sensor for the conveyer tracking board, it is necessary to configure each camera by using the switches provided on the rear panel and inside the camera.

This section describes settings to be modified from the factory defaults. For detailed configuration, refer to the instruction manual of the camera.

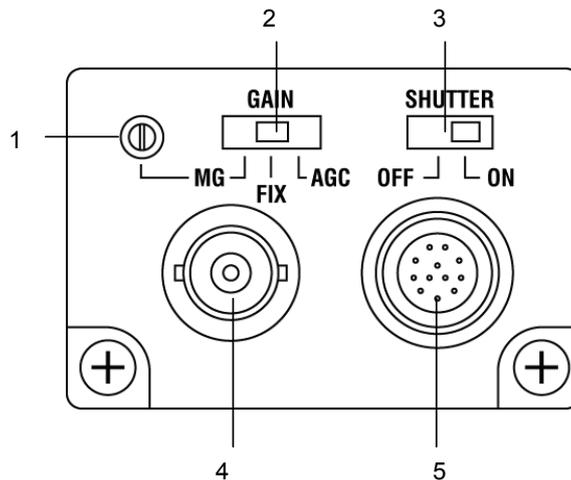
On the rear panel, use the electronic shutter ON/OFF switch.

To access switches inside the camera, remove the camera cover and use the mode change switch (DIP switch) and the electronic shutter speed switch (rotary switch).

1.6.1 Setting on the Rear Panel

Turn the electronic shutter ON/OFF switch (SHUTTER, #3 in the figure below) to the ON position. The default is OFF.

Do not disturb other switch settings.

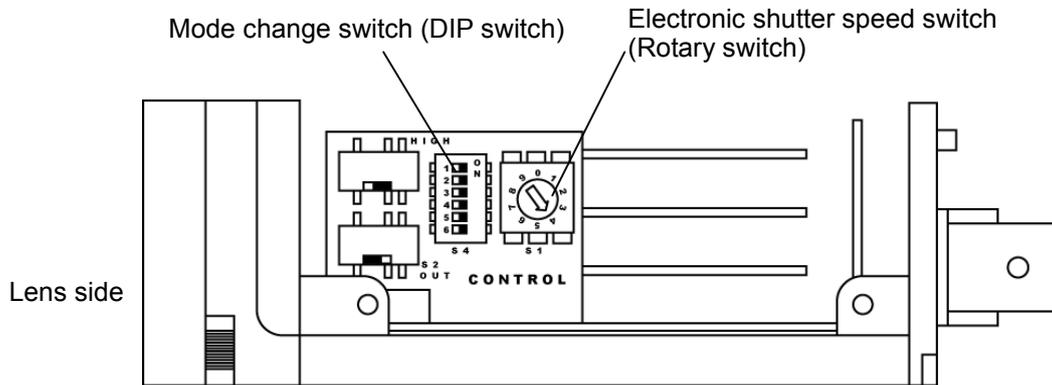


1	Gain adjuster	Adjusts the manual gain. The factory default should be retained.
2	Gain switch	Switches between manual gain, fixed gain, and automatic gain control. The factory default (FIX) should be retained.
3	Electronic shutter ON/OFF switch	Turns the electronic shutter on and off. This should be turned on for the μ Vision board.
4	VIDEO OUT terminal	Video output terminal for monitoring. This is not used for the μ Vision board.
5	DC IN/SYNC terminal	Terminal used for power supply to the camera, shutter signal input, and image output. This should be connected to the μ Vision board.

1.6.2 Setting Inside a Camera

Remove the five screws from the camera cover with a precision screwdriver and take off the cover.

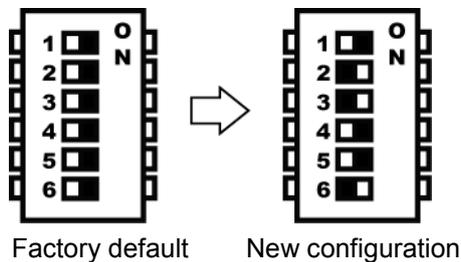
Change the settings of the mode change switch (DIP switch) and electronic shutter speed switch (rotary switch) as detailed below. Do not disturb other switch settings.



View from the side, after removal of the cover

[1] Mode change switch (DIP switch)

Turn selectors 2, 3, and 6 on the mode change switch to the ON position. Other selectors should be OFF.



Selector #	Function name	Switch configuration (New settings in gray)	
		OFF	ON
1	Gamma correction	1.0	0.45
2	CCD accumulation mode	Frame accumulation	Field accumulation
3	Restart/reset	OFF	ON
4	Special shutter	OFF	ON
5	Reserved.	—	—
6	VD/FLD output	VD output	FLD output

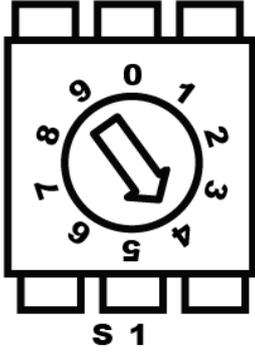
[2] Electronic shutter speed switch (Rotary switch)

Use the electronic shutter speed switch to select the appropriate shutter speed. The factory default is 1/1000 second (position "4").

The relationship between the switch position and shutter speed is shown below.

It is necessary to adjust the electronic shutter speed depending upon the illumination, angle of view, camera-subject distance, conveyer speed, and other environmental conditions.

If the image blurs, increase the shutter speed. If the image is dark, adjust the illumination or the aperture value of the lens to make the image properly bright.

	No.	Shutter speed
	0	Normal
	1	1/125
	2	1/250
	3	1/500
	4	1/1000 (factory default)
	5	1/2000
	6	1/4000
	7	1/10000
	8	Flickerless
9	Flickerless	

[3] Notes

To enable the settings made above, use the CAMIN command to switch to the conveyer tracking mode and the CAMMODE command to specify the "Reset function" and "Field reading" with their parameters.

For details, refer to the PROGRAMMER'S MANUAL I.

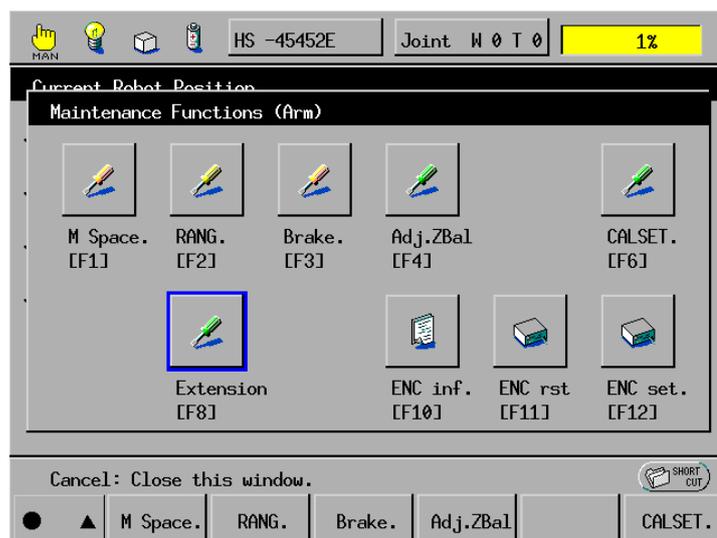
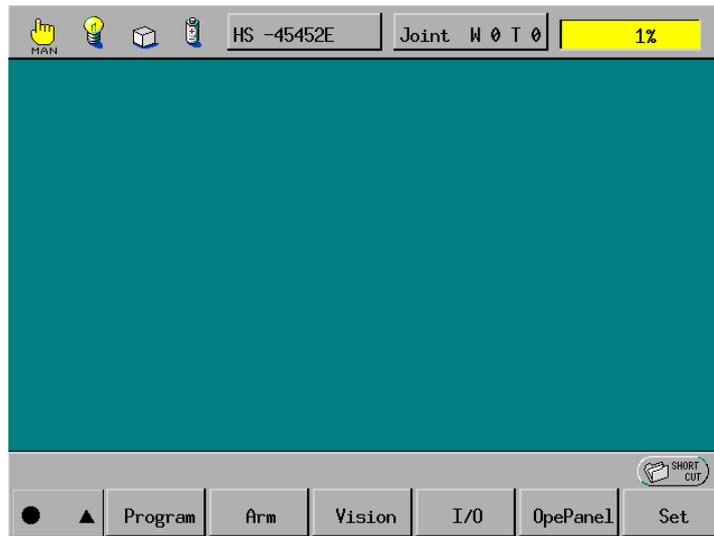
Chapter 2 Conveyor Tracking Adjustment

2.1 Setting Conveyor Tracking Parameters

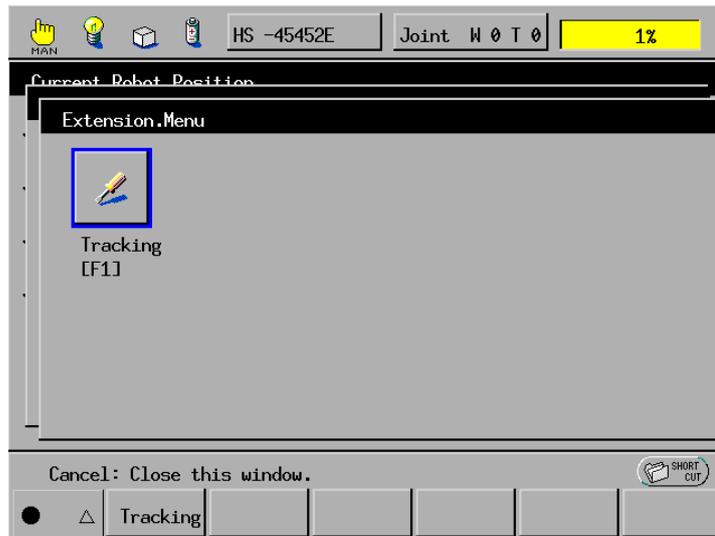
To use the conveyor tracking function, you need to set conveyor tracking parameters with a teach pendant.

2.1.1 How to Set Conveyor Tracking Parameters

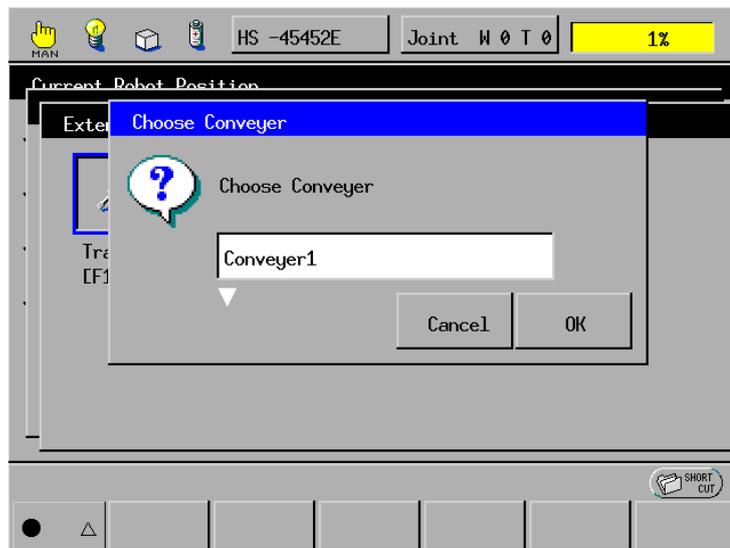
- (1) From the basic screen, select function items, in this order: [F2 Arm] → Shift → [F12 Maint.].



(2) From the Maintenance screen, select [F8 Extension].



(3) Pressing [F1 Tracking] displays the parameter setting screen.
This screen is displayed only when the conveyer tracking function is enabled as an extension function.

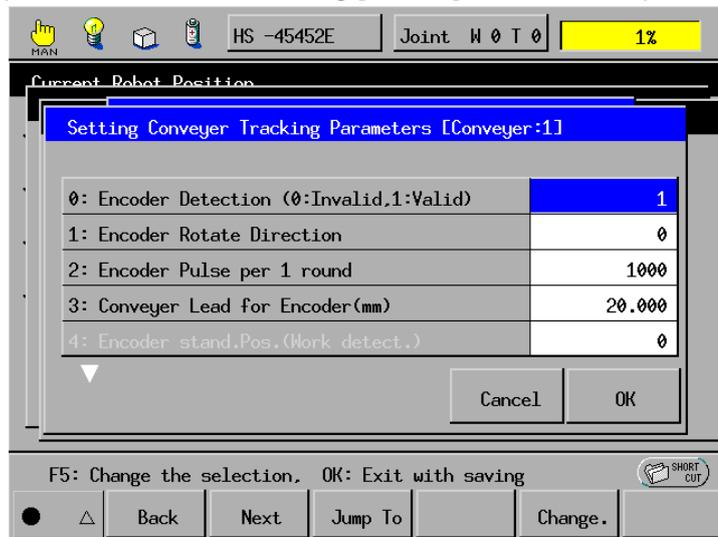


(4) Use the Δ key, ∇ key, or the jog dial to select the conveyer number (Conveyer1, Conveyer2).

The conveyer number is the same as the encode number, which depends on the connector number (E1, E2) of the encoder cable. The encoder connected to E1 is referred to as encoder 1, and the encoder connected to E2 is called encoder 2. For details, refer to “1.3.1 Encoder Configuration” in “Hardware Setting for Conveyer Tracking”.

When using two conveyers, you need to set parameters for each of conveyers 1 and 2.

- (5) Selecting the conveyer number and then pressing [OK] displays the parameter screen. Pressing [Cancel] returns to the preceding screen.



- (6) Select a parameter you want, press [Change], and then enter a numeric value. Pressing [OK] now changes the parameter value to the new one. For the content of parameters, see “Table: List of Conveyer Tracking Parameters”.

Note: There are two types of parameters: those immediately enabled after their values are changed and those enabled after the controller power is turned off and then on. When you change parameters of the latter type, turn the controller power off and then turn it on. For details, see “Table: List of Conveyer Tracking Parameters”.

Table List of Conveyor Tracking Parameters

Parameter name	Entry range	Factory default	Unit	Description	Remarks	Controller restart
Encoder Detection (0: Invalid, 1: Valid)	0 to 1	0		Set to "1" to use the function.		Needed
Encoder Rotate Direction	0 to 1	0		Set to "1" to invert the direction (+/-) of the encoder count.	Set the parameter so that the direction of the conveyor operation is "+".	Needed
Encoder Pulse per 1 Round	0 to 100000	1000	[P/R]	Set the number of pulses per rotation of the encoder.	The number is multiplied by 4 at the counter board. The encoder count value is 4 times the number of pulses.	Needed
Conveyor Lead for Encoder	0 to 100000	20	[mm]	Set the conveyor operation amount per rotation of the encoder.		Needed
Encoder stand. Pos. (Work detect.)				The encoder count value when the latest workpiece is detected is displayed.	This parameter cannot be changed.	
Encoder stand. Pos. (motion start.)				The encoder count value when the workpiece subject to tracking is detected is displayed.	This parameter cannot be changed.	
Encoder current position				The current encoder count value is displayed.	This parameter cannot be changed.	
Conveyor CALSET (X,Y)	-360 to 360	0	[deg]	Set the direction of the conveyor operation (rotation angle with respect to the X-axis on the XY plane).	See "Conveyor Calibration".	Not needed
Conveyor CALSET (Z)	-360 to 360	0	[deg]	Set the direction of the conveyor operation (rotation angle with respect to the XY plane).	See "Conveyor Calibration".	Not needed
Encoder max. speed (rpm)	0 to 20000	6000	[rpm]	Set the upper speed limit for the encoder.		Not needed
Encoder min. speed (rpm)	0 to 20000	0	[rpm]	Set the lower speed limit for the encoder.		Not needed
Conveyor Track Upper Pos. (mm)	-2000 to 2000	200	[mm]	Set the range (+ side) of robot positions allowing tracking.	(Note1)	Not needed
Conveyor Track Lower Pos. (mm)	-2000 to 2000	-200	[mm]	Set the range (- side) of robot positions allowing tracking.	(Note1)	Not needed
Tracking start Pos.(+) (mm)	-2000 to 2000	200	[mm]	Set the workpiece position (+ side) allowing the start of tracking operation with the tracking operation wait command.	See the description of the operation wait command "WAITTRACKMOVE".	Not needed
Tracking start Pos.(-) (mm)	-2000 to 2000	-200	[mm]	Set the workpiece position (- side) allowing the start of tracking operation with the tracking operation wait command.	See the description of the operation wait command "WAITTRACKMOVE".	Not needed
Tracking accel rate	0 to 100	25	[%]	Set the acceleration at the start of tracking operation and the deceleration at the end of tracking operation.		Not needed
Tracking Position Adjustment value	0 to 50	7		Set the positional compensation amount for delay in tracking operation processing.	Increase the value if the tracking operation position lags in the direction of the conveyor's upstream.	Not needed
I/O Interrupt Detection (0:Invalid, 1:Valid)	0 to 1	0		Set to "1" to detect the workpiece by hand I/O interrupt.		Needed
I/O Interrupt Number	1 to 8	0		Set the hand I/O Interrupt number to be used for workpiece detection.		Needed

Note1: When setting "Conveyor Track Upper Pos." and "Conveyor Track Lower Pos.", pay attention to their changing order. Following cases will occur error:

Entering value of "Conveyor Track Upper Pos." < Current value of "Conveyor Track Lower Pos."

Entering value of "Conveyor Track Lower Pos." > Current value of "Conveyor Track Upper Pos."

(Continued on following page)

Parameter name	Entry range	Factory default	Unit	Description	Remarks	Controller restart
Edge Trigger (0:Up, 1:Down)	0 to 1	0		When the parameter is set to "0", an interrupt occurs at a rising edge (OFF→ON). When the parameter is set to "1", an interrupt occurs at a falling edge (ON→OFF).		Needed
Workpiece detection position accuracy	0 to 1000	5	mm	Set a range in which detected workpieces are recognized as the same after the same workpiece has been detected for multiple times as a result of repeated visual recognition process.	Set the parameter to a value smaller than the minimum distance between workpieces.	Not needed
Interrupt data setting	0 to 10	0		Set a condition for setting encoder data into the conveyer tracking data buffer at hand I/O interrupt.	Set this value to 1 if external visual equipment is used.	Not needed
Interrupt delay correction value	0 to 30	0	2ms	Set a wait length from when a hand I/O interrupt occurs to when an encoder value is obtained.	For adjustment method, refer to "3.4 Conveyer tracking accuracy checking procedures".	Not needed

2.1.2 Detailed Explanation of Parameters

1. Encoder Detection

To use the encoder, set this parameter to "1". If it is set to "0", the encoder value is not counted.

2. Encoder Rotate Direction

Set this parameter so that the encoder count value increases (change in the direction of +) when the conveyer operates.

The encoder count value is displayed in "Encoder current position". Check the encoder count value before the conveyer operation, and close the parameter window. After the conveyer operation, open the parameter window again and check the encoder count value.

If the encoder count value after the conveyer operation decreases, change "Encoder Rotate Direction" accordingly.

3. Encoder Pulse per 1 round

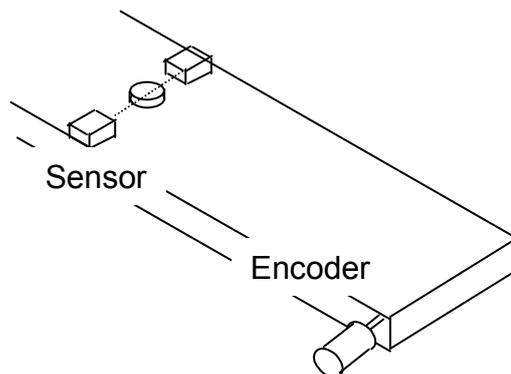
Set the number of pulses per rotation of the encoder. When you use E6B2-CWZ1X, set this parameter to 1000. The encoder pulses undergo frequency division at the conveyer tracking board, where four times the number of pulses is counted. Therefore, the encoder count value per rotation of the encoder is four times the number of encoder pulses.

4. Conveyer Lead for Encoder

Conveyer movement amount per rotation of the encoder. This parameter checks the setting value for correctness when conveyer calibration is being executed.

5. Encoder stand. Pos. (Work detect.)

This parameter indicates the encoder count value when a workpiece is detected with a camera or sensor. The value changes each time a workpiece is detected.



6. Encoder stand. Pos. (motion start.)

This parameter indicates the encoder count value when the workpiece subject to the current tracking is detected. The value changes each time the workpiece subject to tracking changes.

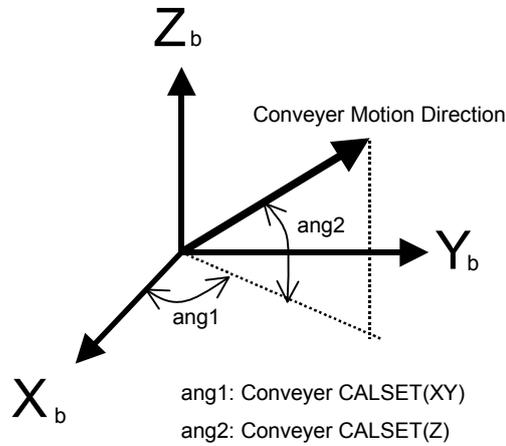
7. Encoder current position

This parameter indicates the current encoder count value.

Confirm that when the encoder makes one rotation, the encoder count value increases four times the encoder pulse value (Encoder Pulse per 1 round).

- 8. Conveyer CALSET(X,Y)
- 9. Conveyer CALSET(Z)

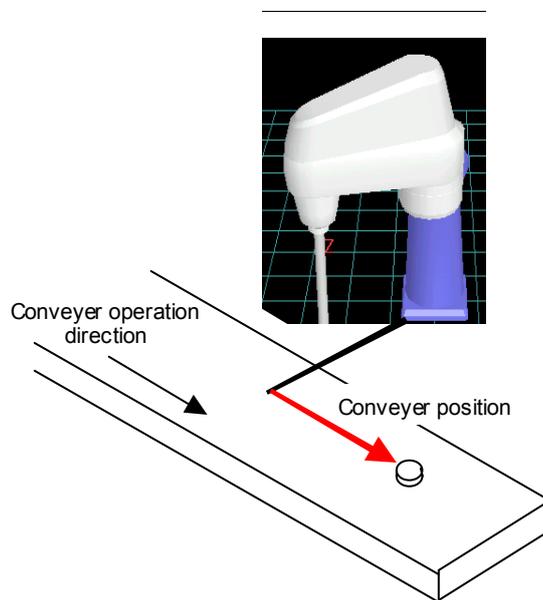
These parameters set conveyer operation directions with respect to the robot base coordinates. Each corresponds to the angle shown in the following figure.



When conveyer calibration is executed, the computation value is applied.

- 10. Conveyer Track Upper Pos. (mm)
- 11. Conveyer Track Lower Pos. (mm)

These parameters set the tracking operation range with conveyer positions. As shown in the figure below, the conveyer position is expressed by the distance from the line, drawn from the robot home position perpendicularly to the direction of conveyer operation, to the point in question in the conveyers. The conveyer operation direction is regarded as the plus (+) side.



- 12. Tracking start Pos.(+) (mm)
- 13. Tracking start Pos.(-) (mm)

These parameters set the tracking operation start position with conveyer positions.

The tracking operation wait command (WAITTRACKMOVE) is placed in the wait status until the workpiece subject to tracking comes into the specified tracking operation start range.

- 14. Tracking accel rate

When the tracking operation starts, the robot is accelerated until its speed becomes the same as the conveyer speed. When the tracking operation terminates, the robot is decelerated from the conveyer speed. This parameter sets such acceleration/deceleration speed at the time of the tracking operation start/stop. The greater the setting value, the shorter the switching time at the start or stop of the tracking operation. Note, however, that too great a setting value may cause “commanded acceleration error.”

- 15. Tracking Pos. Adjustment value

When chucking the workpiece on the conveyer, the chucking position may shift according to a rise in the conveyer speed. If that is the case, use this parameter to adjust the position. If the shift is in the direction of the upstream of the conveyer, increase the value. If the shift is in the direction of the downstream of the conveyer, decrease the value.

Note: If the chucking position shifts when the conveyer speed is low or when the conveyer stops, re-teaching is required.

- 16. I/O Detection

To detect workpieces with Hand I/O input interrupts, set this parameter to “1”. Also, confirm that “Encoder Detection” is set to 1.

- 17. I/O Port Number

This parameter sets interrupt numbers for Hand I/O input connection destination. The following table shows the correspondence between Hand I/O input interrupt numbers, terminal numbers, and port numbers:

Hand I/O (CN9)			
Terminal No.	Name	Port No.	I/O interrupt No.
9	Hand input1	48	1
10	Hand input2	49	2
11	Hand input 3	50	3
12	Hand input 4	51	4
13	Hand input 5	52	5
14	Hand input 6	53	6
15	Hand input 7	54	7
16	Hand input 8	55	8

- 18. Edge Trigger

This parameter sets the interrupt occurrence timing for Hand I/O input.

When “0” is set, an interrupt is generated to detect a workpiece when the input signal specified in “I/O Port Number” changes from OFF to ON (rising edge).

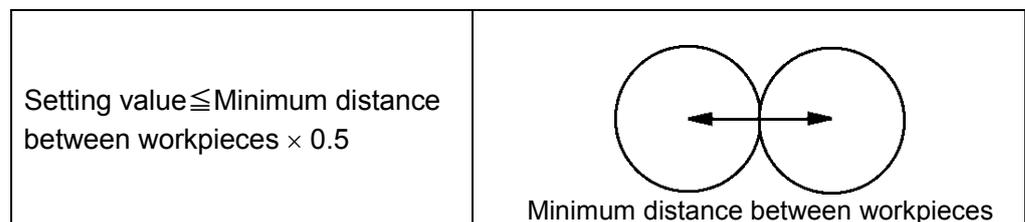
When “1” is set, an interrupt is generated to detect a workpiece when the input signal specified in “I/O Port Number” changes from ON to OFF (falling edge).

19. Workpiece detection position accuracy

The same workpiece may be detected for multiple times on a conveyor as a result of repeated visual recognition process or others. In this case, detected workpieces are checked if they are the same based on the difference in the workpiece positions on the conveyor. This parameter sets a workpiece position error range in which workpieces are recognized as the same.

If the parameter is set to a value larger than the minimum distance between workpieces shown below, two different workpieces are falsely recognized as the same. And if the parameter is set to a value smaller than the minimum distance between workpieces, the same workpiece is falsely recognized as multiple, different workpieces.

Set the parameter using the following guideline.



20. Interrupt data setting

This parameter sets a condition for setting encoder value into the conveyor tracking data buffer when a hand I/O interrupt occurs.

0: Upon execution of TrackDataSet command, all the encoder values obtained during hand I/O interrupts are set into the conveyor tracking data buffer.

1: Upon execution of TrackDataSet, the encoder values obtained during the latest hand I/O interrupt are set into the conveyor tracking data buffer.

If you use the shutter ON signal as the hand I/O interrupt signal by using external visual equipment, set this parameter to 1.

21. Interrupt delay correction value

This parameter sets a wait length from when a hand I/O interrupt occurs to when an encoder value is obtained.

The wait length is "Set value $\times 2$ (ms)".

For how to adjust setting values, refer to the procedures (3) in "3.4.2 When visual equipment is used" of "3.4 Conveyor tracking accuracy checking procedures".

2.2 Conveyor Calibration

The setting of conveyor operation direction is indispensable to conveyor tracking. To set the conveyor operation direction, use the conveyor tracking parameters (Conveyor CALSET(X,Y), Conveyor CALSET(Z)). Conveyor calibration refers to the work of setting conveyor tracking parameters (Conveyor CALSET(X,Y), Conveyor CALSET(Z)).

2.2.1 How to Execute Conveyor Calibration

There are two methods as shown below:

- (1) Calculate conveyor tracking parameters (Conveyor CALSET(X,Y), Conveyor CALSET(Z)) based on the layout drawing of the conveyor and then enter the calculated value.
- (2) Use the library **CONVCAL** for automatic setting.
The method in (2) above uses actual equipment and its precision is higher than (1). The procedure of (2) is explained in the following section.

2.2.2 Conveyor Calibration Execution Procedure

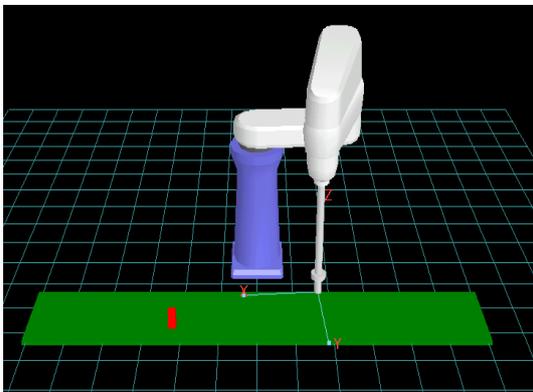
(1) Setting conveyor tracking parameters

Check whether the following conveyor tracking parameters are set correctly:

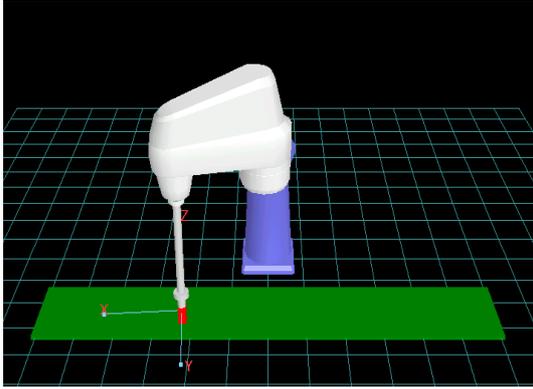
- a) Encoder Detection
Check that "1" (Valid) is set for this parameter.
- b) Encoder Rotate Direction
Check that when the conveyor operates, the encoder count value is incremented.
- c) Encoder Pulse per Round
- d) Conveyor Lead for Encoder
Check the conveyor operation amount per rotation of the encoder.

(2) Execution of the conveyor calibration library (**CONVCAL**)

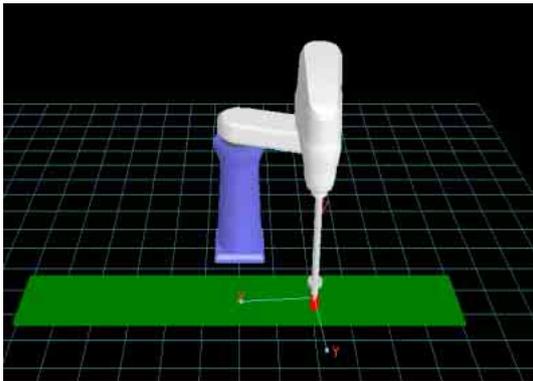
(2-1) Outline of the conveyor calibration library (**CONVCAL**)



- a) Set a reference point on the conveyor (placing a workpiece or a mark).
- b) Move the conveyor so that the reference point falls within the robot operation range.



- c) Move the robot to the reference point and read the position data by teaching.
(Measurement point 1)



- d) Operate the conveyer and move the reference point.
- e) Move the robot to the reference point again and read the position data by teaching.
(Measurement point 2)

- f) The conveyer calibration data is computed based on the robot positions and encoder values for measurement points 1 and 2.

Note: Specify accurate positions for measurement points 1 and 2. If you give rough positions, the tracking operation precision will deteriorate.

(2-2) Execution procedure for the conveyer calibration library (**CONVCAL**)

The execution procedure for the conveyer calibration library (**CONVCAL**) is explained below using the pendant screen.

- a) Set a reference point on the conveyer and move the conveyer until the reference point comes within the robot operation range. Move the robot manually to the reference point. This position is measurement point 1.
- b) Switch to the teach check mode and execute the program **CONVCAL**.
The library argument setting window is displayed.
CONVNO is the conveyer number. Enter the conveyer number for calibration.

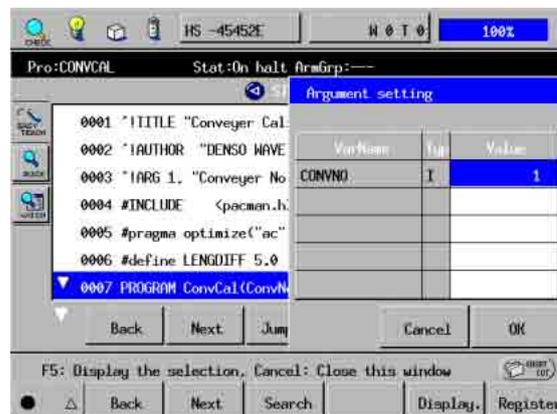


Press "Display".

The parameter display screen is displayed.

Press "Change" and enter the conveyer number.

Pressing the Cancel key after entering a numeric value will return to the original screen.



Check that the value of CONVNO is the same as the conveyer number for calibration.

- c) Execute the program step by step. After several steps are executed, the following message is displayed:



If the conveyer number is set incorrectly, an error is indicated and the program terminates.

- d) Execute the library step by step until the following message is displayed:



Confirm that the robot has moved to measurement point 1 and then press OK. If the robot has not reached measurement point 1, switch to the manual mode and move the robot to that point.

- e) Execute the library step by step until the following message is displayed:



Press OK.

Operate the conveyer to move the reference point. Move it to a position away from measurement point 1 within the robot operation range, and then stop the conveyer.

Switch to the manual mode and move the robot to the reference point. This position is measurement point 2.

- f) Switch to the teach check mode and execute the library step by step. When the following message is displayed, conveyor calibration is completed.



If the conveyer tracking parameters (Encoder Pulse per Round, Conveyer Lead for Encoder) are set incorrectly, an error is indicated. In that case, correct the conveyer tracking parameters and execute conveyer calibration once again.

2.3 Camera Calibration

2.3.1 General

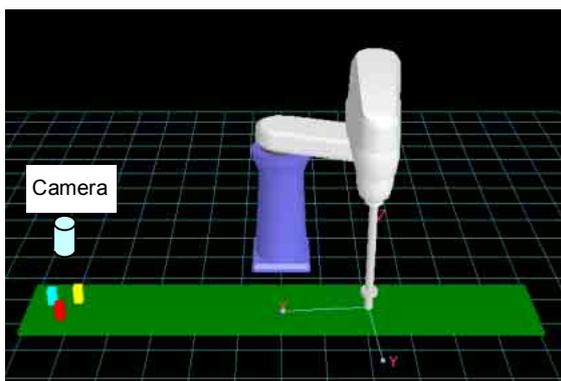
To recognize workpieces with camera, camera calibration is required. Camera calibration is a task of computing relational formulas between robot coordinates and camera coordinates by using WINCAPSIII Vision Manager. (Refer to the Vision Manager in the WINCAPSIII Guide for details.)

In camera calibration, the robot is moved to the mark position recognized visually, and the relational formula between the robot coordinates and camera coordinates is computed. However, at the time of conveyer tracking, the visual recognition position may be distant from the robot work position and therefore it may be impossible to move the robot to the mark position recognized visually. In that case, execute camera calibration in the following procedure using the library CALCCAMCALPOS:

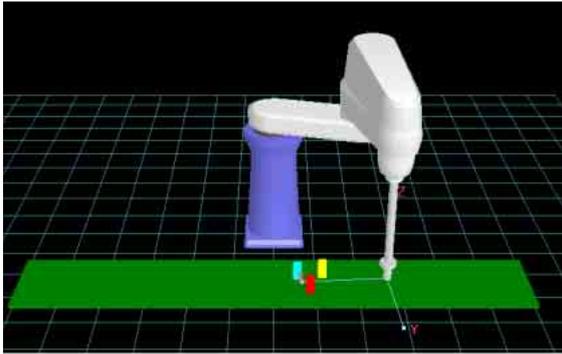
- (1) Prepare three calibration marks and make visual recognition.
- (2) Operate the conveyer, and when three calibration marks come within the robot operation range, stop the conveyer.
- (3) Move the robot to the three calibration mark positions and read the position data by teaching. Based on the conveyer movement amount, convert the teaching position data to the camera recognition position data. Using the conversion result, compute the relational formula between the robot coordinates and the camera coordinates.

Note: Before executing camera calibration, complete conveyer calibration. If you execute camera calibration with conveyer calibration incomplete, correct results of camera calibration cannot be obtained.

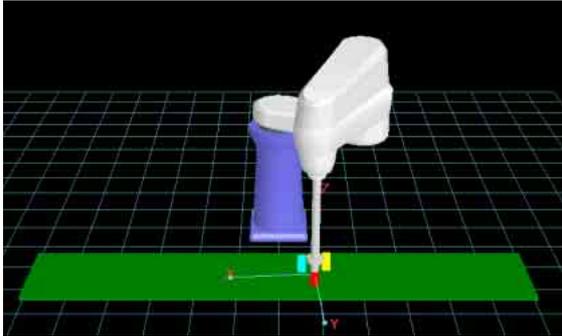
Execution of camera calibration is outlined below.



- a) Prepare three calibration marks, and visually recognize the calibration marks with the conveyer placed in stop status.
- b) Enter the vision coordinates using calibration in the WINCAPSIII Vision Manager.



c) Operate the conveyor, and when the three calibration marks come within the robot operation range, stop the conveyor.



d) Read the position data of three calibration marks by teaching them one by one.

e) Based on the conveyor movement amount, convert the teaching position data to the camera recognition position data. The conversion result is stored in variables.

f) Execute camera calibration using the camera calibration tool of the vision manager.

2.3.2 Execution Procedure for Camera Calibration Library (CALCCAMCALPOS)

The execution procedure for the camera calibration library (CALCCAMCALPOS) is explained below using the pendant screen.

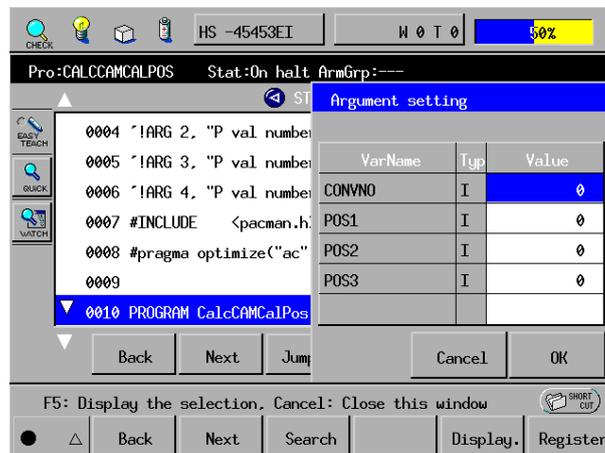
a) Prepare three camera calibration marks, and visually recognize the camera calibration marks with the conveyor placed in stop status.

b) Enter the camera coordinates using calibration in the WINCAPSIII Vision Manager.

In camera calibration, find the robot coordinates and camera coordinates for each of the three marks and compute the relational formula between the robot coordinates and the camera coordinates. The three marks are called mark 1, mark 2, and mark 3 in the following explanations:

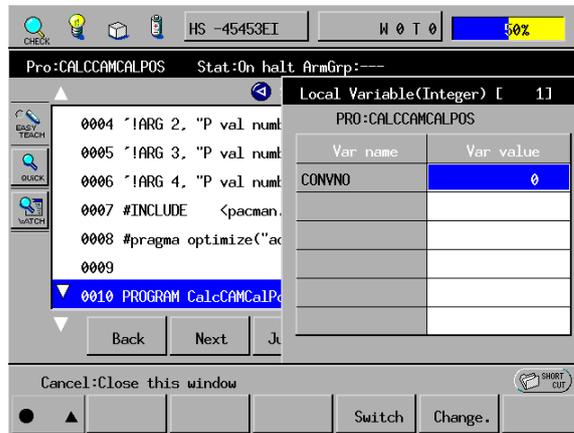
c) Switch to the teach check mode and execute the library CALCCAMCALPOS.

The library argument setting window is displayed.



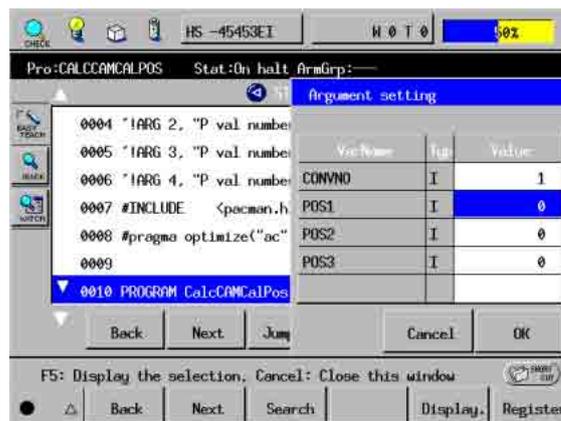
CONVNO is the conveyer number. Enter the conveyer number for calibration.
 POS1 is a P-type variable number to store the robot coordinates for mark 1. Likewise, POS2 and POS3 are P-type variable numbers to store the robot coordinates for mark 2 and mark 3, respectively. Set the numbers so that they are stored in separate P-type variables.

Move the cursor to CONVNO and press "Display" to display the parameter display screen.



Press "Change" and enter the conveyer number.

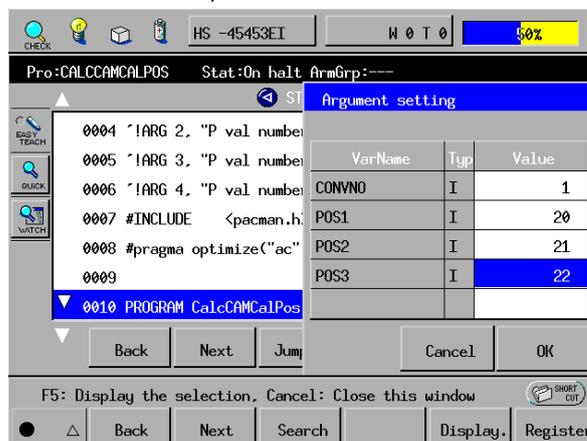
Pressing the Cancel key after entering the numeric value returns to the original screen.



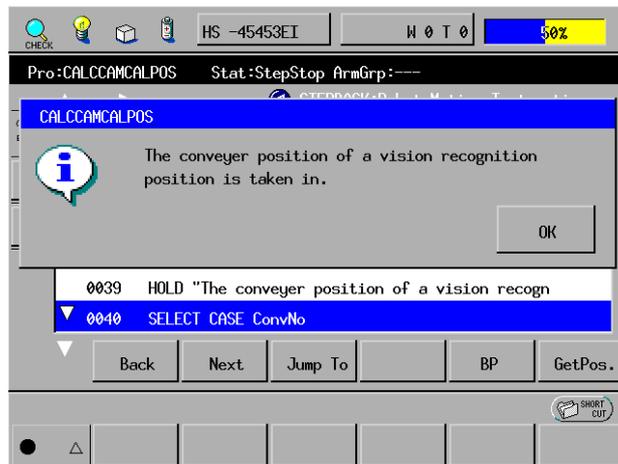
Likewise, enter numeric values sequentially by moving the cursor to POS1 to POS3.

Enter all numeric values for CONVNO and POS1 to POS3.

(The diagram below is an example of storing conveyer number 1, and mark 1 in P20, mark 2 in P21 and mark 3 in P22.)

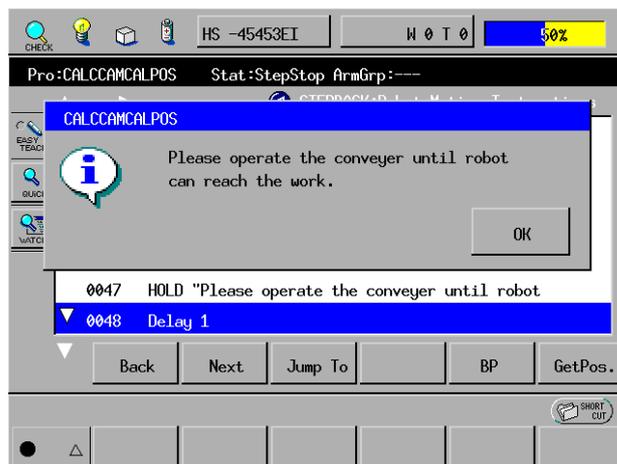


- d) Execute the library step by step. When several steps are executed, the following message is displayed:



Confirm that the conveyer is placed in the stop status at the camera calibration mark recognition position, and then press OK. Execute the library step by step continuously.

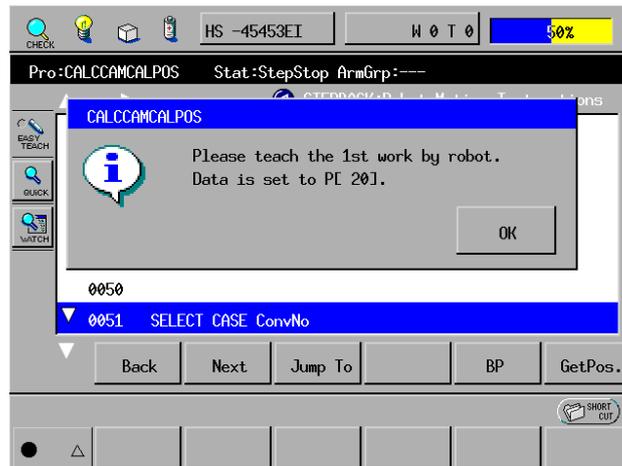
- e) Execute the library step by step until the following message is displayed:



Press OK.

Operate the conveyer, and when the three calibration marks come into the robot operation range, stop the conveyer.

f) Execute the library step by step until the following message is displayed:

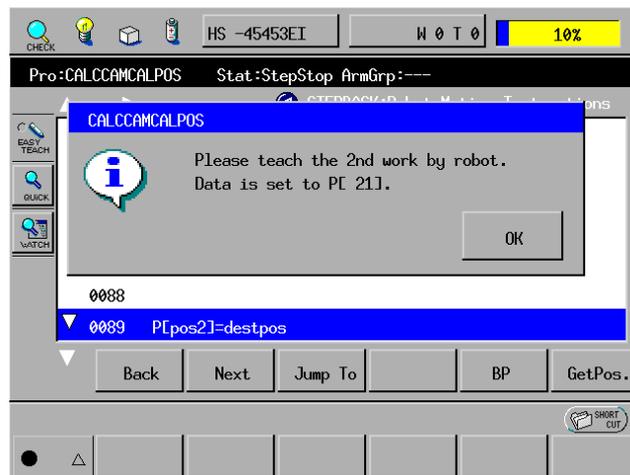


Press OK.

Switch to the manual mode and move the robot to the position of mark 1.

When the movement is completed, switch to the teach check mode again and execute the program step by step.

g) Execute the library step by step until the following message is displayed:

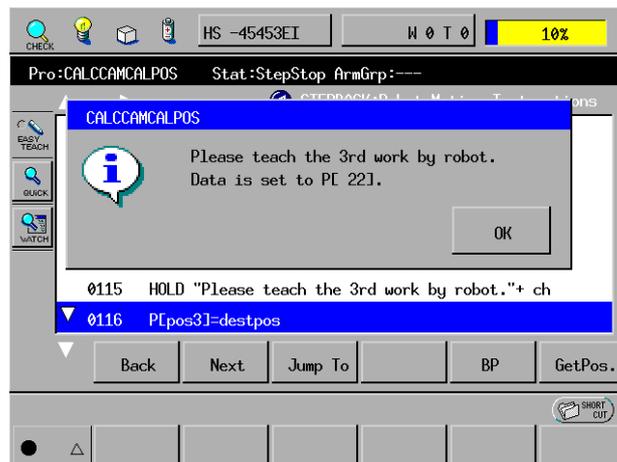


Press OK.

In the same way as for mark 1, switch to the manual mode and move the robot to the position of mark 2.

When the movement is completed, switch to the teach check mode again and execute the program step by step.

h) Execute the library step by step until the following message is displayed:



Press OK.

In the same way as for marks 1 and 2, switch to the manual mode and move the robot to the position of mark 3.

When the movement is completed, switch to the teach check mode again and execute the program step by step.

- i) Execute the library to END. The robot coordinates for the calibration marks are computed and the results are stored in the specified P-type variables.
- j) Import the robot coordinates saved by the WINCAPSIII Vision Manager, then execute calibration. This completes the calibration procedure.

2.4 Teaching of Workpiece Position when Hand I/O Input Interrupt is in Use

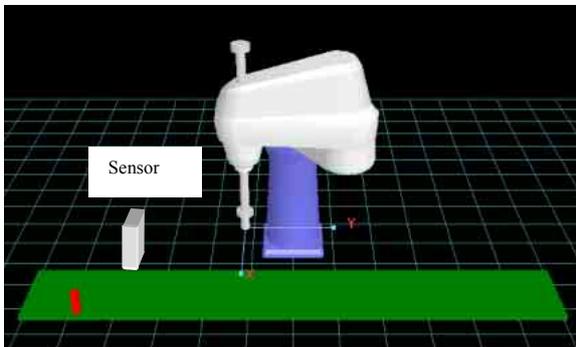
2.4.1 General

To detect workpieces using Hand I/O input interrupts, it is necessary to teach the workpiece position when it is detected. However, at the time of conveyer tracking, the workpiece detection position may be distant from the robot work position and therefore it may be impossible to move the robot to the workpiece detection position.

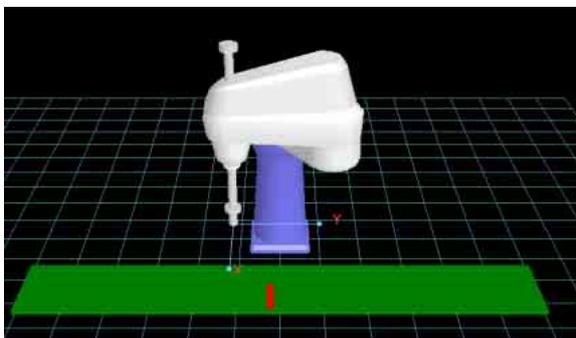
In that case, teach the workpiece detection position in the following procedure using the library CalcIOTeachPos:

Note: Before teaching the workpiece position for Hand I/O input interrupts, set conveyer tracking parameters and then execute conveyer calibration.

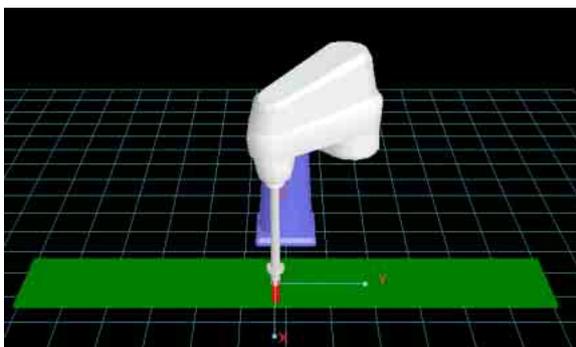
Execution of workpiece position teaching is outlined below.



- a) Prepare one workpiece for teaching.
- b) Switch to the auto mode
- c) Operate the conveyer and detect the workpiece with the sensor.



- d) Operate the conveyer, and when the robot operation range is reached, stop the conveyer.

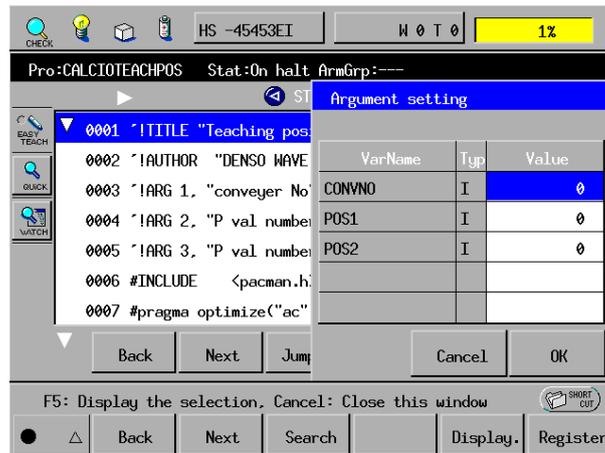


- e) Switch to the manual mode and teach the workpiece chucking position.

2.4.2 Execution Procedure for the Workpiece Position Teaching Library (CALCIOTEACHPOS)

The execution procedure for the workpiece position teaching library (CALCIOTEACHPOS) is explained below using the pendant screen.

- Switch to the auto mode and place a workpiece on the conveyer and operate the conveyer. After detecting the workpiece with the sensor, when the workpiece comes within the robot operation range, stop the conveyer.
- Switch to the manual mode and manually operate the robot to the workpiece holding position.
- Switch to the teach check mode and execute the library CALCIOTEACHPOS.

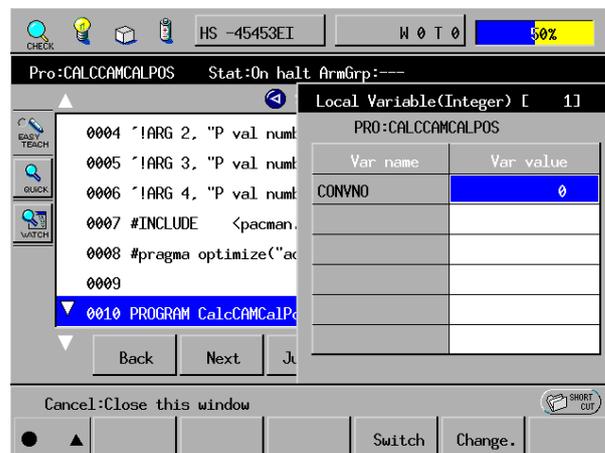


The library argument setting window is displayed. CONVNO is the conveyer number. Enter the conveyer number for calibration.

POS1 is a P-type variable number to store the workpiece position obtained by teaching.

POS2 is a P-type variable number to store the workpiece position when the workpiece is detected by the sensor (Hand I/O input interrupt). Set numbers for POS1 and POS2 in such a way that they are stored in separate P-type variables.

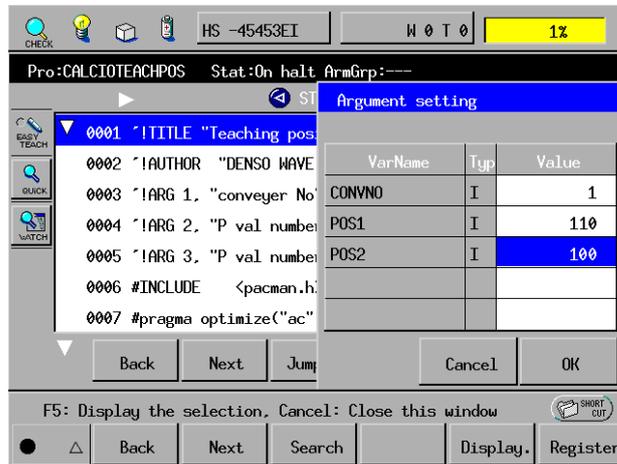
Move the cursor to CONVNO and press "Display" to display the parameter display screen.



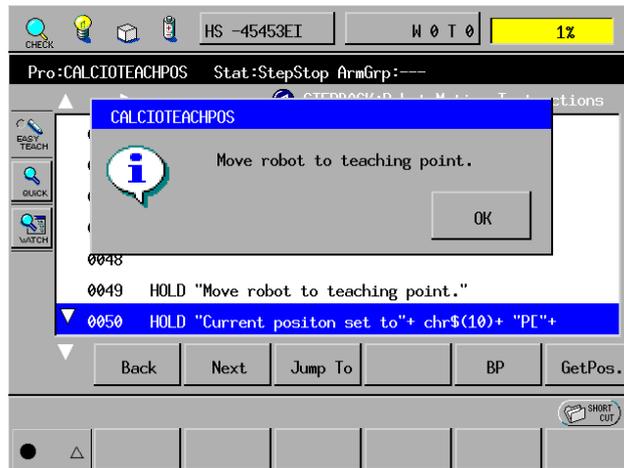
Press "Change" and enter the conveyer number.

Pressing the Cancel key after entering the numeric value returns to the original screen.

Next move the cursor to POS and press “Display”, and set the storage number by pressing “Change”. Change the setting for POS2 too. Enter all numeric values for CNVNO, POS1 and POS2. (The diagram on the left is an example of storing conveyer number 1 and workpiece teaching position in P110, and the workpiece position when it is detected in P100.)



d) Execute the library step by step. When several steps are executed, the following message is displayed:

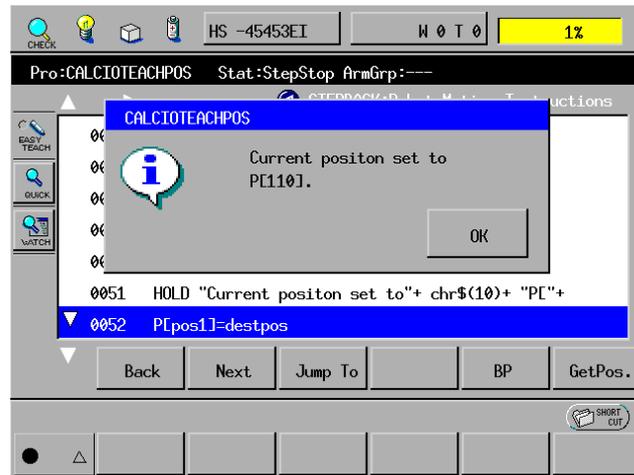


Press OK.

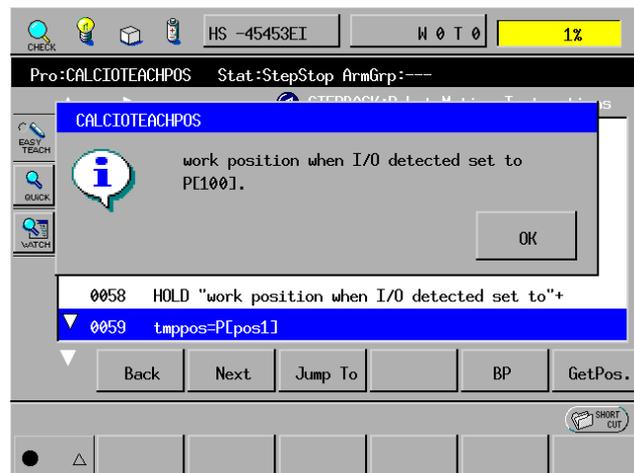
If the robot is not at the workpiece teaching position when the message is displayed, operate the robot manually.

After confirming that the robot is at the workpiece teaching position, execute the library step by step.

- e) When the library is executed several steps, the following message is displayed:
 The workpiece position is taken into the specified P-type variable.
 Press OK, and execute the library step by step.



- f) When several steps are executed, the following message is displayed:

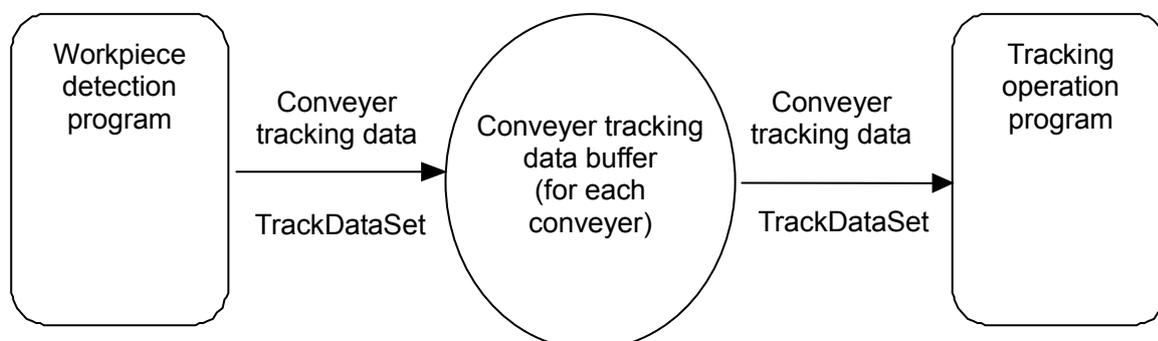


The workpiece position when it is detected is taken into the specified P-type variable
 Press OK and continue to execute the library step by step.

- g) Execute the library step by step up to the end of the program. When the program is terminated, the setting of teaching positions is completed.

Chapter 3 Conveyer Tracking Program

3.1 General



The conveyer tracking program consists of two programs, the workpiece detection program and tracking operation program, and the conveyer tracking data buffer for storing detected workpiece data. The conveyer tracking data buffer is provided on each conveyer (conveyer 1, conveyer 2). Tracking data for the conveyer specified by the workpiece detection program is stored in the buffer. The tracking operation program obtains the tracking data of the specified conveyer and executes tracking operation. Execute the workpiece detection program and the tracking operation program under a multi-task environment.

Cautions at executing the conveyer tracking motion

- (1) The conveyer tracking program must be executed in the auto mode. In the teach check mode, conveyer tracking motion is forbidden .
- (2) Conveyer tracking motion is stopped when the conveyer tracking program is completed or the mode switch is setted in the manual mode or GIVEARM command is executed.
- (3) During tracking operation, do not use the following operation commands:
 - a) Encoder value checking operation (@E is assigned to a point from which operation command pass starts)
An error 6651 “Check command time over” happens if the encoder value checking operation is used.
 - b) Operation command changing an operation pattern of a robot
A robot operation pattern cannot be changed during the tracking operation even at PTP operation. If you execute a command to change the pattern, errors 6844 “Robot posture mismatch (track)” or 667A “Destination position out of motion space 2” happen.
- (4) The following functions cannot be used during the tracking operation.
Current limiting、OffSrvLock、OffPWM
An error 6847 “Time out of waiting tracking motion” occurs if those functions are used.
- (5) If tracking operation stops (stop by motor-OFF, halt, emergent stop and stop by error), the operation returns to the normal mode from the tracking mode and an error 6841 “Interrupted track motion” occurs.

Note: If tracking operation stops by step stop, the tracking mode continues.
To start tracking operation again, execute the tracking-enabling command again.

3.2 Program Examples

3.2.1 Workpiece Detection Program

(1) When using μ Vision

```
PROGRAM Vision
DEFPOS VisResult,PosResult
PosResult=(0,0,240,0,0)

TrackDataInitialize 1           'Initialize conveyer 1 data buffer
TAKEVIS
VISSCREEN 0,0,0
VISCLS
VISBRIGHT 221
VISPLNOUT 0,1
VISWORKPLN 0
VISDEFCHAR 2,2,3
WINDMAKE R,0,510,479,0,2

DO                               'Always monitor the recognition result of the camera
CAMMODE 1,1,1                   'Specify Reset function and Field reading *1
CAMIN 1,0,0,1
    'Capture the image from a camera on conveyer 1 (Camera #1) and enable conveyer tracking
    VISPLNOUT 0
    BLOB 0,0,0,1,125,254,1,2000,0
    Label%=0
    IF VISSTATUS(0)=0 THEN      'Normal end of recognition
        IF VISSTATUS(1)<>0 THEN
            Label%=VISSTATUS(1)
            SHMODEL 0,0,0,3,80,0,Label%,0,0
            IF VISSTATUS(0)=0 THEN
                cnt%=VISGETNUM(0,0)      'Obtain the number of recognized workpieces
                FOR cnt1%=1 TO cnt%     'Convert camera coordinates to robot coordinates
                    LETX VisResult=VISGETNUM(cnt1%-1,1)
                    LETY VisResult=VISGETNUM(cnt1%-1,2)
                    CALL viTran6(0, POSX(VisResult),POSY(VisResult),PosResult)
                    'Save the recognized workpiece data into conveyer tracking data buffer
                    TrackDataSet 1,cnt%,PosResult
                NEXT
            END IF
        END IF
    END IF
LOOP UNTIL IO[220]=ON          'Repeat until the workpiece recognition completion signal comes ON
END
```

*1 Before using CAMIN, be sure to specify the reading method with CAMMODE.

(2) When using a Hand I/O input interrupt

```
PROGRAM IODetect
TrackDataInitialize 1           'Initializing conveyer 1 data buffer
DO                               'Workpiece recognition results are always monitored
    'The detected workpiece data is saved to the conveyer tracking data buffer
    'The number of detected workpieces is determined within TrackDataSet
    TrackDataSet 1,1,P100       'P100 indicates the position where the workpiece is detected
                                (computation result of CalcIOteachPos)
    Delay 200                   'Waiting for 0.2(s)m
LOOP UNTIL IO[220]=ON          'Repeated until the workpiece recognition processing termination
                                signal (such as 220) is ON
END
```

(3) When using an external visual equipment

For the Interrupt data setting of conveyer tracking parameters, set the value to 1 when using an external visual equipment. The sample program using RS-232C is shown as follows.

```
PROGRAM ExtVision

DEFINT ErrNo = 0
DEFSNG VisData(4),Masterang
DEFPOS VisResult
PosResult=(0,0,240,0,0)
TrackDataInitialize 1
TAKEVIS
FLUSH #1                                'Initializing serial communication buffer

DO
LINEINPUT #1,S[1]                        'Receiving visual data via RS-232C
cnt% = VAL(LEFT$(S1,1))                  'Obtaining the number of recognized workpieces
IF cnt% <> 0 THEN                          'When the number of recognized workpieces is except 0
  TmpBuff$ = MID$(S1,3,LEN(S1)-2)
  TmpBuff$ = TmpBuff$ + ",E"
  FOR cnt1%=1 TO cnt%
    FOR cnt2%=1 TO 3
      ii% = STRPOS(TmpBuff$,",")          'Comma "," searching
      IF ii% = 0 THEN                      'When data is abnormal
        ErrNo = 2                          'Setting data error flag
        EXIT FOR
      END IF
      VisData(cnt2%) = VAL(LEFT$(TmpBuff$,ii%-1)) 'Sampling data
      IF VisData(cnt2%) >= 999.0 THEN        'When data is abnormal'
        ErrNo = 3                          'Setting data error flag
      END IF
      TmpBuff$ = MID$(TmpBuff$,ii%+1,LEN(TmpBuff$)-ii%)
    NEXT cnt2%
    LETX VisResult = VisData(1)             'Assigns X element
    LETY VisResult = VisData(2)             'Assigns Y element
    CALL viTran6_s(POSX(VisResult),POSY(VisResult),PosResult) 'Converts to robot coordinate
    TrackDataSet 1,cnt%,PosResult          'The recognized workpiece data is
                                           saved in the conveyer tracking buffer
  NEXT cnt1%
END IF
LOOP UNTIL IO[220]=ON                    'Repeats until the end signal of recognizing workpiece outputs ON
GIVEVIS

END
```

3.2.2 Tracking Operation Program

```
PROGRAM Main
  DEFPOS PassPos

  RUN IODetect      'Executing the workpiece detection program under a multi-mask environment
  TAKEARM
  MOVE P,P10,S=20   'P10 Home Position
  DO
    IF TrackDataNum(1)>0 THEN      'If the conveyer tracking data buffer has data,
      'data is obtained from the conveyer tracking data buffer
      TrackDataGet 1,0,rest%,P11
      IF rest%>=0 THEN            'When workpiece data is obtained normally
        LETZ P11=F11              'Set a position in the direction of Z.
        LETT P11=0                'Set a posture at the tracking operation. (Note 1)
        LETF P11=1                'Set a robot figure at the tracking operation.

        VEL!=CurTrackSpd(1)

        CALL SetTrackStartArea(1,VEL!*F13,1000) 'The tracking start range is changed
                                                according to the conveyer speed
        WAITTRACKMOVE 1,P11,30 'Waiting until the workpiece subject to tracking reaches the
                                tracking start range
        PassPos=CurTrackPos( 1,P11,1) 'Obtaining the operation terminating position in the case
                                        of PTP operation for the workpiece subject to tracking

        LETZ PassPos=F11+70
        MOVE P,@P PassPos,S=100
        CALL SetTrackMove(1) 'Conveyer 1 tracking is effective
        APPROACH P,P11,@P 70,S=100
        MOVE P,P11,S=100

        DELAY 100 'Chucking a workpiece
        DEPART P,@P 30,S=100
        CALL ResetTrackMove 'Canceling conveyer 1 tracking

        APPROACH P,P21,30,S=100
        MOVE P,@E P21,S=100 'Position where the workpiece is ejected
        DEPART P,@P 70,S=100
      ENDIF
    ENDIF
  LOOP UNTIL IO[128]=ON 'Repeated while the tracking operation signal (such as 128) is ON
END
```

Note 1: (1) If the robot is a six-axis type, set a posture at the tracking operation as follows.
LETT P11=RVEC(P20) 'P20: The position and figure on the robot working
(2) When using a Hand I/O input interrupt, this command is not necessary.

3.3 Program Confirmation Procedure

- (1) Check the encoder count value.
Operate the conveyer and check the conveyer tracking parameter “Encoder current position”. Confirm that the value changes according to the conveyer operation.
- (2) Check the encoder count value when a workpiece is detected.
Switch to the auto mode. Then, operate the conveyer and check “Encoder stand. Pos. (Work detect.)”. Confirm that the value changes every time a workpiece is detected.
- (3) Check the workpiece detection program.
 - (3-1) Execute the workpiece detection program and let workpieces flow on the conveyer. Count the number of workpieces visually at this time.
 - (3-2) Stop the conveyer.
 - (3-3) Execute the following program and check the number of data items within the conveyer tracking data buffer:

```
PROGRAM CHKBUFNUM
I50=TrackDataNum(1) 'For conveyer 1, I50: Assign an unused I-type variable.
END
```

The number of data items within the conveyer tracking data buffer is set in “I50”. Check the value in “I50” and confirm that it matches the number of detected workpieces that was confirmed visually.

- (4) Set operation inhibiting areas.
The conveyer tracking operation may sometimes travel to an unexpected position because of a programming mistake. Use the area function to set operation inhibited area.
- (5) Check the conveyer tracking operation.
Move the robot near the center of the conveyer and execute the following program. Confirm that the robot follows the conveyer.
When conveyer tracking speed and conveyer speed deviate from each other.
Check the encoder lead amount of conveyer tracking parameter.
When conveyer tracking direction and conveyer moving direction deviate from each other.
Execute conveyer calibration again.

```
PROGRAM CHKCONVEYERTRACK
TakeArm
CALL SetTrackMove(1)
DELAY 1000 'Following conveyer 1 for 1(s)
CALL ResetTrackMove
END
```

If the robot gets out of the operation range while following the conveyer, an error occurs.

If the robot moves to a position outside the tracking range that was set in the conveyer tracking parameters “Conveyer Track Upper Pos. (mm)” and “Conveyer Track Lower Pos. (mm)”, an error occurs.

(6) Check the tracking operation using the workpiece to be detected.

Check the tracking operation using one detected workpiece.

Start with low conveyer speed and robot speed and increase each speed gradually.

Use the following program to confirm tracking of the workpiece to be detected.

```
PROGRAM CHKWORKTRACK
```

```
DEFPOS PassPos
```

```
RUN IODetect      'Executing the workpiece detection program under a multi-mask  
environment
```

```
TAKEARM
```

```
MOVE P,P10,S=20      'P10 Home Position
```

```
DO
```

```
IF TrackDataNum(1)>0 THEN      'If the conveyer tracking data buffer has data,  
'data is obtained from the conveyer tracking data buffer
```

```
TrackDataGet 1,0,rest%,P11
```

```
IF rest%>=0 THEN      'When workpiece data is obtained normally
```

```
LETZ P11=F11      'Set a position in the direction of Z (50 mm above the conveyer).
```

```
LETT P11=0      'Set a posture at the tracking operation. (Note 1)
```

```
LETF P11=1      'Set a robot figure at the tracking operation.
```

```
'The tracking starts when the workpiece reaches the conveyer position 0
```

```
CALL SetTrackStartArea(1,0,1000)
```

```
WAITTRACKMOVE 1,P11,30
```

```
PassPos=CurTrackPos( 1,P11,1) 'Obtaining the operation terminating position in the case  
of PTP operation for the workpiece subject to tracking
```

```
LETZ PassPos=F11
```

```
MOVE P,PassPos
```

```
CALL SetTrackMove(1)      'Conveyer 1 tracking is effective
```

```
MOVE P,P11
```

```
DELAY 1000      'Following the workpiece for 1(s)
```

```
CALL ResetTrackMove      'Canceling conveyer 1 tracking
```

```
MOVE P, P21      'Position where the workpiece is ejected
```

```
ENDIF
```

```
ENDIF
```

```
LOOP
```

```
END
```

Note 1: (1) If the robot is a six-axis type, set a posture at the tracking operation as follows.

LETT P11=RVEC(P20) 'P20: The position and figure on the robot working

(2) When using a Hand I/O input interrupt, this command is not necessary.

(7) Execute the tracking operation program.

Start with low conveyer speed and robot speed, and increase each speed gradually.

Start with one workpiece and increase the number of workpieces gradually.

Reduce the workpiece interval gradually.

3.4 Conveyor Tracking Accuracy Checking Procedures

3.4.1 Encoder Installation Check

Check encoder and conveyor rotation for any slip or looseness between them.
Correct any slip or looseness if found.

3.4.2 When Using Visual Equipment (External visual equipment such as μ Vision)

(1) Calibration and teaching operation checking procedure

- (1-1) Place a workpiece on a conveyor at a position within a visual recognition area, and execute the visual recognition checking program (CheckVision). For the visual recognition checking program (CheckVision), refer to the program example 1.
- (1-2) Move the conveyor until the workpiece enters into both the robot movable area and the tracking operation area, and stop the conveyor.
- (1-3) Set the robot operation speed to low (SP10 or less) and execute the tracking checking program (CheckTracking). Set the timer (I30 in the program) to a value 200 (0.2(s)) or more after the chuck position moving command is executed. For the tracking checking program (CheckTracking), refer to the program example 2.
- (1-4) If the workpiece chucking position is correct, go to the procedure (2); if any deviation is found in the chucking position, go to the procedure (1-5).
- (1-5) Check if the conveyor calibration result is correct by referring to “(5) Check the conveyor tracking operation” in “3.3 Program confirmation procedures”. If any deviation is found in the conveyor calibration, execute the conveyor calibration again by referring to “2.2 Conveyor Calibration”.
- (1-6) Check if the visual recognition checking program (CheckVision) is correct. As shown in the program example 1, assign the visual recognition checking result to the P-type variable and check the value of the P-type variable (P22).
- (1-7) Check if the tracking operation program (CheckTracking) is correct. As shown in the program example 2, assign the tracking operation position to the P-type variable (P11) and check the value of the P-type variable.
- (1-8) If no problem was found in the procedures (1-6) and (1-7), execute the camera calibration again by referring to the “2.3 Camera Calibration”. After the calibration, start the procedures again from (1-1).
- (1-9) If the deviation in the workpiece chucking position cannot be corrected, correct the position to the conveyor operation direction as shown in the position correcting program (PosAdjust) in the program example 4.

(2) Robot speed-increase checking operation

- (2-1) Execute the procedures (1-1)-(1-3) while increasing the robot speed gradually.
- (2-2) If the workpiece chucking position deviates from normal position as the speed increases, increase the timer value (I30 in CheckTracking) after the chuck position moving command is executed.

(3) Workpiece recognition process check

- (3-1) Set the conveyer tracking parameter "Workpiece detection position accuracy" to 2 mm (the same value as operation accuracy).
- (3-2) Place a workpiece on a conveyer at a position within a visual recognition area, and execute the workpiece detection program (Vision). For the workpiece detection program (Vision), refer to "3.2.1 Workpiece detection program".
- (3-3) Move the conveyer until the workpiece moves out of the visual recognition area.
- (3-4) Terminate the workpiece detection program (Vision), and execute a program (CHKBUFNUM) for checking the number of data in the conveyer tracking data buffer. Verify the number of data (I50) is 1. For the data count program (CHKBUFNUM), refer to "(3) Check the workpiece detection program" of "3.3 Program Confirmation Procedure".
- (3-5) Repeat the procedures (3-2)-(3-4) several times. If the number of data is 1 for all the checks, go to the procedure (3-12); if not, go to (3-6).
- (3-6) Place a workpiece on a conveyer at a position within a visual recognition area, and execute the workpiece detection program (Vision) without moving the conveyer.
- (3-7) After a while, the workpiece detection program (Vision) terminates. Verify that the number of data (I50 in CHKBUFNUM) is 1 by executing the program (CHKBUFNUM) for checking the number of data in the conveyer tracking data buffer.
- (3-8) Verify the number of data is 1 by repeating the procedures (3-6)-(3-7) several times. If the number of data is 1, go to (3-9); otherwise, the image processing program has a problem. Check the image processing program.
- (3-9) For external visual equipment, check that the hand I/O interrupt signal is the shutter-on signal (image-capturing start signal). If any delay is found between the shutter-on signal and image-pickup, set the conveyer tracking parameter "Interrupt delay correction value" to the delay time.
- (3-10) Execute the procedures (3-2)-(3-5) again.
- (3-11) Restore the conveyer tracking parameter "Workpiece detection position accuracy" to an original value.

(4) Conveyer speed-increase checking operation

- (4-1) Set the timer (I30), in the tracking operation program (Main), to a value adjusted in (2) after the chuck position moving command is executed. For the tracking operation program (Main), refer to "3.2.2 Tracking operation program". (The example of workpiece detection program is named as IODetect (RUN IODetect). Change the name to "Vision (RUN Vision)).
- (4-2) Set both the robot speed and conveyer speed to low, and execute the tracking operation program (Main).

- (4-3) Placing a workpiece on a conveyer, move the conveyer to check the tracking operation and the workpiece chucking position. If something is wrong with the tracking operation, check if the tracking operation program is correct.
- (4-4) Execute the procedure (4-3), while keeping the conveyer speed at low and increasing robot speed. If the workpiece chucking position deviates depending on the robot speed, adjust the timer (I30 in Main) after the chuck position moving command is executed.
- (4-5) Increasing the conveyer speed gradually, check the workpiece chucking position. If the workpiece chucking position deviates from normal position as the conveyer speed increases, adjust the conveyer tracking parameter "Tracking position adjustment value".
However, a limitation to the chuck position accuracy is (conveyer speed (mm/s) × 0.008) (mm) + a value equivalent to the camera pixels.

3.4.3 When Position is Recognized by a Sensor (Hand I/O Interrupt)

(1) Calibration and teaching operation checking procedures

- (1-0) Execute the tracking data initialize program (DataInitialize).
- (1-1) Place a workpiece upstream from a sensor on a conveyer. Change the operation mode to automatic and start the conveyer at low speed. Detect the workpiece position using a sensor while the conveyer is at operation.
- (1-2) Move the conveyer until the workpiece enters into both the robot movable area and the tracking operation area, and stop the conveyer.
- (1-3) Execute the hand I/O interrupt checking program (CheckIODetect). For the hand I/O interrupt checking program (CheckIODetect), refer to the program example 3.
- (1-4) Setting the robot operation speed to low (SP10 or less), execute the tracking checking program (CheckTracking). Set the timer (I30 in CheckTracking) to a value 200 (0.2(s)) or more after the chuck position moving command is executed. For the tracking checking program (CheckTracking), refer to the program example 2.
- (1-5) If the workpiece chucking position is correct, go to the procedures (2); if any deviation is found in the chucking position, go to the procedure (1-6).
- (1-6) Check if the conveyer calibration is correct by referring to "(5) Check the conveyer tracking operation" in "3.3 Program confirmation procedures". If any deviation is found in the conveyer calibration, execute the conveyer calibration again.
- (1-7) Check the hand I/O interrupt checking program (CheckIODetect) for any abnormality. As shown in the program example 3, assign the workpiece detection position result to the P-type variable, and check the value of the P-type variable.
- (1-8) Check the tracking operation program (CheckTracking) for any abnormality. As shown in the program example 2, assign the tracking operation position to the P-type variable, and check the value of the P-type variable.
- (1-9) Perform again the workpiece position teaching when the hand I/O interrupt is executed. Refer to "2.4 Teaching of Workpiece Position when Hand I/O Input Interrupt is in Use". Start the procedures from (1-1) again.

(1-10) If the deviation in the workpiece chucking position cannot be corrected, correct the position to the conveyer operation direction as shown in the position correcting program (PosAdjust) as shown in the program example 3.

(2) Robot speed-increase checking operation

- (2-1) Execute the procedures (1-1)-(1-4) while increasing the robot speed gradually.
- (2-2) If the workpiece chucking position deviates from normal position as the speed increases, increase the timer value (I30 in CheckTracking) after the chuck position moving command is executed.

(3) Workpiece recognition process check

- (3-1) Set the conveyer tracking parameter "Workpiece detection position accuracy" to 2 mm (the same value as operation accuracy).
- (3-2) Place a workpiece upstream from a sensor on a conveyer. Execute the workpiece detection program (IODetect) and detect the workpiece using the sensor. For the workpiece detection program (IODetect), refer to "(2) When using an external visual equipment" in "3.2.1 Workpiece detection program".
- (3-3) Verify that the number of data is 1 by executing a program (CHKBUFNUM) for checking the number of data in the conveyer tracking data buffer. For the data count program (CHKBUFNUM), refer to "(3) Check the workpiece detection program" of "3.3 Program Confirmation Procedure".
- (3-4) Execute the procedures (3-2) and (3-3) several times while increasing the conveyer speed gradually. There is no problem if the number of data is 1. Restore the conveyer tracking parameter "Workpiece detection position accuracy" to an original value and go to the procedures (4); if not, go to the procedure (3-5).
- (3-5) Check if the sensor signal has been turned on more than once for the same workpiece. If so, adjust the layout of the sensor or others. If chattering is found to happen in the sensor signal, set the conveyer tracking parameter "Workpiece detection position accuracy" to the following value or lower to neglect the chattering zone.
"Workpiece detection position accuracy" > Conveyer moving distance in the chattering zone
- (3-6) If the number of detected data increases as the conveyer speed increases, check the conveyer tracking parameter "Interrupt timing". Whenever the "Interrupt timing" is changed, teach again the workpiece position when an I/O interrupt occurs. Refer to "2.4 Teaching of Workpiece Position when Hand I/O Input Interrupt is in Use".
- (3-7) Execute the procedures (3-2)-(3-4) again.

(4) Conveyer speed-increase checking operation

- (4-1) Set the timer (I30), in the tracking operation program (Main), to a value adjusted in (2) after the chuck position moving command is executed. For the tracking operation program (Main), refer to “3.2.2 Tracking operation program”.
- (4-2) Set both robot speed and conveyer speed to low, and execute the tracking operation program (Main).
- (4-3) Placing a workpiece on a conveyer, move the conveyer to check the tracking operation and the workpiece chucking position. If something is wrong with the tracking operation, check if the tracking operation program is correct.
- (4-4) Execute the procedure (4-3) keeping the conveyer speed at low while increasing robot speed. If the workpiece chucking position deviates depending on the robot speed, adjust the timer (I30 in Main) after the chuck position moving command is executed.
- (4-5) Increasing the conveyer speed gradually, check the workpiece chucking position. If the workpiece chucking position deviates from normal position as the conveyer speed increases, adjust the conveyer tracking parameter “Tracking position adjustment value”. However, a limitation to the chuck position accuracy is $(\text{conveyer speed (mm/s)} \times 0.008)$ (mm).

3.5 Tact time Decreasing Operation

- (1) Adjust the conveyer tracking parameter “Tracking acceleration rate”. Refer to “14. Tracking accel rate” in “2.1.2 Detailed explanation of parameters”.
- (2) A wait is produced when the conveyer tracking mode switching operation (CALL SetTrackMove or CALL ResetTrackMove) is executed. Decrease the wait using the pass operation or NEXT operation.
- (3) As shown in the program example 5 (PassMove), if you operate the conveyer based on a prediction using CurTrackPos command, the operation command (APPROACH P,P11,@P 70 in PassMove) becomes a moving command for very small distance and the pass operation becomes an ending operation. Use the position corrected to the conveyer predicted operation position as a point to be gone through.
- (4) To decrease a tact time by decreasing the timer (I30 in Main) after the chuck position moving command is executed, adjust the conveyer tracking parameter “Tracking position adjustment value”. The maximum value of the conveyer tracking parameter “Tracking position adjustment value” is 50. If the chuck position deviates upper stream even when the parameter is set to the maximum value, shift the position obtained with TrackDataSet toward conveyer operation direction as shown in the program example 4 (PosAdjust).

3.6 Program Examples

(1) Program example 1 Visual recognition checking program

```
PROGRAM CheckVision
DEFINT ErrNo = 0
DEFSNG VisData(4),Masterang
DEFPOS VisResult
PosResult=(0,0,240,0,0)
TrackDataInitialize 1
TAKEVIS
FLUSH #1                                'Initializing serial line
LINEINPUT #1,S[1]                        'Receiving visual data via RS232C
cnt% = VAL(LEFT$(S1,1))                  'Obtaining the number of recognized workpieces
IF cnt% <> 0 THEN                          'When the number of recognized workpiece is excepto
  TmpBuff$ = MID$(S1,3,LEN(S1)-2)
  TmpBuff$ = TmpBuff$ + ",E"
  FOR cnt1%=1 TO cnt%
    FOR cnt2%=1 TO 3
      ii% = STRPOS(TmpBuff$,",")         ', Comma searching
      IF ii% = 0 THEN                     'When data is abnormal
        ErrNo = 2                         'Setting data error flag
        EXIT FOR
      END IF
      VisData(cnt2%) = VAL(LEFT$(TmpBuff$,ii%-1)) 'Sampling data
      IF VisData(cnt2%) >= 999.0 THEN     'When data is abnormal
        ErrNo = 3                         'Setting data error flag
        EXIT FOR
      END IF
      TmpBuff$ = MID$(TmpBuff$,ii%+1,LEN(TmpBuff$)-ii%)
    NEXT cnt2%
    LETX VisResult = VisData(1)           'Assigns X element
    LETY VisResult = VisData(2)           'Assigns Y element
    CALL viTran6_s(POSX(VisResult),POSY(VisResult),PosResult)'Converts to robot coordinate
    TrackDataSet 1,cnt%,PosResult         'Store recognized workpiece data into conveyer
                                          tracking data buffer.
  NEXT cnt1%
END IF
GIVEVIS
END
```

(2) Program example 2 Visual recognition checking program

```
PROGRAM CheckTracking
DEFPOS PassPos
TAKEARM
MOVE P,P10,S=20          'P10 Home Position

IF TrackDataNum(1)>0 THEN 'If data exists in conveyer tracking data buffer,
  TrackDataGet 1,0,rest%,P11 'obtain data from conveyer tracking data buffer.
  IF rest%>=0 THEN          'When workpiece data is obtained normally.
    LETZ P11=F11           'Set a position in the direction of Z.
    LETT P11=0             'Set a posture at the tracking operation. (Note 1)
    LETF P11=1             'Set a robot figure at the tracking operation.
    WAITTRACKMOVE 1,P11,30 'Wait until target workpiece enters into tracking start area.
    PassPos=CurTrackPos( 1,P11,1) 'Obtain operation termination position when PTP motion
                                   is carried out to a tracking-target workpiece position.

    LETZ PassPos=F11+70
    MOVE P,@P PassPos
    CALL SetTrackMove(1)    'Conveyer 1 tracking is enabled.
    APPROACH P,P11,@P 70
    MOVE P,P11
    DELAY I30
    CALL HandChuck         'Workpiece chuck
    DEPART P,@P 30
    CALL ResetTrackMove    'Conveyer 1 tracking is released.
    APPROACH P,P21,30
    MOVE P,@E P21         'P21: Workpiece output position
  ENDIF
ENDIF
ENDIF
END
```

Note 1: (1) If the robot is a six-axis type, set a posture at the tracking operation as follows.

LETT P11=RVEC(P20) 'P20: The position and figure on the robot working

(2) When using a Hand I/O input interrupt, this command is not necessary.

(3) Program example 3 Hand I/O interrupt checking program

```
PROGRAM CheckIODetect
TrackDataSet 1,1,P100    'P100 is a workpiece detecting position (Calculation result
                          of CalcIOteachPos)
END
```

```
PROGRAM DataInitalize
TrackDataInitialize 1    'Initialize conveyer 1 data buffer.
END
```

(4) Program example 4 Position correction program

```
PROGRAM PosAdjust
DEFVEC CnvVec
IF TrackDataNum(1)>0 THEN      'If data exists in conveyer tracking data buffer,
    'obtain data from conveyer tracking buffer.
    TrackDataGet 1,0,rest%,P11
    IF rest%>=0 THEN          'When workpiece data is obtained normally
        VEL:=CurTrackSpd(1)  'Obtain conveyer speed (mm/s).
        CALL CalcConvVec(1,CnvVec) 'Calculate conveyer 1 vector in operation direction.
        'F99: Correction coefficient for conveyer speed
        LETX P11=POSX(P11)+POSX(CnvVec)*VEL!*F99
        LETY P11=POSY(P11)+POSY(CnvVec)*VEL!*F99
        LETZ P11=F11          'Set a position in the direction of Z.
        LETT P11=0           'Set a posture at the tracking operation. (Note 1)
        LETF P11=1          'Set a robot figure at the tracking operation.
    ENDIF
ENDIF
END
```

(5) Program example 5 Pass operation program

```
PROGRAM PassMove
DEFPOS PassPos
IF TrackDataNum(1)>0 THEN      'If data exists in conveyer tracking data buffer,
    TrackDataGet 1,0,rest%,P11 'obtain data from conveyer tracking buffer.
    IF rest%>=0 THEN          'When workpiece data is obtained normally
        LETZ P11=F11          'Set a position in the direction of Z.
        LETT P11=0           'Set a posture at the tracking operation. (Note 1)
        LETF P11=1          'Set a robot figure at the tracking operation.
        WAITTRACKMOVE 1,P11,30 'Wait until target workpiece enters into tracking start area.
        PassPos=CurTrackPos( 1,P11,1) 'Obtain operation termination position when PTP
                                         motion is carried out to a tracking-target workpiece
                                         position.
        LETZ PassPos=F11+70
        MOVE P,@P PassPos
        CALL SetTrackMove(1)    'Conveyer 1 tracking is enabled.
        APPROACH P,P11,@P 70
        MOVE P,P11
        DELAY I30
        CALL HandChuck          'Workpiece chuck
        DEPART P,@P 30
        CALL ResetTrackMove    'Conveyer 1 tracking is released.
        APPROACH P,P21,30
        MOVE P,@E P21          'P21: Workpiece output position
    ENDIF
ENDIF
END
```

Note 1: (1) If the robot is a six-axis type, set a posture at the tracking operation as follows.

LETTR P11=RVEC(P20) 'P20: The position and figure on the robot working

(2) When using a Hand I/O input interrupt, this command is not necessary.

3.7 List of Commands Related to the Conveyer Tracking Function

TRACKDATAINITIALIZE (statement)

Function

Initializes data within the conveyer tracking data buffer

Format

TRACKDATAINITIALIZE <Initialization mode>

Explanation

This command initializes data within the conveyer tracking data buffer. Specify the initialization content with <initialization mode>.

Initialization mode value

0: Initializes conveyer-1 and conveyer-2 data.

1: Initializes only conveyer-1 data.

2: Initializes only conveyer-2 data.

Related item

TRACKDATASET

Example

```
PROGRAM PRO1  
TRACKDATAINITIALIZE 0 'Initializing conveyer-1 and conveyer-2 data'  
END
```

Notes

Execution of TRACKDATAINITIALIZE erases the data saved with TRACKDATASET.

TRACKDATASET (statement)

Function

Saves data in the conveyer tracking buffer.

Format

TRACKDATASET <conveyer_number>, <number_of_recognized_workpieces>, <recognized_workpiece_position>

Explanation

This command saves P-type data given in <recognized_workpiece_position> into the tracking buffer for the conveyer specified in <conveyer_number>. It sets the number of workpieces recognized simultaneously in <number_of_recognized_workpieces>. For instance, if three workpieces have been recognized by a camera, it sets 3 in <number_of_recognized_workpieces>.

Related item

TRACKDATAGET, TRACKDATAINITIALIZE

Example

```
PROGRAM PRO1
:
:'If three workpieces (pos1, pos2, pos3) are recognized simultaneously by a camera on
conveyer 1
TRACKDATASET 1,3,pos1      'The first recognized workpiece position is saved in the
                           buffer
TRACKDATASET 1,3,pos2      'The second recognized workpiece position is saved in
                           the buffer
TRACKDATASET 1,3,pos3      'The third recognized workpiece position is saved in the
                           buffer
END
```

Notes:

Be sure to execute this command as many times as the number set for the number of workpieces to be recognized.

Otherwise, the tracking operation position may be abnormal.

Data in <recognized_workpiece_position> is compared with the previously recognized workpiece position. If the system determines that the workpiece is at the same position, the data in <recognized_workpiece_position> is not saved but discarded.

Data saved in the tracking buffer with TRACKDATASET can be obtained from the buffer using TRACKDATAGET. The maximum number of data items that can be retained (the number of data items that have not been read out with TRACKDATAGET) is 1000 (100 in the controller system Ver. 2.73 or earlier).

When the power for the controller is turned off, data is erased.

Retained data can be erased with TRACKDATAINITIALIZE.

TRACKDATAGET (statement)

Function

Obtains data from the conveyer tracking buffer

Format

TRACKDATAGET <conveyer_number>, <data_number>,
<number_of_remaining_data_items>, <recognized_workpiece_position>

Explanation

This command obtains the workpiece position having the number specified in <data_number> from the tracking buffer of the conveyer specified in <conveyer_number> and then sets the position in <recognized_workpiece_position>. The number of data items remaining in the tracking buffer is set in <number_of_remaining_data_items>. If there is no data and an attempt to obtain data fails, the value of <number_of_remaining_data_items> is "-1". <data_number> indicates the sequence number saved with TRACKDATASET. If "0" is specified, the data item that is the oldest of all retained data items (data items that have not been obtained with TRACKDATAGET) is applicable.

Related item

TRACKDATASET, TRACKDATAINITIALIZE

Example

```
PROGRAM PRO1
DEFINT REST
DEFPOS POS1
TAKEARM
MOVE P,P1
TRACKDATAGET 1,0,REST,POS1           'Obtaining the 0th workpiece recognition
                                     position on conveyer 1
IF(REST>=0) THEN                     'If there is obtained data
  CALL SETTRACKMOVE(1)               'Starting conveyer 1 tracking
  MOVE P.,POS1                       'Tracking operation at the workpiece (POS1) position on
                                     the conveyer
  CALL RESETTRACKMOVE                'Terminating the conveyer 1 tracking operation
ENDIF
END
```

Notes

If no data is set by TRACKDATASET, data cannot be obtained.
If the value of "number_of_remaining_data_items" is 0, data cannot be obtained.
Data obtained by TRACKDATAGET is erased from the buffer.

TRACKDATAINFO (statement)

Function

Obtains information within the conveyer tracking buffer

Format

TRACKDATAINFO <conveyer_number>, <data_number>,
<encoder_value_at_recognition>, <availability>,
<workpiece_position_at_recognition>

Explanation

This command obtains data specified in <data_number> from the tracking buffer of the conveyer specified in <conveyer_number>. The workpiece position is set in <workpiece_position_at_recognition>, and the encoder count value at workpiece recognition is set in <encoder_value_at_recognition>. If there is no data corresponding to the specified number, the value of <availability> is "-1". <data_number> indicates the sequence number saved with TRACKDATASET. If "0" is specified, the data item that is the oldest of all retained data items (data items that have not been obtained with TRACKDATASET) is applicable.

Related item

CALCWORKPOS, TRACKDATANUM

Example

```
PROGRAM PRO1
DEFINT LI1,ENC
DEFPOS POS1
TRACKDATAINFO 2,0,ENC,LI1,POS1      'The 0th data on conveyer 2 is obtained
                                     The availability is obtained in LI1
END
```

Notes

If there is no data corresponding to the specified number, the value of "availability" is "-1". When data acquisition is completed, the value of "availability" is "0". TRACKDATAINFO is only used to obtain information and the information is not erased from the buffer.

TRACKDATANUM (statement)

Function

Obtains the number of data items retained with TRACKDATASET

Format

<number_of_saved_data_items>=TRACKDATANUM(<conveyer_number>)

Explanation

This command obtains the number of data items retained in the tracking buffer of the conveyer specified in <conveyer_number>.

Example

```
PROGRAM PRO1
DEFINT LI1
LI1=TRACKDATANUM (1)                'The number of data items retained for conveyer 1 is set
                                     in LI1
END
```

CURTRACKPOS (statement)

Function

Obtains the position of the workpiece subject to tracking as a P type

Format

<tracking-target_workpiece_position>=CURTRACKPOS(<conveyor_number>, <P_TYPE_POSITION_DATA>, <mode>)

Explanation

This command sets the position of the workpiece subject to tracking in the conveyor specified in **<conveyor_number>** in **<P_TYPE_POSITION_DATA>**. The workpiece subject to tracking is the one for which data is obtained with TRACKDATAGET before this command is executed. Information about the position of the workpiece subject to tracking shown below can be obtained from **<mode>**.

0: Obtains the current position of the target workpiece.

1: Obtains the position of the workpiece at the termination of the operation when the robot is moved to the target workpiece position by PTP operation.

2: Obtains the position of the workpiece at the termination of the operation when the robot is moved to the target workpiece position by CP operation.

Related item

TRACKDATAGET, TRACKDATASET

Example

```
PROGRAM PRO1
DEFINT REST
DEFPOS POS1,POS2
TAKEARM
```

```
MOVE P,P1
```

```
TRACKDATAGET 1,0,REST,POS1
```

```
POS2= CURTRACKPOS(1,POS1,1)
```

```
MOVE P POS2
```

```
CALL SETTRACKMOVE(1)
```

```
MOVE P,POS1
```

```
CALL RESETTRACKMOVE
```

```
END
```

'Obtaining the 0th workpiece data from conveyor 1

'Setting the workpiece position at termination of PTP operation in POS2

'Operation to POS2

'Starting conveyor 1 tracking operation

'Tracking operation for the workpiece in POS1

'Terminating conveyor 1 tracking operation

Notes

If a tracking-target workpiece position is out of the robot movable area when specifying the mode, "Destination position out of motion space" error occurs.

CURTRACKSPD (statement)

Function

Obtains the speed of the conveyer specified in <conveyer_number>.

Format

<conveyer_speed>=CURTRACKSPD(<conveyer_number>)

Explanation

This command obtains the speed of the conveyer specified in <conveyer_number> in units of [mm/sec].

Related item

Example

```
PROGRAM PRO1
DEFSNG LF1
LF1=CURTRACKSPD(1) 'Obtaining the speed of conveyer 1 in LF1
END
```

WAITTRACKMOVE (statement)

Function

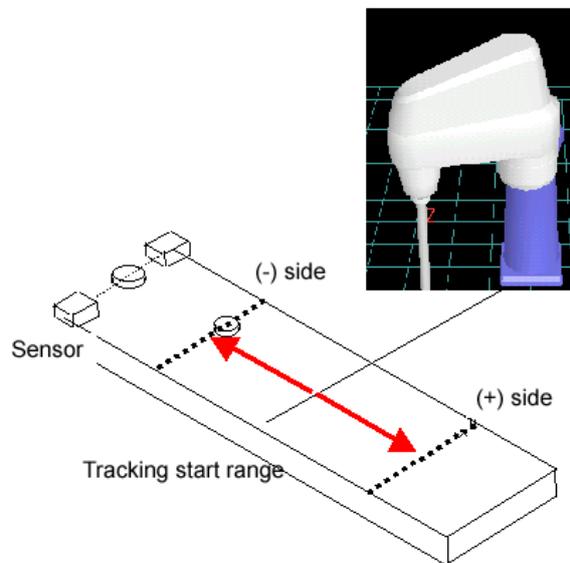
Waits for the position of the workpiece subject to tracking to enter the tracking start range

Format

WAITTRACKMOVE <conveyer_number>,<recognized_workpiece_position>,<timeout>

Explanation

This command waits for the workpiece specified in <recognized_workpiece_position> on the conveyer specified in <conveyer_number> to enter the tracking start range. <timeout> indicates the maximum wait time in seconds. If the workpiece does not fall into the tracking start range within the specified time-out period, a timeout error occurs.



Related item

TRACKDATAGET

Example

```

PROGRAM PRO1
DEFINT REST
DEFPOS POS1
TAKEARM
MOVE P,P1
TRACKDATAGET 1,0,REST,POS1           'Obtaining the 0th workpiece position
                                      from the conveyer 1 buffer
WAITTRACKMOVE 1,POS1,60             'Waiting for the workpiece to enter the tracking
                                      range for a maximum of 60 seconds
CALL SETTRACKMOVE(1)                'Starting conveyer 1 tracking
MOVE P.,POS1                         'Tracking operation to the work (POS1) position on theconveyer
CALL RESETTRACKMOVE                  'Terminating tracking operation to conveyer 1
END
    
```

Notes

The tracking start range is a conveyer tracking parameter. Set this parameter either with a pendant or the tracking start range setting library.

While executing WAITTRACKMOVE, never allow WAITTRACKMOVE to be executed in other programs.

When a target workpiece is downstream from the tracking start area, an error 684B "Out of track start area" occurs.

The suspended stop by Halt is calculated as wait time.

CALCWORKPOS (statement)

Function

Obtains the current position of the specified workpiece.

Format

<current_workpiece_position>=CALCWORKPOS (<conveyor_number>, <workpiece_position_at_recognition>, <encode_value_at_recognition>)

Explanation

This command calculates the current position of the workpiece on the conveyor specified in **<conveyor_number>**. Information (**<workpiece_position_at_recognition>** and **<encode_value_at_recognition>**) of the workpiece subject to tracking is obtained with TRACKDATAINFO.

Related item

TRACKDATAINFO

Example

```
PROGRAM PRO1
DEFINT LP1,LI1,ENC,PNUM
DEFPOS POS1,POS2
```

```
PNUM=TRACKDATANUM(2)
```

'Obtaining the number of retained data items in conveyor 2

```
FOR LP1=0 TO PNUM-1
```

'Referencing data items as many as PNUM

```
TRACKDATAINFO (2,LP1,ENC,LI1,POS1)
```

'Obtaining conveyor 2 data sequentially starting with the 0th

```
IF(LI1=0) THEN
```

```
    POS2=CALCWORKPOS(2,POS1,ENC)
```

'Setting the current position of the workpiece obtained with TRACKDATAINFO in POS2

```
    :
```

```
    :
```

```
    ENDIF
```

```
    NEXT LP1
```

```
END
```

CURTRACKPOSEX (statement)

Function

Obtains a tracking-target workpiece position in the P-type form.

Format

<tracking-target_workpiece_position>=CURTRACKPOSEX (<conveyer_number>, <P-TYPE_POSITION_DATA>, <mode>, <error_information>)

Explanation

Sets a tracking-target workpiece position on a conveyer, specified by **<conveyer_number>**, into a **<P-TYPE_POSITION_DATA>**. The tracking-target workpiece has been obtained with TRACKDATAGET before execution of this command. The following tracking-target workpiece position information can be obtained according to **<mode>**.

0: Obtains a current position of a target workpiece.

1: Obtains a target workpiece position at operation termination when the robot has been moved to the target workpiece position using the PTP operation.

2: Obtains a target workpiece position at operation termination when the robot has been moved to the target workpiece position using the CP operation.

<error_information> becomes 0 after a tracking-target workpiece position has been calculated, and becomes 1 when a tracking-target workpiece position is out of the robot movable area.

Related item

CURTRACKPOS

Example

```
PROGRAM PRO1
DEFINT REST,ERINF
DEFPOS POS1,POS2
TAKEARM
```

```
MOVE P,P1
TRACKDATAGET 1,0,REST,POS1
```

'Obtain the zeroth workpiece data from conveyer 1 data.

```
POS2= CURTRACKPOSEX(1,POS1,1,ERINF)
```

'Assign to POS2 a workpiece position after PTP operation has been terminated.

```
IF ERINF=0 THEN
```

```
    MOVE P POS2
```

'Move to POS2.

```
    CALL SETTRACKMOVE(1)
```

'Start conveyer 1 tracking.

```
    MOVE P.,POS1
```

'Track a workpiece at POS1.

```
    CALL RESETTRACKMOVE
```

'Terminate conveyer 1 tracking.

```
ENDIF
```

```
END
```

WAITTRACKMOVEEX (statement)

Function

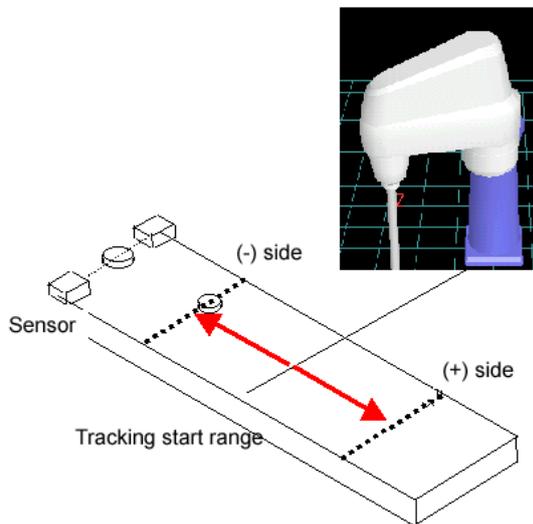
Waits for tracking-target workpiece to enter into a tracking start area.

Format

<error_information>=WAITTRACKMOVEEX(**<conveyer_number>**,
<recognized_workpiece_position>, **<timeout_duration>**)

Explanation

Waits for a workpiece specified by **<recognized_workpiece position>** to enter into a tracking start area on a conveyer specified by **<conveyer_number>**. **<timeout_duration>** is a maximum wait duration in seconds. A timeout error occurs if a workpiece does not enter into a tracking start area. The value of **<error_information>** is "0" while a target workpiece is out of a tracking start area, "1" after a timeout, and "2" when the workpiece is downstream from the tracking start area.



Related item

WAITTRACKMOVE

Example

```
PROGRAM PRO1
DEFINT REST, ERINF
DEFPOS POS1
TAKEARM
MOVE P,P1
TRACKDATAGET 1,0,REST,POS1
```

'Obtain the zeroth workpiece position from conveyer1 buffer.

```
ERINF=WAITTRACKMOVEEX(1,POS1,60)
```

'Wait for up to 60 seconds until a workpiece enters into a tracking area.

```
IF ERINF=0 THEN
```

```
CALL SETTRACKMOVE(1)
```

'Start conveyer 1 tracking.

```
MOVE P.,POS1
```

'Perform tracking to a workpiece position POS1 on a conveyer.

```
CALL RESETTRACKMOVE
```

'Terminate tracking of conveyer 1.

```
ENDIF
```

```
END
```

Notes

Tracking start area is a conveyer tracking parameter. Set this value using either the pendant or the tracking start area setting library.

Do not execute WAITTRACKMOVE while it is used in another program.

The suspended stop by Halt is calculated as wait time.

CAMIN (Statement)

Function

Captures an image from a camera into the specified image memory (processing screen).

Format

CAMIN <camera#>[,<memory#> [,<table#>]]

Explanation

<camera#>

Specifies a camera number. (1 or 2)

<memory#>

Specifies a memory number (processing screen number). (0 to 3)

Omitting this parameter automatically specifies "0."

<table#>

Specifies a lookup table number to use when getting an image. (0 to 15)

Omitting this parameter automatically specifies "0."

Related item

CAMMODE, CAMLEVEL, VISDEFTABLE

Example

```
CAMMODE 1,0,0      'Capture the image of camera 1 into memory 0
                   'after conversion with table 0 (same brightness level as the camera image)
DELAY 2000          'Stop for 2 seconds
CAMIN 1             'Same execution result as CAMIN 1,0,0
DELAY 2000          'Stop for 2 seconds
I1 = 1 '
I2 = 0 '
I3 = 3 '
CAMIN I1,I2,I3     'Capture the image of camera 1 into memory 0
                   'after conversion with table 3 (inversion)
VISPLNOUT 0        'Output the image held in memory 0 to the monitor (static image)
```

Notes

No connection of a camera or no correct data entry due to camera failure results in an error. If <table#> is non-zero, switching to the specified table is made. It may disturb the screen, but it is not a failure. After execution of this command, the table number automatically reverts to "0."

This command requires a µVision board (option) to be mounted in the robot controller.

CAMMODE (Statement)

Function

Specifies camera image capturing conditions.

Format

CAMMODE <camera#>, <capturing_option>, <capture_mode>

Explanation

<camera#>

Specifies a camera number. (1 or 2)

<capturing_option>

Specifies a capturing option. (0 or 1)

0: Normal (Usual camera setting)

1: Reset (After resetting a camera, capture a camera image.)

<capture_mode>

Specifies an image capture mode. (0 or 1)

0: Frame capture mode

This mode captures a single frame of a camera image with the maximum vertical resolution.

1: Field capture mode

This mode is selected to use the shutter feature of the field shutter camera. Using this mode causes no field-to-field imaging delay (1/60 sec.), enabling capture of non-blurred images.

Note that the vertical resolution comes to be half.

Related item

CAMIN, VISSTATUS

Example

```
CAMMODE 1,0,0
```

```
'Specify "Normal" capturing option and "Frame capture mode" for camera 1
```

```
I1 = VISSTATUS(0)
```

```
'If camera is normal, I1 = 0
```

```
IF I1 = 0 THEN
```

```
CAMIN 1
```

```
'Capture the image of camera 1 into memory 0
```

```
'after conversion with table 0 (same brightness level as the camera image)
```

```
VISPLNOUT 0
```

```
'Output the image held in memory 0 to the monitor (static image)
```

```
VISLOC 10,10
```

```
'Specify the display location
```

```
VISPRINT "Capturing normal"
```

```
'Display characters on the screen
```

```
ELSE
```

```
VISLOC 10,10
```

```
'Specify the display location
```

```
VISPRINT "Camera abnormal"
```

```
'Display characters on the screen
```

```
END IF
```

Notes

If this command is not specified, the factory settings apply.

This command preserves the factory settings and restarting the controller reverts to the factory settings. User-specified settings are lost.

Execution of this command checks connection of a camera. If a camera is judged as abnormal, VISSTATUS (0) returns "-1"; if as normal, it returns "0."

This command requires a μ Vision board (option) to be mounted in the robot controller.

3.8 List of Libraries Related to the Conveyer Tracking Function

SETTRACKMOVE (library)

Function

Starts the tracking operation for the specified conveyer

Format

SETTRACKMOVE(<conveyer_number>)

Explanation

This command starts the tracking operation for the conveyer specified in <conveyer_number>.

Related item

RESETTRACKMOVE

Example

```
PROGRAM PRO1
DEFINT REST
DEFPOS POS1
TAKEARM
MOVE P,P1
TRACKDATAGET 1,1,REST,POS1      'Obtaining the 1st workpiece position from
                                conveyer 1 data
WAITTRACKMOVE 1,POS1,60        'Waiting for the workpiece to enter the tracking
                                range for 60 seconds
CALL SETTRACKMOVE(1)           'Starting tracking for conveyer 1
MOVE P,POS1                     'Tracking operation for POS1
CALL RESETTRACKMOVE            'Terminating the tracking operation for conveyer 1
END
```

Notes

When you execute SETTRACKMOVE in the auto mode, the system enters the tracking operation mode.

When you execute SETTRACKMOVE in the teach check mode, an error 6002 "Wrong operation mode" occurs. In the teach check mode, the tracking operation mode is forbidden. During the tracking operation mode, the robot follows in the direction of the conveyer operation even if no operational instruction is executed.

To return from the tracking operation mode to the normal operation mode, execute RESETTRACKMOVE.

When the conveyer tracking program is completed or the mode switch is setted in the manual mode or GIVEARM command is executed, the operation mode returns from the tracking operation mode to the normal operation mode and an error 6841 "Interrupted track motion" occurs.

RESETTRACKMOVE (library)

Function

Switches from the tracking operation mode to the normal operation mode

Format

RESETTRACKMOVE

Explanation

This command stops the tracking operation and switches to the normal operation mode.

Related item

SETTRACKMOVE

Example

```
PROGRAM PRO1
DEFINT REST
DEFPOS POS1
TAKEARM
MOVE P,P1
TRACKDATAGET 1,1,REST,POS1           'Obtaining the 1st workpiece position from
                                     conveyer 1 data
WAITTRACKMOVE 1,POS1,60             'Waiting for the workpiece to enter the tracking
                                     range for 60 seconds
CALL SETTRACKMOVE(1)                'Starting tracking for conveyer 1
MOVE P,POS1                          'Tracking operation for POS 1
CALL RESETTRACKMOVE                  'Terminating the tracking operation for conveyer 1
END
```

Notes

During the tracking operation mode, the robot follows in the direction of the conveyer operation. When you execute RESETTRACKMOVE, the robot decelerates its speed and then stops, returning to the normal operation mode.

CONVCAL (library)

Function

Executes conveyer calibration

Format

CONVCAL(<conveyer_number>)

Explanation

This command executes calibration for the conveyer specified in <conveyer_number>.

Related item

Notes

For the use method, see "2.2 Conveyer Calibration".

CALCCAMCALPOS (library)

Function

Compute the robot coordinate values to be specified in XX when the WINCAPSIII Vision Manager executes calibration.

Format

CALCCAMCALPOS(<conveyer_number>, <P_type_variable_storage_number_1>, <P_type_variable_storage_number_2>, <P_type_variable_storage_number_3>)

Explanation

This command stores the three workpiece positions to be used for camera calibration on the conveyer specified in <conveyer_number> in the P-type variables specified in <P_type_variable_storage_number_1> to <P_type_variable_storage_number_3>.

Notes

For the use method, see “2.3 Camera Calibration”.

CALCIOTEACHPOS (library)

Function

Computes the workpiece position when Hand I/O interrupt is used

Format

CALCIOTEACHPOS (<conveyer_number>,<P_type_variable_storage_number_1>, <P_type_variable_storage_number_2>)

Explanation

This command computes the workpiece position at Hand I/O interrupt based on the teaching position and conveyer movement amount.

Notes

For the use method, see “2.4 Teaching of Workpiece Position when Hand I/O Input Interrupt is in Use”.

SetTrackStartArea (library)

Function

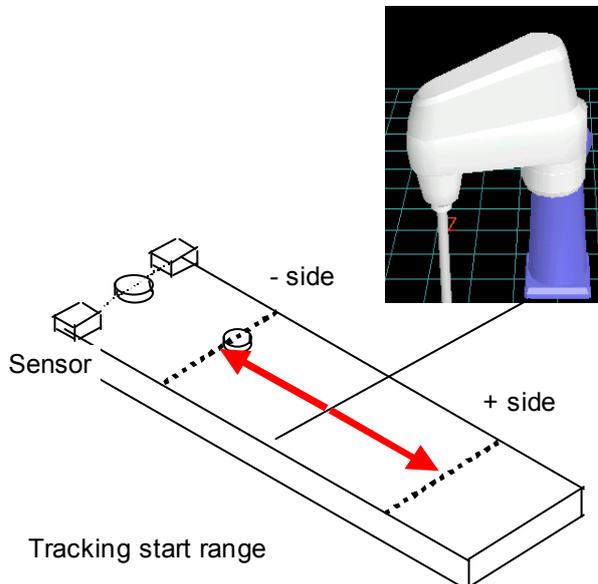
Sets the tracking start range at the time of WAITTRACKMOVE

Format

SetTrackStartArea
(<conveyor_number>,<conveyor_upstream_(-)side_tracking_start_position>,
<conveyor_downstream_(+)side_tracking_start_position>)

Explanation

This command sets the tracking start range at the time of WAITTRACKMOVE for the conveyor specified in <conveyor_number>.
<conveyor_upstream_(-)side_tracking_start_position > and
<conveyor_downstream_(+)side_tracking_start_position> are specified with conveyor positions.



Related item

WAITTRACKMOVE

Example

CALL SetTrackStartArea (1,-100,1000) 'Specifying that the tracking start range for conveyor 1 is between -100 mm and +1000 mm
TRACKDATAGET 1,0,REST,POS1 'Obtaining the 0th workpiece position from the conveyor 1 buffer
WAITTRACKMOVE 1,POS1,60 'Waiting for the workpiece to enter the tracking range for up to 60 seconds

CalcConvPos (library)

Function

Converts the work position of robot coordinate system to the conveyer position.

Format

CalcConvPos (<conveyer_number>, <work_position>, <conveyer_position>)

Explanation

Converts the work position specified in <work_position> of robot coordinate system to the conveyer position specified in <conveyer_number>.

Related item

CurTrackPos

Example

```
DEFPOS Pos1, WPos
```

```
DEFSNG CnvPos
```

```
WPos=CurTrackPos(1, Pos1, 1)
```

'Caluculates the work position when conveyer 1 finishes PTP motion.

```
CALL CalcConvPos(1, WPos, CnvPos)
```

'Caluculates the conveyer position of conveyer 1.

SetConvLowVelErr (library)

Function

Sets the low speed error detecting function.

Format

SetConvLowVelErr((<conveyer_number>)

Explanation

Enables the low speed error detecting function of the conveyer specified in <conveyer_number>.

Entering 0 for <conveyer_number> disables the low speed error detecting function.

The low speed error detecting function detects when the conveyer speed value becomes lower than the setting value in conveyer tracking parameter.

Example

```
CALL SetConvLowVelErr(1,)
```

'Enables the low speed error detecting function of conveyer 1.

CalcConvVec (library)

Function

Calculates conveyer vector.

Format

CalcConvVec (<conveyer_number>, <conveyer_vector>)

Explanation

Calculates conveyer vector of a conveyer specified by <conveyer_number>.

Related item

CalcConvPos

Example

```
DEFVEC CnvVec
CALL CalcConvVec(1,CnvVec)    'Calculate a conveyer vector of conveyer 1.
```

SortTrackData (library)

Function

Calculates the number of a data item, in a conveyer tracking buffer, that is the most downstream on a conveyer.

Format

SortTrackData (<conveyer_number>, <conveyer_buffer_data_count>, <conveyer_vector>, <the_number_of_the_most_downstream_data_item>)

Explanation

Calculates the number of the most downstream data item selected from the data specified by <Conveyer buffer data count> in a conveyer tracking data buffer of a conveyer specified by <Conveyer number>, and assigns the number of the data to <The most downstream data number>.

Obtain <Conveyer buffer data count> using TrackDataNum.

Calculate <Conveyer vector> using CalcConvVec.

Related item

TrackDataGet, TrackDataNum, CalcConvVec

Example

```
DEFPOS TrackPos
DEFVEC CnvVec
DEFINT datnum, resnum
CALL CalcConvVec(1,CnvVec)    'Calculation of the conveyer vector of conveyer 1
CALL SortTrackData(1,TrackDataNum(1),CnvVec, datnum)
TrackDataGet 1,datnum,resnum,TrackPos
```

SortTrackAllData (library)

Function

Calculates the number of a data item, from all data in a conveyer tracking buffer, that is the most downstream on a conveyer.

Format

SortTrackAllData (<conveyer_number>,
<the_number_of_the_most_downstream_data_item>)

Explanation

Calculates the number of the most downstream data item selected from all data in a conveyer tracking data buffer of a conveyer specified by <Conveyer number>, and assigns the number of the data to <The most downstream data number>.

Related item

SortTrackData

Example

```
DEFPOS TrackPos  
DEFINT datnum, resnum  
CALL SortTrackAllData(1, datnum)  
TrackDataGet 1,datnum,resnum,TrackPos
```

RC7M CONTROLLER CONVEYER TRACKING BOARD

OPTIONS MANUAL (SUPPLEMENT)

First Edition	August 2005
Third Edition	March 2010
Fourth Edition	April 2011

DENSO WAVE INCORPORATED

4N**C

The purpose of this manual is to provide accurate information in the handling and operating of the robot. Please feel free to send your comments regarding any errors or omissions you may have found, or any suggestions you may have for generally improving the manual.

In no event will DENSO WAVE INCORPORATED be liable for any direct or indirect damages resulting from the application of the information in this manual.

