FANUC Robot series

R-30*i*B CONTROLLER

*i*RVision Visual Tracking

OPERATOR'S MANUAL

B-83304EN-4/01

• Original Instructions

Before using the Robot, be sure to read the "FANUC Robot Safety Manual (B-80687EN)" and understand the content.

The products in this manual are controlled based on Japan's "Foreign Exchange and Foreign Trade Law". The export from Japan may be subject to an export license by the government of Japan.
Further, re-export to another country may be subject to the license of the government of the country from where the product is re-exported. Furthermore, the product may also be controlled by re-export regulations of the United States government.
Should you wish to export or re-export these products, please contact FANUC for advice.

In this manual we have tried as much as possible to describe all the various matters.
However, we cannot describe all the matters which must not be done, or which cannot be done, because there are so many possibilities.
Therefore, matters which are not especially described as possible in this manual should be regarded as "impossible".

SAFETY PRECAUTIONS

Thank you for purchasing FANUC Robot.
This chapter describes the precautions which must be observed to ensure the safe use of the robot.
Before attempting to use the robot, be sure to read this chapter thoroughly.

Before using the functions related to robot operation, read the relevant operator's manual to become familiar with those functions.

If any description in this chapter differs from that in the other part of this manual, the description given in this chapter shall take precedence.

For the safety of the operator and the system, follow all safety precautions when operating a robot and its peripheral devices installed in a work cell.
In addition, refer to the "FANUC Robot SAFETY HANDBOOK (B-80687EN)".

1 WORKING PERSON

The personnel can be classified as follows.

Operator:
- Turns robot controller power ON/OFF
- Starts robot program from operator's panel

Programmer or teaching operator:
- Operates the robot
- Teaches robot inside the safety fence

Maintenance engineer:
- Operates the robot
- Teaches robot inside the safety fence
- Maintenance (adjustment, replacement)

- An operator cannot work inside the safety fence.
- A programmer, teaching operator, and maintenance engineer can work inside the safety fence. The working activities inside the safety fence include lifting, setting, teaching, adjusting, maintenance, etc.
- To work inside the fence, the person must be trained on proper robot operation.

During the operation, programming, and maintenance of your robotic system, the programmer, teaching operator, and maintenance engineer should take additional care of their safety by using the following safety precautions.

- Use adequate clothing or uniforms during system operation
- Wear safety shoes
- Use helmet

2 DEFINITION OF WARNING, CAUTION AND NOTE

To ensure the safety of user and prevent damage to the machine, this manual indicates each precaution on safety with "Warning" or "Caution" according to its severity. Supplementary information is indicated by "Note". Read the contents of each "Warning", "Caution" and "Note" before attempting to use the oscillator.

> ⚠**WARNING**
> Applied when there is a danger of the user being injured or when there is a danger of both the user being injured and the equipment being damaged if the approved procedure is not observed.

> ⚠**CAUTION**
> Applied when there is a danger of the equipment being damaged, if the approved procedure is not observed.

> **NOTE**
> Notes are used to indicate supplementary information other than Warnings and Cautions.

- Read this manual carefully, and store it in a sales place.

3 WORKING PERSON SAFETY

Working person safety is the primary safety consideration. Because it is very dangerous to enter the operating space of the robot during automatic operation, adequate safety precautions must be observed.
The following lists the general safety precautions. Careful consideration must be made to ensure working person safety.

(1) Have the robot system working persons attend the training courses held by FANUC.

> FANUC provides various training courses. Contact our sales office for details.

(2) Even when the robot is stationary, it is possible that the robot is still in a ready to move state, and is waiting for a signal. In this state, the robot is regarded as still in motion. To ensure working person safety, provide the system with an alarm to indicate visually or aurally that the robot is in motion.
(3) Install a safety fence with a gate so that no working person can enter the work area without passing through the gate. Install an interlocking device, a safety plug, and so forth in the safety gate so that the robot is stopped as the safety gate is opened.

> The controller is designed to receive this interlocking signal of the door switch. When the gate is opened and this signal received, the controller stops the robot (Please refer to "STOP TYPE OF ROBOT" in SAFETY PRECAUTIONS for detail of stop type). For connection, see Fig.3 (a) and Fig.3 (b).

(4) Provide the peripheral devices with appropriate grounding (Class A, Class B, Class C, and Class D).

(5) Try to install the peripheral devices outside the work area.
(6) Draw an outline on the floor, clearly indicating the range of the robot motion, including the tools such as a hand.
(7) Install a mat switch or photoelectric switch on the floor with an interlock to a visual or aural alarm that stops the robot when a working person enters the work area.
(8) If necessary, install a safety lock so that no one except the working person in charge can turn on the power of the robot.

The circuit breaker installed in the controller is designed to disable anyone from turning it on when it is locked with a padlock.

(9) When adjusting each peripheral device independently, be sure to turn off the power of the robot
(10) Operators should be ungloved while manipulating the operator’s panel or teach pendant. Operation with gloved fingers could cause an operation error.
(11) Programs, system variables, and other information can be saved on memory card or USB memories. Be sure to save the data periodically in case the data is lost in an accident.
(12) The robot should be transported and installed by accurately following the procedures recommended by FANUC. Wrong transportation or installation may cause the robot to fall, resulting in severe injury to workers.
(13) In the first operation of the robot after installation, the operation should be restricted to low speeds. Then, the speed should be gradually increased to check the operation of the robot.
(14) Before the robot is started, it should be checked that no one is in the area of the safety fence. At the same time, a check must be made to ensure that there is no risk of hazardous situations. If detected, such a situation should be eliminated before the operation.
(15) When the robot is used, the following precautions should be taken. Otherwise, the robot and peripheral equipment can be adversely affected, or workers can be severely injured.
- Avoid using the robot in a flammable environment.
- Avoid using the robot in an explosive environment.
- Avoid using the robot in an environment full of radiation.
- Avoid using the robot under water or at high humidity.
- Avoid using the robot to carry a person or animal.
- Avoid using the robot as a stepladder. (Never climb up on or hang from the robot.)

(16) When connecting the peripheral devices related to stop(safety fence etc.) and each signal (external emergency , fence etc.) of robot. be sure to confirm the stop movement and do not take the wrong connection.
(17) When preparing trestle, please consider security for installation and maintenance work in high place according to Fig.3 (c). Please consider footstep and safety bolt mounting position.

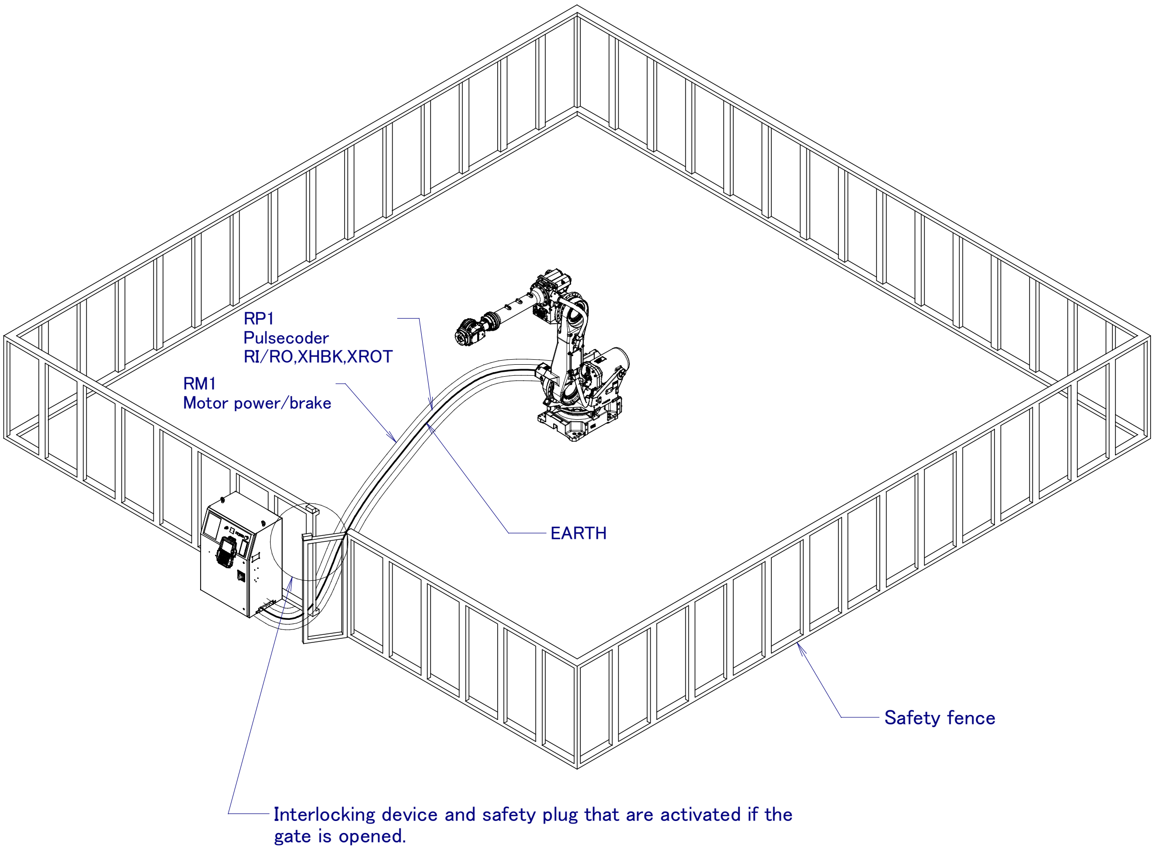
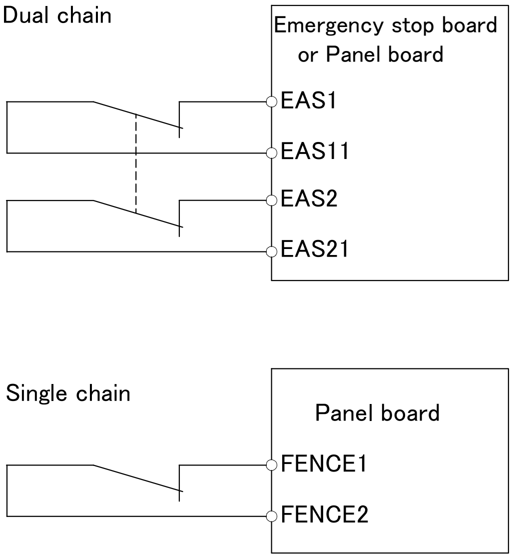


Fig. 3 (a) Safety Fence and Safety gate



(Note)

In case of R-30*i*B
Terminals EAS1,EAS11,EAS2,EAS21 are provided on the emergency stop board.

Refer to the ELECTRICAL CONNCETIONS Chapter of CONNECTION of controller maintenance manual for details.

Fig. 3 (b) Limit switch circuit diagram of the safety fence

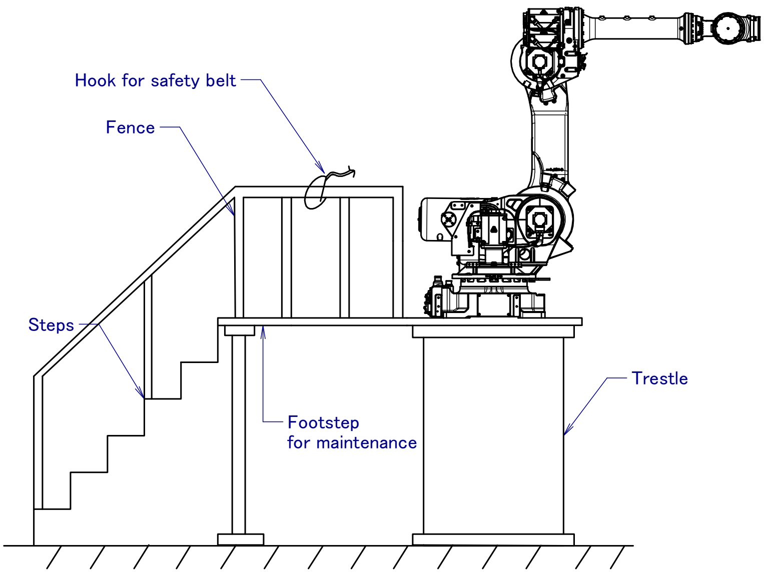


Fig.3 (c) Footstep for maintenance

3.1 OPERATOR SAFETY

The operator is a person who operates the robot system. In this sense, a worker who operates the teach pendant is also an operator. However, this section does not apply to teach pendant operators.

(1) If you do not have to operate the robot, turn off the power of the robot controller or press the EMERGENCY STOP button, and then proceed with necessary work.
(2) Operate the robot system at a location outside of the safety fence
(3) Install a safety fence with a safety gate to prevent any worker other than the operator from entering the work area unexpectedly and to prevent the worker from entering a dangerous area.
(4) Install an EMERGENCY STOP button within the operator's reach.

The robot controller is designed to be connected to an external EMERGENCY STOP button. With this connection, the controller stops the robot operation (Please refer to "STOP TYPE OF ROBOT" in SAFETY PRECAUTIONS for detail of stop type), when the external EMERGENCY STOP button is pressed. See the diagram below for connection.

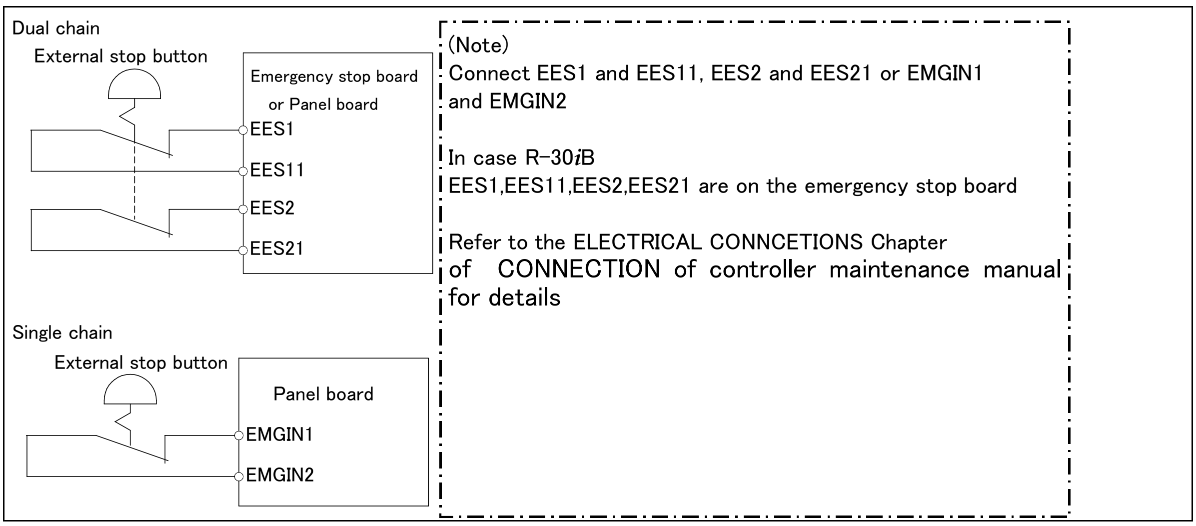


Fig.3.1 Connection Diagram for External Emergency Stop Button

3.2 SAFETY OF THE PROGRAMMER

While teaching the robot, the operator must enter the work area of the robot. The operator must ensure the safety of the teach pendant operator especially.

(1) Unless it is specifically necessary to enter the robot work area, carry out all tasks outside the area.
(2) Before teaching the robot, check that the robot and its peripheral devices are all in the normal operating condition.
(3) If it is inevitable to enter the robot work area to teach the robot, check the locations, settings, and other conditions of the safety devices (such as the EMERGENCY STOP button, the DEADMAN switch on the teach pendant) before entering the area.
(4) The programmer must be extremely careful not to let anyone else enter the robot work area.
(5) Programming should be done outside the area of the safety fence as far as possible. If programming needs to be done in the area of the safety fence, the programmer should take the following precautions:
   - Before entering the area of the safety fence, ensure that there is no risk of dangerous situations in the area.
   - Be prepared to press the emergency stop button whenever necessary.
   - Robot motions should be made at low speeds.
   - Before starting programming, check the entire system status to ensure that no remote instruction to the peripheral equipment or motion would be dangerous to the user.

Our operator panel is provided with an emergency stop button and a key switch (mode switch) for selecting the automatic operation mode (AUTO) and the teach modes (T1 and T2). Before entering the inside of the safety fence for the purpose of teaching, set the switch to a teach mode, remove the key from the mode switch to prevent other people from changing the operation mode carelessly, then open the safety gate. If the safety gate is opened with the automatic operation mode set, the robot stops (Please refer to "STOP TYPE OF ROBOT" in SAFETY PRECAUTIONS for detail of stop type). After the switch is set to a teach mode, the safety gate is disabled. The programmer should understand that the safety gate is disabled and is responsible for keeping other people from entering the inside of the safety fence.

Our teach pendant is provided with a DEADMAN switch as well as an emergency stop button. These button and switch function as follows:

(1) Emergency stop button: Causes an emergency stop (Please refer to "STOP TYPE OF ROBOT" in SAFETY PRECAUTIONS for detail of stop type) when pressed.
(2) DEADMAN switch: Functions differently depending on the teach pendant enable/disable switch setting status.
   (a) Disable: The DEADMAN switch is disabled.
   (b) Enable: Servo power is turned off when the operator releases the DEADMAN switch or when the operator presses the switch strongly.

   Note) The DEADMAN switch is provided to stop the robot when the operator releases the teach pendant or presses the pendant strongly in case of emergency. The R-30*i*B employs a 3-position DEADMAN switch, which allows the robot to operate when the 3-position DEADMAN switch is pressed to its intermediate point. When the operator releases the DEADMAN switch or presses the switch strongly, the robot stops immediately.

The operator's intention of starting teaching is determined by the controller through the dual operation of setting the teach pendant enable/disable switch to the enable position and pressing the DEADMAN switch. The operator should make sure that the robot could operate in such conditions and be responsible in carrying out tasks safely.

Based on the risk assessment by FANUC, number of operation of DEADMAN SW should not exceed about 10000 times per year.

The teach pendant, operator panel, and peripheral device interface send each robot start signal. However the validity of each signal changes as follows depending on the mode switch and the DEADMAN switch of the operator panel, the teach pendant enable switch and the remote condition on the software.

In case of R-30*i*B Controller

Mode	Teach pendant enable switch	Software remote condition	Teach pendant	Operator panel	Peripheral device
AUTO mode	On	Local	Not allowed	Not allowed	Not allowed
		Remote	Not allowed	Not allowed	Not allowed
	Off	Local	Not allowed	Allowed to start	Not allowed
		Remote	Not allowed	Not allowed	Allowed to start
T1, T2 mode	On	Local	Allowed to start	Not allowed	Not allowed
		Remote	Allowed to start	Not allowed	Not allowed
	Off	Local	Not allowed	Not allowed	Not allowed
		Remote	Not allowed	Not allowed	Not allowed

T1,T2 mode:DEADMAN switch is effective.

(6) To start the system using the operator’s panel, make certain that nobody is the robot work area and that there are no abnormal conditions in the robot work area.
(7) When a program is completed, be sure to carry out a test operation according to the procedure below.
  (a) Run the program for at least one operation cycle in the single step mode at low speed.
  (b) Run the program for at least one operation cycle in the continuous operation mode at low speed.
  (c) Run the program for one operation cycle in the continuous operation mode at the intermediate speed and check that no abnormalities occur due to a delay in timing.
  (d) Run the program for one operation cycle in the continuous operation mode at the normal operating speed and check that the system operates automatically without trouble.
  (e) After checking the completeness of the program through the test operation above, execute it in the automatic operation mode.
(8) While operating the system in the automatic operation mode, the teach pendant operator should leave the robot work area.

3.3 SAFETY OF THE MAINTENANCE ENGINEER

For the safety of maintenance engineer personnel, pay utmost attention to the following.

(1) During operation, never enter the robot work area.
(2) A hazardous situation may arise when the robot or the system, are kept with their power-on during maintenance operations. Therefore, for any maintenance operation, the robot and the system should be put into the power-off state. If necessary, a lock should be in place in order to prevent any other person from turning on the robot and/or the system. In case maintenance needs to be executed in the power-on state, the emergency stop button must be pressed.
(3) If it becomes necessary to enter the robot operation range while the power is on, press the emergency stop button on the operator panel, or the teach pendant before entering the range. The maintenance personnel must indicate that maintenance work is in progress and be careful not to allow other people to operate the robot carelessly.
(4) When entering the area enclosed by the safety fence, the maintenance worker must check the entire system in order to make sure no dangerous situations exist. In case the worker needs to enter the safety area whilst a dangerous situation exists, extreme care must be taken, and entire system status must be carefully monitored.

(5) Before the maintenance of the pneumatic system is started, the supply pressure should be shut off and the pressure in the piping should be reduced to zero.
(6) Before the start of teaching, check that the robot and its peripheral devices are all in the normal operating condition.
(7) Do not operate the robot in the automatic mode while anybody is in the robot work area.
(8) When you maintain the robot alongside a wall or instrument, or when multiple workers are working nearby, make certain that their escape path is not obstructed.
(9) When a tool is mounted on the robot, or when any moving device other than the robot is installed, such as belt conveyor, pay careful attention to its motion.
(10) If necessary, have a worker who is familiar with the robot system stand beside the operator panel and observe the work being performed. If any danger arises, the worker should be ready to press the EMERGENCY STOP button at any time.
(11) When replacing a part, please contact FANUC service center. If a wrong procedure is followed, an accident may occur, causing damage to the robot and injury to the worker.
(12) When replacing or reinstalling components, take care to prevent foreign material from entering the system.
(13) When handling each unit or printed circuit board in the controller during inspection, turn off the circuit breaker to protect against electric shock.
If there are two cabinets, turn off the both circuit breaker.
(14) A part should be replaced with a part recommended by FANUC. If other parts are used, malfunction or damage would occur. Especially, a fuse that is not recommended by FANUC should not be used. Such a fuse may cause a fire.
(15) When restarting the robot system after completing maintenance work, make sure in advance that there is no person in the work area and that the robot and the peripheral devices are not abnormal.
(16) When a motor or brake is removed, the robot arm should be supported with a crane or other equipment beforehand so that the arm would not fall during the removal.
(17) Whenever grease is spilled on the floor, it should be removed as quickly as possible to prevent dangerous falls.
(18) The following parts are heated. If a maintenance worker needs to touch such a part in the heated state, the worker should wear heat-resistant gloves or use other protective tools.
- Servo motor
- Inside the controller
- Reducer
- Gearbox
- Wrist unit

(19) Maintenance should be done under suitable light. Care must be taken that the light would not cause any danger.
(20) When a motor, reducer, or other heavy load is handled, a crane or other equipment should be used to protect maintenance workers from excessive load. Otherwise, the maintenance workers would be severely injured.
(21) The robot should not be stepped on or climbed up during maintenance. If it is attempted, the robot would be adversely affected. In addition, a misstep can cause injury to the worker.
(22) When performing maintenance work in high place, secure a footstep and wear safety belt.
(23) After the maintenance is completed, spilled oil or water and metal chips should be removed from the floor around the robot and within the safety fence.
(24) When a part is replaced, all bolts and other related components should put back into their original places. A careful check must be given to ensure that no components are missing or left not mounted.
(25) In case robot motion is required during maintenance, the following precautions should be taken :
- Foresee an escape route. And during the maintenance motion itself, monitor continuously the whole system so that your escape route will not become blocked by the robot, or by peripheral equipment.
- Always pay attention to potentially dangerous situations, and be prepared to press the emergency stop button whenever necessary.

(26) The robot should be periodically inspected. (Refer to the robot mechanical manual and controller maintenance manual.) A failure to do the periodical inspection can adversely affect the performance or service life of the robot and may cause an accident

(27) After a part is replaced, a test operation should be given for the robot according to a predetermined method. (See TESTING section of “Controller operator’s manual”.) During the test operation, the maintenance staff should work outside the safety fence.

4 SAFETY OF THE TOOLS AND PERIPHERAL DEVICES

4.1 PRECAUTIONS IN PROGRAMMING

(1) Use a limit switch or other sensor to detect a dangerous condition and, if necessary, design the program to stop the robot when the sensor signal is received.

(2) Design the program to stop the robot when an abnormal condition occurs in any other robots or peripheral devices, even though the robot itself is normal.

(3) For a system in which the robot and its peripheral devices are in synchronous motion, particular care must be taken in programming so that they do not interfere with each other.

(4) Provide a suitable interface between the robot and its peripheral devices so that the robot can detect the states of all devices in the system and can be stopped according to the states.

4.2 PRECAUTIONS FOR MECHANISM

(1) Keep the component cells of the robot system clean, and operate the robot in an environment free of grease, water, and dust.

(2) Don’t use unconfirmed liquid for cutting fluid and cleaning fluid.

(3) Employ a limit switch or mechanical stopper to limit the robot motion so that the robot or cable does not strike against its peripheral devices or tools.

(4) Observe the following precautions about the mechanical unit cables. When theses attentions are not kept, unexpected troubles might occur.
   - Use mechanical unit cable that have required user interface.
   - Don’t add user cable or hose to inside of mechanical unit.
   - Please do not obstruct the movement of the mechanical unit cable when cables are added to outside of mechanical unit.
   - In the case of the model that a cable is exposed, Please do not perform remodeling (Adding a protective cover and fix an outside cable more) obstructing the behavior of the outcrop of the cable.
   - Please do not interfere with the other parts of mechanical unit when install equipments in the robot.

(5) The frequent power-off stop for the robot during operation causes the trouble of the robot. Please avoid the system construction that power-off stop would be operated routinely. (Refer to bad case example.) Please execute power-off stop after reducing the speed of the robot and stopping it by hold stop or cycle stop when it is not urgent. (Please refer to "STOP TYPE OF ROBOT" in SAFETY PRECAUTIONS for detail of stop type.)
   (Bad case example)
   - Whenever poor product is generated, a line stops by emergency stop.
   - When alteration was necessary, safety switch is operated by opening safety fence and power-off stop is executed for the robot during operation.
   - An operator pushes the emergency stop button frequently, and a line stops.
   - An area sensor or a mat switch connected to safety signal operate routinely and power-off stop is executed for the robot.

(6) Robot stops urgently when collision detection alarm (SRVO-050) etc. occurs. The frequent urgent stop by alarm causes the trouble of the robot, too. So remove the causes of the alarm.

5 SAFETY OF THE ROBOT MECHANISM

5.1 PRECAUTIONS IN OPERATION

(1) When operating the robot in the jog mode, set it at an appropriate speed so that the operator can manage the robot in any eventuality.
(2) Before pressing the jog key, be sure you know in advance what motion the robot will perform in the jog mode.

5.2 PRECAUTIONS IN PROGRAMMING

(1) When the work areas of robots overlap, make certain that the motions of the robots do not interfere with each other.
(2) Be sure to specify the predetermined work origin in a motion program for the robot and program the motion so that it starts from the origin and terminates at the origin.
Make it possible for the operator to easily distinguish at a glance that the robot motion has terminated.

5.3 PRECAUTIONS FOR MECHANISMS

(1) Keep the work areas of the robot clean, and operate the robot in an environment free of grease, water, and dust.

5.4 PROCEDURE TO MOVE ARM WITHOUT DRIVE POWER IN EMERGENCY OR ABNORMAL SITUATIONS

For emergency or abnormal situations (e.g. persons trapped in or by the robot), brake release unit can be used to move the robot axes without drive power.
Please refer to controller maintenance manual and mechanical unit operator’s manual for using method of brake release unit and method of supporting robot.

6 SAFETY OF THE END EFFECTOR

6.1 PRECAUTIONS IN PROGRAMMING

(1) To control the pneumatic, hydraulic and electric actuators, carefully consider the necessary time delay after issuing each control command up to actual motion and ensure safe control.
(2) Provide the end effector with a limit switch, and control the robot system by monitoring the state of the end effector.

7 STOP TYPE OF ROBOT

The following three robot stop types exist:

Power-Off Stop (Category 0 following IEC 60204-1)

Servo power is turned off and the robot stops immediately. Servo power is turned off when the robot is moving, and the motion path of the deceleration is uncontrolled.
The following processing is performed at Power-Off stop.

- An alarm is generated and servo power is turned off.
- The robot operation is stopped immediately. Execution of the program is paused.

Controlled stop (Category 1 following IEC 60204-1)

The robot is decelerated until it stops, and servo power is turned off.
The following processing is performed at Controlled stop.

- The alarm "SRVO-199 Controlled stop" occurs along with a decelerated stop. Execution of the program is paused.
- An alarm is generated and servo power is turned off.

Hold (Category 2 following IEC 60204-1)

The robot is decelerated until it stops, and servo power remains on.
The following processing is performed at Hold.

- The robot operation is decelerated until it stops. Execution of the program is paused.

> ⚠ **WARNING**
> The stopping distance and stopping time of Controlled stop are longer than the stopping distance and stopping time of Power-Off stop. A risk assessment for the whole robot system, which takes into consideration the increased stopping distance and stopping time, is necessary when Controlled stop is used.

When the emergency stop button is pressed or the FENCE is open, the stop type of robot is Power-Off stop or Controlled stop. The configuration of stop type for each situation is called *stop pattern*. The stop pattern is different according to the controller type or option configuration.

There are the following 3 Stop patterns.

Stop pattern	Mode	Emergency stop button	External Emergency stop	FENCE open	SVOFF input	Servo disconnect
A	AUTO	P-Stop	P-Stop	C-Stop	C-Stop	P-Stop
	T1	P-Stop	P-Stop	-	C-Stop	P-Stop
	T2	P-Stop	P-Stop	-	C-Stop	P-Stop
B	AUTO	P-Stop	P-Stop	P-Stop	P-Stop	P-Stop
	T1	P-Stop	P-Stop	-	P-Stop	P-Stop
	T2	P-Stop	P-Stop	-	P-Stop	P-Stop
C	AUTO	C-Stop	C-Stop	C-Stop	C-Stop	C-Stop
	T1	P-Stop	P-Stop	-	C-Stop	P-Stop
	T2	P-Stop	P-Stop	-	C-Stop	P-Stop

P-Stop: Power-Off stop
C-Stop: Controlled stop
-: Disable

The following table indicates the Stop pattern according to the controller type or option configuration.

Option	R-30*i*B
Standard	A (*)
Controlled stop by E-Stop (A05B-2600-J570)	C (*)

(*) R-30*i*B does not have servo disconnect.

The stop pattern of the controller is displayed in "Stop pattern" line in software version screen. Please refer to "Software version" in operator's manual of controller for the detail of software version screen.

"Controlled stop by E-Stop" option

When "Controlled stop by E-Stop" (A05B-2600-J570) option is specified, the stop type of the following alarms becomes Controlled stop but only in AUTO mode. In T1 or T2 mode, the stop type is Power-Off stop which is the normal operation of the system.

Alarm	Condition
SRVO-001 Operator panel E-stop	Operator panel emergency stop is pressed.
SRVO-002 Teach pendant E-stop	Teach pendant emergency stop is pressed.
SRVO-007 External emergency stops	External emergency stop input (EES1-EES11, EES2-EES21) is open.
SRVO-218 Ext. E-stop/Servo Disconnect	External emergency stop input (EES1-EES11, EES2-EES21) is open.
SRVO-408 DCS SSO Ext Emergency Stop	In DCS Safe I/O connect function, SSO[3] is OFF.
SRVO-409 DCS SSO Servo Disconnect	In DCS Safe I/O connect function, SSO[4] is OFF.

Controlled stop is different from Power-Off stop as follows:

- In Controlled stop, the robot is stopped on the program path. This function is effective for a system where the robot can interfere with other devices if it deviates from the program path.
- In Controlled stop, physical impact is less than Power-Off stop. This function is effective for systems where the physical impact to the mechanical unit or EOAT (End Of Arm Tool) should be minimized.
- The stopping distance and stopping time of Controlled stop is longer than the stopping distance and stopping time of Power-Off stop, depending on the robot model and axis. Please refer to the operator's manual of a particular robot model for the data of stopping distance and stopping time.

When this option is loaded, this function cannot be disabled.

The stop type of DCS Position and Speed Check functions is not affected by the loading of this option.

> ⚠ **WARNING**
> The stopping distance and stopping time of Controlled stop are longer than the stopping distance and stopping time of Power-Off stop. A risk assessment for the whole robot system, which takes into consideration the increased stopping distance and stopping time, is necessary when this option is loaded.

120919

TABLE OF CONTENTS

1 PREFACE

This chapter describes an overview of this manual, which should be read before operating the *i*RVision Visual Tracking function.

1.1 OVERVIEW OF THE MANUAL

Overview

This manual describes how to operate *i*RVision controlled by the R-30*i*B controller.
In this manual, only the operation and the technique of programming for the dedicated sensor functions are explained, assuming the installation and the setup of the robot are completed. Refer to the "R-30iB CONTROLLER OPERATOR'S MANUAL (Basic Operation)" about other operations of FANUC Robots.

> ⚠**CAUTION**
> This manual is based on R-30*i*B system software version V8.10P/04. Note that the functions and settings described in this manual may not be available, and those not described in this manual may be available, and some notation differences are present, depending on the software version.

> **NOTE**
> This manual describes start-up procedure of a visual line tracking robot system using *i*RVision. This manual describes how to program, points to keep in mind, and examples of various scenarios of a visual tracking system. Additionally, this manual also supports queue managed line tracking system. If the system performs queue managed line tracking system, please replace the term "visual tracking" with "queue managed line tracking".

Contents of this manual

Chapter 1	How to use this manual
Chapter 2	Overview
Chapter 3	Preparation
Chapter 4	Setup
Chapter 5	Variation
Chapter 6	Multi-view
Chapter 7	Other functions
Chapter 8	Troubleshooting
Chapter 9	Appendix

1.2 RELATED MANUALS

This section introduces related manuals.

R-30*i*B CONTROLLER OPERATOR'S MANUAL (Basic Operation) B-83284EN

This is the main manual of R-30*i*B Controller. This manual describes the following items for manipulating workpieces with the robot:

- Setting the system for manipulating workpieces
- Operating the robot

- Creating and changing a program
- Executing a program
- Status indications
- Backup and restore robot programs.

This manual is used on an applicable design, robot installation, robot teaching.

R-30*i*B CONTROLLER MAINTENANCE MANUAL B-83195EN

This manual describes the maintenance and connection of R-30*i*B Controller.

R-30*i*B CONTROLLER OPERATOR'S MANUAL (Alarm Code List) B-83284EN-1

This manual describes the error code listings, causes, and remedies of R-30*i*B Controller.

R-30*i*B CONTROLLER *i*RVision OPERATOR'S MANUAL (Reference) B-83304EN

This manual is the reference manual for *i*RVision on the R-30*i*B controller. This manual describes each functions which are provided by *i*RVision. This manual describes the meanings (e.g. the items on *i*RVision setup screen, the arguments of the instruction, and so on.

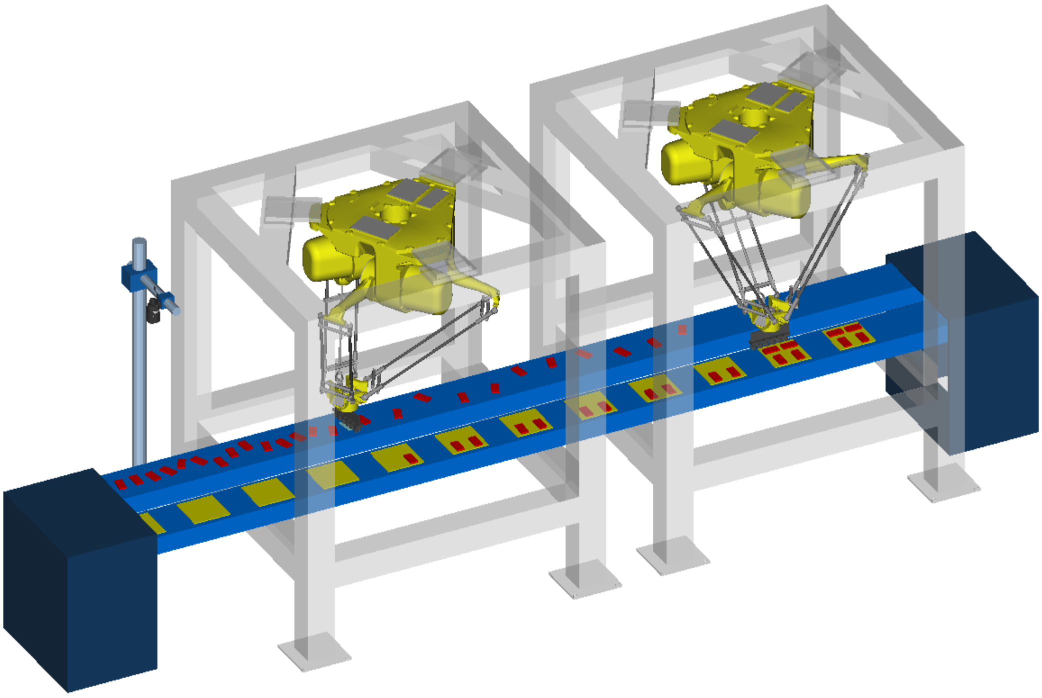
R-30*i*B CONTROLLER Sensor Mechanical Unit / Control Unit OPERATOR'S MANUAL B-83434EN

This manual describes the connection between sensors which is a camera or 3D Laser Sensor and R-30*i*B Controller, and maintenance of sensors.

2 OVERVIEW

Visual tracking and queue managed line tracking are applications, which detect many workpieces flowing on a conveyor, and guides robots to pick up the workpieces as the workpieces move along the conveyor belt. *i*RVision visual tracking has an excellent queue management function, which allows multiple robots to share the work of picking up workpieces flowing on the conveyor. It also supports multiple conveyors, enabling a conveyor-to-conveyor application in which each robot picks up workpieces from one conveyor and puts them on the other conveyor. Non camera tracking solutions are also supported, where sensors on the conveyors detect the parts instead of cameras.

This manual sequentially describes the procedure to build a visual tracking system. An *i*RVision visual tracking system can be built by performing the operations described in Chapter 2 through Chapter 4 according to the specified procedure. Chapter 5 introduces many variations and sample applications that are not contained in Chapters 2 to 4. A queue managed line tracking system can also be built in the same way. Please replace the term “visual tracking” with “queue managed line tracking”.



2.1 CONFIGURATION

This section describes device configuration of *i*RVision visual tracking system and queue managed line tracing system.

2.1.1 Visual Tracking

A visual tracking system consists of the following devices.

Devices required for system running

- Robot controller
- Robot
- Conveyor
- Sensor such as a photo eye (optional)

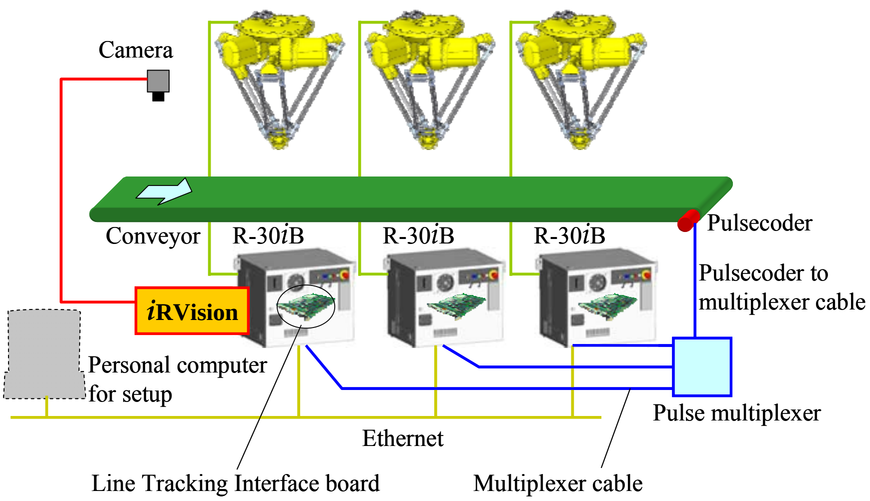
- Camera
- Lens
- Camera cable
- Lighting
- Pulsecoder
- Line tracking interface board (installed in the robot controller)
- Pulsecoder cable
- Pulse multiplexer and multiplexer cables (required when multiple robots are used and the Ethernet Encoder is not used)
- Ethernet communication cable
- Ethernet hub (required when multiple robots are used)

Devices required only for system start-up

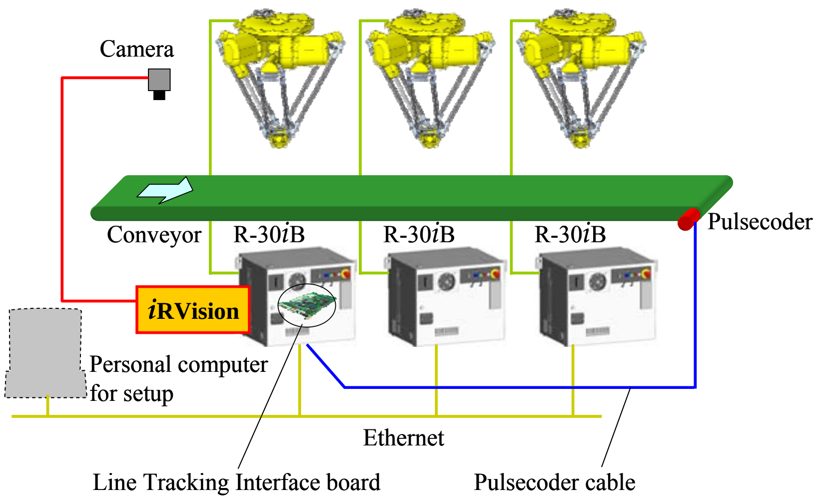
- Personal computer for setup and Ethernet communication cable for the personal computer
- Calibration Grid
- Pointer

Sample configuration

When a pulse multiplexer is used



When the Ethernet Encoder is used



2.1.2 Queue Managed Line Tracking

A queue managed line tracking system consists of the following devices.

Devices required for system running

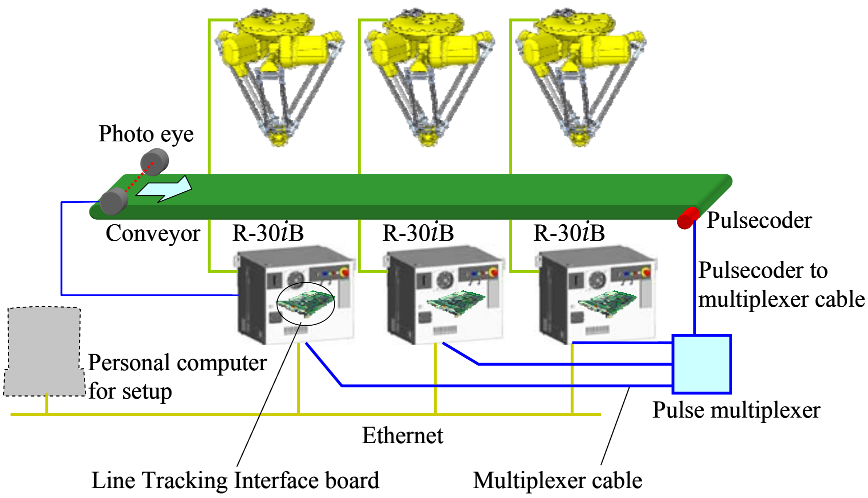
- Robot controller
- Robot
- Conveyor
- Sensor such as a photo eye
- Pulsecoder
- Line tracking interface board (installed in the robot controller)
- Pulsecoder cable
- Pulse multiplexer and multiplexer cables (required when multiple robots are used and the Ethernet Encoder is not used)
- Ethernet communication cable
- Ethernet hub (required when multiple robots are used)

Devices required only for system start-up

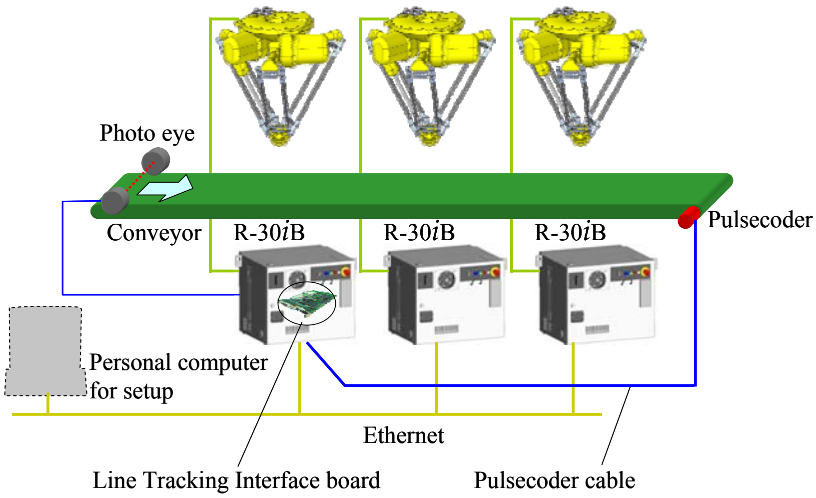
- Personal computer for setup and Ethernet communication cable for the personal computer
- Pointer
- Fixture which has a target that the robot can touch up accurately with the pointer

Sample configuration

When a pulse multiplexer is used



When the Ethernet Encoder is used



2.2 KEY CONCEPTS

This section describes terms used in this manual.

Line, Work area, and Sensor task

The following three concepts are used when a system configuration is considered to build a visual tracking system:

Work area

Area on a conveyor in which a robot performs a task, or data that represents it. The robot picks up a workpiece from a work area and puts it in another work area.

Line

Array data in which ordered work areas are arranged. A Line indicates how a workpiece on a conveyor sequentially flows from a work area to another work area. A Line represents a logical conveyor.

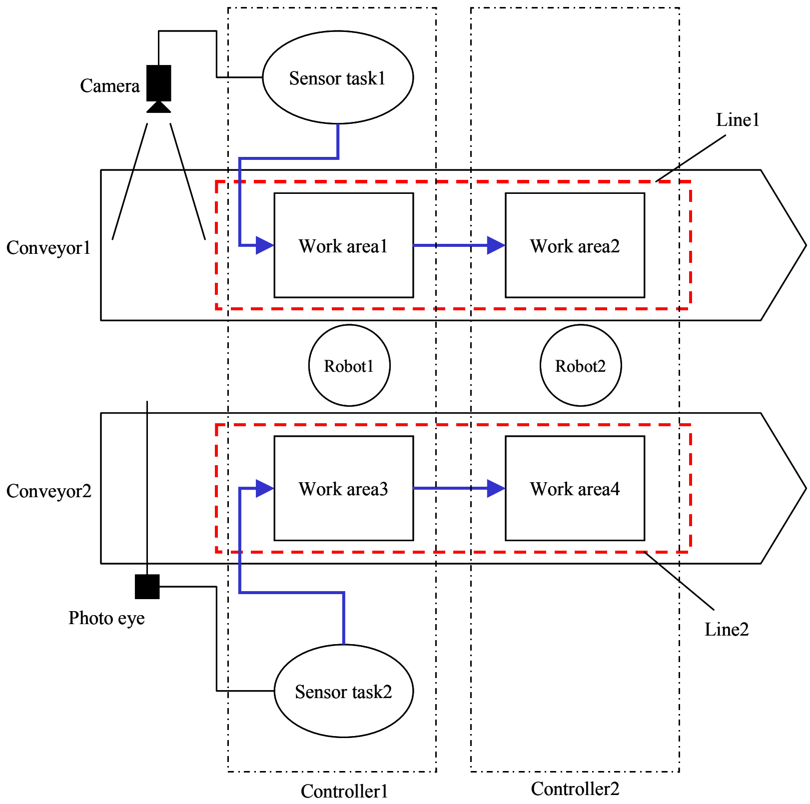
Sensor task

Robot program to detect workpieces on a conveyor and input this data to the most upstream work area defined by a Line, which is written in KAREL. It is available both for detecting workpieces using a vision system and for detecting them using a sensor without using a vision system.

> **NOTE**
> In most cases, you can use the standard KAREL program provided for sensor tasks. However, if needed, you can customize it. The source code of the KAREL program is not provided as standard. If you need to customize it, contact FANUC.

An example is given below, which will help you understand work areas, Lines, and sensor tasks more specifically.

The following figure shows a visual tracking application consisting of two robots and two conveyors. Conveyor 1 is used as an infeed conveyor and conveyor 2 is used as an outfeed conveyor. Both robots 1 and 2 pick up workpieces from conveyor 1 and put them on conveyor 2. One camera is installed at the upstream end of conveyor 1, which is used to detect each flowing workpiece. One sensor is installed at the upstream end of conveyor 2, which is used to detect each flowing box. This system picks up workpieces from conveyor 1 and places them in boxes flowing on conveyor 2.



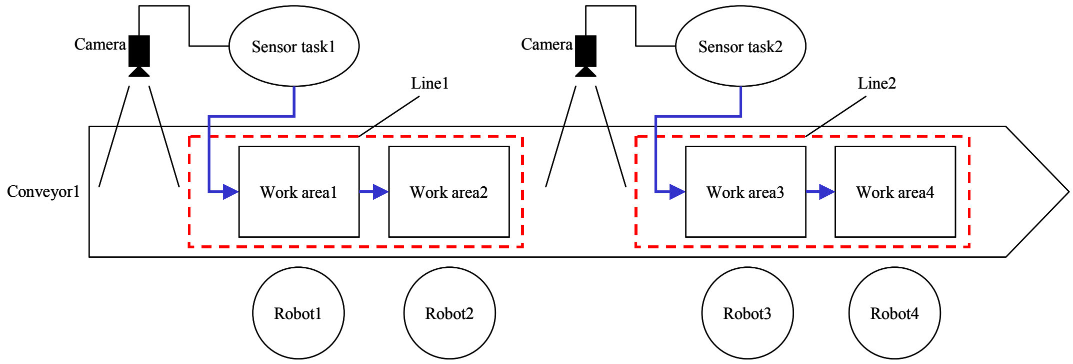
This system has four work areas, work areas 1 to 4. Work areas 1 and 2 are defined on conveyor 1 and work areas 3 and 4 are defined on conveyor 2. Robot 1 picks up workpieces from work area 1 on conveyor 1 and puts them in work area 3 on conveyor 2. Robot 2 picks up workpieces from work area 2 on conveyor 1 and puts them in work area 4 on conveyor 2.

This system has two Lines, Lines 1 and 2. Line 1 is defined on conveyor 1 and two work areas, work area 1 and work area 2, are defined in Line 1 in this order. Line 2 is defined on conveyor 2 and two work areas, work area 3 and work area 4, are defined in Line 2 in this order.

This system has two sensor tasks, sensor tasks 1 and 2.
Sensor task 1 detects a workpiece on conveyor 1 using a camera and inputs the information of the detected workpiece to Line 1. Sensor task 2 detects a workpiece on conveyor 2 using a sensor and inputs the information of the detected workpiece to Line 2.
The information of the workpiece sensor task 1 inputs to Line 1 flows from work area 1 to work area 2 as the workpiece flows. Similarly, the information of the workpiece sensor task 2 inputs to Line 2 flows from work area 3 to work area 4.

Each robot receives the information of the flowing workpieces from the work areas and performs work. For example, they pick a workpiece or put into a box. If a workpiece does not flow to a work area, its information is not passed to the robot. If a workpiece passes through a work area, its information is sent to the next work area. The information of the flowing workpieces is given to each robot based on information such as the position of each flowing workpiece and predetermined allocation ratio.

Generally, one Line is defined for one conveyor. If two sensors are arranged on the same conveyor in series, two Lines may be defined on one conveyor (figure below). In this case, two sensor tasks are also defined for one conveyor.



Tray pattern, Cell, and Tray frame

In the sample system described above, “boxes” flow on the outfeed conveyor and the robots place workpieces picked up from the infeed conveyor into boxes. Generally, multiple workpieces are placed in a “box”. For *i*RVision visual tracking, this “box” is called a Tray.

Tray pattern

A Tray pattern is data, which defines a “box”. That is, items such as how many workpieces are to be put in a Tray or picked from a Tray, how the workpieces are arranged in the box, and the order in which workpieces are put or picked are specified.

Cell

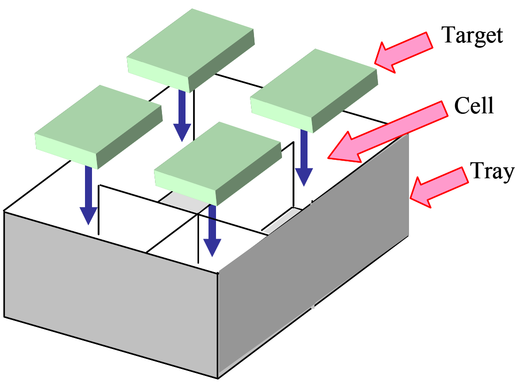
Each place where a workpiece is placed or picked is called a cell. For example, a box in which four workpieces are laid out is defined by a Tray pattern having four cells.

Tray frame

A tray frame is a frame used to define a cell layout in a Tray pattern. The position and orientation of each cell is set as coordinates in the tray frame. A tray frame can be defined in any position in the Tray.

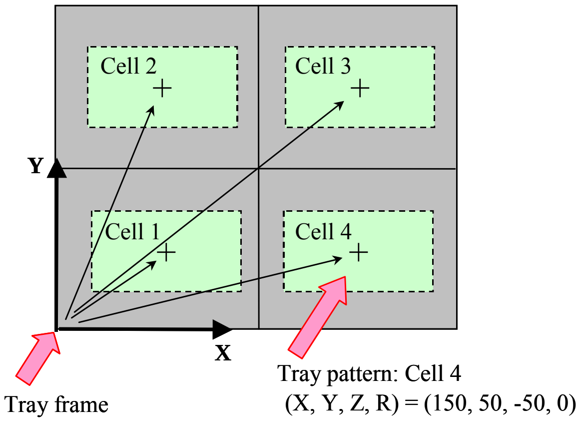
Defining a Tray pattern and teaching the robots the position of only one cell in the Tray pattern can make the robots place or pick workpieces in other cells. Since the cell in which a robot places or picks a workpiece is managed, multiple robots can easily be made to share the work of handling workpieces in a box.

An example of a Tray having four cells is given below, which will help you understand a Tray pattern more specifically.



In the example shown in the above figure, four workpieces can be placed in a box. The box itself is called a “Tray” and each place at which to place a workpiece in the box is called a “cell”. That is, this Tray has four cells.

The following figure shows the Tray pattern representing this Tray. In this figure, the above Tray is viewed from the top. A tray frame is defined at the lower-left corner of the box.



The + mark indicates the center of each cell. The position of each cell is set as the relative position viewed from the tray frame.

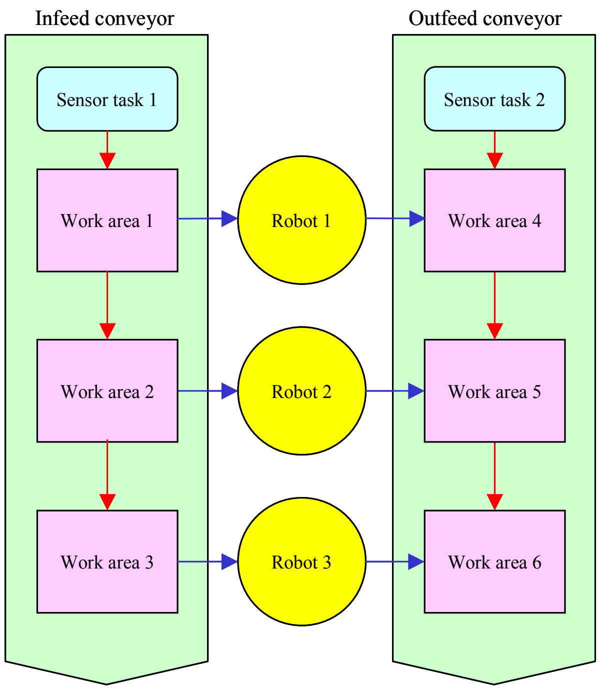
The tray frame can be determined in any position in the Tray without restrictions. To teach the robot the position of the tray frame, however, determine the position where the touch-up operation can be performed easily because the origin of the tray frame and a point along the X-axis must be touched up to set the TCP of each robot.

2.3 SAMPLE SYSTEM CONFIGURATIONS

This section introduces some sample system configurations to help you understand *i*RVision visual tracking system configurations.

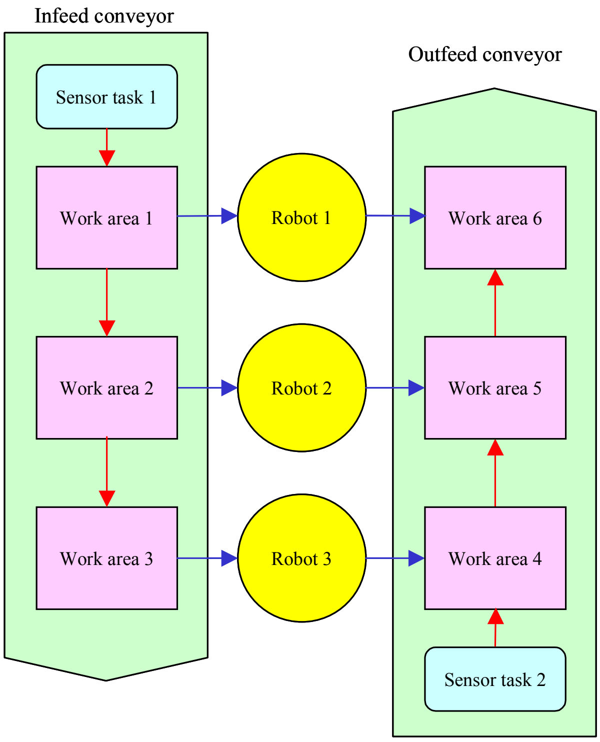
Sample configuration 1

This is the most basic configuration. Each robot picks up workpieces flowing on the infeed conveyor and places them on the outfeed conveyor.



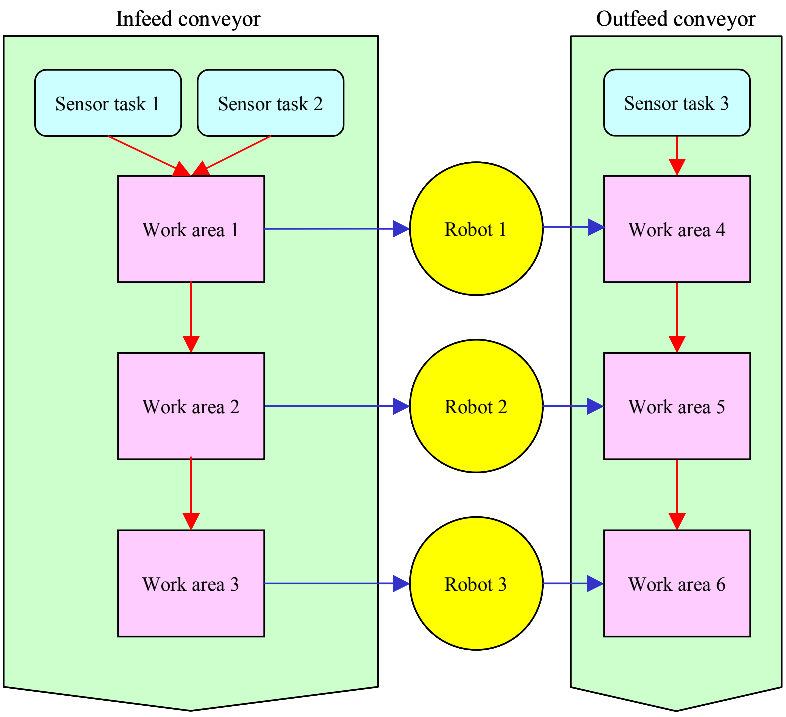
Sample configuration 2

This sample is basically the same as sample configuration 1. The flow direction is different between the infeed and outfeed conveyors.



Sample configuration 3

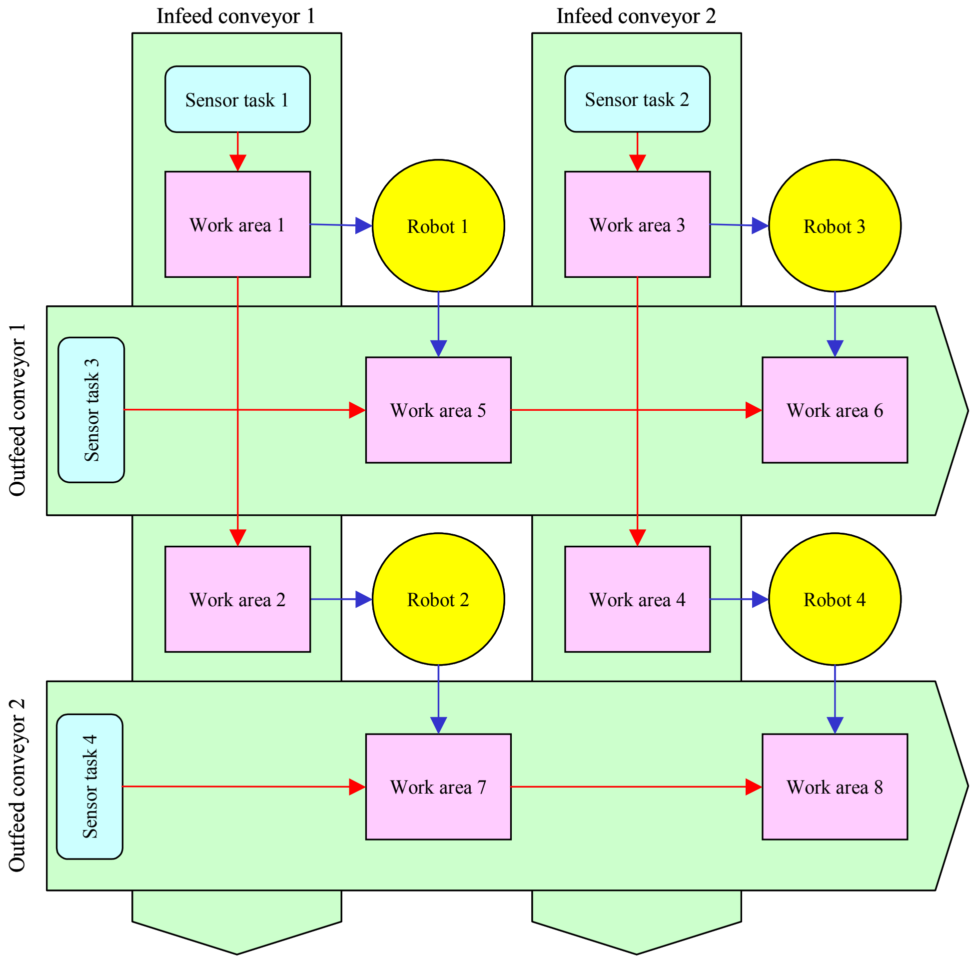
This sample is also basically the same as sample configuration 1. In this sample, conveyor 1 is wide and cannot be covered with one camera. In this case, install two cameras on the infeed conveyor, one covers the right half and the other covers the left half, and configure the system so that two sensor tasks input the information of the flowing workpieces to the same work area 1.



Sample configuration 4

This system has a little more complicated configuration, in which two input and outfeed conveyors are installed and arranged crossways.

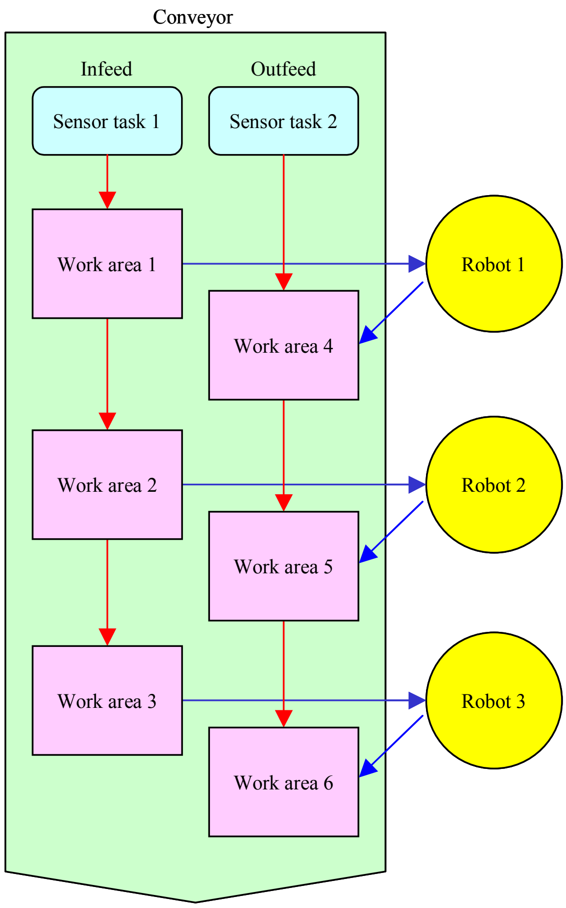
This system is seemingly complicated, but when you focus on operation of each robot, you can see that robot operation itself is almost the same as that in the sample configuration 1. For example, robot 1 picks up workpieces from work area 1 and places them in work area 5.



Sample configuration 5

In this system, the infeed and outfeed conveyors are physically the same conveyor.

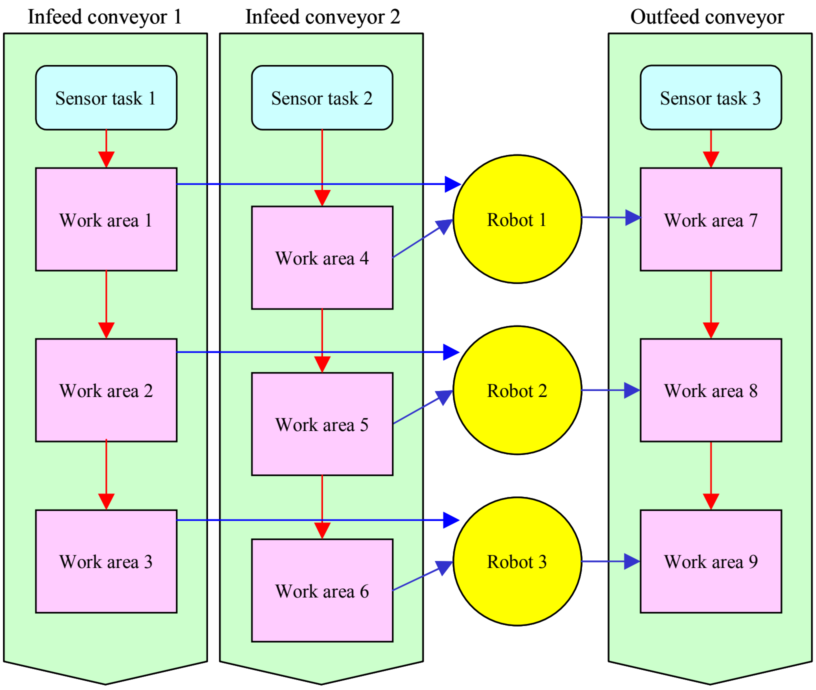
This configuration applies to the case where workpieces disorderly flowing on a conveyor are rearranged on the same conveyor. Making sensor task 1 detect workpieces on the conveyor using a camera and sensor task 2 detect markings made at equal spaces on the conveyor using a sensor such as a photo eye can arrange disorderly flowing workpieces on the conveyor at equal spaces.



Sample configuration 6

Two infeed conveyors are installed. Each robot picks up a workpiece from either infeed conveyor and places it on the outfeed conveyor. This configuration applies to the case where there are two types of workpieces, workpieces of type A flow on the infeed conveyor 1, workpieces of type B flow on the infeed conveyor 2, and workpieces of types A and B are mixed in a box flowing on the outfeed conveyor.

To configure this system, the robot program is complex. An example of the actual program is introduced in Chapter 5, “VARIATION”.



2.4 STUDY FOR APPLICATION

This section describes notes and others on considering the system configuration, layout, and so on.

System layout

Design a layout of robots, conveyors, pallets, and other devices to minimize the distance moved by the robot. This becomes important when the distance from the workpiece pickup location to the workpiece placement location decreases. In this case, a shorter time is required for each back and forth motion.

Robot

Determine the type of robot, the number of robots, the type of gripper, and so forth by considering the required transport quantity, workpiece weight, and so forth.

The maximum robot speed depends on the type of robot. The distance moved by a robot and the gripper open/close time needs to be considered. A higher-speed robot can make more back-and-forth motions per minute over a specified distance, handling a larger transport quantity. In general, however, such a robot can bear less weight.

When a gripper that can hold multiple workpieces is used, the number of back-and-forth motions for placing workpieces is reduced, so that one robot can handle a larger transport quantity. On the other hand, the gripper becomes heavier. So, care is needed to ensure that the weight of the gripper plus workpieces does not exceed the transportable weight of the robot. Generally, the gripper becomes larger in this case, so that interference with peripheral equipment and robots can occur.

> ⚠ **CAUTION**
> In case of using a 4 or 5 axes robot, the robot and the conveyor should be installed such that the conveyor surface and the XY plane of the robot world frame are parallel to each other. For a 4 or 5 axes robot, the XY plane of the tracking frame, that means the conveyor surface, will be parallel to the XY plane of the robot world frame automatically if the tracking frame is set with following the method described in this manual. So, if the conveyor surface and the XY plane of the robot world frame are not parallel to each other, the robot tracking accuracy will lower. If the procedures described in this manual for a 4 or 5 axis robot are properly followed the yaw and pitch of the tracking frame will be 0.00 degrees.

Workpiece detection method

The sensor to be used to detect workpieces flowing on each conveyor needs to be studied. There are two main types of methods.

Detection using a sensor

A sensor is installed in the upstream area of the conveyor to detect a workpiece by checking whether it cuts across in front of the sensor. With a sensor, only the position in the conveyer moving direction can be recognized. The orientation and longitudinal conveyor position of flowing workpieces need to be constant.

Detection using a camera

A camera is installed in the upper area of the conveyor, an image on the conveyor is snapped, and workpieces in the image are detected with *i*RVision. A workpiece placed at any position in any orientation on the conveyor can be picked up correctly because the position and orientation are recognized with the vision system.

When workpieces are detected using a camera, there are the following two methods: With one method, the system snaps an image using the camera for every conveyor movement over a certain distance and continues monitoring the conveyor. With the other method, the system snaps an image using the camera only when a workpiece causes an input from a trigger sensor to come on.

> ⚠ **CAUTION**
> When using multiple cameras with multiple robots, it will reduce the image processing load if the cameras are connected to different controllers.

Distribution of pulse count

When multiple robots are used, the method to distribute the pulse count of the Pulsecoder on the conveyor needs to be studied. There are two types of methods.

Distribution using a pulse multiplexer

The pulse count is input to each robot controller via a pulse multiplexer. The pulse count becomes equal in all robot controllers, accurate tracking is possible. However, the more line tracking hardware is required. The line tracking interface board is necessary to all robots. Also, multiplexer cables and Pulsecoder to multiplexer cables are necessary.

Distribution using the Ethernet Encoder

The pulse count is input to one robot controller that is connected to Pulsecoder. The other robot controllers get the pulse count via inter-controller communication. Compared to distribution using a multiplexer, you can reduce the cost of the hardware since it does not require the pulse multiplexer, multiplexer cables, Pulsecoder to multiplexer cables, and the line tracking interface board for slave controllers. However, when each robot controller recognizes the pulse count, some variation may occur due to the effect of communication. Typically this variation may be between -2.0 ms and +3.0 ms. For example, if the conveyor speed is 400mm/s, the variation of handling position will be between -0.8 mm and + 1.2mm

> ⚠ **CAUTION**
> 1 If you use the Ethernet Encoder, connect the Pulsecoder directly with the master controller of the robot ring in order to decrease the communication traffic.
> 2 If you use the Ethernet Encoder, connect the camera or sensor directly with the controller where the Pulsecoder is connected in order to obtain an accurate pulse count at the moment that workpieces are detected. Combined with the previous item, connect the camera or sensor directly with the master controller of the robot ring.
> 3 If you use multiple cameras, it is recommended to use the pulse multiplexer. If you do not connect two or more cameras in one robot controller as described in the previous caution, you cannot avoid connecting a camera to the slave controller of the robot ring in this case. As a result, the variation of handling position may be greater than mentioned above because the slave controller cannot obtain the accurate pulse count at the moment that workpieces are detected.

Conveyor speed

A transport quantity is determined by the conveyor speed and the density of workpieces on the conveyor. When the conveyor speed is 200 mm/s and workpieces are located at average intervals of 100 mm, the transport quantity is 120 workpieces/min. On the other hand, when the conveyor speed is 100 mm/s, but workpieces are located at intervals of 50 mm, the transport quantity is unchanged, namely, 120 workpieces/min. In this case, as the conveyor is slower, the robot tracking precision and vision detection precision are improved. This means that a slower conveyor offers better overall system performance.

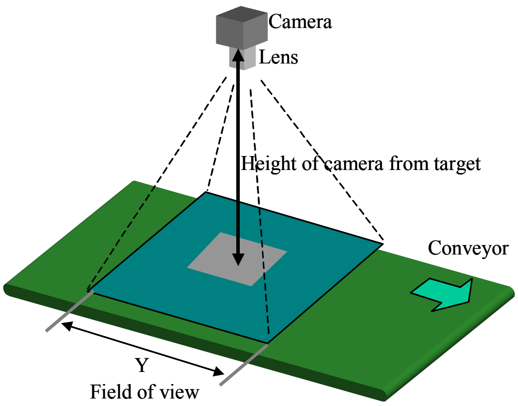
Field of view

When you use the camera to detect a workpiece, you have to consider the field of view. Field of view is determined by the kind of lens and camera and the height of camera from the workpiece. The expression to compute field of view is as follows:

$Y = y \times ((L - f) \div f)$

Y: Field of view
y: CCD size
L: Height of camera from workpiece
f: Focal length

The following figure is the physical relationship of the camera and workpiece.



CCD size is the size of CCD (Charge-Coupled Device).
CCD size = Cell size × Image size (pixels)

The following table shows the information of cameras that FANUC can provide as standard.
You can change the image size of digital cameras in the configuration menu. For details, refer to the section, "KOWA DIGITAL CAMERA" in the chapter "CAMERA SETUP" in the "*i*RVision OPERATOR'S MANUAL (Reference)".

Camera	Cell size	Image size	CCD size
Kowa digital camera (Grayscale camera)	6.700μm	1280 × 1024	8.576mm × 6.861mm
Kowa digital camera (Color camera)	6.400μm	1024 × 768	6.554mm × 4.915mm
Analog camera	7.400μm	640 × 480	4.736mm × 3.552mm

For example, when the focal length is 8mm, the distance from a camera to a workpiece is 1000mm, cell size is 6.7μm and the image size is 1280 × 1024pixels,, the field of view is computed by the following expression.

Horizontal direction: 8.576(mm) × ((1000(mm) – 8(mm)) ÷ 8(mm)) = 1063(mm)
Vertical direction: 6.861(mm) × ((1000(mm) – 8(mm)) ÷ 8(mm)) = 850.8(mm)

NOTE

It is desirable that the field of view is as small as possible. The smaller the field of view, the greater the resolution. Because the real-space area per pixel in that the image is projected is smaller if the field of view is smaller.

Exposure time

Study the proper camera exposure time when the camera is used to detect the workpiece. If the workpiece moves during exposure, the image displacement occurs. As a result, the detection accuracy lowers. The camera exposure time is determined to reduce the image displacement.
A movement of workpiece depends on the conveyor speed. Accordingly, the camera exposure time depends on the conveyor speed. As the conveyor speed is higher, the camera exposure time has to be shorter. On the other hand, as the conveyor speed is lower, the camera exposure time can be longer. However, exposure time is directly added to the image processing time. So, even when the conveyor moves slowly, a relatively short exposure time should be used.

An exposure time suitable for a conveyor speed is calculated using the expression below.

$E \leqq \angle \div V$

E : Exposure time (ms)
⊿ : Image displacement (mm)
V : Conveyor speed (mm/s)

For example, when the conveyor speed is 100 mm/s, the expected exposure time to make the image displacement 0.1 mm or less is computed by the following expression:

0.1(mm) ÷ 100(mm/s) = 1(ms)

Accordingly, set an exposure time of 1 ms or less.

As the exposure time is shorter, stronger lighting is needed. Prepare lighting to enable a sufficiently bright image to be picked up with the exposure time calculated above.

Vision system trigger interval

To detect workpieces on a conveyor using a camera, pay attention to the intervals at which to execute the vision process. The vision system needs to perform image processing at least one time for every conveyor movement over a certain distance so that workpieces on the conveyor are not missed. Accordingly, the vision system is highly affected by the conveyor speed. That is, as the conveyor moves faster, the image processing needs to be performed more frequently, resulting in an increased load on the vision system. On the contrary, if the conveyor moves slowly to enable multiple workpieces to be detected in one image processing period, the load on the vision system is reduced without changing the transport quantity.

The vision system trigger interval is roughly calculated using the following expression:

Start interval (s) = 0.5 × Camera field of view (mm) ÷ Conveyor speed (mm/s)

As the field of view is enlarged, the vision system trigger interval is increased. However, the detection precision is decreased as the field of view is enlarged.

Handling precision

Handling precision is affected by factors such as the tracking frame setting error, TCP setting error, robot position teaching error and robot tracking delay other than the error of the detection of vision system.

Furthermore, the handling precision may be degraded by pulsation occurring in the movement of the conveyor or by a large workpiece swinging on the conveyor.

By taking the following measures, however, handling precision can be improved:

- To set the TCP for touch-up correctly and accurately
- To detect many workpieces at a time
- To decrease the conveyor speed
- To shorten the exposure time
- To adjust tracking motion

Load balancing

When multiple robots share the work of handling workpieces from one conveyor, the quantity and type of workpieces each robot is to handle can be specified. This specification of “the type of workpiece the robot is to handle, and the quantity of workpieces the robot is to handle” is called load balancing.

The load balancing function can be used to specify the ratio of the quantity of workpieces each robot is to handle. A typical allocation method is “equal distribution”. For equal distribution, for example, when two robots are used, specify 1:1. If the robots have different processing capabilities, you may also specify 2:1.

When the load balancing function is disabled, each robot attempts to handle as many workpieces as it can. As a result, the maximum transport capability of the system is exerted. If small number of workpieces flow, however, the robot in the upstream area handles more workpieces and only few workpieces flow to the robot in the downstream area.

When workpieces are detected using a camera, the robot, which is to handle a type of workpiece in addition to the quantity of workpieces to be handled, can be specified. For example, the specification can be made so that robot 1 handles round workpieces and robot 2 handles triangle workpieces.

For allocation according to the type, model IDs of *i*RVision are used. Model IDs are typically used to determine types of workpieces. Using the *i*RVision conditional execution tool together with model IDs allows various types of allocation in addition to the determination of types of workpieces.

For example,

- Allocation based on the workpiece detection position (For example, robot 1 handles the workpieces on the right side of the conveyor and robot 2 handles the workpieces on the left side of the conveyor.)
- Allocation based on the workpiece orientation (For example, robot 1 handles the workpieces of which orientation is between 0 and 180 degrees and robot 2 handles the workpieces of which orientation is between -180 and 0 degrees.)

It is possible to combine these allocation methods. For specific allocation methods, see Chapter 5, “VARIATION”.

Controller, which executes a sensor task

In some systems introduced in Section 2.3, “SAMPLE SYSTEM CONFIGURATIONS”, multiple sensor tasks are used. In the most general system, one sensor task for the infeed conveyor and one sensor task for the outfeed conveyor, a total of two sensor tasks, is generally used.

As means for detecting workpieces on the conveyor with the sensor task, the following two main means are introduced: Detecting workpieces using a camera and detecting them using a sensor. No special consideration is needed for a sensor task using a sensor because the amount of processing for the sensor task is small. Since a sensor task using a camera performs image processing, a considerable processing load is applied on the robot controller.

For this reason, when two or more sensor tasks use the vision system, it is recommended that the system is configured so that the same robot controller does not process multiple sensor tasks. Specifically, configure the system so that two or more cameras are not connected to one robot controller.

2.5 BASIC START-UP PROCEDURE

Use the following procedure to start the *i*RVision visual tracking system:

1. Setting of a personal computer for setup
   Set up a personal computer used to set up the queue management.
2. Setting of robot ring
   When performing tracking with multiple robots, setup the robot ring. This step is unnecessary when only one robot is used.
3. Pulsecoder installation and setting
   Install and connect a Pulsecoder, then set the parameters of the Pulsecoder.
4. Camera installation and connecting
   Install and connect a camera. This step is unnecessary in the queue managed line tracking system.
5. Setting of the TCP for touch-up
   Set the TCP for touch-up used for a purpose such as frame setting. Do this work on each robot.
6. Setting of Lines and work areas
   Set items such as Lines and work areas related to the system configuration. Also, set the tracking frame.
7. Tray pattern setting
   When using a Tray, teach the Tray pattern. This step is unnecessary when no Tray is used.
8. Vision process teaching
   Perform camera calibration and vision process teaching. This step is unnecessary in the queue managed line tracking system.
9. Robot program teaching
   Create programs for robots, which perform tracking motion.
10. Sensor task teaching
    Sets sensor tasks.
11. Fine adjustment of tracking motion
    Make a fine tracking motion adjustment while performing a tracking motion actually.

2.6 RESTRICTIONS

As of V8.10P/04, the following restrictions are imposed on the *i*RVision visual tracking function and queue managed line tracking function:

1. The *i*RVision visual tracking system and the queue managed line tracking system support line tracking only. Rail tracking and circular tracking are not supported.
2. The *i*RVision visual tracking system and the queue managed line tracking system do not support the hot start mode. Please disable the hot start.

3 PREPARATION

This chapter describes hardware connection and other preparations for the start-up of a visual tracking system.

3.1.4 “Setting of the Robot Ring” describes the settings only needed when the system contains multiple robot controllers. When the system contains only one robot controller, these settings are unnecessary.
3.3 “PULSECODER INSTLATION AND CONNECTION” describes a procedure corresponding to the system configuration. This procedure differs according to whether the system contains a single robot controller or multiple controllers. In addition, this procedure differs according to whether using a pulse multiplexer or the Ethernet Encoder when the system contains multiple robot controllers.
3.4 “CAMERA INSTALLATION AND CONNECTION” and 3.6 “GRID PATTERN CALIBRATION” describe procedures only needed in the *i*RVision visual tracking system. These procedures are unnecessary in the queue managed line tracking system.
All other procedures are necessary in any case.

3.1 CREATE A NETWORK

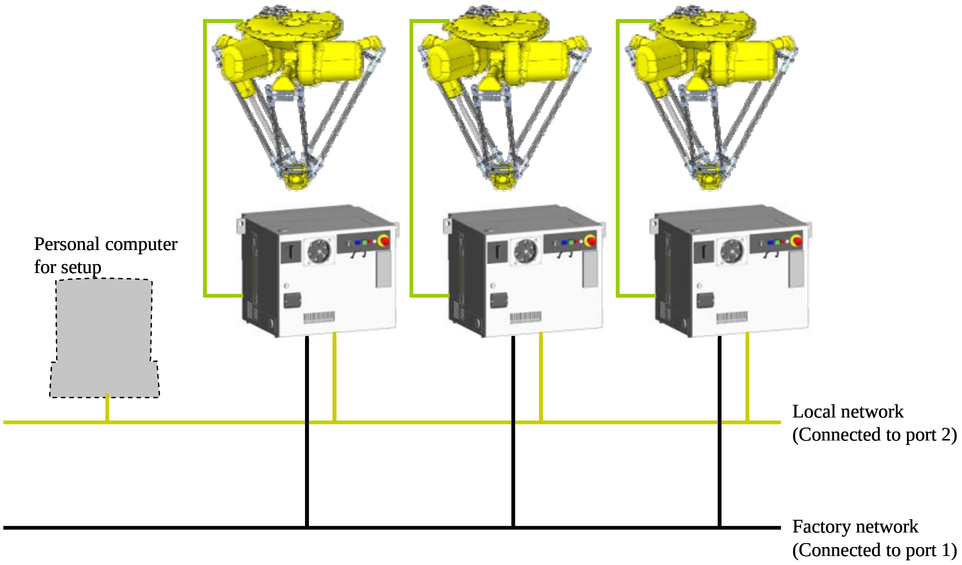
When multiple robot controllers are used, connect these controllers via a network.

Even when a single robot controller is used, this setup operation is required since a personal computer is connected to the network in order to set up the queue management.

3.1.1 Connecting a Network Cable to the Robot

In an *i*RVision visual tracking system, a robot controller communicates a lot of information to other robot controllers via a network. Therefore, if the transmission is delayed, it affects the system performance. Therefore, create a local network, which is configured with only robot controllers, so that the communication traffic reduces.

The robot controllers have two Ethernet ports. Accordingly, the robot controllers can belong to two different networks. If a robot controller needs to connect to an external network, for example which is a factory network, port 1 is used for the external network and port 2 is used for the robot controller network as shown in the figure below. The network using port 2 is connected by the robot controllers used for visual tracking and a personal computer to set up the queue management.



When port 1 is used, connect the Ethernet cable to the CD38A connecter on the MAIN board. When port 2 is used, connect the Ethernet cable to the CD38B connecter on the MAIN board.

NOTE

1. If a robot controller is not connected to an external network, you can use whichever port you prefer.
2. A personal computer to set up the queue management is connected to the local network to communicate among robot controllers.

3.1.2 Caution on Setting up a Network

When a network is set up, set up each step below while noting the following cautions.

- As mentioned in “3.1.1 Connecting a Network Cable to the Robot”, there are two ports in one controller. With reference to the “3.1.3 Setting IP Address”, set the IP address and the subnet mask of the port used.
- If you want to use both ports 1 and 2, you should set the different network address to each port. In detail, refer to the "3.1.3 Setting IP Address".

The followings are restrictions when there is more than one robot controller in the system. For a system with only one robot, please skip.

- When the IP address is set, each robot controller needs different name.
- When the IP address is set, the IP Addresses must be continuous. For example, if there are four robot controllers and port 2 is used, the IP address of port 2 of each robot controller is set as 172.16.0.1, 172.16.0.2, 172.16.0.3, and 172.16.0.4.
- If the Ethernet Encoder is used, the IP address of the robot controller connected with the Pulsecoder must be the first/lowest number. For the previous example, the IP address of port 2 of the robot controller connected with the Pulsecoder is set as 172.16.0.1.

3.1.3 Setting IP Address

Set up the IP address for each robot controller by following this procedure:

1. Press [MENUS] on the teach pendant, then select [6 SETUP].
2. Press F1, [TYPE], then select [Host Comm].

3 Select [TCP/IP], and press [ENTER] button.
4 Input this robot controller name to [Robot name].

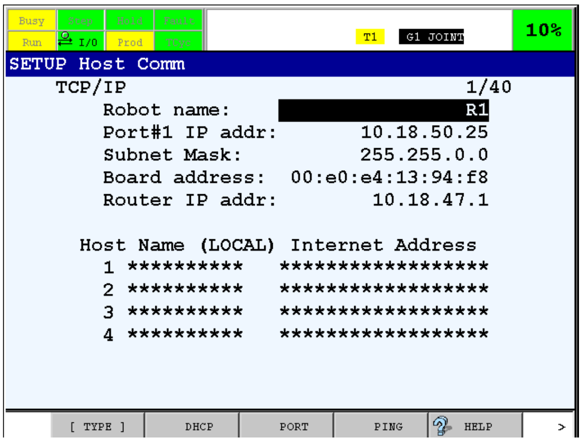
> ⚠ **CAUTION**
> Only alpha numeric characters and the minus sign can be used for robot name. Only an alphabet can be used as the first character. The minus sign cannot be used as the last character. The robot name cannot contain any spaces.

5 Input the IP address for this robot controller to [Port#1 IP addr].

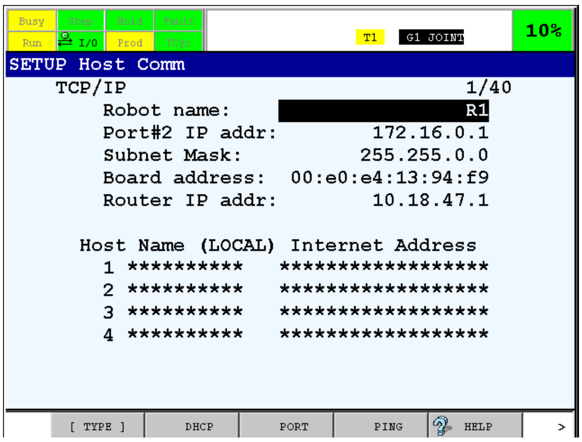
> ⚠ **CAUTION**
> You should not insert spaces or unnecessary "0" to the IP address. If there are spaces or unnecessary "0" in the IP address, the communication will fail.

6 Input the subnet mask to [Subnet Mask].
7 Input the IP address of router to [Router IP addr].
8 Press F3, [PORT], and select the port.
9 Set IP address and subnet mask by the same way as the procedure from 7 to 8.
10 Turn off the power to the robot controller then turn on the power to the robot again.

> ⚠ **CAUTION**
> If you don't cycle power of the robot controller, following setup is not performed correctly. Cycle power of the robot controller necessarily.



Setting example of Port 1



Setting example of Port 2

> ⚠ **CAUTION**
> If you use both port 1 and port 2, they need to be set to the different network address. For example, when subnet mask is 255.255.0.0 and IP address is 172.16.0.1, network address is 172.16 and host address is 0.1. If port 1 and port 2 are set to the IP address whose network address is same, the alarm “HOST-179 IP Address mis-configuration” is posted when the controller is turned on. Then, only port 1 becomes valid and port 2 becomes invalid.

3.1.4 Setting of the Robot Ring

When the system contains multiple controllers, set the robot ring. When the system contains only one robot, you do not need to set the robot ring. In this case, skip this subsection.

Robot ring internally uses the communication function, named as "ROS interface packet over Ethernet (RIPE)." In detail of RIPE, refer to “ROS INTERFACE PACKETS OVER ETHERNET (RIPE)” in the “Ethernet Function OPERATOR’S MANUAL” (B-82974EN).

Master and Slave

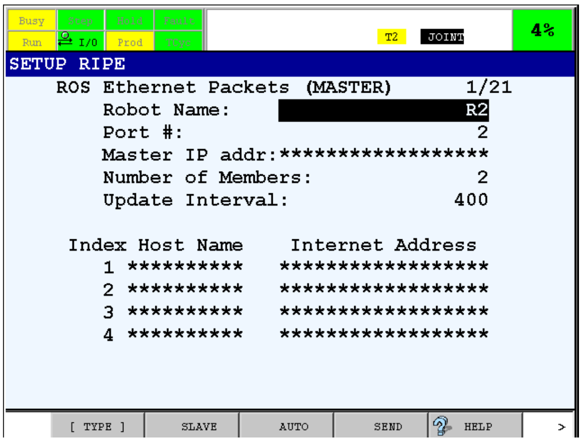
Robot ring consists of one master controller and the other slave controllers. In the visual tracking system, any robot controller can be the master controller. When the Ethernet Encoder function is used, it is recommended that the Pulsecoder is connected directly with the master controller for decreasing the communication traffic.

At first, set the slave controller when robot ring is set. Set the master controller after all the slave controllers are set. After you finish setting the master controller, the power of all the controllers is cycled automatically, and the setup is finished. The following is the detail procedure.

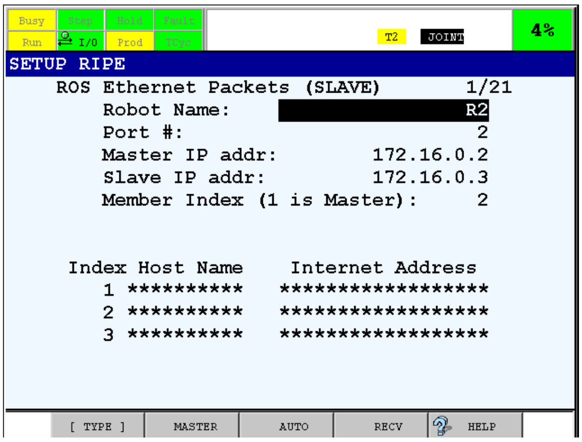
Slave Controller Setup

Set up the slave controller according to the following procedure.

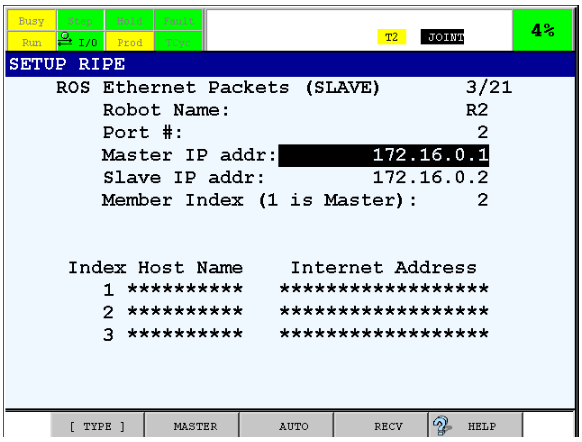
1 Press [MENUS] on the teach pendant, then select [6 SETUP].
2 Press F1 [TYPE], then select [Host Comm].
3 Select [RIPE], and press [ENTER] button.



4 If the above master controller screen is displayed, press F2 [SLAVE], and display the screen “ROS Ethernet Packet (SLAVE)” as the following. (If slave controller screen is already open and F2 button is MASTER, this procedure should be skipped.)

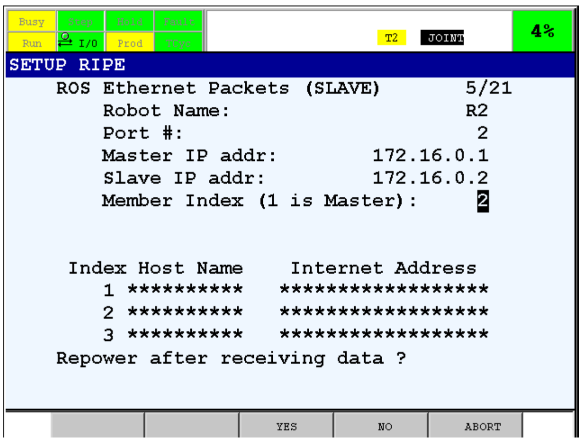


5 Set [Master IP addr] to the IP address of the master controller in the port used in the robot ring.

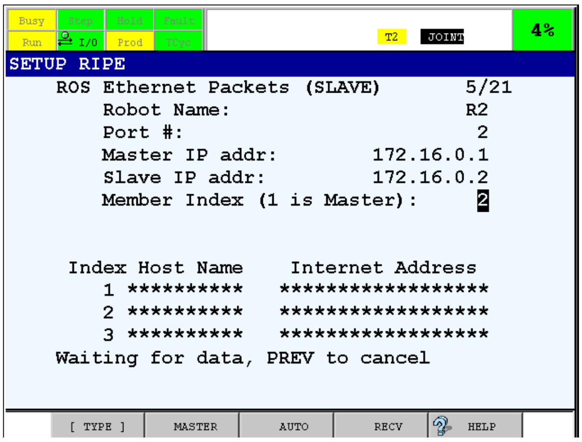


6 Set [Member Index]. In “3.1.3 Setting IP Address”, you already defined the IP address of the port used in the robot ring to the continuous number and also assigned the top number to the master controller. Input the IP address order of this robot controller from the top. According to this order setting, [Slave IP addr] is set automatically. Verify that the displayed IP address and the IP address which you assigned to this controller are same.

7 After finishing the above procedure, press F3 AUTO. The following screen is displayed. You will be asked whether this controller can cycle the power after receiving data, then press F3 YES.



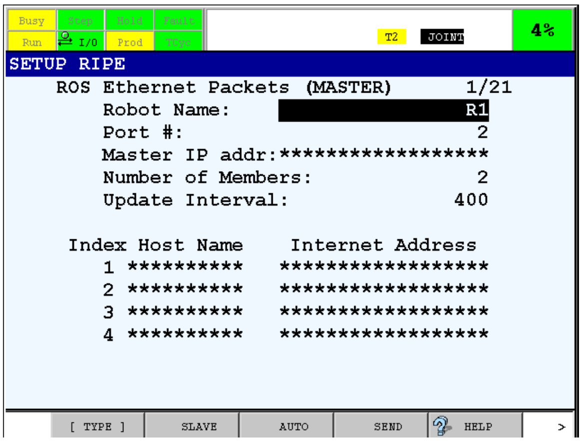
8 Perform the above procedure on all the slave controllers. The status of all the slave controllers should be “Waiting for data” from the master controller.



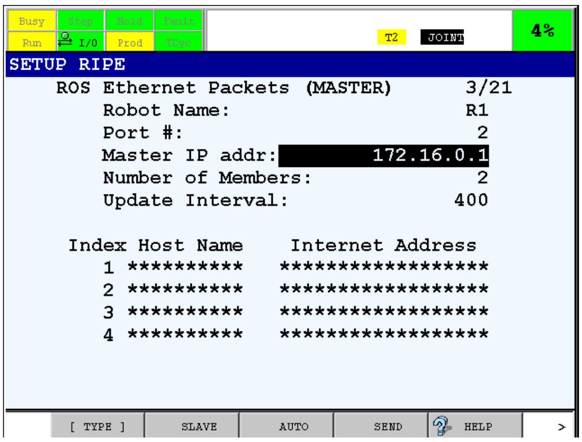
Master Controller Setup

After you finish setting all the slave controllers, set up the master controller according to the following procedure.

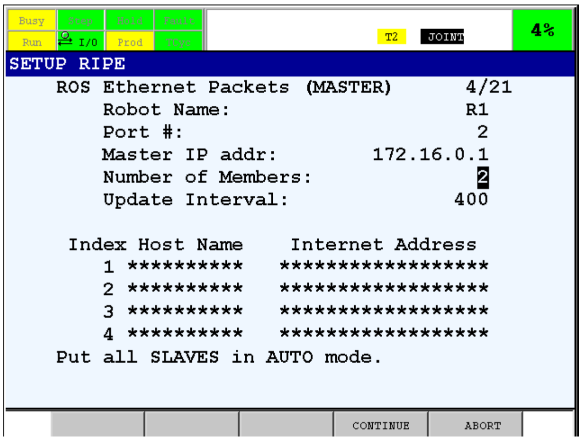
1 Press [MENUS] on the teach pendant, then select [6 SETUP].
2 Press F1 [TYPE], then select [Host Comm].
3 Select [RIPE], and press [ENTER] button.
4 If the screen “ROS Ethernet Packet (SLAVE)” is displayed, press F2 [MASTER], and change the following screen of the master controller. (If master controller screen is already open and F2 button is SLAVE, this procedure should be skipped.)



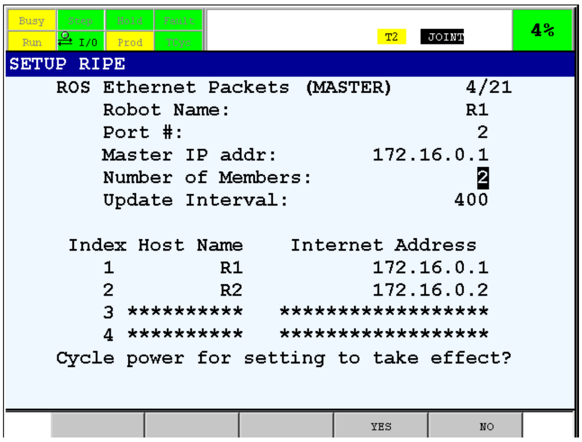
5 Set [Master IP addr] to the IP address of the master controller in the port used in the robot ring. This is the IP address of this controller.



6 Set [Number of Members] to the number of controllers which join the robot ring.
7 Set [Update Interval] if necessary. This is the interval to check whether the robot is online.
8 After finishing the above procedure, press F3 AUTO. The following screen is displayed.



9 The massage, “Put all SLAVES in AUTO mode.” is displayed. According to the previous “Slave Controller Setup”, because all the slave controllers are waiting for data from the master controller, press F4 CONTINUE. The following screen is displayed.



10 The robot names and IP addresses of all controllers which join the robot ring are displayed on [Host Name]. After you are asked to cycle the power of the controller in order to reflect the setting, press F4 CONTINUE. Then, all the controllers will cycle the power and whole the setting will be finished.

> ⚠ **CAUTION**
> Don’t change the robot name and the host name after you finish setting the robot ring. The robot name and the host name need to be same. If you want to change them, restart the setting of robot ring from the beginning.

3.2 SETTING OF A PERSONAL COMPUTER FOR SETUP

Set up a personal computer used to set up the queue management.

The personal computer is needed only to set up the queue management and can be disconnected during run time.

NOTE
A personal computer to set up the queue management is connected to the local network (not the factory network) to use the robot ring.

3.2.1 Setup PC

The tested PC and browser are Windows 7 (32bit) and Internet Explorer 9 (32bit).

⚠ **CAUTION**
1 The queue management supports only Japanese version and US version of Windows.
2 All Windows versions need the latest Service Pack.
3 Log into your PC as a user with the Administrator privilege.

3.2.2 Communication Cable

A cable is used to connect the robot controller and the PC to set up the queue management. Choose a 10BASE-T or 100BASE-T cable that meets the specifications as below.

Cable	Twisted pair
Shield	Shielded
Type	The R-30*i*B controller is auto detect, so a Cross over or Straight cable is appropriate.

3.2.3 Connecting a Communication Cable

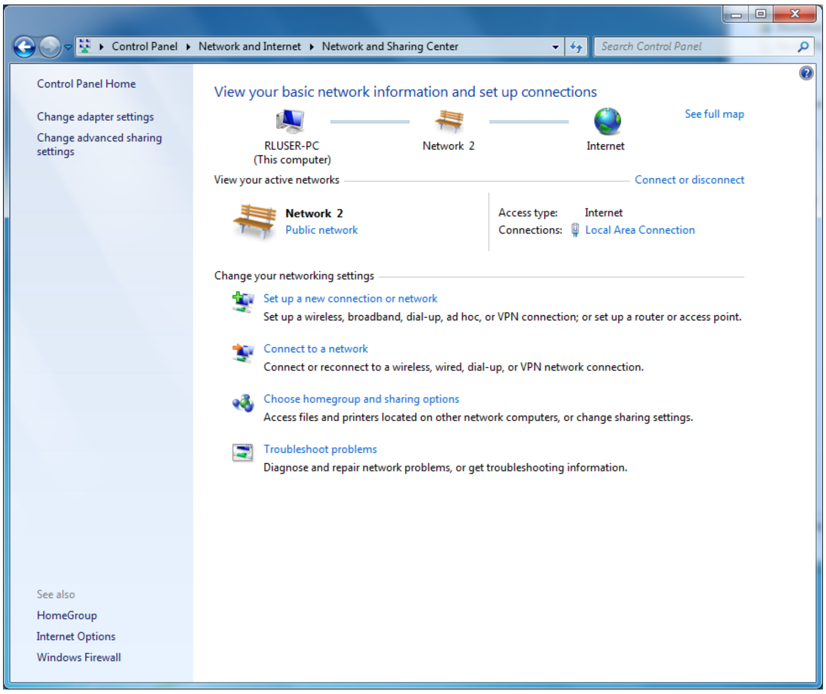
Connect the robot controller with the PC using an Ethernet cable. On the robot controller side, plug the cable into the Ethernet connector on the front of the MAIN board. On the PC side, plug the cable into the network connector, usually marked 品.

3.2.4 Setting the IP Address of the PC

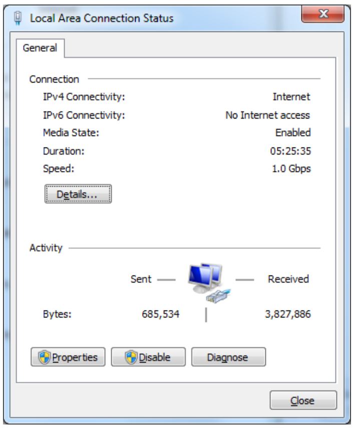
Set the IP address of the PC. For example, when three robot controllers and a PC are connected and not connected to any other network device, the IP addresses can be set as shown below.

Robot controller 1	172.16.0.1
Robot controller 2	172.16.0.2
Robot controller 3	172.16.0.3
PC	172.16.0.4
Gateway	172.16.0.5
Subnet mask	255.255.0.0

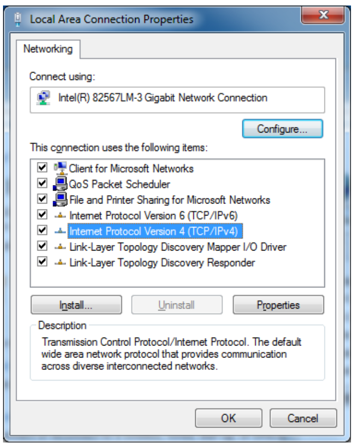
1 In the Control Panel window, open [Network and Sharing Center].



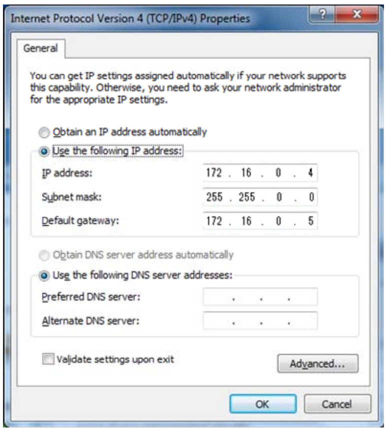
2 Click [Local Area Connections] in [View your active networks].



3 Click the [Properties] button.



4 Select [Internet Protocol Version 4 (TCP/IPv4)], and click the [Properties] button.

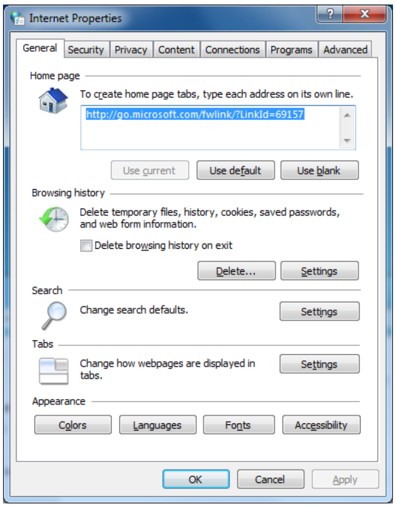


5 Check the [Use the following IP address] box, and enter values in [IP address], [Subnet mask], and [Default gateway].
6 Click the [OK] button to close the window.

3.2.5 Modifying Settings of Internet Explorer

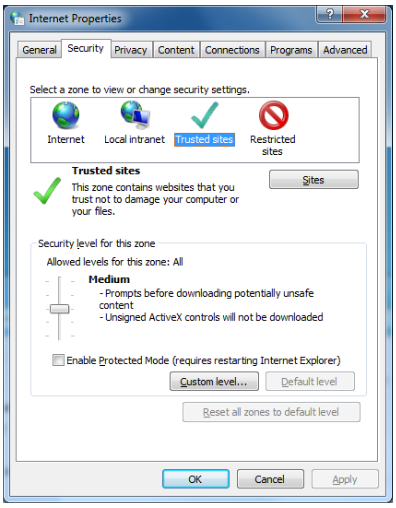
Set Internet Explorer to prevent Windows from blocking communication with the robot controller.

1 In the Control Panel window, open [Internet Options].

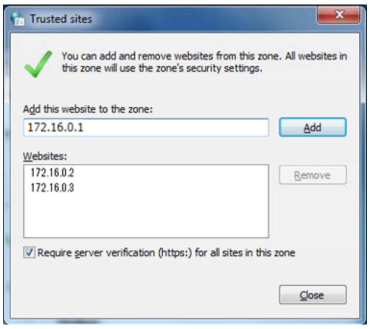


Trusted Sites

1 Select the [Security] tab.



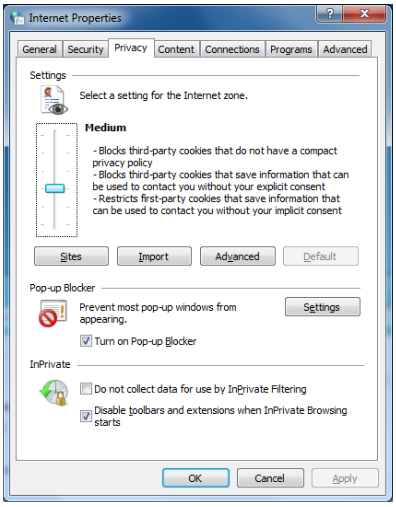
2 Select [Trusted Site], and then click the [Sites] button.



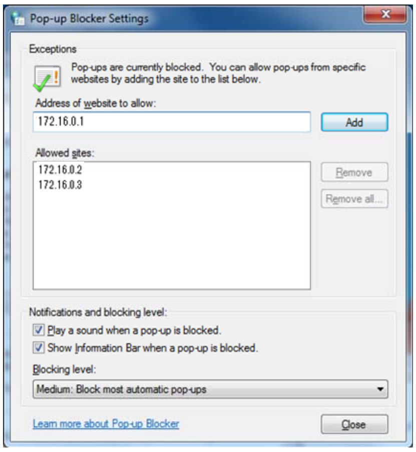
3 Uncheck the [Require server verification (https:) for all the sites in this zone] box.
4 In the [Add this Web site to the zone] textbox, enter the IP address of the robot controller (or the last digit of the IP address can be replaced by *). Then, click the [Add] button.
5 Click the [Close] button to close the dialog box.

Pop-up Blocker

1 Select the [Privacy] tab.



2 Click the [Settings] button of [Pop-up Blocker].

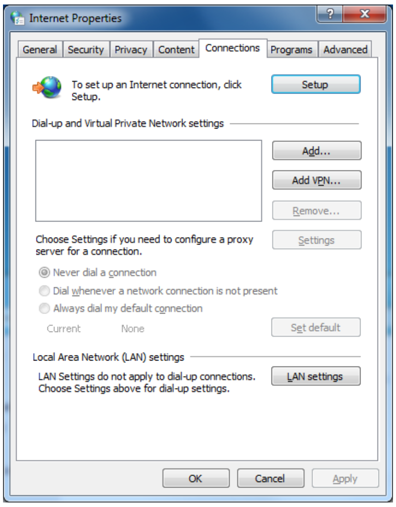


3 Enter the IP address of the robot controller in the [Address of Web site to allow] textbox, and click the [Add] button.

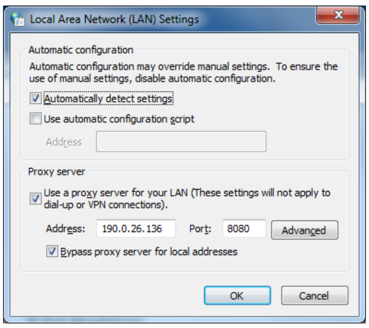
4 Click the [Close] button to close the dialog box.

Proxy Server

1 Select the [Connections] tab.

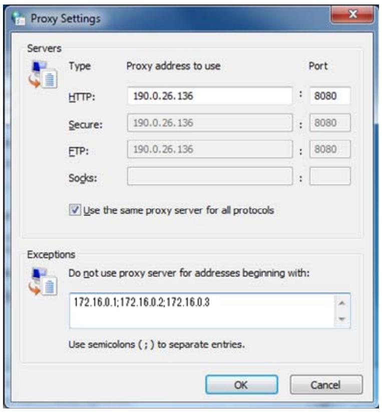


2 Click the [LAN Settings] button.



3 When the [Use a proxy server for your LAN] check box is not checked, proceed to the step 7. When it is checked, perform the steps 4 to 6.

4 Click the [Advanced…] button of [Proxy server].

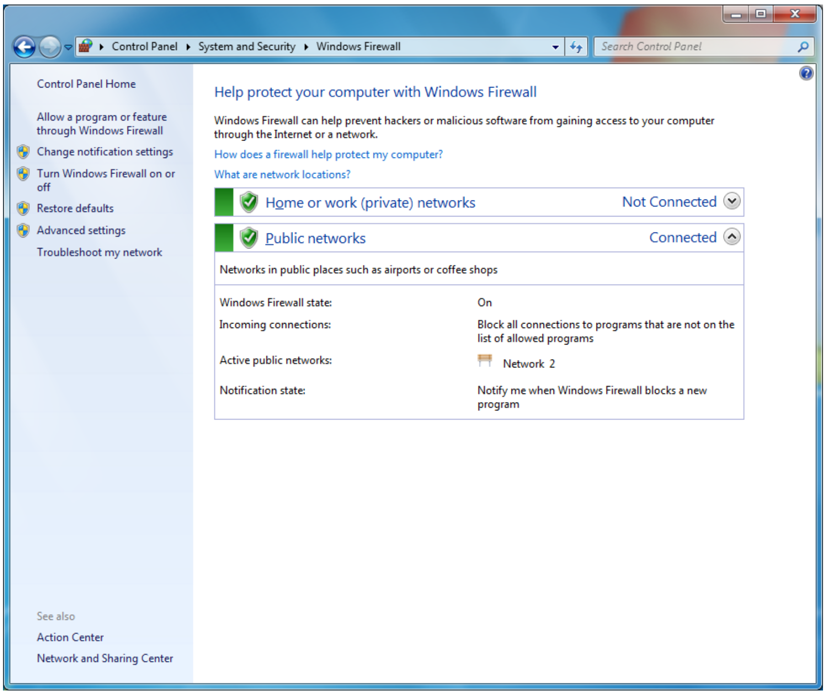


5 Enter the IP address of the robot controller in the text box under [Exceptions].
6 Click the [Close] button to close the dialog box.
7 Click the [OK] button to close the Internet property page.

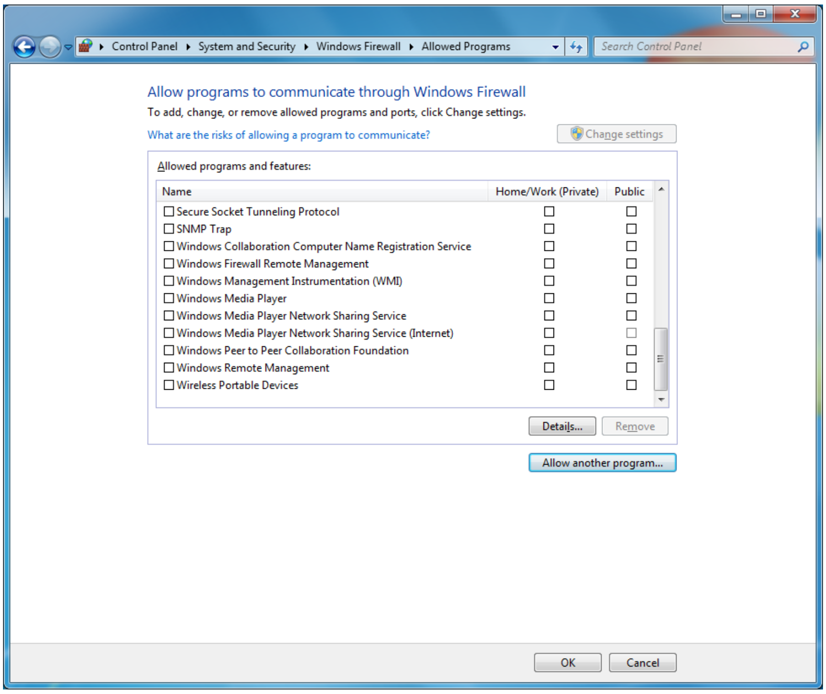
3.2.6 Modifying Setting of Windows Firewall

Modify the settings of Windows Firewall to prevent Windows Firewall from blocking communication with the robot controller.

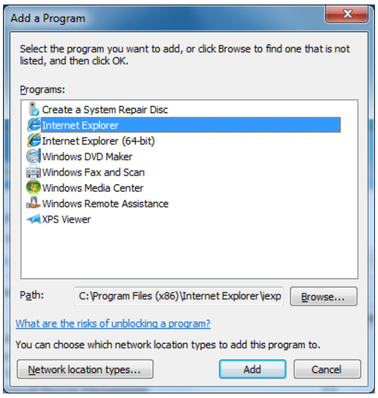
1 In the Control Panel window, open [Windows Firewall].



2 Click [Allow a program or feature through Windows Firewall].



3 Click the [Change settings] button.



4 Select [Internet Explorer] in the list, and click the [Add] button.
5 Click the [OK] button to close the window.

NOTE

Communication with the robot controller might be prevented due to a cause other than the above, which is, for example, a Microsoft® Internet Explorer add-on or security software installed in your PC. If an error occurs during setting up of the queue management, refer to Section 8.11, "PC UIF Troubles" first.

3.2.7 Opening Queue Management Pages

First, display the robot homepage by following the steps below.

1. Click the [Start] button on the PC screen, and start Internet Explorer.

2. Enter the IP address or the host name of the robot controller in [Address].

The robot homepage is not dedicated to *i*RVision Visual Tracking but also is provided for every robot controller. When the robot controller has the *i*RVision Visual Tracking option or the queue managed Line Tracking option, the following link for the queue management appears on the robot homepage.

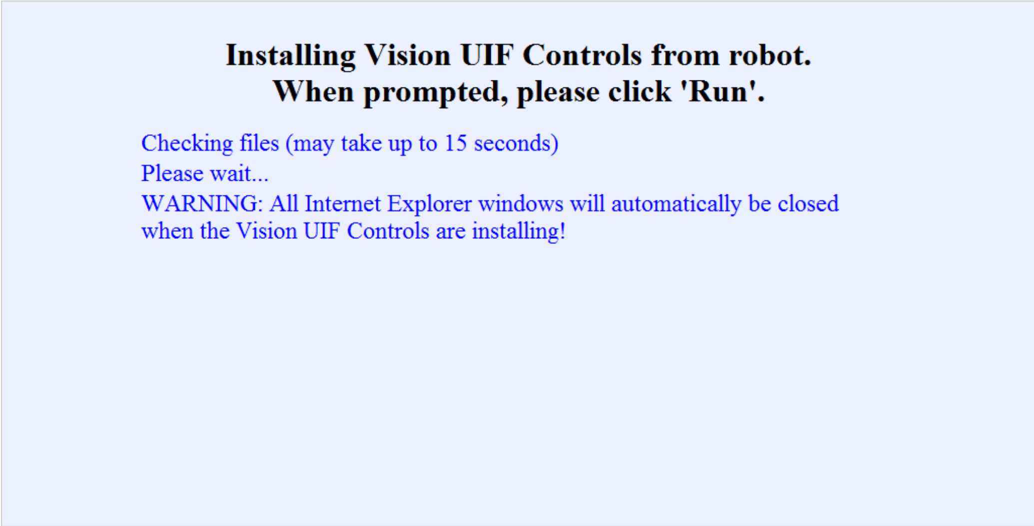
QUEUE MANAGEMENT

Displays the Line / Tray Pattern Setup page. In detail, refer to Section 4.1, “SETTING OF A LINE AND A TRAY PATTERN”.

Installing Vision UIF Controls

You must install Vision UIF Controls on your PC in order to display the queue management user interface. You can install Vision UIF Controls from the robot controller when you click a Line / Tray Pattern Setup link. Follow the steps below:

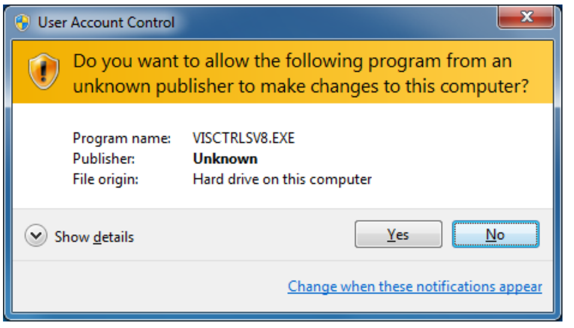
1 Click [Line / Tray Pattern Setup] in the queue management section.
If Vision UIF Controls are already installed in the PC used, the Line / Tray Pattern Setup Page opens.
If Vision UIF Controls are not installed in the PC, the following screen appears:



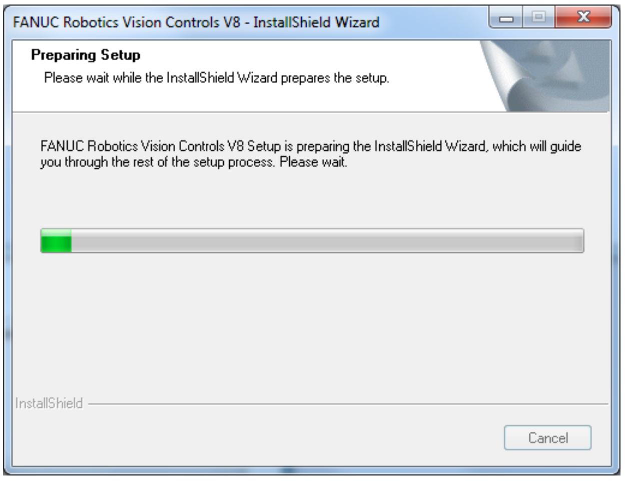
2 After a while, the following dialog appears.

3 Click the [Run] button.
4 After a while, the following dialog appears.

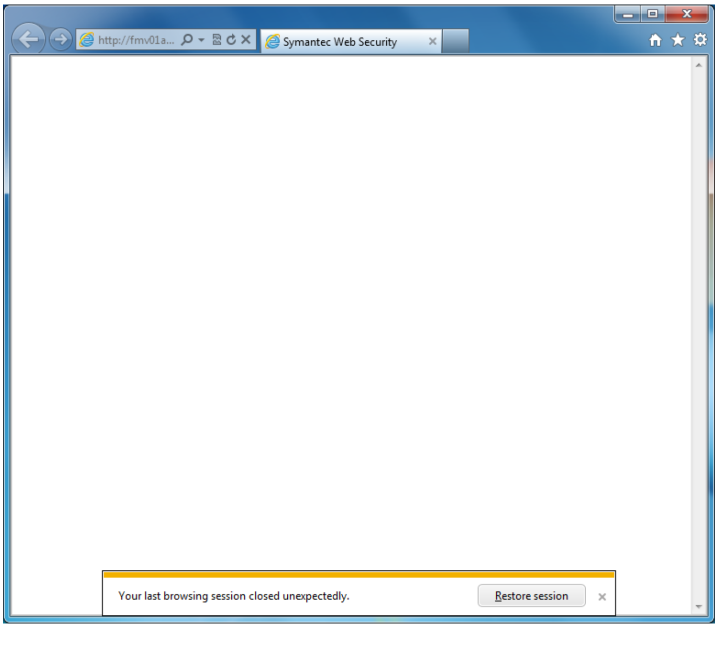
5 Click the [Run] button.
6 The following dialog box appears.



7 Click the [Yes] button.
8 Installation of Vision UIF Controls starts.



9 When the installation is completed, all Internet Explorer windows are closed.
10 Start Internet Explorer again, and open the homepage of the robot.
11 The following message will appear when you start Internet Explorer. Close the message by clicking the [X] button.



3.3 PULSECODER INSTALLATION AND CONNECTION

Install a Pulsecoder on the conveyor then connect the Pulsecoder to the robot controller.

> **NOTE**
> This manual refers to the device for detecting the amount of movement of the conveyor as “Pulsecoder”. In the teach pendant or the configuration screen of the queue management, the term "encoder" is used as the same meaning of "Pulsecoder".

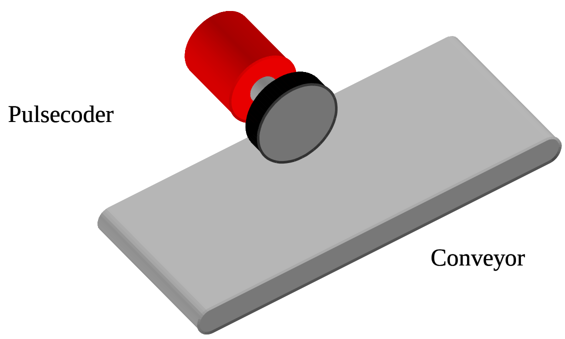
3.3.1 Installing a Pulsecoder

Install the Pulsecoder on the conveyor.

Install the Pulsecoder at a location outside the tracking area so that the Pulsecoder does not interfere with the robot during operation. When a camera is used to detect workpieces, install the Pulsecoder outside the field of view.

Attaching a rotary disk to the tip of the Pulsecoder so that the rotary disk rotates by directly touching the conveyor belt enables the move amount of the conveyor to be measured more accurately than attaching the Pulsecoder directly to the drive/follower axis of the conveyor.

Ensure that no slippage occurs between the rotary disk and conveyor belt. If a slippage occurs, the move amount of the conveyor cannot be measured accurately, resulting in degraded robot handling precision. Pressing the Pulsecoder against the conveyor using a spring is effective in securing the Pulsecoder.

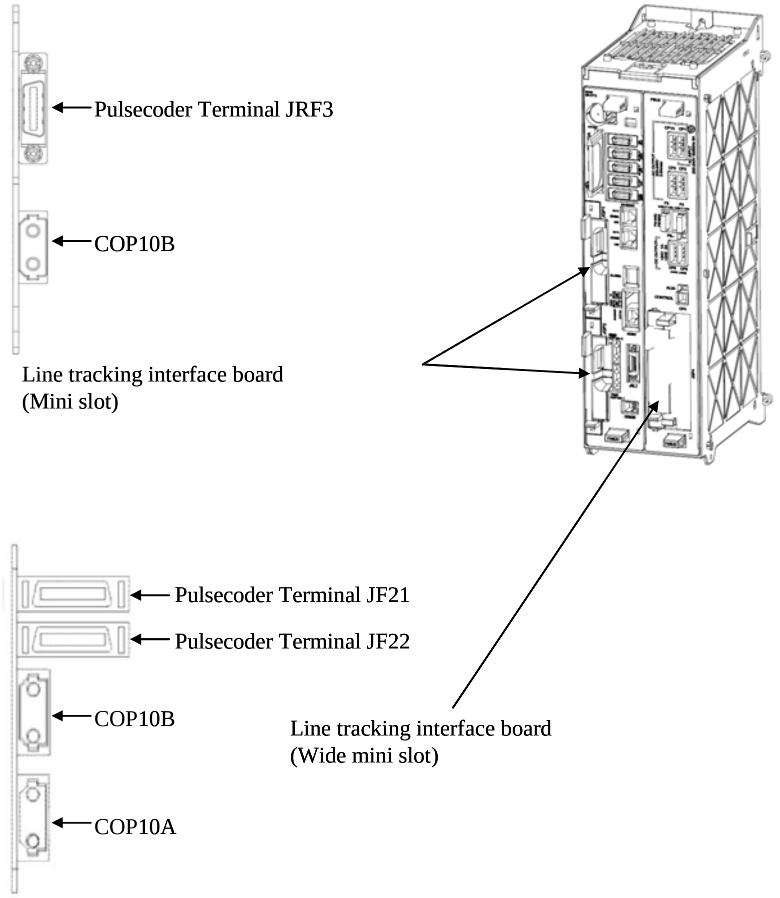


3.3.2 Connecting the Pulsecoder

Connect the Pulsecoder to the robot controller.

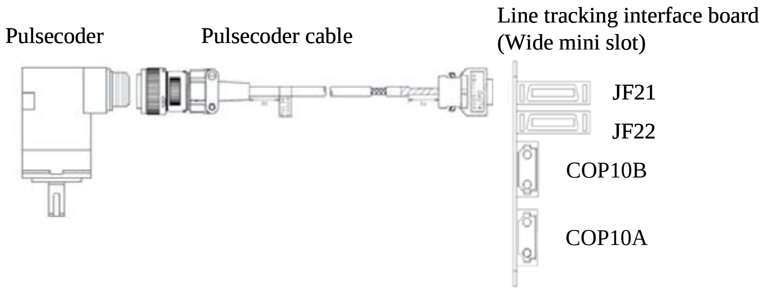
Line tracking interface board

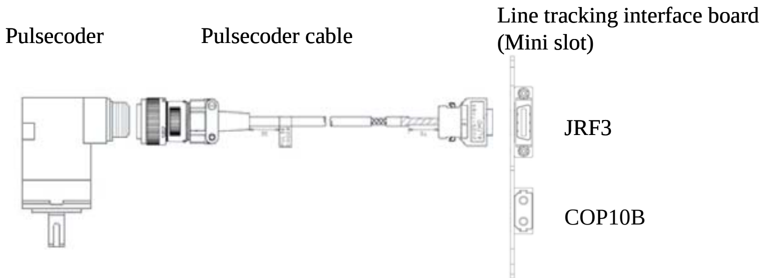
Line tracking interface board for wide mini slot or for mini slot can be selected.

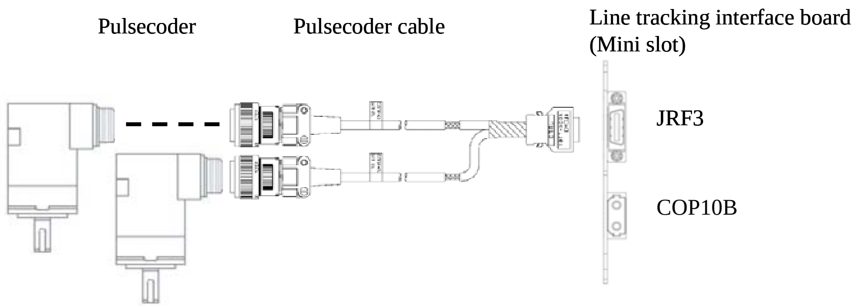

Pulsecoder Terminal JRF3
COP10B
Line tracking interface board
(Mini slot)
Pulsecoder Terminal JF21
Pulsecoder Terminal JF22
COP10B
COP10A
Line tracking interface board
(Wide mini slot)

When only one robot is used

Connect the Pulsecoder with the robot controller directly with a cable.





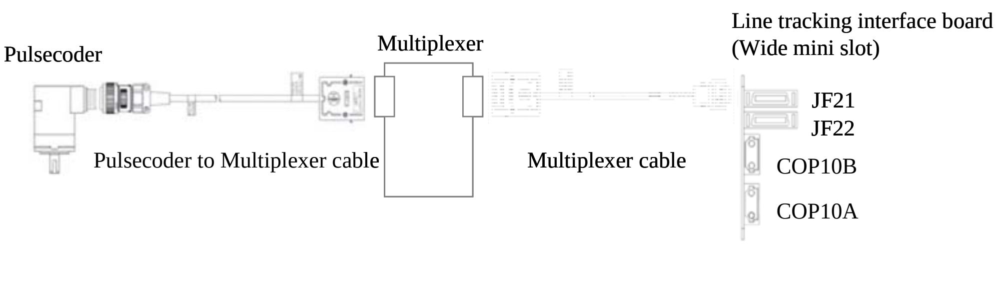


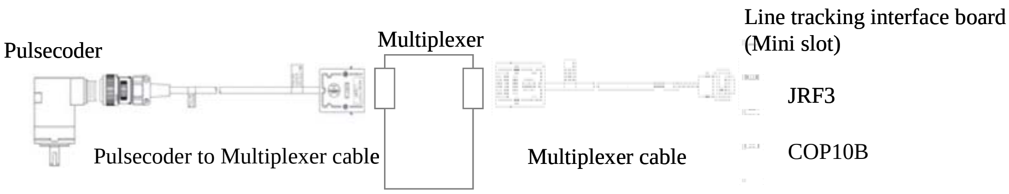
When using line tracking interface board for wide mini slot and only one Pulsecoder, connect the line tracking cable to the JF21 connector.

When multiple robots are used

When using pulse multiplexer

Connect the Pulsecoder to the robot controllers via a pulse multiplexer.





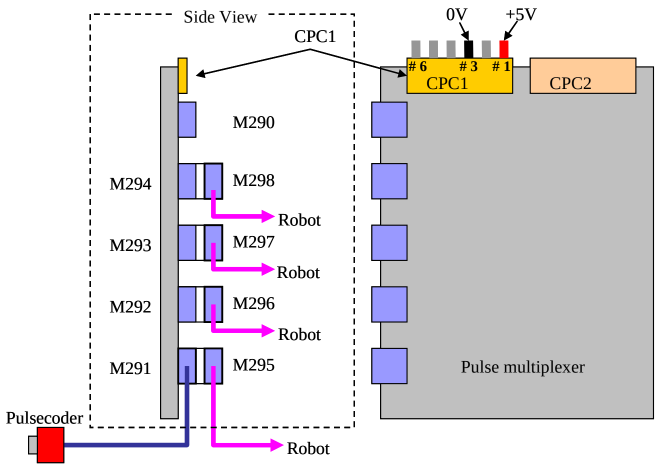
When using line tracking interface board for wide mini slot and only one Pulsecoder, connect the line tracking cable to the JF21 connector of each line tracking board. The method of connecting line tracking cables to the pulse multiplexer is detailed in the next subsection.

When using the Ethernet Encoder

As in the case that only one robot is used, connect the Pulsecoder to the robot controller. In order to reduce the load of the communication, connect the Pulsecoder to the master robot controller of the robot ring.

3.3.3 Connecting Cables to the Pulse Multiplexer

When using multiple robots and the pulse multiplexer, connect the line tracking cables and power cable to the pulse multiplexer as shown below.



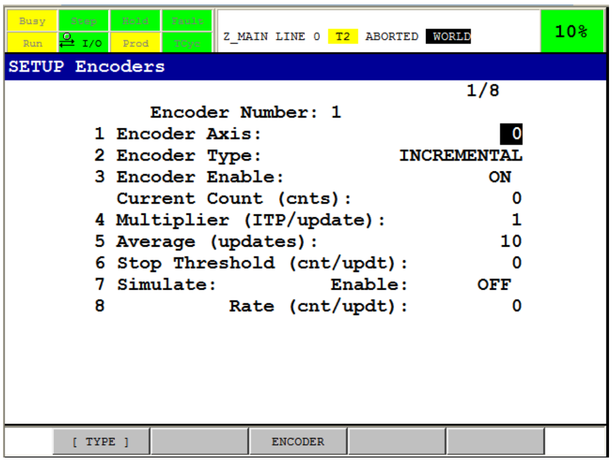
To one pulse multiplexer, up to four cables (pulse multiplexer ⇔ controller) can be connected.

3.3.4 Setting of the Pulsecoder

Set the parameters of the Pulsecoder as follows:

When only one robot is used or when multiple robots are used with the pulse multiplexer

1 Press MENUS on the robot teach pendant.
2 Select [6 SETUP].
3 Press F1 [TYPE].
4 Select [Encoders]. A menu as shown below is displayed.



5 To refer or set information about another encoder number, press F3 ENCODER then enter the desired encoder number. When using the line tracking interface board for wide mini slot, select 1 if you connect the encoder to JF21 and select 2 if you connect the encoder to JF22. When using the line tracking interface board for mini slot, select 1 if you use only one Pulsecoder. If you use two Pulsecoders, select the number that is printed on the cable tag.

6 Place the cursor on [1 Encoder Axis] then enter a desired encoder axis number. Generally, an encoder of incremental type is used. In this case, a value from 1 to 8 may be entered. Usually, enter 1.

> ⚠ **CAUTION**
> If the HDI signal is used on the conveyor using a sensor such as the photo eye, please select 1 as Encoder Axis

7 Place the cursor on [5 Average (updates)] then enter the number of sample counts used to calculate the current speed of the conveyor. Usually, enter 10.

> ⚠ **CAUTION**
> If the value of Average (updates) is small and the conveyor in use does not move smoothly, the robot also may not move smoothly because it follows the motion of the conveyor. In such a case, increase the [5 Average (updates)]

8 Please leave other setting items as default value.

9 If the encoder axis number is changed, the robot controller needs to be started in the cold start mode. In this case, turn off the power to the robot then turn on the power to the robot again.

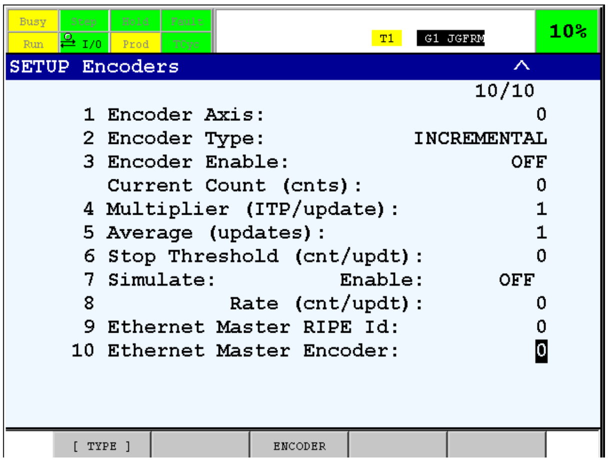
> ⚠ **CAUTION**
> The encoder counts of an incremental encoder are set to 0 when the power to the robot controller is turned on. In a system in which multiple robots work for one conveyor, all of the robot controllers must be powered on in the conveyor stopped state. If the robot controllers are powered on while the conveyor is moving, or if only one of the controllers is powered on again after the conveyor starts moving, the encoder counts for each robot controller become different, resulting in incorrect tracking.

When multiple robots are used with the Ethernet Encoder

Master Controller Setup

Use the following procedure for the robot that is connected to the encoder.

1 Follow the same procedure as steps 1 to 8 described in "When only one robot is used or when multiple robots are used with the pulse multiplexer." However, the screen of "SETUP Encoders" is different. The setup screen is shown below. When the Ethernet Encoder option is specified, the item is added that [9 Ethernet Master RIPE Id] and [10 Ethernet Master Encoder].



2 Place the cursor on [9 Ethernet Master RIPE Id] then enter the RIPE Id of this robot controller. RIPE Id means a unique id of each robot controller on a RIPE Ethernet network. It is set into the system variable $PH_ROSIP.$MY_INDEX. If this robot controller is the master of the RIPE network, RIPE_id ($PH_ROSIP.$MY_INDEX) is 1.

3 Place the cursor on [10 Ethernet Master Encoder] then enter the encoder number that is being selected. When using the line tracking interface board for wide mini slot, select 1 if you connect the encoder to JF21 and select 2 if you connect the encoder to JF22. When using the line tracking interface board for mini slot, select 1 if you use only one Pulsecoder. If you use two Pulsecoders, select the number that is printed on the cable tag.

4 Cycle the robot controller power.

Slave Controller Setup

1 Open the Encoders setup page. It does not matter which encoder number is selected, typically 1 is selected.

2 Enter the same value as entered at the master controller to [9 Ethernet Master RIPE Id] and [10 Ethernet Master Encoder].

3 There is no need to change any other items.

4 Cycle the robot controller power.

3.4 CAMERA INSTALLATION AND CONNECTION

Install a camera then connect the camera to the robot controller. This step is unnecessary in the queue managed line tracking system.

Selecting the Camera

The *i*RVision visual tracking system supports high resolution camera, color camera, and analog camera.

Color Camera

Please note the following tips when using a color camera.

- Use white strobe lighting. Adjust the light emission time of strobe lighting in order to avoid image blur.
- Not to be affected by ambient light, cover the camera and lighting by the light shielding plate. There are two purposes for this.

- To reduce the negative effect of ambient light since a color image is strongly affected by the light hitting to the workpiece.
- To avoid image blur of the moving part use strobe lighting,

For basic use of color cameras, refer to the subsection “Color Camera” in the section “KOWA DIGITAL CAMERA” in the chapter “CAMERA SETUP” in the “*i*RVision OPERATOR’S MANUAL (Reference)”.
In addition to the usual settings of color camera, set the following.

- Check [Use Strobe].
- Set [Def. Exposure Time] as the strobe light emitting time plus something extra. This something extra is the delay time of strobe light emitting but this needs not be exact.
- The strobe light emitting time should be tuned by the knob of the strobe light power supply.

Checking the Camera Setting

Change the setting of the camera unit to match *i*RVision.
For details, refer to the section “SETUP” in the “R-30*i*B CONTROLLER Sensor Mechanical Unit / Control Unit OPERATOR’S MANUAL”.

Connecting the Camera

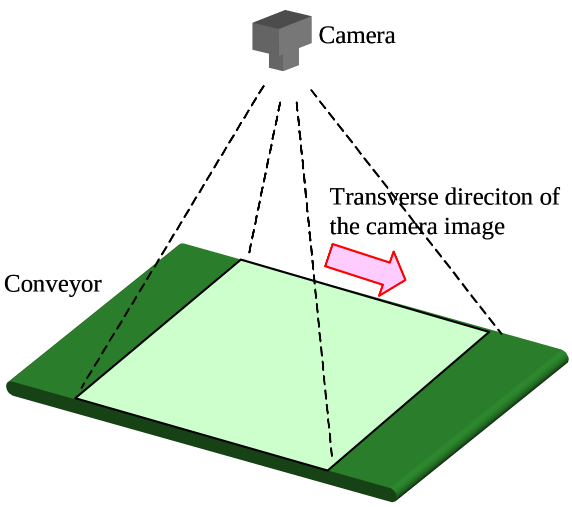
Connect the camera to the robot controller.
For details, refer to the section “SETUP” in the “R-30*i*B CONTROLLER Sensor Mechanical Unit / Control Unit OPERATOR’S MANUAL”.

> ⚠ **CAUTION**
> If you use the Ethernet Encoder, connect the camera directly with the master controller of the robot ring in order to obtain an accurate encoder count at the moment workpieces are detected.

Installing the Camera

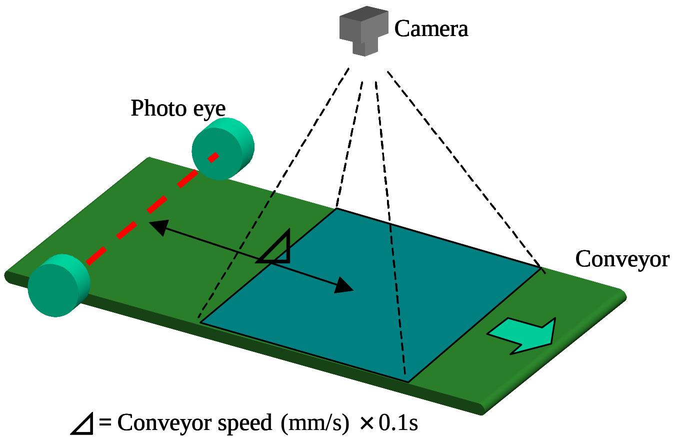
Attach the lens to the camera then install the camera over the conveyor.
Orient the optical axis of the camera so that the optical axis is approximately perpendicular to the conveyor plane.

Install a camera so that the direction in which the conveyor moves is the size larger direction of the camera image. The workpieces should move from left to right or right to left in the image to take advantage of the long side of the rectangular field of view.



Installing the camera and the sensor such as the photo eye

The sensor task watches a DI signal when the camera snaps an image after the input of a sensor such as a photo eye. When a DI signal is input from the sensor, the sensor task waits for the conveyor to move a specified distance then snaps. Depending on the hardware configuration there is approximately a 0.1 second delay to snap the image once the DI signal is turned on. It is recommended that you install the camera in the downstream direction from the photo eye by distance of the conveyor speed × 0.1. In this case, you can snap an image in the desired position of a workpiece by adjusting the "Trigger Offset" in the sensor task. Please refer to Section 4.4, "SETTING OF SENSOR TASK" for details on setting the "Trigger Offset".



3.5 SETTING OF THE TCP

Set the TCP for touch-up used for tracking frame setting and camera calibration. When multiple robots are used, this setting is required for all robots.

Secure a pointer to the wrist of a robot then set a user tool at the tip of the pointer. Assign a user tool data item to the pointer and make a setting so that the origin of the user tool matches the pointer tip. The setting methods include a 3-point teach method, 6-point teach method, and direct teach method. For a detailed description of operation, refer to the section "SETTING COORDINATE SYSTEMS" in the chapter "SETTING UP THE ROBOT SYSTEM" in the "R-30*i*B CONTROLLER OPERATOR'S MANUAL (Basic Operation)".

The pointer is used for tracking frame setting, camera calibration, sensor/tray position setting, and other purposes. After the completion of the required work, the point can be detached. After the pointer is detached, however, the set TCP is invalidated and a TCP needs to be set again unless the pointer is precisely restored mechanically. Accordingly, it is recommended that the mechanical design for the pointer is made to be restored to robot in preparation for requiring the pointer to touch up afterward. And it is recommended that the pointer be left mounted immediately before robot program position teaching described in Subsection4.5.2, "Robot Position Teaching".

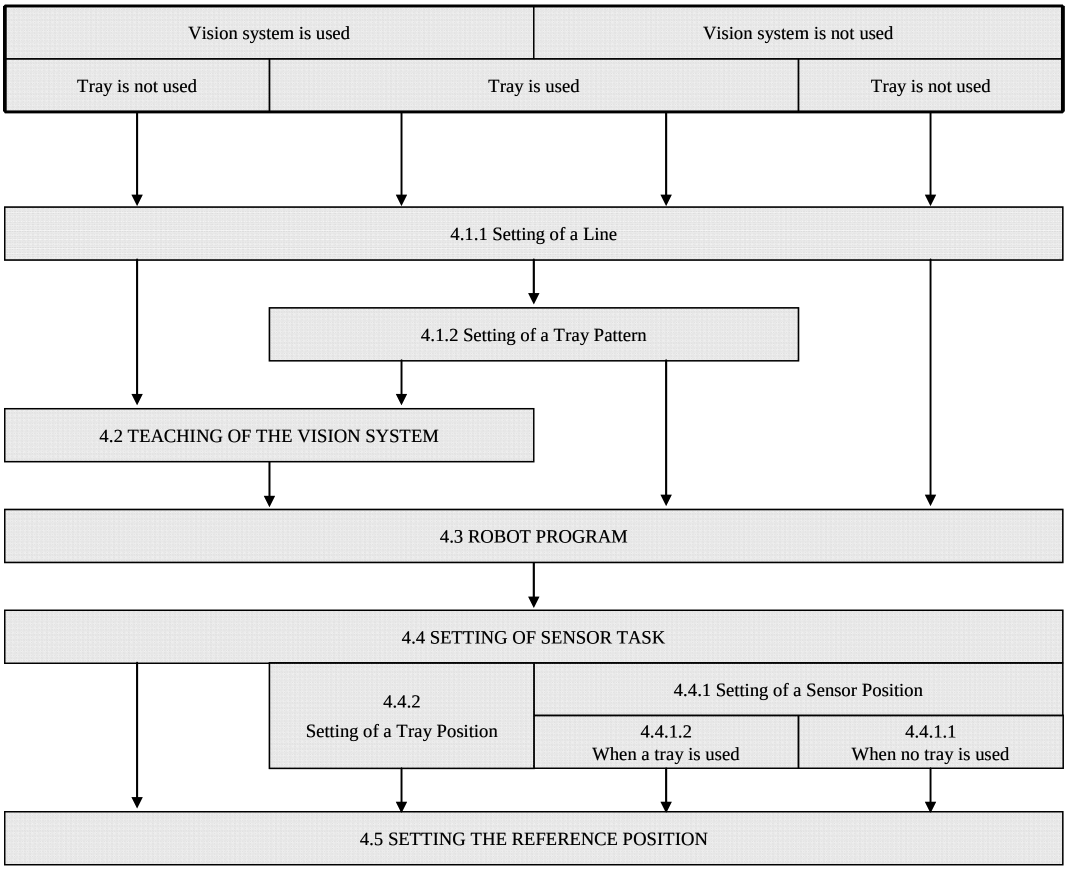
3.6 GRID PATTERN CALIBRATION

For visual tracking camera calibration, a grid pattern is used. Create and prepare a calibration grid in advance. Notes and others on creating a calibration grid are described in the chapter "CALIBRATION GRID" in the "*i*RVision OPERATOR'S MANUAL (Reference)". From FANUC, a standard set of calibration grids is available. The use of a FANUC supplied calibration grid is recommended.

4 SETUP

This chapter describes the basic visual tracking setup procedure. When a system consists of more than one Line (conveyor), the procedure described in this chapter must be repeated as many times as the number of the Lines (conveyors).

The setup will be completed by performing the operations described from 4.1 “SETTING OF A LINE AND A TRAY PATTERN” to 4.5 “SETTING THE REFERENCE POSITION”. The setup operations differ depending on whether using the vision system or not. In addition, the setup operations differ depending on whether using a tray or not. The setup procedures are shown in the following figure. Though you can skip some procedure depending on your system, you should perform the following operation process step by step.



4.6 “FINE ADJUSTMENT OF A TRACKING MOTION” describes how to optimize the tracking motion of a robot after the robot position teaching is completed. 4.7 “SYSTEM START-UP” describes notes on system startup for the multiple robot system.

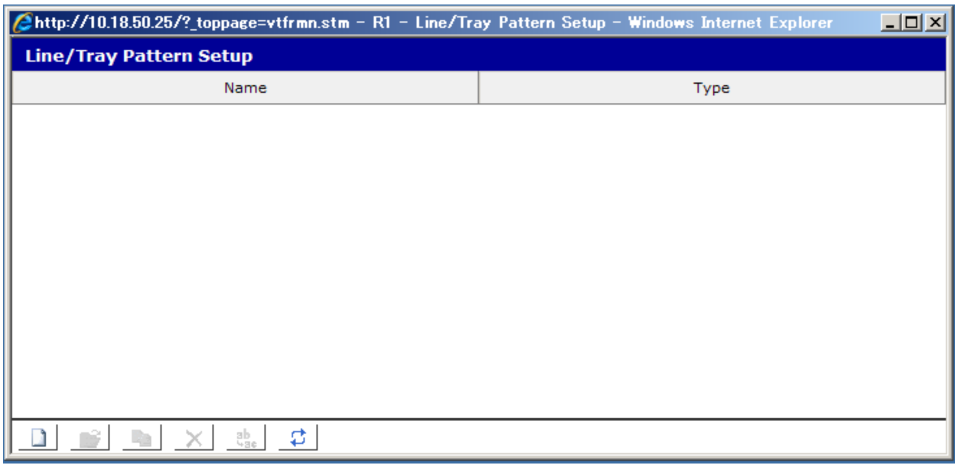
4.1 SETTING OF A LINE AND A TRAY PATTERN

Perform setup specific to visual tracking such as a Line and a Tray pattern.

When a Line and a Tray pattern are set, the same data is set in all the robot controllers in the system variable. If robot ring has been set, the data is transferred to each robot controller and saved automatically.

> ⚠ **CAUTION**
> If the power to a robot controller is off during setting a Line and a Tray pattern, the data in that robot controller becomes inconsistent with the data in the other robot controllers. Therefore, when performing the setting, make sure that all the robot controllers are turned on and robot ring is set correctly.

On the robot homepage, click [Line / Tray Pattern Setup] of [QUEUE MANAGEMENT].



> **NOTE**
> 1 A Line and a Tray pattern can be set from any robot controller in the system if *i*RVision Visual Tracking option or Queue Managed Line Tracking option, and Vision UIF Control option are installed in that robot controller. However, it is impossible to edit the same data from two robot controllers at the same time. Note that vision data such as camera setup, camera calibration, and vision process can be edited from only robot controllers in which *i*RVision Visual Tracking option is installed, and the camera is connected.
> 2 Vision data and a line / a tray pattern cannot be edited by the same controller at the same time. For example, when you edit the vision data with TP at one controller, you cannot edit a line / a tray pattern with a personal computer at the same controller.

Setting the same data in all the controllers

If a robot controller of which power is not turned on is present in the system when settings of a Line or a Tray pattern are changed, data in that robot controller becomes inconsistent with data in the other robot controllers. In such a case, turn on the power to all the robot controllers in the system, and perform the following operation to transfer data again so that the Line and Tray pattern settings are made consistent among all the controllers:

Click the  button. You are asked whether to transfer settings to all the controllers. Click the [OK] button.

When the above operation is performed, setting data in the robot controller in which *i*RVision Visual Tracking option or Queue Managed Line Tracking option is installed currently operated is copied to all the robot controllers in the system.

NOTE

When you add new robots to the system, you need to transfer existing settings to the new robots. Otherwise, the existing settings might be modified by the settings of new robots.
After transferring settings to the new robots, you need to set the Lines again that the new robot joins. Because it is necessary to set a common tracking frame to all the robots again, described in the subsection 4.1.1.2, "Setting a tracking frame and a scale". After setting the tracking frame, all other procedures such as work area setup, camera calibration, reference data setup and robot position teaching should be done again.

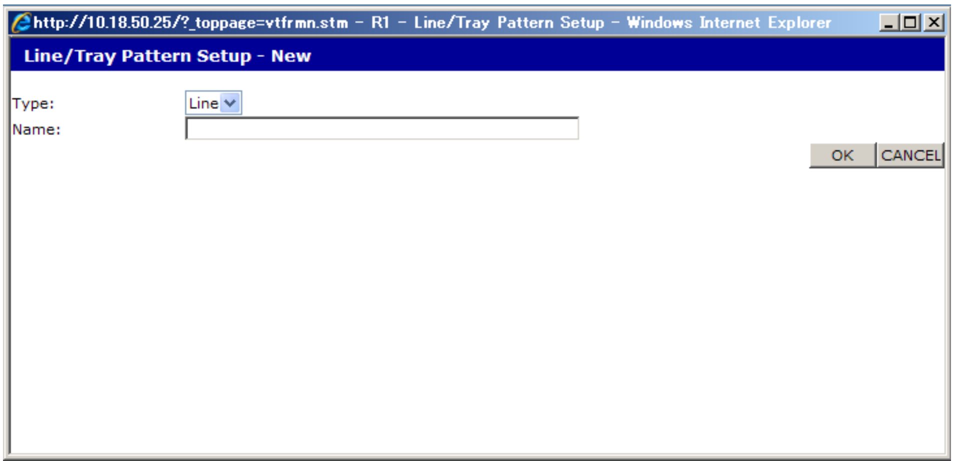
4.1.1 Setting of a Line

Set a Line.

Creating a new Line

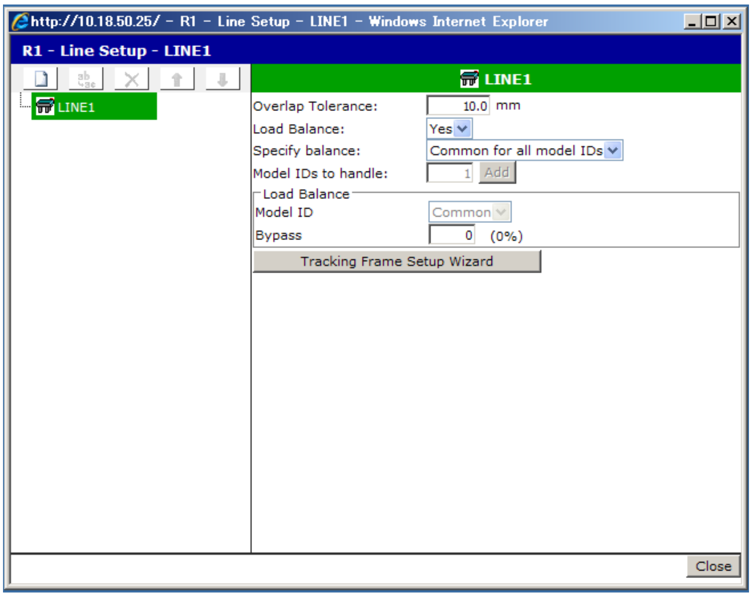
Create a new Line.

1 On the [Line / Tray Pattern Setup] list screen, click the button. The following screen is displayed:



2 In [Type], select [Line].
3 In [Name], enter a Line name. (Example: LINE1)
A name consisting of up to 20 one-byte characters can be specified. The name may not begin with a numeric character or contain a "-" (minus sign).
4 Click the [OK] button.
The created Line is added to the list.

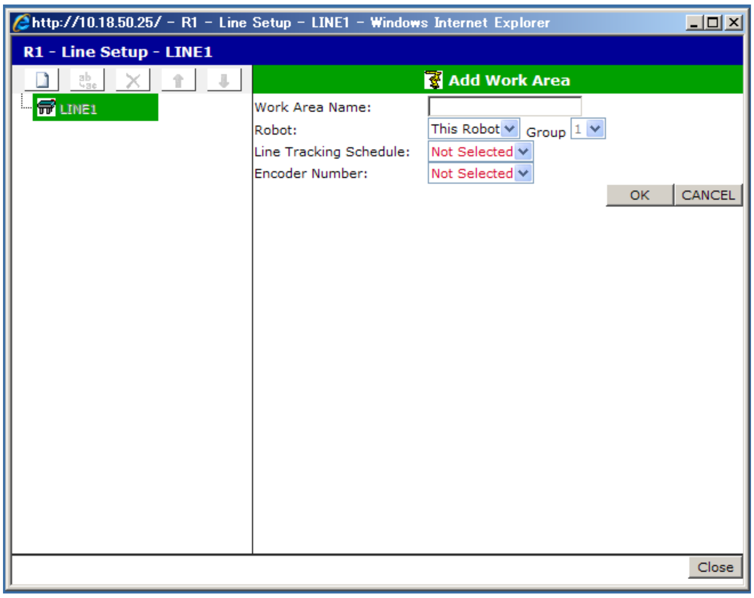
5 Select the created Line and click the button.
A screen as shown below is displayed.



Adding a work area

Add a work area to the Line.

1 To add a work area to the Line, click the button. A screen as shown below is displayed.



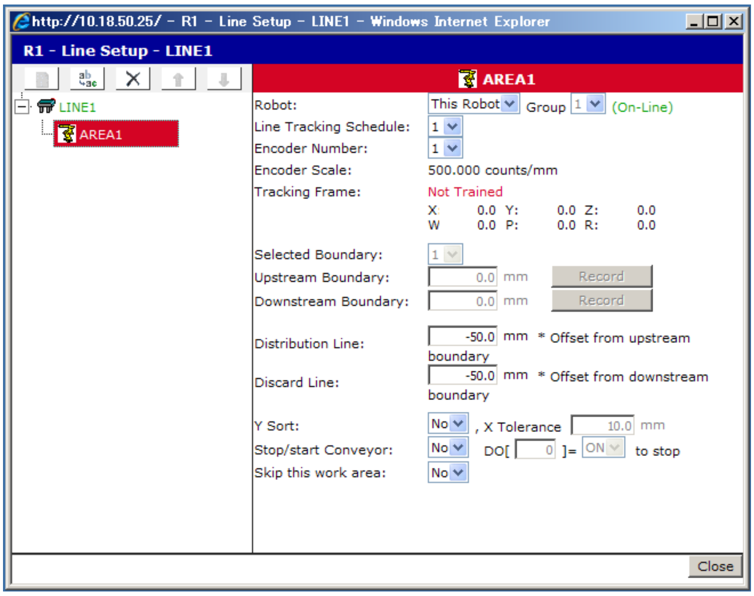
2 Enter the name of a work area in [Work Area Name].
A name consisting of up to 20 one-byte characters can be specified. “-”(minus sign) cannot be used.

3 Set a robot that operates in this work area in [Robot]. To specify a robot, select [This Robot] or the name of other robot controller. A robot of a controller placed offline cannot be added. Perform teaching with the power on all of the controllers.

4 Set a tracking schedule number used for this work area in [Line Tracking Schedule].

5 Set the number of an encoder used for this work area in [Encoder Number].

6 Click the [OK] button to add the work area to the Line.



7 When there is more than one work area on this Line, repeat the above procedure as many times as the number of the work areas.

Deleting a work area

A work area can be deleted from the Line.

1 In the tree view, select a work area to be deleted.
2 Click the ✕ button.

Changing the order of a work area

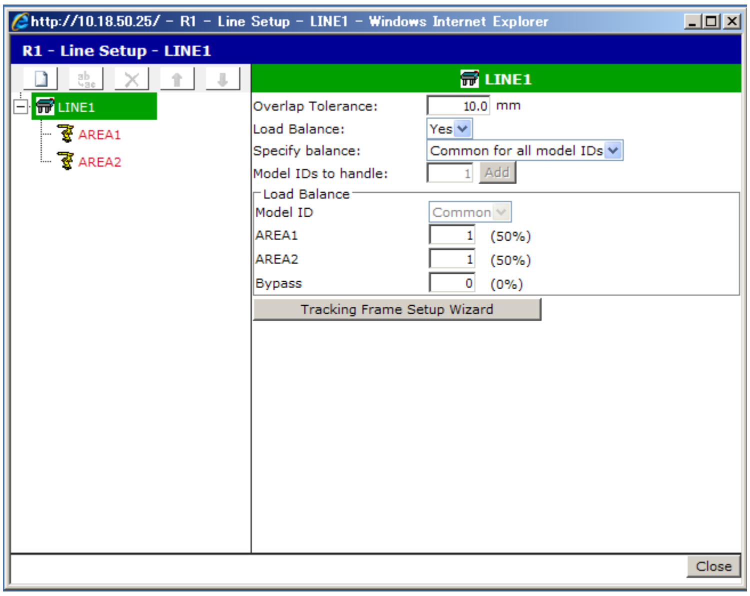
The order of a work area added to the Line can be changed.

1 To move the work area to an upstream location, click the ⇧ button.
2 To move the work area to a downstream location, click the ⇩ button.

> ⚠ **CAUTION**
> Place the work areas so that the order of the work areas displayed in the tree view on the screen matches the order of the robots on the conveyor arranged from upstream to downstream.

4.1.1.1 Setting of a line in detail

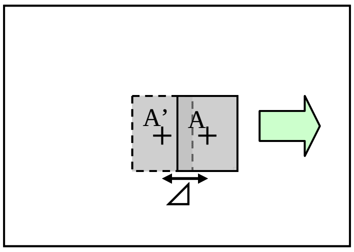
Set a Line in detail.
In the tree view on the left side, click .



Overlap Tolerance

When a workpiece is found in more than one snap as it travels through the field of view, this tolerance is used to help determine if it is the same already found workpiece or a new workpiece. If it is determined that the found workpiece has already been found in an earlier snap then it is not put into the queue multiple times. The default overlap tolerance is 10 mm. During runtime, if the robot tends to pick the same workpiece multiple times, raise the overlap tolerance.

For example, if ⊿ is less than Overlap Tolerance, vision leaves only A’ found in (n-1)th snap.



A’ : Current position of the workpeice found in (n - 1)th snap
A : Current position of the workpeice found in (n) th snap
⊿ : Distance between A and A’

> ⚠ **CAUTION**
> Even though a new workpiece is different from any one of already found workpieces, it may not be put into the queue. If the distance between the new found workpiece and one of already found workpieces is less than Overlap Tolerance, the new found one is not put into the queue even though the model ID of the new one is different from that of already found one.

Load Balance

Specify a method of allocating workpieces to the robots. If load balance is enabled, the information of the workpieces is allocated so that each robot handles workpieces at a specified rate. If load balance is disabled, each robot handles workpieces as many as possible.

Specify balance

When enabling load balance, select a method of specifying a balance (a rate of workpieces handled by each robot). If [Common for all model IDs] is selected, only a single set of balance can be specified. The information of the workpieces is allocated with the specified balance regardless of the model ID of workpieces. If [For each model ID] is selected, a balance can be specified for each model ID. The balance to be used changes according to the model ID of workpieces.

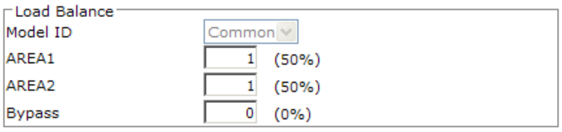
> **NOTE**
> For how to specify a load balance using a model ID, several use examples are given in Chapter 5, “VARIATION”.

Model IDs to handle

If load balance is enabled, and selecting [For each model ID] in [Specify balance], specify model IDs. If you use more than one model ID, you can add the text box for specifying a model ID by clicking the [Add] button. Up to eight model IDs can be specified. If there are unused text boxes, you can remove them by clicking [Delete] button.

Load balance box

The below figure indicates [Load balance box].



If load balance is enabled, specify the rate of the quantity of workpieces to be handled by each robot. Model ID is selected in [Model ID]. Specify the rate of workpieces to be handled in each work area.
For example, when there are three work areas on a Line, and the workpieces is to be equally allocated regardless of the model ID, specify 1 for every work area. Then, the workpieces are allocated at a rate of 1:1:1. 1:1:1 and 2:2:2 are exactly the same.

If [For each model ID] is specified in [Specify balance], select a model ID then set a rate. Set a rate for every model ID set in [Model IDs to handle].

If a value other than 0 is set in [Bypass], this Line intentionally lets workpieces at a set rate to pass by without handling them up. When more than one Line is defined on the same conveyor and it is necessary to convey a certain quantity of workpieces to the downstream side, set a non-zero value.

4.1.1.2 Setting a tracking frame and a scale

Set the tracking frame and the encoder scale.

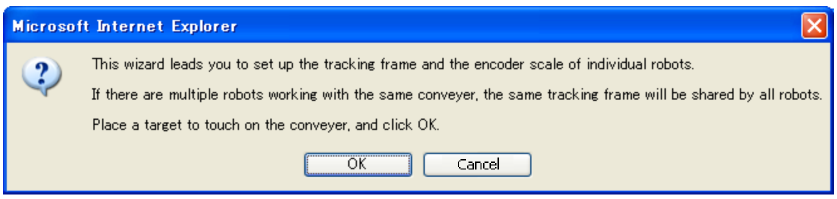
The tracking frame and the encoder scale can be set on the tracking schedule setting menu on the teach pendant of a robot. In particular, when tracking is performed with multiple robots, it is necessary to set a tracking frame common to all the robots. So the method described here is recommended.

> **⚠ CAUTION**
>
> If you use robot with 4 axes or 5 axes, note that the following issues.
> W or P value in the tracking frame might be a non-zero value if the tracking frame is set on the SETUP tracking schedule screen. On the other hand, W and P value in the tracking frame will be always zero if the tracking frame is set as the following procedure. Because 4 or 5 axes robot can perform tracking motion only when the face plate surface faces to the vertical direction, set the tracking frame as the following procedure. The following procedure does not support the tracking motion changing the vertical component of the robot position. If such tracking motion is needed, set the tracking frame on the SETUP tracking schedule screen. You should perform the fine tune by directly entering the component of the tracking frame because the face plate surface should face to the vertical direction while the robot performs tracking motion.
> For 4 and 5 axes robots the W and P values of the tracking frame must be 0.00.

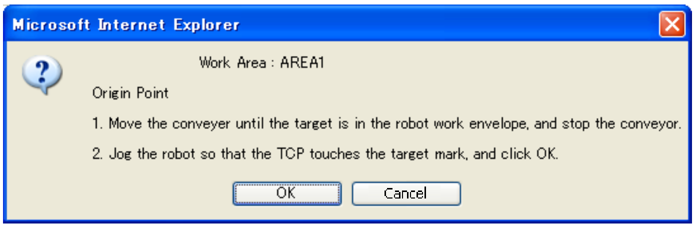
When setting the tracking frame, convey a fixture on the conveyor. Any fixture may be used if it has a target that the robot can touch up accurately with the TCP. When a calibration grid has been purchased from FANUC, a dot on the calibration grid can also be used. We explain here using the calibration grid.

> **⚠ CAUTION**
>
> 1 When the tracking frame is changed after work area setting, camera calibration, reference position setting and robot position teaching needs to be performed again.
> 2 When there are multiple robots, check the encoder schedule menu of each controller when the conveyor is stopped to see whether the [Current Count (cnts)] values of all the controllers match. If they do not match, turn the power to all the controllers off then on.
> 3 When there are multiple robots, fix the fixture on the conveyor belt. If the robot or other things interfere with the fixture when the fixture is conveyed on the conveyor, tracking frame should be set again from the beginning.
> 4 The tracking frame must be accurately taught or the handling precision of the robot may not be good.
> 5 If the robot's posture changes during the teaching of the tracking frame, the tracking frame may not be set up correctly. Jog the robot in only the X, Y, and Z-axis direction so that a robot's posture does not change.

1 Click the [Tracking Frame Setup Wizard] button. The following screen is displayed:

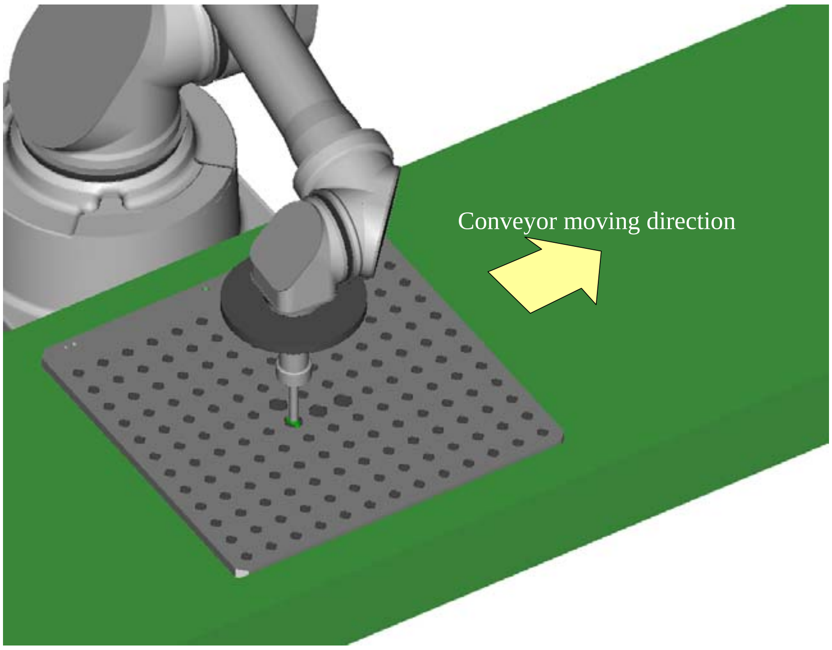


2 Place a fixture that the robot can touch up with the TCP in the most upstream area of the conveyor, and click the [OK] button. The following screen is displayed:

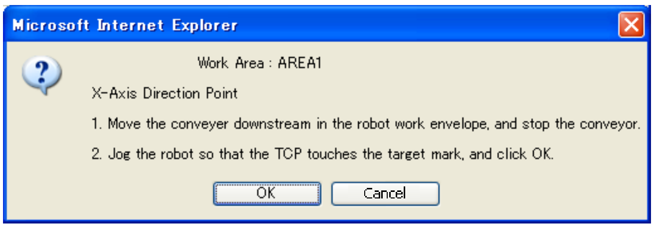


3 Select user tool frame that is set in Section 3.5, “SETTING OF THE TCP”.

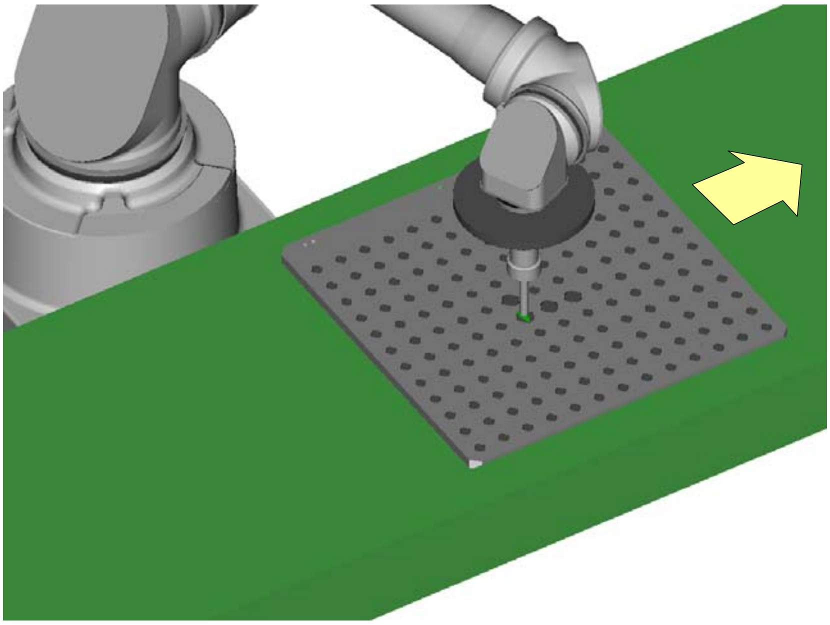
4 Move the conveyor so that the fixture reaches the upstream area within the robot operation range, and then stop the conveyor. Jog the robot to touch up the fixture with the TCP. For example, we touch up the origin that is located at the intersection of large dots arranged in the backwards L shape.



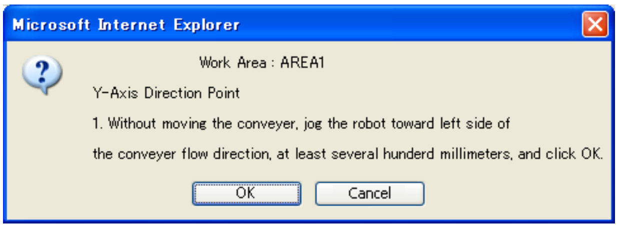
Click the [OK] button in the state where the fixture is touched up. The following screen is displayed:



5 After moving the robot to a position where the TCP does not interfere when the conveyor is moved, move the conveyor so that the fixture reaches the downstream area within the robot operation range, and then stop the conveyor. Next, Jog the robot again to touch up the same position of the fixture you touched up shortly before with the TCP.



Click the [OK] button in the state where the fixture is touched up. The following screen is displayed:

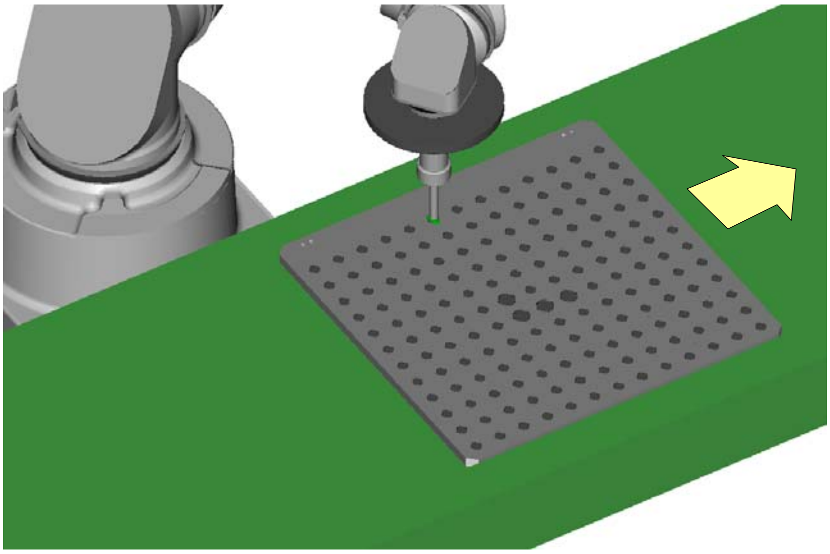


6 Jog the robot at least several hundred millimeters to the left relative to the move direction of the conveyor without moving the conveyor, and click the [OK] button. At this point, touch up accurately the surface of fixture with the TCP.

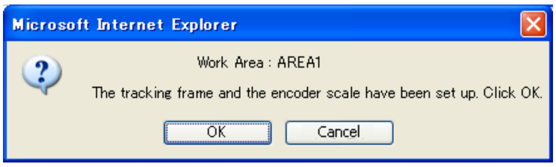
⚠ **CAUTION**

1 If you use robot with 4 axes or 5 axes, this step is skipped. Also the robot and the conveyor should be installed so that the conveyor surface and the XY plane of the world frame should be parallel to each other because the XY plane of the tracking frame becomes parallel to the XY plane of the world frame by this procedure.

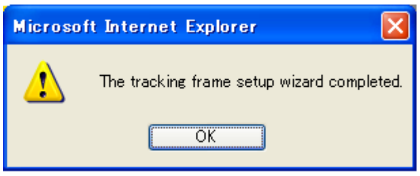
2 If the conveyor is vertical, for example when the workpieces are hanging on hooks from the conveyor, move the robot down to teach the Y-axis direction point instead of towards the left.



The following screen is displayed:



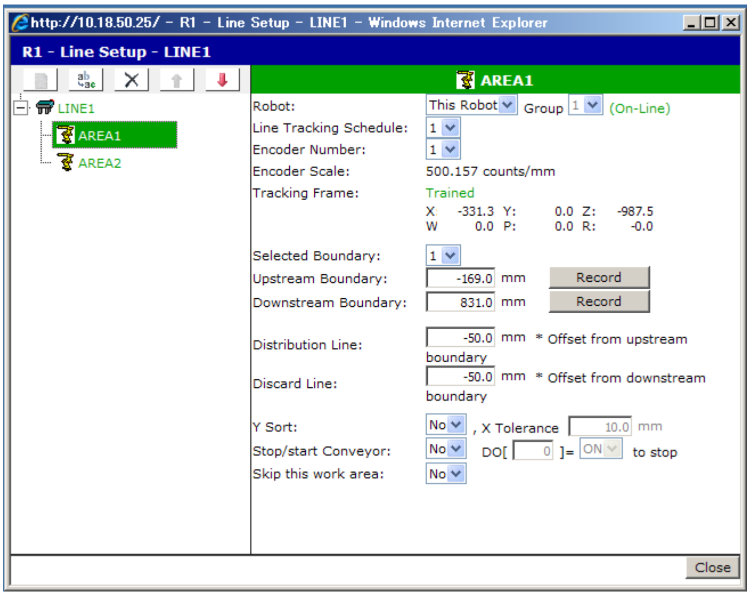
7 The steps above set a tracking frame for the first robot. If there are multiple robots on the same Line, setting of a tracking frame for the second robot starts when the [OK] button is clicked. Repeat steps 3 through 6 for each robot. Upon completion of these settings for all robots, the following screen is displayed:



8 Click the [OK] button. For each robot, the tracking frame and the scale are set.

4.1.1.3 Setting a work area

Set each work area. In the tree view of a Line, click the work area. A screen as shown below is displayed:



> ⚠ **CAUTION**
> Most setting items on the work area setup page cannot be set unless the tracking frame has been set up. Before setting a work area, set the tracking frame.

Robot
A robot that operates in this work area is indicated.

Line Tracking Schedule
A tracking schedule number used for this work area is indicated.

Encoder Number
The number of an encoder used for this work area is indicated.

Encoder Scale
The scale of the set encoder is indicated.

Tracking Frame
The status of the set tracking frame and coordinates are indicated.

Selected Boundary
Select a boundary number to be used. Usually, 1 is selected.

Upstream Boundary
Specify the position of the upstream boundary of the tracking area. [Upstream Boundary] is the most upstream position the robot can reach.
While teaching the boundaries make sure the user tool frame used in the teach pendant program is active.
Move the TCP by jogging to the upstream boundary of the tracking area and click the [Record] button. Then, the X coordinate value in the tracking frame is set. It is also possible to change the value by manual entry.

Downstream Boundary
Specify the position of the downstream boundary of the tracking area. [Downstream Boundary] is the most downstream position the robot can reach.

Move the TCP by jogging to the downstream boundary of the tracking area and click the [Record] button. Then, the X coordinate value in the tracking frame is set. It is also possible to change the value by manual entry.
When you have finished setting up the upstream boundary and the downstream boundary, select the tool frame that is set in Section 3.5, "SETTING OF THE TCP" for subsequent setup.

Distribution Line

Specify the position at which the information of workpieces is allocated to the robot. Specify an offset to the upstream boundary of the tracking area ([Upstream Boundary]). Usually, set a relatively small negative value. Then, the information of workpieces to be handled is allocated to the robot slightly before workpieces enter the tracking area, and as soon as the workpiece enters the tracking area, the robot can handle the workpiece.

> ⚠ **CAUTION**
> Ensure that [Distribution Line] for the most upstream work area is not within the field of view. If [Distribution Line] is within the field of view, the information of the workpieces still seen within the field of view is allocated to the robot, which causes the robot to attempt to handle the same workpiece twice.

Discard Line

Specify the position the workpiece is no longer within the pickable range of the robot. Specify an offset to the downstream boundary of the tracking area ([Downstream Boundary]). Usually, set a relatively small negative value. Then, when a workpiece passes this Line, the workpiece is determined not to be able to be handled, slightly before the workpiece leaves from the tracking area.

Determine the value of the discard line based on the average time between the beginning of tracking motion and the end of work for the workpieces and the conveyor speed as follows:

If the robot operation time is about 0.8 seconds and the conveyor speed is 100 mm/s, the free-running distance of the conveyor is calculated as follows: 0.8 s × 100 mm/s = 80 mm.
Therefore, set the discard line to around −100 mm with some allowances.

> **NOTE**
> If the distance between the distribution line and the discard line is too narrow, the situation where there is no workpiece in the work area may occur sometimes. If there is no workpiece in the work area, the robot waits until the next workpiece passes the distribution line. If there is always at least one workpiece in the work area, the robot with single hand can continue to pick workpieces smoothly. For example, when workpieces are supplied almost at even interval, the robot can be better able to continue to pick workpieces smoothly by setting the distance between the distribution line and the discard line wider than the supply interval.

Y SORT

With this function, the information of the workpieces on the conveyor are rearranged in the Y direction of the tracking frame, that is, in the direction perpendicular to the conveyor movement direction, so that the workpieces can be handled in the order from the workpiece at an edge of the conveyor.

In [X Tolerance], specify a range in the conveyor movement direction. Workpieces within this range are regarded as being in the same row. For example, if 100 mm is specified in [X Tolerance], a workpiece located at a position that has the smallest (or largest) Y value is found among workpieces located within a distance of 100 mm from the most downstream workpiece, and the found information of the workpiece is allocated to the robot first.

Stop/start Conveyor

This function automatically stops the conveyor when a workpiece has passed the discard line. When the robot handles that workpiece, the conveyor is restarted.

When using this function, specify a DO signal for stopping and restarting the conveyor, and the polarity indicating whether to turn the signal on or off to stop the conveyor.

When this function is enabled, the set criterion for the discard line changes. Specify the distance the conveyor coasts during the time from the issuance of the conveyor stop signal until the conveyor actually stops. When the conveyor coasts by a distance of 150 mm until it stops, for example, the discard line should be set to around –150 mm.

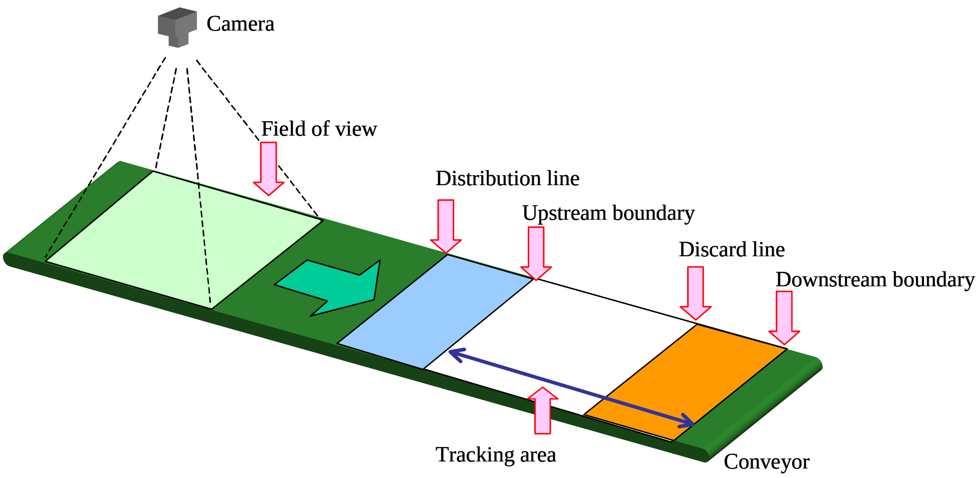
> ⚠ **CAUTION**
> 1 When multiple work areas are defined on the same controller, if the same DO number is specified for the stop conveyor function, and the polarity of DO is reversed (that is, the polarity is set for one work area so that the conveyor is stopped when the DO is turned on, and the polarity is set for another work area so that the conveyor is stopped when the DO is turned off), the stop conveyor function does not work correctly.
> 2 When stop/start conveyor function is enabled, please specify 10 or a greater value as the [5 Average (updates)] on the encoder schedule menu of the teach pendant. When this value is default value 1 and the conveyor is stopped or restarted, “Move error excess” may occur because the robot follows the motion of the conveyor with highly sensitive.

Skip this work area

When this is enabled, no information of the workpieces is fed into this work area. Enable this function when the robot is down intentionally.

Positional relationship of items

The positional relationship among [Upstream Boundary], [Downstream Boundary], [Distribution Line], and [Discard Line] is indicated as shown below.



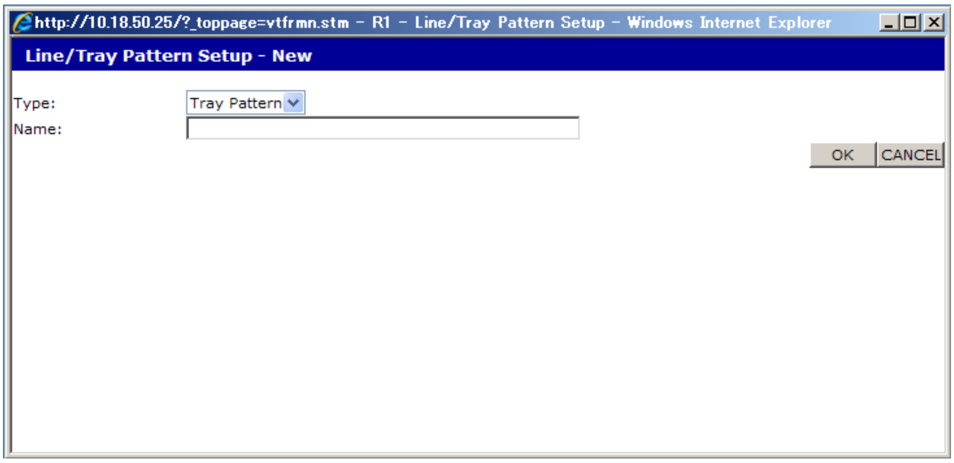
4.1.2 Setting of a Tray Pattern

Set a Tray pattern.

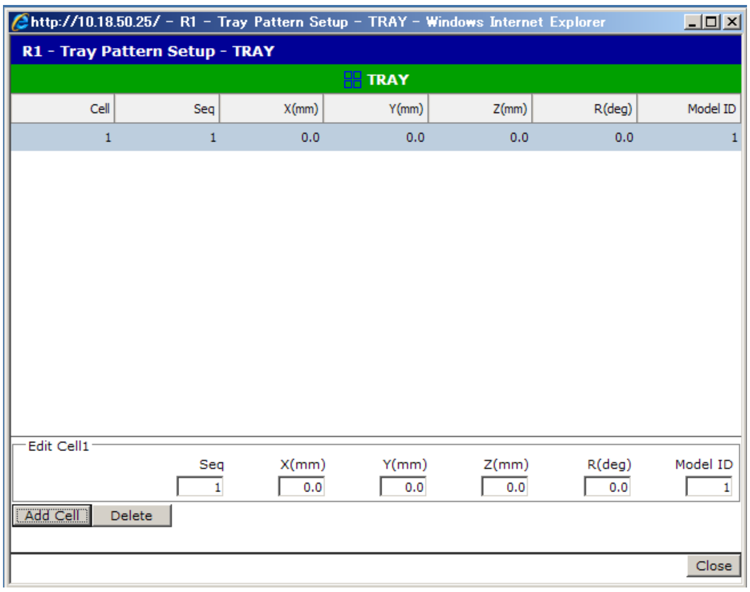
When no Tray is to be used, setting of a Tray pattern is unnecessary.
In this case, skip this section.

Creating a new Tray

1 On the [Line / Tray Pattern Setup] list screen, click the button. The following screen is displayed:



2 In [Type], select [Tray Pattern].
3 In [Name], enter a Tray name. (Example: TRAY1)
A name consisting of up to 20 one-byte characters can be specified. The name may not begin with a numeric character.
4 Click the [OK] button.
The created Tray is added to the list.
5 Select the created Tray and click button.
A screen as shown below is displayed:



Add Cell

Cells are added to the Tray.

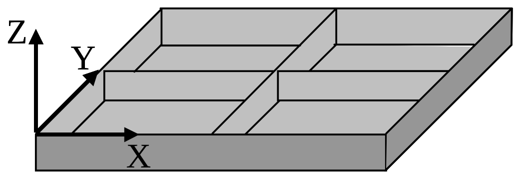
> **NOTE**
> 1 The order in which cells are added is not related to the order in which workpieces are placed or put in a Tray during execution. When workpieces are actually handled in the Tray, they are handled sequentially from a cell on the downstream side depending on the situation. To control the order intentionally, use [Seq] described later.
> 2 Up to 160 cells can be added to a Tray pattern.

X, Y, Z, and R

Enter the coordinates of each cell in the tray frame.
When four workpieces are handled in a Tray, set coordinates as follows:
First, determine where the tray frame is to be defined.
For example, suppose that the tray frame is to be set at an upper corner of a box as shown in the figure below.

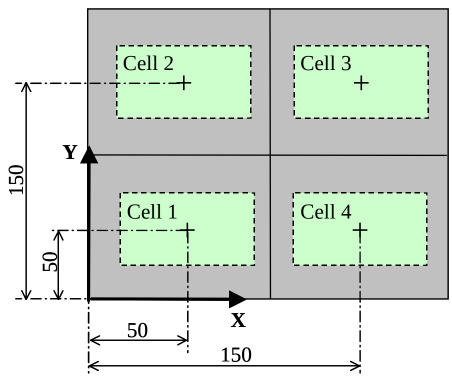
> **NOTE**
> The posture of the tray frame is determined the following way.
> Z-axis direction of the tray frame is the same as Z-axis direction of the tracking frame. XY plane of the tray frame is parallel to XY plane of the tracking frame. Accordingly, when the X-axis direction of the tray frame is determined, the other axis direction of the tray frame is determined.

The tray frame may be defined freely on a Tray. When the tray frame is taught to a robot, the origin of the tray frame and a point in the X-axis direction need to be touched up with the TCP of the robot. So, determine an easy-to-touch-up position.

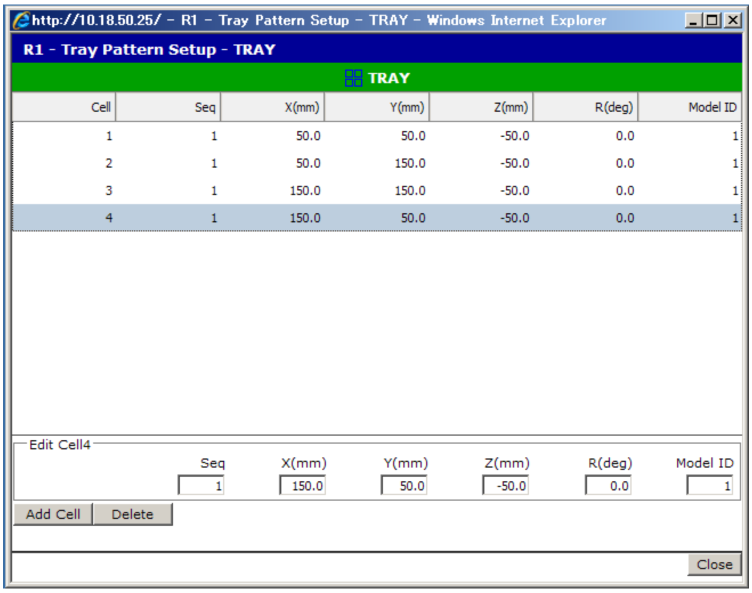


Add cells to the Tray pattern, and enter the coordinates of each cell.

For example, the Tray shown in the above figure is shown as follows when viewed from above:



The cross marks (+) represent the center positions of the individual cells.
The coordinates of each cell are set as follows:



Model ID

A model ID can be set for each cell. Model IDs are used in teach pendant programs in robots when, for example, processing is to be changed according to the model ID of an allocated cell.
If these IDs are not required in particular, the default, 1, need not be changed.

An example using model IDs is presented in Chapter 5, “VARIATION”.

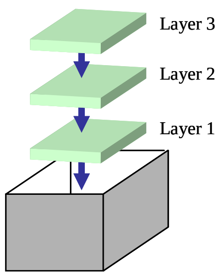
Seq

Cell priority. A smaller value means a higher priority, and a larger value means a lower priority. When 0 or less is set for a cell, the cell is regarded as being nonexistent.

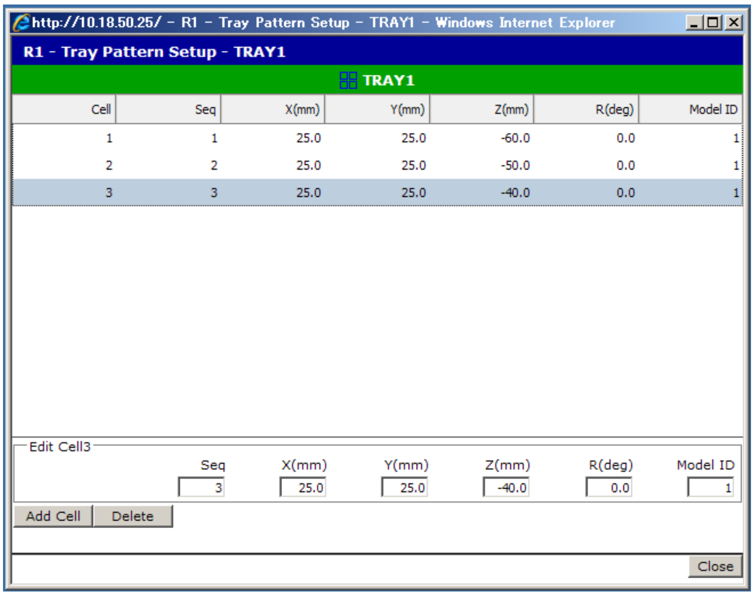
Usually, the same priority is specified for all cells. When a Tray contains many cells having the same priority, the cell order in which workpieces are handled is determined depending on the situation. That is, workpieces are handled sequentially starting from a cell on the downstream side in the tracking area of a robot.

When a Tray has multiple layers, and a workpiece cannot be placed in an upper layer unless placement in the lower layers is completed, the order in which workpieces are placed in a Tray may have to be controlled. In such a case, priority is used. A smaller value means a higher priority. When there are still cells having a smaller priority value, a workpiece cannot be placed in a cell having a larger priority value.

An example of using cell priority is presented below. Workpieces are placed in three layers in a box. In this case, the Tray has three cells. Priority is given as follows. Workpieces cannot be placed in the second and third layers unless a workpiece is placed in the first layer, and a workpiece cannot be placed in the third layer unless a workpiece is placed in the second layer.



In this case, set a Tray pattern as follows:

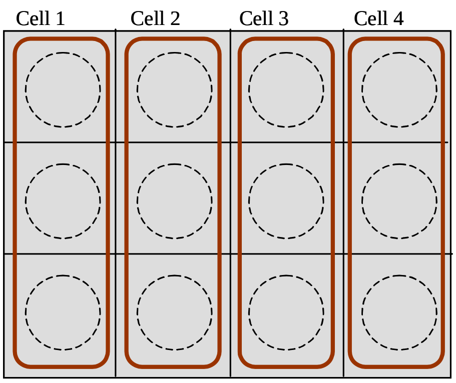


Cell 1 is the first layer (the bottom), cell 2 is the second layer (the middle), and cell 3 is the third layer (the top). The X values and Y values of the cells are the same, but an upper cell has a larger Z value.

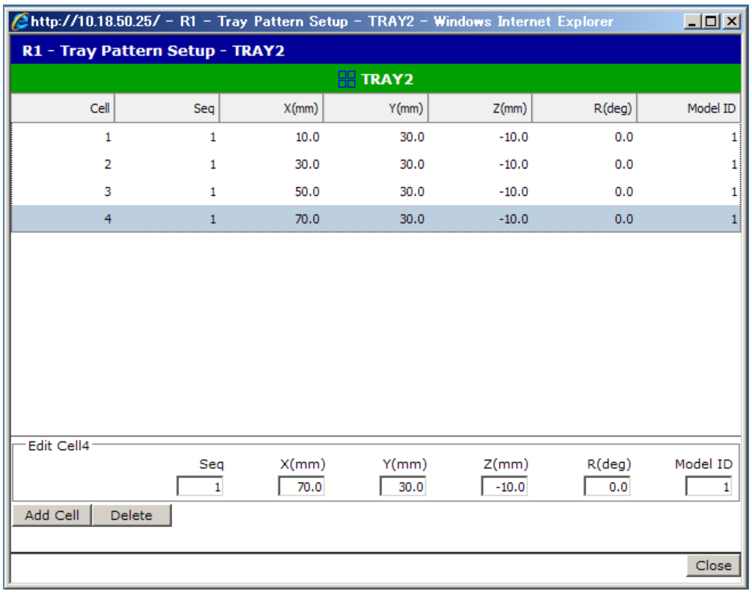
Priority is set to 1 for cell 1, 2 for cell 2, and 3 for cell 3. With this setting, the cell 2 and cell 3 are not allocated to the robot and no workpiece is placed in these cells until a workpiece is placed in cell 1, which has the smallest priority value.

Tray pattern when a robot hand that can hold multiple workpieces is used

This section explains the case when a gripper that can hold multiple workpieces is used and the multiple workpieces are placed by one motion.
For example, consider the case where a robot has a gripper that can hold three workpieces and a Tray has twelve places as shown in the figure below. The circle drawn with a dotted line indicates the workpiece.



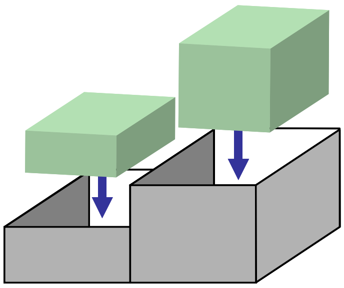
The robot can handle the three workpieces at a time. In such a case, consider the three places that are handled at a time to be one cell. Accordingly, a Tray pattern has four cells in this case as shown in the figure below:



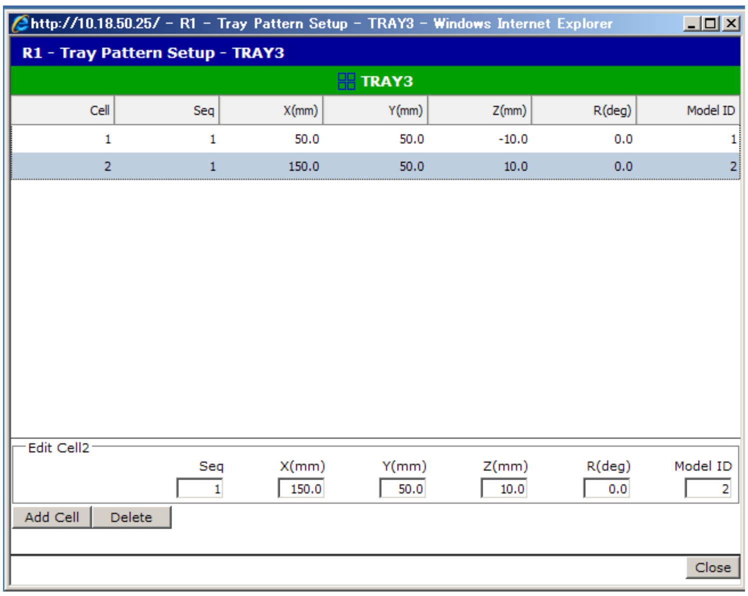
Tray pattern when workpieces have different height

When workpieces have different heights, use the Z value of a cell.

For example, consider the case where there are two cells and the workpieces that are put into these cells have different height as shown in the figure below.



In such a case, modify the Z value of each cell to fit the workpiece’s height. For example, perform setup as shown below:



4.2 TEACHING OF THE VISION SYSTEM

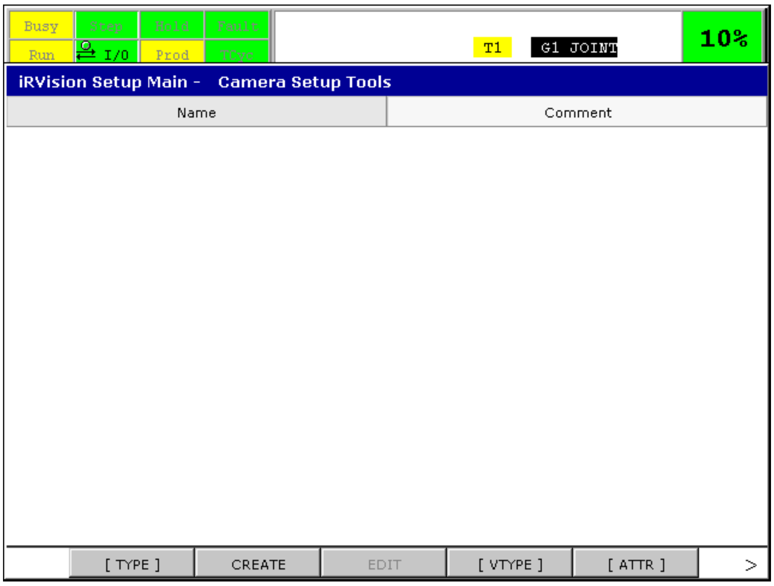
Perform vision system teaching including camera setup and calibration and vision process teaching to detect a workpiece.

This step is unnecessary in the queue managed line tracking system.

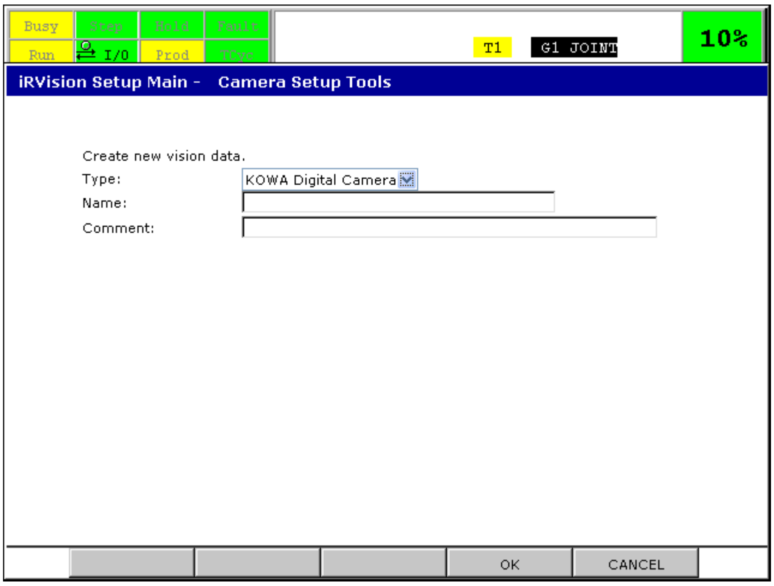
4.2.1 Camera Setup

Perform the camera setup.

1 Press [MENUS] on the *i*Pendant. Select [8 *i*RVision], and then [1 Vision setup].
2 Press F4 [VTYPE], and select [Camera Setup Tools]. A screen as shown below is opened.

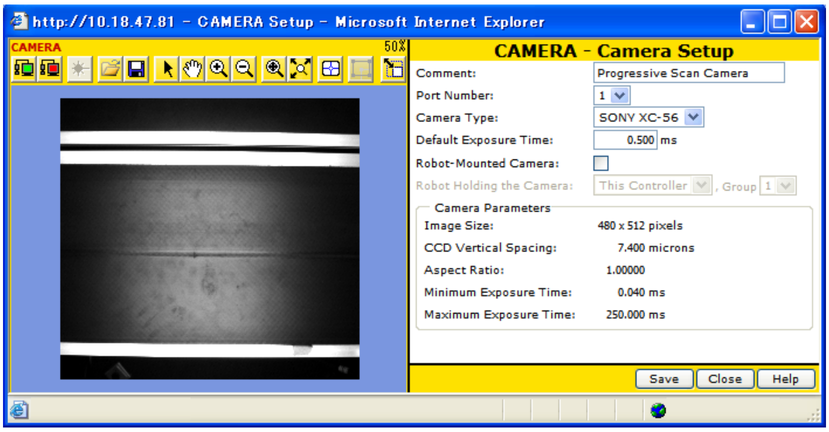


3 Press F2 CREATE. A screen as shown below is displayed



4 Select the type of camera in [Type]. If you use high resolution camera or color camera, select [KOWA Digital Camera].

5 Enter a desired camera name in [Name]. (Example: Camera)
A name consisting of up to 34 alphanumeric characters can be specified. The name may not begin with a numeric character.
6 Press F4 OK. The created camera setup is displayed in the list.
7 Tap the created camera setup and press F3 EDIT. A screen as shown below is displayed.



8 By pressing F3 SNAP, check that a camera image can be snapped normally.
9 Place a workpiece in the field of view then press F3 SNAP again to check the image. Check that the workpiece is located within the field of view and that the image is viewed directly from above. If the workpiece is not snapped normally, adjust the position of the camera.
10 By pressing F2 LIVE, adjust the focus and diaphragm of the camera.
11 Press F5 END EDIT to close the screen.

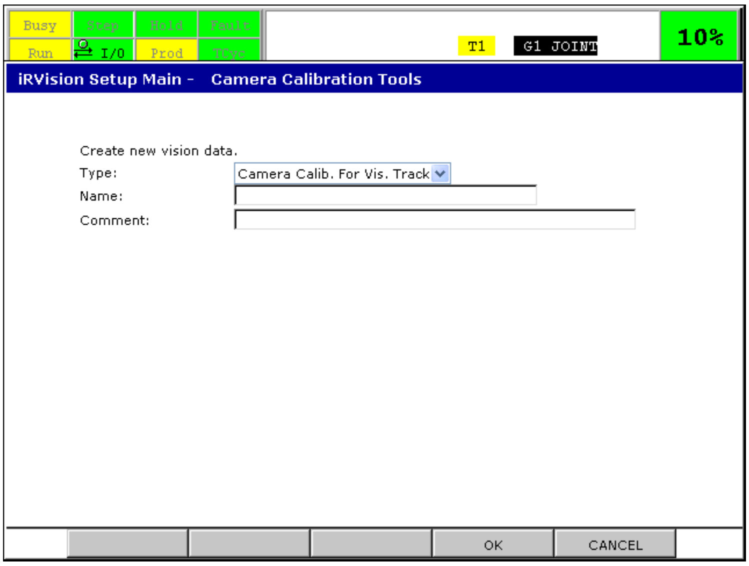
4.2.2 Camera Calibration

Perform camera calibration.
Visual tracking supports the visual tracking calibration only.

Position data on an image detected by performing camera calibration can be converted to position data in the tracking frame.
Camera calibration relates the camera frame with the tracking frame.

Setting the calibration parameters

1 Press [MENUS] on the *i*Pendant. Select [8 *i*RVision], and then [1 Vision setup].
2 Press F4 [VTYPE], and select [Camera Calibration Tools].
3 Press F2 CREATE. A screen as shown below is displayed.

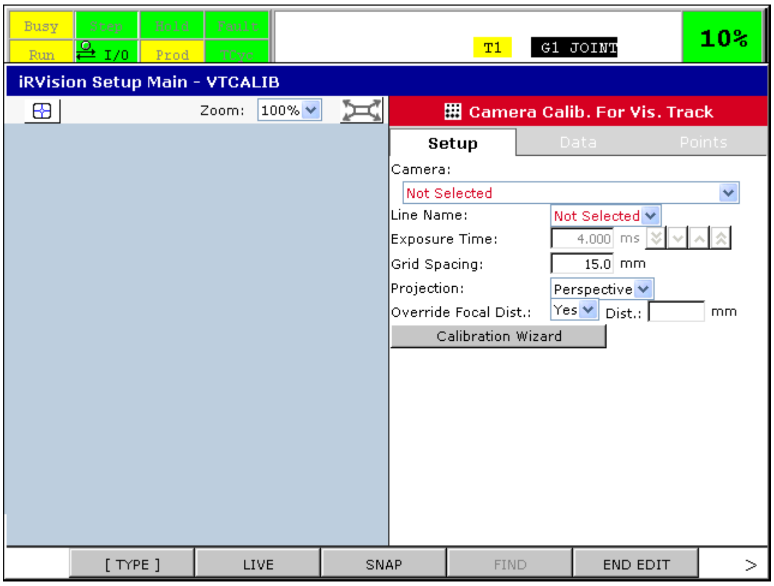


4 Select [Camera Calib. For Vis. Track] in [Type].

5 Enter a desired calibration name in [Name]. (Example: VTCALIB)
A name consisting of up to 34 alphanumeric characters can be specified. The name may not begin with a numeric character.

6 Press F4 OK. The created camera calibration is displayed in the list.

7 Tap the created camera calibration and press F3 EDIT. A screen as shown below is displayed.



8 Select a camera setup in [Camera].

9 Select a Line in [Line].

10 Adjust an exposure time in [Exposure Time].

11 Enter a desired value in [Grid spacing] for the grid pattern.

12 Select [Perspective] in [Projection].

> ⚠ **CAUTION**
> [Orthogonal] can be selected only when workpieces to be detected have the same height and the height of the grid pattern plane is the same as that of the top surface of workpieces to be detected.

13 Select a calculation method in [Override Focal Distance]. Select [Yes].

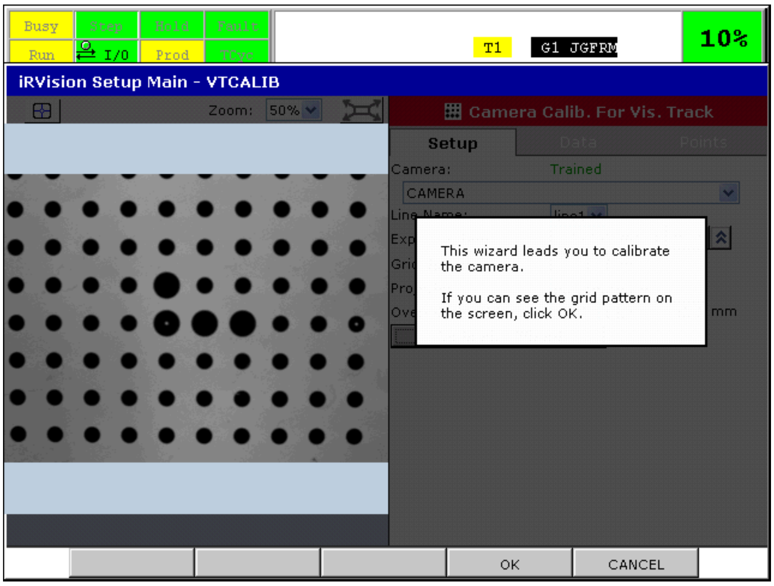
> ⚠ **CAUTION**
> If [No] is selected in [Override Focal Distance], an incorrect focal length calculation may be made theoretically because the calibration grid on the conveyor is approximately parallel with the imaging area of the camera.

14 Set the nominal focal length of the lens.

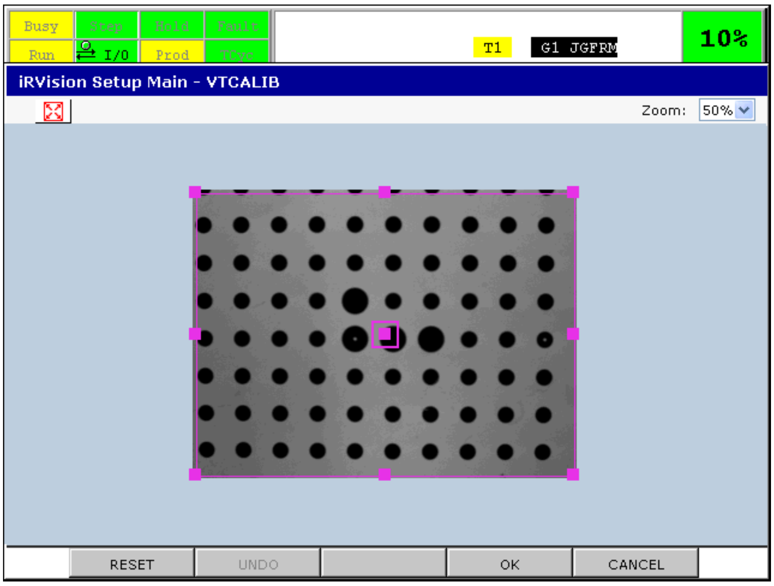
> ⚠ **CAUTION**
> Confirm the focal length of the lens used. If an incorrect focal length is entered, camera calibration cannot be performed correctly.

Executing calibration

1 Tap the [Calibration Wizard] button with the calibration grid placed within the field of view. The screen shown below is displayed.



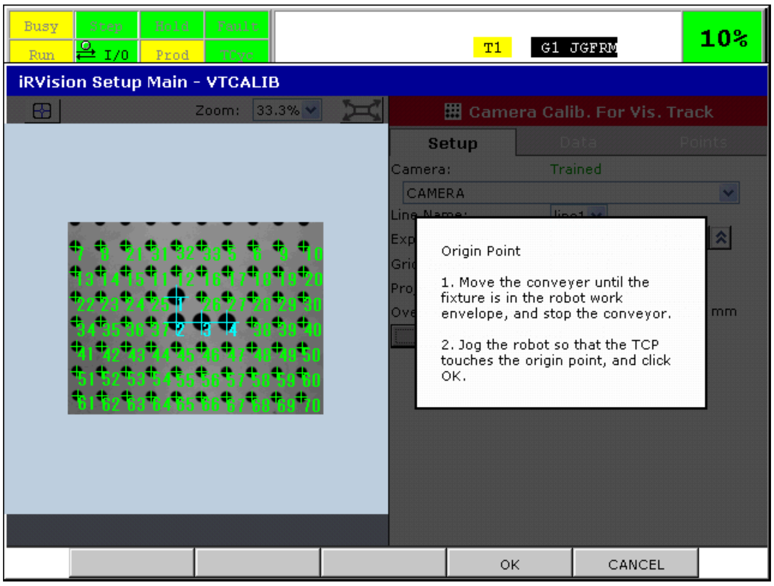
2 Check that the grid pattern on the calibration grid is displayed on the screen, and then press F4 OK. A screen as shown below is displayed.



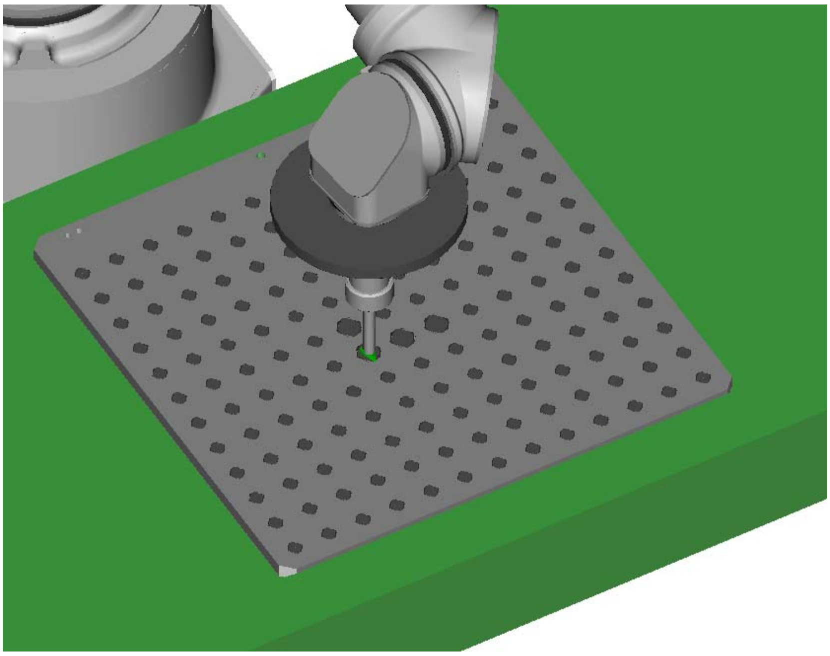
NOTE

Use of a calibration grid that fills up the entire field of view allows the lens distortion to be calculated correctly, therefore improving accuracy in handling a workpiece that appears in a corner of the camera's field of view.

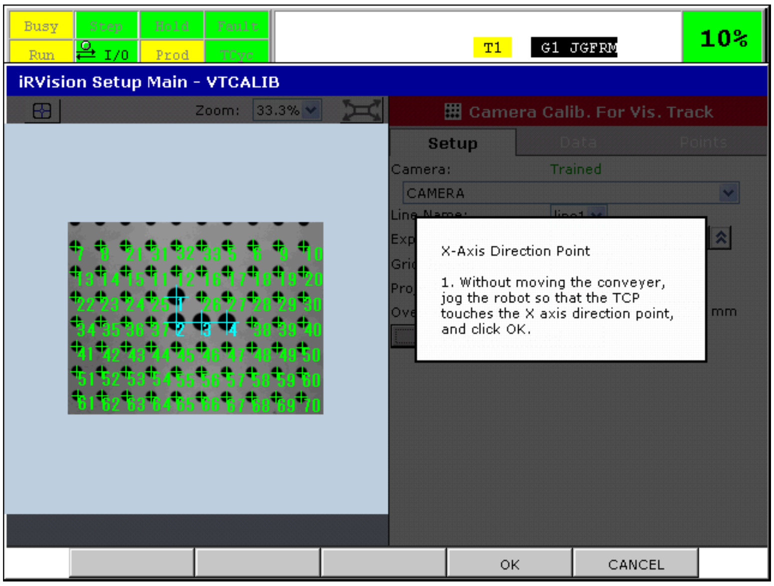
3 Enclose the grid pattern with a magenta rectangular window, and then press F4 OK.
The grid pattern is detected then a screen as shown below is displayed.



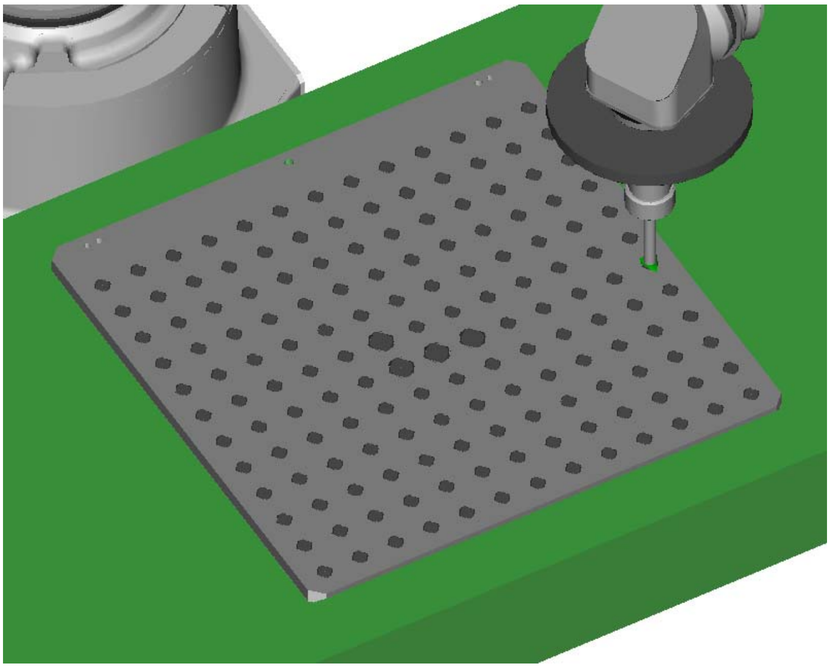
4 Select the tool frame that is set in Section 3.5, "SETTING OF THE TCP".
5 Move the conveyor so that the calibration grid is placed in front of the robot at which vision setup is executed.
6 Touch up the origin of the calibration grid.
The origin is located at the intersection of large dots arranged in the shape of the character L. See the figure shown below.



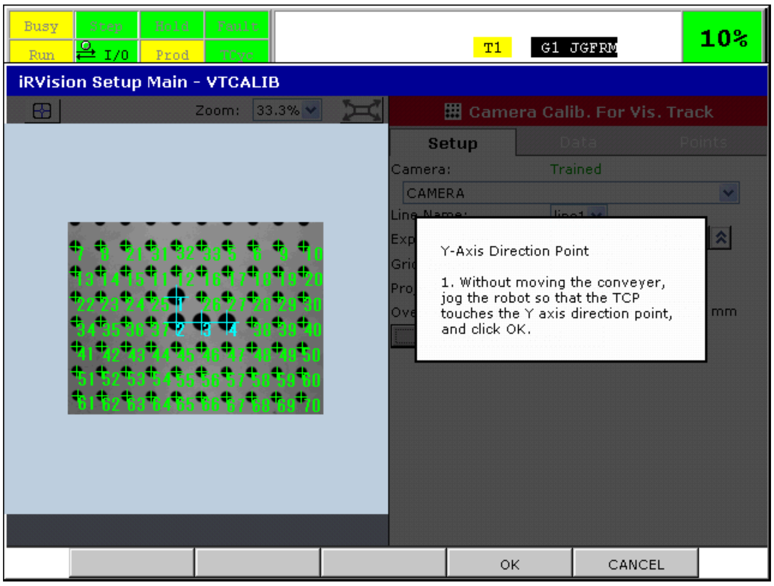
7 Press F4 OK with the origin touched up. The screen shown below is displayed.



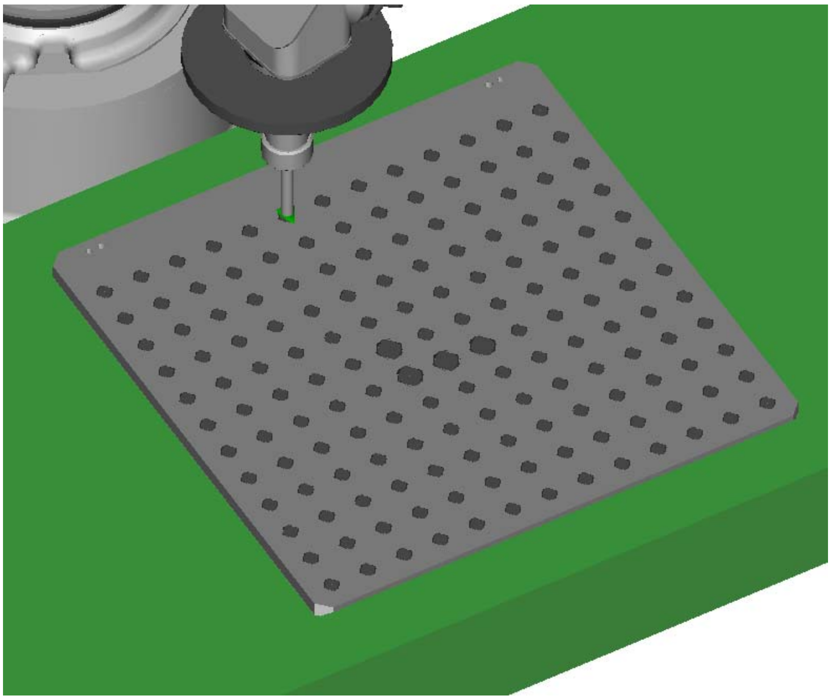
8 Touch up a dot along the X-axis with the TCP.
The X-axis represents the direction in which a greater number of large dots among those arranged in the shape of L are placed. Touch up the dot farthest from the origin. See the figure shown below.



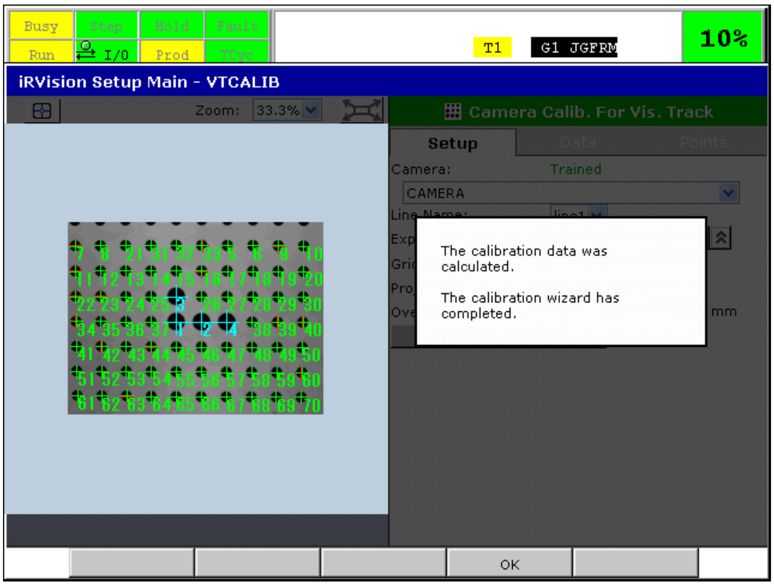
9 Press F4 OK with the dot touched up. The screen shown below is displayed.



10 Touch up a dot along the Y-axis with the TCP.
The Y-axis represents the direction in which a less number of large dots among those arranged in the shape of L are placed. Touch up the dot farthest from the origin. See the figure shown below.



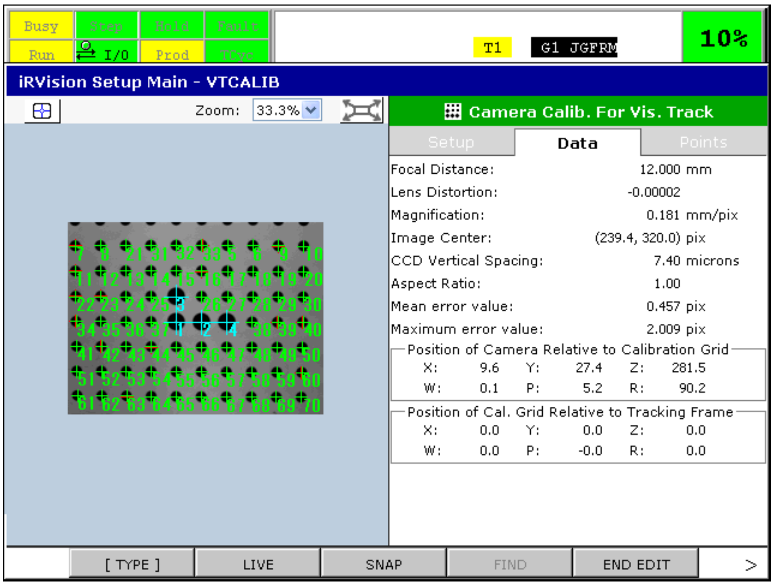
11 Press F4 OK with the dot touched up. The screen shown below is displayed.



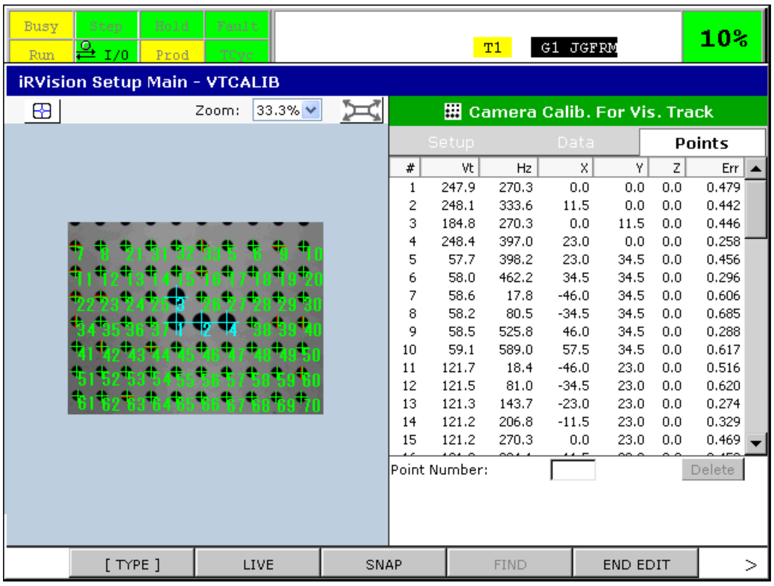
12 When the screen above is displayed, the calibration is completed. Press F4 OK to complete the processing.

Checking the result of calibration

1 To view the result of calibration, click the [Data] tab. An image as shown below is displayed. Check the result of calibration with the image.



2 If detection points (+ marks) displayed on the image do not match dots of the grid pattern, delete the wrong detection points. Tap the [Points] tab. A screen as shown below is displayed.



3 Enter the point number to be deleted in the box after [Point Number] then tap the [Delete] button. The specified point disappears from the image and another calibration calculation is made.
4 Press F10 SAVE to save the data.
5 Press F5 END_EDIT to close the screen.

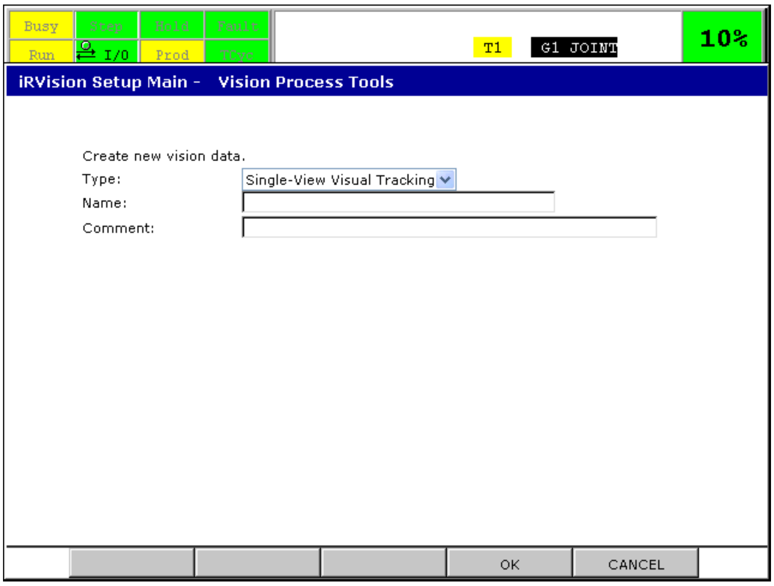
4.2.3 Vision Process Teaching and Test

Teach a vision process and execute a test.

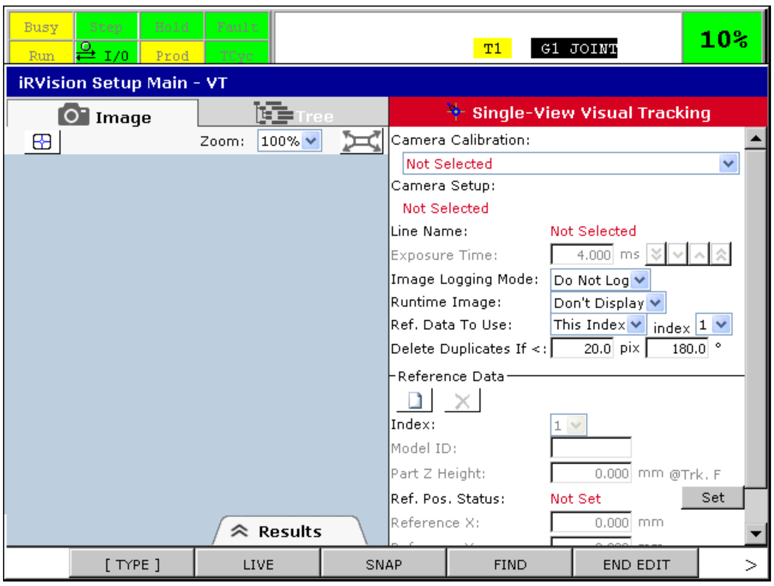
This subsection describes a GPM locator tool as a tool used for detection. Other locator tools such as a blob locator tool are also usable. For details, refer to the chapter, “COMMAND TOOLS” in the “*i*RVision OPERATOR’S MANUAL (Reference)”.

4.2.3.1 Teaching a vision process

1 Press [MENUS] on the *i*Pendant. Select [8 *i*RVision], and then [1 Vision setup].
2 Press F4 [VTYPE], and select [Vision Process Tools].
3 Press F2 CREATE. A screen as shown below is displayed.



4 Select [Single-View Visual Tracking] in [Type].
5 Enter a desired vision process name in [Name]. (Example: VT)
A name consisting of up to 34 alphanumeric characters can be specified. The name may not begin with a numeric character.
6 Press F4 OK. The created vision process is displayed in the list.
7 Tap the created vision process and press F3 EDIT. A screen as shown below is displayed.



8 Select a camera calibration in [Camera Calibration].
9 Adjust an exposure time in [Exposure Time].
10 Select an image logging in [Image Logging Mode].

NOTE
Often, failed images are logged for the purpose of later examination. It takes considerable time to save images, however. So, in visual tracking, [Don't Log] is generally selected not to increase processing time. In the initial adjustment stage, for example, in which increase in cycle time raises no problem, [Log Failed Images] or [Log All Images] may be selected to save images and use them for detection parameter adjustments and so on. Images are saved on the memory card.

11 Select a display mode in [Runtime Image].

NOTE
Usually, [Display With 100%] is selected. Selecting [Display With 50%] or [Don't Display] can save processing time a little during execution.

12 Select an option in [Ref. Data To Use].

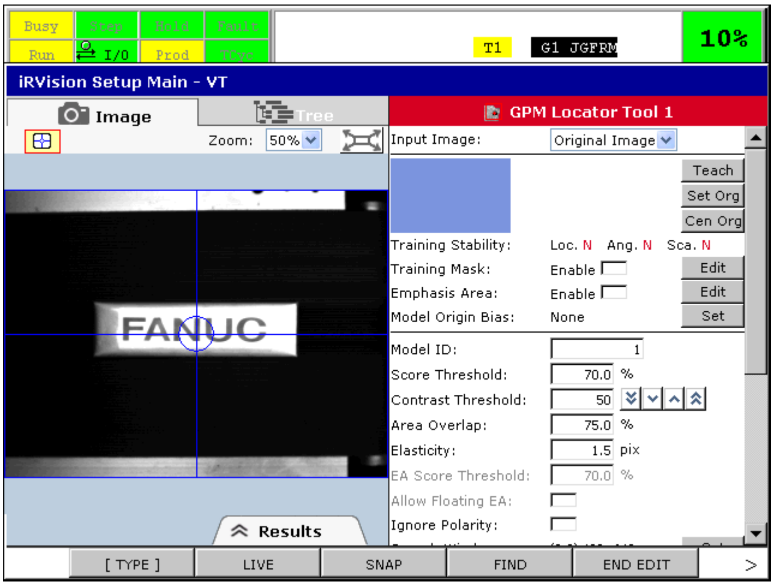
⚠ **CAUTION**
Enter the height of the found part of the workpiece as seen from the tracking coordinate system in [Part Z Height]. This is not the height from the surface of the conveyor. For example, if a thick calibration grid is used to set up the tracking frame, then [Part Z Height] is (the height of the found part of the workpiece - the thickness of the calibration grid) when the direction of z-axis of tracking frame is the upward against the conveyor surface.

NOTE
Usually, select [This Index], and specify [ID] of the reference data used at all times. When multiple types of workpieces with different heights are conveyed, for example, the reference data may have to be changed depending on the detected workpiece. In such a case, select [Model ID], and set [Reference Data] for each model ID. Specify an appropriate value as [Part Z Height] for each [Reference Data].

4.2.3.2 Teaching a GPM locator tool

1 To teach a GPM locator tool, tap [GPM Locator Tool 1] in the tree view located in the left area of the teach window. When multiple model IDs of workpieces are to be detected, multiple GPM locator tools can be created by clicking the button with [Single-View Visual Tracking] selected.

NOTE
If it is needed to detect workpieces and Trays by one camera. It is necessary to prepare separate vision processes to detect Trays and workpieces.



2 Select the image display control in the left area of the setup window and place a workpiece to be taught in the center of the field of view then press F3 SNAP to snap the image.
3 Tap the [Teach] button then enclose the workpiece with a magenta rectangular window.
4 Set the other items as required. For details, refer to the section, “GPM LOCATOR TOOL” in the chapter, “COMMAND TOOLS” in the “*i*RVision OPERATOR’S MANUAL (Reference)”.

4.3 ROBOT PROGRAM

Create a program for the robot according to the system. This section provides examples of basic programs and explains the meanings of these programs. Based on the provided sample programs, tailor the programs for your system.

First, we describe tracking programs. A program that performs a pickup motion and a program that performs a placement motion are to be a tracking program. Unlike ordinary programs, tracking programs require specific settings. Next, we provide a sample set of programs for a robot that performs visual tracking. And last, we explain the specifications of the KAREL Programs used for visual tracking.

> ⚠ **CAUTION**
> If KAREL programs are called in teach pendant programs, it needs $KAREL_ENB to change to 1 in advance.

This section mainly deals with the creation of program logic only. For actually teaching robot positions, see Section 4.5, “SETTING THE REFERENCE POSITION”.

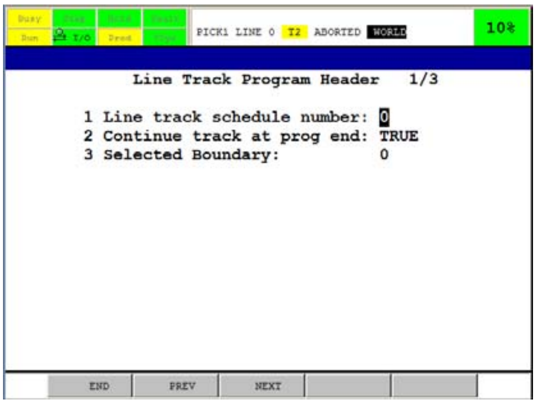
4.3.1 Tracking Program

Programs of robots that track a workpiece are called tracking programs. Unlike ordinary programs, tracking programs require a tracking schedule number and other data to be set on the program detail menu in advance.

The setting method is as follows:

1 Press the [Select] button on the teach pendant.
2 Select a tracking program from the list.

3 Press F2 [DETAIL].
4 Press F3 [NEXT].
5 A menu as shown below is displayed:



6 In [Line track schedule number], set the tracking schedule number specified for the work area.
7 Set [Continue track at prog end] to TRUE.

NOTE
If [Continue track at prog end] is set to FALSE, then the robot stops tracking when the program finishes. In this case, even if the next tracking program starts quickly and the robot starts tracking, the robot motion is not smooth. Set [Continue track at prog end] to TRUE in order to pick up and place workpieces smoothly.

8 Leave [Selected Boundary] set to 0. When this item is set to 0, the tracking area specified for the work area is used.

4.3.2 Sample Program

This subsection provides a set of sample programs for a robot that performs visual tracking. Based on the program provided here, perform customization to match each system.

Generally in a visual tracking system, there are the following three programs:

- Main program
- Pick program
- Place program

The program to perform a pickup motion and the program to perform a placement motion are both called from the main program. When workpieces are picked up from a conveyor and placed on a table, only a program to perform a pickup motion is a tracking program. When workpieces are picked up from a conveyor and placed on a conveyor, a program to perform a pickup motion and a program to perform a placement motion are tracking programs.

The sample program provided here performs a general task; that is, picking up workpieces from an infeed conveyor and placing them on an outfeed conveyor. When a pickup motion or placement motion is not a tracking motion, create a robot program that performs an ordinary operation instead of using a sample program provided here that performs a pickup motion or placement motion (an example of such a robot program is not provided in this manual).

> ⚠ **CAUTION**
> In visual tracking system, disable the Restart position check function. Set the system variable $USRTOL_ENB to FALSE.

The sample programs use the registers and position registers listed below. Change register numbers as necessary.

Registers

R[1]	Cycle stop flag. When 1 is set in this register, a cycle stop occurs in the system.
R[2]	VSTKGETQ status
R[3]	Robot gripper counter*
R[4]	Number of grippers of a robot. When using a hand that can hold one workpiece, set 1 in advance; when using a hand that can hold N workpieces, set N in advance.

* If you use a 4 workpiece gripper and the robot picks each workpiece individually, the robot gripper counter is 4. If you use a 4 workpiece gripper and the robot picks all 4 workpieces at once in one pick, the robot gripper counter is 1.

Position registers

PR[1 to 9]	Pickup position for the Nth gripper
PR[10]	Tool offset to create an approach point for pickup motion
PR[11]	Placement position on the outfeed conveyor
PR[12]	Tool offset to create an approach point for placement motion

4.3.2.1 Main program

The following shows a system main program:

MAIN1.TP

```
  1:   R[1:CYCLE_STOP]=0
  2:   CALL VSTKCLRQ('IN_AREA')
  3:   CALL VSTKCLRQ('OUT_AREA')
  4: J P[1] 50% FINE
  5:   CALL VSTKSTRT
  6:
  7:   LBL[100]
  8:   CALL PICK1
  9:   IF R[1:CYCLE_STOP]=1,JMP LBL[900]
 10:   CALL DROP1
 11:   IF R[1:CYCLE_STOP]=1,JMP LBL[900]
 12:   JMP LBL[100]
 13:
 14:   LBL[900]
 15:   CALL VSTKSTOP
[End]
```

Description

On line 8, PICK1.TP is called to pick up a workpiece, and on line 10, DROP1.TP is called to drop the workpiece. The main program mainly repeats a PICK1.TP call and a DROP1.TP call alternately. When a gripper that can hold multiple workpieces is used, PICK1.TP picks up multiple workpieces and DROP1.TP places them all at one time.

R[1] is a cycle stop flag. The cycle stop flag is checked between pickup and placement motions. If the flag is set, a cycle stop takes place.

On line 5, VSTKSTRT is called to start a sensor task. When a cycle stop occurs, VSTKSTOP is called on line 15 to stop the sensor task.

On lines 2 and 3, VSTKCLRQ is called to initialize work areas. Line 2 initializes the work area of the infeed conveyor, and line 3 initializes the work area of the outfeed conveyor. The work areas must be initialized before sensor tasks start. If the supply side or discharge side is not a tracking program, delete line 2 or line 3 as necessary.

In this sample program, if the cycle stop flag is set at the point of time on line 9, a cycle stop occurs immediately regardless of whether the robot holds a workpiece or not. If a cycle stop is to occur after the robot places a workpiece when the robot holds the workpiece, modify line 9 as follows:

```
  9:  IF R[1:CYCLE_STOP]=1 AND R[3:GRIPPER]=1,JMP LBL[900]
```

The program to perform a pickup motion, PICK1.TP, described later increments the value of R[3] each time the robot picks up a workpiece. Therefore, when the robot is holding a workpiece, R[3] contains 2 or a larger value. By using this, whether the robot is holding a workpiece or not is determined.

When multiple robots are used

Even when multiple robots are used, the main program is the same as MAIN1.TP. The other robots without a sensor task use a main program like MAIN2.TP shown below.

MAIN2.TP

```
     R[1:CYCLE_STOP]=0
     CALL VSTKCLRQ('IN_AREA2')
     CALL VSTKCLRQ('OUT_AREA2')
   J P[1] 50% FINE

     LBL[100]
     CALL PICK2
     IF R[1:CYCLE_STOP]=1,JMP LBL[900]
     CALL DROP2
     IF R[1:CYCLE_STOP]=1,JMP LBL[900]
     JMP LBL[100]

     LBL[900]
 [End]
```

Compared to MAIN1.TP, there is no need to call the instruction to start or stop a sensor task.

> **NOTE**
> You can use MAIN1.TP for the robot that has no sensor task by copying the program. In this case, when you call VSTKSTRT, you will see "All sensors disabled" on the teach pendant. This is not an issue.

When multiple robots have each sensor task, a main program should be as follows for the robots that has a sensor task.

MAIN3.TP

```
  1:    R[1:CYCLE_STOP]=0
  2:    CALL VSTKCLRQ('IN_AREA')
  3:    CALL VSTKCLRQ('OUT_AREA')
  4:  J P[1] 50% FINE
  5:    WAIT 10.00(sec)
  6:    CALL VSTKSTRT
  7:
  8:    LBL[100]
  9:    CALL PICK1
 10:    IF R[1:CYCLE_STOP]=1,JMP LBL[900]
 11:    CALL DROP1
 12:    IF R[1:CYCLE_STOP]=1,JMP LBL[900]
 13:    JMP LBL[100]
 14:
 15:    LBL[900]
 16:    CALL VSTKSTOP
[End]
```

Compared to MAIN1.TP, there is a wait instruction on line 5. All work areas must be initialized before the sensor tasks start. The robots that have a sensor task wait for the initialization of work areas in other robot controllers. Please refer to Section 4.7, "SYSTEM START-UP". The waiting time should be adjusted according to the system.

4.3.2.2 Pick program

The following shows a program to perform a pickup motion with an infeed conveyor:

PICK1.TP

```
   UTOOL_NUM=0
   UFRAME_NUM=0
   R[3:GRIPPER]=1

   LBL[100]
   STOP_TRACKING
   CALL VSTKGETQ('IN_AREA', 1, 100, 2, R[3:GRIPPER])
   IF R[2:GETQ_STATUS]=0,JMP LBL[200]
   IF R[1:CYCLE_STOP]=1,JMP LBL[900]
   JMP LBL[100]

   LBL[200]
L  PR[R[3:GRIPPER]] 4000mm/sec CNT100 VOFFSET,VR[1] Tool_Offset,PR[10]
L  PR[R[3:GRIPPER]] 4000mm/sec CNT3 VOFFSET,VR[1]
   CALL VACUUM_ON
L  PR[R[3:GRIPPER]] 4000mm/sec CNT100 VOFFSET,VR[1] Tool_Offset,PR[10]
   CALL VSTKACKQ('IN_AREA', 1, 1)

   R[3:GRIPPER]=R[3:GRIPPER]+1
   IF R[3:GRIPPER]<=R[4:NUM_GRIPPER],JMP LBL[100]
   LBL[900]
```

```
[End]
```

Description

On the program detail menu, [Line track schedule number] must be the same number as specified one in the work area IN_AREA setup page

On lines 1 and 2, a user tool frame number and a user frame number are selected. As user tool frame number, typically select 0 . In tracking programs, 0 must always be selected as user frame number. In tracking programs user frame 0 refers to the tracking frame not world frame.

On line 7, VSTKGETQ is called to wait for up to 100 millisecond before a workpiece to be picked up next is allocated. When a workpiece is allocated, vision offset is stored in VR[1] and 0 is stored in R[2]. After a workpiece has been allocated, a jump to line 12 is made to perform a pickup motion, and on line 17, VSTKACKQ is called to notify the work area of the completion of the pickup motion.

> ⚠ **CAUTION**
> When using a gripper that can hold multiple workpieces, workpieces should be handled sequentially like PICK1.TP. For example, if all the information of workpieces handled by the gripper is gotten in advance, then the robot cannot pick all of them at the correct positions because the trigger values of tracking motions are not right except for only the trigger value for the last workpiece. The trigger value is set when VSTKGETQ gets the information of one workpiece.

Because the third argument of VSTKGETQ sets the time-out period to 100 ms, if there is no workpiece to be picked up in the work area when VSTKGETQ is called, control exits with a wait time of 100. In this case, R[2] contains a non-zero value. PICK1.TP checks the cycle stop flag, calls VSTKGETQ again, and waits for allocation of a workpiece.

> ⚠ **CAUTION**
> Robot controller where a camera is connected handles TP programs and the sensor task in parallel. Therefore, if the robot controller is overloaded with a considerable processing load of TP programs, the sensor task takes longer to complete the detection process. Then there could be happen like that even though the workpiece on the conveyor has already been conveyed in the tracking area, the robot cannot get the information of it because the sensor task has not finished the detection of it. In such a case, lighten the processing load by inserting the wait instruction in the repetition processing of the TP program. For example, if the third argument of VSTKGETQ which specifies how long the robot waits before the timeout is set to 0 ms, then TP program is busy because VSTEGETQ is called over and over again at a fast rate, and as a result it requires considerable time to detect workpieces by the sensor task.

In R[4], set the number of workpieces the gripper handles. When using a gripper that can hold one workpiece, set R[4] to 1. When using a gripper that can hold multiple workpieces, set the number of the workpieces.

R[3] indicates the number of the gripper to be used for the next pickup motion. It is initialized to 1 on line 3. Each time one workpiece is picked up, the register is incremented by one on line 19. For a gripper that can hold multiple workpieces, therefore, when the first workpiece is picked up, R[3] is set to 1; when the second workpiece is picked up, R[3] is set to 2. In this way, the register is incremented. When the register has reached the number specified in R[4], as many workpieces as the number of all grippers have been picked up.

The fifth argument of VSTKGETQ on line 7 specifies R[3], which holds a gripper number. If the fifth argument is set to 1, VSTKGETQ performs allocation according to the specified load balance. If the argument is set to 2 or a greater value, it allocates workpieces successively regardless of the load balance. Therefore, when a gripper that can hold N workpieces is used, the robot picks up N successive workpieces then lets M, which is determined by the system configuration, successive workpieces pass. To pick up every second workpiece even with a gripper that can hold multiple workpieces, pass 1 as a constant to the fifth argument of VSTKGETQ.

NOTE

M is determined by the specified load balance. For example, consider the case where a gripper that holds four workpieces is used and workpieces are allocated with equal percentage in a two-robot system. In this case, the upstream robot handles 4 successive workpieces and lets 4 successive workpieces pass. On the other hand, consider the case where a gripper that holds four workpieces is used and workpieces are allocated with equal percentage in 3 robots system. In this case, the upstream robot handle 4 successive workpieces and lets 8 successive workpieces pass. Even if the two-robot system is used, when a load balance is set to 1 : 2, the upstream robot handle 4 successive workpieces and lets 8 successive workpieces pass.

⚠ **CAUTION**

The workpieces that come through the [Discard Line] are passed to the downstream robots although the robot handles successive workpieces.

When you want a robot not to pick up successive workpieces even with a gripper that can hold multiple workpieces, set 1 as a constant to the fifth argument of VSTKGETQ. How many workpieces are passed to the downstream robot depends on the load balance settings. When workpieces fill a gripper, tracking program is exited in such a case too.

On lines 13 to 16, a pickup motion is performed. Robot positions are provided by Position registers. This makes fine adjustment of taught points easier during program execution. Position registers are specified indirectly through R[3] in which a gripper number is stored. Therefore, the robot moves to a taught point for a selected gripper.

[CNT3] is used in the pickup point because if [FINE] is used, the robot may not able to pick up a workpiece in a high speed visual tracking application. Please refer to Subsection 4.6.5, “Continuous Termination Type Specification for a Taught Position” for the detail.
You can optimize the tracking motion more by a using the distance before and the time before functions. Please refer to Subsection 4.6.6, “Specification of a Linear Disatnce” and 4.6.7, “Time Before Function” for the detail.

NOTE

In tracking programs, it is recommended that only one TCP that is set at the mechanical interface be used regardless of whether the gripper used can hold one workpiece or multiple workpieces. The reason is to prevent the robot from failing to reach a destination position. During tracking motion, whether the destination position is within the tracking area or not is checked at the TCP position of the robot. So, if the TCP is set at an offset position from the mechanical interface of robot wrist, the robot wrist may be out of the tracking area even when the TCP at the destination position is within the tracking area.

The approach point and retract point of a pickup motion are not determined by directly teaching their positions, but is determined with respect to the pickup position by using a tool offset set in a position register in advance. This makes fine adjustment of the pickup position, approach point, and retract point easier.

A STOP_TRACKING instruction is called on line 6. This instruction stops a tracking motion. In this sample program, [Continue track at prog end] is enabled as set in Subsection 4.3.1, “Tracking Program”. This setting is to reduce the delay between successive pickup and placement motions. If a workpiece to be picked up is not conveyed in a certain period of time, however, the robot continues the previous tracking motion while waiting for the workpiece, and consequently the robot goes beyond the tracking area. To prevent this, the tracking motion is stopped temporarily. Even in this case, if the workpiece information is obtained immediately, a tracking motion is re-started smoothly.

> **NOTE**
> When a program is created for single gripper, the program can become slightly simpler than PICK1.TP. See the program to perform a placement motion as described later.

4.3.2.3 Placement program

The following shows a program to perform a placement motion on an outfeed conveyor:

DROP1.TP

```
  1:   UTOOL_NUM=0
  2:   UFRAME_NUM=0
  3:
  4:   LBL[100]
  5:   STOP_TRACKING
  6:   CALL VSTKGETQ('OUT_AREA', 2, 100, 2, 1)
  7:   IF R[2:GETQ_STATUS]=0,JMP LBL[200]
  8:   IF R[1:CYCLE_STOP]=1,JMP LBL[900]
  9:   JMP LBL[100]
 10:
 11:   LBL[200]
 12: L PR[11] 4000mm/sec CNT100 VOFFSET,VR[2] Tool_Offset,PR[12]
 13: L PR[11] 4000mm/sec CNT3 VOFFSET,VR[2]
 14:   CALL VACUUM_OFF
 15: L PR[11] 4000mm/sec CNT100 VOFFSET,VR[2] Tool_Offset,PR[12]
 16:   CALL VSTKACKQ('OUT_AREA', 2, 1)
 17:
 18:   LBL[900]
[End]
```

Description

Basically, this program is the same as the program to perform a pickup motion.

On the program detail menu, [Line track schedule number] must be the same number as specified one in the work area OUT_AREA setup page

A major difference is that because even when a gripper that can hold multiple workpieces is used, the workpieces are placed all at one time. Whether the gripper can hold one workpiece or multiple workpieces need not be considered, and so gripper management using R[3] and R[4] is unnecessary. For the same reason, the last argument of VSTKGETQ.PC on line 5 is set to constant 1.

4.3.3 Visual Tracking Instructions

The visual tracking option and the queue managed line tracking option provide KAREL programs shown below. These KAREL programs can be called using the CALL instruction to use various functions. This subsection explains the specifications of the provided KAREL Programs.

> ⚠ **CAUTION**
> 1 If KAREL program is called in teach pendant program, it needs $KAREL_ENB to change to 1 in advance.
> 2 These KAREL programs do not support the hot start. Please disable the hot start when you use these KAREL programs.

4.3.3.1 VSTKCLRQ.PC

This program clears information of workpieces present in a specified work area. At this point, the information of the workpieces in the work area is entirely erased. Usually, this program is called just once when the system is started.

Argumen1:
Specify a name of a work area with a character string.

4.3.3.2 VSTKGETQ.PC

This program gets the information of one workpiece from a work area. The information of the gotten workpiece is stored in a vision register. The value of the encoder for the gotten workpiece is set as the trigger of a tracking motion. When there is no workpiece to be handled in the work area, the robot waits by a specified time until a workpiece is actually conveyed in the work area.

Argument 1:
Specify a name of a work area with a character string.

Argument 2:
Specify a vision register number to store information on allocated workpiece.

Argument 3:
Specify time in milliseconds for which the robot waits for a workpiece when there is no workpiece to be handled in the work area. When a negative value is specified, the robot waits indefinitely until a workpiece is actually conveyed in the work area. When zero is specified, this instruction returns immediately.

Argument 4:
Specify a register number to store the processing status. When the information of a workpiece could be gotten normally, 0 is stored in the register. When the robot stops waiting because of a time-out, 1 is stored in the register.

Argument 5:
Usually, specify 1. To allocate workpieces successively regardless of the load balance to each work area, which is specified in the Line setup page, specify 2 or a greater value. The detailed explanation is introduced in Subsection 4.3.2.2, “Pick program”.

Argument 6(optional):

When workpiece information is to be acquired by specifying a particular model ID, specify the model ID. This argument can be omitted. Usually, this argument is not specified. A use example is provided in Chapter 5, “VARIATION”.

4.3.3.3 VSTKACKQ.PC

This program notifies the work area of the information about how the allocated workpieces was handled.

Argument 1:

Specify a work area name with a character string.

Argument 2:

Specify a vision register number that holds information on the allocated workpiece.

Argument 3:

Specify one of the following values as a processing result:

1: Specified when the allocated workpiece has been handled correctly.

2: Specified when the allocated workpiece has not been handled according to plan. This applies when, for example, the position or model ID of the allocated workpiece is evaluated by the teach pendant program of the robot and handling of the workpiece is canceled intentionally. In this case, this workpiece is passed to the next work area.

3: Specified when the handling of the allocated workpiece fails. This applies when, the allocated workpiece is picked up, and then vacuum confirmation shows an unacceptable result. In this case, because the robot has moved the workpiece, a downstream robot cannot handle this workpiece even when the information of the workpiece is returned to the work area. So, the information of the workpiece is deleted.

When you enable the stop conveyor function and this function stops the conveyor because the robot cannot handle a workpiece before the workpiece passes the discard line, the conveyor is re-started after VSTKACKQ is called.

4.3.3.4 VSTKSTRT.PC

This program starts sensor tasks. All sensor tasks enabled in Section 4.4, “SETTING OF SENSOR TASK” are started. There is no argument.

> ⚠ **CAUTION**
> If an error occurs in the sensor task, this KAREL program will abort.

4.3.3.5 VSTKSTOP.PC

This program stops sensor tasks.
There is no argument.

4.3.3.6 VSTKSTLB.PC

This program changes the load balance data of each work area in a line. The change is reflected immediately.

Argument 1:

Specify a name of a line with a character string.

Argument 2:

Specify a register number to store load balance information. The value stored to R[the register number] is used as load balance info for the most upstream work area. The value stored to R[the register number + 1] is used as load balance info for the second work area. The effective range of the values stored to the registers is 0 ~ 255.

Argument 3:

Specify the number of work areas where you want to set the load balance. Generally, you specify the total number of work areas in a line. For example, when there are three work areas in a line and you specify 3 here, you can set the load balance for all work areas.

Argument 4:

Specify a value of [Bypass].

Argument 5(optional):

Specify a model ID to set the load balance. In the case of selecting "Common for all model IDs", you can omit this argument.

For example:

When the line name is LINE1 and there are three work areas in the line and you want to set the load balance to 1:2:3 and set [Bypass] to 0 for model ID 8, you call the following instruction. The three registers are used. R[11] = 1, R[12] = 2, R[13] = 3.

```
CALL VSTKSTLB('LINE1',11,3,0,8)
```

4.3.3.7 VSTKENLB.PC

This program enables or disables a load balance function. The change is reflected immediately.

Argument 1:

Specify a name of a line with a character string.

Argument 2:

Specify one of the following values:

1: Specified when you want to enable the load balance function.

0: Specified when you want to disable the load balance function.

⚠ CAUTION

If 1 is specified as the argument 2 to enable the load balance function, please be careful with the timing of calling this program. In particular, be careful when there are some workpieces which pass the discard line because many workpieces flow to the robot. If a workpiece passes the discard line soon after the load balance function enabled by VSTKENLB, then the robot consider itself missing the first workpiece. In that case, the robot may pick 2 workpieces consecutively at one time contrary to the setting of picking every other workpiece. In order to prevent things like this, please call this program when there are workpeces to be handled at more upstream, for example, right after getting information of a workpiece by VSTKGETQ.

4.3.3.8 VSTKENSC.PC

This program enables or disables the stop conveyor function for each work area in a line. The change is reflected immediately.

Argument 1:

Specify a name of a line with a character string.

Argument 2:

Specify a register number to store a flag to enable or disable the stop conveyor function. The value stored to R[the register number] is used to enable or disable the stop conveyor function for the most upstream work area. The value stored to R[the register number + 1] is used to enable or disable the stop conveyor function for the second work area. The effective range of the values stored to registers is 0 or 1. If you want to enable the stop conveyor function, specify 1. Otherwise, specify 0.

Argument 3:

Specify the number of work areas where you enable or disable the stop conveyor function. Generally, you specify the total number of work areas in a line. For example, when there are three work areas in a line and you specify 3 here, you can enable or disable the stop conveyor function for all work areas.

For example:

When the line name is LINE1 and there are three work areas in the line and you want to enable the stop conveyor function for the most upstream work area and the third work area and you want to disable the stop conveyor function for the second work area, you call the following instruction. The three registers are used. R[11] = 1, R[12] = 0, R[13] = 1.

```
CALL VSTKENSC('LINE1',11,3)
```

4.3.3.9 VSTKSTVN.PC

This program changes the vision process name that the sensor task runs.
If you have already ran a sensor task by running VSTKSTRT.PC, please stop the sensor task by running VSTKSTOP.PC and call this program.

When you want to handle many kinds of workpiece, there are two methods.

1 You make one vision process. That vision process has models for many kinds of workpiece. But this method may need much processing time because that vision process find a workpiece using all model.
2 You make vision processes every kind of workpiece. And you switch each vision process that is run in sensor task depending on the situation. This program is used in this method.

Argument 1:

Specify a sensor task number.

Argument 2:

Specify a name of a vision process name with a character string.

Argument 3(optional):

Specify a name of a second vision process name with a character string when you use the multi-view functions . In this case, specify a name of a first vision process name as argument 2. You must NOT specify this argument when you use a single view.

4.3.3.10 VSTKSTTN.PC

This program changes the tray name of a specified sensor task.

Argument 1:
Specify a sensor number.

Argument 2:
Specify the name of a tray pattern by a character string.

For example:
When you want to change the tray name of sensor task 1 to TRAY2, you call the following instruction.

```
CALL VSTKSTTN(1,'TRAY2')
```

4.3.3.11 VSTKGETT.PC

This program returns the estimated time to take until the next workpiece that can be picked arrives at the distribution line. The unit of the time is millisecond. The time is calculated by using the conveyer speed when this program is called.
If the workpiece has not passed the distribution line, this program returns a positive value. If the workpiece has already passed the distribution line, this program returns a negative value. If conveyor is stopped, this program returns 2147483646 when the workpiece does not pass the distribution line. If conveyor is stopped, this program returns –2147483647 if the workpiece has already passed the distribution line. Please call this program after VSTKACKQ and before VSTKGETQ. If this program is called after VSTKGETQ and before VSTKACKQ, then it returns the error of “No AckQueue before next GetQueue.”

Argument 1:
Specify a name of a work area with a character string.

Argument 2:
Specify a register number to store the time it takes until the next workpiece to be picked arrives at the distribution line.

Argument 3:
Usually, specify 1. To get a possibility time of workpieces successively regardless of the load balance to each work area, which is specified in the Line setup page, specify 2 or a greater value.

Argument 4(Optional):
When a possibility time of workpieces is to be gotten by specifying a particular model ID, specify the model ID. This argument can be omitted. Usually, this argument is not specified.

4.3.3.12 VSTKPFRT.PC

This program returns the rate at which the workpieces flow through the work area in one minute (expected workpiece flow rate).
The expected workpiece flow rate is calculated as follows.

Expected workpiece flow rate (workpieces/min) = Expected number of workpieces (workpieces) * Conveyor speed (mm/sec) * 60 (sec/min) / Distance that corresponds to the work area (mm)

Where "Expected number of workpieces" and "Distance that corresponds to the work area" are calculated as follows.

For each work area, the region for calculation is defined. The expected number of workpieces is calculated by use of workpieces contained only in this region. The downstream boundary of this region is the discard line of the work area. The upstream boundary of this region is any one of the following three.

1. If the work area is the most upstream one, the upstream boundary is the most upstream detection position of the workpieces which have been held by the work area when VSTKPFRT is called for the first time with the work area holding at least one workpiece.
2. If the work area is not the most upstream one and the "Y Sort" is disabled, the upstream boundary is the discard line of the previous work area in the line.
3. If the work area is not the most upstream one and the "Y Sort" is enabled, the upstream boundary is the position more downstream by the "X Tolerane" of the "Y Sort" from the discard line of the previous work area in the line.

Expected number of workpieces

This is the number of workpieces which are contained in the region for calculation and expected to be handled at the work area on the basis of the settings of "Load Balance" of the line, "Y Sort" of the work area and "Seq" of the tray pattern and the current performance. For example, if the setting of "Load Balance" is 50% at the most upstream work area, and before now 4 workpieces out of 7 have been handled at this work area, and 3 workpieces are contained in the region for calculation, then the expected number of workpieces is 1 (=(7+3)/2－4).

Distance that corresponds to the work area

This is the distance between the upstream and downstream boundaries of the region for calculation of the expected number of workpieces.

Argument 1:

Specify the name of a work area by a character string.

Argument 2:

Specify the index number of a register to store the expected workpiece flow rate.

For example:

When you want to output the expected workpiece flow rate of AREA1 to R[12], you call the following instruction.

```
CALL VSTKPFRT('AREA1',12)
```

4.3.3.13 VSTKNPRT.PC

This program returns the number of workpieces that are held by a specified work area.

Argument 1:

Specify the name of a work area by a character string.

Argument 2:

Specify the index number of a register to store the number of workpieces that are held by the work area.

For example:

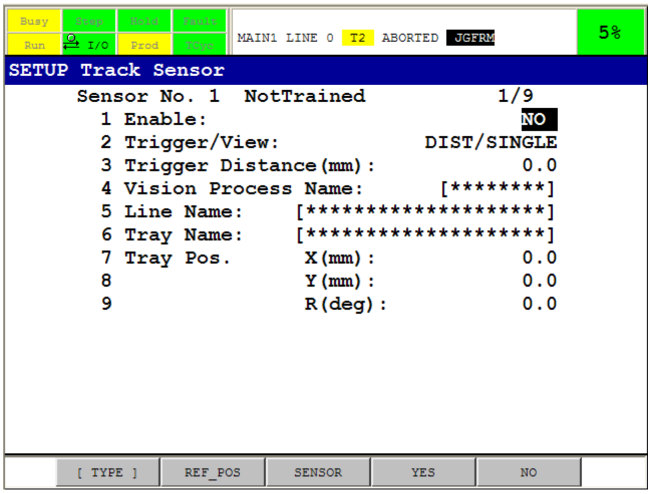
When you want to output the number of workpieces that are held by the work area AREA1 to R[11], you call the following instruction.

```
CALL VSTKNPRT('AREA1',11)
```

4.4 SETTING OF SENSOR TASK

This section describes how to setup the sensor task.
If there are multiple controllers on the same line, setup the sensor task only for robot controllers that install the camera or the sensor such as the photo eye.

When [6 SETUP] is selected from [MENUS] on the teach pendant and [Track Sensor] from F1, [TYPE], a screen as shown below appears.



Selecting a task

Up to four sensor tasks can be defined for each controller. The sensor task number currently displayed appears in the upper-left of the screen. Press F3, [SENSOR] and select the sensor task to be set.

Enable Sensor

This item specifies whether the sensor task selected on the screen is used. When using the sensor task, specify [YES]. When not using the sensor task, specify [NO].

Trigger/View

This item is used to change the trigger type and the number of camera views. Place the cursor on [DIST/SINGLE] of [2 Trigger/View], and press F4 [SELECT]. The popup menu shown below appears. Select the trigger type and the number of camera views. The setting screen changes depending on the item selected.

There are five options.

DIST/SINGLE

Each time the conveyor moves the specified distance, a vision process is executed once to find workpieces traveling on the conveyor.

DI/NONE

A sensor such as a photo eye installed on the conveyor finds workpieces traveling on the conveyor. Sensor task detects a rising edge (OFF→ON).

DI/SINGLE

Upon the input of a DI signal from a sensor such as a photo eye installed on the conveyor, a vision process is executed once to find workpieces traveling on the conveyor. Sensor task detects a rising edge (OFF→ON)

DI/DOUBLE

When the conveyor moves the specified distance after a DI signal is input from a sensor such as a photo eye installed on the conveyor, the first vision process is executed once. When the conveyor moves the specified distance again, the second vision process is executed once. The found results of the two vision processes are combined to calculate a workpiece position. Use this option when handling a large workpiece that cannot fit in a single camera's field of view. Sensor task detects a rising edge (OFF→ON). Please refer to Chapter 6, “MULTI-VIEW” for a detail.

HDI/NONE

A sensor such as a photo eye installed on the conveyor finds workpieces traveling on the conveyor. This option offers higher precision than the DI option but can be used only once per controller.

Trigger Distance (mm)

This item specifies the distance in mm that the conveyor moves before a vision process is executed.

Vision Process Name

When using *i*RVision, enter the name of the vision process used in this sensor task.

Line Name

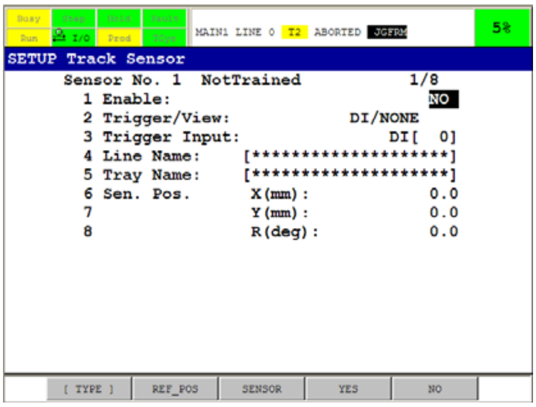
This item specifies the name of the line to which the information about found workpieces by the sensor task are sent.

Tray Name

This item specifies the name of a tray pattern when the traveling workpiece is a tray.

When [DI/NONE] is selected as the [Trigger/View]

In this case, the following screen appears.



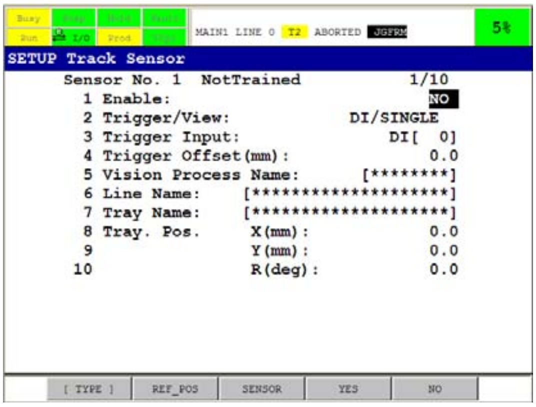
Trigger Input

This item specifies the number of the DI port to which the signal of a sensor such as a photo eye is to be input.

> ⚠ **CAUTION**
> You can set up trigger input number in line tracking schedule on a teach pendant. But set it up in the sensor task setup screen only.

When [DI/SINGLE] is selected as the [Trigger/View]

In this case, the following screen appears.

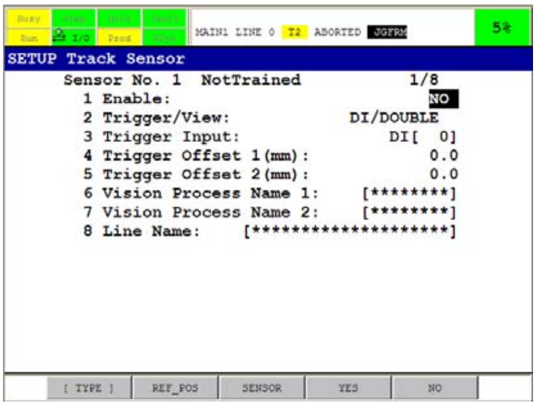


Trigger Offset (mm)

When the conveyor moves the distance specified here after the input of the DI signal, the vision system attempts to find workpieces. Use this item when the camera is mounted at a distance from the sensor such as a photo eye.

When [DI/DOUBLE] is selected as the [Trigger/View]

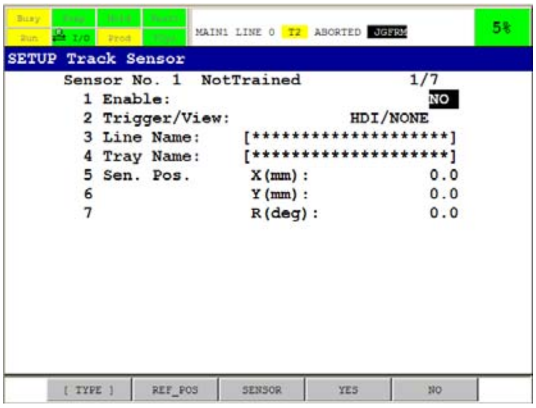
In this case, the following screen appears.



Please refer to Chapter 6, “MULTI-VIEW” for a detail.

When [HDI/NONE] is selected as the [Trigger/View]

In this case, the following screen appears.

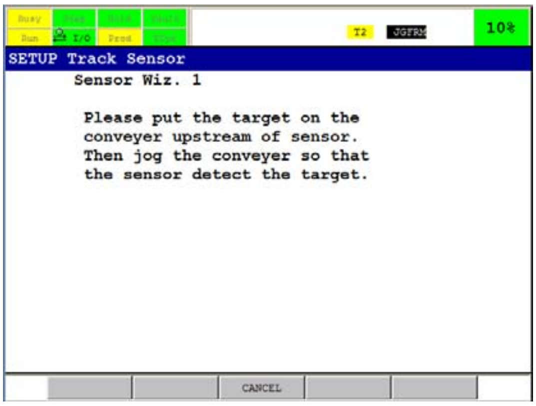


4.4.1 Setting of a Sensor Position

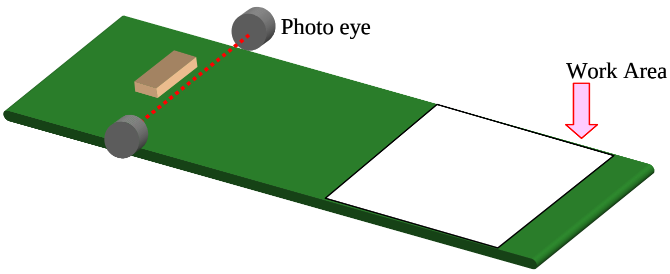
For a sensor task that uses no vision system ([DI/NONE], [HDI/NONE]), teach the position on which the sensor such as a photo eye is installed. The procedure slightly differs depending on whether a tray is used.

4.4.1.1 When no tray is used

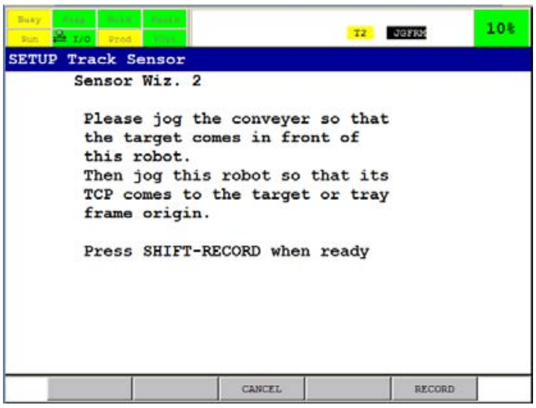
1 Move the cursor to [Sensor Pos. X(mm)].
2 Stop the conveyor.
3 Press F5 [SEN_WIZ]. A screen as shown below appears.



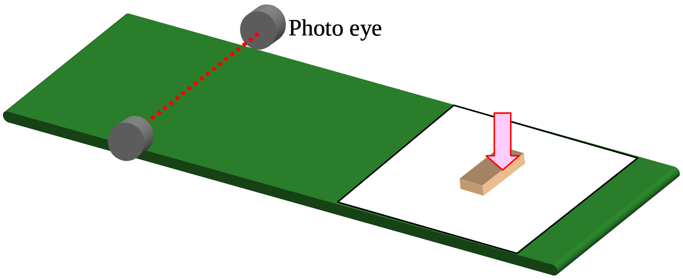
4 Place a workpiece upstream of the sensor such as a photo eye and move the conveyor to make the workpiece pass in front of the photo eye.



5 When the sensor detects the workpiece, a screen as shown below appears.



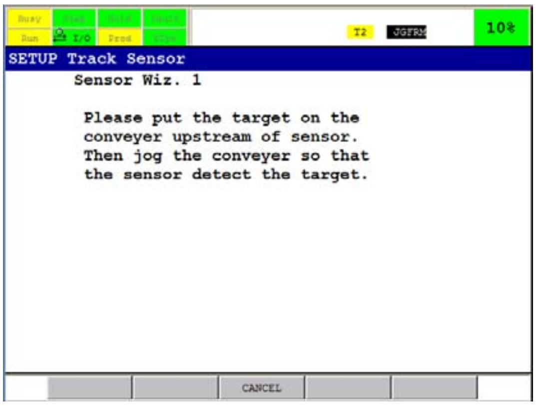
6 Move the conveyor until the workpiece positions in front of the robot.
7 Jog the robot and touch up the end on the downstream side of the workpiece, which is detected point by the sensor, with the TCP.



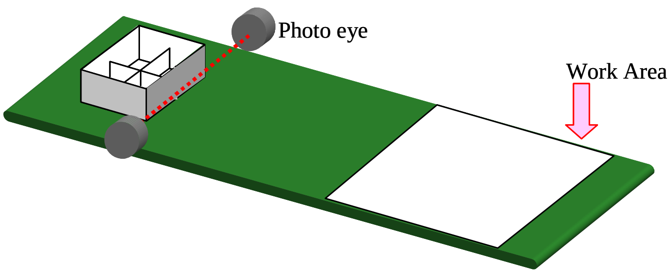
8 Press the F5 [RECORD] while holding the shift key.
9 Only the X value of [Sensor Pos] is set. (The Y and R values remain at 0.)

4.4.1.2 When a tray is used

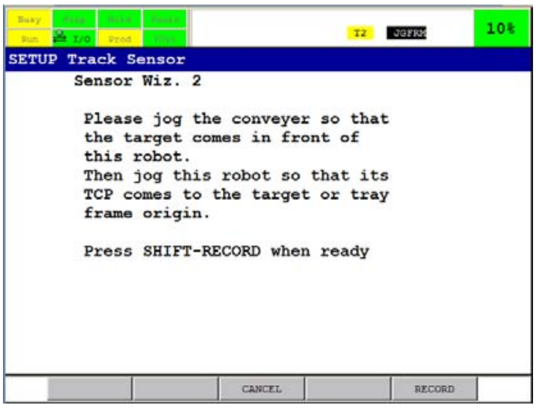
1 Move the cursor to [Sensor Pos. X(mm)].
2 Stop the conveyor.
3 Press F5 [SEN_WIZ]. A screen as shown below appears.



4 Place the tray upstream of the sensor such as a photo eye and move the conveyor to make the tray pass in front of the sensor.

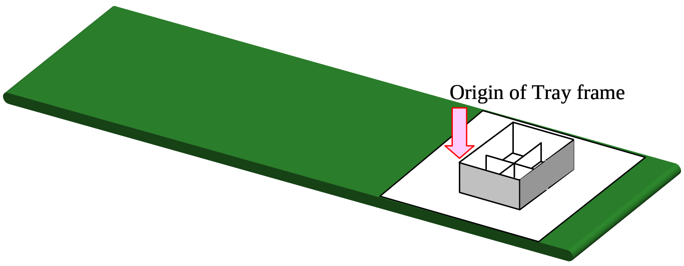


5 When the sensor detects the tray, a screen as shown below appears.

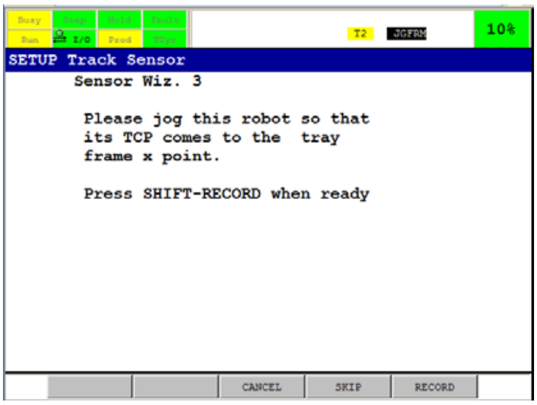


6 Move the conveyor until the tray positions in front of the robot.

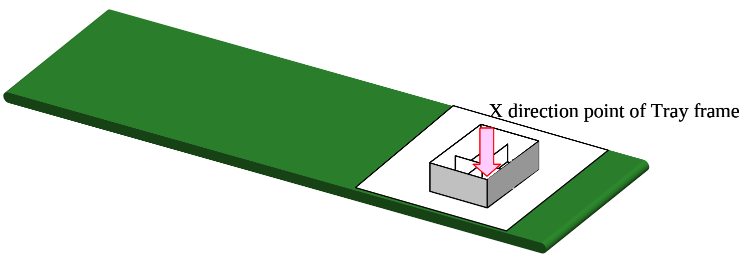
7 Jog the robot and touch up the origin of the tray frame.



8 Press the F5 [RECORD] while holding the shift key. A screen as shown below appears.



9 Jog the robot and touch up the X direction point of the tray frame. If the X direction of the tray frame is the same as one of the tracking frame, you can skip this step.



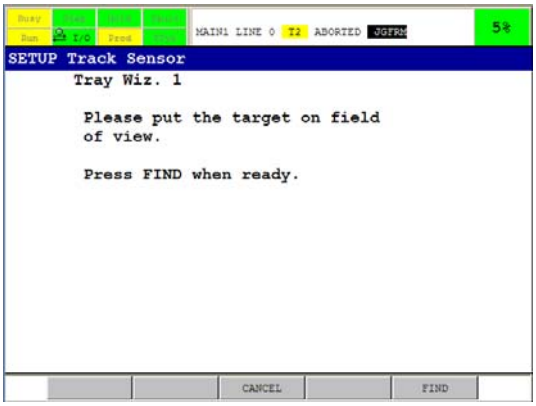
10 The X, Y, and R values of the tray position are set.

4.4.2 Setting of a Tray Position

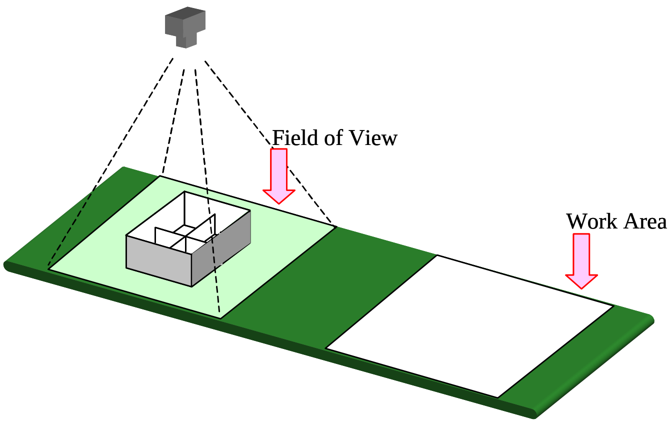
For a sensor task that uses a vision system and tray, set the position of the tray relative to the tracking frame. It is needed that camera calibration and teaching vision process are completed in advance.

4.4.2.1 [DIST/SINGLE] is used

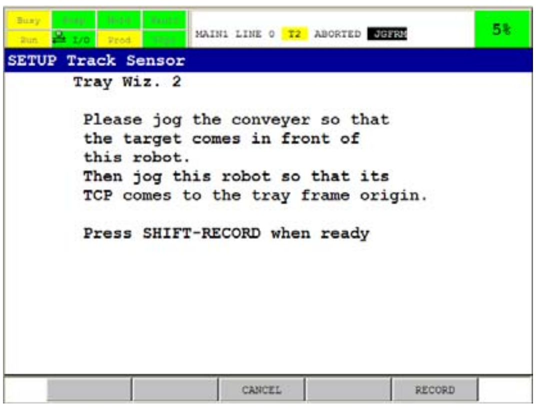
1 Move the cursor to [Tray Pos. X(mm)].
2 Stop the conveyor.
3 Press F5 [TRAY_WIZ]. A screen as shown below appears.



4 Place the tray in the field of view of the camera and press F5 [FIND].

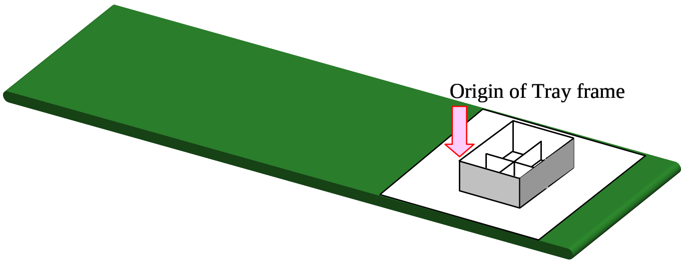


5 The sensor detects the tray and a screen as shown below appears.



6 Move the conveyor until the tray positions in front of the robot.

7 Jog the robot and touch up the origin of the tray frame.



8 Press the F5 [RECORD] while holding the shift key.
9 A screen as shown below appears.


MAIN1 LINE 0 T2 ABORTED JGFRM 5%

SETUP Track Sensor

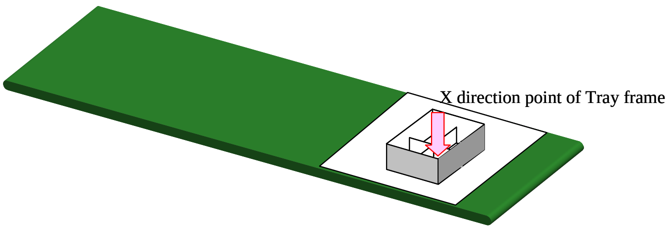
Tray Wiz. 3

Without moving the conveyer,
please jog this robot so that
its TCP comes to the tray
frame x point.

Press SHIFT-RECORD when ready

CANCEL RECORD

10 Jog the robot and touch up the X direction point of the tray coordinate system.

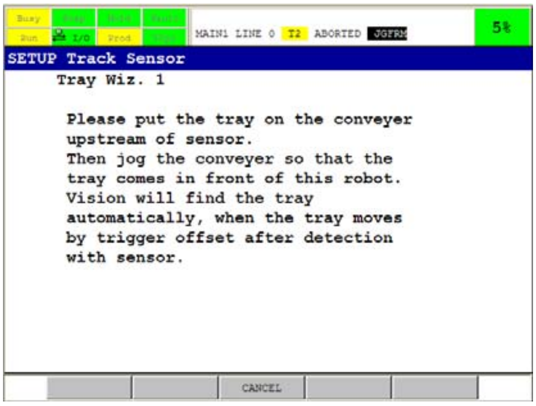


11 Press the F5 [RECORD] while holding the shift key.
12 The X, Y, and R values of the tray position are set.

4.4.2.2 [DI/SINGLE] is used

1 Move the cursor to [Tray Pos. X (mm)].
2 Stop the conveyor.

3 Press F5 [TRAY_WIZ]. A screen as shown below appears.

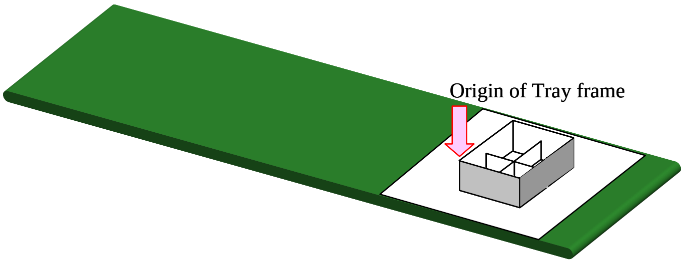


4 Let the tray travel from the uppermost section of the conveyor. When the conveyor moves by the trigger offset after the sensor such as a photo eye detects workpieces, *i*RVision automatically finds the workpieces. The conveyor does not need to be stopped when *i*RVision finds the workpieces.

5 When iRVision succeeds in finding the workpieces, a screen as shown below appears.

6 Move the conveyor until the tray is positioned in front of the robot, and stop.

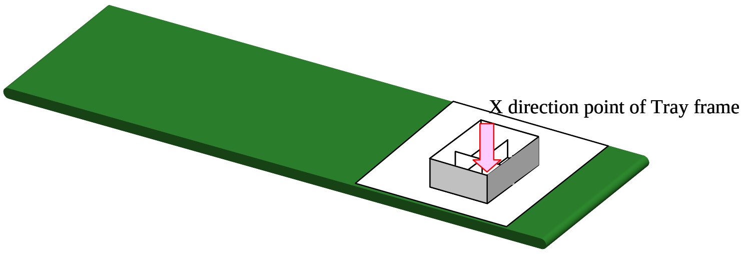
7 Jog the robot and touch up the origin of the tray frame.



8 Press the F5 [RECORD] while holding down the Shift key.

9 A screen as shown below appears.

10 Jog the robot and touch up the X direction point of the tray frame.



11 Press the F5 [RECORD] while holding down the Shift key.
12 The X, Y, and R values of the tray position are set.

4.5 SETTING THE REFERENCE POSITION

The reference position is the position of a workpiece when a robot positions are taught. Although the reference position is set in a normal *i*RVision system, it is necessary to set the reference position common to all robots that work on the same conveyor and set the same trigger, particularly in visual tracking.

The reference position setting function on the sensor task menu can be used to easily set the reference position and trigger common to all robots.

NOTE

In the vision process, if [Model ID] is selected in [Ref. Data To Use], the reference position setting and the robot position teaching have to be executed for each model ID. Repeat the following steps for each model ID.

4.5.1 Setting the Reference Position

The reference position is set on the setting menu of the sensor task. Robot position teaching has to be performed while the workpiece position is kept unchanged on the conveyor.

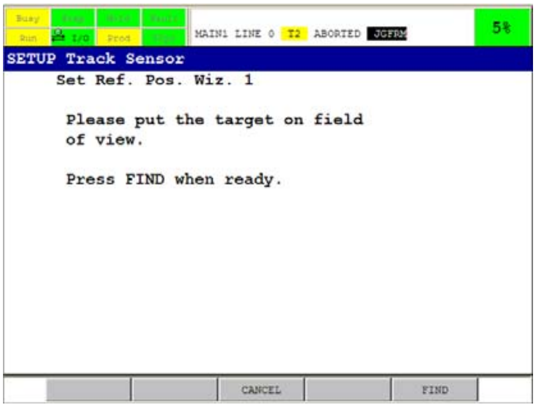
> ⚠ **CAUTION**
> 1 Do not move the workpiece on the conveyor until robot position teaching is completed after the reference position is set. When there are multiple work areas on the line, do not move the workpiece until robot position teaching is completed in all work areas. If the workpiece is accidentally moved, repeat the procedure from the beginning with the setting of the reference position.
> 2 When there are multiple robots, check the encoder schedule menu of each controller when the conveyor is stopped to see whether the [Current Count (cnts)] values of all the controllers match. If they do not match, cycle power on all the controllers with the conveyor stopped.

> **NOTE**
> When tracking both the pickup position and placement position, perform reference position setting and robot position teaching with a program to perform a pickup motion first. Then, move the conveyor, have the robot pick up the workpiece through a tracking motion, and perform reference position setting and robot position teaching with a program that performs a placement motion.

The setting procedure slightly differs depending on whether a vision sensor is used with a sensor task.

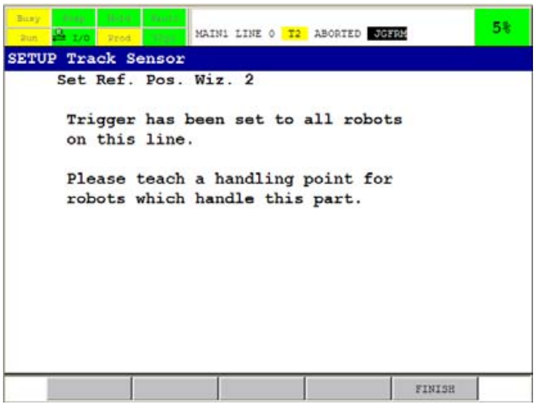
4.5.1.1 [DIST/SINGLE] is used

1 Stop the conveyor.
2 Press F2 [REF_POS]. A screen as shown below appears.



3 Place a workpiece approximately the center of the field of view of the camera.
4 Press F5 [FIND].

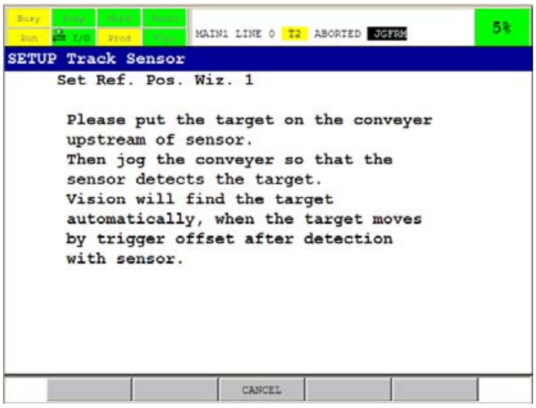
5 When iRVision detects the workpiece, a screen as shown below appears.



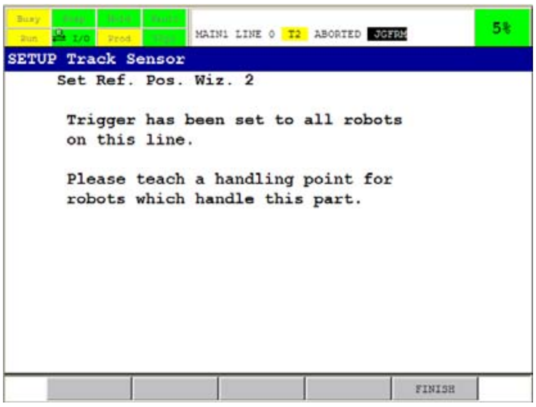
6 Press F5 [FINISH]. Now, the reference position has been set.
7 Next, perform robot position teaching. It is introduced in Subsection 4.5.2, “Robot Position Teaching”.

4.5.1.2 [DI/SINGLE] is used

1 Stop the conveyor.
2 Press F2 [REF_POS]. A screen as shown below appears.



3 Let the workpiece travel from the uppermost section of the conveyor. When the conveyor moves by the trigger offset after the sensor such as a photo eye detects the workpiece, *i*RVision automatically finds the workpiece. The conveyor does not need to be stopped when *i*RVision finds the workpiece.
4 When *i*RVision succeeds in finding the workpiece, a screen as shown below appears.



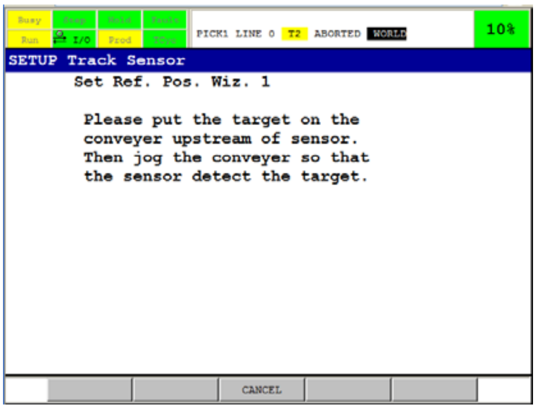
5 Press F5 [FINISH]. This completes the setting of the reference position.
6 Next, perform robot position teaching. It is introduced in Subsection 4.5.2, “Robot Position Teaching”.

4.5.1.3 [DI/DOUBLE] is used

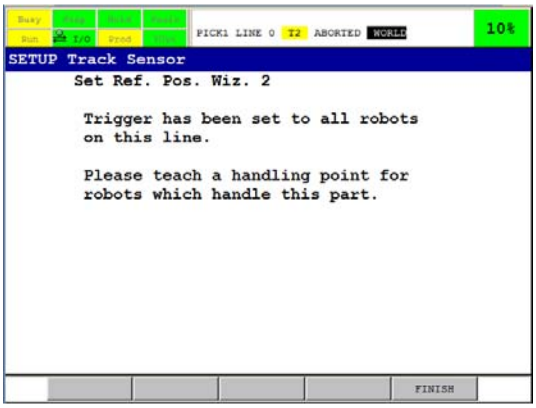
Please refer to Chapter 6, “MULTI-VIEW” for detail.

4.5.1.4 [DI/NONE], [HDI/NONE] is used

1 Stop the conveyor.
2 Press F2 [REF_POS]. A screen as shown below appears.



3 Place a workpiece upstream of the sensor such as a photo eye and move the conveyor to make the workpiece pass in front of the sensor.
4 When the sensor detects the workpiece, a screen as shown below appears.



5 Press F5 [FINISH]. Now, the reference position has been set.
6 Next, perform robot position teaching. It is introduced in Subsection 4.5.2, “Robot Position Teaching”.

4.5.2 Robot Position Teaching

After setting the reference position, perform robot position teaching.

1 When the workpiece enters the tracking area after the reference position is set on the setting menu of a sensor task, stop the conveyor.
2 From the program list, open the tracking program.
3 Check that user frame and tool frame used in the tracking program are selected.

4 Jog the robot to teach the pickup position (or the placement position).

> ⚠ **CAUTION**
> 1 Teach a pickup position (or a placement position) in a tracking program in order to teach a position represented by tracking frame. Even though a position register is used as a pickup position (or a placement position), do not teach the position at the position register screen. Place the cursor on the row number of the motion instruction moving a robot tool to the pickup position (or the placement position) and press the F5(TOUCHUP) key while pressing the SHIFT key when the robot tool is at the position. At this time, you are asked whether to subtract the vision register value. Select [NO].
> 2 When using Trays, teach the placement (or pickup) position for the cell 1. The positions of cell 2 and subsequent cells are automatically calculated based on the Tray pattern, and the robot is moved to each cell by VOFFSET instruction.
> 3 Since 5-axis robot cannot move under a full control of its tool attitude, there are some attitudes which the robot cannot reach. For this reason, the face plate surface faces to the vertical direction at the pickup position (or the placement position). For details, refer to the “Motion of 5-axis robot” in the subsection, “Motion Format” in the section, “MOTION INSTRUCTIONS” in the chapter, “PROGRAM STRUCTURE” in the “R-30*i*B CONTROLLER OPERATOR’S MANUAL (Basic Operation)”.

> **NOTE**
> As mentioned before, the positions of the approach point and retract point are created with respect to the taught pickup position (or placement position) by using Tool_Offset instruction, so these positions need not be taught.

5 When there are multiple work areas on the Line, repeat the above steps for each work area.

> **NOTE**
> When the grippers of robots have the same shape, position teaching for the second and subsequent robots may be omitted, and the taught position for the first robot can be manually copied to the other robots.

4.6 FINE ADJUSTMENT OF A TRACKING MOTION

This section explains how to optimize the tracking motion of a robot. The robot position teaching has to be completed before this section.

Adjust the position as the robot is actually tracking.

If the handling precision has an error, carefully investigate in what condition the displacement occurs, to what direction the displacement occurs, and how much the displacement occurs. Then pinpoints the cause. It is also important to implement appropriate measures to fix the cause. Otherwise, it cannot be expected to improve the handling precision.

The following subsections explain the coping strategy according to the cause.

4.6.1 First Fine Adjustment

After robot position teaching, the adjustment explained below must be performed first. Convey a workpiece several times to fine-adjust the taught position. Perform this adjustment for the individual robots.

1 Insert a WAIT instruction immediately after the pickup position in the tracking program.
  The reason for inserting the WAIT instruction is to check for a displacement visually. A wait period of approximately 30 seconds should be enough.
2 Move the conveyor at the operating speed.
3 Start the main program as in ordinary operation, and convey one workpiece on the conveyor.

> ⚠ **CAUTION**
> When using a camera to detect workpieces, place a workpiece in the same direction as when the reference position was set.

4 When the robot reaches exactly above the position at which the workpiece is to be picked up, stop the conveyor. Leave the program in the execution state.
5 After the conveyor stops completely, stop the program temporarily.
6 Check for a displacement between the position of the robot and the position of the workpiece.
7 If a displacement is found, jog the robot to the correct position, and touch up the position. Then place the cursor on the row number of the motion instruction moving a robot tool to the pickup position (or the placement position) and make a position modification. At this time, you are asked whether to subtract the offset value. Select [YES].
8 Delete the inserted WAIT instruction from the tracking program.

4.6.2 Adjustment of Delay in DI Signal Recognition

If there is a time lag from the moment at which a workpiece moves across the sensor such as a photo eye until the robot controller recognizes the signal from the sensor, an error occurs in tracking motion. This error occurs in conveyors using a sensor only in the direction in which the conveyor moves.

This error can be decreased by adjusting the DI signal recognition delay time of the robot controller by following the steps explained below. Make an adjustment only for robot controllers that input this DI signal.

> ⚠ **CAUTION**
> If [DI/SINGLE] or [DI/DOUBLE] is used, you need not to adjust a DI signal recognition delay because the DI delay does not have an affect on tracking accuracy in these cases.

1 Insert a motion instruction and a WAIT instruction immediately after the approach position in the tracking program temporarily.

```
L PR[R[3:GRIPPER]] 4000mm/sec FINE VOFFSET,VR[1] Tool_Offset,PR[15]
  WAIT 10.00(sec)
```

  The reason for inserting the motion instruction is to stop the robot motion immediately above a picking up position. Set a small Z value to PR[15] (the tool offset of an approach point for pickup motion. X, Y, W, P and R are zero).
  The reason for inserting the WAIT instruction is to check for a displacement visually. A wait period of approximately 10 seconds should be enough.
2 Set the system variable $ENC_IOD_ENB[encoder-number] to TRUE.

3 Set the system variable $ENC_ IODELAY[encoder-number] to 0.
4 Start the conveyor at the operating speed.
5 Start the main program as in ordinary operation, and convey one workpiece on the conveyor.
6 When the robot reaches exactly the approach position, stop the conveyor. Leave the program in the execution state.
7 After the conveyor stops completely, stop the program temporarily.
8 Check for a displacement between the position of the robot and the position of the workpiece.
9 In $ENC_IODELAY[encoder-number], input a value(ms) calculated as follows: Displacement amount (mm) ÷ conveyor speed (mm/s). The conveyor speed is the speed at which the workpiece moves across the sensor. The sign of the displacement amount is positive if the robot is displaced from the workpiece in the upstream direction of the conveyor; the sign is negative if the robot is displaced in the downstream direction. The conveyor speed is the speed at which the workpiece moves across the sensor.
10 Delete the inserted motion instruction and WAIT instruction from the tracking program.

Example:

When the conveyor speed at which the workpiece moves across the sensor is 500 mm/s, and the robot position is displaced from the workpiece in the upstream direction of the conveyor by 3 mm:
$ENC_IODEAY[encoder-number] = 3 ÷ 500 = +6(ms)

4.6.3 Adjustment of a Tracking Frame Setting Error

When a camera is used to detect workpieces, if the tracking frame recognized by the camera and the tracking frame recognized by each robot do not match, an error occurs in the tracking motion. This error occurs as follows: If a workpiece is not rotating, the error can be corrected properly, but if the workpiece is rotating, a displacement is generated.

This error is due to incorrect setting of the tracking frame. To eliminate such an error, the tracking frame should be set up again starting from the TCP setting. However, the error might be decreased when an adjustment amount is added to the vision offset calculated by *i*RVision by following the steps explained below. Make this adjustment for each robot separately.

A sample program given here uses the position register 20 as areas for storing an adjustment amount. Modify the storage locations as necessary.

1 Insert a motion instruction and a WAIT instruction immediately after the approach position in the tracking program temporarily.

```
L PR[R[3:GRIPPER]] 4000mm/sec FINE VOFFSET,VR[1] Tool_Offset,PR[15]
  WAIT 10.00(sec)
```

The reason for inserting the motion instruction is to stop the robot motion immediately above a picking up position. Set a small Z value to PR[15] (the tool offset of an approach point for pickup motion. X, Y, W, P and R are zero).
The reason for inserting the WAIT instruction is to check for a displacement visually. A wait period of approximately 10 seconds should be enough.

2 Add the following one line after VSTKGETQ in the tracking program:

```
  CALL ADJ_OFS(1, 1, 20, 3)
```

3 The vision offset adjusted by the above instruction is stored to VR[3]. In order to use VR[3] for the vision offset, change the tracking program as follow:

```
   L PR[R[3:GRIPPER]] 4000mm/sec CNT100 VOFFSET,VR[3] Tool_Offset,PR[10]
   L PR[R[3:GRIPPER]] 4000mm/sec FINE VOFFSET,VR[3] Tool_Offset,PR[15]
     WAIT 10.00(sec)
   L PR[R[3:GRIPPER]] 4000mm/sec CNT3 VOFFSET,VR[3]
     CALL VACUUM_ON
   L PR[R[3:GRIPPER]] 4000mm/sec CNT100 VOFFSET,VR[3] Tool_Offset,PR[10]
```

4. Set PR[20] to (0, 0, 0, 0, 0, 0).
5. Start the conveyor at the operating speed.
6. Start the main program as in ordinary operation, and convey one workpiece on the conveyor. At this time, place the workpiece in the same orientation as when the reference position was set.
7. When the robot reaches exactly the approach position, stop the conveyor. Leave the program in the execution state.
8. After the conveyor stops completely, stop the program temporarily.
9. Check for a displacement between the position of the workpiece and the position of the robot.
10. If a displacement is found, jog the robot and teach the corrected position. At this time, you are asked whether to subtract the compensation value. Select [YES].
11. Restart the main program, and convey another workpiece.
    This time, place the workpiece in the orientation 180° opposite to the orientation of the workpiece placed when the reference position was set.
12. When the robot reaches exactly the approach position, stop the conveyor. Leave the program in the execution state.
13. After the conveyor stops completely, stop the program temporarily.
14. Check for a displacement between the position of the robot and the position of the workpiece.
15. If a displacement is found, measure the displacement in the X direction (the direction of conveyor movement) and in the Y direction (the direction perpendicular to the conveyor movement direction) separately. Set the adjustment amount in the X direction in X component of PR[20], and set the adjustment amount in the Y direction in Y component of PR[20]. If there is a displacement, the adjustment amount is set to a half of the displacement.
16. Delete the inserted motion instruction and WAIT instruction from the tracking program.

About ADJ_OFS

ADJ_OFS is a KAREL program that adds a specified adjustment amount to vision offset stored in a vision register. The meanings of the arguments are described below. For details, refer to the section, “VISION SUPPORT TOOLS” in the chapter, “OTHER OPTIONS” in the “*i*RVision OPERATOR’S MANUAL (Reference)”.

Argument 1: The type of the register storing offset data (Set 1 for a vision register)
Argument 2: The number of the vision register if offset data is stored in the vision register
Argument 3: The number of the position register storing the amount of adjustment
Argument 4: The number of the vision register storing adjusted offset data

You can specify the same number in arguments 2 and 4.

> **NOTE**
> The visual tracking option and the queue managed line tracking option always provide vision support tools even if the vision support tool option is not ordered.

4.6.4 Adjustment of a Dynamic Error in a Robot

When the conveyor is stopped, a robot may reach exactly above the pickup position of a workpiece; however, when the conveyor is moving, there may be a displacement. This error is due to a tracking delay in robot operation.

NOTE
Before adjusting a dynamic error, make adjustments described in Subsections 4.6.1 to 4.6.3 to make sure that no error occurs when the conveyor is stopped.

Adjust this displacement by following the steps explained below.

1 Insert a motion instruction and a WAIT instruction immediately after the approach position in the tracking program temporarily.

```
L PR[R[3:GRIPPER]] 4000mm/sec FINE VOFFSET,VR[1] Tool_Offset,PR[15]
  WAIT 10.00(sec)
```

The reason for inserting the motion instruction is to stop the robot motion immediately above a picking up position. Set a small Z value to PR[15] (the tool offset of an approach point for pickup motion. X, Y, W, P and R are zero).
The reason for inserting the WAIT instruction is to check for a displacement visually. A wait period of approximately 10 seconds should be enough.

2 Start the conveyor at the operating speed.
3 Start the main program as in ordinary operation, and convey one workpiece on the conveyor. At this time, place the workpiece in the same orientation as when the reference position was set.
4 During conveyor operation and program execution, check that the position of the robot is tracking the approach position of the workpiece accurately.
5 If a displacement is found, in $LNCFG_GRP[group number].$SOFT_DELAY, input a value(ms) calculated as follows: Displacement amount (mm) ÷ conveyor speed (mm/s). The sign of the displacement amount is positive if the robot is displaced from the workpiece in the upstream direction of the conveyor; the sign is negative if the robot is displaced in the downstream direction. The conveyor speed is the speed at which the robot tracks the workpiece.
6 Delete the inserted motion instruction and WAIT instruction from the tracking program.

Example:

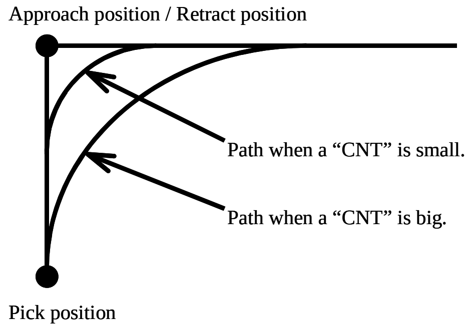
When the conveyor is moving at 500 mm/s, and the workpiece pickup position is displaced 3 mm toward the upstream of the conveyor
$SOFT_DELAY = 3 ÷ 500 = + 6 (ms)

4.6.5 Continuous Termination Type Specification for a Taught Position

Usually in a program that performs a pickup motion and a program that performs a placement motion, there are three operations. First the robot moves to an approach position, then to a pickup (or placement) position, and finally to a retract position. When Fine(fine position) or CNT(continuous) is set for these three operations, not only the robot operation path changes, but also the cycle time is affected.

Approach position and retract position

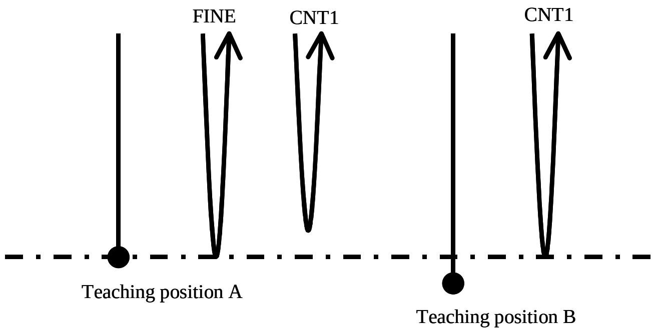
The approach position and retract position need not necessarily be passed exactly, so specifying a larger value in CNT can shorten the cycle time. A value ranging from 0 to 100 might be specified in CNT, and CNT100 produces the highest speed. However, as the value specified in CNT becomes greater, the robot operation path curves inward. When a workpiece has a certain degree of thickness, it can catch another workpiece on the conveyor as the robot turns inward. Therefore, either adjust the degree of CNT while checking a balance of the operation path and the cycle time or raise the position to allow for the rounding of the path.



Pickup position and placement position

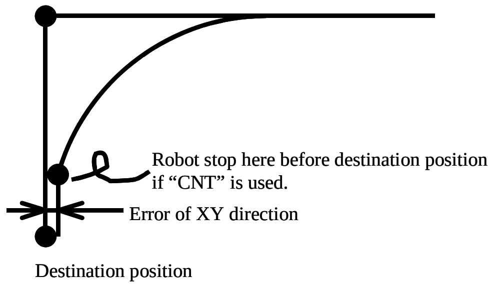
The pickup position (or placement position) must be reached exactly, so FINE instead of CNT is usually specified for the position. When FINE is specified, however, the robot operates more slowly than when CNT is specified. To improve the cycle time without affecting the path where possible, specify a small value in CNT. A value from 0 to 100 may be specified in CNT. Adjust the value to set to around 0 to 2 while checking a balance between the operation path and the operation time.

When CNT is specified, however, the robot turns inward a little even if a small value is specified, and the robot begins to move toward the retract position before reaching the taught pickup position (or placement position). When position A in the figure below is taught, for example, if FINE is specified, the robot lowers to a specified level, but if CNT is specified, the robot does not lower to the specified level and begins to rise toward the retract position.
To prevent this, teach a position slightly lower than a position that is to be actually reached, as shown in taught position B in the figure. Lowering the position is dangerous, so adjust the position in small increments.



4.6.6 Specification of a Linear Distance

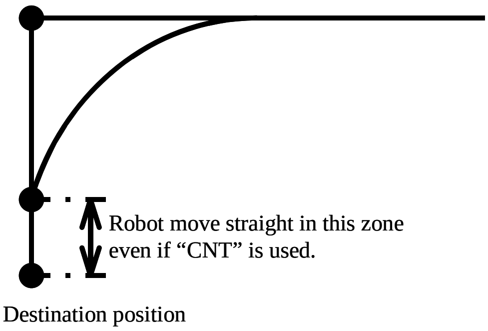
When a large value is set in CNT for the approach position and retract position, and CNT instead of FINE is specified for the pickup position (or placement position), the robot may not reach the destination position not only in height but also in the X and Y directions. The unreached distance seems to be added to the handling error.



In such a case, add a linear distance specification instruction to the operation instruction at the pickup position (or placement position).

> ⚠ **CAUTION**
> For using the linear distance specification function, the ADV-CP Path CTRL option is required separately.

When this instruction is added, the robot moves linearly only a specified distance immediately before the destination position even if CNT is specified for the preceding position.
However, because specifying a long linear distance increases the cycle time, adjust the distance while checking a balance of the robot operation and the cycle time.



A program modification example is shown below. This program is a part of the program given in Subsection 4.3.2.2, “Pick program”.

```
13: L PR[R[3:GRIPPER]] 4000mm/sec CNT100 VOFFSET,VR[1] Tool_Offset,PR[10]
14: L PR[R[3:GRIPPER]] 4000mm/sec CNT3 VOFFSET,VR[1] AP_LD100
15:   CALL VACUUM_ON
16: L PR[R[3:GRIPPER]] 4000mm/sec CNT100 VOFFSET,VR[1] Tool_Offset,PR[10] RT_LD100
```

On lines 14 and 16, the linear distance specification instruction is added. The robot definitely moves 100mm linearly before and after the destination position.

4.6.7 Time Before Function

If a signal output instruction is specified in a subprogram, this function allows a signal to be output during robot operation. It can also eliminate the wait time associated with the transfer of data to and from peripheral devices, thus reducing the cycle time.
When robots in a visual tracking system move at a high speed, the wait time associated with the transfer of data to turn on/off because of vacuum has an effect on system performance. In such a case, add the TB instruction to the pick or place motion instruction.
If the TB instruction is added, the wait time can be eliminated and the cycle time can be reduced because this function allows a signal to be output during robot operation.

A program modification example is shown below. This program is a part of the program given in Subsection 4.3.2.2, “Pick program”.

```
13: L PR[R[3:GRIPPER]] 4000mm/sec CNT100 VOFFSET, VR[1] Offset,PR[10]
14: L PR[R[3:GRIPPER]] 4000mm/sec CNT3 VOFFSET, VR[1] TB .5sec,CALL VACUUM_ON
15:
16: L PR[R[3:GRIPPER]] 4000mm/sec CNT100 VOFFSET, VR[1] Offset,PR[10]
```

On lines 14, the time before instruction is added. "VACUUM_ON" is executed before 0.5 second of a robot arriving at the destination position. Signal output can be turned on at the right time before a robot arrives at the destination position according to adjusting the specified time.

4.7 SYSTEM START-UP

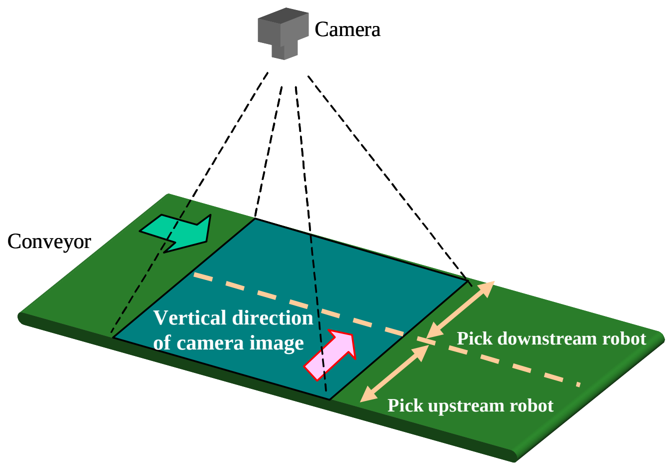
In visual tracking systems with multiple robots, start the robot controller program with the sensor task after all other robots have been started. This allows for the initialization of all work areas for each robot prior to the start of the sensor task.
When multiple robots have each sensor task, all work areas must be initialized before all sensor tasks start. For this reason, insert a WAIT instruction before VSTKSTRT in the system main program as described in Subsection 4.3.2.1, "Main program".

5 VARIATION

This chapter describes an example of a system variation, which is not described in Chapter 4.

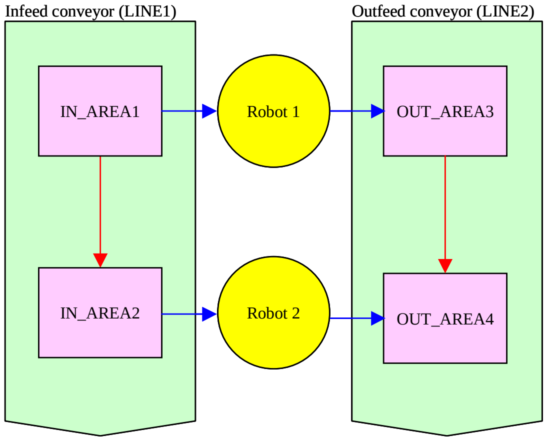
5.1 USING DIFFERENT ROBOTS TO PICK UP WORKPIECES DEPENDING ON THEIR POSITIONS ON THE CONVEYOR

This section describes the case in which different robots are used to pick up workpieces depending on their positions on the conveyor. Specifically, two robots are assumed to be available; the upstream robot picks up the workpieces on the right half of the conveyor and the downstream robot picks up the workpieces on the left half of the conveyor. A camera is used to sense workpieces on the conveyor.



In addition, the following assumptions are made.

- Install the camera so that the horizontal direction of the camera image is parallel to the movement direction of the conveyor. That is, the width direction of the conveyor is aligned with the vertical direction of the camera image. In *i*RVision, Row indicates the coordinate in the vertical direction of the camera image.
- Set Model ID to 1 in GPM locator tool. If the Row of the detected workpiece is greater than 240, set the model ID to 2 by the conditional execution tool. (When you use the digital camera, the Row of the middle of the conveyor on screen is the half of vertical size of [Mode] of Kowa Digital Camera setup page. If you select 1/3" VGA (640x480) as [Mode], this is 240. When you use the analog camera, this is half of the vertical size of [Image Size] of Camera Parameters.)
- Define the configuration of work areas and Lines as shown below.
  This configuration is the same as configuration 1 shown in Section 2.3, “SAMPLE SYSTEM CONFIGURATIONS”.

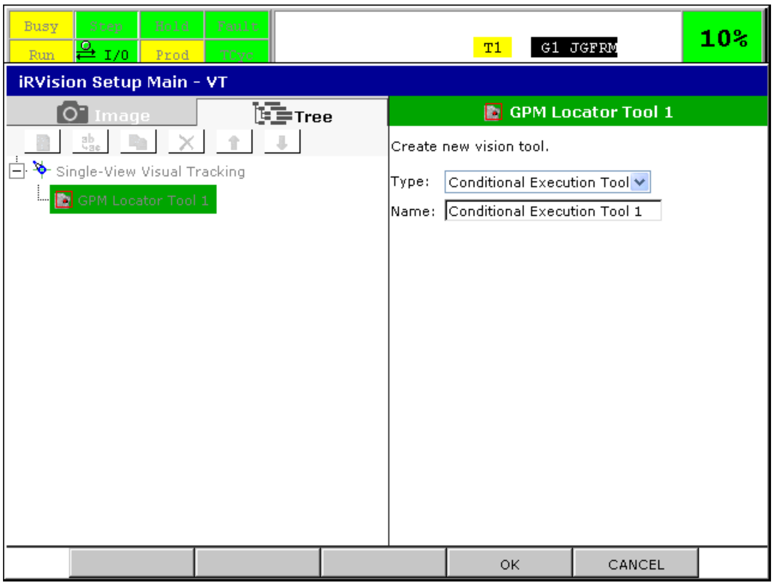


IN_AREA1 and IN_AREA 2 are the work areas of the infeed conveyor and OUT_AREA3 and OUT_AREA4 are the work areas of the outfeed conveyor.

Inserting the conditional execution tool

The following procedure inserts the conditional execution tool into the vision process.

1 In the tree view at the left of the teach window, tap [GPM Locator Tool 1] and click the button. A screen as shown below appears.

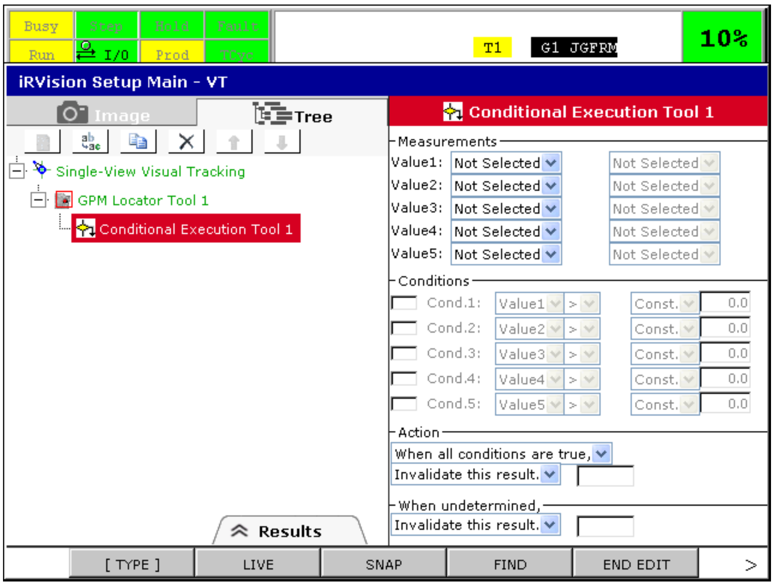


2 Select [Conditional Execution Tool] in the [Type].
3 Enter a tool name such as "Conditional Execution Tool 1" in the [Name] field and press F4 OK button.

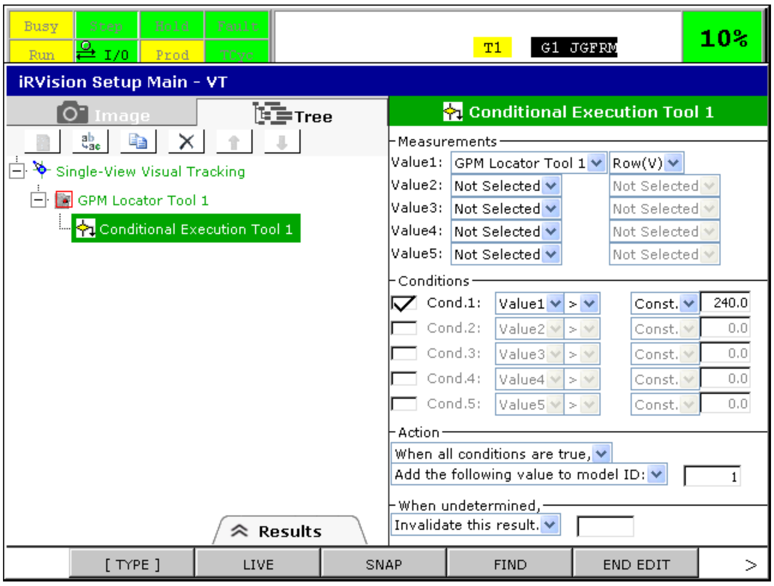
Setting the conditional execution tool

The following procedure makes settings so that 1 is added to the model ID if Row is greater than 240.

1 Select [Conditional Execution Tool 1] in the tree view. A screen as shown below appears.



2 Under [Measurements], select [GPM Locator Tool 1] and [Row(V)] next to [Value1].
3 Under[Conditions], check [Cond.1] and select [Value1], [>], [Const.], and [240].
4 Under [Action], select [Add the following value to model ID], and specify 1.

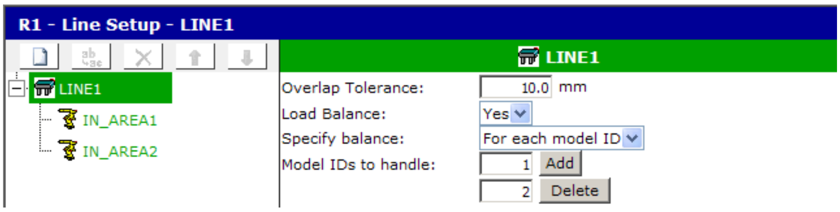


Setting a load balance

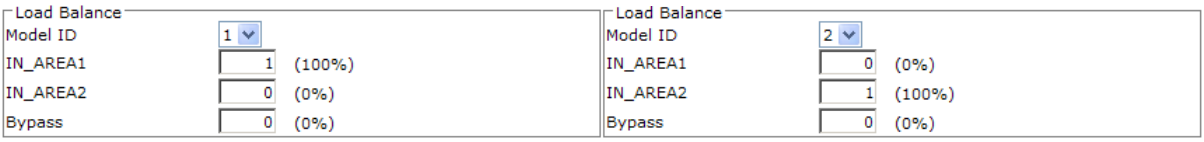
The following procedure sets the load balance of a Line so that all workpieces with a model ID of 1 are handled by the upstream robot and all workpieces with a model ID of 2 are handled by the downstream robot.

Open the Line setup page and set the load balance. In the following figure, only the infeed conveyor is indicated.

1 Open the line setup page. A screen as shown below appears.

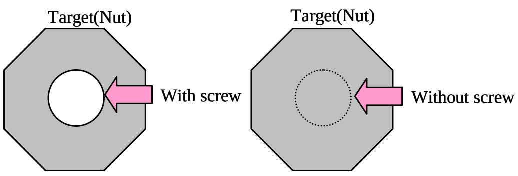


2 Select [Yes] in [Load Balance].
3 Select [For each model ID] in [Specify balance].
4 In the [Model IDs to handle] fields, enter 1 and 2.
5 Under [Load Balance], perform setup so that all workpieces with a model ID of 1 are handled by the upstream robot and all workpieces with a model ID of 2 are handled by the downstream robot.



5.2 USING DIFFERENT ROBOTS TO PICK UP WORKPIECES DEPENDING ON THEIR CHARACTERISTICS

This section describes the case in which different robots are used to pick up workpieces depending on the characteristics of the workpieces. Specifically, the conveyor is assumed to send two types of workpieces: one with a hole and the other with no hole. The delivery destination is switched depending on whether a hole is present. The following figure assumes that the workpieces are two types of nuts.



In addition, the following assumptions are made.

- Whether a hole is present is determined by measuring the standard deviation of brightness with the histogram tool. When the surface of a nut is homogeneous, the standard deviation of brightness is large if a hole is present or the standard deviation is small if no hole is present.

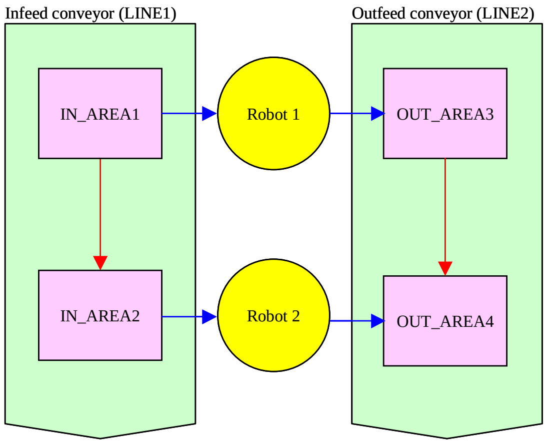
	With a hole	With no hole
Standard deviation	Large value	Small value

> ⚠ **CAUTION**
> The above method may not be used depending on the background. This method is only an example.

> **NOTE**
> Whether a hole is present can be determined by the blob tool or pattern match tool. This section introduces the method of identifying a hole with a histogram.

- The model ID of the workpiece taught to the GPM locator tool is assumed to be 1.
- If a hole is present, the conditional execution tool sets the model ID to 2.
- The work areas and Lines are defined as shown in the figure below.

This configuration is the same as configuration 1 shown in Section 2.3, “SAMPLE SYSTEM CONFIGURATIONS”.

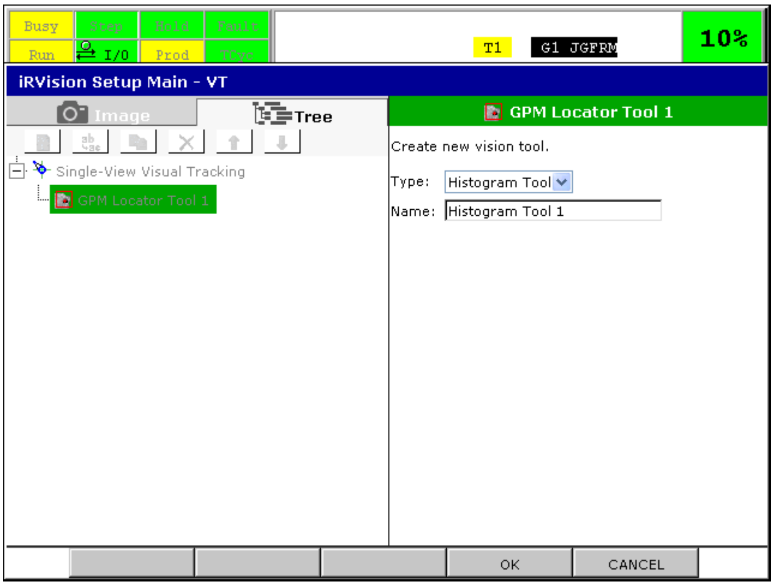


IN_AREA1 and IN_AREA 2 are the work areas of the infeed conveyor and OUT_AREA3 and OUT_AREA4 are the work areas of the outfeed conveyor.

Inserting the histogram tool

The following procedure inserts the histogram tool into the vision process.

1 In the tree view at the left of the teach window, tap [GPM Locator Tool 1] and click the button. A screen as shown below appears.



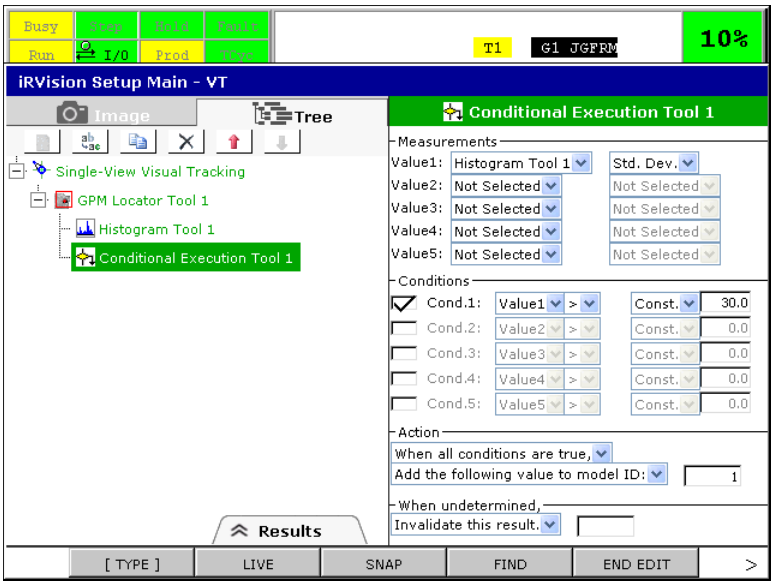
2 Select [Histogram Tool] in the [Type].
3 Enter a tool name such as “Histogram Tool 1” in the [Name] field and press F4 OK.
4 Tap [Histogram Tool 1] above the tree view.
5 Click the [SET] button to teach [the histogram].
6 Press F10 SAVE.

Setting the conditional execution tool

The following procedure sets the conditional execution tool so that the model ID is set to 2 if the workpiece has a hole. For how to insert the conditional execution tool, see Section 5.1, “USING

DIFFERENT ROBOTS TO PICK UP WORKPIECES DEPENDING ON THEIR POSITIONS ON THE CONVEYOR".

If the standard deviation is approx. 50 when the workpiece has a hole and approx. 10 when the workpiece has no hole, the appropriate threshold is 30, which is the middle of them. The following procedure sets the conditional execution tool so that 1 is added to the model ID if the standard deviation of a histogram is 30 or more.

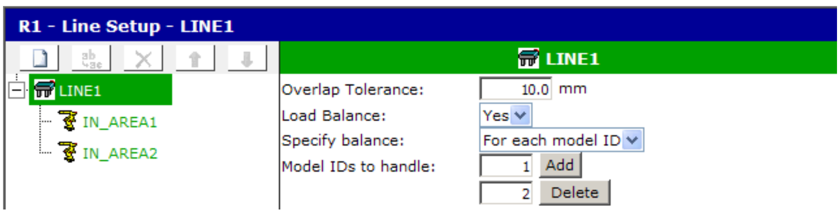


1 Next to [Value1] under [Measurements], select [Histogram Tool 1] and [Std. Dev.].
2 Under [Conditions], check [Cond.1] and select [Value1], [>], [Const.], and [30].
3 Under [Action], select [Add the following value to model ID] and set 1 in the field to the right.

Setting a load balance

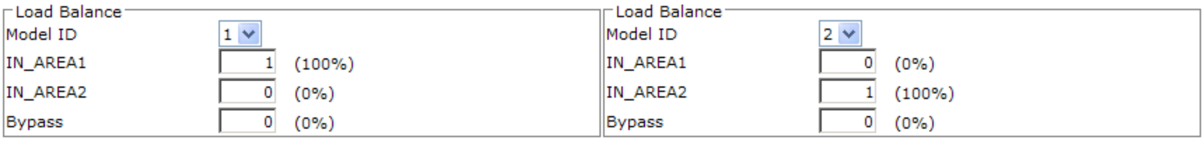
The following procedure sets the load balance of a Line so that all workpieces with a model ID of 1 are handled by the upstream robot and all workpieces with a model ID of 2 are handled by the downstream robot.
Open the Line setup page and set the load balance. In the following figure, only the infeed conveyor is indicated.

1 Open the Line setup page. A screen as shown below appears.



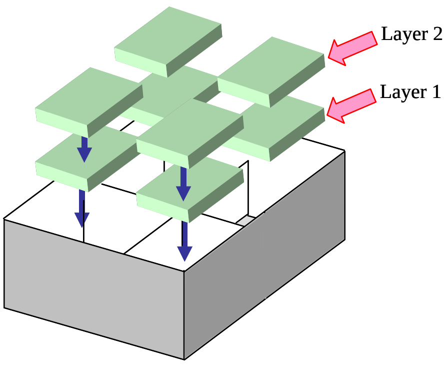
2 Select [Yes] in the [Load Balance].
3 Select [For each model ID] in [Specify balance].
4 Enter 1 and 2 in [Model IDs to handle].

5 Be sure that all the workpieces with a model ID of 1 are handled by the upstream robot and all the workpieces with a model ID of 2 are handled by the downstream robot under [Load Balance].



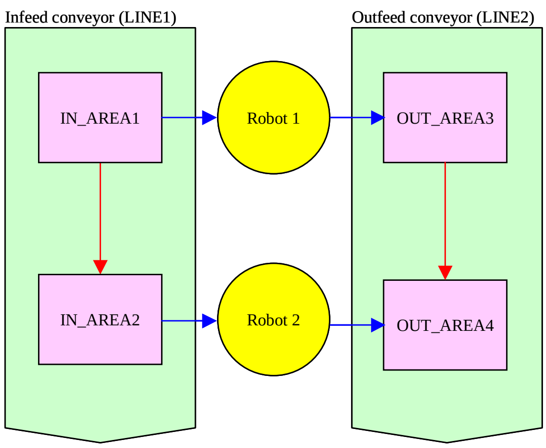
5.3 PLACING WORKPIECES IN LAYERS

This section describes the case in which workpieces are placed in two layers. Four workpieces is placed in each layer. Tray is used in this case.



In this case, workpiece cannot be placed in the upper layer unless the lower layer is filled with four workpieces. To observe this rule, it is necessary to specify [Seq] number in the Tray pattern setup page so that workpiece cannot be placed in the upper layer unless the lower layer is filled with four workpieces.

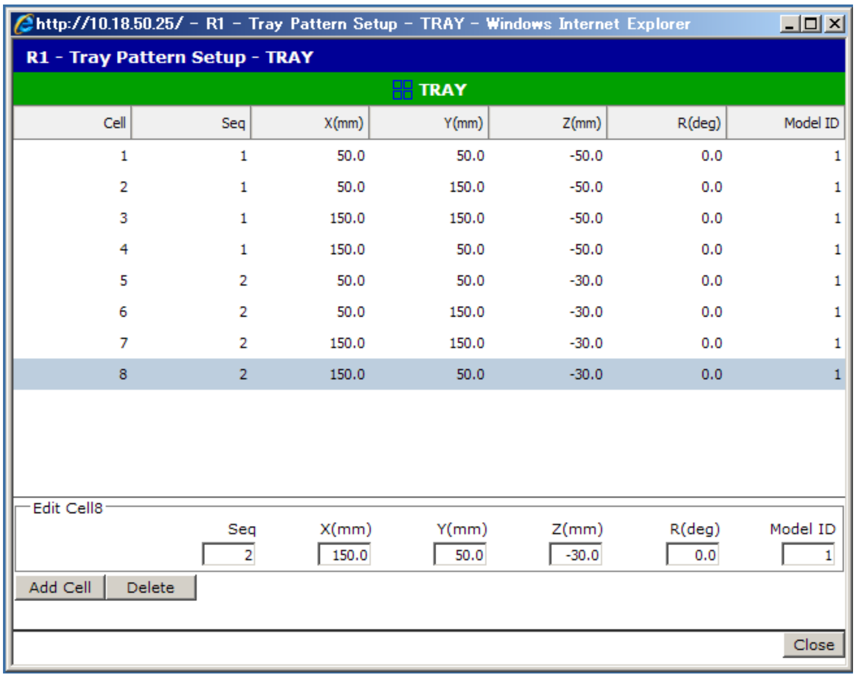
Work areas and Lines are assumed to be set as shown below. This configuration is the same as the configuration 1 shown in Section 2.3, “SAMPLE SYSTEM CONFIGURATIONS”.



IN_AREA1 and IN_AREA 2 are the work areas of the infeed conveyor and OUT_AREA3 and OUT_AREA4 are the work areas of the outfeed conveyor.

Setting a Tray

The Tray pattern is shown in the figure below.

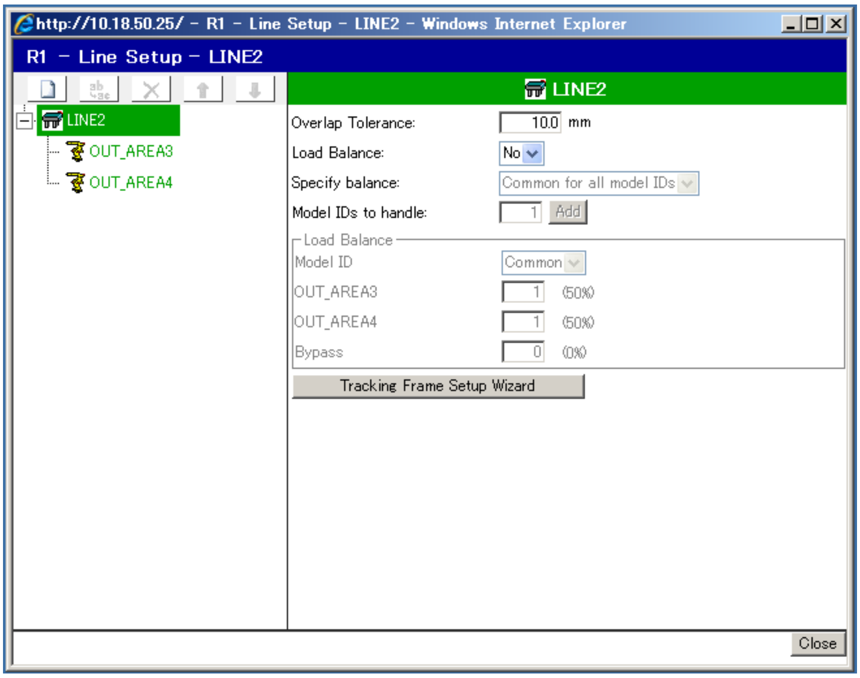


Cells 1 to 4 with seq 1 are set to be placed in the first layer and cells 5 to 8 with seq 2 are set to be placed in the second layer. If the Tray pattern is set as shown above, cells 5 to 8 in the second layer are not allocated to robots unless the first layer is filled with cells.

The cells in the same layer must be set to the same seq. In such a setting, workpieces are placed in the order from downstream to upstream in the same layer.

Setting a Line

A Line is set as shown in the figure below. The settings only for the outfeed conveyor are shown here. The settings for the infeed conveyor do not need to be changed.



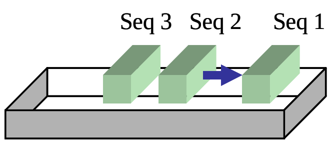
If the load balance is disabled, the robot places as many workpiece as possible on the cell in the specified order.

NOTE

The load balance is disabled in this example, but it is possible to enable it. In this case, the cell information is allocated in the specified order according to the specified load balance.

5.4 PLACING WORKPIECES IN SEQUENCE FROM THE END OF A BOX

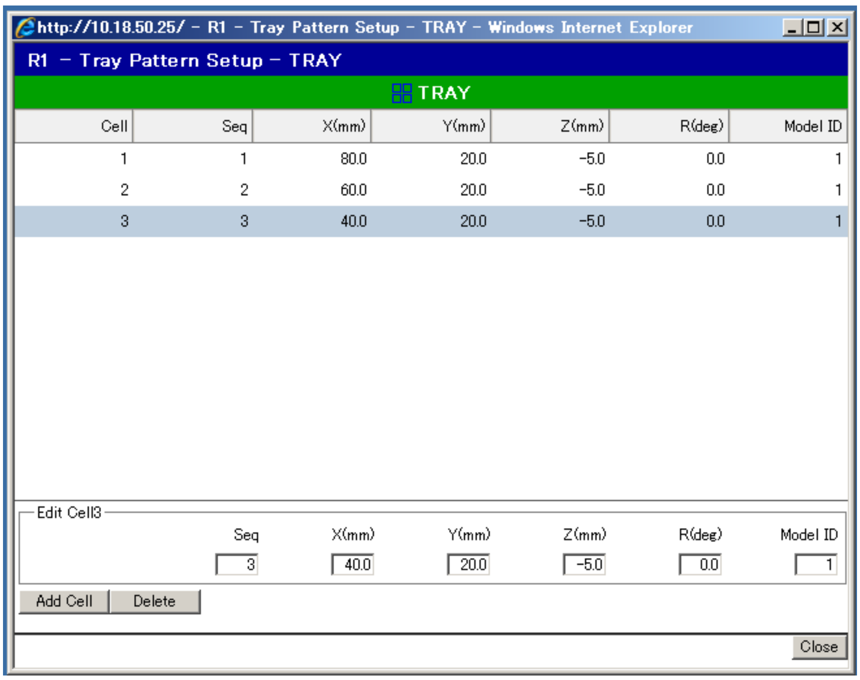
This section describes the case in which workpieces are placed in sequence from the end of a box that is not separated.



To place workpieces from the end, even when there is only one layer in the box, specify [Seq] number in a Tray pattern setup page so that a workpiece is not placed in the adjacent cell unless the end cell is filled with a workpiece.

Example of setting a Tray pattern

To place workpieces in sequence from the end, perform setup as shown below.



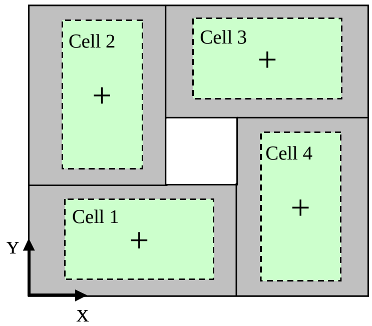
The [Seq] fields of cells 1 to 3 are set to in order. Cell 1 is located at the end. In these settings, workpieces are placed in the Tray in the order of cell 1, cell 2, and cell 3.

Setting a Line

Perform setup as in Section 5.3, “PLACING WORKPIECES IN LAYERS”.

5.5 PLACING WORKPIECES IN DIFFERENT DIRECTIONS IN A BOX

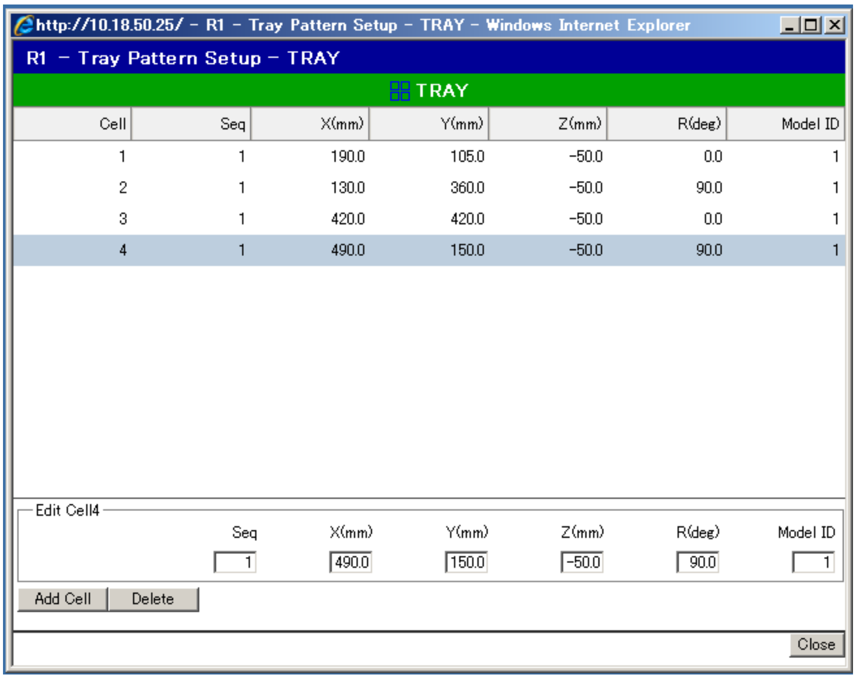
This section describes the case in which workpieces are placed in different directions for each cell. For example, the direction of cell 2 and cell 4 differs by 90° from that of cell 1 and cell 3 in the figure below.



In this case, only the settings of the Tray pattern need to be changed.

Setting a Tray

An example of setting a Tray pattern is shown below.



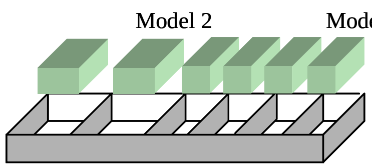
The angle of cell 1 is assumed to be 0° and the angles of the other cells are represented, relative to cell 1. The angle of cell 2 and cell 4 is 90°.

Position teaching for the robot is performed only in cell 1 as in a normal Tray. Since it is indicated in the Tray pattern that the direction of cell 2 and cell 4 differs by 90° from that of cell 1 and cell 3, the robot automatically changes the direction 90° when placing a workpiece in cell 2 or cell 4.

5.6 PLACING WORKPIECES IN A DIFFERENT PLACE OF A BOX DEPENDING ON MODEL ID

This section describes the case in which two types of workpieces are placed in different places of a box depending on their types.

As shown in the figure below, there are big and small workpieces and there are big cells for a big workpiece and small cells for a small workpiece. In this case, cell in which a workpiece placed is determined depending on the model ID of a workpiece picked up.

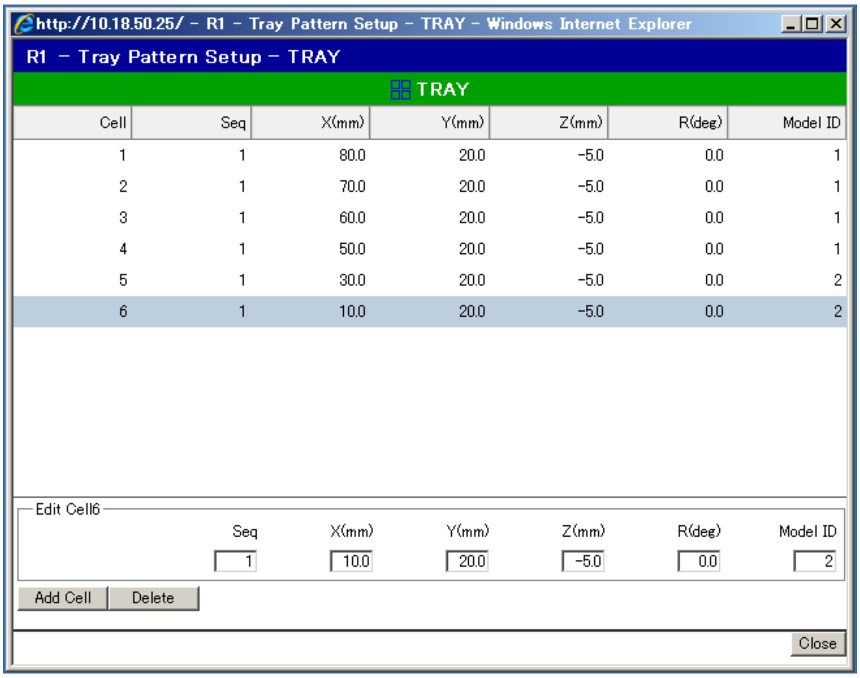


To place workpieces this way, set the model IDs for cells when setting the Tray pattern and specify the model ID of cells to be delivered during a call of VSTKGETQ.PC.

It is assumed for the infeed conveyor that the model ID of small workpieces is set to 1 and the model ID of big workpieces is set to 2.

Setting a Tray pattern

An example of setting a Tray pattern is shown below.



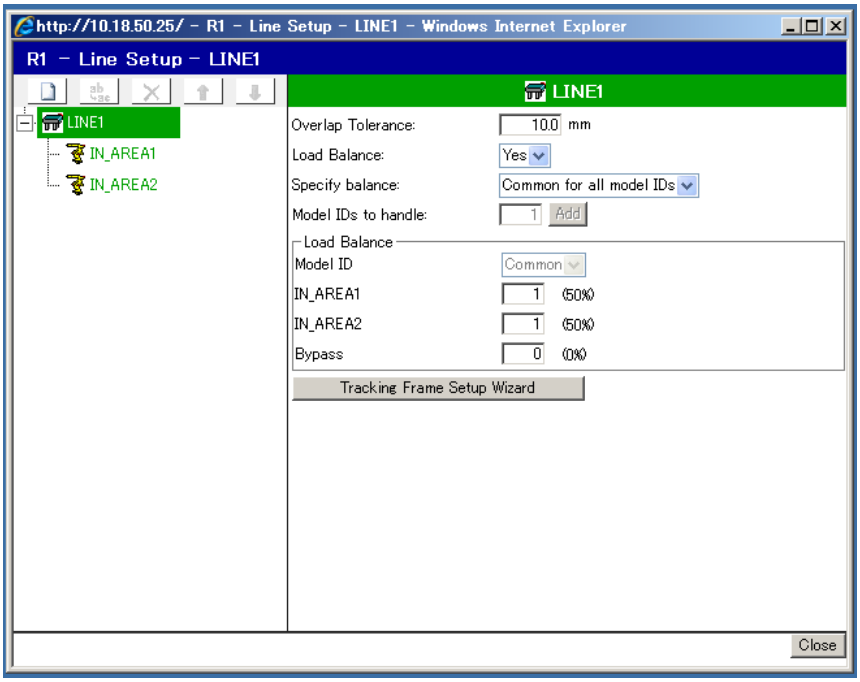
Cell	Seq	X(mm)	Y(mm)	Z(mm)	R(deg)	Model ID
1	1	80.0	20.0	-5.0	0.0	1
2	1	70.0	20.0	-5.0	0.0	1
3	1	60.0	20.0	-5.0	0.0	1
4	1	50.0	20.0	-5.0	0.0	1
5	1	30.0	20.0	-5.0	0.0	2
6	1	10.0	20.0	-5.0	0.0	2

Cells 1 to 4 are cells for a small workpiece and cells 5 and 6 are cells for a big workpiece.
There are two cell model IDs (1 and 2), which are the same as workpiece model IDs (1 and 2) for the infeed conveyor. A cell with a model ID of 1 is placed in a cell with a model ID of 1 and a cell with a model ID of 2 is placed in a cell with a model ID of 2. It is not necessary to set the cell model IDs this way, but this makes a workpiece model ID and its corresponding cell model ID identical.

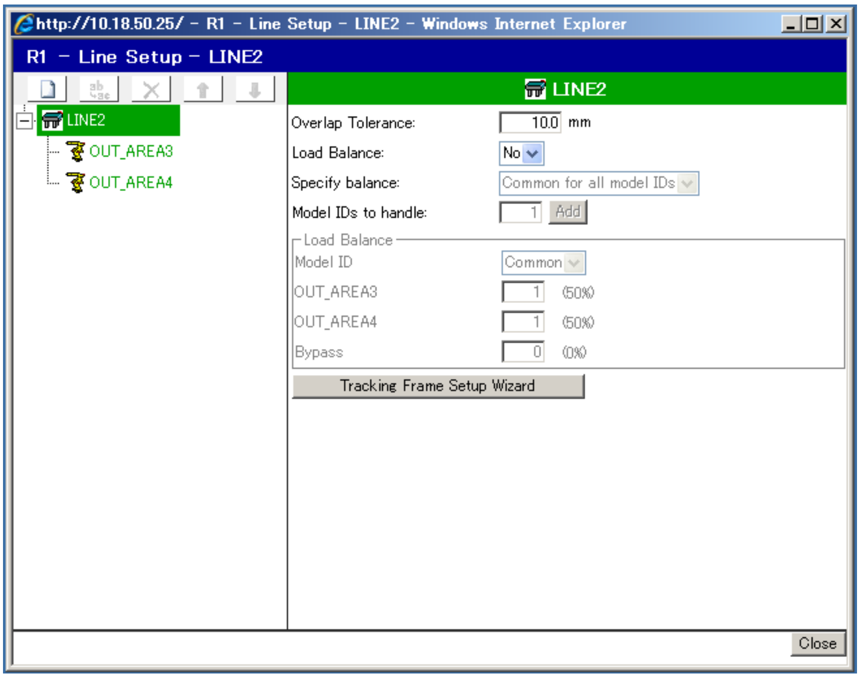
All cells need to be set in the same [Seq] number. When different [Seq] number is given to cells, even if a cell with a model ID of 1 is required, a cell with a model ID of 1 may not be delivered because of [Seq] number constraints.

Setting a Line

An example of setting a Line is shown below.



Infeed conveyor



Outfeed conveyor

For the infeed conveyor, select [Common for all model IDs] in [Specify balance] and set the load balance to 1:1. Therefore, each robot handles two types of traveling workpieces equally regardless of their types.

For the outfeed conveyor, select [No] in [Load Balance]. This makes each robot wait for the cells corresponding to the model IDs of picked workpieces and place them in as many places as possible.

Example of changing the main program

In this case, the program for placement motion by a robot needs to be customized. The main program and the pickup motion program do not need to be modified.

Modify the program shown in Subsection 4.3.2.3, "Placement program" as shown below. The major changes are indicated by an underline.

DROP1.TP

```
  1:    UTOOL_NUM=0
  2:    UFRAME_NUM=0
  3:
  4:    LBL[100]
  5:    STOP_TRACKING
  6:    R[5:MODELID]=VR[1].MODELID
  7:    CALL VSTKGETQ('OUT_AREA3', 2, 100, 2, 1, R[5:MODELID])
  8:    IF R[2:GETQ_STATUS]=0,JMP LBL[200]
  9:    IF R[1:CYCLE_STOP]=1,JMP LBL[900]
 10:    JMP LBL[100]
 11:
 12:    LBL[200]
 13: L PR[11] 4000mm/sec CNT100 VOFFSET,VR[2] Tool_Offset,PR[12]
 14: L PR[11] 4000mm/sec CNT3 VOFFSET,VR[2]
 15:    CALL VACUUM_OFF
 16: L PR[11] 4000mm/sec CNT100 VOFFSET,VR[2] Tool_Offset,PR[12]
 17:    CALL VSTKACKQ('OUT_AREA3', 2, 1)
 18:
 19:    LBL[900]
[End]
```

On line 6, the model ID of the workpiece picked from the infeed conveyor is stored in R[5]. VSTKGETQ.PC on line 7 adds an argument and specifies R[5] as a model ID. This ensures that the cell with the model ID specified in R[5] is delivered.

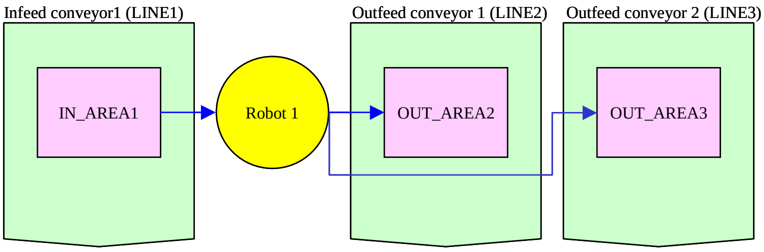
> ⚠ **CAUTION**
> If there is no cell that corresponds to a picked workpiece, the robot cannot place this workpiece. To operate such a system normally, the rate at which workpieces 1 and 2 travel on the infeed conveyor must match the rate at which cells where workpieces 1 and 2 are placed.

5.7 PLACING WORKPIECES ON A DIFFERENT CONVEYOR DEPENDING ON MODEL ID

This section describes the case in which there are multiple outfeed conveyors and, depending on the type of a workpiece picked up from the infeed conveyor, the outfeed conveyor on which the workpiece is to be placed is selected.

Consider the case where there is one infeed conveyor and two outfeed conveyors, and each of these conveyors has one work area, as shown in the figure below.

The model IDs of workpieces that travel on the infeed conveyor are 1 and 2; the workpieces with a model ID of 1 are placed on outfeed conveyor 1 and the workpieces with a model ID of 2 are placed on outfeed conveyor 2.



Setting a Line

Perform setup so that all workpieces on the infeed conveyor are picked up regardless of the model ID. Disable the load balance for this purpose. For the two outfeed conveyors, also disable the load balance.

Example of changing the main program

In this case, the main program of the robot needs to be customized. The DROP2.TP program, which places workpieces on the conveyor 2, is newly required. The program for pickup motion does not need to be modified.

Modify the program shown in Subsection 4.3.2.1, “Main program“ as shown below. The major changes are indicated by an underline.

MAIN1.TP

```
   R[1:CYCLE_STOP]=0
   CALL VSTKCLRQ('IN_AREA1')
   CALL VSTKCLRQ('OUT_AREA2')
   CALL VSTKCLRQ('OUT_AREA3')
L  P[1] 1000mm/sec FINE
   CALL VSTKSTRT

  LBL[100]
  CALL PICK1
  IF R[1:CYCLE_STOP]=1,JMP LBL[900]

  R[5:MODELID]=VR[1].MODELID
  IF R[5:MODELID]=2,JMP LBL[200]
  CALL DROP1
  JMP LBL[300]
  LBL[200]
  CALL DROP2
  LBL[300]
  IF R[1:CYCLE_STOP]=1,JMP LBL[900]
  JMP LBL[100]

  LBL[900]
  CALL VSTKSTOP
[End]
```

On line 12, the model ID of the workpiece picked up from the infeed conveyor is stored in R[5]. If the R[5] value is 1, DROP1.TP is called so that the workpiece is placed on the outfeed conveyor 1. If the R[5] value is 2, DROP2.TP is called so that the workpiece is placed on the outfeed conveyor 2.

An example of DROP2.TP, which performs the placement motion for the outfeed conveyor 2, is shown below.

DROP2.TP

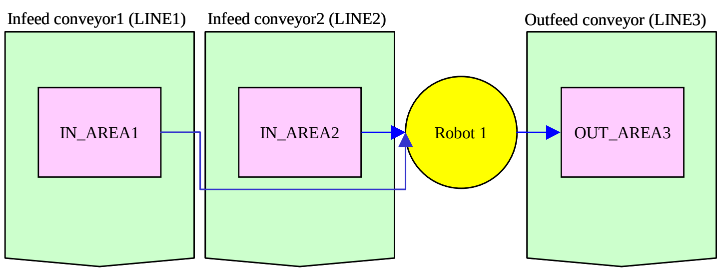
```
   UTOOL_NUM=0
   UFRAME_NUM=0

   LBL[100]
   STOP_TRACKING
   CALL VSTKGETQ('OUT_AREA3', 3, 100, 2, 1)
   IF R[2:GETQ_STATUS]=0,JMP LBL[200]
   IF R[1:CYCLE_STOP]=1,JMP LBL[900]
   JMP LBL[100]

   LBL[200]
L  PR[20] 4000mm/sec CNT100 VOFFSET,VR[3] Tool_Offset,PR[12]
L  PR[20] 4000mm/sec CNT3 VOFFSET,VR[3]
   CALL VACUUM_OFF
L  PR[20] 4000mm/sec CNT100 VOFFSET,VR[3] Tool_Offset,PR[12]
   CALL VSTKACKQ('OUT_AREA3', 3, 1)

   LBL[900]
[End]
```

This program is basically the same as the DROP1.TP program shown in Subsection 4.3.2.3, “Placement program”, except for the number of PR, which is the robot teaching position. The Lines of the two outfeed conveyors are different, so it is necessary to use a different position register for each outfeed conveyor. Here, PR[20] is used as the placement position for outfeed conveyor 2.

5.8 PICKING UP WORKPIECES FROM TWO OR MORE INFEED CONVEYORS

This section describes the case in which there are multiple infeed conveyors, workpieces are picked up from infeed conveyors, and then workpieces are placed on the outfeed conveyor. This is described in Configuration 6 in Section 2.3, “SAMPLE SYSTEM CONFIGURATIONS”.

Consider the case where there are two infeed conveyors and one outfeed conveyor, and each of these conveyors has one work area, as shown in the figure below.



The robot picks up a workpiece from these conveyors alternately. If the robot fails to pick up a workpiece, it picks up a workpiece on another conveyor.

Example of changing the main program

In this case, the pickup motion program and main program of the robot need to be customized. A program that picks up workpieces from infeed conveyor 2 is newly required.

An example of changing the MAIN1.TP program shown in Subsection 4.3.2.1, "Main program" and an example of changing the PICK1.TP program shown in Subsection 4.3.2.2, "Pick program" are shown below. The major changes are indicated by an underline.

MAIN1.TP

```
   R[1:CYCLE_STOP]=0
   R[3:GRIPPER]=1
   R[5:SELECT_CNV]=1
   CALL VSTKCLRQ('IN_AREA1')
   CALL VSTKCLRQ('IN_AREA2')
   CALL VSTKCLRQ('OUT_AREA3')
L  P[1] 1000mm/sec FINE
   CALL VSTKSTRT

   LBL[100]
   STOP_TRACKING
   IF R[5:SELECT_CNV]=2,JMP LBL[200]
   CALL PICK1
   JMP LBL[300]
   LBL[200]
   CALL PICK2
   LBL[300]
   IF R[1:CYCLE_STOP]=1,JMP LBL[900]

   R[5: SELECT_CNV]=3-R[5: SELECT_CNV]

   IF R[3:GRIPPER]<=R[4:NUM_GRIPPER] OR R[2:GETQ_STATUS]<>0,JMP LBL[100]
   R[3:GRIPPER]=1

   CALL DROP1
   IF R[1:CYCLE_STOP]=1,JMP LBL[900]
   JMP LBL[100]

   LBL[900]
   CALL VSTKSTOP
[End]
```

R[5] is used to switch between conveyors. When R[5] is 1, pickup is performed from infeed conveyor 1. When R[5] is 2, pickup is performed from infeed conveyor 2.

NOTE

If a workpiece to be picked up is present on both conveyors, the application program determines the conveyor from which a workpiece is picked up. In this program, the conveyor is alternately selected.

PICK1.TP is a tracking program for the pickup motion for the infeed conveyor 1 and PICK2.TP is a tracking program for the pickup motion for the infeed conveyor 2.

On line 22, gripper number and VSTKGETQ.PC status is checked. If R[4] is more than 1, the robot pick up workpieces from both conveyors consecutively. When gripper is filled with the workpieces, gripper number is reset on line 23.

PICK1.TP

```
  1:   UTOOL_NUM=0
  2:   UFRAME_NUM=0
  3:
  4:   CALL VSTKGETQ('IN_AREA1', 1, 100, 2, R[3:GRIPPER])
  5:   IF R[2:GETQ_STATUS]<>0,JMP LBL[900]
  6:
  7: L PR[R[3:GRIPPER]] 4000mm/sec CNT100 VOFFSET,VR[1] Tool_Offset,PR[10]
  8: L PR[R[3:GRIPPER]] 4000mm/sec CNT3 VOFFSET,VR[1]
  9:   CALL VACUUM_ON
 10: L PR[R[3:GRIPPER]] 4000mm/sec CNT100 VOFFSET,VR[1] Tool_Offset,PR[10]
 11:   CALL VSTKACKQ('IN_AREA1', 1, 1)
 12:   R[3:GRIPPER]=R[3:GRIPPER]+1
 13:
 14:   LBL[900]
[End]
```

No lines were added to the original PICK1.TP and some lines were deleted. Unlike the original PICK1.TP, if VSTKGETQ.PC on line 4 exits due to a timeout, that is, there are no workpieces in the area then, PICK1.TP terminates immediately. This calls PICK2.TP and picks up a workpiece from infeed conveyor 2.

PICK2.TP

```
  1:   UTOOL_NUM=0
  2:   UFRAME_NUM=0
  3:
  4:   CALL VSTKGETQ('IN_AREA2', 2, 100, 2, R[3:GRIPPER])
  5:   IF R[2:GETQ_STATUS]<>0,JMP LBL[900]
  6:
  7:   R[6:PR_NUM]=R[3:GRIPPER]+20
  8: L PR[R[6:PR_NUM]] 4000mm/sec CNT100 VOFFSET,VR[2] Tool_Offset,PR[10]
  9: L PR[R[6:PR_NUM]] 4000mm/sec CNT3 VOFFSET,VR[2]
 10:  VACUUM_ON
 11: L PR[R[6:PR_NUM]] 4000mm/sec CNT100 VOFFSET,VR[2] Tool_Offset,PR[10]
 12:   CALL VSTKACKQ('IN_AREA2', 2, 1)
 13:   R[3:GRIPPER]=R[3:GRIPPER]+1
 14:
 15:   LBL[900]
[End]
```

This program is basically the same as the PICK1.TP changed above. Unlike PICK1.TP, R[6] is used instead of R[3] to specify indirectly PR because the robot teaching position is different between infeed conveyors 1 and 2. PR[21 to 29] are assumed to be used as the teaching positions of infeed conveyor 2, so R[6] is set to R[3] plus 20.

5.9 REPORTING FAILURE TO PICK UP A WORKPIECE

This section describes the case in which failure to pick up a workpiece is found after the robot tried to pick up it.
In such a case, the position of a workpiece could differ from the position of it previously detected by the sensor, so set the third argument of VSTACKQ.PC to 3 to prevent any robot from picking it again.
It is only needed to change PICK1.TP introduced in Subsection 4.3.2.2, “Pick program”. An example is shown below. The major changes are indicated by an underline.

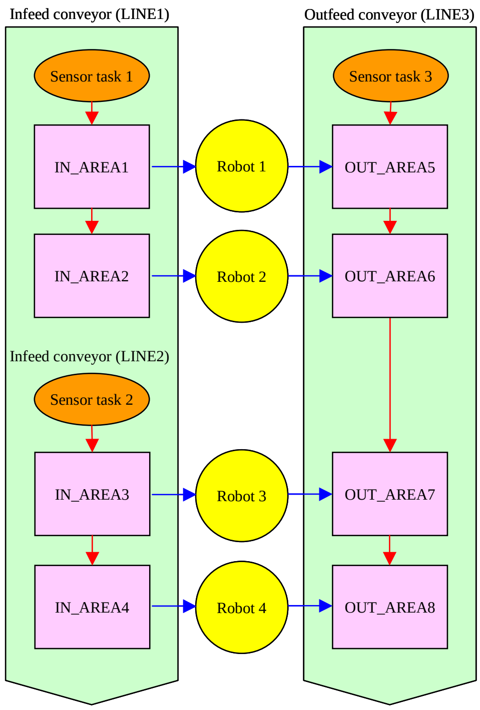
PICK1.TP

```
  1:   UTOOL_NUM =0
  2:   UFRAME_NUM =0
  3:   R[3:GRIPPER]=1
  4:
  5:   LBL[100]
  6:   STOP_TRACKING
  7:   CALL VSTKGETQ('IN_AREA', 1, 100, 2, R[3:GRIPPER])
  8:   IF R[2:GETQ_STATUS]=0,JMP LBL[200]
  9:   IF R[1:CYCLE_STOP]=1,JMP LBL[900]
 10:   JMP LBL[100]
 11:
 12:   LBL[200]
 13: L PR[R[3:GRIPPER]] 4000mm/sec CNT100 VOFFSET,VR[1] Tool_Offset,PR[10]
 14: L PR[R[3:GRIPPER]] 4000mm/sec CNT3 VOFFSET,VR[1]
 15:   CALL VACUUM_ON
 16: L PR[R[3:GRIPPER]] 4000mm/sec CNT100 VOFFSET,VR[1] Tool_Offset,PR[10]
 17:   IF RI[R[3:GRIPPER]]=ON,JMP LBL[300]
 18:   CALL VSTKACKQ('IN_AREA', 1, 3)
 19:   JMP LBL[100]
 20:
 21:   LBL[300]
 22:   CALL VSTKACKQ('IN_AREA', 1, 1)
 23:   R[3:GRIPPER]=R[3:GRIPPER]+1
 24:   IF R[3:GRIPPER]<=R[4:NUM_GRIPPER],JMP LBL[100]
 25:   LBL[900]
[End]
```

On line 17, it is checked whether a workpiece was picked up . Check RI corresponding to the number of the gripper to confirm if the picking workpiece succeeded or not. If it is successful, RI is ON. If it failed, RI is OFF. If the picking workpieces are successful, this program normally finishes after jumping to label[300]. If it failed, the program reports 3 (failure to pick up) to the system by VSTKACKQ.PC at line 18 and jumps to LBL[100]. The robot requests the next workpiece information by VSTKGETQ.PC.

5.10 ARRANGING SENSORS IN SERIES IN THE SAME CONVEYOR

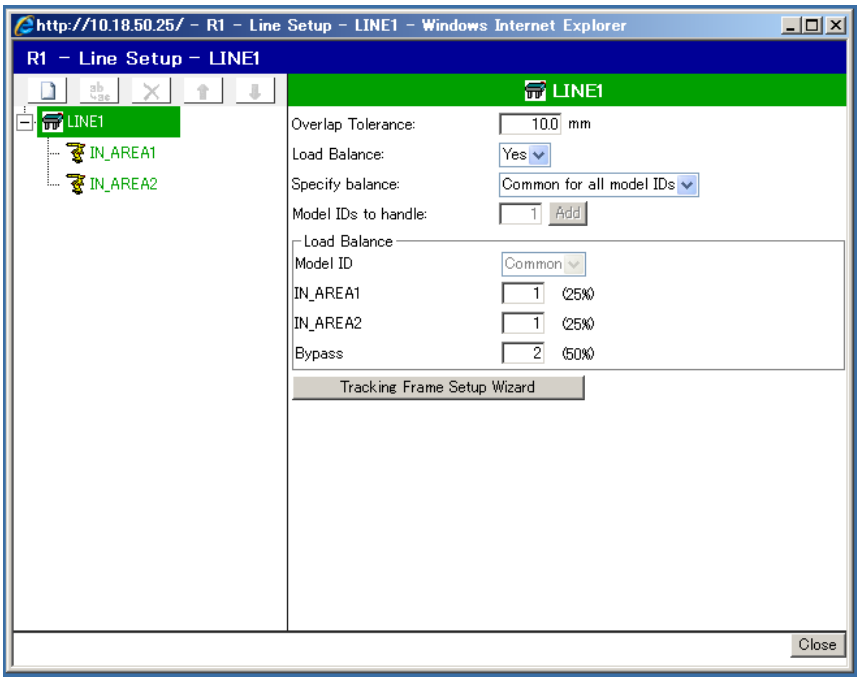
This section describes the case in which the workpieces can be transferred from work area to work area. The work areas can be distant from each other. In such a case, workpieces which have been detected upstream once are found again downstream and it is allocated to the downstream robot on a conveyor.
This case is introduced in Section 2.2, “KEY CONCEPTS”.
Consider the case where there is one infeed conveyor and one outfeed conveyor and two sensors are arranged in an infeed conveyor as shown in the figure below.



Sensors are arranged ahead of each work area (IN_AREA1, IN_AREA3) in an infeed conveyor. In such a case, it is only needed to change the Line setting.

Setting a Line

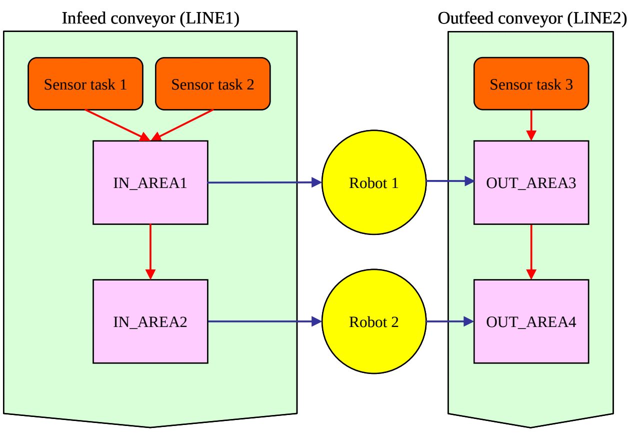
Make two Lines in an infeed conveyor. Each Line has two work areas. Setting data of LINE1 is shown below.



In LINE1 including IN_AREA1 and IN_AREA2, set [Bypass] in the setting of the load balance to 2, this makes half of the workpieces flow on the conveyor past IN_AREA1 and IN_AREA2 to IN_AREA3 and IN_AREA4.
In LINE2 including IN_AREA3 and IN_AREA4, set the load balance so that all of the remaining workpieces is handled in IN_AREA3 and IN_AREA4.

5.11 USING TWO CAMERAS TOO WIDE CONVEYOR

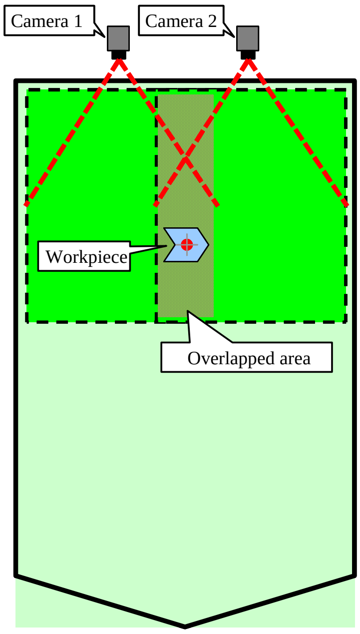
When an infeed conveyor is too wide for one camera to cover whole conveyor, we recommend to install two cameras on the infeed conveyor. One covers the right half and the other covers the left half. Configure the system so that two sensor tasks input the information of the flowing workpieces to the same work area. This case is already introduced as “sample configuration 3” in “2.3 SAMPLE SYSTEM CONFIGURATIONS”.



Suppose a tracking frame is already taught to all two robots.

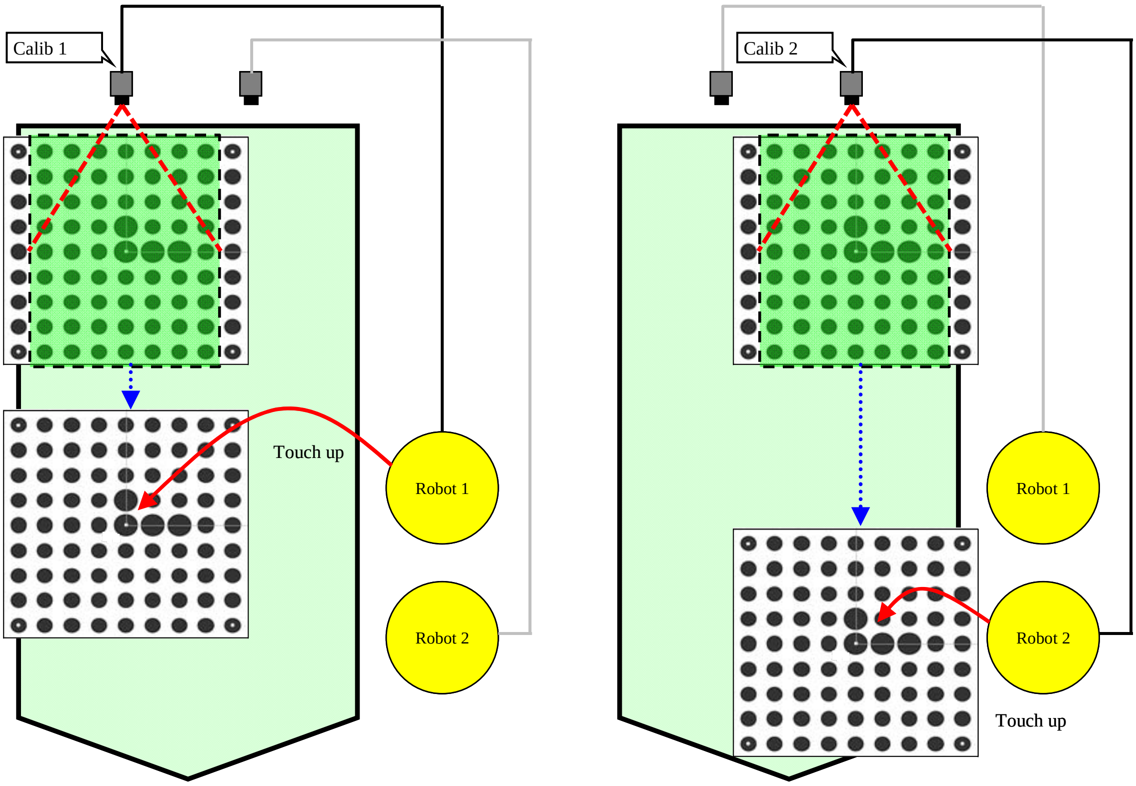
Layout

The field of view of two cameras (Camera1, Camera2) should be overlapped so that all workpieces can be found. Overlapped area should be able to include the workpiece completely.



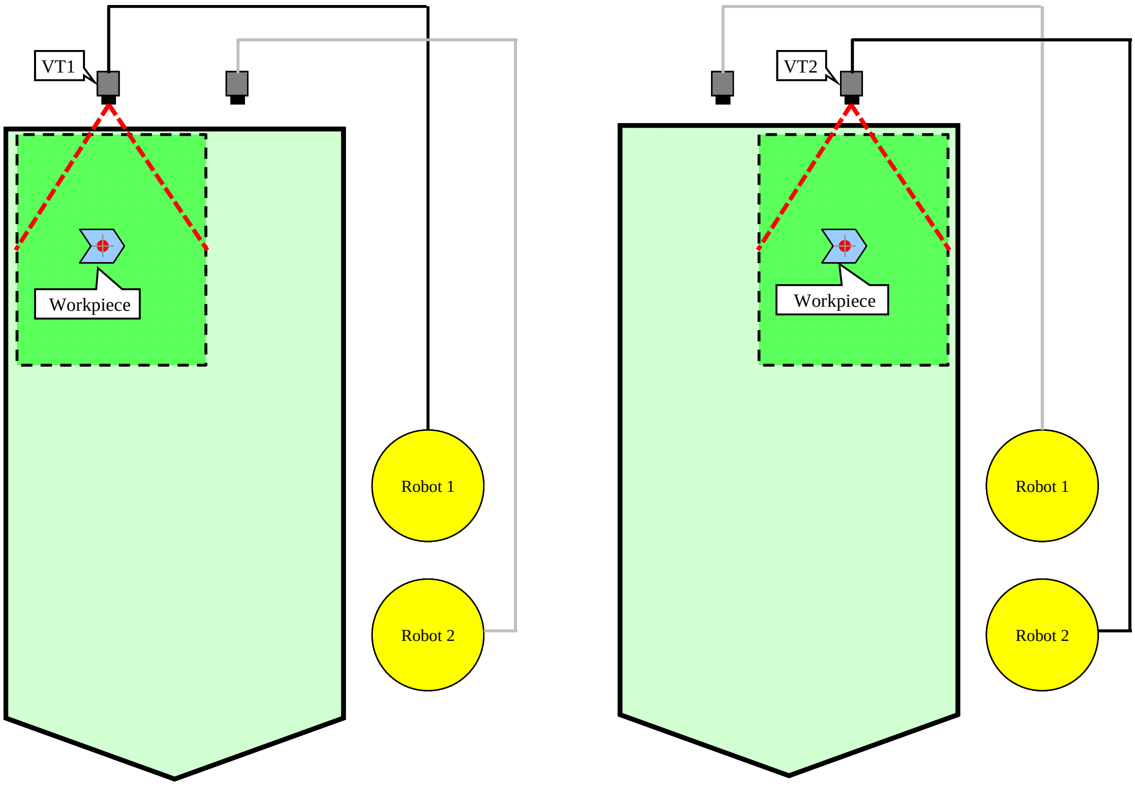
Camera calibration

Calibrate each camera with the robot that each camera is connected to.



Vision process

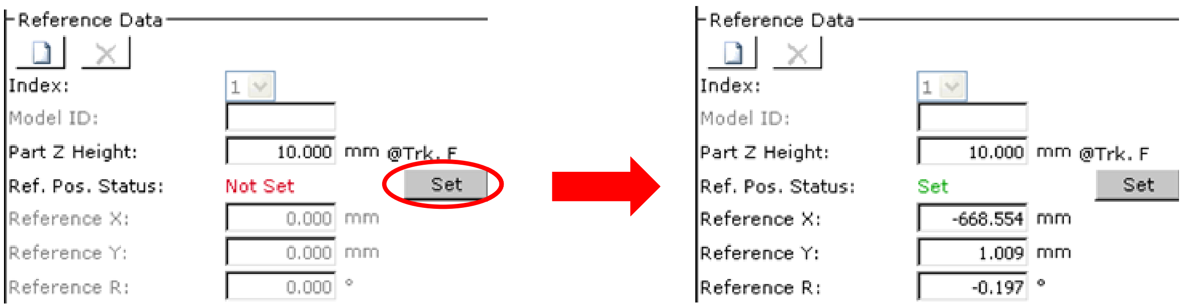
Create vision processes (VT1, VT2) on the robot controller that is connected to each camera. Those vision processes should use the above camera calibration data. The same GPM locator tool should be taught for each vision process. Also each model origin should be set at the same position on the workpiece.



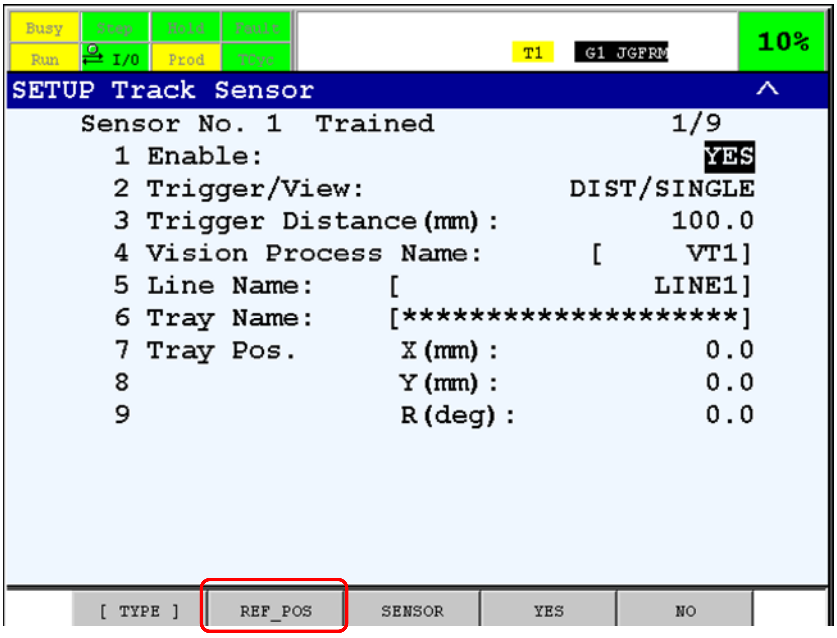
Reference position setting

The same reference position should be set to each vision process. After setting the reference data to one vision process (VT1) with standard procedure, set the same reference data to the other vision process (VT2) manually with “direct input”. Before the reference data is set (= [Ref. Pos. Status] is [not set] as the bellow figure), the reference data cannot be input directly in the vision process setting menu. Before setting directly, place the workpiece into the field of view of Camera 2 , find it and set the dummy reference data in the VT2 setting menu, then manually enter the proper reference data.

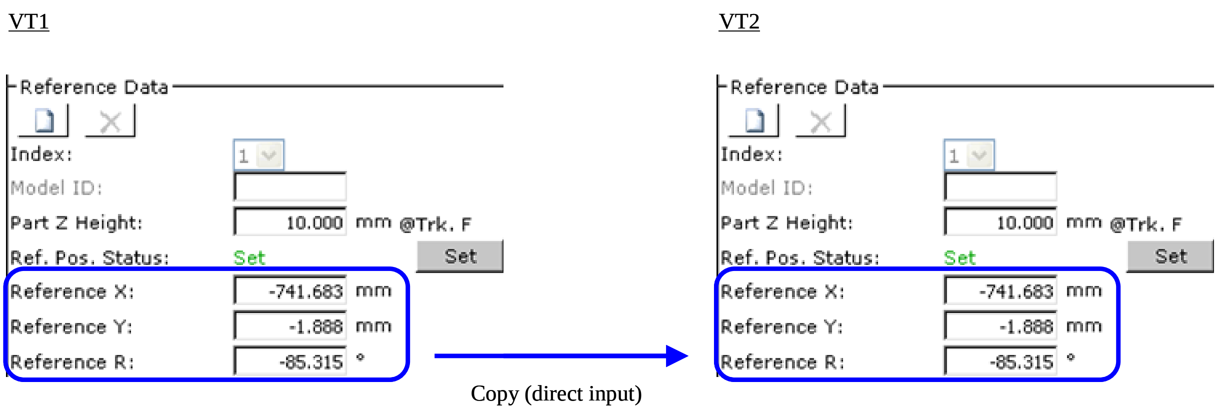
VT2



Place the workpiece into the field of view of Camera 1. Then, set the reference data for VT1 in the sensor task setting menu.

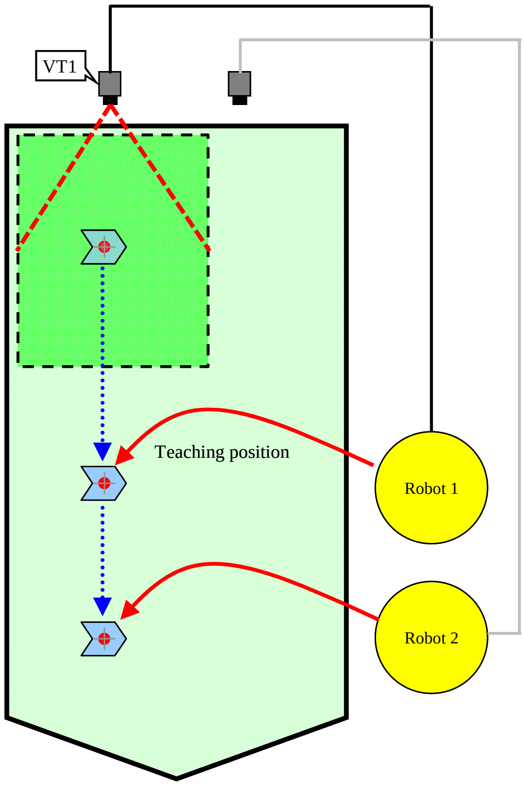


Once the reference data is set, do not move the workpiece on the conveyor until robot position teaching is completed in all work areas. Confirm [Reference X, Y, R] of [Reference Data] in VT1 setting menu and manually input the same values to one in VT2 setting menu. If the message, "If you set the reference position, you may need to reteach all of the robot points that use this offset. Click OK to set the reference position." is displayed, press F4 OK.



Robot position teaching

After setting the reference data, perform each robot position teaching.

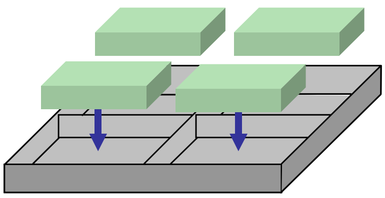


> ⚠ **CAUTION**
> 1 Each camera had better been connected to different robot controllers.
> 2 If two cameras are connected to one robot controller, you should take care of processing time and trigger interval. Even if triggers are input at a same time, two vision programs for each camera cannot be processed at a same time. After one vision program is processed, the other is processed. So you should set longer trigger interval than a total processing time of each vision program.

5.12 CONFIRMING WHETHER ALL CELLS ARE FILLED

This section describes the case in which the last robot picks up a Tray if all its cells are filled.

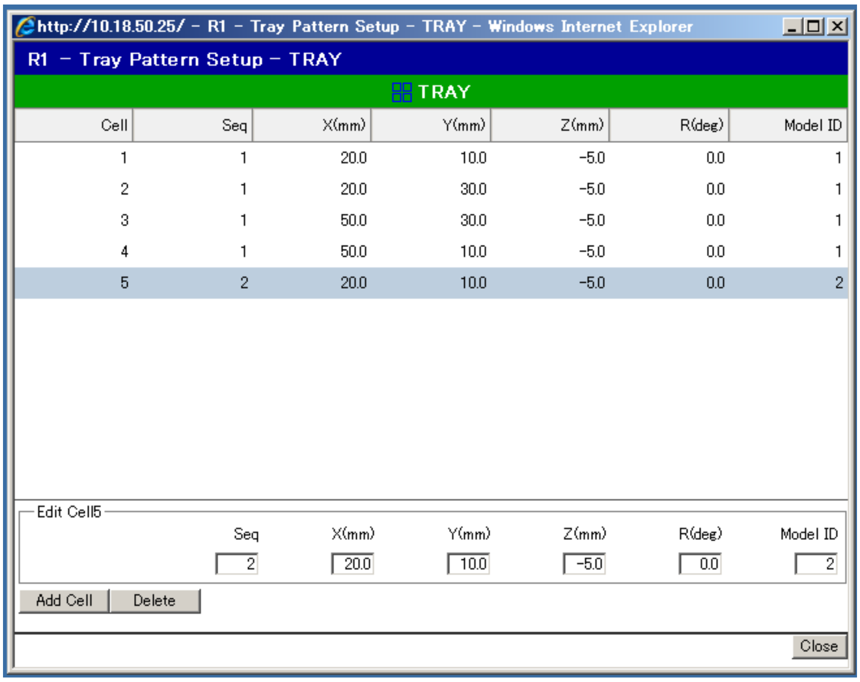
Consider the case where there are four cells in a Tray and the last robot picks up the Tray only when all its cells are filled with workpieces. It is assumed the last robot only picks up Trays and has a different gripper from other robots.



In such a case, the change of the Tray pattern and the load balance is needed. In this case, add one more cell to the Tray pattern. Set [Seq] of this cell to a bigger value than the value of the other cells so that a workpiece is placed in the last cell after all cells are filled. In addition, set the [Model ID] of the last cell to a different number from other cells so that the last cell is allocated only to the last robot.

Setting a Tray pattern

A Tray pattern is set such as the figure shown below.



Cell	Seq	X(mm)	Y(mm)	Z(mm)	R(deg)	Model ID
1	1	20.0	10.0	-5.0	0.0	1
2	1	20.0	30.0	-5.0	0.0	1
3	1	50.0	30.0	-5.0	0.0	1
4	1	50.0	10.0	-5.0	0.0	1
5	2	20.0	10.0	-5.0	0.0	2

Though a Tray actually has four cells, define five cells in a Tray pattern. Set the cell 5 as shown below.

- [Seq] is set bigger value than value of other cells.
- [Model ID] is set different number from other cells.
- Position(X, Y, Z, R) is set same value as cell 1.

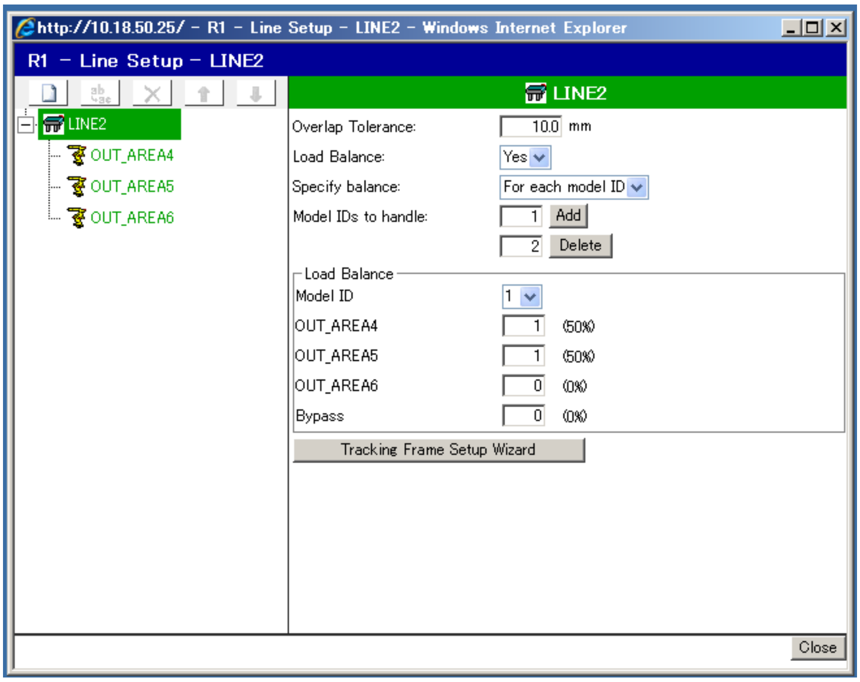
The reason the position (X, Y, Z, R) of the cell is set to the same value as cell 1 is that the position for the last cell can be taught differently from other cells.

NOTE

Ordinarily, the robot position only for cell 1 is taught, and the position for other cells, which are defined in the Tray pattern, are automatically compensated. However, the work for cell 5 to pick up a Tray itself is different from the work for other cells to put down a workpieces on a cell, so the position for cell 5 should be taught in addition to the position for cell 1. In this case, make the position (X, Y, Z, R) of cell 5 the same position as that of cell 1 for convenience. If you do it, the offset calculated based on cell 1 become zero because cell 1 and cell 5 are the same position. Consequently, the position for cell 5 can be taught separately as well as cell 1.

Setting a Line

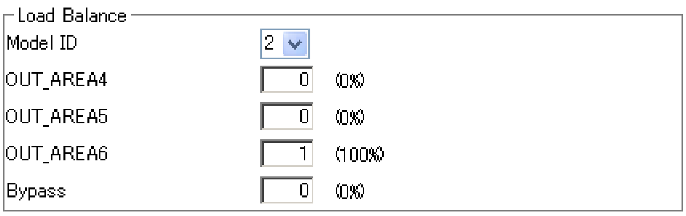
The load balance of the Line is set as shown in the image shown below.



This Line has three work areas. Suppose that the upstream two robots put down workpieces on the cells in a Tray and the downstream third robot picks up a Tray itself.

To the upstream two work areas, only the cell whose [Model ID] is 1 is needed to be allocated, so the load balance whose [Model ID] is 1 is set as a figure shown above (each first and second work area has 50%, third work area has 0%).

On the other hand, only the cell whose [Model ID] is 2 is needed to be allocated to the last work area, so the load balance whose [Model ID] is 2 is set as shown in the figure shown below.



Though the load balance whose [Model ID] is 2 in the last work area is 100%, the cell whose [Seq] is 2 (cell 5) is not allocated until all cells whose [Seq] is 1 (cell 1 ~ 4) in a Tray are filled. As a result, the last robot will only get the Trays in which all cells are filled with workpieces.

Reference position setting and robot position teaching

When each robot teaches the position after setting the reference position, only the last robot is taught not only the position of cell 1 but also the position in which the robot picks up a Tray.

> ⚠ **CAUTION**
> Here, It is assumed that the last robot picks up only Trays and cell 5 has the same position (X, Y, Z, R) as cell 1 in the Tray pattern.

Changing the program

The program is not required to change.
The last robot gets cell 5 by using VSTKGETQ.PC and picks up a Tray in the same way as other robots.

5.13 SWITCHING THE SENSOR TASK NUMBER

This section describes the way to switch a sensor task number in TP program. For example, this way is used when a system configuration is changed.

The procedure to switch a sensor task number is as following.

1 Set 1 to $VTSENS[n].$ENABLED by parameter instruction. n is a sensor task number you want to enable.

   For example, $VTSENS[2].$ENABLED = 1.

2 Set 0 to $VTSENS[m].$ENABLED. m is a sensor task number you want to disenable.

   For example, $VTSENS[1].$ENABLED = 0.

3 Call VSTKSTRT.PC again.

6 MULTI-VIEW

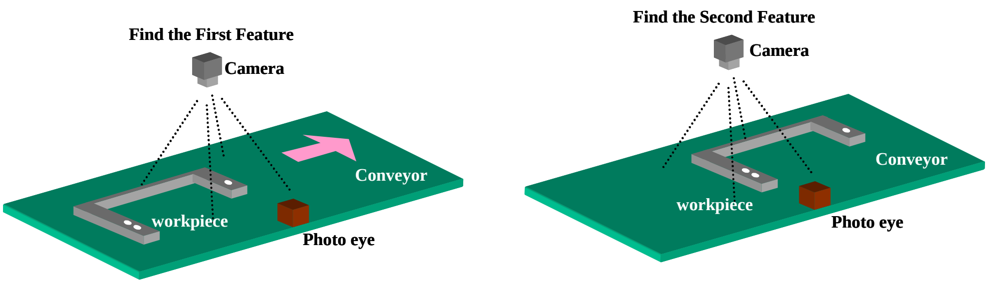
Previous chapters explain the setup of a single view using only one camera. This chapter explains the setup of multi-view using one or two cameras.

6.1 OVERVIEW

The multi-view function is useful in a case that a whole workpiece cannot be detected by one measurement for visual tracking application. A large size workpiece such as an automotive floor panel moving on a conveyer belt is given as an example.

6.1.1 Workpiece Detection Procedure

When you use the multi-view function, a workpiece is detected as follows.
A sensor such as a photo eye detects the workpiece. After the workpiece moves by specified distance, the first feature on the workpiece is snapped and found. Then the second feature on the workpiece is snap and found when the workpiece moves by another specified distance. The sensor task combines the found results and calculates the offset data.



6.1.2 Limitations

- When you use two cameras, you need to connect them to the same robot controller.
- You cannot use the multi-view function and the tray function at the same time.
- You cannot use a single view vision process and a multi-view vision process simultaneously.
  However you can switch them exclusively by $VTSENS[n].ENABLE settings. Supposing that you want to enable Sensor Task 1, you set 1 to $VTSENS[1].ENABLE while setting 0 to $VTSENS[2].ENABLE for Sensor Task 2 which you want to disable. After such setting changes, you can run VSTKSTRT again in a TP program.
- Multi-view function cannot find two or more workpieces at the same time.

6.2 PREPARATION FOR CAMERA

Number of Cameras

You can use one or two cameras.
When you use one camera, the vision detects two different features on a workpiece with the same camera. Therefore, these features go through a field of view for one camera to detect these features.
When you use two cameras, it detects the first feature on the workpiece by one camera and the second one by the other camera. In this case, you can arrange the cameras layout. For example, you can place the cameras at a certain distance to get enough intervals between the two vision detections, giving enough margins for the image processing times.

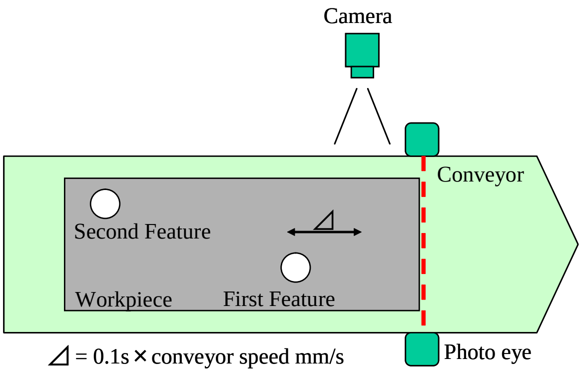
Connection

When you use two cameras, you have to connect them to the same robot controller.

Camera Layout

You can arrange the camera layout.

But the time until camera snaps image after a DI signal is input is about 0.1 second depending on a hardware configuration as mentioned in Section 3.4, “CAMERA INSTALLATION AND CONNECTION”. Therefore, it is recommended that you install a camera in the downstream side from the photo eye by about (conveyor speed × 0.1).



A camera layout example

6.3 SETUP

The setup procedures of Line, Camera, Calibration data are the same as those for a single view configuration. The procedures for other data are described below.

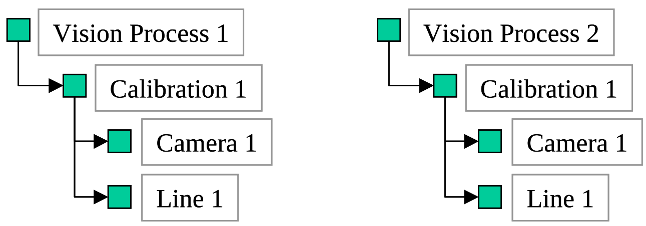
6.3.1 Vision Process

Case of one camera

Create two single view vision processes. One is for the first feature on a workpiece (view 1), and the other for the second feature on the same workpiece (view 2).

Train each vision process after making each feature be placed right under the corresponding camera. The training procedure itself is the same as that for a single view vision process.

The process diagram is shown below.



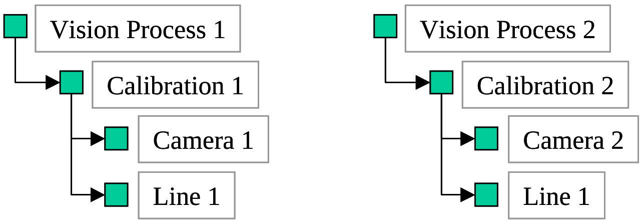
View 1 for first feature **View 2 for second feature**

Case of two cameras

Create two camera data and two calibration data for each camera first. Each procedure is the same as that for a single view configuration.

Then create two single view vision processes. One is for the first feature on a workpiece (view 1), and the other for the second feature on the same workpiece (view 2).

Train each vision process after making each feature be placed right under the corresponding camera. The training procedure itself is the same as that for a single view vision process. The process diagram is shown below.



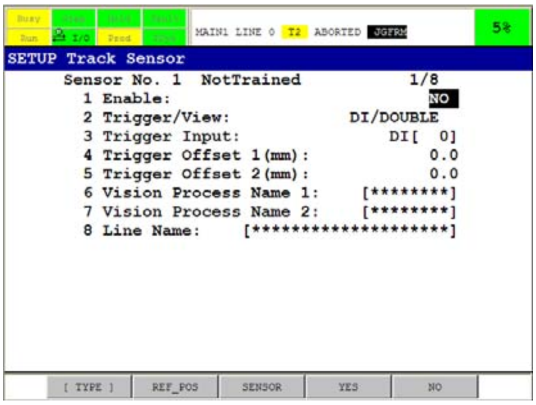
View 1 for first feature **View 2 for second feature**

6.3.2 Robot Program

You create and setup robot programs just as a single-view configuration.

6.3.3 Settings of Sensor Task

Press [MENUS] on the teach pendant, select [6 SETUP], F1 [TYPE] and then [Track Sensor].
In the displayed screen, set the cursor at [2 Trigger/View] and press F4 [SELECT]. The following screen is displayed after selecting [DI/DOUBLE].



Trigger Input

Specify DI number where the signal of sensor such as a photo eye is input.

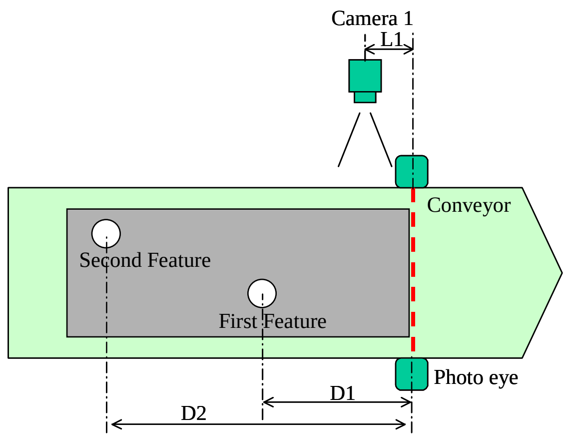
Trigger Offset 1(mm), Trigger Offset 2(mm)

[Trigger Offset 1] is a distance in mm by which the conveyer moves after the workpiece passes by the sensor such as a photo eye until the first feature is snapped and found by the vision.
[Trigger Offset 2] is another distance in mm by which the conveyer moves after the workpiece passes by the sensor such as a photo eye until the second feature is snapped and found by the vision.

Case of one camera

When you use one-camera configuration shown in the figure below, specify [Trigger Offset 1] and [Trigger Offset 2] as follows.

Trigger Offset 1 = D1(mm) - L1(mm) - 0.1(s) × Conveyor speed(mm/s)
Trigger Offset 2 = D2 (mm) - L1(mm) - 0.1(s) × Conveyor speed(mm/s)

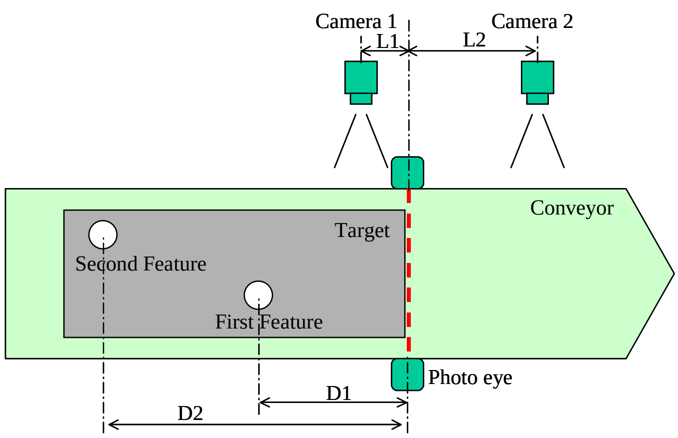


Example of one-camera configuration

Case of two cameras

When you use two-camera configuration shown in the figure below, specify [Trigger Offset 1] and [Trigger Offset 2] as follows.

Trigger Offset 1 = D1(mm) - L1(mm) - 0.1(s) × Conveyor speed(mm/s)
Trigger Offset 2 = D2(mm) + L2(mm) - 0.1(s) × Conveyor speed(mm/s)



Example of two-camera configuration

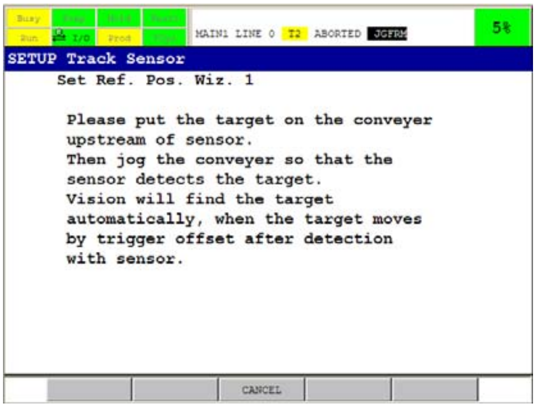
Vision Process Name 1, Vision Process Name 2

Specify the vision process name (view 1) for the first feature to [Vision Process Name 1] and the vision process name (view 2) for the second feature to [Vision Process Name 2].

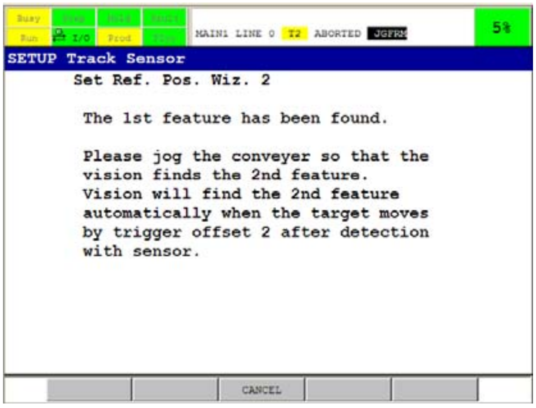
6.3.4 Reference Position Setting and Robot Position Teaching

You set reference positions according to the Wizard instructions.

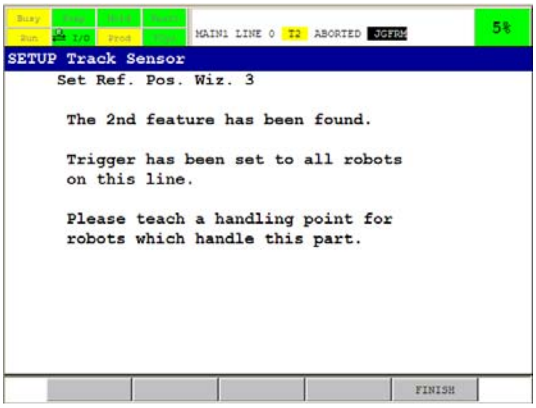
1 Put a workpiece at upstream side of the sensor such as a photo eye position on the conveyor.
2 Press F2 [REF_POS] button to start the reference position setting Wizard menu. The following screen is displayed.



3 Move the conveyer to make the photo eye detect the workpiece.
4 When the workpiece moves by [Trigger offset 1] distance after the photo eye detection, the first feature will be snapped automatically by the vision. The following screen is displayed if the workpiece is successfully found. Then move the conveyor furthermore.



5 When the workpiece moves by [Trigger offset 2] distance after the photo eye detection, the second feature will be snapped automatically by the vision. The following screen is displayed if the workpiece is successfully found. Then move the conveyor furthermore.



6 Stop the conveyor when the workpiece reaches in front of each robot.
7 Teach the robot positions. This teaching procedure is the same as that for a single view configuration.

8 Press F5 [Finish] to finish the Wizard menu.

6.4 RUN MULTI-VIEW SYSTEM

A procedure to run the multi-view system is the same as a single view system. Also it is run by VSTKSTRT and stopped by VSTKSTOP.

7 OTHER FUNCTIONS

This chapter describes the functions that are useful in the *i*RVision visual tracking system. These functions can be used to construct a more stable and flexible system.

7.1 RESTRICTING THE MOVABLE RANGE OF THE FINAL AXIS

The R-30*i*A minimizes the rotation angle of an axis when linear motion accompanying posture changes is performed. In a visual tracking or other system in which the rotation angle of the final axis changes every time on both the pickup and placement sides, the angle of the final axis may fall outside the movable range during repetition of minimum rotation, possibly causing a stroke limit alarm.

This function prevents the stroke limit of the final axis from being violated by rotating to an opposite side intentionally when a minimum rotation issues a stroke limit alarm.

This function is enabled only when the following conditions are satisfied.

- The tracking type is a “Line”.
- Line tracking is operating.
- A linear motion is performed.

To use this function, set the following system variables.

- $LNCFG_GRP[group-number].$LMT_CHK_ENB
  This is the enable/disable flag of this function. Set this variable to TRUE.
- $LNCFG_GRP[group-number].$LMT_CHK_UL
  This is the offset from the upper limit of the angle of the final axis. Set this value as needed.
- $LNCFG_GRP[group-number].$LMT_CHK_LL
  This is the offset from the lower limit of the angle of the final axis. Set this value as needed.

A change in these system variables takes effect immediately.

For example, when the stroke limit of the J6-axis ranges from −270 to 270 degrees in a 6-axis robot, if $LMT_CHK_UL is set to 90 degrees and $LMT_CHK_LL to 90 degrees, the operation range of the J6-axis is between −180 to 180 degrees.

7.2 STOPPING TRACKING MOTION OUTSIDE THE AREA

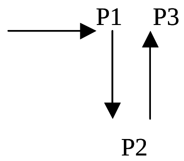
When the discard line is properly set in the Line setup page, if a workpiece crosses the discard line, the robot determines that it is not possible to overtake the workpiece and the workpiece is allocated to the next work area. If the speed override of the robot is low, the conveyor travels at a higher speed than expected, or the robot is away from the workpiece, then it may not be possible to overtake the workpiece even though it is located upstream of the discard line. This function interrupts the tracking of the workpiece in this case.

Generally, a tracking program consists of motion instructions for traveling to the following three positions.

P1: Approach position

P2: Pickup position

P3: Retraction position



This function determines whether the destination position falls within the tracking area during execution of a motion instruction for traveling to any of the above positions. If the destination position is determined to fall outside the range, this function skips the tracking motion instruction for traveling to the position and proceeds to the next instruction.

To use this function, set the following system variables.

- $LNCFG_GRP[group-number].$SKIP_OBNDMV
  This is the enable/disable flag of this function. Set this variable to TRUE.
- $LNCFG_GRP[group-number].$SKP_ADJ_MS
  The value set by this variable is used to determine whether the destination position falls within the tracking area at the start of motion. The unit is ms.

  For example, when the travel distance to P1 is up to 200 mm and the specified speed for P1 is 2000 mm/s, the setting of $SKP_ADJ_MS is shown below.
  $SKP_ADJ_MS = 200(mm) / 2000(mm/s) = 0.1(s) = 100(ms)

  When a travel to P1 is started, whether the destination position falls within the tracking area is determined at a position where the robot travels for 100 ms. If the position is determined to fall outside the tracking area, all of the P1, P2, and P3 operation instructions are skipped.

- $LNCFG_GRP[group-number].$SKP_FLG_NO
  This is a flag number. If a skip occurs, the flag specified by this variable is turned on. However, the system software only turns on the flag. When the flag is turned on, it must be turned off by a robot program.

NOTE

This function is not valid if the destination position falls outside the area when the robot continues the tracking motion while waiting for a workpiece after holding the previous workpiece with a hand for holding multiple workpieces.
In this case, as described in Subsection 4.3.2.2, “Pick program”, execute the STOP_TRACKING instruction and stop tracking before VSTKGETQ.PC.

Example of a program

When using this function, the tracking program for pickup motion or placement motion needs to be modified. The following shows an example of changing such a program. The major changes are indicated by an underline. It is assumed that the flag number is 1 and the maximum travel distance to P1 is 400 mm.

$LNCFG_GRP[1].$SKP_FLG_NO=1
$LNCFG_GRP[1].$SKP_ADJ_MS=400(mm) / 4000(mm/s)=0.1(s)=100(ms)

PICK1.TP

```
   UTOOL_NUM=0
   UFRAME_NUM=0
   R[3:GRIPPER]=1

   LBL[100]
   STOP_TRACKING
   F[1]=(OFF)
   CALL VSTKGETQ('IN_AREA', 1, 100, 2, R[3:GRIPPER])
   IF R[2:GETQ_STATUS]=0,JMP LBL[200]
   IF R[1:CYCLE_STOP]=1,JMP LBL[900]
   JMP LBL[100]

   LBL[200]
L  PR[R[3:GRIPPER]] 4000mm/sec CNT100 VOFFSET,VR[1] Tool_Offset,PR[10]
   IF (F[1]=ON),JMP LBL[300]
L  PR[R[3:GRIPPER]] 4000mm/sec CNT3 VOFFSET,VR[1]
   IF (F[1]=ON),JMP LBL[300]
   CALL VACUUM_ON
L  PR[R[3:GRIPPER]] 4000mm/sec CNT100 VOFFSET,VR[1] Tool_Offset,PR[10]

   CALL VSTKACKQ('IN_AREA', 1, 1)
   R[3:GRIPPER]=R[3:GRIPPER]+1
   IF R[3:GRIPPER]<=R[4:NUM_GRIPPER],JMP LBL[100]
   JMP LBL[900]

   LBL[300]
   CALL VSTKACKQ('IN_AREA', 1, 2)
   JMP LBL[100]

   LBL[900]
[End]
```

When the destination position is determined to fall outside the area, specify 2 for VSTKACKQ.PC to indicate that this workpiece cannot be picked up and make the next robot pick up this workpiece.

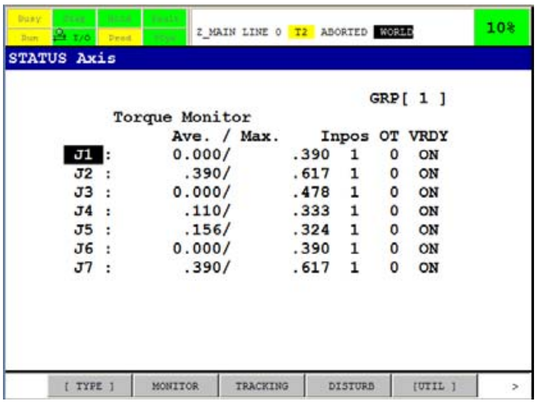
The flag needs to be turned off explicitly in the program.

7.3 CHECKING THE ROBOT TORQUE

Check that the motor is not overheated from the torque of each axis during tracking motion.

The method of displaying the torque monitor menu is as follows:

1 Press MENUS on the teach pendant.
2 Select [0 – NEXT].
3 Select [4 STATUS].
4 Press F1, [TYPE], then select [Axis] from the options.
5 Press F6 then select F2 [MONITOR]. The menu shown below is displayed.



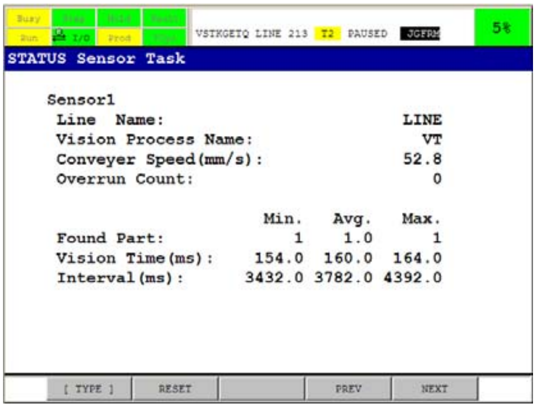
While the system continues steady operation, if the [Ave.] value of each axis increases gradually, it is possible that the motor has been overheated. In such a case, it is necessary to reduce the speed specified in the program for tracking motion.

7.4 CHECKING THE STATE OF A SENSOR TASK

The load or other states of a sensor task can be checked.

1 Press MENUS on the teach pendant.
2 Select [0 - - NEXT - -].
3 Select [4 STATUS].
4 Press the F1, [TYPE], and select [Sensor Task] from the options.
5 Select a sensor task by pressing the F4, [PREV] button and F5, [NEXT]button.

7.4.1 Single View



Line Name:
Name of the Line on which the workpiece detected by this sensor travels.

Vision Process Name:
Name of the vision process used by this sensor task.

Conveyor Speed (mm/s):
Current speed of the conveyor monitored by this sensor task.

Overrun Count:
Number of times the next trigger was input before the previous image processing was completed

when a vision system was used. If this value is not 0, reduce the frequency at which a trigger is input because system performance is insufficient for the image processing.

Found Part:
Number of workpieces detected (number of workpieces in the field of view) each time image processing is performed when a vision system is used.

Vision Time (ms):
Time to process an image and push found workpieces to a most upstream work area. If this value is greater than [Interval], [Overrun Count] is increased.

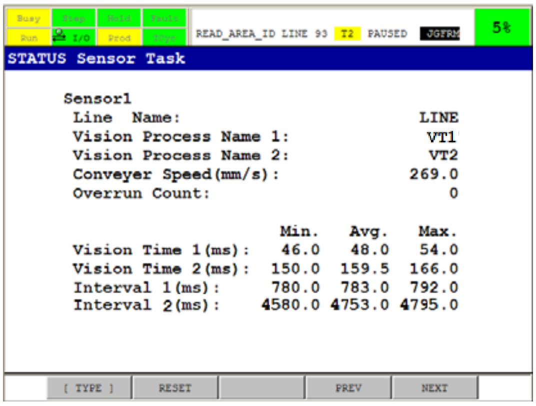
Interval (ms):
Time interval at which a trigger occurs.

RESET (F2 key):
When this key is pressed, data such as [Found Part] is reset. After a reset, a new record is created.

PREV (F4 key):
When this key is pressed, the state of the previous sensor task is displayed. The sensor task before sensor 1 is sensor 4.

NEXT (F5 key):
When this key is pressed, the state of the next sensor task is displayed. The sensor task next to sensor 4 is sensor 1.

7.4.2 Multi-View



Vision Process Name 1:
Name of the vision process used by this sensor task for view 1.

Vision Process Name 2:
Name of the vision process used by this sensor task for view 2.

Vision Time 1(ms):
Time to process an image for view 1. If this value is greater than [Interval 1(ms)], [Overrun Count] is increased.

Vision Time 2(ms):
Time to process an image for view 2 and push a found workpiece to a most upstream work area. If this value is greater than [Interval 2], [Overrun Count] is increased.

Interval 1(ms):
Time interval from a snap for view 1 to a snap for view 2.

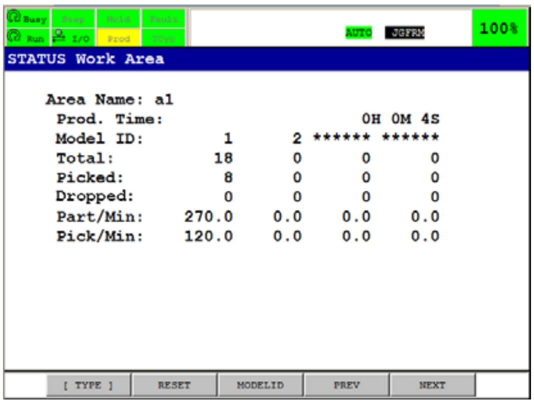
Interval 2(ms):
Time interval from a snap for view 2 to a snap for view 1.

7.5 CHECKING THE STATE OF A WORK AREA

The processing state of workpieces in each work area can be checked.

1 Press MENUS on the teach pendant.
2 Select [0 - - NEXT - -].
3 Select [4 STATUS].
4 Press F1, [TYPE], and select [Work Area] from the options.
5 Select a work area by pressing the F4, [PREV] button and F5, [NEXT] button.
6 When five or more model IDs are specified, press F3, [MODELID]. The states of the other model IDs are displayed. To return to the previous display, press the F3, [MODELID] button again.
7 If you want to display the status of the workpieces in a work area, press the [F→] button and press the F1, [PARTQ] button.

7.5.1 Status of Work Area



Area Name:
Name of a work area

Prod. Time:
Total time that has elapsed since production was started. If the F2 [RESET] button is pressed, this data is cleared.

Model ID:
Model ID used in this work area. See this item only when [For each model ID] is selected in [Specify balance] in Line setup page.

Total:
Total number of workpieces that have traveled to this work area since production was started. If the F2, [RESET] button is pressed, this data is cleared.

Picked:
Total number of workpieces that have been picked up by the robot since production was started. If the F2, [RESET] button is pressed, this data is cleared.

Dropped:

Total number of workpieces that has been dropped by the robot since production was started. If F2, [RESET] is pressed, this data is cleared.

Part/Min:

Number of workpieces traveled to this work area per minute. This value is an average in the period since production was started. If the F2, [RESET] button is pressed, this data is cleared.

Pick/Min:

Number of workpieces that were picked up by the robot per minute in this work area. This value is an average in the period since production was started. If F2, [RESET] is pressed, this data is cleared.

RESET (F2 key):

If this key is pressed. All data items including [Prod. Time] are reset. After a reset, a new record is created.

MODELID (F3 key):

If this key is pressed, the states of the model IDs not currently displayed are displayed. This key is used only when five or more model IDs are specified in the Line setup page. The displayed model IDs are changed each time this key is pressed.

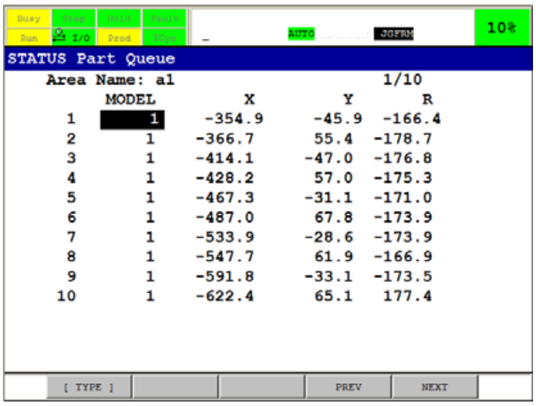
PREV (F4 key):

If this key is pressed, the state of the previous work area is displayed.

NEXT (F5 key):

If this key is pressed, the state of the next work area is displayed.

7.5.2 Status of Workpieces



Model ID:

Model ID of the workpieces through this work area.

X, Y, R:

Current position of a workpiece. These positions are represented by a tracking frame. 10 workpieces are displayed from a downstream workpiece at the maximum.

PREV (F4 key):

If this key is pressed, the state of workpieces for the previous work area is displayed.

NEXT (F5 key):

If this key is pressed, the state of workpieces for the next work area is displayed.

8 TROUBLESHOOTING

This chapter describes troubles that are like to occur in the *i*RVision visual tracking system and their remedies.

8.1 WHEN THE ROBOT DOES NOT PICK UP A WORKPIECE

The possible causes are described below.

- The work area is disabled.
  If [Skip this work area] is enabled in a work area setup page, no workpiece is allocated to the robot in the work area. Disable the skip area.
- The load balance is set improperly.
  Check that the load balance of the robot is not zero. If the load balance for putting down workpieces is invalid, it is not able to pick up workpieces properly. Check that both load balance for picking up and putting down is properly set.
- An incorrect model ID is specified.
  Check that the model ID of the workpiece detected by the vision process is identical to the model ID specified by the load balance of the Line.
  In the case of a Tray, check that the model ID of a cell is identical to the model ID of the load balance.
  When a model ID is specified during a VSTKGETQ call, check that the model ID specified is identical to the model ID of the vision process or cell.
- Workpieces are not detected by the vision process.
  Check that workpieces are detected correctly with the *i*RVision runtime image.
- No sensor task is started.
  Check that a sensor task has been started.

8.2 WHEN A WORKPIECE IS MISSED FREQUENTLY

The possible causes are described below.

- Processing by the robot is too slow.
  Disable Hot Start.

> ⚠ **CAUTION**
> The *i*RVision visual tracking system and the queue managed line tracking system do not support the hot start.

- The motion of the robot is too slow.
  If the motion of the robot is too slow, not all of the traveling workpieces can be picked. Adjust the speed in the motion program. It is also effective to add an ACC instruction to the motion program. Check that an appropriate override is selected.
- A workpiece is not detected by the vision process.
  If parameters of the vision process are inappropriate, workpieces may not be detected. Adjust the parameters of the GPM locator tool so that workpieces are detected correctly.
- Detection is too slow.
  If vision processing is too slow as compared with the conveyor speed, the next vision process cannot be executed at the requested timing and workpieces pass by the field of view being detected. Check whether vision execution is not slow with the sensor task status monitor and, if it is too slow,

decrease the speed of the conveyor or enlarge the field of view.
When multiple sensor tasks are using vision systems on one controller, distribute the sensor tasks over multiple controllers.

- Vision processing time is too long.
  Because the priority of vision processing is lower than the executing TP program, if executing TP program shares CPU time so often, the vision processing cannot run as often. As a result, the vision processing time becomes longer. Especially, if there is no workpiece to be handled on the conveyer and the zero distance motion to the waiting position is executed repeatedly, the TP program will take a large portion of the CPU time. In this case, wait the TP program so that vision processing can run by setting a positive value in the third argument of VSTKGETQ as mentioned at "4.3.2.2 Pick program". Also mentioned at "4.3.3.2 VSTKGETQ.PC", when a positive value is set to the third argument of VSTKGETQ, VSTKGETQ will work as follows.
  - If there are workpieces to be handled, it will not wait.
  - If there is not any workpiece to be handled, it will wait for specified time. However if any pickable workpiece comes in a work area before specified time passes, it will get the workpiece information and finish waiting.

  In the case of using a WAIT instruction, even if any workpiece to be handled comes in a work area, the program will wait definitely for the specified time. To wait the program appropriately as actual infeed, set the waiting time as the third argument of VSTKGETQ. Usually, 100 is enough.
  For your reference, if zero is set to the third argument of VSTKGETQ, when there is no workpiece to be handled, the loop to call VSTKGETQ that means line 5 to line 10 of "4.3.2.2 Pick program" will be executed repeatedly. It will share CPU time largely and may cause too long vision processing time.

- The setting of the discard line is inappropriate.
  If the position of the discard line is too upstream, a workpiece that can be picked up is determined to be impossible and is missed. Adjust the appropriate position of the discard line by operating the line.
- The tracking area is set inappropriately.
  Check that the tracking area is appropriate.
- The model ID is incorrect.
  Check that the model ID specified by the vision process matches [Model IDs to handle] in the Line setup page. If these are different from each other, the workpieces with the model ID will not be picked up.

8.3 WHEN THE SAME WORKPIECE IS PICKED UP TWICE

The possible causes are described below.

- Camera calibration is inappropriately.
  Check that the pointer TCP used for camera calibration is correct. Check that the grid spacing and focal distance of camera calibration are correct.
- The [Distribution Line] setting is inappropriately.
  Check that [Distribution Line] is not set within the field of view. If [Distribution Line] is set within the field of view, a workpiece is allocated even though the workpiece is located within the field of view. Since the workpiece is located within the field of view yet, it may be detected again. The robot cannot make an overlap check because the workpiece was already delivered to the robot to pick, so the robot tries to pick up the same workpiece twice.
- The [Overlap Tolerance] setting is inappropriate.
  Check that this setting is not tool small.
- A phantom workpiece falsely is detected.
  Check that a wrong workpiece is not detected in its vicinity.

8.4 COMPENSATION DEVIATES WHEN A WORKPIECE ROTATES

The tracking frame is not set correctly. See Subsection 4.6.3, “Adjustment of a Tracking Frame Setting Error”.

8.5 COMPENSATION DEVIATES FOR A WORKPIECE NEAR THE CONVEYOR END

The calibration grid used for camera calibration may be too small to calculate the lens distortion correctly. If the lens distortion cannot be calculated correctly in camera calibration, the position of a workpiece at the end of the field of view cannot be calculated correctly. Make the calibration grid larger so that it is fully covers the field of view, and calibrate it again.

If the [Part Z Height] setting specified by the vision process is incorrect, this symptom may appear. [Part Z Height] is the height of the feature to be detected in the tracking frame. Set [Part Z Height] to the height of the feature to be detected in the tracking frame properly.

8.6 WHEN ROBOT MOTION IS NOT SMOOTH

If the conveyor in use does not move smoothly, the robot also may not move smoothly because it follows the motion of the conveyor. In such a case, increase the [5 Average (updates)] on the encoder schedule menu of the teach pendant. This value is the number of sample counts used to calculate the current speed of the conveyor. The default value is 1. Set the value to 10 initially and check the operation of the robot.

8.7 STOP CONVEYOR DOES NOT WORK

Check that in multiple work areas defined in the same controller, the same DO number is not specified for the stopped conveyor, and polarities of DO are not set opposite to each other (in some work areas conveyor stop when DO is ON, but in others conveyor stop when DO is OFF).

8.8 SNAP TIMING IS DELAYED

Vision may be over load because vision saves data to the log on a memory card. Please un-check [Enable Logging] in the Vision Config screen of *i*RVision.

8.9 “Vision overrun” IS OCCURED OFTEN

This warning is occurred because [Vision time] is longer than [Interval]. Although this warning is posted, your system does not stop. But the next process is done after finishing the previous process. Therefore, some workpieces may pass the field of view while vision waits for the process to finish. In this case, you have to improve a system configuration.

DIST/SINGLE

Select [6 SETUP] from [MENUS] on the teach pendant and [Track Sensor] from F1, [TYPE]. Increase [Trigger Distance(mm)] to a larger value as much as possible and check if this warning continues to get posted.

DI/SINGLE, DI/DOUBLE

Increase the interval to infeed workpieces to a value as long as so that an interval of a signal input is longer.

Changing the setting could reduce [Vision time]. Following measures could be effective.

- If the vision log is enabled, then disable this function.
- If the vision runtime is displayed, then un-check [Enable Logging] on the Vision Config screen of *i*RVision and select [Don't Display] in [Runtime Image], the setting item of a Vision Process.
- Adjust the location parameters of GPM Locator tool. In particular, if the aspect ratio is enabled, then disable it. If the found part of the workpiece is not tilt, then the aspect ratio is not needed. If the workpiece is pouched and the model pattern looks different depending on the workpiece because the model pattern is printed on the surface of pouch, then adjust Elasticity instead of Aspect.

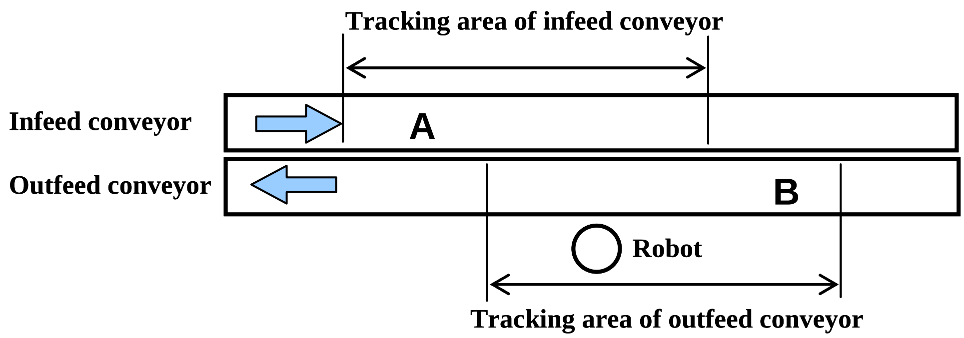
Reducing processes other than vision could also reduce [Vision time]. For example, if multitasking is used, try to stop using it.

8.10 "Track destination gone error" IS OCCURRED EVEN IF THERE IS A WORKPIECE IN THE TRACKING AREA

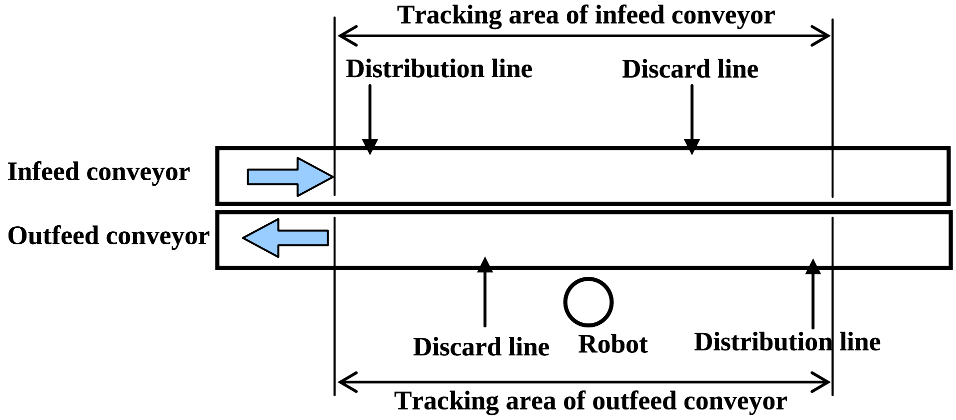
Even if there is a workpiece in the tracking area, "TRAK-005 Track destination gone error" could be occurred. This alarm occurs not only when there is a workpiece out of the tracking area but also when the robot position is out of the tracking area.
There is a system variable to overcome this problem. This problem is solved by setting $LNCFG.$COMP_SW. Add 4096 to $LNCFG.$COMP_SW if the quotient is an even number when $LNCFG.$COMP_SW is divided by 4096. If the quotient is an odd number when $LNCFG.$COMP_SW is divided by 4096, the flag to solve this problem has already been on.
There are other measures to overcome this problem. If the system has one tracking line, then this problem is solved by adjusting the boundaries of the tracking area or setting the passing point in the tracking area in order to prevent the robot position from coming out of the tracking area. On the other hand, if the system has multiple conveyors, each boundary of each conveyor need to be located in the same position. Please refer to the following examples.

For example 1:

Consider the configuration where the robot and the tracking areas of infeed/outfeed conveyors are located as shown in the following figure.



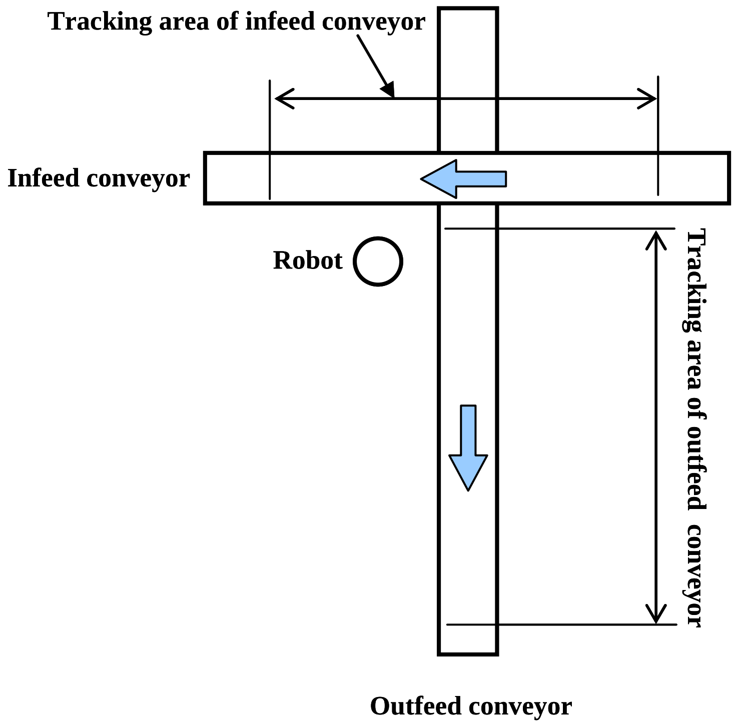
If the robot picks a workpiece at A and directly tries to perform the tracking motion to track a workpiece in the outfeed conveyor, then the alarm is posted and the robot cannot perform the tracking motion because the robot position is out of the tracking area of the outfeed conveyor. Even if the target position is in the tracking area, "Track destination gone error" alarm is posted. Similarly, if the robot drops a workpiece at B and directly tries to perform the tracking motion to track a workpiece in the infeed conveyor, then the alarm is posted, too. In that case, set the same boundary of the tracking area on both the infeed conveyor and the outfeed conveyor and adjust the work space of the robot by Distribution Line and Discard Line as shown in the following figure.



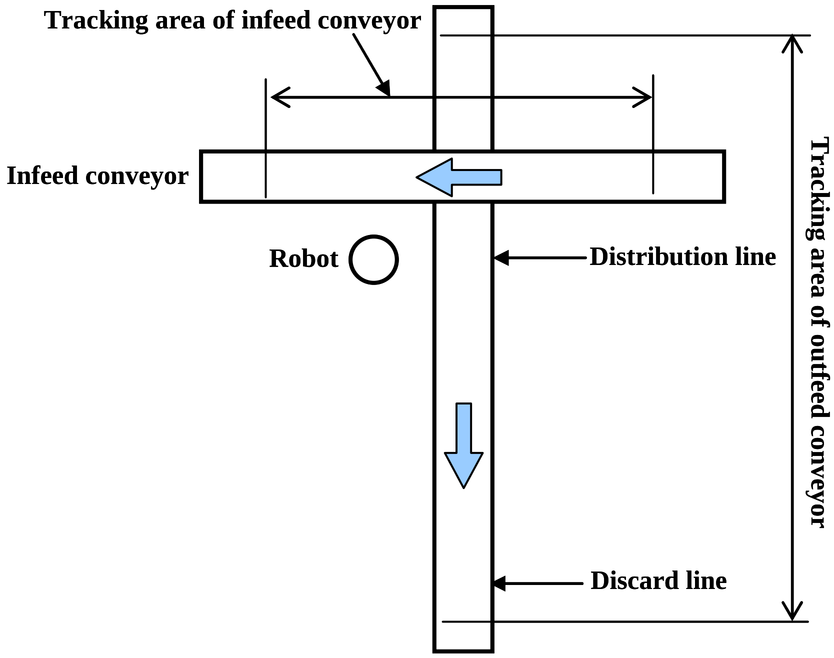
If the system has time to spare, then the alarm is also avoided by passing through another point in the tracking area before performing the tracking motion.

For example 2:

The alarm could also occur when the tracking area does not overlap the other tracking area as shown in the following figure even when lines are not parallel each other.



In that case, make the tracking areas overlap the other tracking area and adjust the work space of the robot by Distribution Line and Discard Line as shown in the following figure.



Outfeed conveyor

If the system has time to spare, then the alarm is also avoided by passing through another point in the tracking area before performing the tracking motion.

8.11 PC UIF Troubles

If there is a problem with the queue management setup operation on a PC, first check this subsection.

The robot home page cannot be opened.

If Internet Explorer of your PC is configured to use the proxy server, the PC and controller may not be able to communicate with each other correctly. Set it as described in Section 3.2, "SETTING OF A PERSONAL COMPUTER FOR SETUP".

When you click QUEUE MANAGEMENT Line / Tray Pattern Setup, the message “Failed to login Vision Setup” appears.

The Windows firewall might be set incorrectly. Set it as described in Section 3.2, "SETTING OF A PERSONAL COMPUTER FOR SETUP".

When you open the QUEUE MANAGEMENT Line / Tray Pattern Setup, the message “Enables popup on Internet Explorer” appears.

Internet Explorer might be set incorrectly. Set it as described in Section 3.2, "SETTING OF A PERSONAL COMPUTER FOR SETUP".

When you create a new line / tray pattern, a runtime error occurs.

Internet Explorer might be set incorrectly. Set it as described in Section 3.2, "SETTING OF A PERSONAL COMPUTER FOR SETUP"..

Clicking QUEUE MANAGEMENT Line / Tray Pattern Setup displays the alarm [70: Cannot write].

Internet Explorer might be set incorrectly. Set it as described in Section 3.2, "SETTING OF A PERSONAL COMPUTER FOR SETUP".

No window opens even though QUEUE MANAGEMENT Line / Tray Pattern Setup is clicked.

The Windows firewall might be set incorrectly. Set it as described in Section 3.2, "SETTING OF A PERSONAL COMPUTER FOR SETUP"..

If security software is installed in your PC, communication might be blocked by the security software. Disable the security software.

The alarm [PMON-001 Failed to notify PC Monitor] is displayed on the teach pendant of the robot.

The Windows firewall might be set incorrectly. Set it as described in Section 3.2, "SETTING OF A PERSONAL COMPUTER FOR SETUP"..

If security software is installed in your PC, communication might be blocked by the security software. Disabled the security software.

Clicking QUEUE MANAGEMENT Line / Tray Pattern Setup stops processing at icon copy time.

Communication with the robot controller may not be performed normally due to the influence of the add-on software of Internet Explorer. Disable all add-on's issued by other than FANUC Robotics North America or FRNA, by choosing "Manage Add-on's" from the "Tools" menu of Internet Explorer. In this state, check whether the queue management setup operation can be performed normally. If no problem arises, enable the disabled add-on's one at a time while checking that the queue management setup operation is not affected.

Clicking QUEUE MANAGEMENT Line / Tray Pattern Setup displays [A problem occurred] and closes Internet Explorer.

Communication with the robot controller may not be performed normally due to the influence of the add-on software of Internet Explorer. Disable all add-on's issued by other than FANUC Robotics North America or FRNA, by choosing "Manage Add-on's" from the "Tools" menu of Internet Explorer. In this state, check whether the queue management setup operation can be performed normally. If no problem arises, enable the disabled add-on's one at a time while checking that the queue management setup operation is not affected.

The hourglass-shaped mouse cursor remains displayed on the Line / Tray Pattern Setup screen and another operation cannot be performed.

If Internet Explorer in your PC is set to use a proxy server, the PC might not normally communicate with the robot controller. Open the Internet Explorer option setting screen and disable the proxy server setting.

Vision UIF Control Cannot be Installed

Check that the "*i*RVision UIF Controls" option (J871) is ordered. If the option is not ordered, contact your FANUC sales representative.

9 APPENDIX

9.1 SYSTEM VARIABLE FILES

System variable files for *i*RVision visual tracking system is shown below.

SYSVTCFG.SV
In this file, Line and Tray patterns are saved.

SYSVTSNS.SV
In this file, sensor task data is saved.

9.2 KAREL PROGRAM

Visual tracking uses the KAREL programs as shown below.

VSTKCLRQ.PC
VSTKGETQ.PC
VSTKACKQ.PC
VSTKSTRT.PC
VSTKSTOP.PC
VSTKSTLB.PC
VSTKENLB.PC
VSTKENSC.PC
VSTKSTVN.PC
VSTKSTTN.PC
VSTKGETT.PC
VSTKPFRT.PC
VSTKNPRT.PC
VSTKSENS.PC
VSTKSENM.PC

VSTKCLRQ~VSTKNPRT are used by calling from robot program. To get detailed information, see Subsection 4.3.3, “Visual Tracking Instructions”.

VSTKSENS is an execution program for the sensor task of single view. VSTKSENM is an execution program for the sensor task of multi-view. By calling VSTKSTRT, it is started internally. This KAREL program does not PAUSE or ABORT even while the teach pendant was operated. It can be stopped by calling VSTKSTOP. However, if an error occurs in the sensor task, an ABORT error is caused.

INDEX

REVISION RECORD

Edition	Date	Contents
01	Nov., 2012	

B-83304EN-4/01

* B - 8 3 3 0 4 E N - 4 / 0 1 *