EPSON

Epson RC+ 8.0 Ver.8.0
User's Guide
Project Management and Development

Original instructions

Rev.2
ENM24ZS6946F

Table of Contents

1. Introduction

1.1 Introduction

Thank you for purchasing this Epson robot system.
This manual provides the information necessary for correctly using the software.
Before using the system, please read this manual and related manuals to ensure correct use.
After reading this manual, store it in an easily accessible location for future reference.

Epson conducts rigorous testing and inspection to ensure that the performance of our robot systems meets our standards. Please note that if the Epson robot system is used outside the operating conditions described in the manual, the product will not perform up to its basic performance.

This manual describes potential hazards and problems that are foreseen. To use the Epson robot system safely and correctly, be sure to follow the safety information contained in this manual.

1.2 Trademarks

1.3 Notation

Microsoft® Windows® 10 operating system
Microsoft® Windows® 11 operating system
In this manual, the above operating systems are referred to as Windows 10 and Windows 11, respectively. Windows 10 and Windows 11 are sometimes collectively referred to as Windows.

1.4 Terms of Use

The information in this document is subject to change without notice.
Please contact us if you find any errors in this document or if you have any questions about the information in this document.

1.5 Manufacturer

SEIKO EPSON CORPORATION

1.6 Contact Information

Contact information details are listed in the "Supplier" section in the following manual.
Note that the contact information may vary depending on your region.
"Safety Manual - Contact Information"

The Safety Manual is also available at the following site.
URL: https://download.epson.biz/robots/

1.7 Before Use

Before using this manual, be sure that you understand the following information.

The Installation Folder for Epson RC+ 8.0

You can change the path for the installation folder for Epson RC+ 8.0 anywhere. This manual assumes that Epson RC+ 8.0 is installed in `C:\EpsonRC80`.

2. Epson RC+ Overview

2.1 Welcome to Epson RC+ 8.0

Welcome to the Epson RC+ 8.0 Project Management and Development Environment. Epson RC+ 8.0 is used to develop application software for the Robot Controller.

Epson RC+ 8.0 features

- Operable on Windows
  Integrated application development environment
- SPEL+ programming language
  BASIC-like programming language with support for multi-tasking, robot motion control, I/O control, and networking
- Communicates with the Controller by USB or Ethernet
- Allows you to connect one computer with multiple Controllers
- Multi simultaneous session
- I/O systems including Digital I/O boards and Fieldbus I/O
- TCP/IP and RS-232C communications
- Background task
  Controls entire system
- Database access
- Vision Guide option
  Integrated vision robot guidance
- RC+ API option
  Enables you to control the system using standard Microsoft .NET programming environments including Microsoft Visual Basic and Microsoft Visual C++.
- Security option
  Allows you to administrate all Epson RC+ users on your system. It also includes usage auditing, so you can track how many hours are spent using the system, and if changes were made.
- Conveyor Tracking option
  Enables one or more robots to pick parts from moving conveyors using vision or sensors.
- PG Motion System option
  Allows you to use third party motors and drivers to control auxiliary equipment such as XY tables, slides, etc.
- ECP option
  Supports CP motion relative to a fixed point.
- GUI Builder option
  Integrated GUI development tool
- Force Guide option
  Allows a robot to use torque/force sensing and measurement
- Absolute Accuracy Calibration
  Makes sure the coordinates and trajectory match the actual robot position and trajectory. This function is used for some options and applicable models.
- VRT option
  Enables you to reduce vibration of Robot Motion.
- Part Feeding option
  You can easily supply parts by the Robot.
- Safety Function (Only for the Controller with Safety Board, some of them are option.)
  Sets limit of operation speed and operating area for the robot, it allows you to make an application controlling robot safely.

2.2 System Overview

Epson RC+ 8.0 software, which is installed to the PC connected to the robot controller, contains several components that enable you to control an entire robotic work cell. Epson RC+ 8.0 communicates with the controller using USB or Ethernet.

Epson RC+ 8.0 and the Controller can be used in following environments:

Slave system
: The Controller is PLC or PC cell slave. Application is developed with Epson RC+ 8.0. After saving the object code to the Controller, it does not need to be connected to the computer. The Controller is controlled by I/O or fieldbus.

Standalone system
: Controls the robot and peripheral equipment as the robot controller. Epson RC+ 8.0 displays the simple operator window in AUTO mode. By using RC+ API option, .NET application can be controlled.

Offline development system
: Program edition and project build can be checked on the offline PC.

Simulation system
: Epson RC+ 8.0 on the PC which is connected to the Controller can execute the program without the actual I/O or robot by using the virtual I/O and dry run.

2.2.1 Controller

RC700, RC800 series

Epson RC+ 8.0-compatible firmware version: 7.5.4.x or later

The RC700 and RC800 series Controller is a powerful robotic work cell controller that controls our SCARA robots and 6-axis robots.

Controller features

- Sophisticated yet achieving reliability and stability
- Built in Motion System
  The motion drive system can control up to 6 axes simultaneously and 1 robot, and can add up to three drive units (RC700, RC700-A only)
- Includes standard I/O
- Wide variety of options

For detailed information on the Controller, refer to the following manual:

"Robot Controller Manual"

RC90-B

Epson RC+ 8.0-compatible firmware version: 7.5.4.x or later

The RC90-B Controller is a robot controller that can drive LS-B series manipulators.

Features:

- Built in Motion System
  The motion drive system can control one robot.
- Includes standard I/O
- Optional digital I/O expansion boards
- Optional Fieldbus slave support for DeviceNet, PROFIBUS-DP, CC-Link Ethernet/IP, PROFINET, and EtherCAT.
- RS-232C ports (standard + optional)

For detailed information on the Controller, refer to the following manual:

"Robot Controller Manual"

T series Manipulator

Epson RC+ 8.0 compatible firmware version: 7.5.54.x or later

The T series Manipulators are Controller integrated SCARA robot.

For more details on Controller part, refer to the following manuals:

- "T Series Manipulator Manual"
- "T-B Series Manipulator Manual"

VT series Manipulator

Epson RC+ 8.0 compatible firmware version: 7.5.54.x or later

The VT series Manipulators are Controller integrated 6-axis robot.

For more details on Controller part, refer to the following manual:

"VT Series Manipulator Manual"

2.2.2 Software

Epson RC+ 8.0 needs to be installed to your development PC. To communicate with the Controller, the computer should support USB 1.1/ 2.0 / 3.0 or Ethernet communication.

Software license are available for purchase at the following times.

- Purchased at the same time as the product
- Purchased later when actually used

Using Epson RC+ 8.0, you can develop application software for the SPEL+ language that runs in the RC700 controller.

2.2.3 Simulator Functions

Simulator functions enable easy robot motion checking on your PC, which gives you flexibility to consider the system layout, measure the operation time, and create robot programs.

They are useful in all the way from introduction stage of robot automation to launch of robot system.

See details below.

Simulator

2.2.4 System Requirements

To use Epson RC+ 8.0 adequately, following environment is required.

Supported OS

- Windows 10 64-bit version (version1607 or later)
- Windows 11 64-bit version

(except Windows 10 (S mode), Windows 10 IoT Core, Windows 11 SE)

PC Requirements (Recommended Specifications)

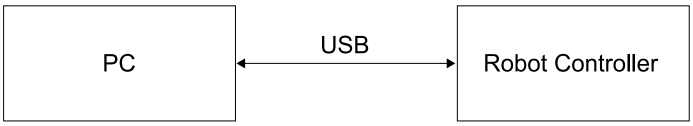
- CPU: Intel Core i5 or higher (Models released in 2017, 8th generation or later)
- Memory: 8 GB or more
- Graphic: DirectX 12 or later

> **KEY POINTS**
>
> Epson RC+ 8.0 does not support high contrast mode.

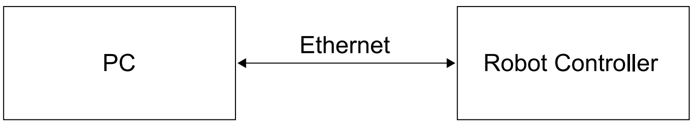
2.2.5 System Block Diagram

The following system block diagram shows methods for connecting a PC running Epson RC+ 8.0 to one or more controllers.

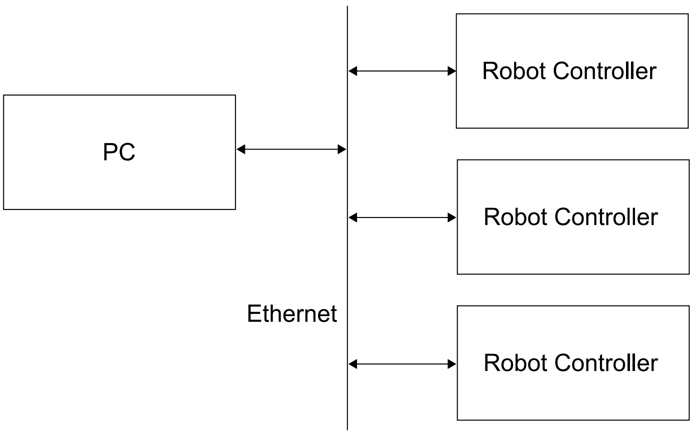
System 1: Connects the PC with one Controller using USB



System 2: Connects the PC with one Controller using Ethernet



System 3: Connects the PC with multiple Controllers using Ethernet



2.3 Options

Epson RC+ 8.0 enables the purchased Controller options license.

For details on enabling the option license, see below.

Installing Controller License

2.4 Notes for Connecting via Ethernet

The robot controller does not support internet protocol version 6 (TCP/IPv6). When you connect the development PC to the robot controller using the Ethernet, make sure to use internet protocol version 4 (TCP/IPv4).

2.5 Notes for Users of EPSON RC+ 7.x or Earlier Versions

Epson RC+ 8.0 is only compatible with projects created with EPSON RC+ 7.0.

For EPSON RC+ 7.0

EPSON RC+ 7.0 is compatible with Epson RC+ 8.0.

Projects created with EPSON RC+7.0 are saved to the following location.

" `\EpsonRC70\projects` " directory

From the Epson RC+ 8.0 menu, select [Project] - [Import Project] to import project files for use.

KEY POINTS

- When importing a project created with EPSON RC+ 7.0 Ver. 7.5.0 or earlier into Epson RC+ 8.0 for use, we recommend using EStopOff output.

  With EPSON RC+ 7.0 Ver. 7.5.1 or later, the remote I/O specifications have changed and the use of EtopOn output is not recommended.

  For details on setting changes, see below:

  Remote I/O
- We do not recommend importing a project created with Epson RC+ 8.0 into EPSON RC+ 7.0.

  When using an Epson RC+ 8.0 project with EPSON RC+ 7.0, you cannot use the features added in Epson RC+ 8.0.

  If, after being imported to EPSON RC+ 7.0 and edited, a project is again imported to Epson RC+ 8.0, the settings related to the functions added in Epson RC+ 8.0 are cleared to default values.

For SPEL for Windows and EPSON RC+ 6.0 or earlier versions

Use EPSON RC+ 7.0 to convert the project file to an Epson RC+ 8.0-compatible EPSON RC+ 7.0 project.

1. From the EPSON RC+ 7.0 menu, select [Project] - [Import Project] to import a SPEL for Windows or Epson RC+ 6.0 or earlier project.
2. From the EPSON RC+ 7.0 menu, select [Project] - [Save Project] to save the project.

   Imported projects are saved to the following location.

   " `\EpsonRC70\projects` " directory
3. From the Epson RC+ 8.0 menu, select [Project] - [Import Project] to import and use an EPSON RC+ 7.0 project file.

For details of EPSON RC+7.0, see below.

"EPSON RC+ 7.0 User's Guide"

2.6 Manuals

All documentation is installed on the PC in PDF format.

To view manuals on the PC:

- In Epson RC+ 8.0, select [Help], and then [Manuals].
- From Windows desktop, click [Start]-[Programs]-[Epson RC+ 8.0]-[Epson Robot Manuals]
- Manuals are also available on the following site.

  URL: https://download.epson.biz/robots/

Available manuals are shown in the table below.

Safety manual
    Information for using the robot system safely
    Paper manual will come with the product
Epson RC+ 8.0 User's Guide
    Explains the functions and basic operations of the integrated application "Epson RC+"
SPEL+ Language Reference
    Information for the SPEL+ Language
Status Code/Error Code
    Provides a list and status codes/error codes, and describes their countermeasures
Hand Function Manual
    Explains hand installation, commands, and use
Vision Guide
    Hardware, software, and reference for Vision Guide
Force Guide
    Hardware, software, and reference for force sensor
Part Feeding
    Integration guideline, introduction, hardware, software for part feeding
Teach Pendant
    Detailed instructions on how to use the TP (Teach Pendant)
RC+ API 8.0, GUI Builder 8.0, Fieldbus IO, PG Motion System, PLC Function Blocks, OPC UA Server, Vibration Reduction Technology, Third-party Sensor Force Control option
    Information for options
Remote Control Reference
    Information for Remote I/O control extended function
Manipulator manual
    Information for the purchased Manipulator
    Each series has its own manual
Robot Controller manual
    Information for the purchased Controller
Robot Controller Safety Function Manual
    Information for the system of safety function (Only for the Controller with Safety Board)

KEY POINTS

The "NOTE" sections describe important information to be followed for operating the Robot system.

TIP

The "TIP" sections describe hints for easier or alternative operations.

2.7 Security for Controller Ethernet Connection

Our robot system is provided on the assumption that customers use it in a closed local area network. We considered the use of a public IP address for the Controller that can be accessed via the Internet (directly or through a router) and changed the specification to support password authentication in order to secure the connection.

Password authentication is not performed in case of USB connections.

Be sure to use an IP address in the following private IP address ranges unless a public (global) IP address is required.

Private IP Address Ranges

- 10.0.0.1 to 10.255.255.254
- 172.16.0.1 to 172.31.255.254
- 192.168.0.1 to 192.168.255.254

2.7.1 Setting Password for PC Ethernet Controller Connection

It is necessary to set the authentication password in the Controller and the PC connection client for Ethernet connection when a public (global) IP address is used in the Controller.

To use a public IP address in the Controller, the authentication password must be set in advance. If there is no authentication password configured, you cannot use a public IP address for the controller.

CAUTION

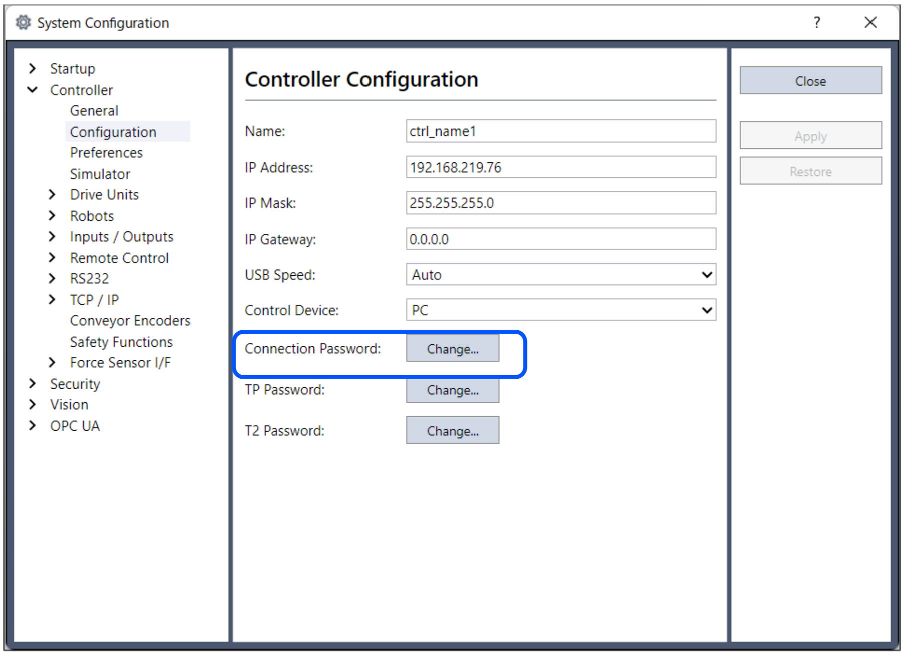
Set a private IP address for the Controller before use.

When setting a global IP address for the Controller, be sure to understand risks such as unauthorized access before use.

Setting a password for Controller

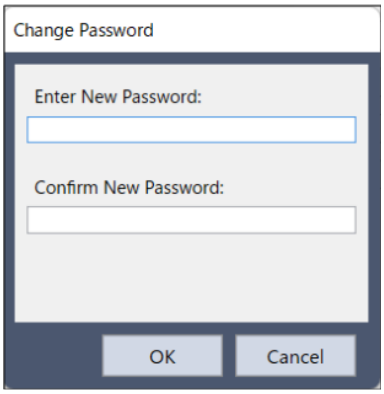
1. Select Epson RC+ 8.0 menu-[Setup]-[System Configuration]-[Controller]-[Configuration].

2. Click [Connection Password]-[Change] button.



3. Set the password. (The password must contain at least 8 characters.)

   You can use alphanumeric characters for the password. It will be valid after the Controller restart.



⚠ CAUTION

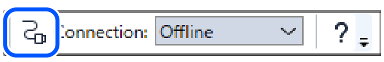
USE EXTREME CAUTION!

Keep a record of the password(s) used for encryption in a safe place. If you forget the password, you will need to initialize the Controller.

Setting a password for PC

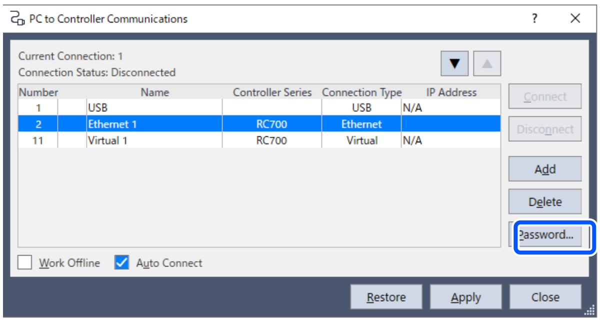
On PC (Epson RC+) side, the password can be set for each connection destination. (Ethernet connection only)

1. Click the following icon on Epson RC+ 8.0 menu.



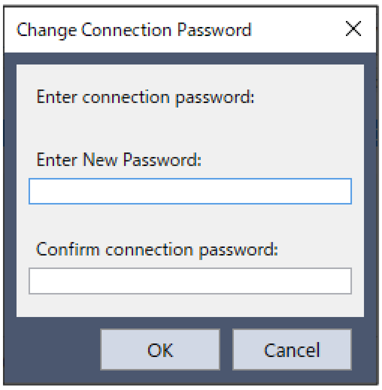
2. The [PC to Controller Communications] dialog box is displayed.

   Select the target "Ethernet" connection. Click the [Password] button.



3. The [Change Connection Password] dialog box is displayed.

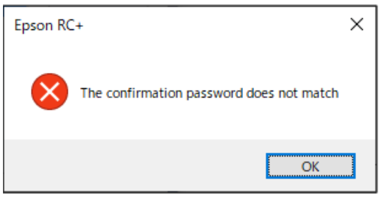
   Enter your password in the [Enter New Password:] and [Confirm connection password:].



4. Click the [OK] button.

5. When both passwords entered into the [Enter New Password:] and [Confirm connection password:] match, the password will be registered and the [PC to Controller Communications] dialog box will close.

   The following message will be displayed when the passwords do not match.



   Click the [OK] button to return to the [PC to Controller Communications] dialog box.

2.7.2 PC Ethernet Connection to Controller

When connecting to a Controller for which a public (global) IP address is configured, connection authentication using password is required.

When connecting to a Controller for which a private (local) IP address is configured, connection authentication using password is optional.

Note: Authentication is performed when the authentication password for a PC Ethernet connection is configured.

2.7.3 Remote Ethernet

To login via Remote Ethernet, password authentication is required.

All commands are not available until you log in via Remote Ethernet.

If the command is executed without login via Remote Ethernet, error "11" occurs. For details on errors, see below.

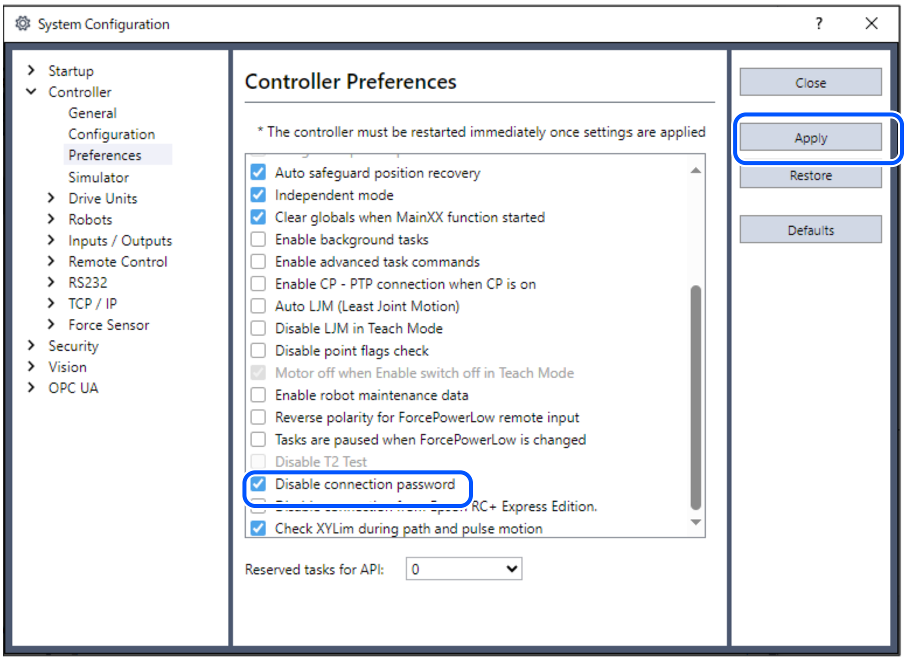
Response - "Error response"

2.7.4 Disable the PC Ethernet Controller Connection Authentication

You can disable PC (Ethernet) connection authentication. (Connection authentication is enabled by default.)

1. Select Epson RC+ 8.0 menu-[Setup]-[System Configuration]-[Controller]-[Preferences].
2. Select the [Disable connection password] checkbox.

   Click the [Apply] button.



⚠ CAUTION

Connection will not be secured if the connection authentication is disabled. Please be very careful when connecting to Internet.

2.8 Security for Compact Vision CV2 Ethernet Connection

Similar to our robot system, a Compact Vision CV2 unit is provided on the assumption that customers use it in a closed local area network. We considered the use of a public IP address for the CV2 that can be accessed via the Internet (directly or through a router) and changed the specification to support password authentication in order to secure the connection.

Be sure to use an IP address in the following private IP address ranges unless a public (global) IP address is required.

Private IP Address Ranges

- 10.0.0.1 to 10.255.255.254
- 172.16.0.1 to 172.31.255.254

- 192.168.0.1 to 192.168.255.254

For instructions on CV2 connection password configuration, refer to the following manual.

"Setup - CV2 Camera Configuration" in "Vision Guide 8.0 Hardware Manual"

2.9 Security for Part Feeding Ethernet Connection

Similar to our robot system, a part feeding (IF-80, IF-240, IF-380, IF-530) is provided on the assumption that customers use it in a closed local area network. Be sure to use an IP address in the following private IP address ranges unless a public (global) IP address is required.

Private IP Address Ranges

- 10.0.0.1 to 10.255.255.254
- 172.16.0.1 to 172.31.255.254
- 192.168.0.1 to 192.168.255.254

The part feeding does not have a security feature such as password authentication to prevent unauthorized access. If it is necessary to set a global (public) IP address to use the part feeding, please carefully consider the risk of unauthorized access via the Internet.

For instructions on the part feeding, refer to the following manual.

"Software Part Feeding Page" in "Part Feeding 8.0 Introduction & Software"

3. Safety

Please read "Safety Manual" before installing the robot system or before connecting cables and check the safety requirements.

For the safety of the Controller with Safety Board, refer to the safety manual for the product you use.

3.1 Conventions

The following symbols are used in this manual to indicate important safety information. Be sure to read the descriptions shown with each symbol.

⚠ WARNING

This symbol indicates an imminently hazardous situation which, if operation is not performed properly, will result in death or serious injury.

⚠ WARNING

This symbol indicates a potentially hazardous situation which, if operation is not performed properly, could result in an injury due to electric shock.

⚠ CAUTION

This symbol indicates a potentially hazardous situation which, if operation is not performed properly, may result in a minor or moderate injury or in property damage only.

3.2 Definitions

3.2.1 Robot Power

The status of robot power is explained below in terms of restriction to operation:

Operation-prohibited status:
    Robot cannot be operated.
Restricted (low power) status:
    Robot can operate at low speed and low torque.
Unrestricted (high power) status:
    Robot can operate without restriction.

The robot will not operate regardless of the control actions taken by the operator when in the operation-prohibited state. During operation, when the safeguard circuit opens, the system will switch to operation-prohibited state.

The robot will operate at low speed and torque in the restricted state (low power). In the unrestricted state (high power), the robot will operate at the programmed speed and torque.

In the event that the robot should make an unexpected movement, the restricted state (low power) decreases operating speed allowing the operator to avoid danger. The torque is also decreased to minimize serious injury to the operator should one be struck by the robot. The maximum values of the decreased speed and torque are set according to the robot used and cannot be changed by the user.

As a safety precaution the initial power state of the robot will be set to either the restricted (low power) state or the operation-prohibited state. The system will not change to the unrestricted (high power) state if the appropriate procedures are not followed.

When the system is in restricted (low power) state or operation-prohibited state, a single failure will not cause a runaway action that surpasses the assigned speed or torque decrease. This is due to the multi-protect circuit and mutual monitoring circuit in the control system.

3.2.2 Safeguard

Safety barriers for safety must be installed around the Manipulator, and safeguards must be installed at the entrance and exit of the safety barriers.

The term "safeguard" mentioned in this manual is a safety device with an interlock that allows entry into the safety barriers. Safeguards include safety door switch, safety barriers, light curtains, safety gates, safety floor mats, and so on. The safeguard is an input that informs the Robot Controller that an operator may be inside the safeguard area.

For RC700-E, RC800-A:

You need to assign at least one Safeguard (SG) in Safety Function Manager. For more details, refer to the following manual:

"Safety I/O Connector" in "Robot Controller Manual"

For other than RC700-E, RC800-A:

Connect the interlock switch to the safeguard input of the controller's EMERGENCY connector. For more details, refer to the following manual:

"EMERGENCY" in "Robot Controller Manual"

When the safeguard is opened, Protective Stop operates to change the safeguard state to open (displayed: SO).

Safeguard open

Operations are prohibited. Further robot operation is not possible until either the safeguard is closed and the latch is released, or TEACH or TEST mode is turned on and the enable circuit is activated.

Safeguard closed

The robot can operate automatically in an unrestricted (high power) state. To ensure safe operation, install a safety system using safety doors, light curtains, safety floor mats, etc.

⚠ WARNING

- If a third person accidentally releases the safeguard while an operator is working inside the Safety barriers, this may result in a hazardous situation. To protect the operator working inside the Safety barriers, implement measures to lock out or tag out the latch release switch.
- To protect operators working near the robot, be sure to connect a safeguard switch and make sure that it works properly.

3.2.3 Operation Mode

The operation mode is defined as the single control point for the controller, therefore you cannot use more than one operation mode at the same time.

There are four operation modes for the controller: AUTO, PROGRAM, TEACH, and TEST. AUTO operation modes allow you to execute programs in the controller when the safeguard is closed. PROGRAM operation mode allows you to execute and debug programs when the safeguard is closed. TEACH operation mode allows you to jog and teach the robot at slow speed while inside the safeguarded area.

TEST operation mode allows you to execute a program at slow speed while the safeguard is opened.

KEY POINTS

Teaching operation described in this manual is the operation of AUTO operation mode or PROGRAM operation mode. This intended for jogging the robot or teaching operation outside the safeguard.

3.2.4 Start Mode

The Start mode specifies the operation mode for Epson RC+ 8.0 when it starts. You can set the Epson RC+ 8.0 to start in AUTO or PROGRAM mode.

For information on how to change the start mode, see below.

Operation

3.2.5 Changing Operation Mode

You can change from AUTO operation mode or PROGRAM operation mode to TEACH mode by setting the mode selector key switch on the Teach Pendant to the TEACH position as described below.

- TP1, TP2: Teach
- TP3, TP4: TEACH/T1, TEACH/T2

When the mode selector key switch is changed as described below to send a release latch input signal, the operation mode is returned to the previous operation mode.

- TP1, TP2: Auto
- TP3, TP4: AUTO

The AUTO operation mode can be changed to PROGRAM mode during the Epson RC+ 8.0 startup sequence. A password can be used to allow only certain personnel to change the startup operation mode.

When Epson RC+ 8.0 starts in AUTO operation mode, the AUTO operation mode cannot be changed to PROGRAM operation mode after the system has started. To change the operation mode, restart the system and log into PROGRAM mode, then set the start mode again and restart Epson RC+ 8.0.

For details on the Start Mode, see below.

Start Mode

To change to TEST operation mode:

- TP1: Switch the mode selector key switch on the Teach Pendant to Teach, and then select Function key F1: Test Mode.
- TP3, TP4: Switch the mode selector key switch on the Teach Pendant to TEACH/T1 or TEACH/T2, and then tap the [Test] tab.

For more details, refer to the following manuals:

- "Functions Operation Mode (TEACH/AUTO/TEST) in Robot Controller option Teach Pendant TP1 manual"
- "Functions Operation Mode (TEACH/AUTO) in Robot Controller option Teach Pendant TP2 manual"

- "Functions Operation Mode (TEACH/AUTO/TEST) in Robot Controller option Teach Pendant TP3 manual"
- "Operation Mode (TEACH/AUTO/TEST) in Robot Controller option Teach Pendant TP4 manual"

KEY POINTS

- T2 mode cannot be used on RC700-A and RC700-D Controllers complying with the UL standards.
- T2 mode can be used on RC700-E and RC800-A Controllers complying with the UL standards.

3.2.6 Emergency Stop

The controller is equipped with an emergency stop input terminal. If the normally closed emergency stop circuit is broken, the power supplied to all motors will be shut off (and enter servo-free status) and the robot will be stopped by dynamic braking.

For more details on wiring, refer to the following manual:

"EMERGENCY" in "Robot Controller Manual"

3.2.7 Teach Pendant

Operators can use the teach pendant to operate the robot in the TEACH or TEST operation mode.

For operation instructions, refer to the following manuals.

- "Robot Controller option Teach Pendant TP1 manual"
- "Robot Controller option Teach Pendant TP2 manual"
- "Robot Controller option Teach Pendant TP3 manual"
- "Robot Controller option Teach Pendant TP4 manual"

3.3 Installation and Design Precautions

For installation and design precautions, refer to the following manuals.

- "Safety Manual"
- "Robot Controller Manual"
- "Manipulator Manual"

3.4 Precautions regarding Robot Operation

For precautions regarding robot operation, refer to the following manuals.

- "Safety Manual"
- "Robot Controller Manual"
- "Manipulator Manual"

3.5 Backup of Projects and Controller

After a project has been created or edited, or after system data including robot parameters has been edited, the project and controller files should be copied and stored in media other than the hard disk on the PC (e.g. USB memory key). Keep the backup media in a safe place in case of damaged data on the hard disk.

To backup, select [Controller Tools] from the Epson RC+ 8.0 [Tools] menu and execute [Controller Backup]. See details below.

[Controller] Command (Tools Menu)

Backup Controller is a function to backup both the project and the controller.

To backup only the project data, select [Copy] from the [Project] menu. See details below.

[Copy] Command (Project Menu)

⚠ CAUTION

If your system cannot be restored by Restore Controller, you must restore robot calibration parameters (Hofs, CalPls) before operating the robot. If you fail to do so, the robot will move to incorrect positions.

4. Getting Started

This chapter contains instructions for setting up and using Epson RC+ 8.0. It is recommended that first time users first read the following before reading through this chapter.

- Hardware Installation
- Software Installation
- Windows Security Administration

4.1 Hardware Installation

Epson RC+ 8.0 is used with the Controller. You need to install the controller and robot before you can use Epson RC+ 8.0 to develop and run SPEL+ applications.

You need to prepare the PC which has Windows which can run the Epson RC+7.0, and which can connect with the Controller using USB or Ethernet.

The Controller comes pre-configured at the factory. For instructions on installation, refer to the following manual.

"Robot Controller Manual"

4.2 Software Installation

Epson RC+ 8.0 should be installed to the PC with Windows. For details of adding the options, version upgrade, and re-installation, refer to the following:

Appendix B: Epson RC+ 8.0 Software

4.3 Windows Security Administration

Users need Administrator rights to use the Epson RC+ 8.0. Other users such as Power User, Limited User, Guest User cannot use Epson RC+ 8.0.

To provide security within the Epson RC+ 8.0 environment, a Security software option is available. This option allows you to manage Epson RC+ 8.0 users and audit development activity. See details below.

Security

5. Operation

This chapter contains instructions for operation of the Epson RC+ 8.0 system. The main topics are:

- System Power Up Procedure
- Starting Epson RC+ 8.0
- Communications with Controller
- Writing your first Program

5.1 System Power Up Procedure

Follow this procedure to power up the system:

1. Ensure that all safeguards are in place and that all personnel are clear of the equipment.
2. Apply power to the Controller, monitor, and I/O devices.
3. Start the Epson RC+ 8.0 software on the PC, if the PC is used in the system.

5.2 Starting Epson RC+ 8.0

There are three ways to start Epson RC+ 8.0. You can also configure the mode that Epson RC+ 8.0 starts in.

- Start Method 1

  Double-click the [Epson RC+ 8.0] icon on the desktop.

- Start Method 2
  1. Click the Windows [Start] button.
  2. Select [Epson RC+ 8.0]-[Epson RC+ 8.0].

- Start Method 3

  Configure Epson RC+ 8.0 to start automatically after Windows starts. See details below.

  Auto Start

> KEY POINTS
>
> When using the RC+ API option, you do not need to start Epson RC+ 8.0. The library created with the RC+ API option will load Epson RC+ 8.0 automatically in the background.

> KEY POINTS
>
> You cannot start Epson RC+ among multiple Windows users. (Windows user switching is not supported.)

5.2.1 Startup Sequence

When Epson RC+ 8.0 starts, it reads initial settings for the current user and local system from information stored on the PC.

The startup sequence depends on the following two factors:

- Control device
- Independent mode

When start mode is other than Independent mode (Any control device)

- If there is no project file specified in the startup command line, the last-opened project will be opened.

- If the start mode is Auto, the [Start Mode] dialog will be displayed. See details below.

  Start Mode Dialog

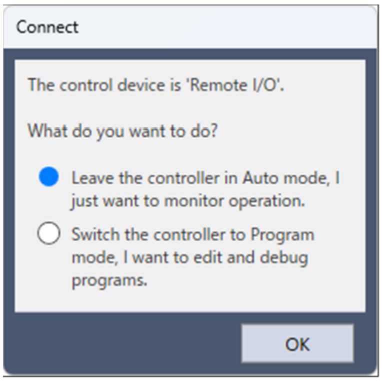
- If the start mode is Program, the Epson RC+ 8.0 GUI will be displayed.

When start mode is Independent mode (using any control device other than PC)

- If there is no project file specified in the startup command line, the last-opened project will be opened as read only.
- If tasks are currently running, Epson RC+ 8.0 will prompt to enter Monitor Mode.

> **KEY POINTS**
>
> The same dialog appears when starting Epson RC+ 8.0 in Program mode and using USB connection.

- If no tasks are currently running, a dialog below will be displayed.



Cooperative mode and Independent mode

The Robot Controller consists of the following two parts.

- Real Part: Controls the SPEL+ program (Specialized for the real time control)
- Windows Part: Controls the Windows applications (GUI)

The main function of the robot can be run by Real Part and some functions of the Controller uses the connected Windows Part. (See below)

Function	Operable with SPEL+ control device	Operable with connected Windows Part
Detail of available function	▪ Vision Guide (PV1) ▪ RC+ API option ▪ Fieldbus master	▪ PC file ▪ PC RS-232C ▪ Database access ▪ DLL calling

Real Part and connected Windows Part are started up separately at the each timing.

To operate the robot system without problem, you should synchronize these two parts. At the shipment of the Robot Controller, the Independent mode which these parts operate individually is applied.

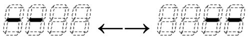
> **KEY POINTS**
>
> According to the design of robot system, it may not need to synchronize Real Part and connected Windows Part. In this case, change to Cooperative mode. For the instructions of this settings, see the section below How to set the Cooperative mode.

When the controller is in Cooperative mode, it has to wait until both of Real Part and connected Windows Part can start up without failure.

Meanwhile, on the front surface of the controller displays as below:

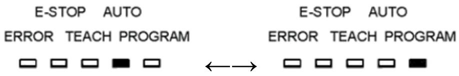
- RC700 series Seven-segment LED

  Repeat

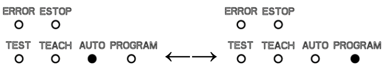
- RC90 series LED

  Repeat

- T, VT series LED *

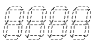
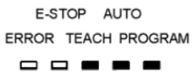
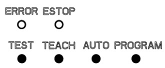
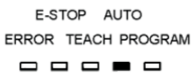
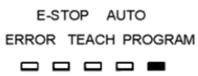
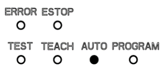
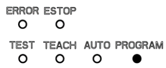
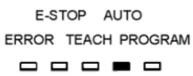
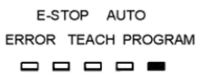
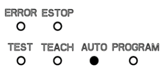
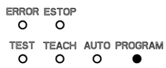
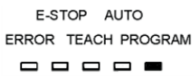
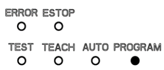
  Repeat

  - Illustration: T series

Then, it also has to wait until connected Windows part is ready and RC+ 7.0 can start up without failure.

The table below shows the startup sequence when the controller is in Cooperative mode:

	RC700, RC800 series Seven-segment LED	RC90 series LED	T, VT series LED *	Console instruction	Background task
(1) Power ON	No display 	Blinking 	Blinking 	Not available	Not started yet
(2) Real Part starts up	Repeat  ←→  alternately.	Repeat  ←→  alternately.	Repeat  ←→  alternately.	Not available	Not started yet
(3) Windows part starts up	Repeat  ←→  alternately.	Repeat  ←→  alternately.	Repeat  ←→  alternately.	Not available	Not started yet
(4) RC+starts up	Blinking 	Blinking 	Blinking 	Available	Already started

(* Illustration: T series)

(Includes the startup of the Operator Window and RC+ API application)

The table below shows the startup sequence when the controller is in Independent mode:

	RC700, RC800 series Seven-segment LED	RC90 series LED	T, VT series LED *	Console instruction	Background task
(1) Power ON	No display	Blinking E-STOP AUTO ERROR TEACH PROGRAM	Blinking ERROR ESTOP TEST TEACH AUTO PROGRAM	Not available	Not started yet
(2) Real Part starts up	Blinking	Blinking E-STOP AUTO ERROR TEACH PROGRAM	Blinking ERROR ESTOP TEST TEACH AUTO PROGRAM	Available *1	Already started
(3) Windows part starts up	Blinking	Blinking E-STOP AUTO ERROR TEACH PROGRAM	Blinking ERROR ESTOP TEST TEACH AUTO PROGRAM	Available *1	Continue
(4) RC+starts up	Blinking	Blinking E-STOP AUTO ERROR TEACH PROGRAM	Blinking ERROR ESTOP TEST TEACH AUTO PROGRAM	Available	Continue

(* Illustration: T series)

*1

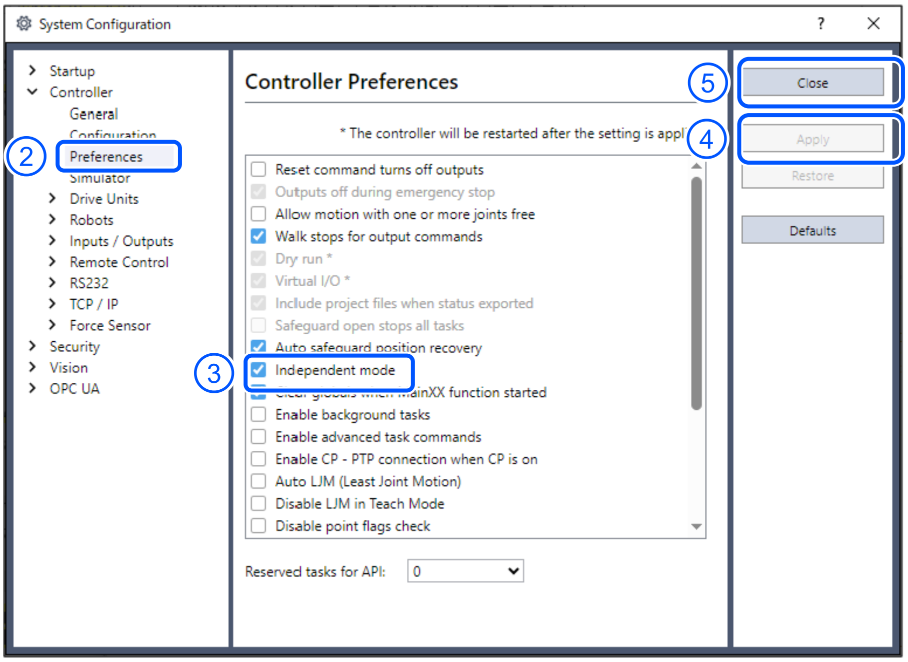
- When the control device is "PC": It waits the command execution from the Operator Window or RC+ API application.
- When the control device is other than "PC": (2) At Real Part starts up, Remote function becomes enable and starts operating.

KEY POINTS

When the controller is in Cooperative mode, the state does not back to wait for the Epson RC+ connection even after Epson RC+ shutdown. Also when the control device is other than "PC", you need to be careful during the Epson RC+ shutdown because the remote command is still executable.

How to set the Cooperative mode

1. The [System Configuration] command opens a dialog that contains several pages that are used to configure the system for the Epson RC+ 8.0 environment.
2. Select [Controller]-[Preferences].



3. Uncheck the [Independent mode] checkbox.

4. Click the [Apply] button.

5. Click the [Close] button.

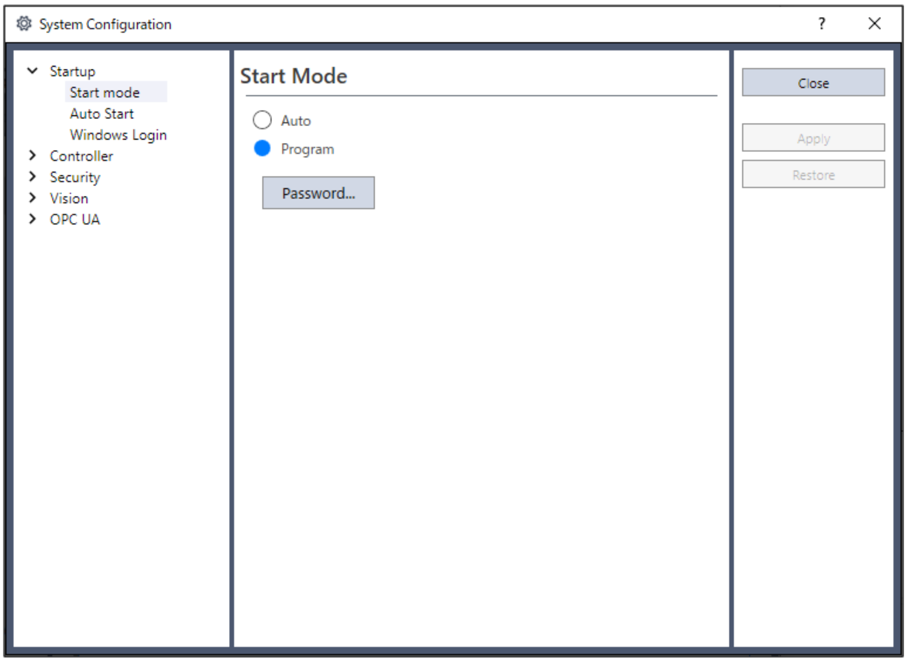
5.2.2 Startup Configuration

To configure startup settings, in the Epson RC+ 8.0 menu, select [Setup]-[System Configuration], and then select the options in [Startup].

The [Startup] section has pages for [Start Mode], [Auto Start], and [Windows Login].

5.2.3 Start Mode

This page has settings for the Epson RC+ 8.0 start mode.



There are two start modes:

- Auto: This mode starts the system and displays the Operator Window.
- Program: This mode allows you to develop your projects. This is the default startup mode.

Use the [Password] button to change the start mode password.

5.2.4 Start Mode Dialog

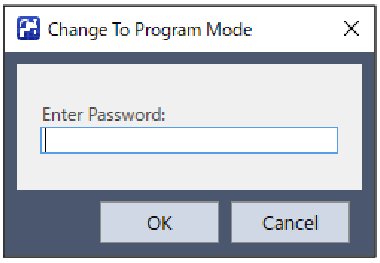
When the start mode is set for Auto, then a dialog is displayed at start up that allows you to change the startup mode using a password.

After a few seconds, if the [Change To Program Mode] button has not been clicked, the system will initialize and the Operator Window will be displayed.

You can disable this startup dialog using command line options. See details below.

Command Line Options

Click the [Change To Program Mode] button. The following dialog appears.



To change the startup mode, enter your password, and then click the [OK] button. This allows authorized personnel to enter Program mode temporarily to make changes or adjustments.

You can also abort startup all together by clicking the [Cancel] button.

KEY POINTS

When you change to PROGRAM mode from this dialog, it is only temporary. The next time Epson RC+ 8.0 runs, the original start mode setting will be used.

5.2.5 Start Mode: Program

Program mode is the default start mode. This is the Epson RC+ 8.0 development environment, from which you can:

- Create / edit projects.
- Configure the controller and set preferences.
- Run and debug programs.

5.2.6 Start Mode: Auto

Auto mode displays the Operator Window. The Operator Window is configured according to the settings in [Project]-[Properties].

The Auto mode is set by the control device as follows:

Control Device: PC
    The Operator Window can be used as a simple operator interface for production.
Control Device: Remote I/O, Remote Ethernet, Remote RS232, TP3
    The Operator window is displayed with no operator buttons to allow any diagnostic messages to be viewed.

5.2.7 Auto Start

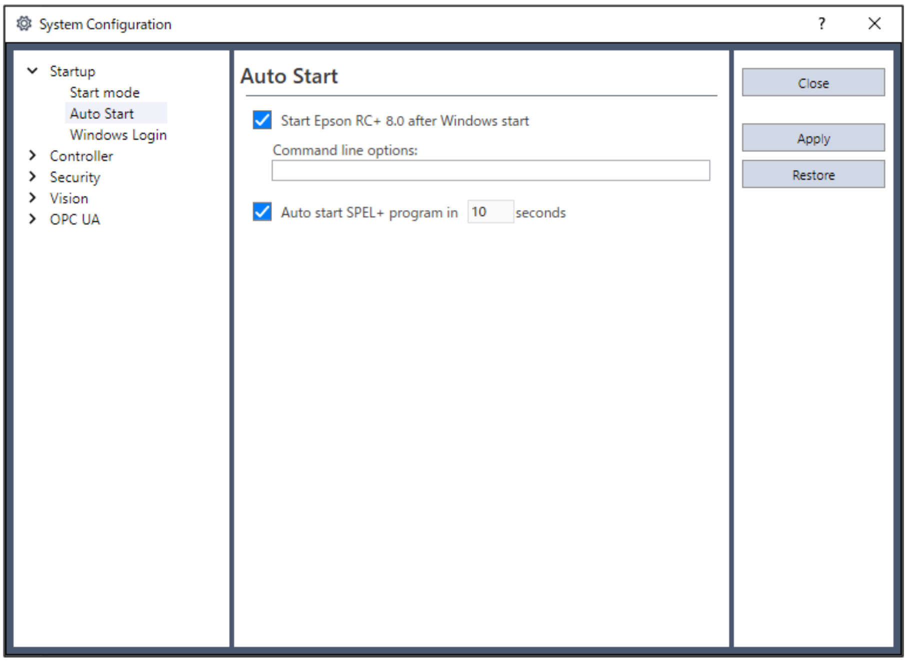
You can configure Epson RC+ 8.0 to automatically start when Windows starts.

In the Epson RC+ 8.0 menu, select [Setup]-[System Configuration]-[Startup]-[Auto Start] page, and then set the [Start Epson RC+ 8.0 after Windows start] checkbox.

In addition, if you set the checkbox above, you can specify Epson RC+ 8.0 command line options (/auto, /nosplash, etc.) in the [Command line options] text box. See details below.

Command Line Options

When the startup mode is Auto, a main function of the SPEL+ program can be started automatically. Check the [Auto start SPEL+ program in ## seconds] checkbox. Time from the Epson RC+ 8.0 startup until a main function starts can be specified in the textbox on the right. In the example below, a main function starts 10 seconds after the Epson RC+ 8.0 run. Startup of the main function can be aborted if it is within the specified time.



KEY POINTS

When using auto start, ensure that your application can automatically start safely. Additionally, inform operators on how to abort the startup.

5.2.8 Using Monitor Mode

Monitor Mode allows you to monitor operation of the controller. In Monitor Mode, you can do the following:

- View print output on the [Run] window.
- Monitor I/O status using the I/O Monitor.
- Monitor task status using the Task Manager.
- Monitor variable values using Display Variables.

To enter the monitor mode, follow the steps below.

When control device is set to a device other than PC, and Independent mode is on

1. Start Epson RC+ 8.0
2. If tasks are running, you will be prompted to connect and monitor operation.

   If tasks are not running, you will be prompted to connect in monitor mode, or switch to the Program mode.

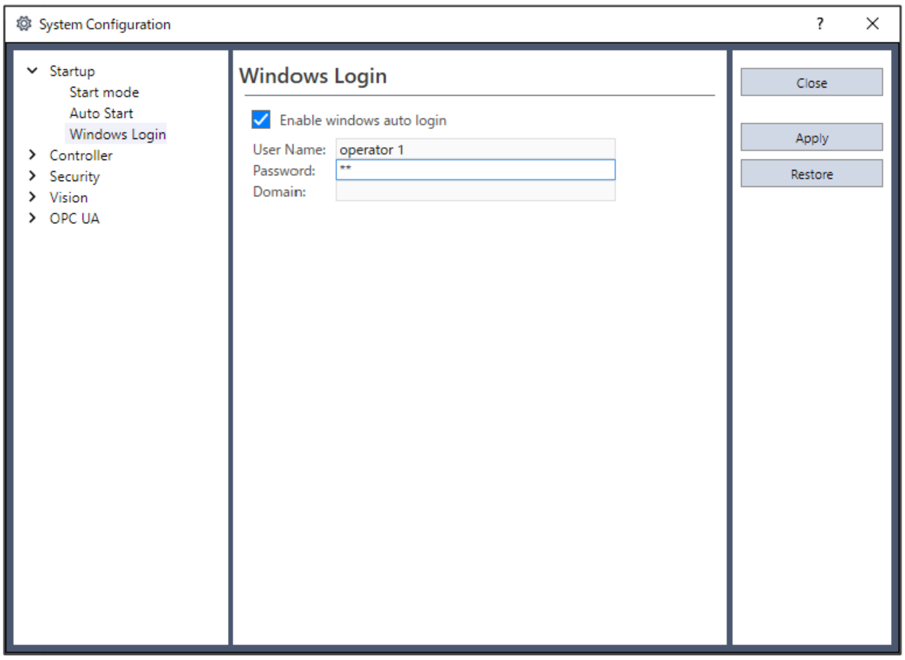
5.2.9 Windows Login

You can configure automatic Windows login from the Epson RC+ 8.0.

In [Setup]-[System Configuration]-[Startup]-[Windows Login] page, check the [Enable windows auto login] checkbox. Then, enter the name and password of the user logging in. Optionally, you can supply a domain, if required.

However, you must have the authority of Windows Administrator to set login parameters.

To configure automatic Windows login from the Epson RC+ 8.0, you must reboot the system the first time. After the reboot, Windows login will be automatic.



5.2.10 Command Line Options

For details on how to use, see below.

Using Command Line Options

There are command line options for the Epson RC+ 8.0 that provide the following functions:

Starting Epson RC+ 8.0 for a specific project

When you start the Epson RC+ 8.0, you can optionally specify a project name in the command line.

```
erc80.EXE  /PROJECT <PathToProjectFolder>
```

`PROJECT <PathToProjectFolder>`
    The project name and the directory path to any project storage folder

Example: Open project myapp saved on the C drive: (At startup)

```
erc80.EXE  /PROJECT "C:\EpsonRC80\projects\myapp"
```

Change Epson RC+ 8.0 startup mode

You can select the startup mode and override the startup dialog using command line options.

- To start in Program mode (no password required)

  ```
  erc80.EXE  /PROG
  ```

- To start in Auto mode

```
erc80.EXE  /AUTO
```

Use these command line options to override and hide the startup dialog and open the Operator Window directly.

If only the AUTO flag is supplied and the control device is PC, Epson RC+ 8.0 will open the project from the last session and display the operator window. Epson RC+ 8.0 will only be visible in the Windows Task Manager. When the operator window is closed, Epson RC+ 8.0 will be terminated.

> **KEY POINTS**
>
> When the control device is PC, you cannot close the operator window while tasks are running.
>
> Example: Open project myapp on drive C and display the operator window:
>
> ```
> erc80.EXE  /PROJECT "C:\EpsonRC80\projects\myapp" /AUTO
> ```
>
> The Controller should be ON before starting Epson RC+ 8.0 with the /AUTO command line option. If Epson RC+ 8.0 cannot communicate with the controller, then an error message will be displayed with the [Retry] button.
>
> See details below.
>
> **Operator Window**

Login

You can automatically login from the command line if you are not using the Auto Login feature for the security Option:

```
erc80.EXE /LOGIN "userID", "password"
```

This is especially useful when you are starting in operator mode.

If the user ID or password is invalid, it will display an error dialog and exit the Epson RC+ 8.0.

Starting Epson RC+ 8.0 specifying the language

You can specify the language to use in Epson RC+ 8.0 GUI.

- Japanese: `erc80.EXE /LANG_JAPANESE` *1
- English: `erc80.EXE /LANG_ENGLISH`
- German: `erc80.EXE /LANG_GERMAN` *2
- French: `erc80.EXE /LANG_FRENCH` *2
- Spanish: `erc80.EXE /LANG_SPANISH` *2
- Chinese (Simplified): `erc80.EXE /LANG_CHINESESIMP` *3
- Chinese (Traditional): `erc80.EXE /LANG_CHINESETRAD` *3

*1 Available for Japanese OS

*2 Available for English, German, French, and Spanish OS

*3 Available for Chinese OS

> ✎ KEY POINTS
>
> The language specified at startup is a temporary setting. To set this by default, in the RC+ menu, set the language in [Setup]-[Preferences]-[Language Preferences]. See details below.
>
> **[Setup]-[Preferences]-[Language] Page**

Disabling the Epson RC+ 8.0 splash window

You can suppress the splash window displayed at startup using the following syntax:

```
erc80.EXE /NOSPLASH
```

5.2.11 Using Command Line Options

Examples of command line options are:

Running from [Run]

You can specify a command from the Windows [Start] menu-[Run]-[Name] text box.

e.g.

```
C:\EpsonRC80\exe\erc80.exe  /PROJECT "C:\EpsonRC80\projects\myapp"
```

Making startup icons for your projects

You can create icons that automatically start Epson RC+ 8.0 for different projects and start Auto or Program modes.

1. Right click on your desktop and select [New]-[Shortcut].
2. Click <Browse…> in the [Create Shortcut] dialog box.

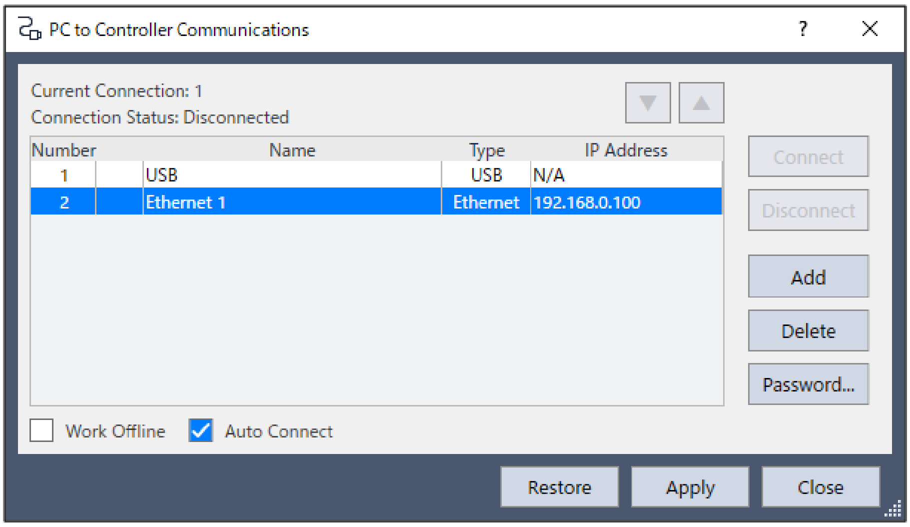
   Select " `C:\EpsonRC80\exe\erc80.exe` " and click the [OK] button. After the dialog changes, click the [Next] button.
3. Type a name for the shortcut and click [Finish].
4. Right click the created icon and select [Properties]. Add an option such as "/AUTO" or "/PROG" to [Target:].

5.3 Communications with Controller

Your PC running Epson RC+ 8.0 can communicate with a Controller using USB or Ethernet.

5.3.1 Configuring Communications with the Controller

In the Epson RC+ 8.0 menu, select [Setup]-[PC to Controller Communications]. The following dialog appears. Configure communications with the Controller.



The dialog has a list of connections. The first connection is for USB and is fixed. You cannot delete or change.

You can add one or more Ethernet connections and give each one a meaningful name.

The name for each connection is also shown in the Connections dropdown list on the main toolbar. If no name is supplied, the Ethernet IP address is shown in the dropdown list.

See details below.

[PC to Controller Communications] Command (Setup Menu)

5.3.2 USB Communications

USB 2.0 or USB 1.1 can be used to communicate with one controller. This is the default communication method for Epson RC+ 8.0 and requires no configuration. To connect to a controller via USB:

1. Connect a USB cable between the PC and the controller.
2. Turn ON the Controller.
3. Start Epson RC+ 8.0
4. Select [Setup PC to robot controller communications] on the toolbar.
5. Ensure that connection #1 (USB) is selected.
6. Click the [Connect] button.
7. Click the [Close] button.

KEY POINTS

EPSON RC+7.0 may already be installed on the same PC. If EPSON RC+ 7.0 is already performing USB communication, Epson RC+ 8.0 cannot perform USB communications. Make sure that EPSON RC+ 7.0 is disconnected before connecting Epson RC+ 8.0.

CAUTION

When performing the USB communication using the computer with Windows 10 or later, communication with the Controller is disconnected as the PC enters a sleep state. Before performing the USB communication, make sure to change the PC setting so that it will not enter the sleep mode.

5.3.3 Ethernet Communications

KEY POINTS

The robot controller does not support internet protocol version 6 (TCP/IPv6). When you connect the development PC to the robot controller using the Ethernet, make sure to use internet protocol version 4 (TCP/IPv4).

You can communicate with one or more Controllers from one PC using Ethernet. For Ethernet communications, each Controller must have a unique IP address. You can set the IP address, mask, and gateway for the controller from [Setup]-[System Configuration]-[Controller]-[Configuration]. The gateway setting is only required if you will be accessing the controller from outside of the local network.

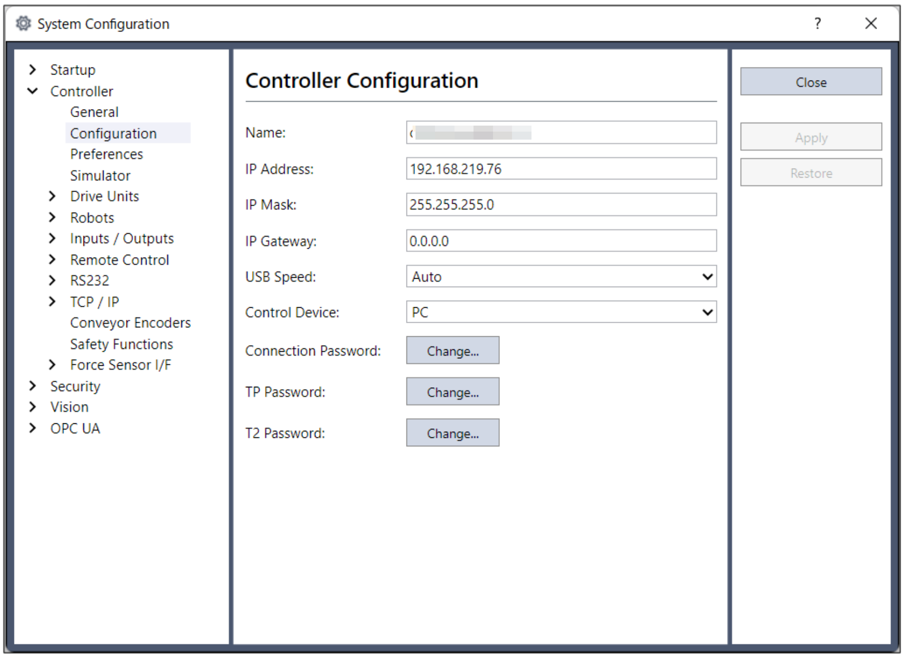
You can connect the controller directly to a PC using an Ethernet cable. Alternatively, you can connect the PC and controller to an Ethernet switch or hub.

Before you can communicate with a controller using Ethernet, you must configure the controller's IP address, IP mask, and IP gateway. This is accomplished by first connecting to the controller with USB, and then from the Epson RC+ 8.0 [Setup]-[System Configuration]-[Controller]-[Configuration] page, set the IP address, IP mask, and IP gateway of the controller as shown below.

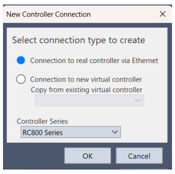
The following is the configuration of the controller at the time of shipment.

- IP Address: 192.168.0.1
- Subnet Mask: 255.255.255.0
- IP Gateway: 0.0.0.0

Use the USB connection to configure Ethernet communications.



In Ethernet communication, the connection method differs depending on the Controller series. Use the [New Controller Connection] window to set up.

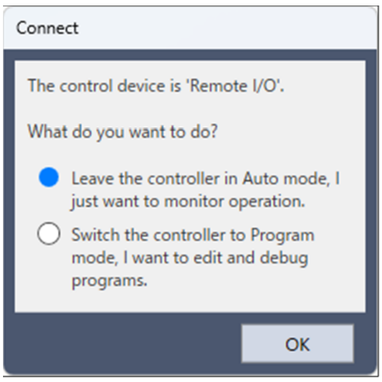


If an error dialog is displayed when connecting, check the Controller to which you are connecting and change the Controller series if necessary.

5.3.4 Connecting a Control Device That is Not a PC

Connecting while control device is not PC and tasks are not running

If your control device is not a PC and tasks are not running, you will see the following message box:



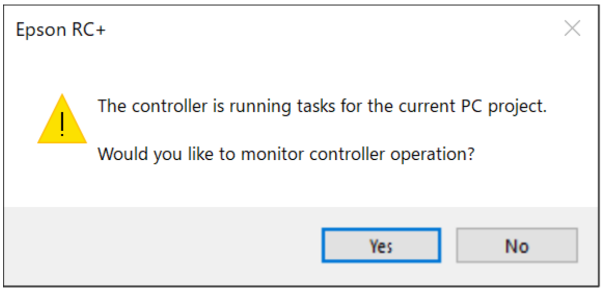
[Leave the controller in Auto mode, I just want to monitor operation.]: Switches to Monitor mode for monitoring the controller status.

[Switch the controller to Program mode, I want to edit and debug programs.]: Switches to Program mode for editing programs. The remote device cannot start programs until remote control has been enabled from the [Run] window.

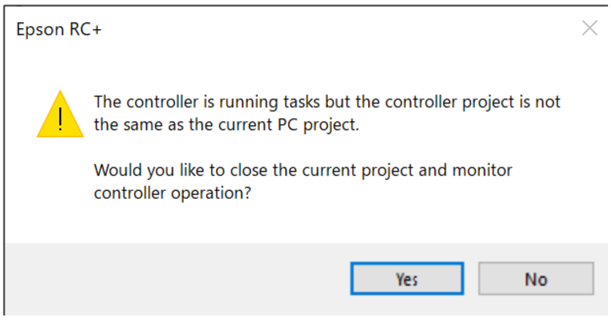
Connecting from a control device that is not a PC while tasks are running

When using a control device that is not a PC and tasks are running, you can connect to a PC to the controller to monitor operation. For example, you can connect to a controller that is running tasks to temporarily monitor display output, tasks, and I/O, and then disconnect while tasks continue to run.

If the project on the PC is the same as on the controller, you will see the following message box when connection is established:



If the project on the PC is not the same as the project in the controller, you will see the following message box when connection is established:



If the [Leave the controller in Auto mode, I just want to monitor operation.] option is selected, the [Run] window will open if Epson RC+ 8.0 is started in Program mode. If Epson RC+ 8.0 is started in Operator mode, the [Operator Window] will appear. From the [Run] window or [Operator Window], you can view results of Print instructions. You can also use the Task Manager and I/O monitor.

When monitoring controller operation, the controller remains in Auto mode. You cannot stop tasks from Epson RC+ 8.0, because the control device is not a PC. Stop the task from the current control device, and then connect the controller from Epson RC+ 8.0. Next, select Program mode to switch the controller to Program mode. (Refer to Connecting while control device is not PC and tasks are not running.)

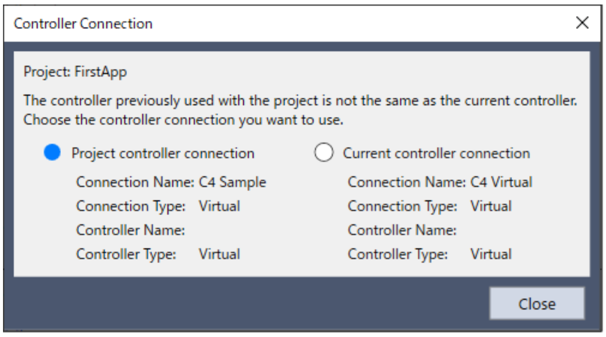
Disconnecting while tasks are running

You can only disconnect from the controller with tasks running when the control device is set to a device other than PC.

1. Stop communications with the controller by selecting [Offline] from the [Connection] dropdown list on the toolbar.
2. You may now disconnect the communications cable between the PC and the controller. Tasks will continue to run in the controller.

5.3.5 Project Controller Tracking

Epson RC+ keeps track of which controller connection was used with a project. This is useful for when multiple projects and controllers are used from the same PC. When connecting Epson RC+ to a different controller for the current project to that used for the previous project, a dialog window will appear with information about the previously connected controller, and the controller you are trying to connect. You can select which controller you want to connect to with the current project.

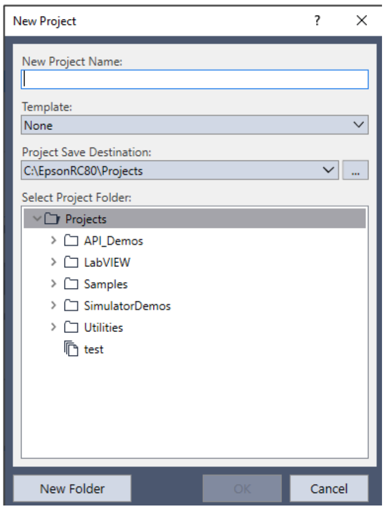


5.4 Writing your first Program

After installing the controller, robot, and Epson RC+ 8.0 software on the RC700 Robot Controller, follow these instructions to create a simple application program so that you will become more familiar with the Epson RC+ 8.0 development environment.

1. Double-click the  [Epson RC+ 8.0] icon on the desktop to start Epson RC+ 8.0.
2. The New command is used to create a new Epson RC+ 8.0 project.

   i. In the Epson RC+ 8.0 menu, select [Project]-[New Project]. The [New Project] dialog box will appear.

   ii. Type in a name for a project in the [New Project Name] box. e.g. FirstApp

   iii. Click the [OK] button to create the new project.

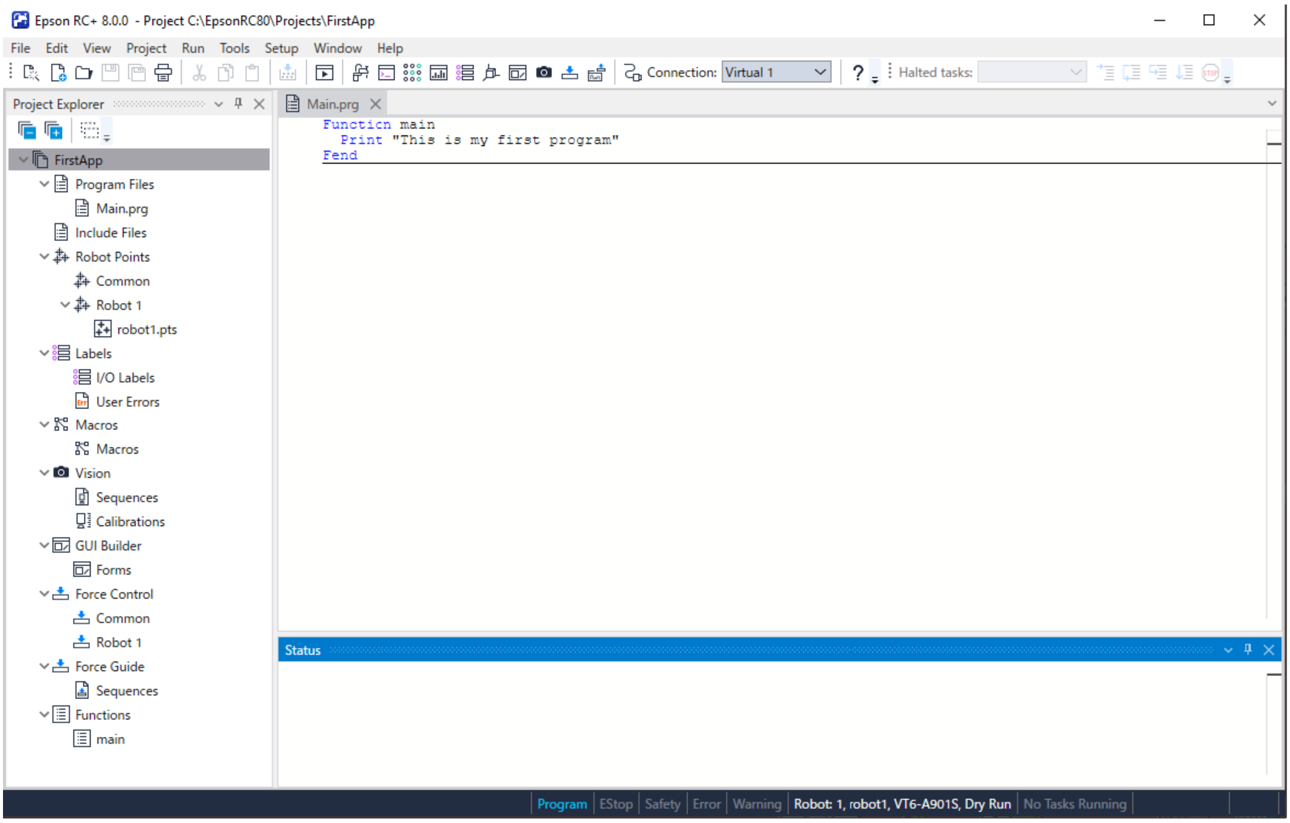
   When the new project is created, a program called "Main.prg" is created.

   You will see the "Main.prg" window open with a cursor flashing in the upper left corner. Now you are ready to start entering your first program.

3. Edit the program.

   Type in the following program lines in the "Main.prg" edit window.

```
Function main
  Print "This is my first program."
Fend
```

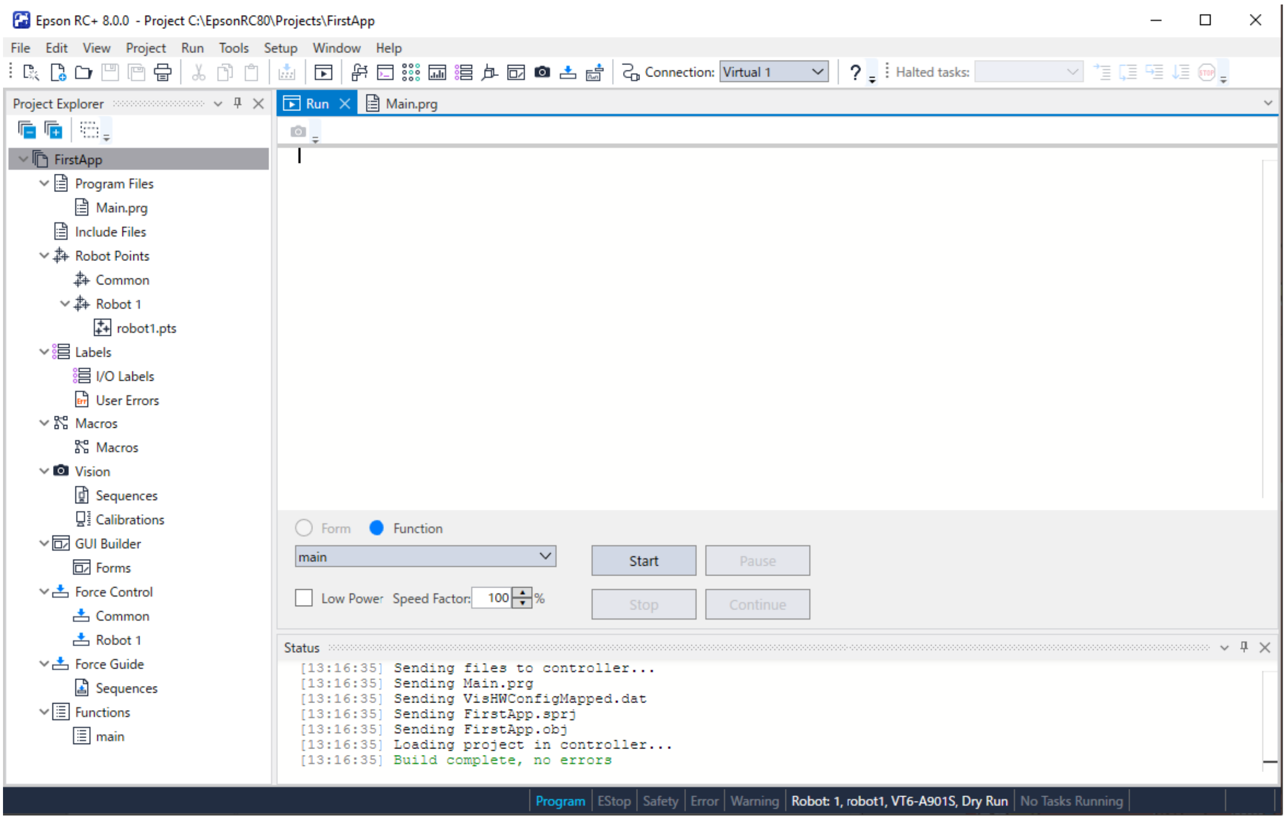


4. Run the program.

   i. In the Epson RC+ 8.0 menu, select [Run]-[Run Window] (F5 is the shortcut).

      The [Status] window showing the build operation status will appear.

   ii. During project build, your program is compiled and linked. Then communications is established with the controller and project files are sent to the controller. If there are no errors during build, the [Run] window will appear.



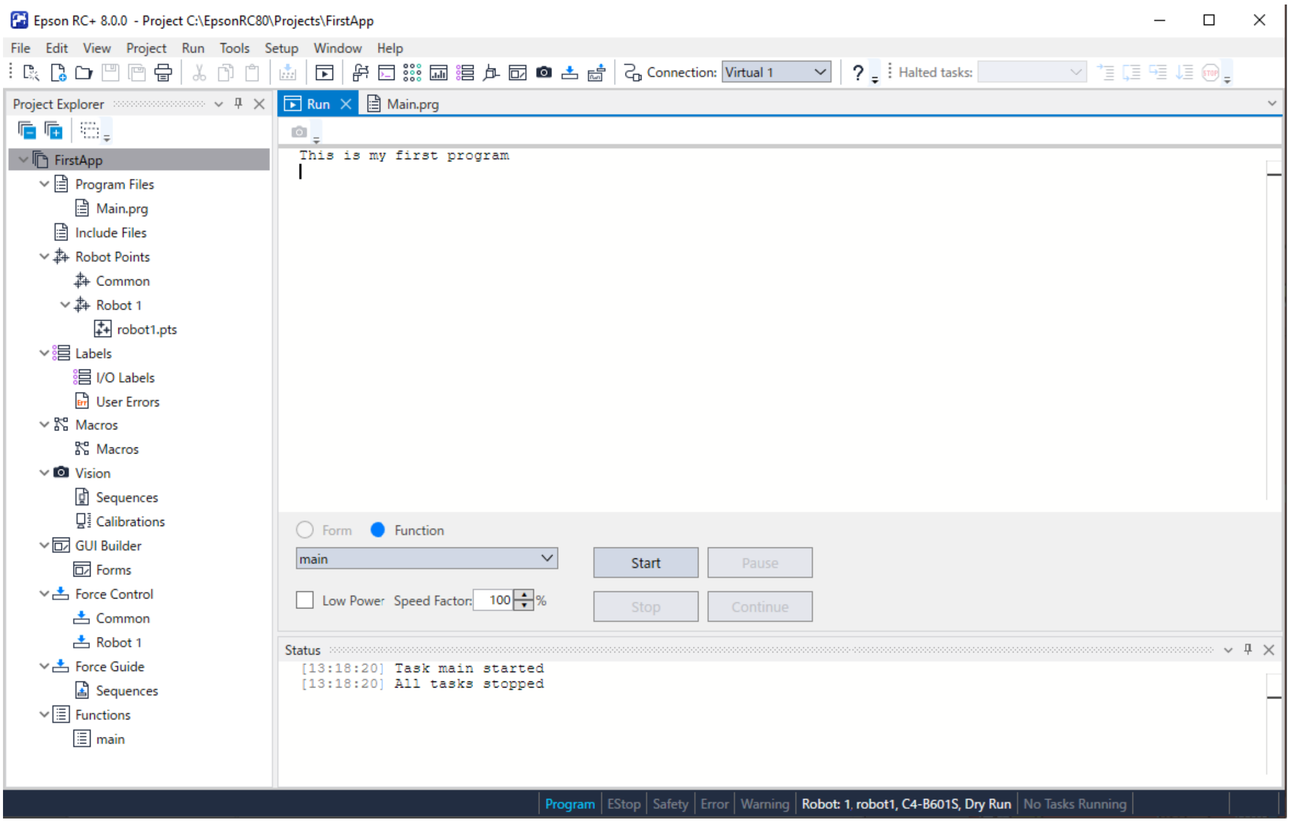
iii. Click the [Start] button on the [Run] window to run the program. You will be prompted to confirm the operation. Click the [Yes] button.

iv. The following tasks will appear in the [Status] window:

> Task main started

> All tasks stopped

The statement output will appear on the [Run] window.



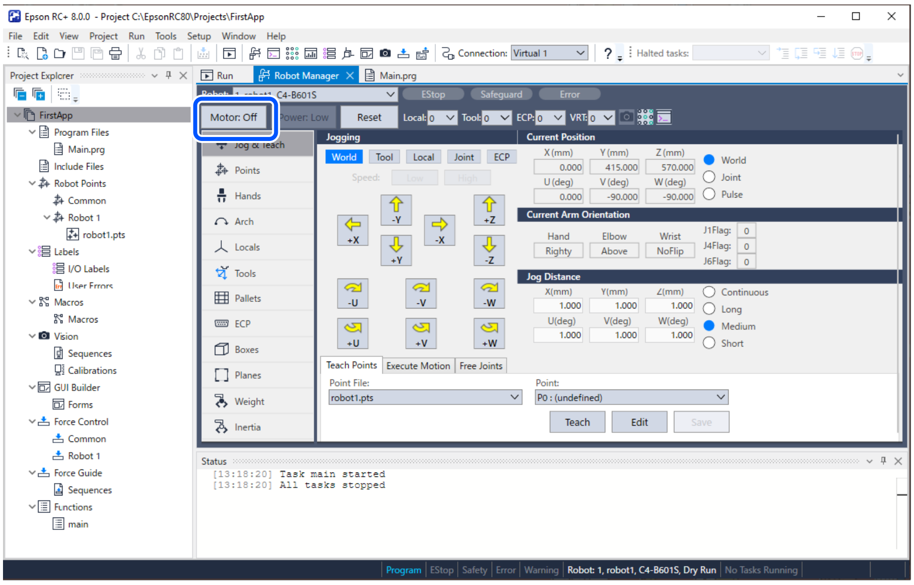
Now let's teach some robot points and modify the program to move the robot.

5. Teach robot points

i. Ensure that it is safe to operate the robot.

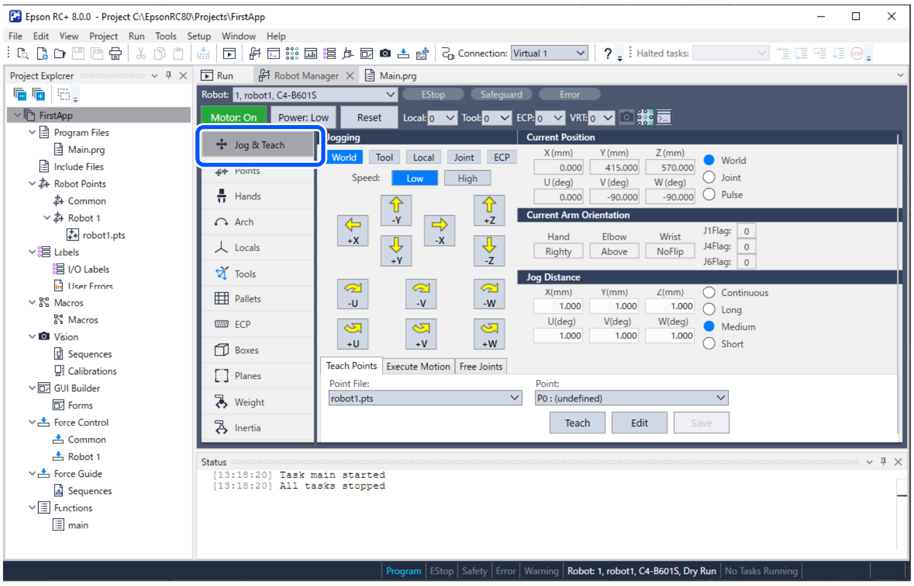
ii. Click the  [Robot Manager] button on the toolbar to display the [Robot Manager] window.

iii. Click the [Motor: Off] button at the top of the [Robot Manager] window to turn the motor on.



iv. You will be prompted to confirm the operation. Click the [Yes] button.

v. Click the [Jog & Teach] tab.



vi. Teach P0. Click the [Teach] button. You will be prompted to enter a point label and comment.

vii. Click the [+Y] button to incrementally jog the robot in the +Y direction. Hold the button down to continue jogging. Let go when the robot is about half way out in the work envelope.

viii. Click the [-Z] button to jog the robot down the robot's Z-axis.

ix. In the [Point] box, select "P1".

x. Click the [Teach] button. You will see a confirmation message to teach the point.

xi. Click the [Yes] button. The current point is set to P1.

xii. Click the [+X] button to incrementally jog the robot in the +X direction.

xiii. In the [Point] box, select "P2".

xiv. Click the [Teach] button. You will see a confirmation message to teach the point.

xv. Click the [Yes] button. The current point is set to P2.

xvi. Click the  [Save all files] button on the toolbar to save the changes.

6. Modify the program to include robot motion commands

   i. Insert three new Go statements into the Main.prg program as shown below:

```
Function main
  Print "This is my first program."
  Motor On
  Go P1
  Go P2
  Go P0
Fend
```

   ii. Press F5 to display the [Run] window.

   iii. Click on the [Start] button to run the program.

   The robot should move to each of the points you taught.

7. Modify the program to change speed of robot motion commands

   i. Insert the Power, Speed, and Accel commands as shown in the program below:

```
Function main
  Print "This is my first program."
  Motor On
  Power High
  Speed 20
  Accel 20, 20
  Go P1
  Go P2
  Go P0
Fend
```

   ii. Press F5 to display the [Run] window.

   iii. Click on the [Start] button to run the program.

   The robot should go to each of the points you taught at 20% speed, acceleration, and deceleration. The Power High statement enables your program to run the robot at high (normal) power, which in turn allows the robot speed and acceleration to be increased.

8. Backup the project and system configuration

   Even though this is only a sample project, we will backup the project and controller configuration. This is easy to do with Epson RC+ 8.0. It is important that you keep regular backups of your applications on an external media such as a USB memory key.

   Follow these steps to backup the project and system configuration:

   i. In the Epson RC+ 8.0 menu, select [Project] - [Copy Project].

   ii. On the [Copy Project] dialog - [Project Save Destination] box, select the file destination.

By clicking on the [...] button to the right, you can add a project save destination.

iii. Click the [OK] button.

Project backup files will be copied to external media.

iv. From the [Tools] menu, select [Controller].

v. Click on the [Backup Controller] button.

vi. In the [Browse For Folder] window, select a drive.

vii. Click the [OK] button. The system configuration will be backed up on the external media.

You have now written your first program. For details on basic application design, see below.

Designing Simple Applications

6. Epson RC+ 8.0 GUI

The chapter contains information on the Epson RC+ 8.0 GUI.

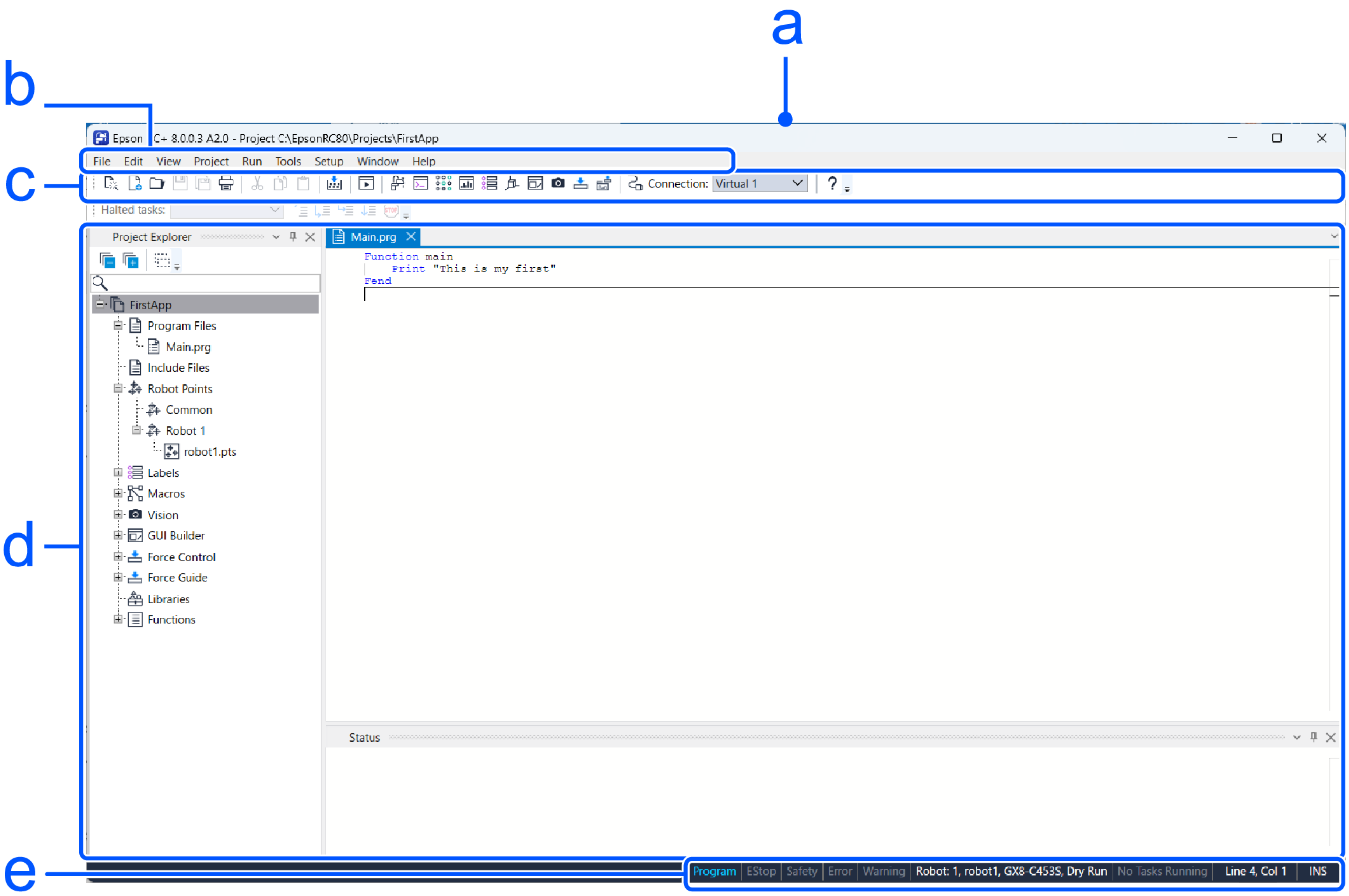
- Overview
- Subwindows
- Window layout
- Tool Bar
- Status Bar
- Help
- [File] Menu
- [Edit] Menu
- [View] Menu
- [Project] Menu
- [Run] menu
- [Tools] Menu
- [Setup] Menu
- [Window] menu
- [Help] Menu

6.1 GUI Overview

Epson RC+ 8.0 is an IDE (Integrated Development Environment) application with a docking window. You can drag and drop to group windows into tabs, and display separate windows on external displays. You can customize the window layout to your liking based on the size and number of monitors in use.

Your customized window layouts can saved for later use. You can load and quickly switch between easy-to-use layouts.

The following window will appear when starting Epson RC+ +8.0.



Symbol	Item	Description
a	Main Window	This appears when starting Epson RC+8.0 in Program mode.
b	Menu Bar	This displays menus of options that can be ran in Epson RC+8.0.
c	Tool Bar	This displays buttons for frequently used commands.
d	Subwindows	This displays a window for the Program Editor, Point Data, Project Explorer, Status Window, and other functions.
e	Status Bar	This displays status messages, such as project build status, system errors and warnings, etc.

6.2 Subwindows

You can customize the position and size of subwindows in Epson RC+8.0.

There are two types of subwindows available.

- Document window: Window used to edit and use program files, point data, and other contents.
- Tool window: Auxiliary windows, such as Project Explorer and Status.

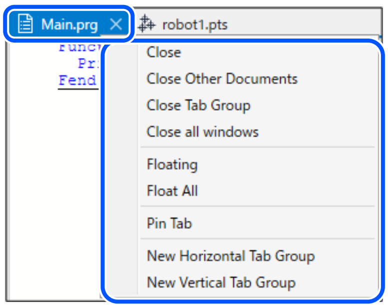
TIP

You can distinguish between subwindow types by putting them in a floating state. The Tool window displays the following at the top of the window.

6.2.1 Window Operation

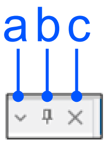
Document window

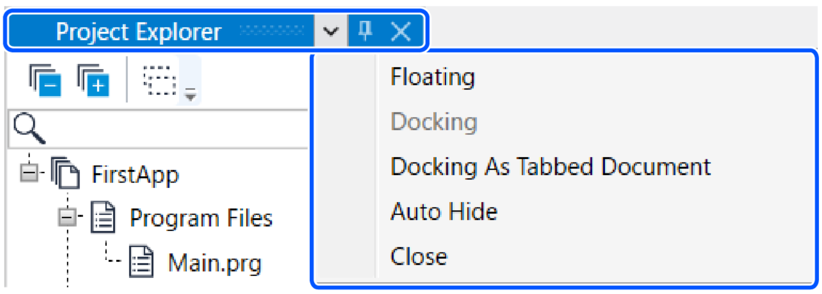
Right-click the window title bar to display the context menu.



Tool window

The following icons controlling the window appears in the top right of the window.



Symbol	Description
a	This displays the context menu. You can also show the context menu by right-clicking on the title bar of the window. 
b	This minimizes the window as a tab. Click  [Auto-Hide] to display the window again.
c	Closes and hides the window.

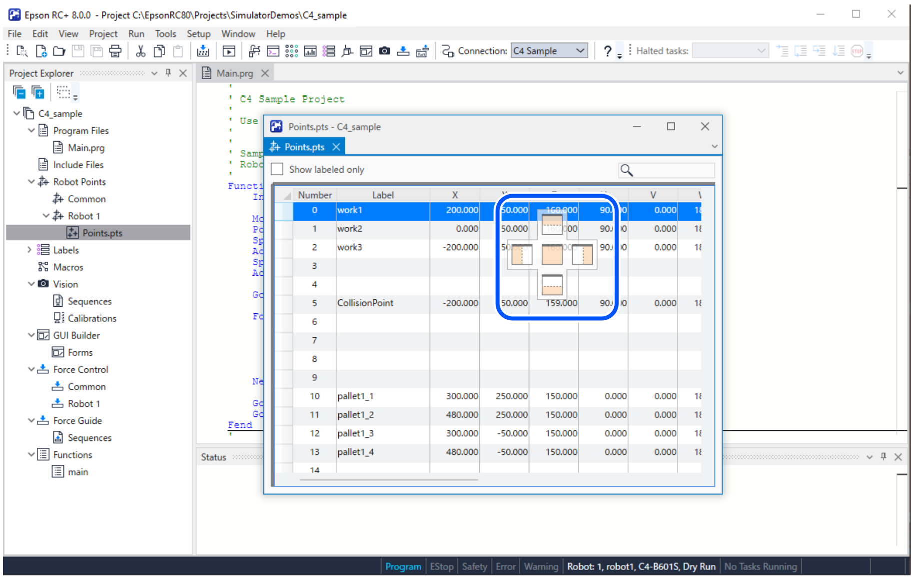
6.2.2 Docking a Subwindow to the Main Window

Drag the title bar of the subwindow you want to dock and place it over the window you want to dock, then the guide appears. Follow the guide and drop the subwindow to dock it.

The docking position will vary depending on the window.

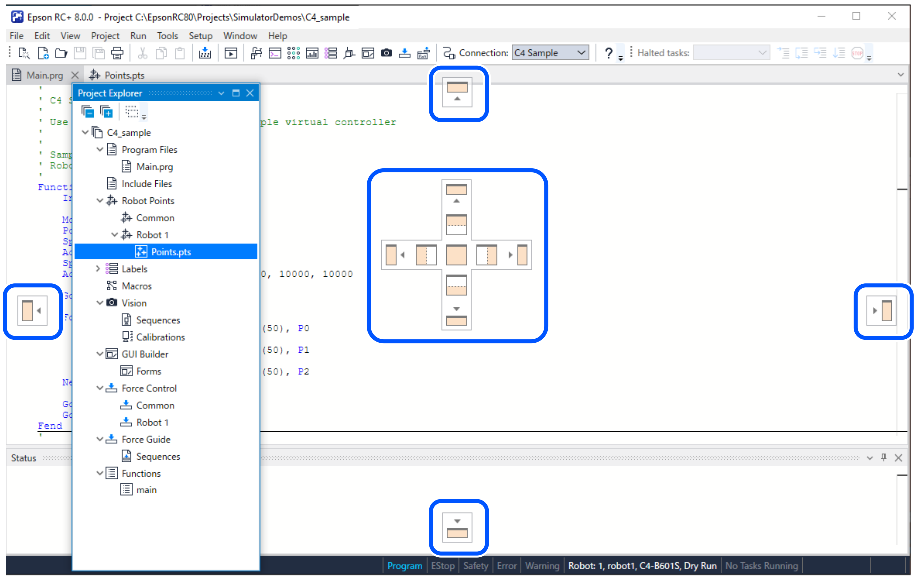
Document window

You can dock subwindows to an open document window.



Tool window

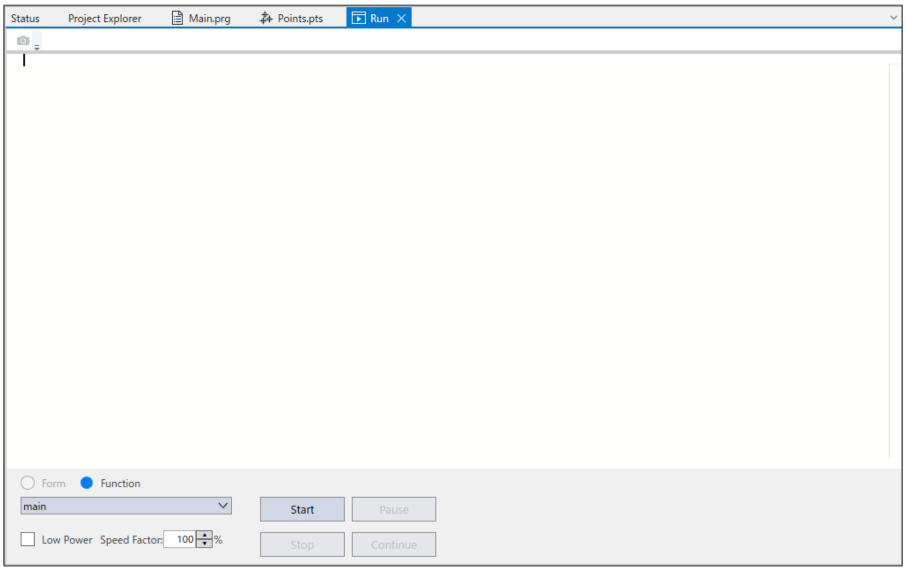
You can dock subwindows to one side of the main window, or an open document or tool window.



> **TIP**
>
> Click [Auto-Hide] in the Tool window to unpin and minimize unused windows as tabs. To display hidden windows again, click on the minimized tab. Tool windows displayed from tabs overlap on the document window, making this useful when working with a small monitor.

6.2.3 Subwindow Tabs

You can drag and drop different subwindows to be displayed on top of one another as tabs.



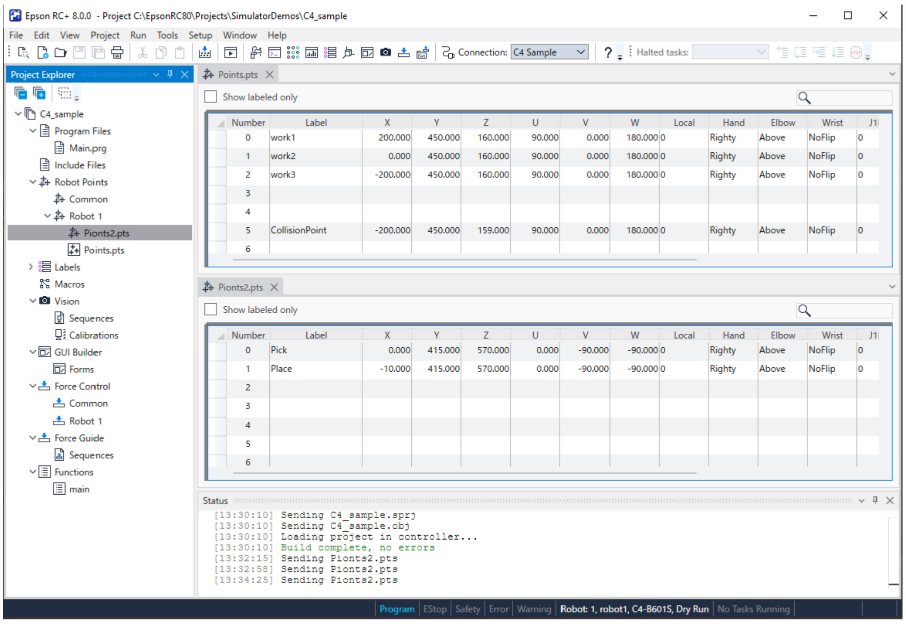
To display an inactive windows to the front, either click on the tab, or press the corresponding shortcut key (the F5 key for the Run window, for example).

6.2.4 Split and Display Subwindows

You can split tabbed subwindows on the top and bottom or left and right to create a new tab group.

Horizontal tab groups (arrange vertically)

This is useful when simultaneously referencing point data and other tabular data.

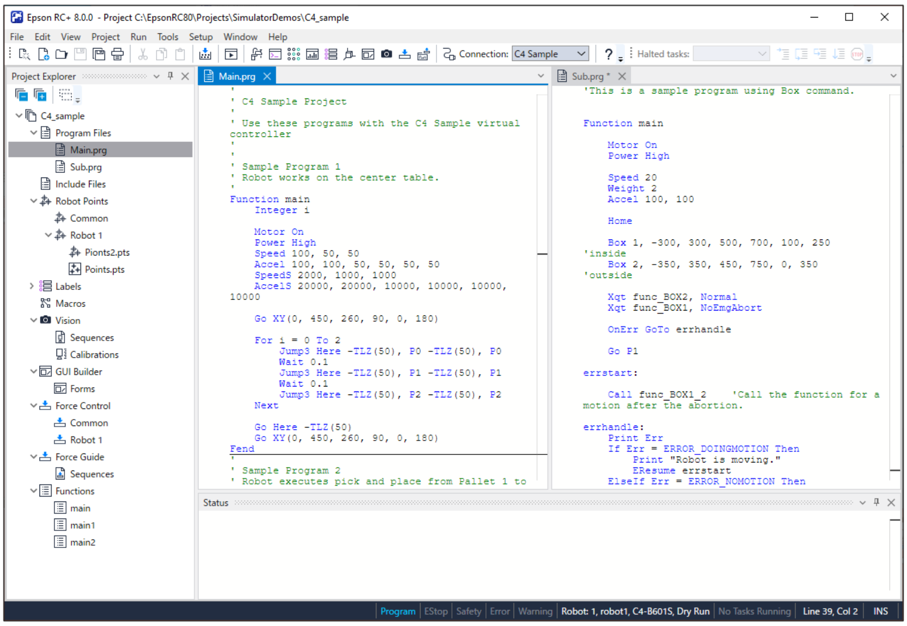


Arrange the subwindows vertically in one of the following ways.

- Right-click on the title bar of the window, and then select [New Horizontal Tab Group].
- In the Epson RC+8.0 menu, select [Window]-[New Horizontal Tab Group].
- Drag the title bar of the window and dock it above or below the other window.

Vertical tab groups (arrange horizontally)

This is useful when simultaneously referencing programs and other vertically formatted data.



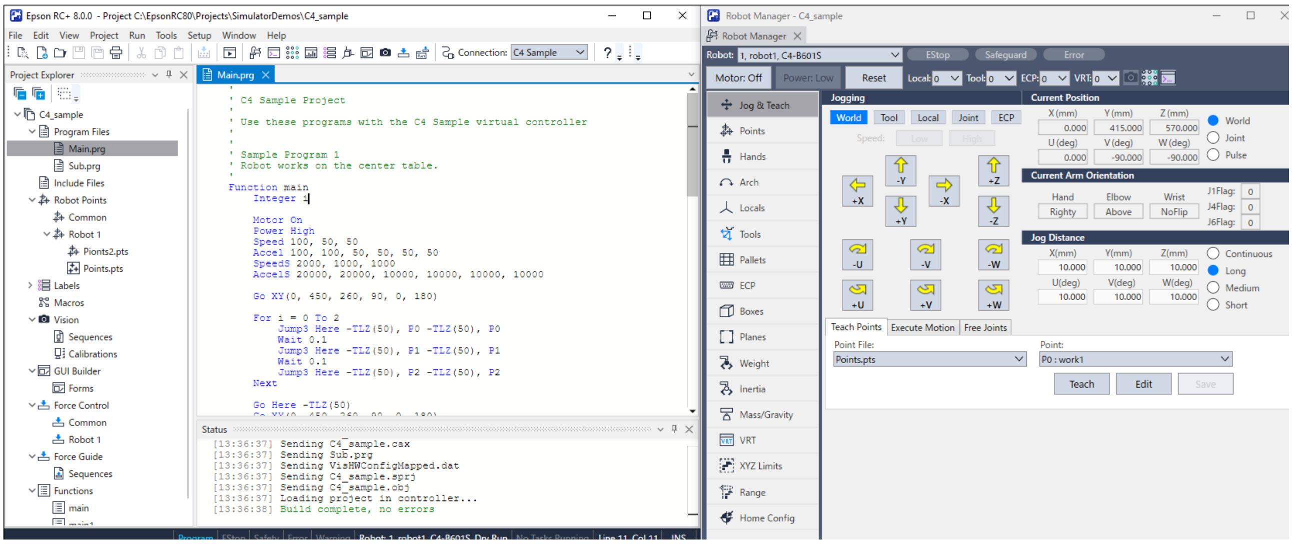
Arrange the subwindows horizontally in one of the following ways.

- Right-click on the title bar of the window, and then select [New Vertical Tab Group].
- In the Epson RC+8.0 menu, select [Window]-[New Vertical Tab Group].
- Drag the title bar of the window and dock it to the left or right of the other window.

6.2.5 Floating Subwindows

Windows can be put into a floating state (display as a separate window). This is useful for maximizing the space available by displaying floating windows on an external display.

Drag the title bar of the subwindow you want to float and drop it wherever the docking guide does not appear.



Floating windows can be displayed at the front of the main window at any time.

Right-click on the title bar of the window, and then select either [Show All Floating Tool Window On Top] or [Show All Floating Document Window On Top].

6.3 Window layout

6.3.1 Save Window Layout

You can name and save the current window layout information. In the Epson RC+ 8.0 menu, select [Window]-[Save Window Layout] to save.

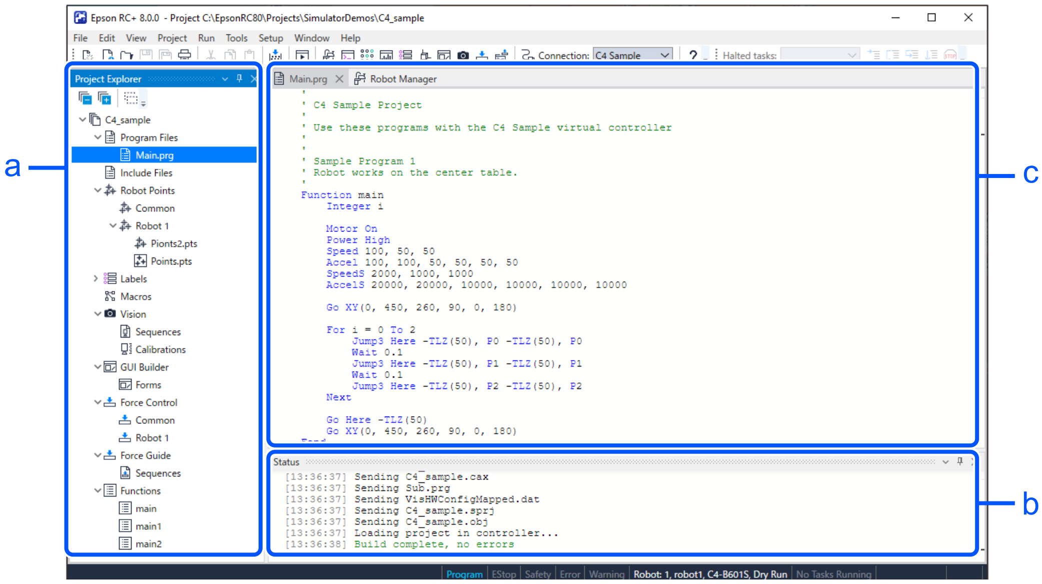
Saved layouts can be loaded from [Window]-[Apply Window Layout].

6.3.2 Reset Window Layout

You can reset the window layout. In the Epson RC+ 8.0 menu, select [Window]-[Reset Window Layout] to save.

When [Standard] is selected

The returns the user to the initial layout. This layout is intended for single displays.



Symbol	Item	Description
a	Project Explorer	[Project Explorer] Command (View Menu)
b	Status Window Pane	[Status Window] (View Menu)
c	Main area	This will place subwindows other than Project Explorer, Status Window, and Search Results.

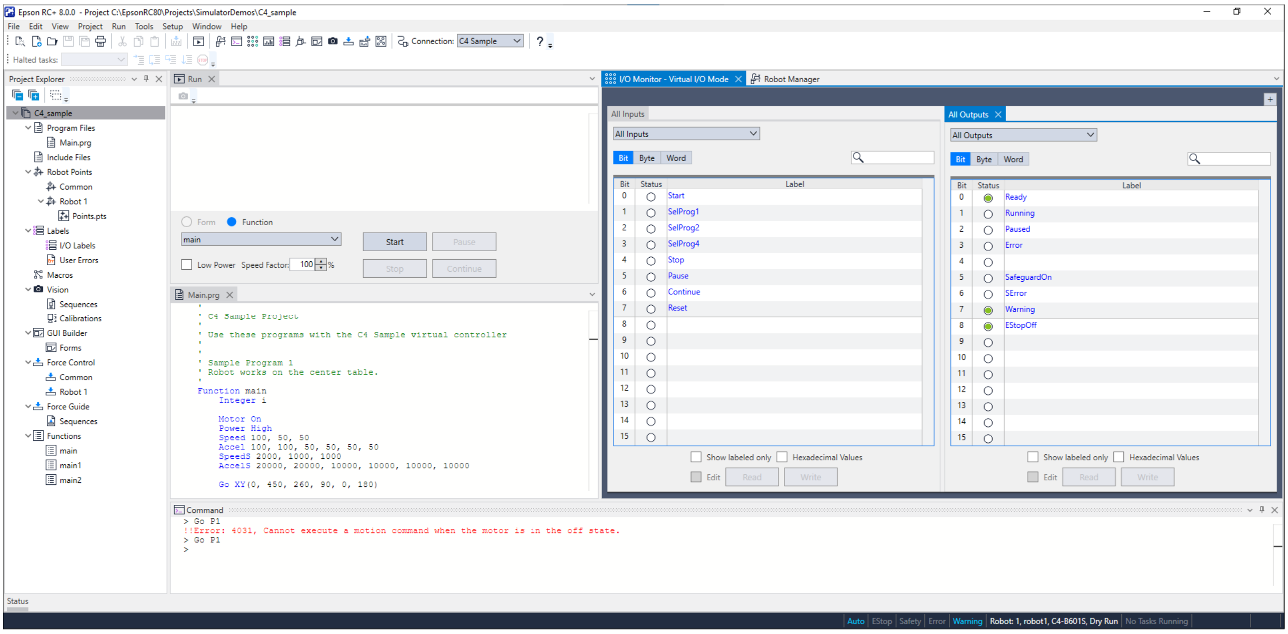
When [Layout for multiple displays] is selected

This will display functions that are easier to use on a larger screen (Vision Guide, ForceGuide, Force Monitor, Simulator, GUI Builder), and the Run window used for debugging in a floating state. This layout is intended for multiple displays. Place windows on the external display.

6.3.3 Window Layout Configuration Examples

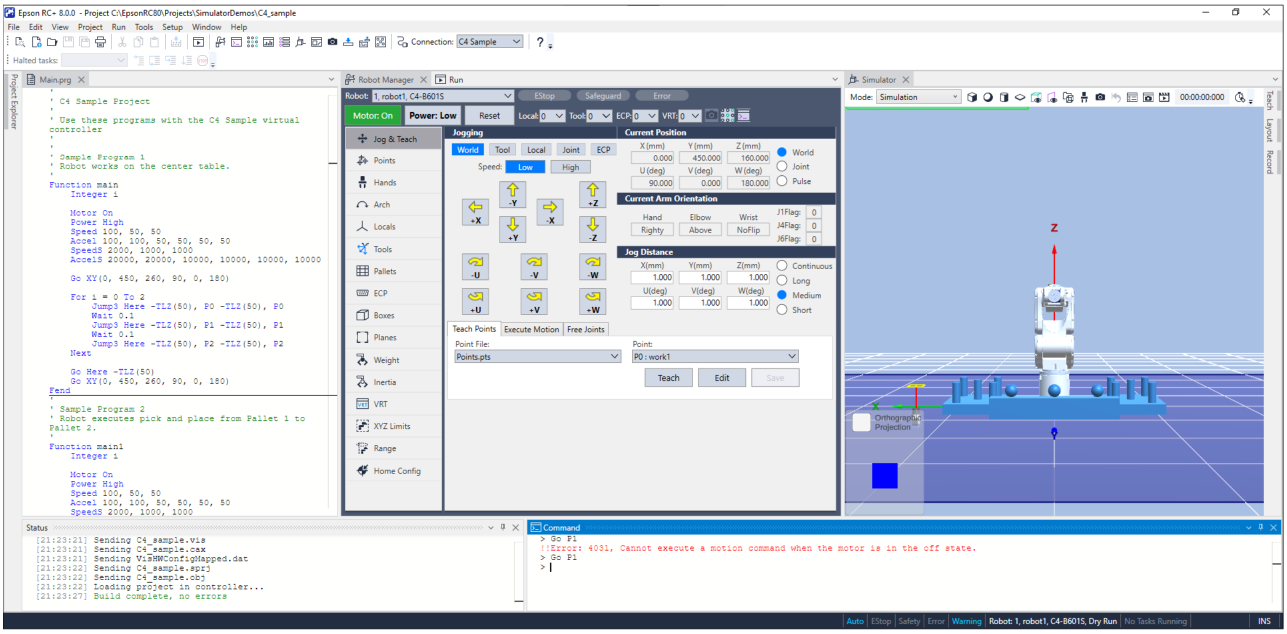
The following section provides examples of window layouts based on tasks performed.

Example 1: Window layout for programming/debugging



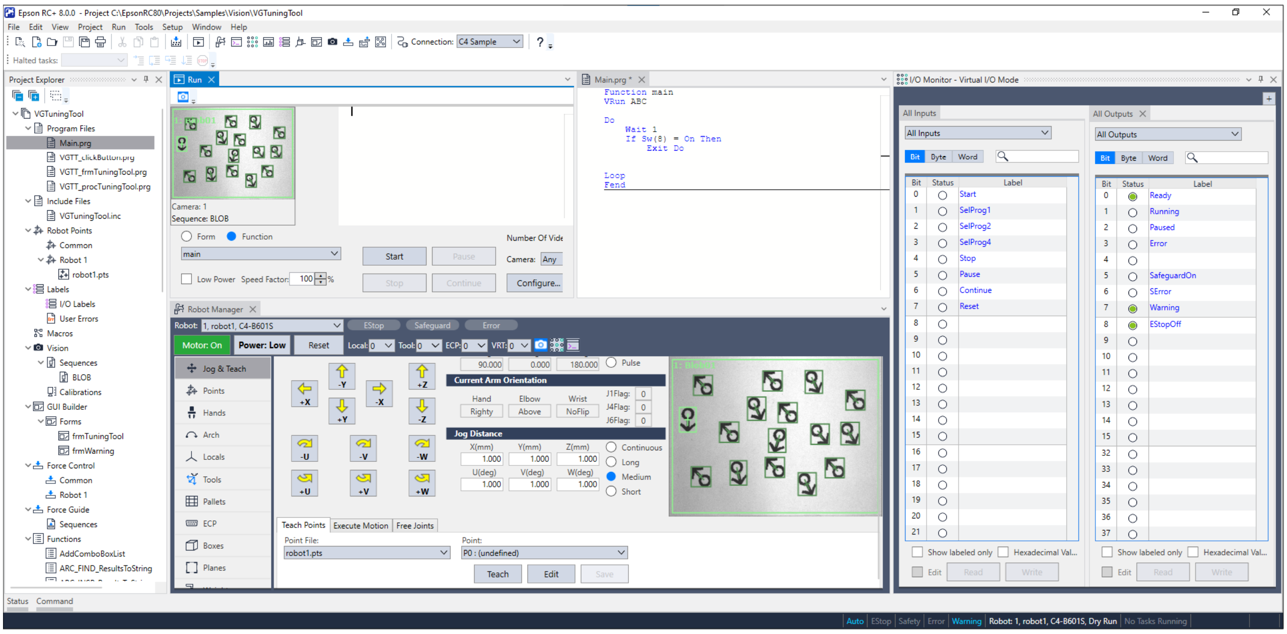
POINT: This layout allows you to debug the program while checking RunWindow information.

Example 2: Window layout for programming while checking robot operation in the Simulator



POINT: To save screen space, the Project Explorer is automatically hidden. This allows for various robot motion controls using sample programs, jog panel, and command window.

Example 3: Programming with Vision



POINT: To save screen space, the Status Window and Command Window are automatically hidden. You can check the vision sequence results from the [Run] window and [Robot Manager].

6.4 Tool Bar

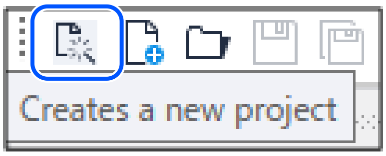
The main tool bar, located under the menu bar on the main window, has buttons for commonly used commands. Each button on the tool bar corresponds to a submenu selection from the menu bar. For example, the [Project Wizard] tool bar button corresponds to the [Project Wizard] command in the [Project] menu.

Button	Shortcut	Description
Project Wizard	-	Starts project wizard.
New Files	Ctrl + N	Creates a new file.
Open File	Ctrl+O	Opens a file.
Saving the File	Ctrl+S	Saves the active files.
Save All	Ctrl+Shift+S	Saves all open files.
Print	Ctrl+P	Opens the [Print] dialog box.
Cut	Ctrl+X	Copies the current selection into the Clipboard and then deletes the selection.
Copy	Ctrl+C	Copies the current selection into the Clipboard.
Paste	Ctrl+V	Puts the contents of the Clipboard into the currently active document starting at the insertion point.
Find	Ctrl+F	Finds a text string in the program.

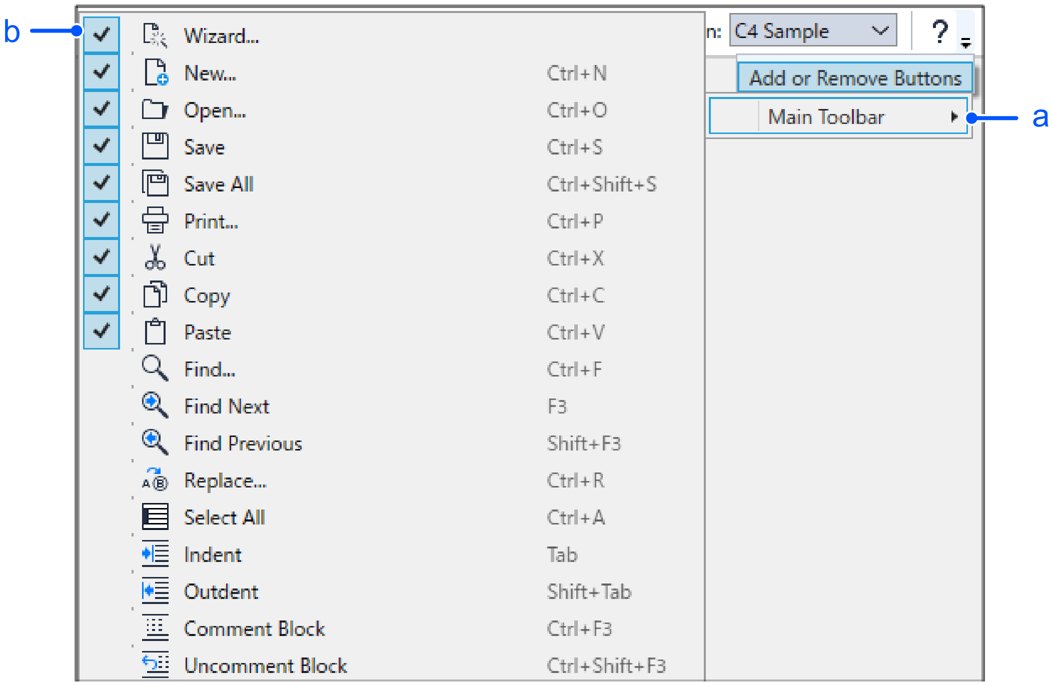
Button	Shortcut	Description
Find Next	F3	Finds the next occurrence of the search text specified in the last [Find] command.
Find Previous	Shift+F3	Finds the previous occurrence of the search text specified in the last [Find] command.
Replace	Ctrl+R	Searches for a text string and replace it with new text.
Go to Line	Ctrl+G	Displays [Go to Line] dialog.
Select All	Ctrl+A	Selects the entire files.
Indent	Tab	Moves the selected line one tab to the right.
Outdent	Shift+Tab	Moves the selected line one tab to the left.
Comment Block	Ctrl+F3	A comment character will be added to the beginning of each of the selected lines.
Uncomment Block	Ctrl+Shift+F3	The first comment character from each of the selected lines will be removed.
Building a project	Ctrl+B	Builds the current project so that it can be executed.
Run window	F5	Opens the [Run] window.
Setting/clearing breakpoints	F9	Sets the selected line as a breakpoint or returns it to normal.
Robot Manager	F6	Opens the [Robot Manager] window.
Command Window	Ctrl+M	Opens the [Command] window.
Monitoring I/O	Ctrl+I	Opens the [I/O Monitor] window.
Task Manager	Ctrl+T	Opens the [Task Manager] window.
Macros	-	Opens the [Macros] Editor.
IO Label Editor	Ctrl + L	Opens [IO Label Editor].
User Error Editor	Ctrl+U	Opens the [User Error] editor.
Simulator	Ctrl+F5	Opens the [Simulator] window.
GUI Builder	Ctrl+F7	Opens the [GUI Builder] window.
Part Feeding	Ctrl+F12	Opens the [Part Feeding] window.

Button	Shortcut	Description
Vision	Ctrl+F9	Opens the [Vision Guide] window.
Force Guide	Ctrl+F11	Opens the [Force Guide] window.
Force Monitor	Ctrl+F10	Opens the [Force Monitor] window.
Controller Connection with PC	-	Opens the [PC to Controller Communications] window.
Connection:	-	Displays the connection status with the controller.
Help	F1	Opens the help in your browser.

Hovering the mouse over a tool bar button will show a tool tip which briefly describes the command. For example, when the mouse pointer is over the "Project Wizard" tool bar button, the tool tip displays "Creates a new project."

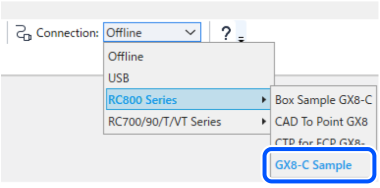


You can add or remove command buttons on the tool bar by clicking the [Add or Remove Buttons] dropdown arrow button, and then select [Main Toolbar]. Next, click the command you want to add or remove. All tool bar command buttons have fixed positions. Adding and removing tool bar buttons is equivalent to showing or hiding the buttons.



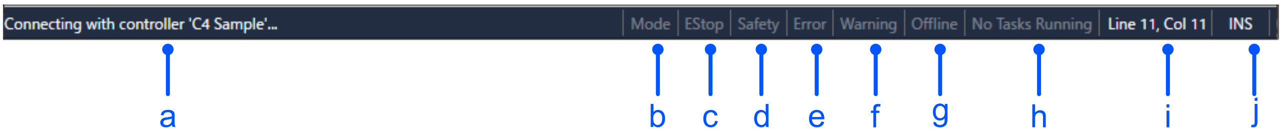
Symbol	Description
a	Click the drop-down arrow button.
b	Check this box to show or hide the button.

If the drop-down list is selected in the [Connection:] toolbar while offline, the most recent connection destination is highlighted in blue.



6.5 Status Bar

The status bar is located at the bottom of the main window. This displays the following.



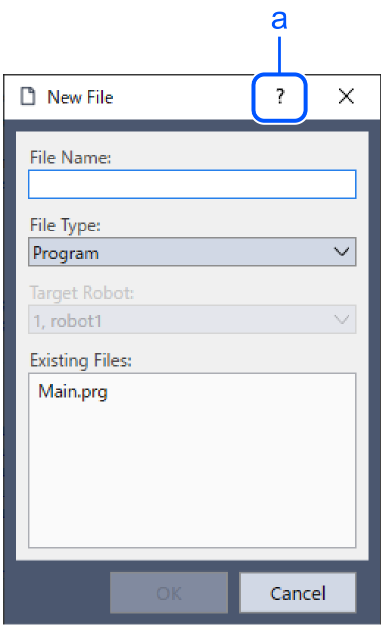
Symbol	Item	Description
a	Message area	Displays the syntax error for the current line and system messages.
b	Operation Mode status	Indicates the controller operation mode.
c	Emergency Stop status	"EStop" is displayed in color if emergency stop is active.
d	Safeguard status	"Safety" is displayed in color if a safeguard circuit is open.
e	Error status	"Error" is displayed in color if the controller is in the error state.
f	Warning status	“Warning” is displayed in color if there is a warning. Put the mouse cursor over the Warning status area to see the warning message.
g	Current robot	Displays the currently selected robot number, name, type number, and the dry run status.
h	Tasks running status	Indicates that one or more tasks are running.
i	Current Line and Column	When a program editor window is active, the current line and column are displayed.
j	INS / OVR	Indicates "insert" or "overtype" mode.

6.6 Help

Epson RC+ 8.0 has a help system that can be easily brought up when starting Epson RC+ 8.0.

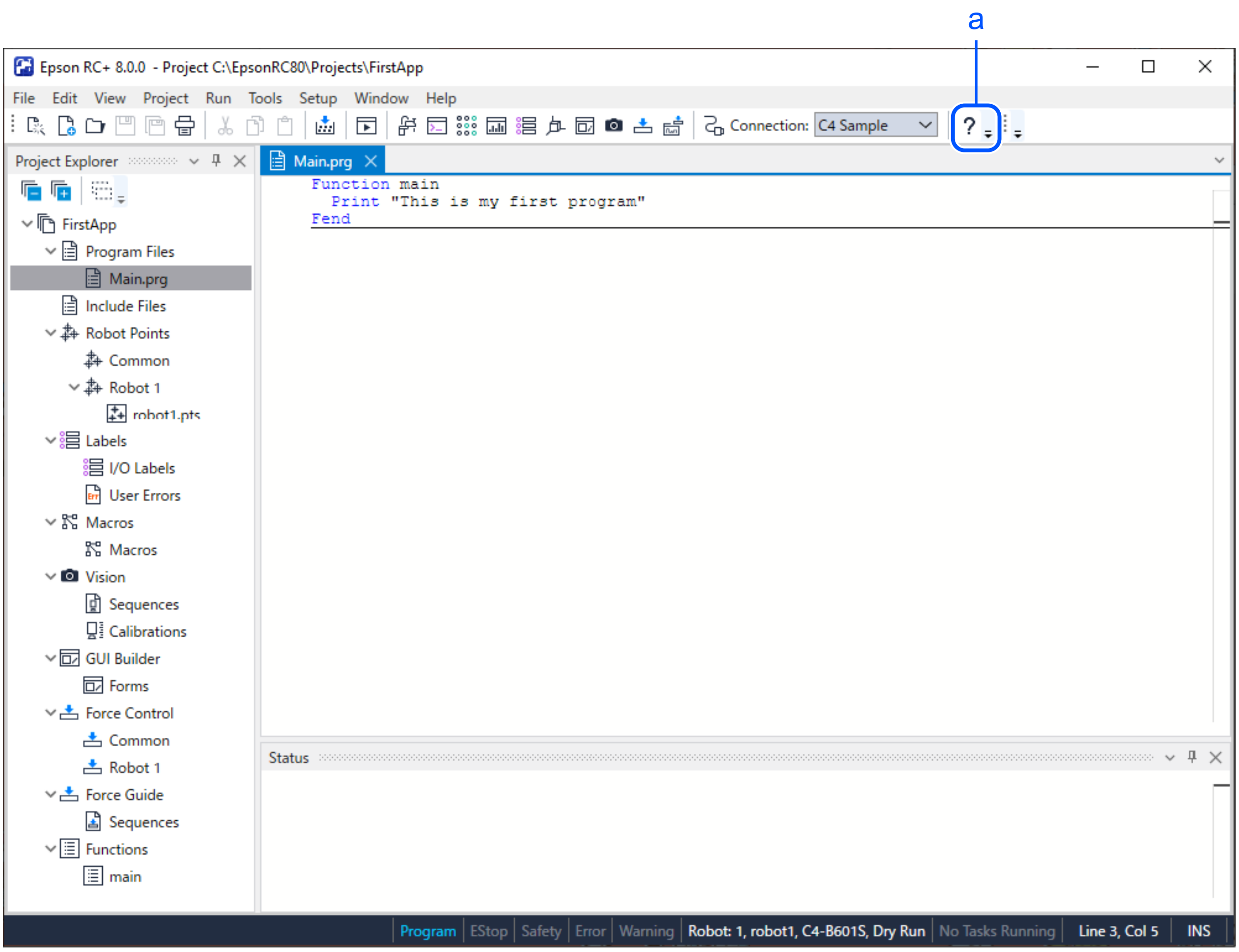
There are several methods to get help.

- From the Windows desktop, click [Start]-[Programs]-[Epson RC+ 8.0]-[Epson RC+ 8.0 Help] to view help topics.
- In the Epson RC+ 8.0 menu, select [Help]-[Help] to browse help topics.
- When editing programs, place the cursor over the keyword of interest and press the [F1] key.
- When a dialog is open, press the [F1] key or click the Help button. For dialogs, the Help button is located in the window title bar on the right side and is shown as a question mark icon as shown below.



Symbol	Description
a	Dialog Help Button

For MDI child windows, the Help button is located on the main toolbar and is also shown as a question mark icon as shown below.



Symbol	Description
a	Help button on the active sub-window

6.7 [File] Menu

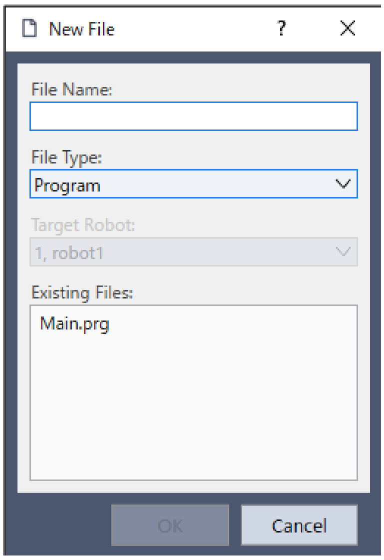
From the Epson RC+ 8.0 menu-[File], you can manage and print files in the current project.

6.7.1 [New File] (File Menu)

: Ctrl + N

The New command is used to add new files to the current project.

When the [New File] is selected, the [New File] dialog is opened.



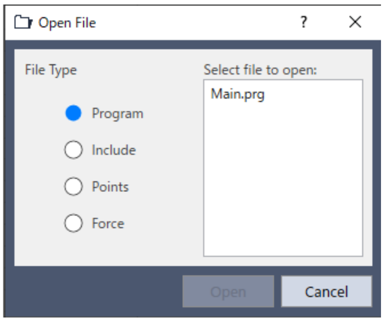
Item	Description
File Name	Enter a name for the new file in this box. Up to 24 characters can be input.
File Type	You can also specify program files, include files, point data, and force data.
Target Robot	Select the robot to which you want to apply the point file or force file. ▪ Common: Applies to all robots ▪ robot x : Only apply to the specified robot
Existing Files	Shows the files for the selected type currently in the Project folder.
OK	Creates a new file.
Cancel	Cancels the operation.

6.7.2 [Open File] (File Menu)

: Ctrl+O

Open one or more files in the current project for editing. (You can open program files, include files, point files, or force data.)

Files excluded from the current project cannot be opened. You must add the file to the project before you can open it. (This also applies to include files, point files, and force data).



Item	Description
File Type	Displays a list of current project files. You can also specify program files, include files, point data, and force data. For details on editing points, see the following: **Editing Points**
Select file to open	Click on the file name you want to open. ▪ You can select more than one file by using the [Ctrl] key or [Shift] key. ▪ The [Ctrl] key allows you to select or deselect any file. ▪ The [Shift] key allows you to select a group of files.
Open	Opens the selected file(s).
Cancel	Cancels the operation.

TIP

You can also double click on a file name in the [Select file to open] box to open the file. You do not need to click the [Open] button.

6.7.3 Close Command (File Menu)

Ctrl+D

Close the currently active window.

Any of the windows can be closed with this command: Programs, Include files, Point files, Command Window, Run window, I/O Label Editor, user errors.

TIP

You can also close a window or dialog box by double clicking on the control box button located in the upper left corner of the window or dialog box.

6.7.4 Save Command (File Menu)

: Ctrl+S

Saves the current file. This menu selection cannot be selected if there is no new file to be saved.

6.7.5 [Restore] (File Menu)

Reloads the program files, include files, I/O labels, user errors, or point files that you are currently editing from the hard disk.

Use this function to change a document to the state it was in since last saved.

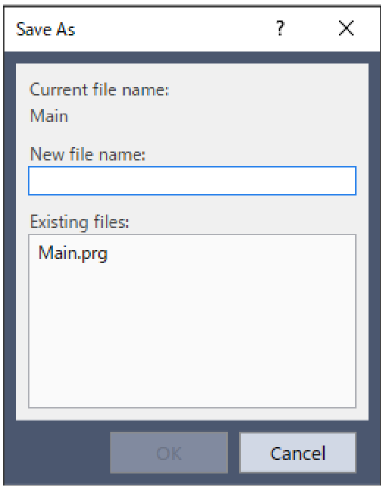
You will be prompted to confirm this operation.

6.7.6 [Save As] (File Menu)

Save the program file, include file, point file, or force file with a new file name. The original file will be removed from the project but will remain on the hard disk.

To display the original excluded file, select [Project Explorer]-[Show files except project build].

If you use [Save As] on an include file, you must rename the file in each of your #include statements that refer to it.



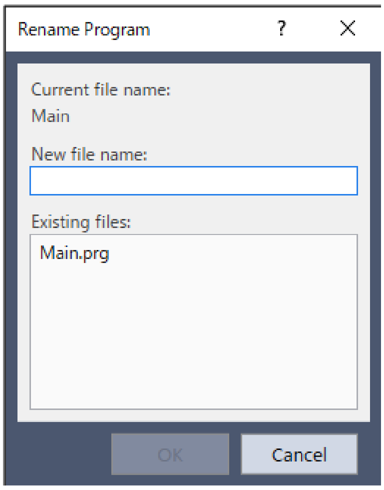
6.7.7 [Save All] (File Menu)

: Ctrl+Shift+S

Saves all open files. This menu selection cannot be selected if there is no new file to be saved.

6.7.8 [Rename Program] (File Menu)

Changes the name of the program file, include file, point file, or force file you are currently editing.



Rename Program

1. Select and activate the file to be renamed.
2. Select the Epson RC+ 8.0 menu-[File]-[Rename Program].
3. Enter a new name for the file, and then click the [OK] button.

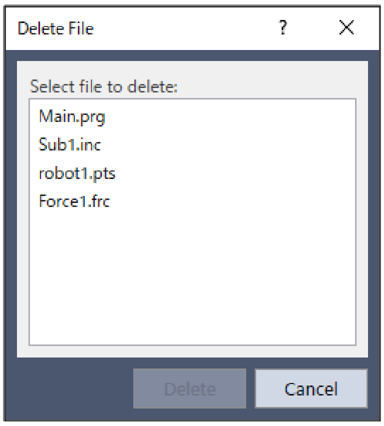
The new file name cannot be the same as the existing files. You will get an error message if you enter a new name that is already being used.

If you use [Save As] on an include file, you must rename the file in each of your #include statements that refer to it.

6.7.9 [Delete File] (File Menu)

This command allows you to delete a file in the current project folder. You can delete the program file, include file, point file, or force file.

The file does not have to be registered in the project to delete.



Item	Description
Select file to delete	Click on the file name you want to delete. This list displays all program files, include files, point files, and force files in the current project folder.
Delete	Deletes the selected file. You will be prompted with a confirmation message before the file is deleted. If the file is currently open, it will be closed and removed from the current project before it is deleted from disk.

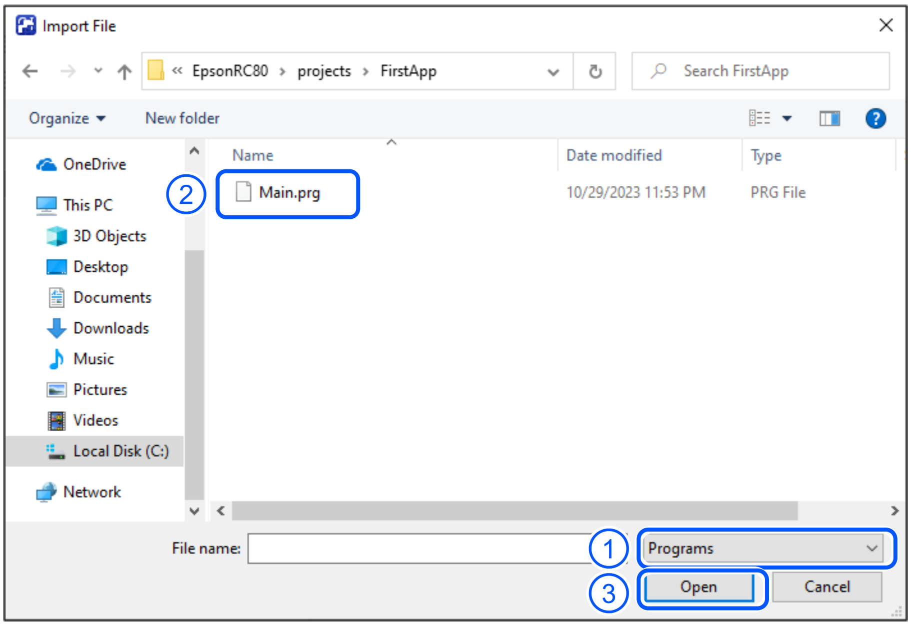
Item	Description
Cancel	Cancels the operation.

6.7.10 Import Command (File Menu)

Import a file from other Epson RC+ 8.0 projects.

The extensions and file names that can be imported are as follows. Be careful.

- Program file names must have a .PRG extension.
- Include file names must have a .INC extension.
- Point file names must have a .PTS extension.
- GUI Builder file names must have a .GUI extension.
- Vision Guide file names must have a .VIS extension.
- Force file names must have a .FRC extension.
- Force Guide file names must have a .FG extension.
- Parts Feeding file names must have a .PF extension.
- I/O labels must have the file name IOLabels.dat
- User errors must have the file name UserErrors.dat
- Macro files must have the file name Macros.dat



To import a file

1. Select the file type from the [File Type] list box.
2. Navigate to the file you want to import.
3. Click the [Open] button.

   If a file name is already used in the project folder, you will be prompted to confirm overwrite. The file will then be copied to the current project's folder.

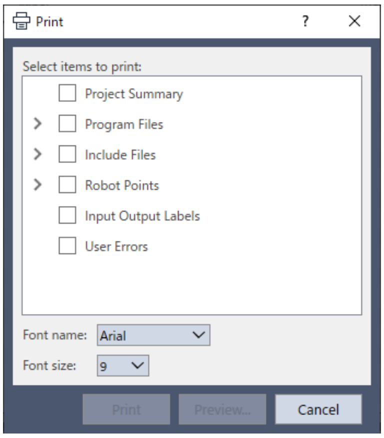
> ✎ KEY POINTS
>
> You can only import files created with Epson RC+ 8.0 or EPSON RC+ 7.0. When importing files created with earlier versions than EPSON RC+ 7.0, import the files into EPSON RC+ 7.0, and then import the imported files into Epson RC+ 8.0.

6.7.11 Print Command (File Menu)

: Ctrl+P

The [Print] dialog appears, allowing you to print multiple program files, include files, point files, I/O labels, and user errors. Also you can print out a project summary.

The page number, product name, project name, file name, and last modified timestamp are printed in the header.



Item	Description
Select items to print	Check the items in the tree that you would like to print out.
Project Summary	Select this checkbox to print a summary of the programs and points used in the current project.
Program Files	Select this checkbox to print all program files. (Click the [>] button to select program files to print.)
Include Files	Select this checkbox to print all include files. (Click the [>] button to select include files to print.) This checkbox is not shown if there are no include files in the current project.
Robot Points	Select this checkbox to print all point files. (Click the [>] button to select point files to print.)
Input Output Labels	Select this checkbox to print a listing of the all of the I/O labels used in the project.
User Errors	Prints a listing of all user errors for the current project. If either the label or message is non-blank, then the error definition will be printed.
Font Name	Opens a dialog for selecting the printer font. The selected font is saved for subsequent printing. This setting will be applied when printing again.
Font Size	Opens a dialog for selecting the printer font size. The selected font size is saved for subsequent printing. This setting will be applied when printing again.

Item	Description
Print	Prints the selected files. This button will be dimmed if nothing is selected to be printed.
Preview	Preview the selected files before printing. This button will be dimmed if nothing is selected to be printed.
Cancel	Cancels the operation and closes the dialog.

6.7.12 Exit Command (File Menu)

Alt+F4

Exits the Epson RC+ 8.0.

If you are running a program from the [Run] window and the control device is PC, you will see a message that a program is running and you will not be allowed to exit. You must stop all tasks first before you can exit.

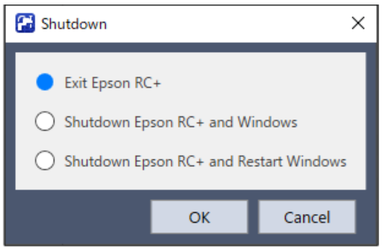
A dialog will appear prompting you to save any unsaved program files, include files, point files, I/O labels, or user errors. Click the [Yes], [No], or [Cancel] button to continue.

- Select [Yes] to save the file and close RC+.
- Select [No] to close RC+ without saving the files.
- Select [Cancel] to return to the Epson RC+ 8.0 main window.

If the display of the dialog at Epson RC+ 8.0 shutdown is enabled, the following dialog will be displayed at the shutdown and you can select a termination process.

For details on the shutdown dialog, see the following:

[Preferences] Command (Setup Menu)



Item	Description
Exit Epson RC+	Exits the Epson RC+ 8.0.
Shutdown Epson RC+ and Windows	Exits the Epson RC+ 8.0 and shutdown the Windows.
Shutdown Epson RC+ and Restart Windows	Exits the Epson RC+ 8.0 and reboot the Windows.
OK	Executes the selected operation.
Cancel	Cancels the operation and closes the dialog.

6.8 [Edit] Menu

You can edit the program files from the Epson RC+ 8.0 menu-[Edit].

TIP

By right-clicking the program edit window, [Edit] can be displayed.

6.8.1 [Undo] Command (Edit Menu)

: Ctrl+Z

Undo the changes to the currently active program since it was open.

6.8.2 [Redo] Command (Edit Menu)

: Ctrl+Y

Redo the previous undo.

6.8.3 [Cut] Command (Edit Menu)

: Ctrl+X

Copies the current selection into the Clipboard and then deletes the selection.

6.8.4 [Copy] Command (Edit Menu)

: Ctrl+C

Copies the current selection into the Clipboard.

6.8.5 [Paste] Command (Edit Menu)

: Ctrl+V

Puts the contents of the Clipboard into the currently active document starting at the insertion point.

6.8.6 [Find] Command (Edit Menu)

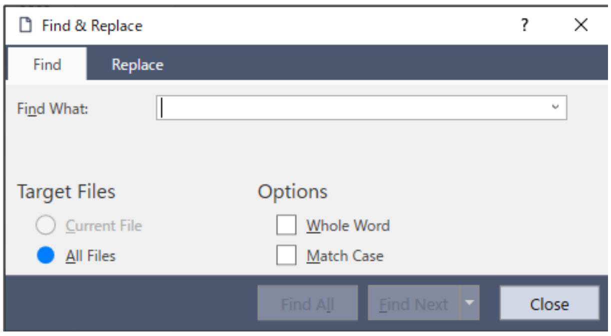
: Ctrl+F

Finds a text string in the program.

Note that encrypted files will not be searched.

For details on using encrypted files, see the following:

Using Encrypt Files



Item	Description
Find What	Type the text you want to search for. If any text was selected when you execute the Find command, it will be displayed here. When executing [Find] with a string selected, the selected string is displayed. If no text was selected, then the text from the last Find will be displayed. You are limited to one line of text. If selecting more than one line before executing Find, the search will not start.
Current File	Searches only in the current program file and include file.
All Files	Searches all files in the project.
Whole Word	Searches for the full word by itself and not as part of another word.
Match Case	Text must also match lower and upper case in order to be found.
Find All	Search for all occurrences in the program, and list the results in the [Search Results] window. Each result shows the file name, line number, and line where the text was found. You can then double click on a result to open the file where the text was found. This button will be dimmed if nothing is entered to be searched.
Find Next / Find Previous	Finds the next, or previous occurrence. (Click the [▼] button to select the search method.) If the text is found in a file that is not open, then the file will be opened to display. This button will be dimmed if nothing is entered to be searched.
Close	Closes the dialog box.

6.8.7 [Find Next] Command (Edit Menu)

: F3

Finds the next occurrence of the search text specified in the last [Find] command.

6.8.8 [Find Previous] Command (Edit Menu)

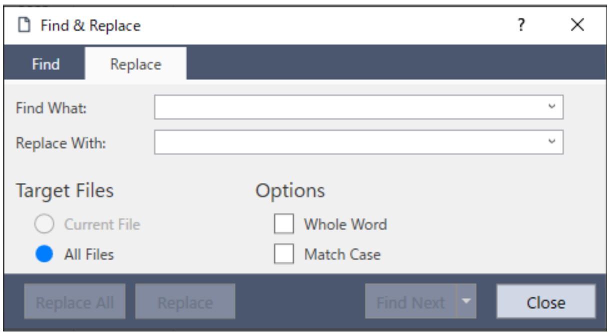
: Shift+F3

Finds the previous occurrence of the search text specified in the last [Find] command.

6.8.9 [Replace] Command (Edit Menu)

: Ctrl+R

Search for a text string and replace it with new text.

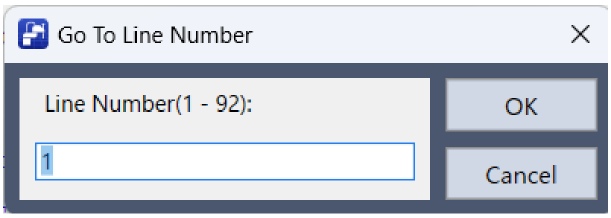


Item	Description
Find What	Type the text you want to search for. When executing [Replace] with a string selected, the selected string is displayed. If no text was selected, then the text from the last Find will be displayed.
Replace With	Enter the replacement text here. The replacement text can be empty.
Current File	Searches only in the current program file and include file.
All Files	Searches all files in the project.
Whole Word	Searches for the full word by itself and not as part of another word.
Match Case	Text must also match lower and upper case in order to be found.
Replace All	Replaces all occurrences.
Replace	If already found, replaces the current find, otherwise searches for the next occurrence.
Find Next / Find Previous	Finds the next, or previous occurrence. (Click the [▾] button to select the search method.)
Close	Closes the dialog box.

6.8.10 [Go to Line] Command (Edit Menu)

: Ctrl+G

Jump to the selected program files and the line number specified by the include files.



Item	Description
Line Number	Enters the line number you wish to jump to.
OK	Jumps to the specified line number.
Cancel	Closes the dialog.

6.8.11 [Select All] Command (Edit Menu)

: Ctrl+A

Selects the entire program file, include file, point file, I/O labels, or user errors. You can then execute Cut or Copy.

6.8.12 [Indent] Command (Edit Menu)

: Tab

Move the selected line one tab to the right.

6.8.13 [Outdent] Command (Edit Menu)

: Shift+Tab

Move the selected line one tab to the left.

6.8.14 [Comment Block] Command (Edit Menu)

: Ctrl+F3

A comment character will be added to the beginning of each of the selected lines. Comments out the selected block of lines by adding the comment character to the beginning of each line.

6.8.15 [Uncomment Block] Command (Edit Menu)

: Ctrl+Shift+F3

The first comment character from each of the selected lines will be removed.

6.8.16 [Go To Definition] Command (Edit Menu)

Shift+F2

Opens the window and sets the line where a function, variable, macro, point label, I/O label, or user error label is defined.

To use,

Click on the identifier in the program window
and select the Epson RC+ 8.0 menu-[Edit]-[Go To Definition].
Right click on the identifier in the program window
and select [Go To Definition] from the Context Menu.

Identifier type	Display
Function name or variable	Program window where a function name or variable is declared.
Point label	Point file which a label is defined.

Identifier type	Display
I/O label	I/O label editor which a label is defined.
User error label	User error which a label is defined.

6.8.17 [Navigate Backward] (Edit Menu)

Ctrl+F2

Move the cursor to the line that was previously displayed by [Go to Definition] command. Repeat the operation to move the cursor through the history of [Go to Definition], starting with the most recent one.

6.8.18 [Navigate Forward] (Edit Menu)

Ctrl+Shift+F2

Move the cursor to the line that was previously displayed by [Navigate Backward] command. Repeat the operation to move the cursor through the history of [Go to Definition] in order from the oldest to the newest.

6.9 [View] Menu

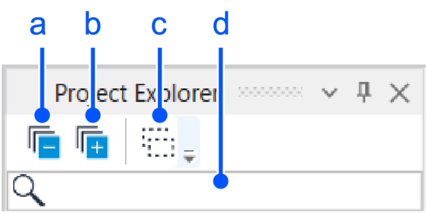
The Epson RC+ 8.0 [View] menu includes commands for opening the Project Explorer, Status Window, Search Results, and Start Window. In addition, there is a command for viewing the system history.

6.9.1 [Project Explorer] Command (View Menu)

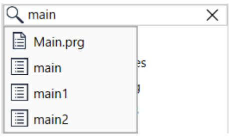
Select this to open the [Project Explorer] pane.

The Project Explorer pane enables you to quickly open any file in the current project or jump to any function. The project files and functions are organized in a sorted tree structure.

Open a file or jump to a function: Double-click on the item.

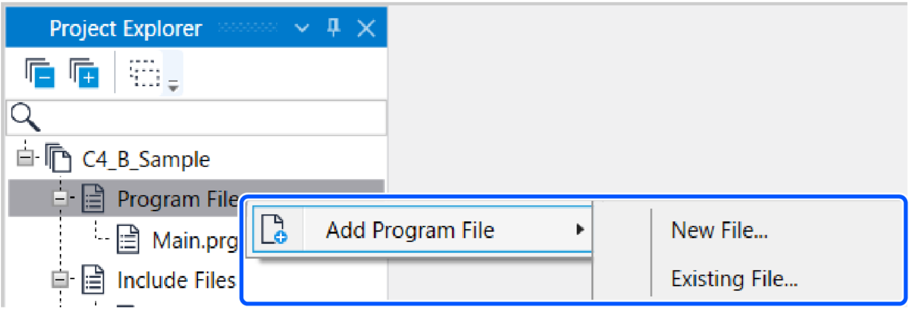


Symbol	Description
a	Collapses all items.
b	Unfolds all items.
c	Displays files except project build. Right-click on an item except the project build. Select [Add To Project] from the displayed context menu to add it to the project.

Symbol	Description
d	Display suggested items that matches the characters entered. By selecting an item from the displayed suggetions, the item will be focused. 

Context Menu

Right-click items in the project tree to display the context menu. To access the context menu, right click on an item in the project tree.



6.9.2 [Status Window] (View Menu)

Select this to display the [Status Window].

The status pane displays status messages, such as project build status, system errors and warnings, etc.

The Status Window is located at the bottom of the main window by default. The Status Window can be moved to any position.

KEY POINTS

If the Status Window is closed and a message is displayed on the Status Window, such as during project build, the Status Window will automatically be opened so that the message can be seen.

6.9.3 [Search Results] Command (View Menu)

Select this to display the [Search Results].

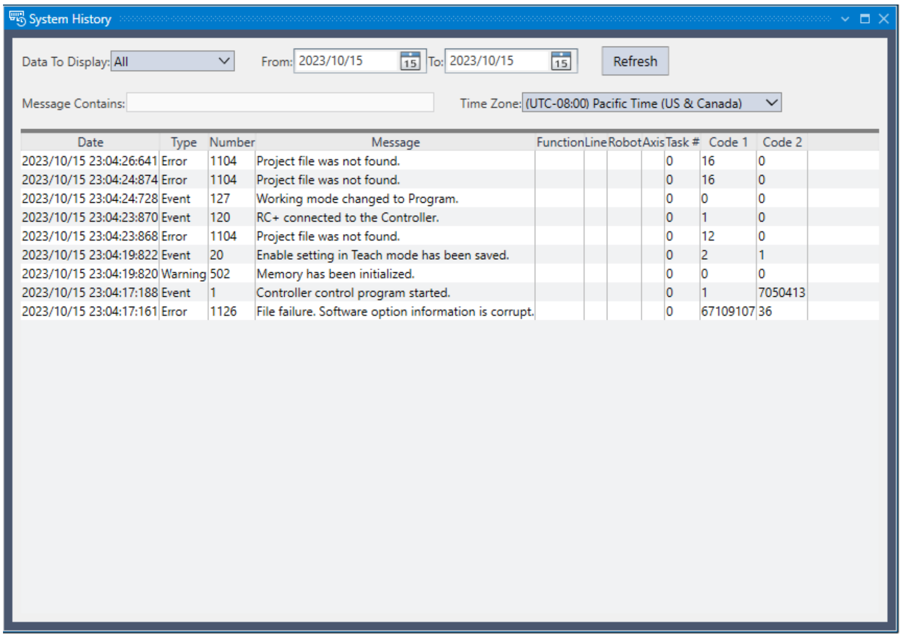
Execute [Find All] in the [Find & Replace] dialog to show the search results.

Each result shows the file name, line number, and line where the text was found. You can then double click on a result to open the file where the text was found.

6.9.4 [System History] (View Menu)

Display [System History]. This window shows events, errors, and warnings that have been logged in the current controller's system history.

The data can be sorted by clicking on any column header. Drop and drop columns to rearrange the order. To sort multiple columns, hold down the shift key and click on multiple columns headers.

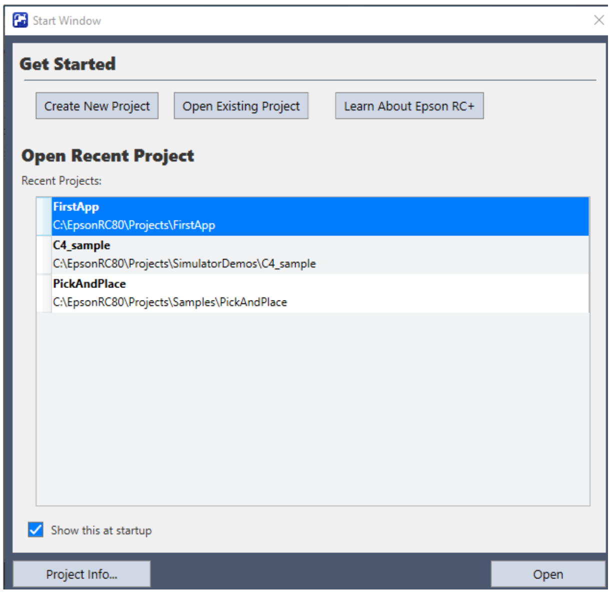


Item		Description
Data To Display		Select which data you would like to view. (All, Error, Event, Warning)
From / To		Select the dates you want to view data from. When the window is first opened, these are automatically set to the first and last dates in the history data.
Message Contains		Enter the error message to search, and then click the [Refresh] button, or press the [Enter] key.
Time Zone		Select a time zone. Time of event, warning, and error occurrences are displayed according to the selected time zone.
Refresh		Click this button to reload the data from the controller.
Type	Event	Information for operation and mode change.
	Warning	Program can be executed continuously, however, needs countermeasure.
	Error	Error occurred in the program or the Robot.
Number		For more details on the number, refer to the following manual: "Status Code / Error Code List" manual
Message		Displays event, warning, and error messages.
	Function and Line number	Function name and the line number are displayed when error occurred while executing a program.
	Robot and axis number	Robot and the axis number are displayed when Robot error occurred.
	Task number	Task number of the task with error is displayed when error occurred while executing the program. "0" is displayed for others.

Item		Description
	Additional information 1 and 2	More details are displayed for some errors. Refer to the following manual: "Status Code / Error Code List" manual

6.9.5 [Start Window] (View Menu)

Display [Start Window]. The Start Window by default opens when RC+ is started.



Item	Description
Create New Project	This button starts the Project Wizard which assists you in creating a new project. See details below. **[Project Wizard] Command (Project Menu)**
Open Existing Project	This button opens the [Open Project] dialog where you can select an existing project to open.
Learn About Epson RC+	This button opens the help system to a page which has topics for learning about Epson RC+.
Recent Projects	This is a list of recently used projects. Each row shows the project name on the first line, and the project path on the second line. You can select a project in the list, then click the [Open] button to open the project. Click the [Project Info] button to view the project information. You can also double click on a project in the list to open it. The list stores a maximum of 8 recent projects. Select the [Setup]-[Preferences]-[Workspace]-[Clear recent project history] checkbox, to delete the history when you exit Epson RC+ 8.0.
Show this at startup	This is a checkbox that allows you to set whether the Start Window opens during startup or not. You can also change this setting in [Setup]-[Preferences]-[Workspace]-[Show start screen].

6.10 [Project] Menu

In Epson RC+ 8.0-[Project], you can manage and build the projects.

6.10.1 [Project Wizard] Command (Project Menu)

The Project Wizard is a tool for creating new projects using step by step procedures. It is especially useful for new users.

You can create the following types of projects:

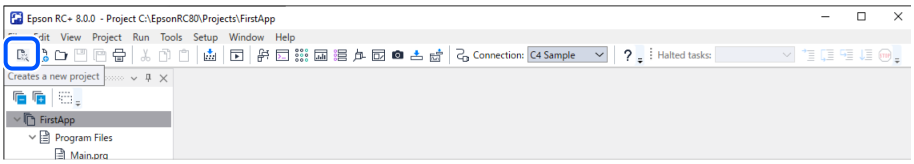
- Empty project
- Project from template
- Pick and place without vision
- Pick and place with vision

> ⚠ CAUTION
>
> Before creating a project that will use a robot, ensure that you are familiar with all safety precautions and procedures. Always use caution when working with the robot, especially when working inside the safeguarded area. See details below.
>
> **Precautions regarding Robot Operation**

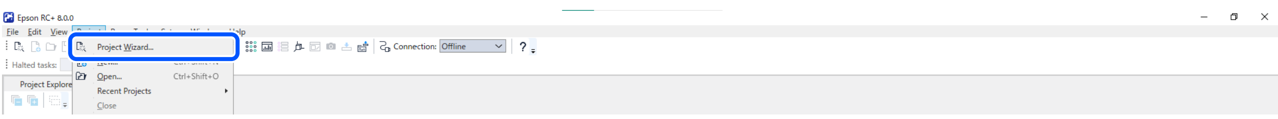
6.10.1.1 How to Use [Project Wizard] Command

Start project wizard by one of three ways.

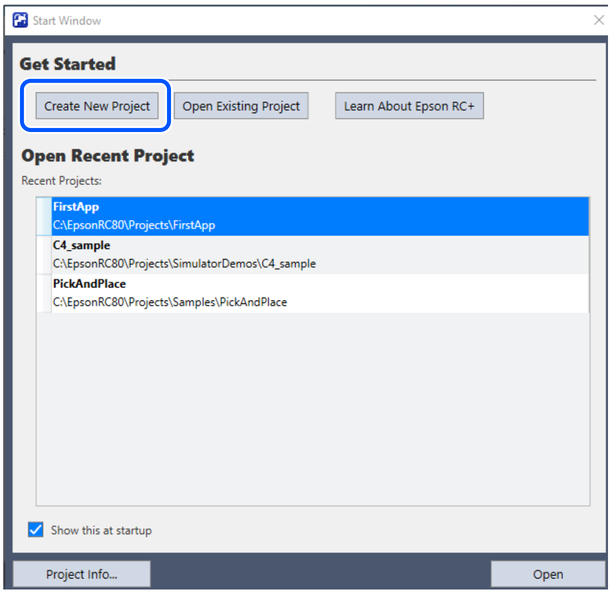
- Click the Epson RC+ toolbar - [Project Wizard]



- On the Epson RC+ 8.0 menu, select [Project]-[Project Wizard].

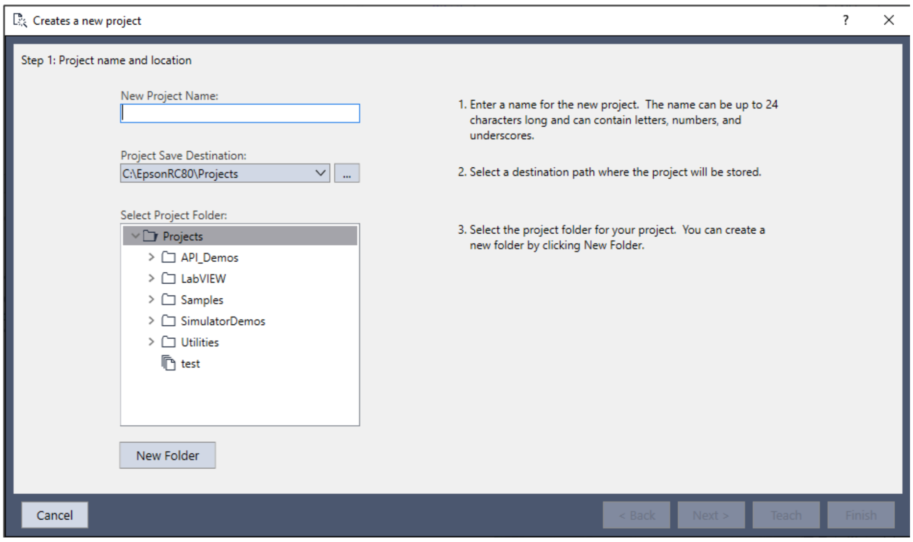


- Select [Create New Project] in [Start Window] that displayed when Epson RC+ 8.0 started.



1. Enter a name and select the save destination for the new project.

> ✎ TIP
>
> When specifying a new destination, click the [...] button to set the save destination. Set the destination in advance in the Epson RC+ menu, [Setup]-[Preferences]-[Workspace]-[Project Save Destination].

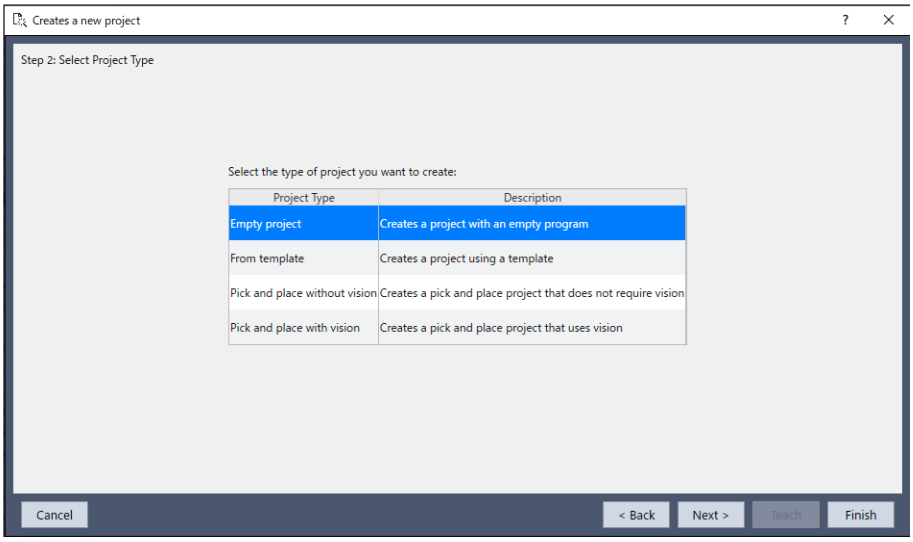


2. Select a project type you want to create, described in next page.
   - A: Empty project
   - B: From template
   - C: Pick and place without vision
   - D: Pick and place with vision

   Next steps vary according to the selected project type.

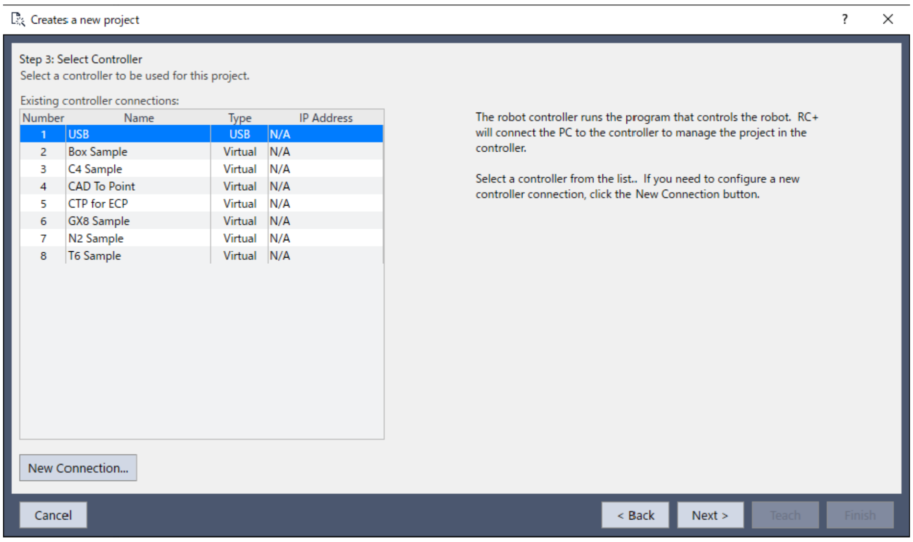
6.10.1.1.1 A: Empty project

4. When the following window is displayed, select [Empty project], and then click [Next].

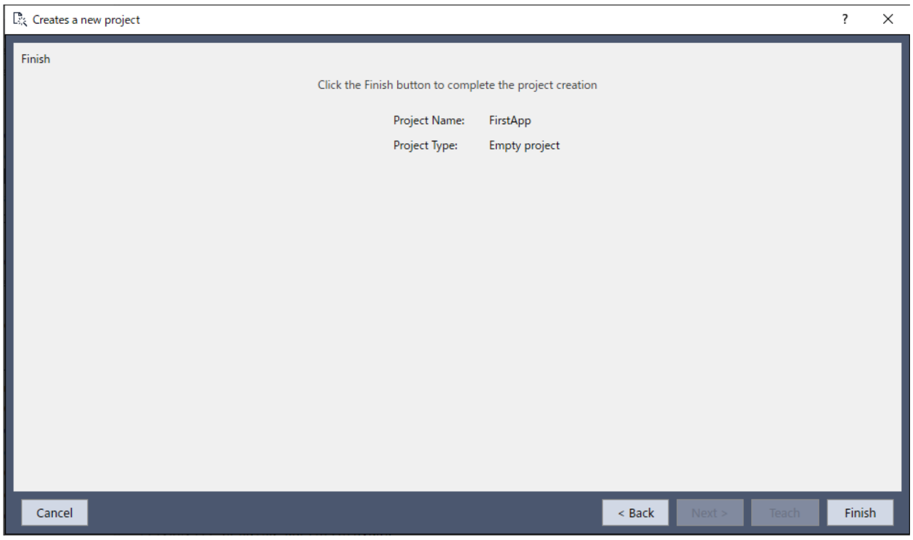


5. Select a Controller to connect. If it doesn't have connecting Controller in the list, click [New Connection] then Controller connecting wizard will start. See details below.

   Controller Connection

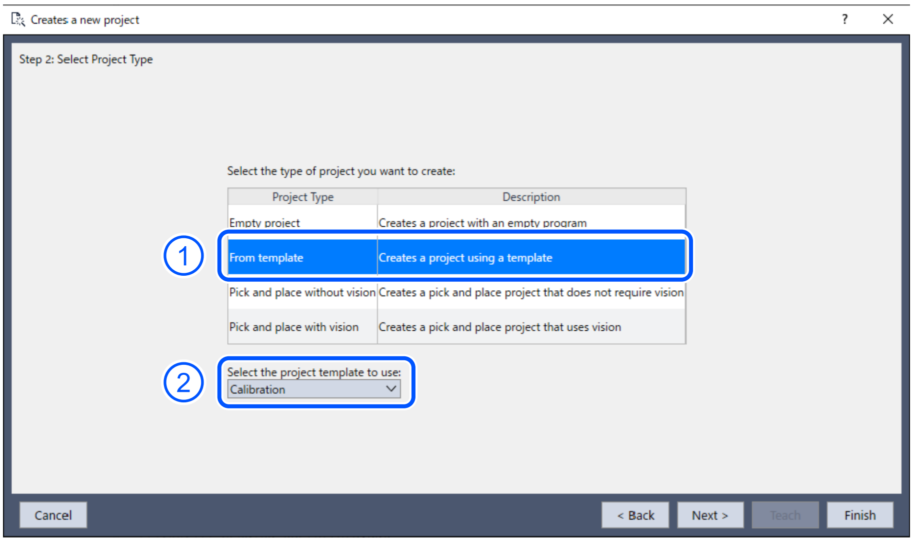


This is the end of the steps for the project wizard.



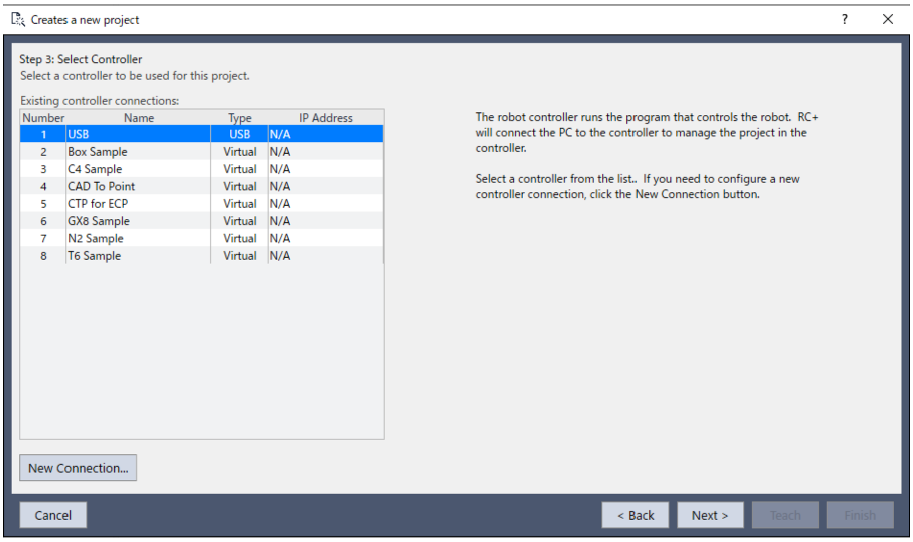
6.10.1.1.2 B: From template

4. When following window displayed, select [From template]. Then [Select the project template to use] will be displayed and select a template you want to create.

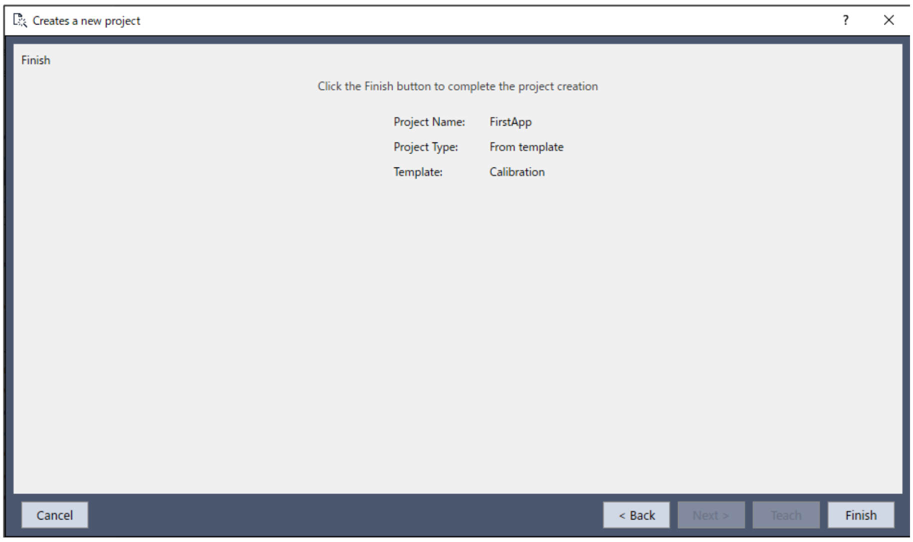


5. Select a Controller to connect. If it doesn't have connecting Controller in the list, click [New Connection] then Controller connecting wizard will start. See details below.

   Controller Connection



This is the end of the steps for the project wizard.

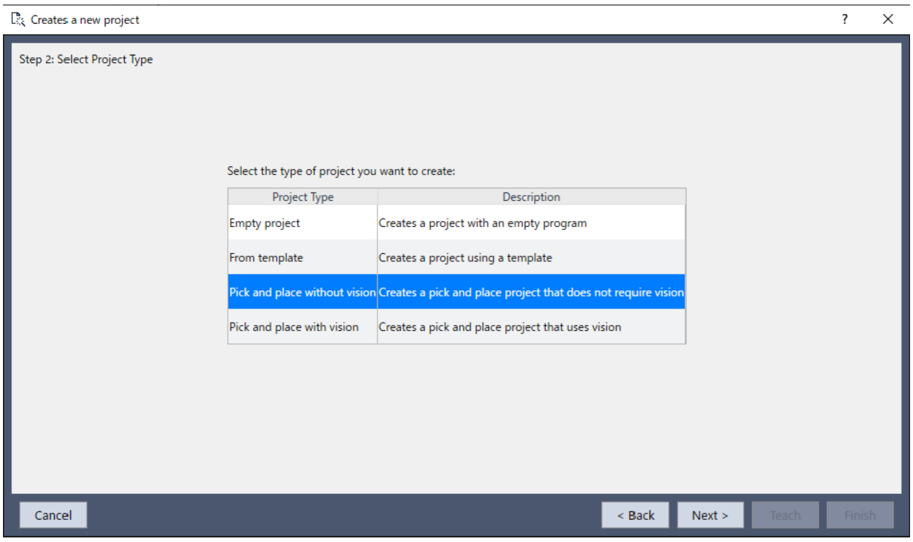


6.10.1.1.3 C: Pick and place without vision

Beforehand, following preparations need to be done.

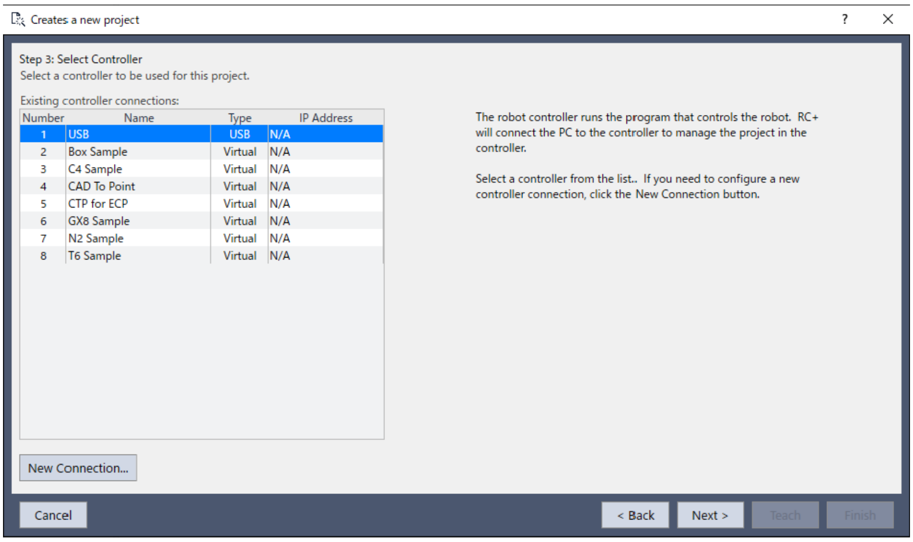
- Make sure that the connection settings of the robot controller are properly configured.
  - By default, Epson RC+ 8.0 will connect to the Controller with USB. Make sure that the PC on which RC+ is started and the robot controller are properly connected with a USB cable.
  - If you need a setting for Ethernet connection, you can set the connection from [Setup]-[PC to Controller Communications] Ethernet.
- Mount a hand for gripping a part. Determine the robot controller output(s) required to open and close the hand.

4. When the following window is displayed, select [Pick and place without vision], and then click [Next].

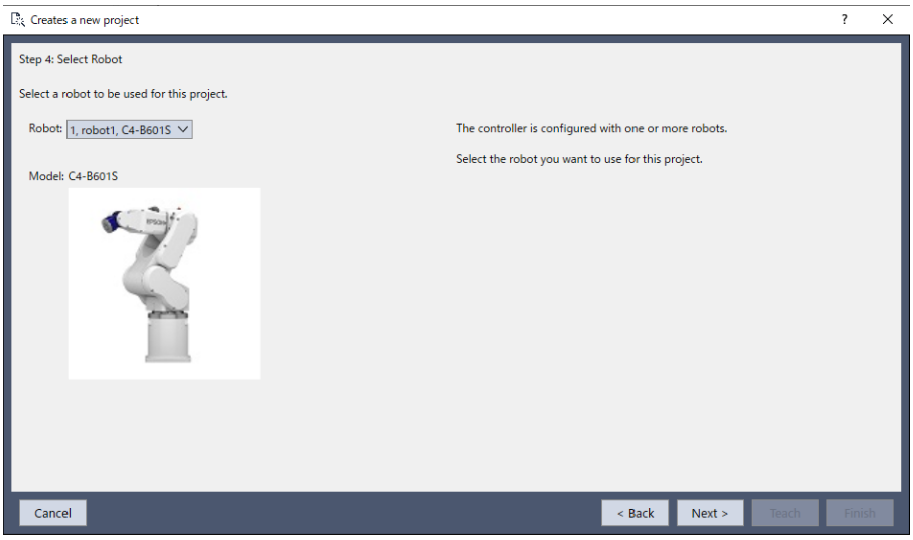


5. Select a Controller to connect. If it doesn't have connecting Controller in the list, click [New Connection] then Controller connecting wizard will start. See details below.

Controller Connection

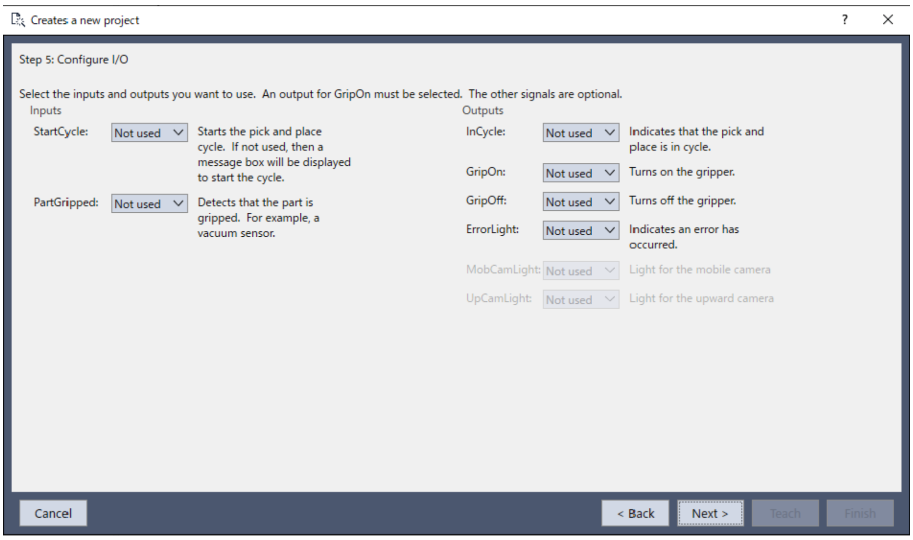


6. Select a robot to use.



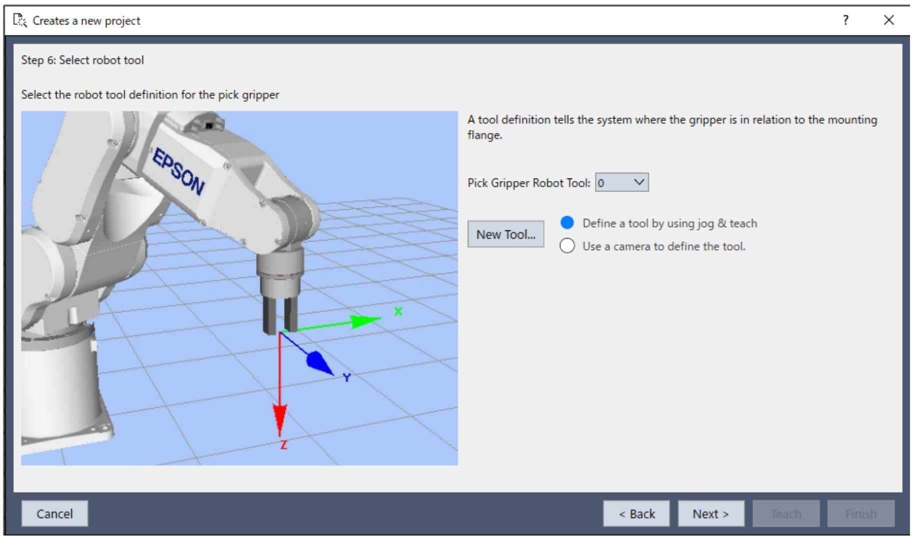
7. Set main I/O to use. Make sure to select output bit "GripOn". For more details of each Input/Output bit, see below.

Inputs and Outputs



8. Select a robot tool. By setting the tool according to the hand you're using, jog operation will be more intuitive. See details below.

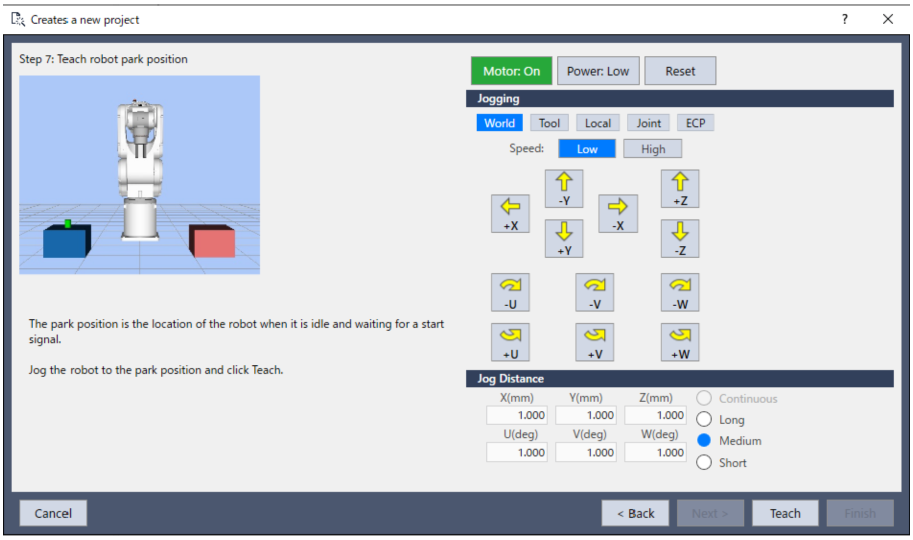
   Robot Tool



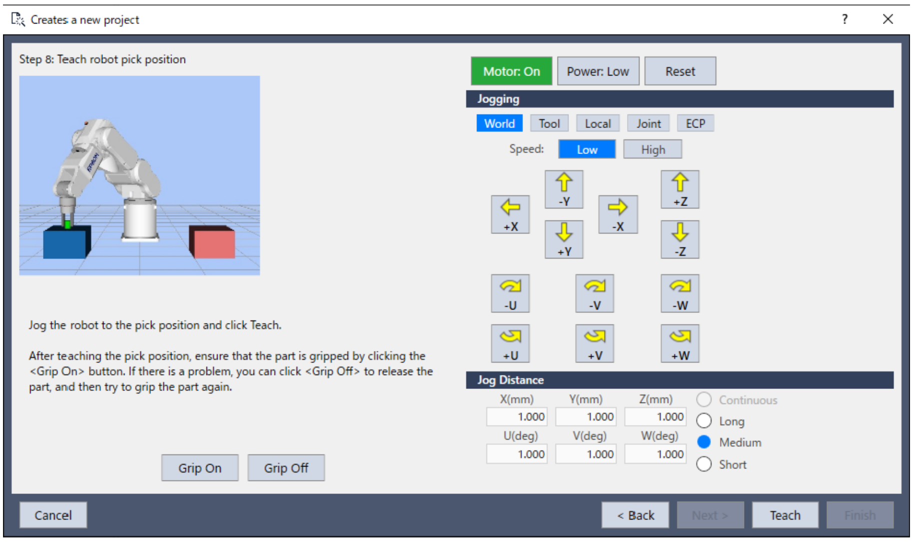
   From here, move the actual robot to teach the position. See details below.

   Jogging the robot and teaching points

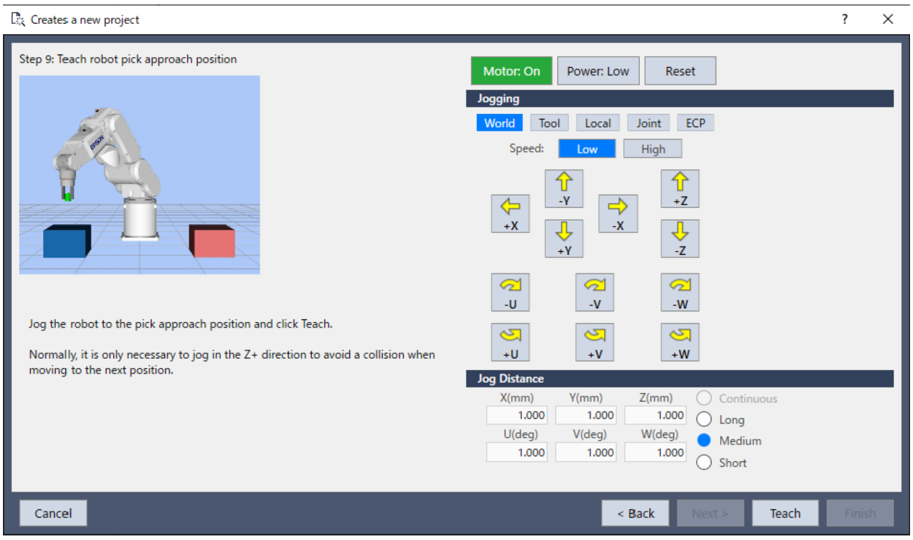
9. Jog the robot and teach the Park robot position. Park robot position is a point where a robot stops moving and waits for signals to start.



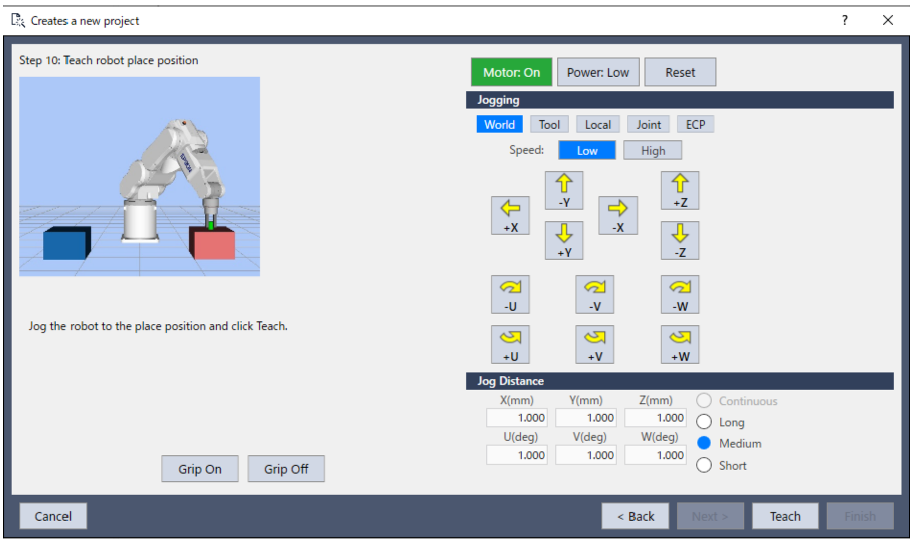
10. Jog the robot and teach the Pick robot position. Pick robot position is a point to hold (pick) a target workpiece.



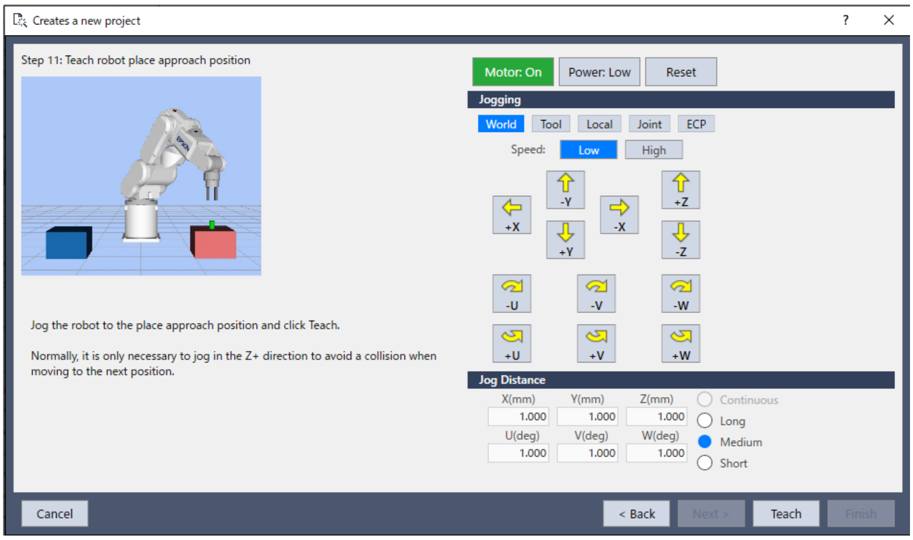
11. Jog the robot and teach the PickAppro (pick approach) robot position. It is recommended to specify the point for PickAppro robot position that is slightly moved in the +Z direction from the pick position to prevent collision with a jig near the pick position.



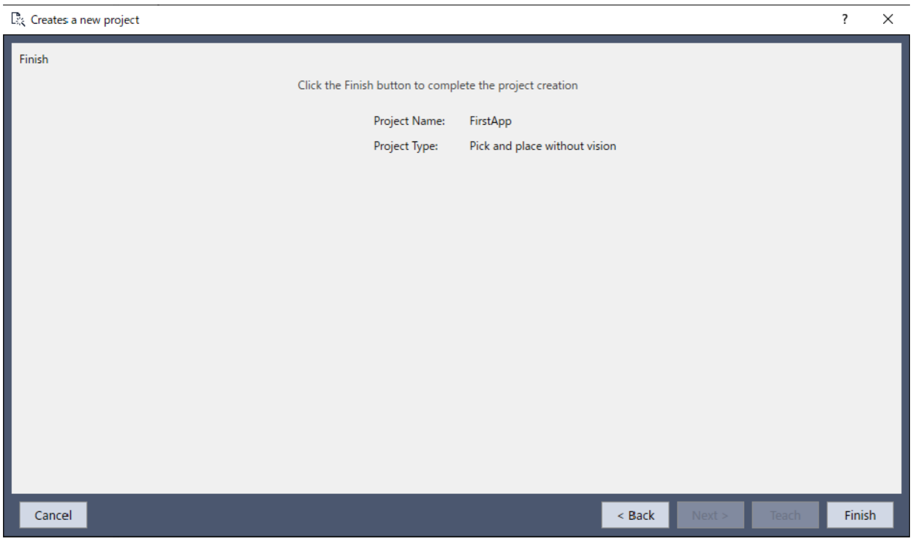
12. Jog the robot and teach the Place robot position. Place robot position is a point to release (place) the target workpiece.



13. Jog the robot and teach the PlaceAppro (place approach) robot position. Same as the PlaceAppro robot position, it is recommended to specify the point that is slightly moved in the +Z direction from the place position.



This is the end of the steps for the project wizard.



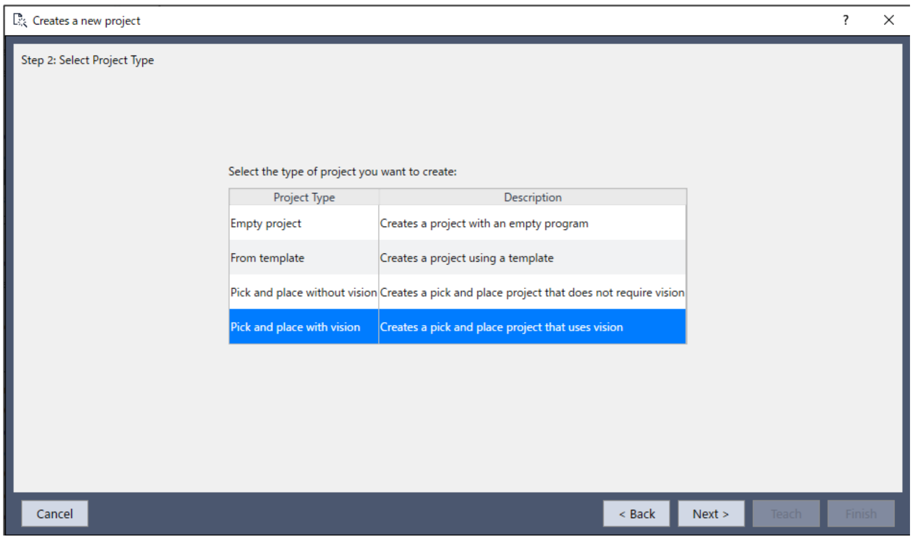
6.10.1.1.4 D: Pick and place with vision

Beforehand, following preparations need to be done.

- Make sure that the connection settings of the robot controller are properly configured.
  - By default, Epson RC+ 8.0 will connect to the Controller with USB. Make sure that the PC on which RC+ is started and the robot controller are properly connected with a USB cable.
  - If you need a setting for Ethernet connection, you can set the connection from [Setup]-[PC to Controller Communications] Ethernet.
- Mount a hand for gripping a part. Determine the robot controller output(s) required to open and close the hand. Also determine whether to use vision for positioning part to pick.
- If you need vision to locate the part for pickup, and if it is necessary, mount a camera on the robot.
  - For SCARA robots, mount the camera on joint 2 or joint 4.
  - For 6 axis robots, mount the camera on joint 5 or joint 6.
  - You may also need to mount a light for the camera. Determine a robot controller output required to operate the light.

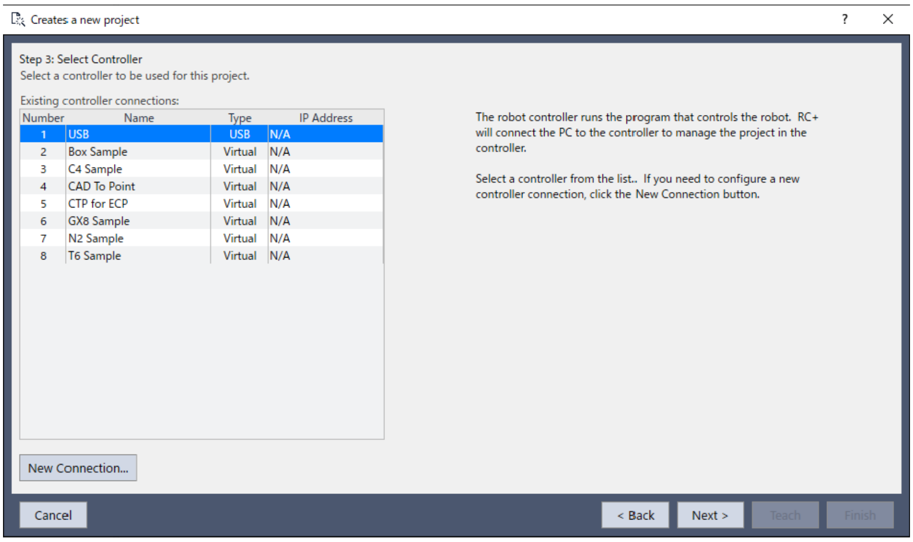
- Determine if you need vision to locate the part on the gripper to calculate place position.
  - This is accomplished with an upward looking camera. After the part has been picked up, the robot moves so that the grasped part is over the upward looking camera. The vision system then determines the place position of the part.
  - You may also need to mount a light for the camera. Determine a robot controller output required to operate the light.

4. When the following window is displayed, select [Pick and place with vision], and then click [Next].

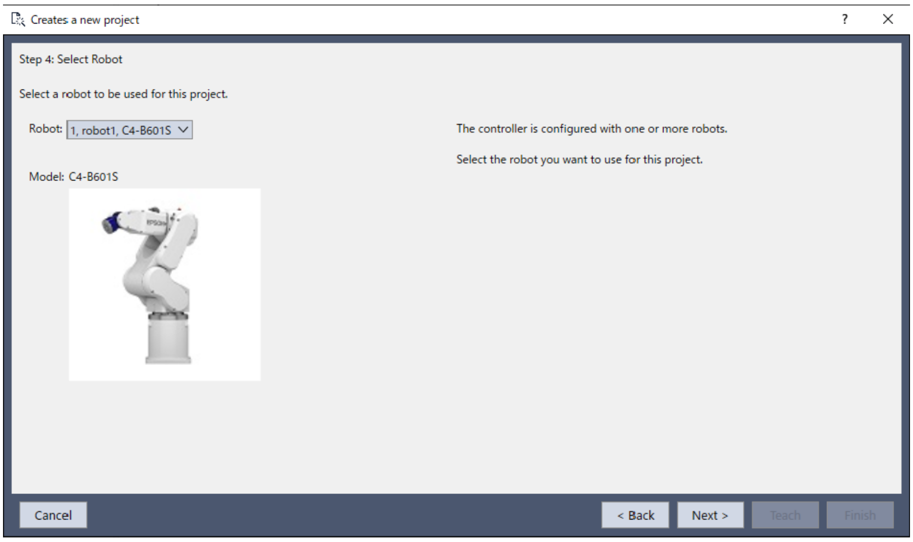


5. Select a Controller to connect. If it doesn't have connecting Controller in the list, click [New Connection] then Controller connecting wizard will start. See details below.

   [Controller Connection](#)

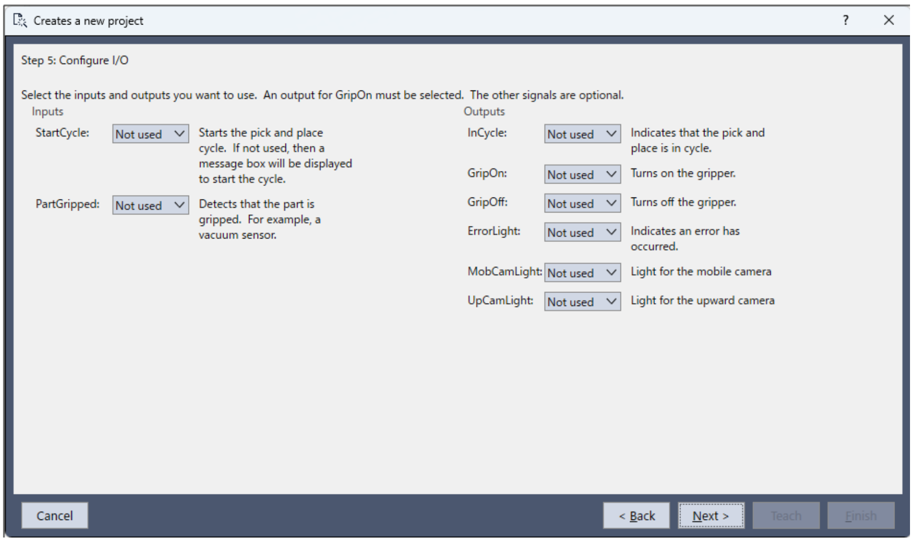


6. Select a robot to use.



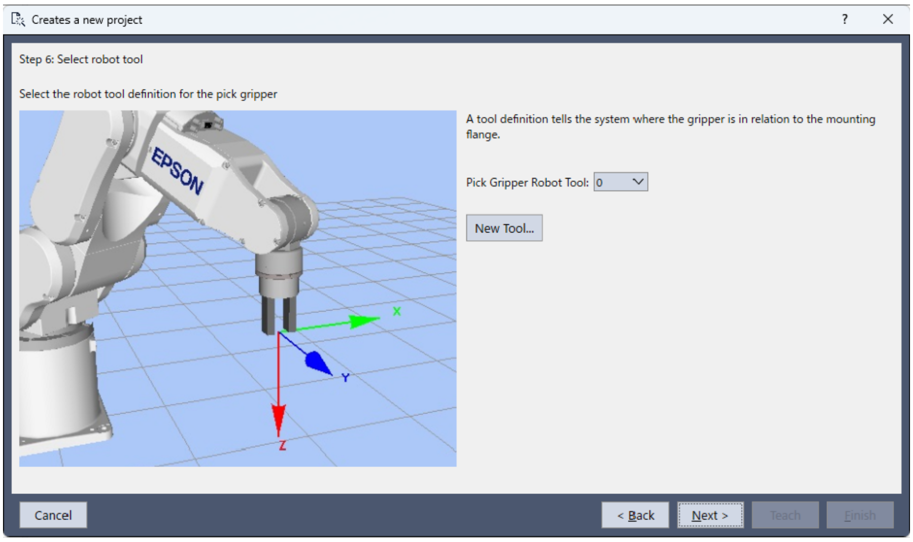
7. Set main I/O to use. Make sure to select output bit "GripOn". For more details of each Input/Output bit, see below.

   Inputs and Outputs



8. Select a robot tool. By setting the tool according to the hand you're using, jog operation will be more intuitive. See details below.

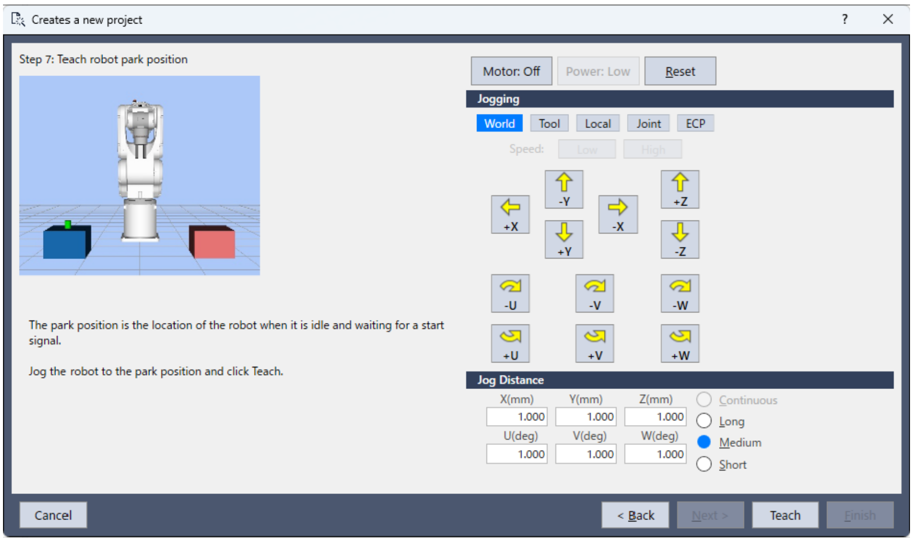
   Robot Tool



From here, move the actual robot to teach the position. See details below.

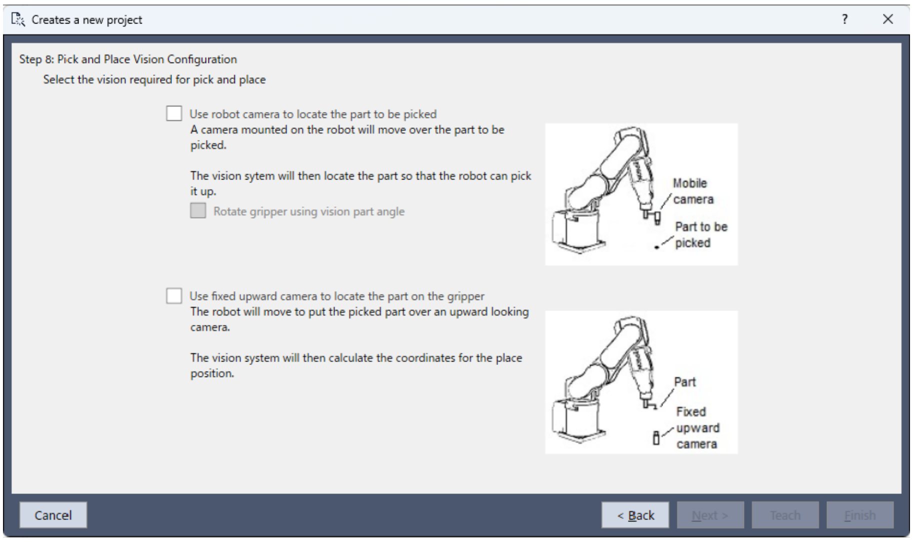
[Jogging the robot and teaching points]

9. Jog the robot and teach the Park robot position. Park robot position is a point where a robot stops moving and waits for signals to start.

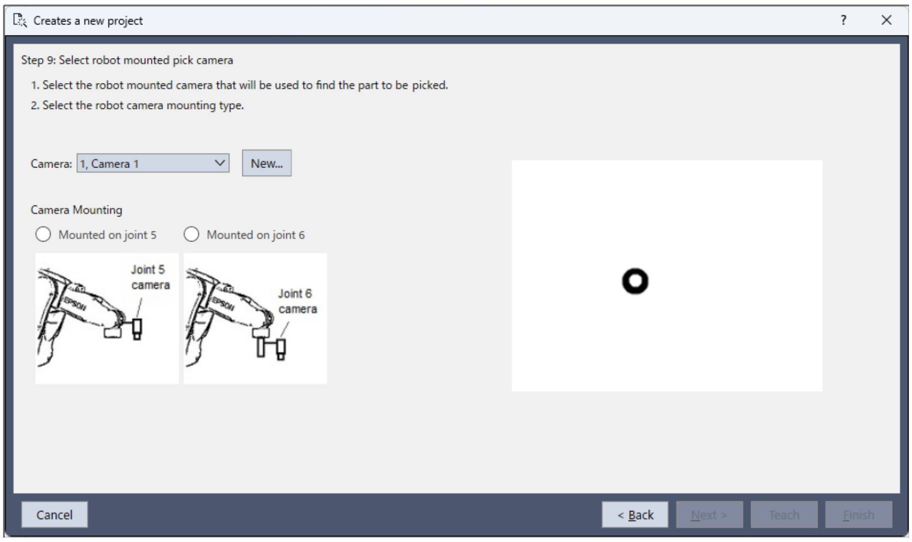


10. Select a position of mounting a camera.

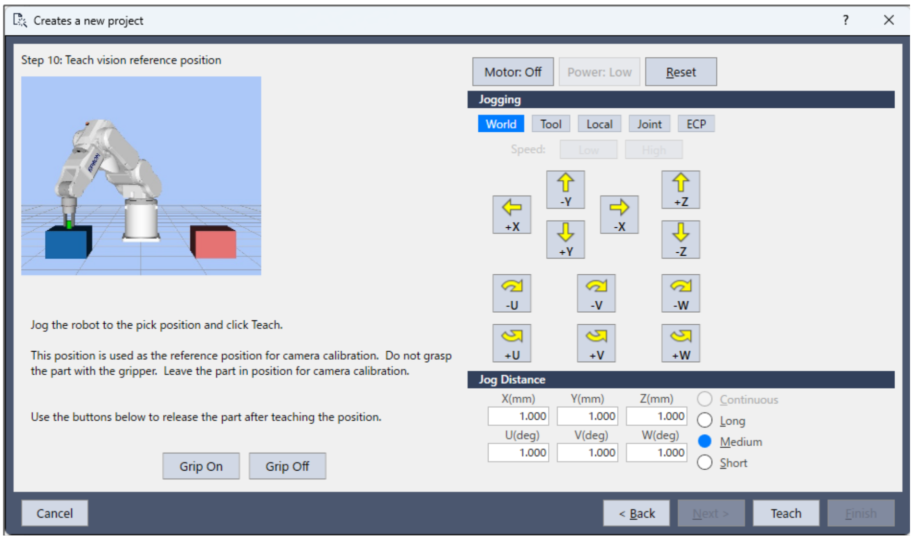
You can configure a mobile camera and/or an upward camera. Check [Use robot camera to locate the part to be picked] if you need to locate the part before picking it. Optionally, you can check [Rotate gripper using vision part angle]. This will cause the robot to rotate the gripper to match the angle of the part. Select the [Use fixed upward camera to locate the part on the gripper] checkbox if you need the robot to adjust for pickup variations when placing the part.



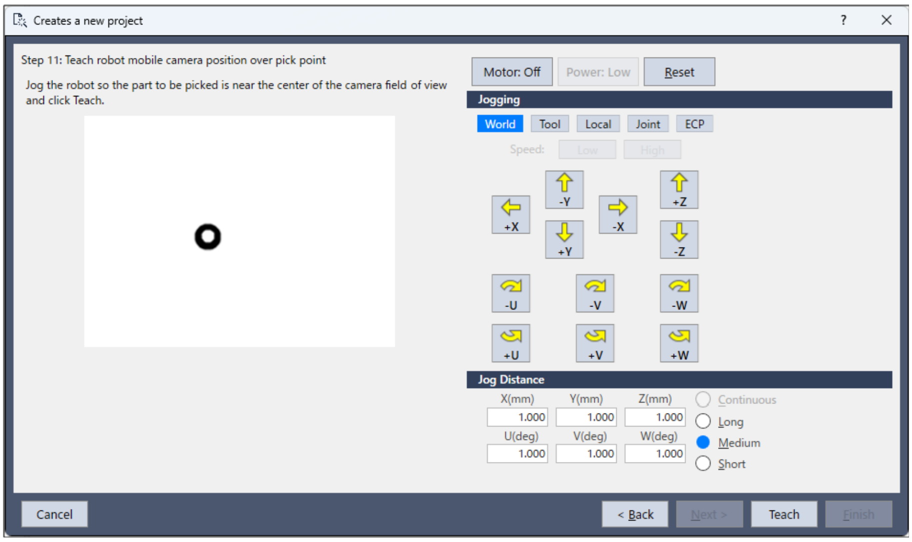
11. When mobile camera is selected in step 10, the wizard moves to the wizard of selecting mobile camera. If you are not using a mobile camera to find the part to pick up, go to step 14.



12. Jog the robot and teach the Pick robot position.

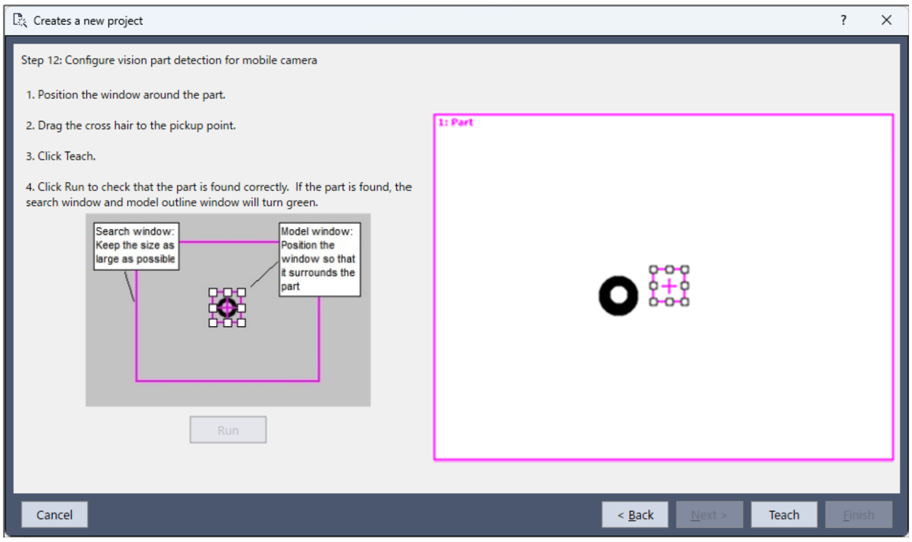


13. Jog the robot until the camera is positioned so that the part is near the center of the field of view.



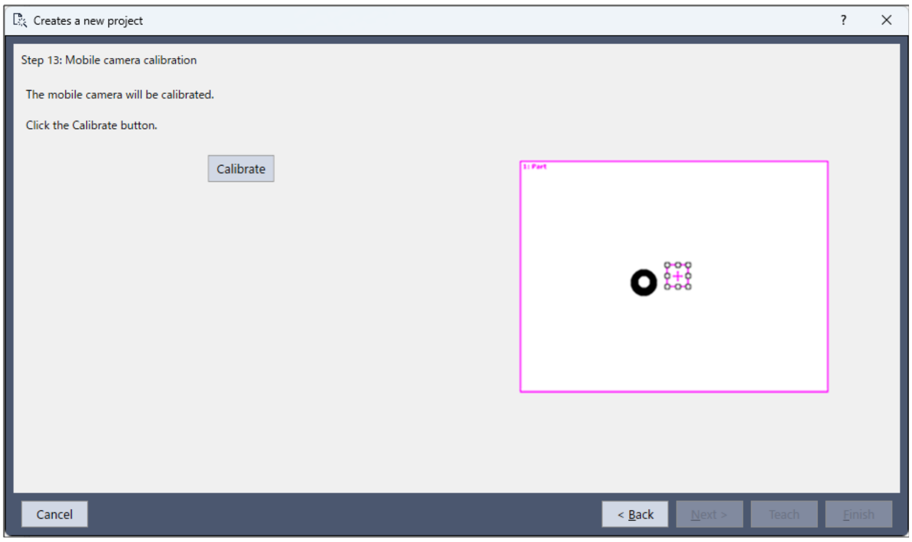
14. Teach the vision system for part detection. Use the mouse to position the model window so that is surrounds the part. Position the cross hair to the pick position on the part. See details below.

Teaching part detection for vision

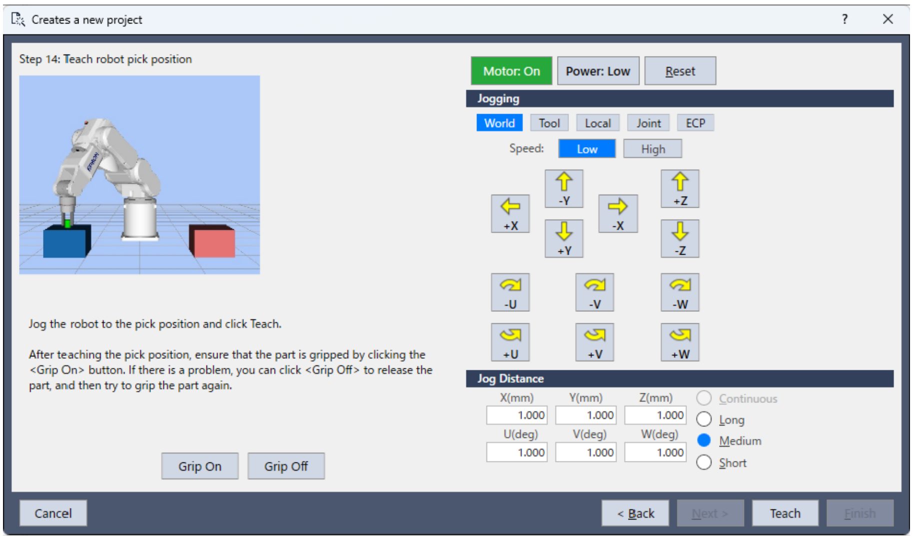


15. Calibrate the mobile camera by clicking the [Calibrate] button.

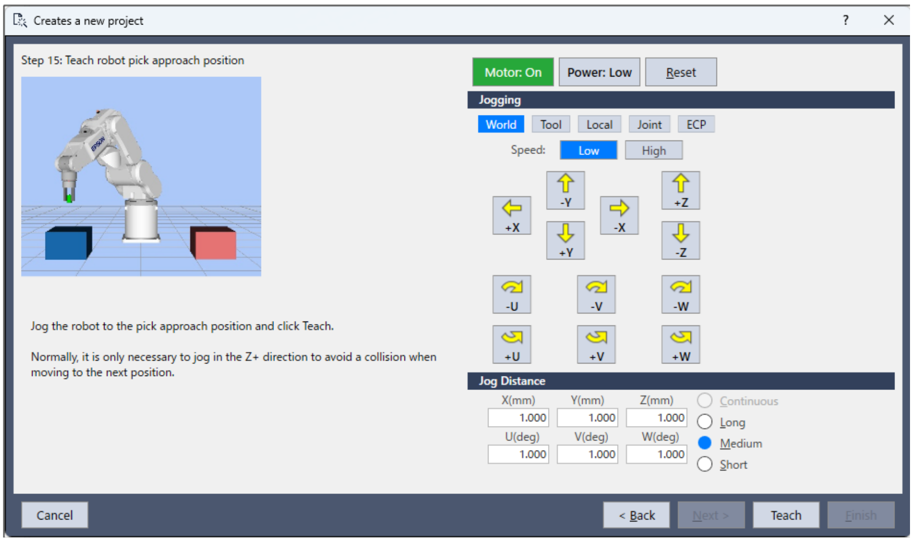
Calibration is automatic and will use the part as the calibration target. Click the [Abort] button if you want to stop the calibration. When the calibration is completed, a message is displayed stating whether the calibration was successful or not.



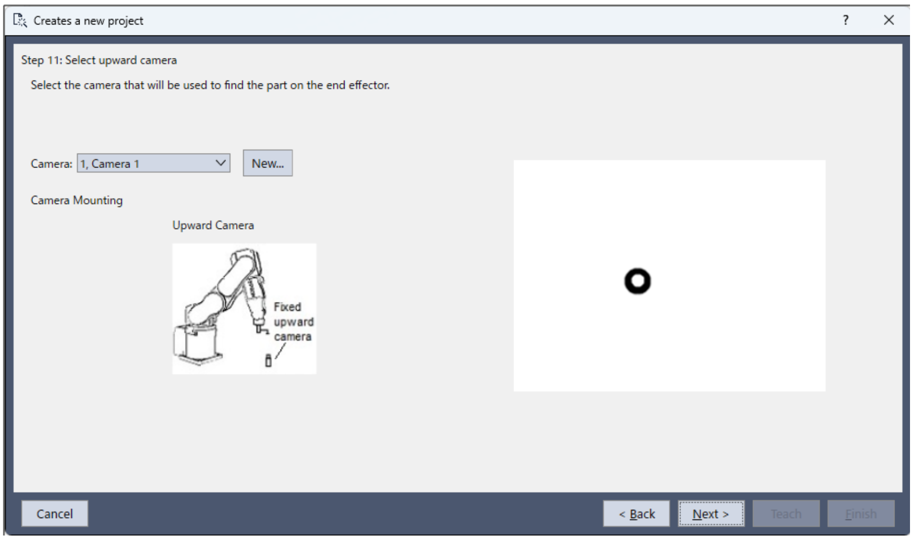
16. Jog the robot and teach the Pick robot position. Pick robot position is a point to hold (pick) a target workpiece.



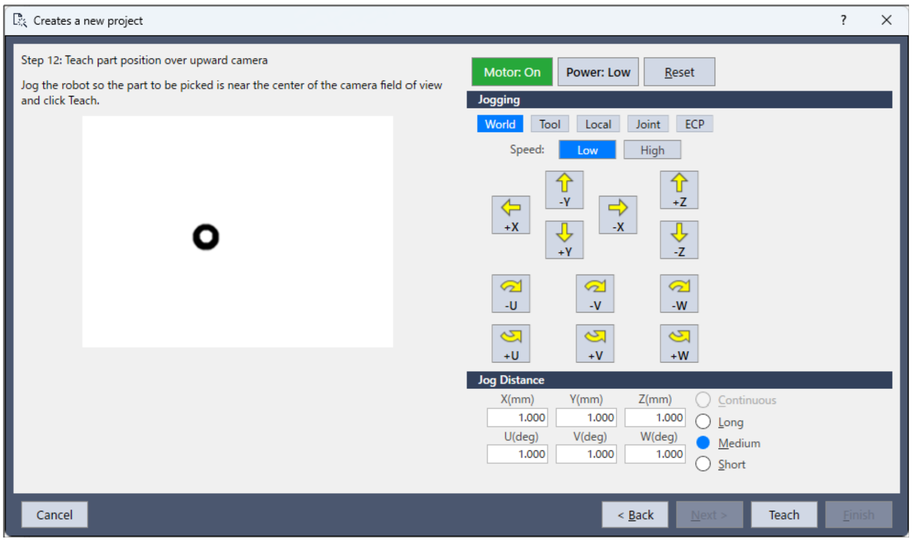
17. Jog the robot and teach the PickAppro (pick approach) robot position. It is recommended to specify the point for PickAppro robot position that is slightly moved in the +Z direction from the pick position to prevent collision with a jig near the pick position.



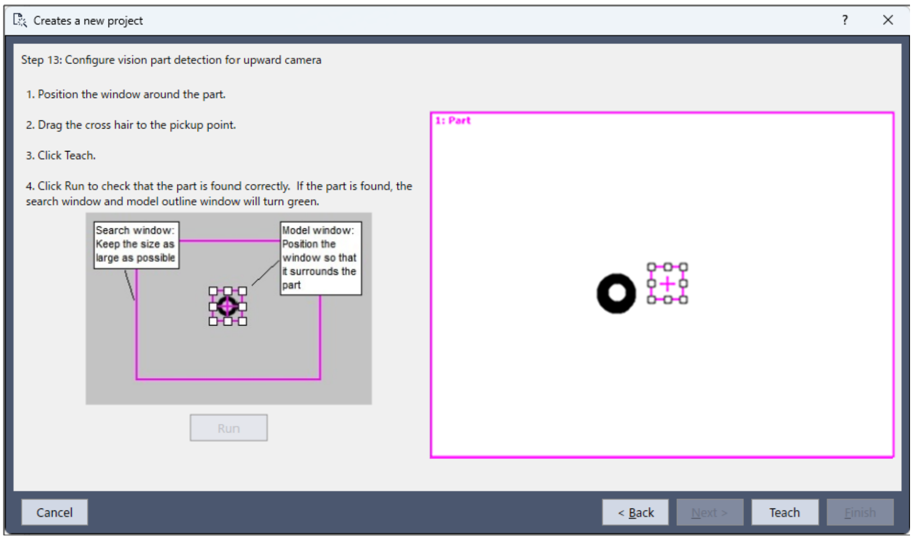
18. When upward camera is selected in the step of selecting a position of mounting a camera, the wizard moves to the wizard of selecting upward camera.



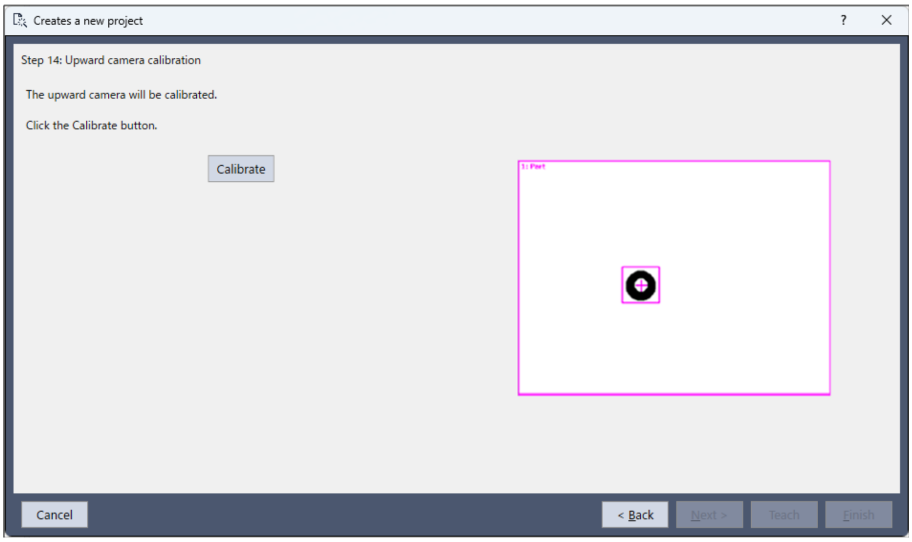
19. Jog the robot until the camera is positioned so that the part is near the center of the field of view.



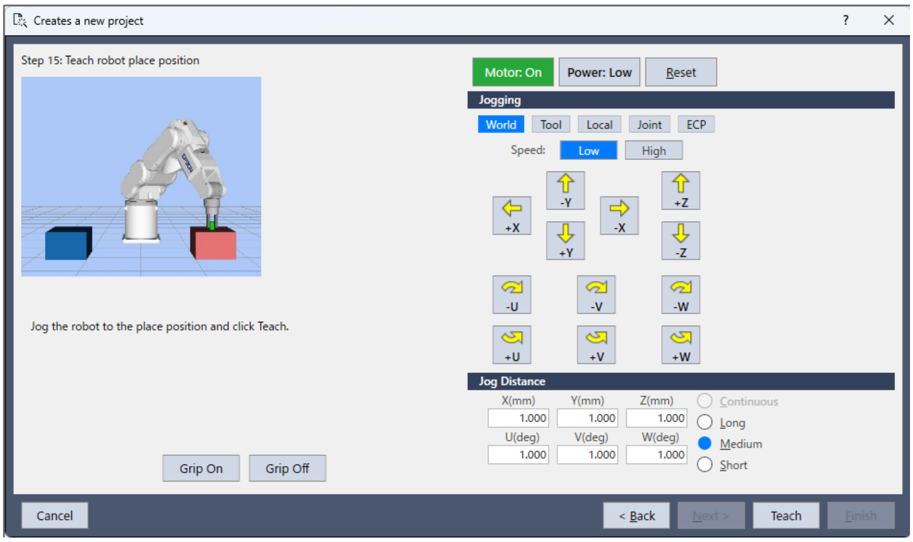
20. Register the vision model of the part.



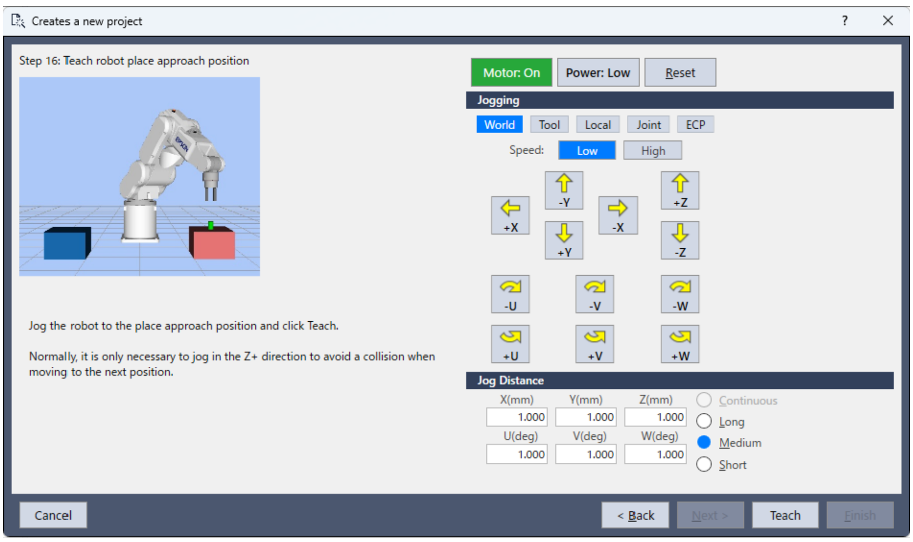
21. Calibrate the upward camera.



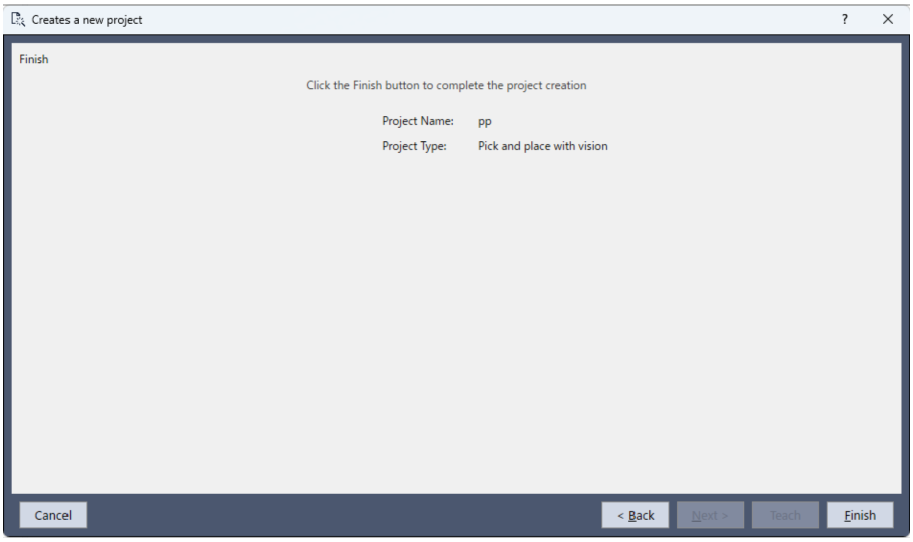
22. Jog the robot and teach the Place robot position. Place robot position is a point to release (place) the target workpiece.



23. Jog the robot and teach the PlaceAppro (place approach) robot position. Same as the PlaceAppro robot position, it is recommended to specify the point that is slightly moved in the +Z direction from the place position.



This is the end of the steps for the project wizard.



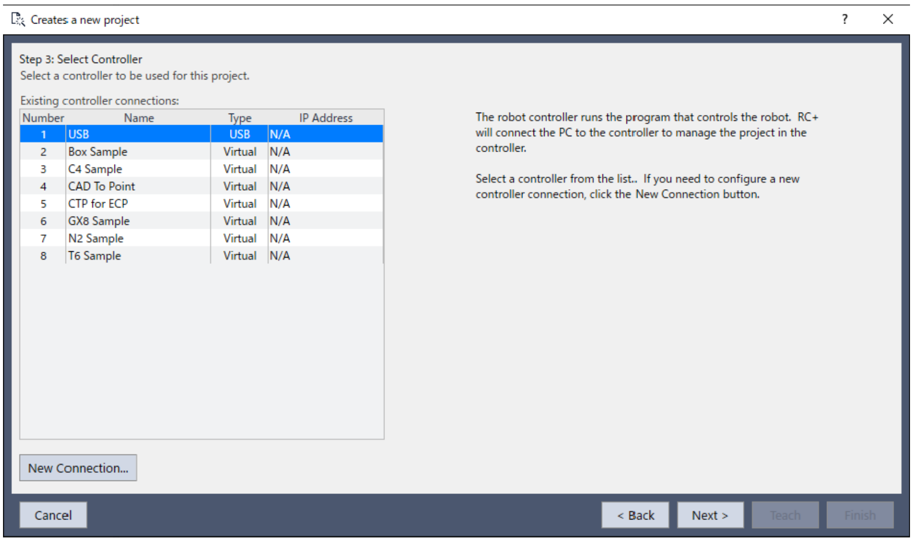
6.10.1.1.5 After Project Wizard

After the wizard finishes, the [Finish] button will be enabled. Click the [Finish] button and create a new project.

1. After clicking [Finish], SPEL+ program code and related objects are automatically generated for a new project.
2. Press F5 to run the new program. Use caution when running a program that moves the robot.
3. You can make changes to the generated program to refine operation to suit customer's needs.

6.10.1.2 Controller Connection

For each project type, you must select a connection to a robot controller when the Project Wizard asks you to select a controller. You can select from existing connections, or you can create a new connection.

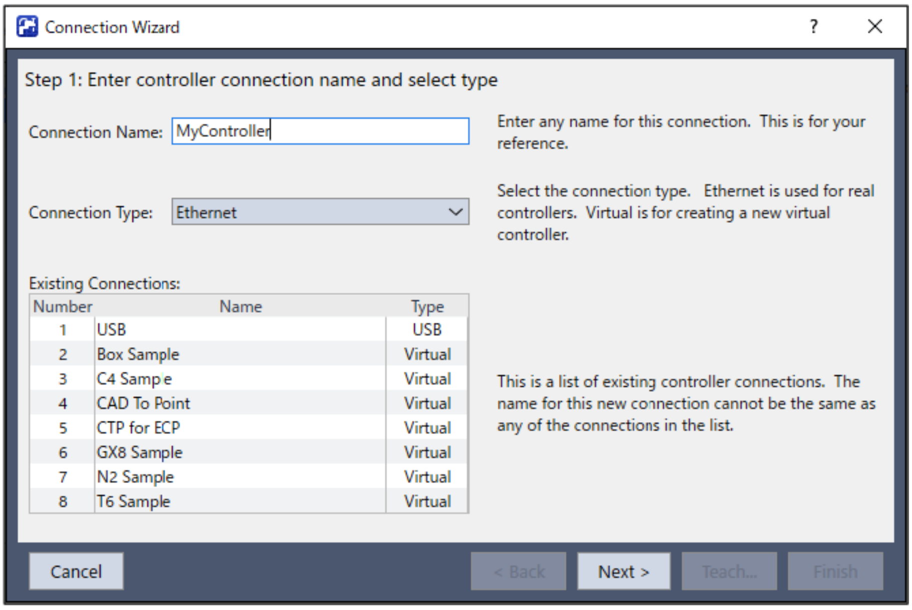


Select an existing connection

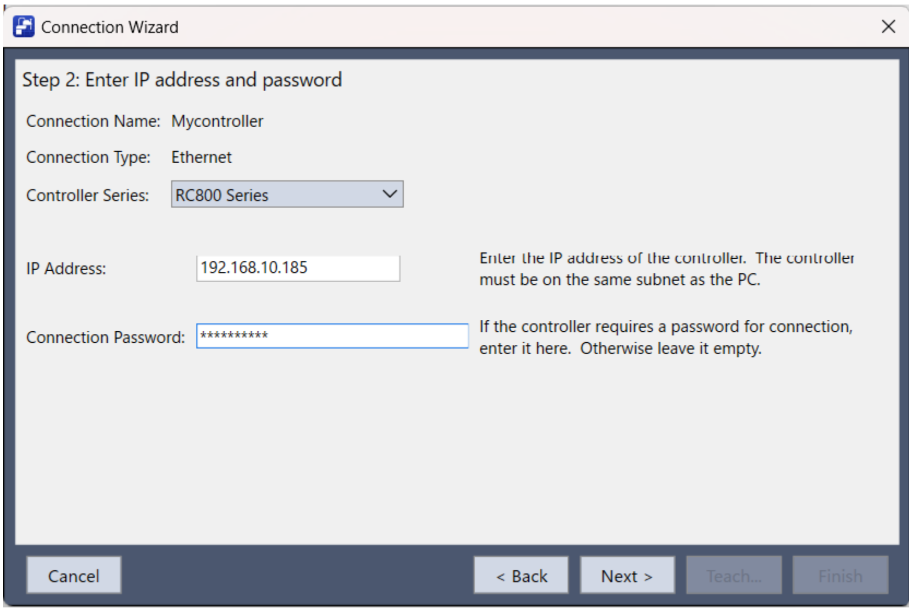
Select a connection in the list of connections, and then click the [Next] button.

Create a new connection

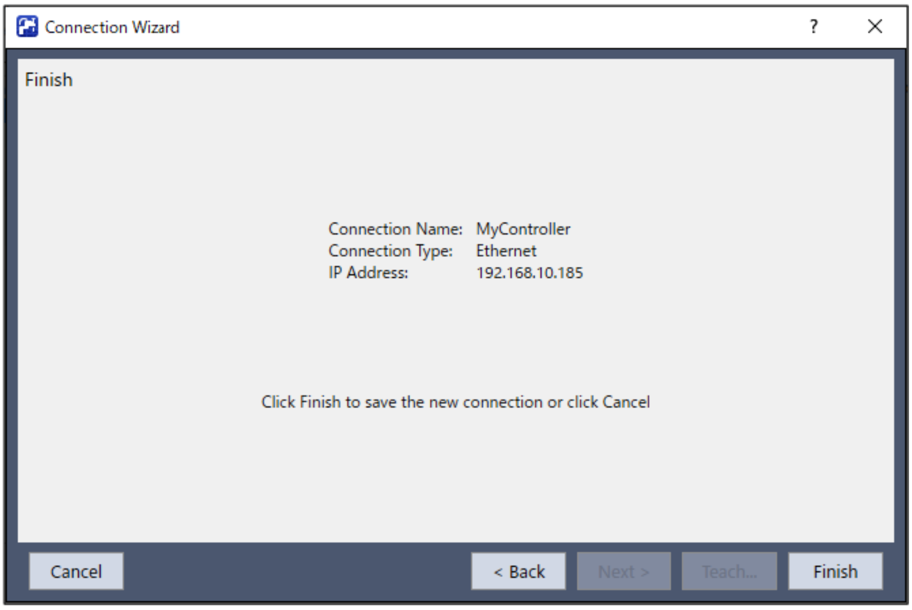
Click the [New Connection...] button. This opens the Connection Wizard. Follow the steps in the wizard to add a new controller connection. When the wizard is completed, the new connection will be selected in the connection list. Click the [Next] button.



When the connection type is Ethernet, enter the IP address. If the controller connection requires a password, enter the connection password. Click the [Next] button. Epson RC+ will attempt to connect with the controller. If unsuccessful, an error message will be displayed. Check the IP address, password, connection and network cables. Click the [Next] button again to start the connection.

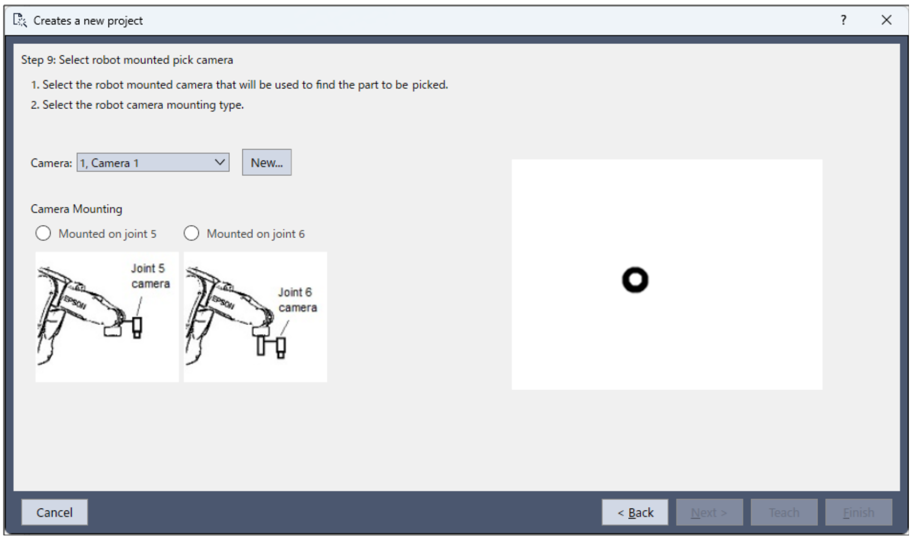


Click the [Finish] button to create a new connection. The new connection will be added to the connection list in the project wizard and it will be selected.



6.10.1.3 Camera Selection

For project types that require vision, you must select a camera where required. You can select from a dropdown list of existing cameras, or you can add a new camera.

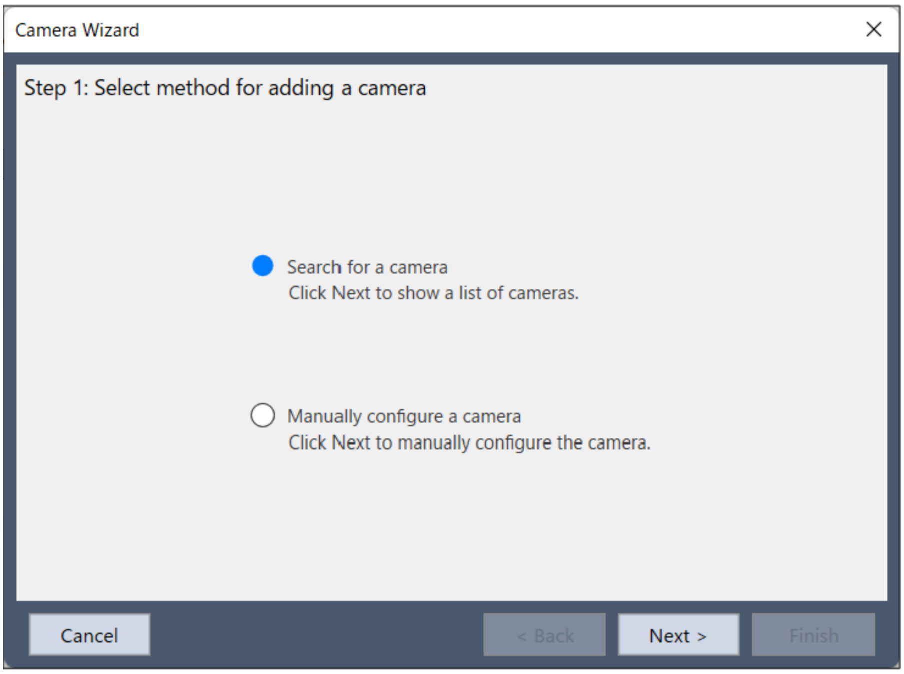


Select an existing camera

Select a camera from the dropdown list, then click the [Next] button.

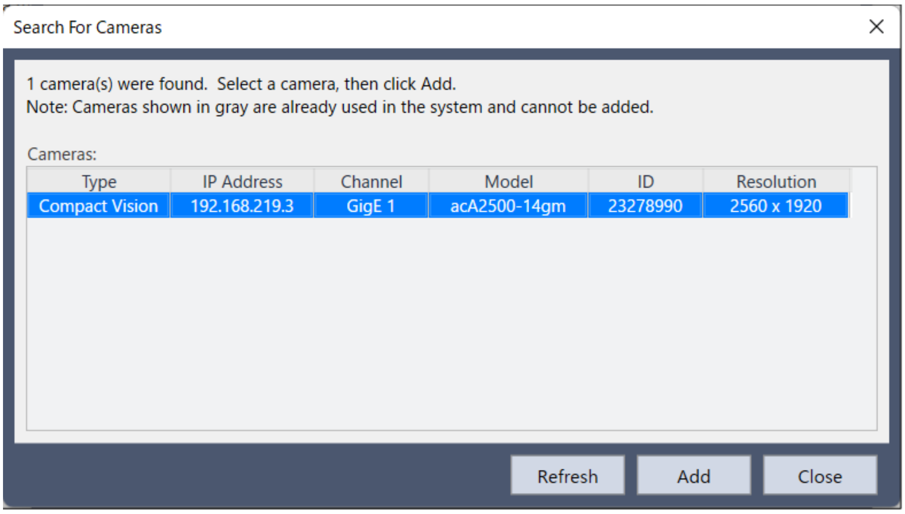
Add a new camera

Click the [New...] button. This opens the Camera Wizard. Follow the steps in the wizard to add a new camera.

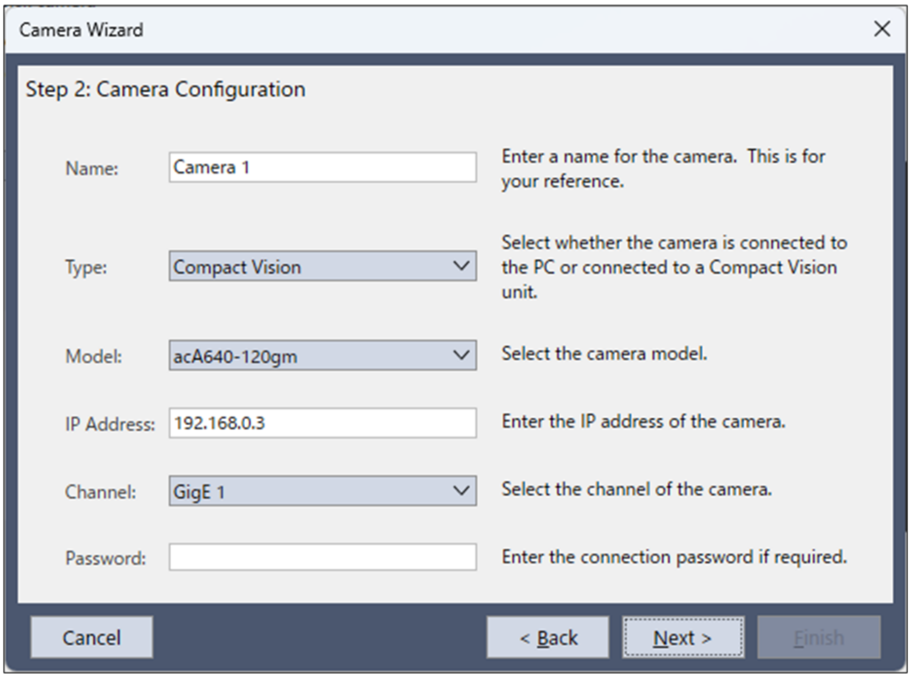


Select whether to automatically search for a camera in [Search for a camera], or [Manually configure a camera], and click the [Next] button.

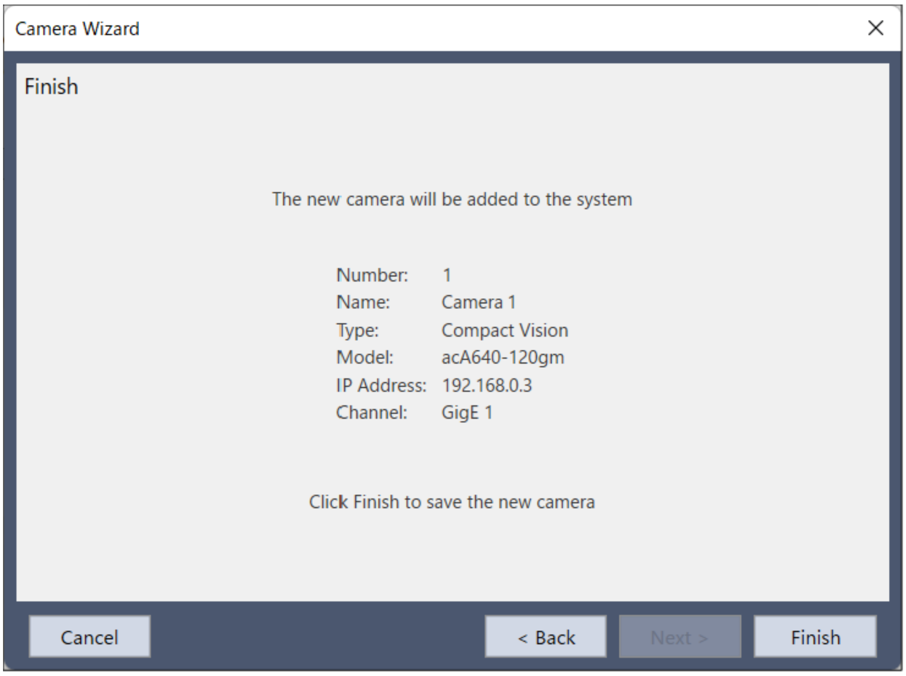
If you selected to [Search for a camera], a list of available cameras is displayed. Select a camera in the list, and then click [Add].



The camera configuration information is populated by the selected camera. Review or enter the information for the new camera, and then click the [Next] button.



Review the camera configuration and click the [Finish] button to add the new camera.



The new camera will be selected in the camera list.

6.10.1.4 Inputs and Outputs

For pick and place project types, you will be prompted by the wizard to configure input and output signals. Only the GripOn output is required. All other I/O signals are optional. I/O labels for the new project are automatically created. The signals that are defined are used in the generated SPEL+ program.

KEY POINTS

Signals that have been reserved for use with remote I/O are not allowed.

Signal Name	Type	Description
StartCycle	Input	Starts a pick and place cycle. If not defined, a message box is added to the generated code that prompts the operator to start a cycle.
PartGripped	Input	Used to detect if the part is grasped or not. For example, if a vacuum gripper is used, this signal could be a vacuum sensor that detects that a part was picked up or released. If this signal is not specified, then a predefined delay is added to the code after turning the part gripper on and off.
InCycle	Output	Indicates that a pick and place cycle is running.
GripOn	Output	Turns on the gripper to grip the part.
GripOff	Output	If specified, this turns off the hand to release the part. The hands with two points requires to specify. If not specified, the part is released when GripOn is Off.
ErrorLight	Output	Indicates that an error occurred.
MobCamLight	Output	Controls the light mounted with the mobile camera.
UpCamLight	Output	Controls the light mounted with the upward camera.

6.10.1.5 Robot Points

For pick and place project types, you will be prompted by the wizard to teach robot points, shown in the table below. Point labels for the new project are automatically created and used in the generated SPEL+ program.

Robot point	Description
Park	This is the idle position for the robot when waiting for the next cycle to start
Pick	This is the part pick position.
PickCam	For vision pick and place. This is the position of the robot mobile camera over the part to be picked.
PickAppro	Used for 6 axis robots. This is the approach position above the pick position.
UpCam	For vision pick and place. This is the position of the grasped part over the upward looking camera.
Place	This is the part place position.
PlaceAppro	Used for 6 axis robots. This is the approach position above the place position.

6.10.1.6 Jogging the robot and teaching points

The project wizard supports integrated simple jogging when it is required to teach a robot point.

Select the jog mode, jog speed, and jog distance from the dropdown lists.

KEY POINTS

For safety, continuous jogging is not used in the project wizard.

Click a jog button to move the robot in the direction shown on the button. If you want to jog multiple steps, hold down the jog button.

When you click a jog button and some condition prevents jogging, such as motors off, EStop active, Safeguard active, etc., you will be prompted for motors to be turned on or to clear Estop or Safeguard. Then you can click the jog button again to jog the robot.

After jogging to the required position, click the [Teach] button to teach the robot point. If necessary, you can jog the robot again, then click [Teach] again to re-teach the point.

6.10.1.7 Robot Tool

For pick and place project types, you will be prompted to select a robot tool. A tool informs the system of the location of the end effector.

For a simple application that uses vacuum end effector and does not require rotation of a part for placement, you may not need to use a robot tool and can use Tool 0 (default).

For an application that uses an end effector that is not centered on the robot mounting flange and needs to rotate the part for pickup and / or for placement, you should define a robot tool. When you select a tool in the wizard, you can click on [New Tool...] to use the Tool Wizard.

You can also define the tool before using the project wizard by running the Tool Wizard from the Robot Manager Tool page.

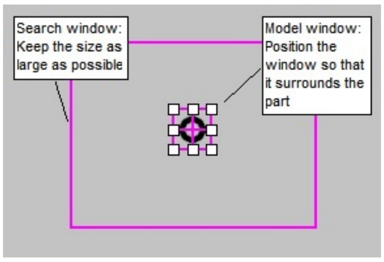
For more details on tool settings, refer to the following manual:

- **[Tools]-[Robot Manager]-[Tools] Page**
- "SPEL+ Language Reference - TLSet Statement"

6.10.1.8 Teaching part detection for vision

When creating a project that requires vision part detection, you will be asked to perform the following:

1. Outline the part using a mouse to change the size and position of the model window.
2. Use the mouse to drag the model reference cross-hair. For a mobile camera to find the part to be picked up, drag the cross-hair to the desired pick position on the part. For an upward camera used to find the part on the end effector, drag the cross-hair so that it is near the center of the part.
3. Optionally, you can click on the search window and change its size. Try to keep the size as large as possible. The part should be able to be found anywhere in the search window.



4. Click the [Teach] button to teach the vision model of the part.
5. Click the [Run] button to check how the part is found by the vision system. You can change the model window and re-teach the model again if desired.

6.10.1.9 Hand Settings

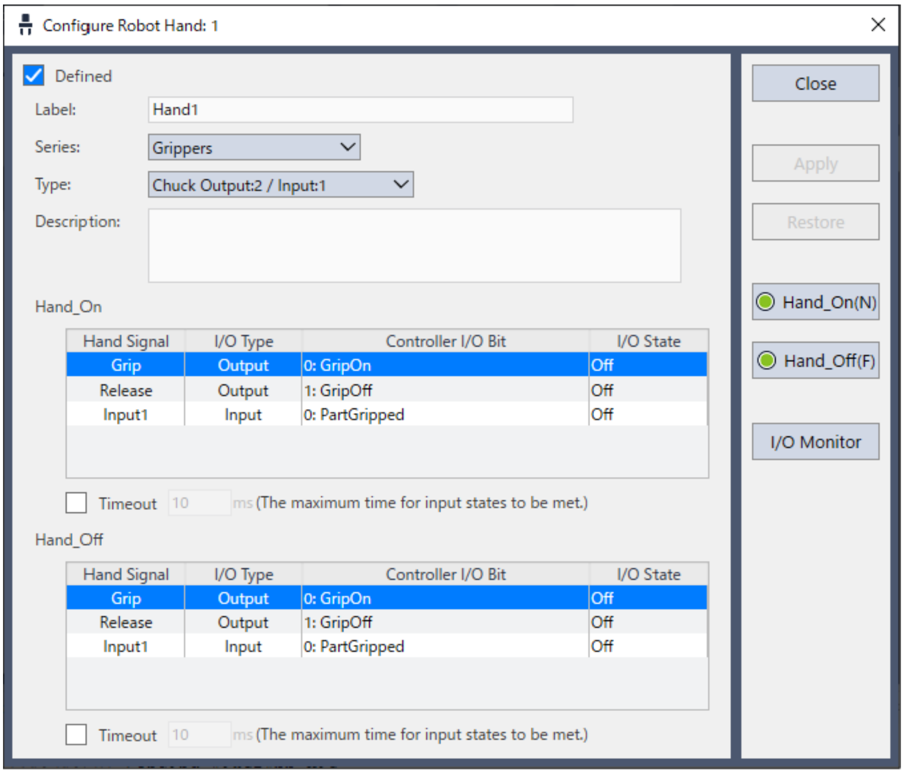
In the created SPEL+ program, commands related to hand On/Off can also be written using hand control commands. By using the hand function, you can easily check the motion including the operation of the hand from the GUI.

This section describes how to register a hand and how to replace it with a hand control command. For more details on hand settings, refer to the following manual:

"Hand Function Manual"

Example of hand registration

Hand registration is performed on the [Configure Robot Hand: *] screen. The figure below is an example when a chuck with 2 outputs and 1 input is set.



Follow the steps below to register a hand on the [Configure Robot Hand: *] screen.

Depending on the hand, the I/O On/Off setting may be the opposite of the value described in the steps.

1. Select the Epson RC+ menu- [Tools]-[Robot Manager]-[Hands] tab.
2. Select any hand number.
3. Click the [Configure...] button. The [Configure Robot Hand *] screen is displayed.
4. Check the [Defined] check box.
5. Set [Series:] to “Grippers”.
6. Refer to the table below and set the type according to the usage of the output bit GripOff and the input bit PartGripped.

	With PartGripped	Without PartGripped
With GripOff	Chuck* Output2 / Input1	Chuck* Output2 / Input0
Without GripOff	Chuck* Output1 / Input1	Chuck* Output1 / Input0

* For suction hands, use Suction instead of Chuck.

7. Select [Hand_On]-[Grip] to set [Controller I/O Bit] to “GripOn”.

   Set [I/O State] to “On”.

8. Select [Hand_Off]-[Grip] to set [Controller I/O Bit] to “GripOn”.

   Set [I/O State] to "Off".

Perform the steps 9 and 10 only when using a hand with two outputs.

9. Select [Hand_On]-[Release] to set [Controller I/O Bit] to “GripOff”.

   Set [I/O State] to "Off".

10. Select [Hand_Off]-[Release] to set [Controller I/O Bit] to “GripOff”.

```
Set [I/O State] to "On".
```

Perform the steps 11 and 12 only when using a hand with input.
11. Select [Hand_On]-[Input1] to set [Controller I/O Bit] to “PartGripped”.

```
Set [I/O State] to “On”.
```

12. Select [Hand_Off]-[Input1] to set [Controller I/O Bit] to “PartGripped”.

```
Set [I/O State] to "Off".
```

13. Click the [Apply] button to complete the hand registration.

How to replace with hand control command

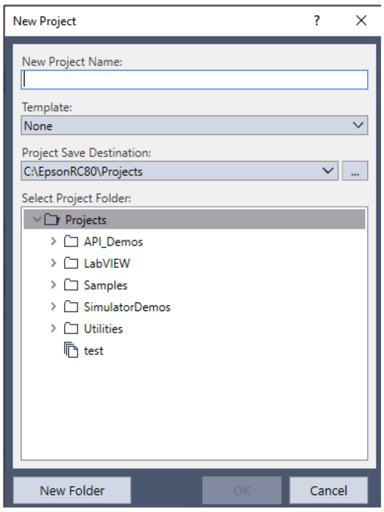
The commands related to hand On/Off can be replaced with hand control commands as shown in the table below.

Before replacement	After replacement
Off GripOff On GripOn	Hand_On {Hand number}
On GripOff Off GripOn	Hand_Off {Hand number}
Wait Sw(PartGripped) = On	Wait_HandOn({Hand number}) = True
Wait Sw(PartGripped) = Off	Wait Hand_Off({Hand number}) = True

6.10.2 [New] Command (Project Menu)

: Ctrl+Shift+N

The New command is used to create a new Epson RC+ 8.0 project.

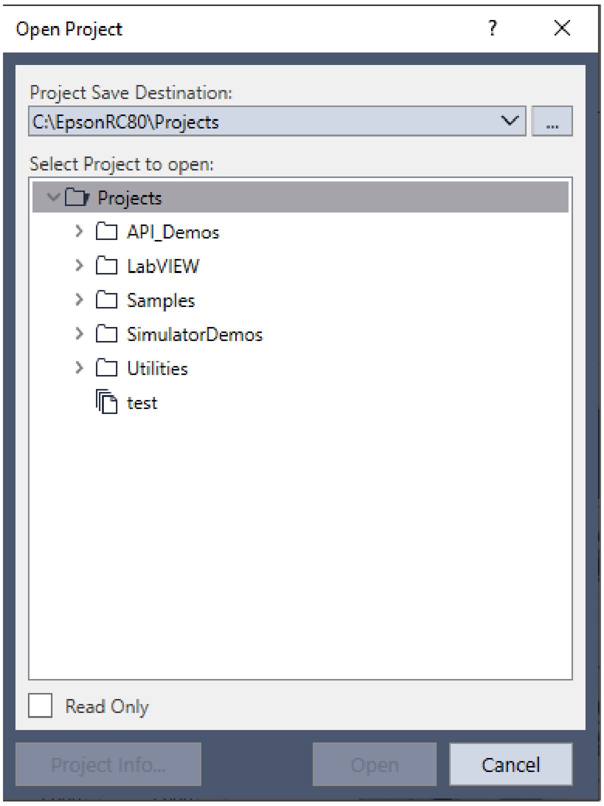


Item	Description
New Project Name	Type in a new name for the project. The name can include alphanumeric characters along with underscores(_).
Template	Select a project template. The new project will be a copy of the template project.
Project save destination	Select the folder to save the new project to. When specifying a new destination, click the [...] button to set the save destination. Set the destination in advance in the Epson RC+ menu, [Setup]-[Preferences]-[Workspace]-[Project Save Destination].
Select Project Folder	This is a list of folders and projects on the selected save destination. If you click on a name of projects in this list, the name appear in the [New Project Name] text box and can be edited. You can then edit the name, or you can create a new project with the same name as one that has already been created. In the later case, you will be prompted to overwrite the old project if it is in the same folder.
New Folder	Creates a new folder in the currently selected folder.
OK	Creates the new project.
Cancel	Cancels the operation and closes the dialog.

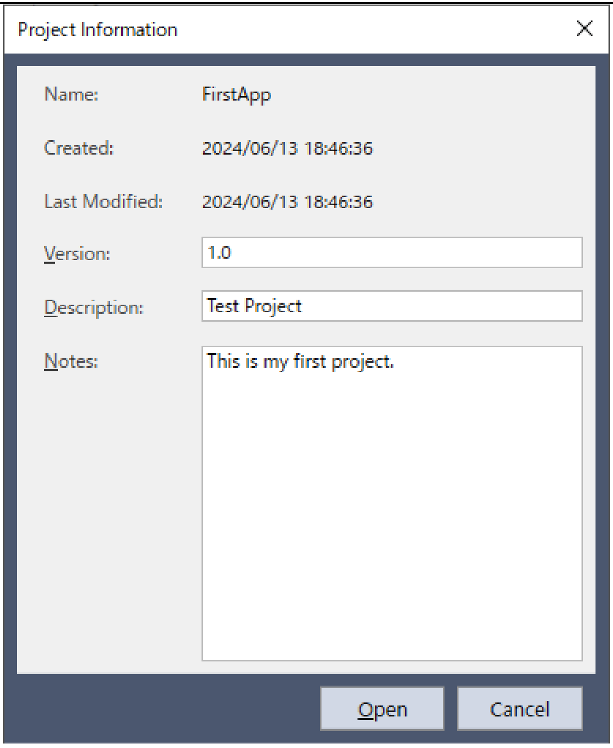
6.10.3 [Open] Command (Project Menu)

Ctrl+Shift+O

Use this command to open an Epson RC+ 8.0 project. When the project is opened, the previous project is closed. You will be prompted to save changes.



Item	Description
Project save destination	Select the save destination for the project you want to open. When specifying a new destination, click the [...] button to set the save destination. Set the destination in advance in the Epson RC+ menu, [Setup]-[Preferences]-[Workspace]-[Project Save Destination].
Select Project to open	Select a project name from the list box. To open a folder, double click on the folder or click ">" located to the left of the folder.
Read Only	If you set this checkbox and open a project, you cannot edit the program file, include file, point file, force file, I/O label, and user error.
Open	Opens the selected project.
Cancel	Cancels the operation and closes the dialog.
Project Info	Displays general project properties for the selected project. To view project information, first select a project in the list, then click the [Project Info] button.

Item	Description
	

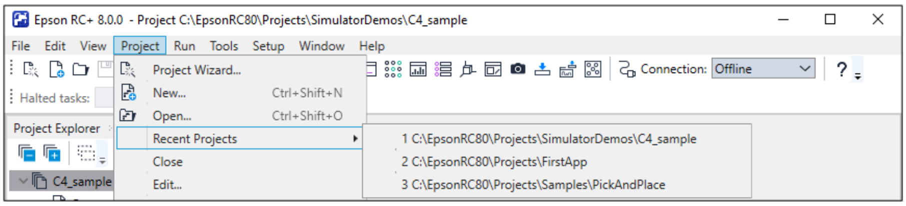
KEY POINTS

Project information for a project can be changed by selecting [Properties] from the [Project] menu after opening the project.

6.10.4 Recent Projects Submenu (Project Menu)

The Recent Projects submenu contains up to eight of the most recently used projects.

When you select a project in the menu, the current project is closed and the selected project is opened the same as if you used the [Open] command from the [Project] menu.



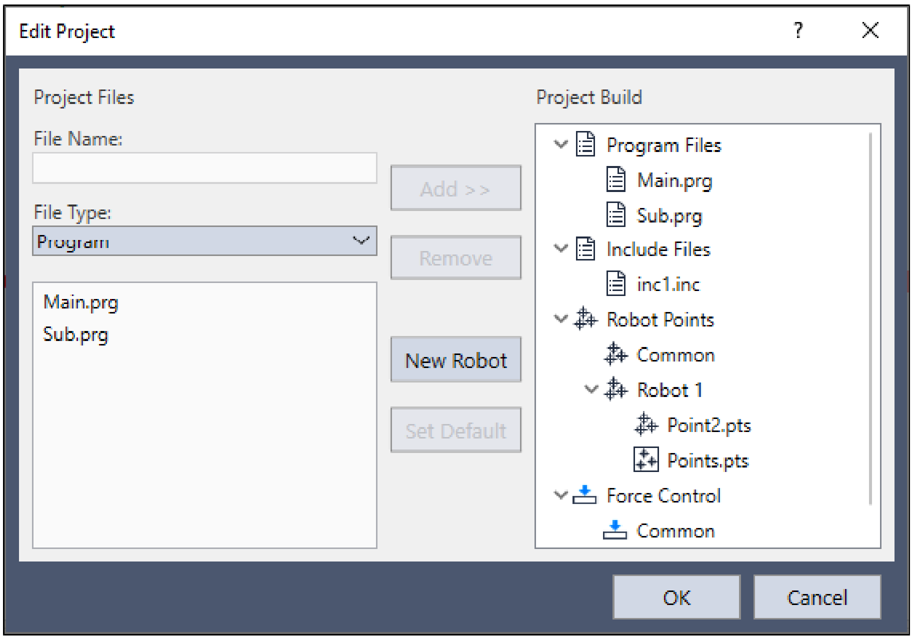
Select the [Setup]-[Preferences]-[Workspace]-[Clear recent project history] checkbox, to delete the history when you exit Epson RC+ 8.0.

6.10.5 [Close] Command (Project Menu)

Use the [Close] command to close the current project. Several menu and toolbar commands will be disabled after the project is closed.

6.10.6 [Edit] Command (Project Menu)

The [Edit] command is used to define which program files, include files, point files, and force files are to be used in the current project. The [Project Files] contains a list of files in the current project folder. You can select which files to view from the [File Type] list box. The [Project Build] on the right shows the projects in the tree. The tree includes program files, include files, point files, and force files.



To use a file in the [Project Files] list in a project, click the [Add] button and register the file in the [Project Build] tree.

TIP

In the [Project Explorer], right-click on the target robot of a program file, include file, point file, or force file to add an existing program from another project from the context menu.

Create a new program file, include file

1. Enter the file name in the [File Name] text box.

   The name of the include file can also be the same name as a program. The name can include alphanumeric characters along with underscores(_).

2. Click the [Add] button. You will be prompted to create a new file. Click the [Yes] button. The file created will be added to the [Project Build] tree.

Add an existing program file or include file to a project

1. Select the file that you want in the [File Type] box.
2. Select the file name to add to the project from the list.
3. Click the [Add] button. Alternatively, double click the selected file. Files are added to the [Project Build] tree.

Add a new point file, force file

1. Enter the file name to create in the [File Name] box.

   The name can include alphanumeric characters along with underscores(_).

2. Select the robot folder you want to register from the [Project Build] tree.
3. Click the [Add] button. You will be prompted to create a new file. Click the [Yes] button. The file created will be added to the robot selected in the [Project Build] tree.

Add an existing point file or force file to the project

1. Select the file that you want in the [File Type] box.
2. Select the robot folder you want to register from the [Project Build] tree.
3. Select the file name to add to the project from the list.
4. Click the [Add] button. The file will be added to the selected robot of the [Project Build] tree.

Remove a program file, include file, point file, force file

1. Select the file you want to remove in the [Project Build] tree.
2. Click the [Remove] button to remove the file from the [Project Build] tree.

   The file is not deleted from the project folder, so you will still see the file in the file list.

To add a new robot

1. Click the [New Robot] button. This adds a robot to [Robot Points], [Force Control] in the [Project Build] tree.

To set a default point file

The default point file is a point file that is automatically loaded to a robot with the project load. Each robot can have one point file as the default.

1. Select a point file to set as the default from each robot of the [Robot Points] folder in the [Project Build] tree.
2. Click the [Set Default] button. The file will be set as the default of the registered robot.

> **KEY POINTS**
>
> The common point file is a point file that is available for all robots on the controller. To use this point file, you need to load it from the SPEL+ program to the robot using LoadPoints command.

6.10.7 [Save] Command (Project Menu)

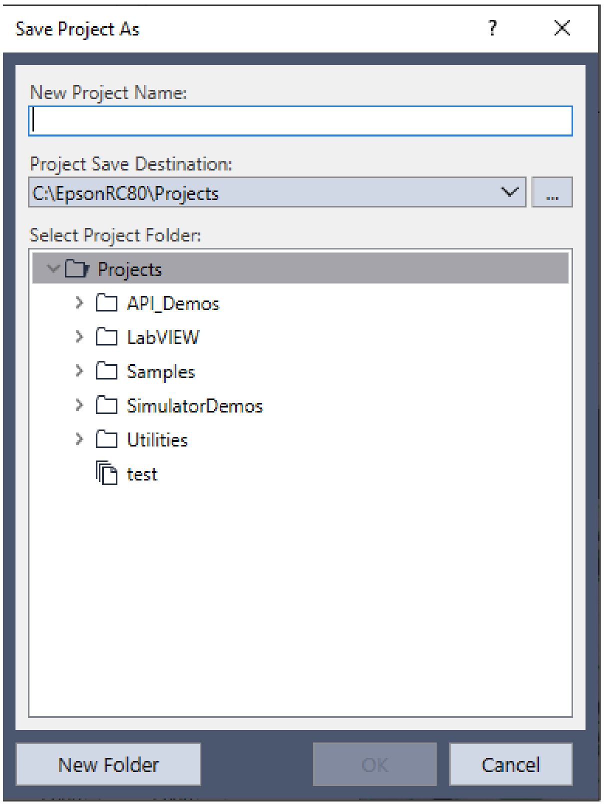
This command saves the active program file, include file, point file, force file, I/O label, or user errors. This menu selection will be dimmed if nothing needs to be saved.

> **TIP**
>
> It's a good idea to save files frequently while you are editing project files. Just click the  [Save all files] button on the toolbar to save all of your files.

6.10.8 [Save As] Command (Project Menu)

After saving and closing the current project, create a duplicate with a different name and open the created project.

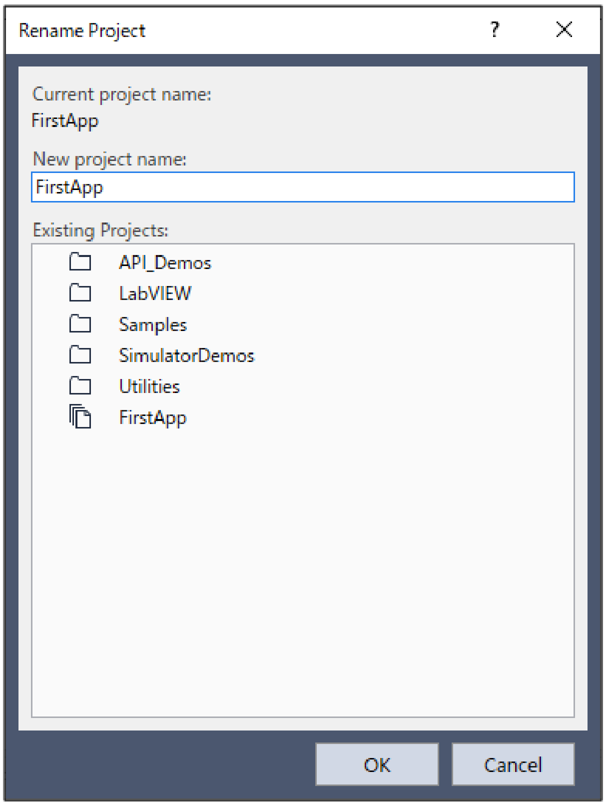
If the [Setup]-[Preferences]-[Workspace]-[Auto_file save] checkbox is selected, select [Save Project As] to save the project at the same time.



Item	Description
New Project Name	Type in a new name for the project. Project names can contain alphanumeric characters along with underscores. The maximum number of characters is 24. You can use the same name as the current project if you select a location different from the current project folder.
Project save destination	The folder that new projects are saved to. When specifying a new destination, click the [...] button to set the save destination. Set the destination in advance in the Epson RC+ menu, [Setup]-[Preferences]-[Workspace]-[Project Save Destination].
Select Project Folder	Click on the desired folder for the project.
New Folder	Click this button to create a new folder under the Projects folder.
OK	Click this button to save a project with a new name to the specified save destination.
Cancel	Cancels the operation and closes the dialog.

6.10.9 [Rename] Command (Project Menu)

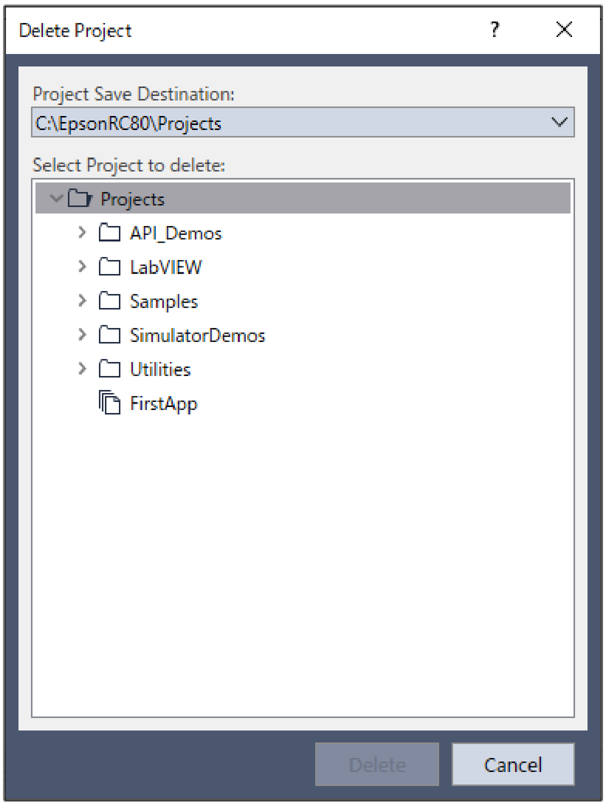
This command renames the current project. The project folder and all associated project files are also renamed.



Item	Description
Current Project Name	The name of the current project.
New project name	Type in a new name for the project. Project names can contain alphanumeric characters along with underscores.
Existing Projects	This shows projects in the selected folder. The new name you choose cannot be one of the names in this list.
OK	Renames the project.
Cancel	Cancels the operation.

6.10.10 [Delete] Command (Project Menu)

This command deletes an entire project from a PC disk. All files in the project folder will be destroyed.



Item	Description
Project save destination	Select the project save destination to delete.
Select Project to delete	Select a project to delete from the list.
Delete	Delete the project. You will be prompted to confirm the operation.
Cancel	Cancels the operation and closes the dialog.

6.10.11 [Import] Command (Project Menu)

The Project Menu Import Command uses a wizard to import projects from a PC, the current controller, or a controller status folder.

When a project is imported, the files are copied to a new project folder, so the original project is not changed.

You can only import projects created with Epson RC+ 8.0 or EPSON RC+ 7.0.

KEY POINTS

When importing projects created with earlier versions than EPSON RC+ 7.0, import the project into EPSON RC+ 7.0, and then import the imported project into Epson RC+ 8.0.

For details on the conversion method, see below.

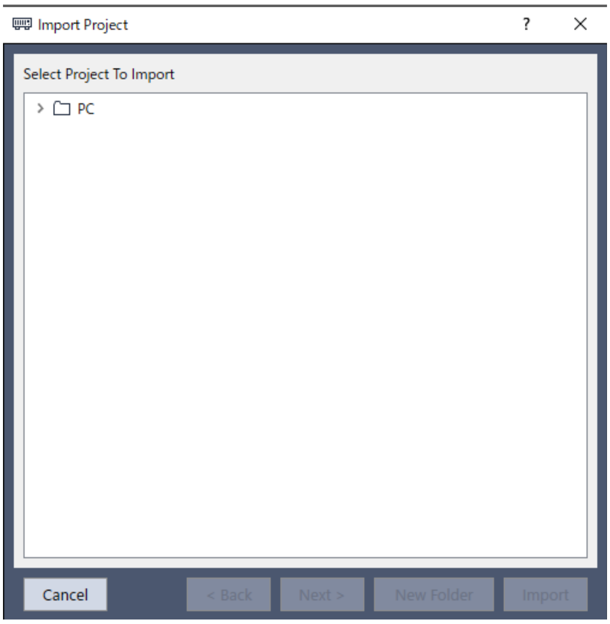
Notes for Users of EPSON RC+ 7.x or Earlier Versions

The sections below have instructions for importing a project from each type of source location.

Importing a PC project

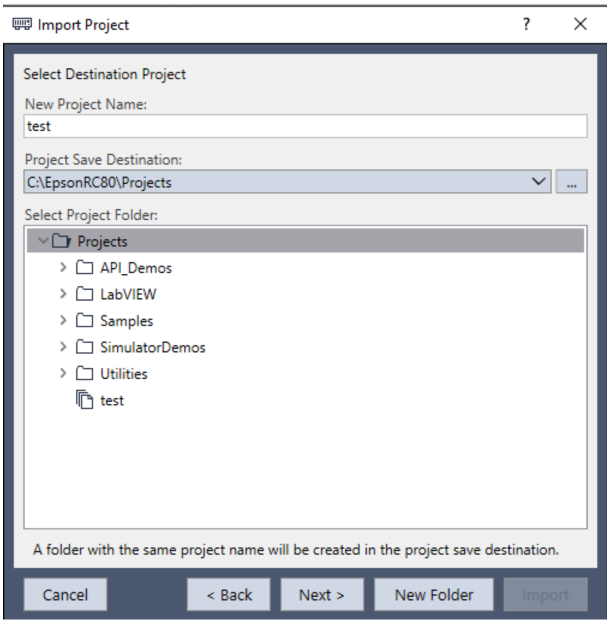
Follow these steps to import a project from a PC:

1. In the Epson RC+ 8.0 menu, select [Project]-[Import Project]-[PC] to open the [Import Project] dialog box.



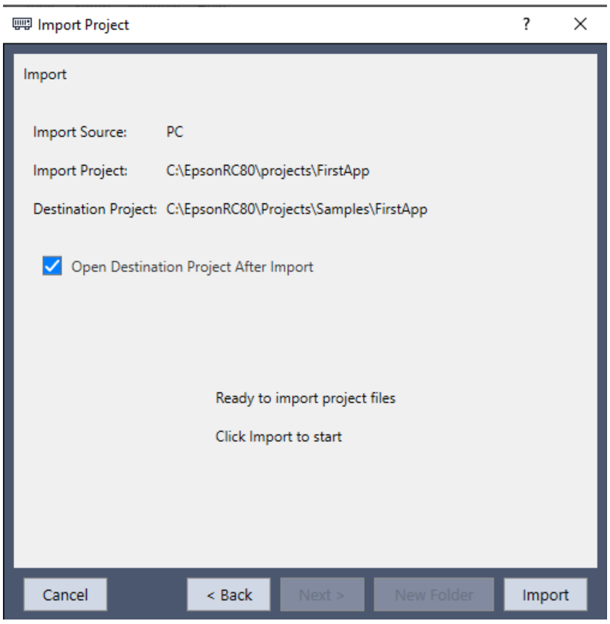
2. Select the project to import, and then click the [Next] button.

3. The new project name is set to the name of the imported project. The name of the new project can be changed. Select the project save destination and project folder, then click the [Next] button. To create a new folder, click [New Folder].



4. Verify the import source, import origin, and save destination.

   Check [Open Destination Project After Import] if you want the project to open after import.



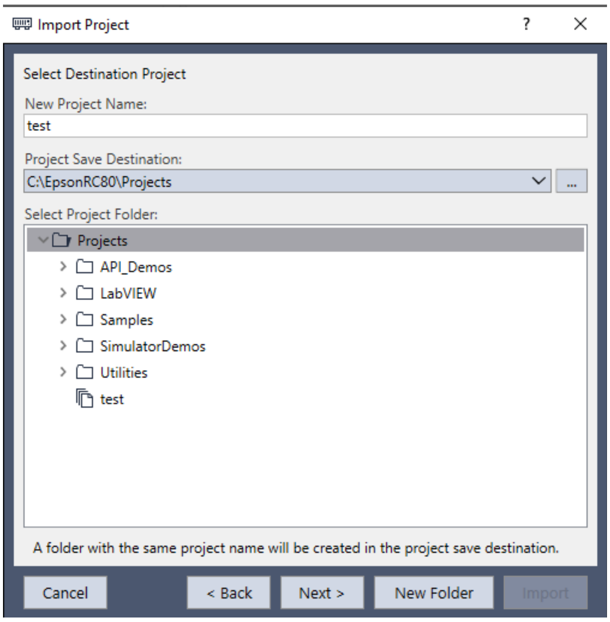
5. Click the [Import] button. If a project with the same name already exists at the import destination, you will be asked if you want to overwrite it.

Importing a Controller project

Follow these steps to import a project from a controller:

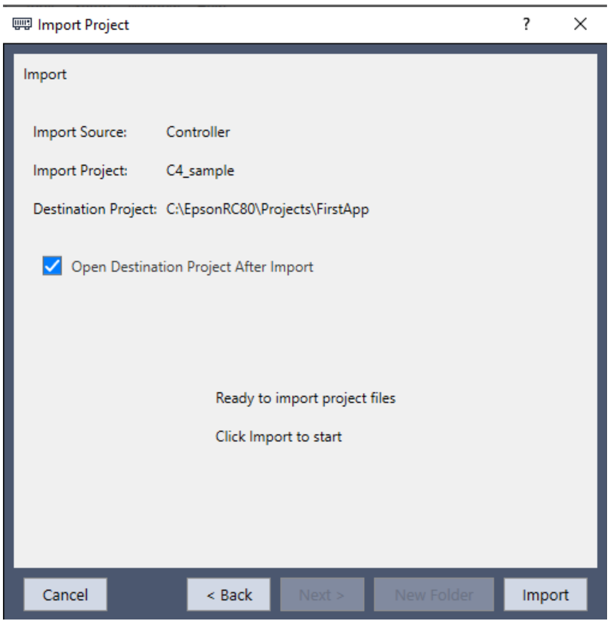
1. In the Epson RC+ 8.0 menu, select [Project]-[Project Import]-[Controller] to open the [Import Project] dialog box.

2. The new project name is set to the name of the current project in the controller. You can modify the new project name if desired. Select the project save destination and project folder, then click the [Next] button.



3. Verify the import source, import project, and destination project.

   Check [Open Destination Project After Import] if you want the project to open after import. If the import destination already exists, you will be asked if you want to overwrite it.



4. Click the [Import] button.

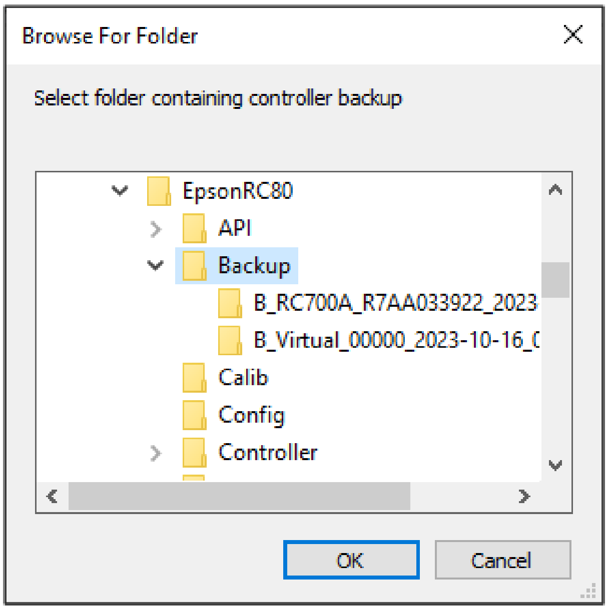
5. The project in the destination project will be built.

Importing a Controller Status project

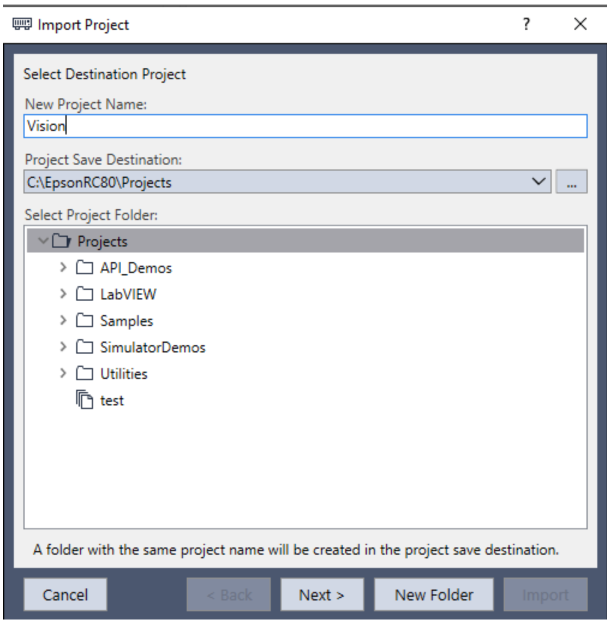
Follow these steps to import a project from the Controller Backup Folder:

1. In the Epson RC+ 8.0 menu, select [Project]-[Import Project]-[Controller Backup Folder] to open the [Browse For Folder] dialog box.

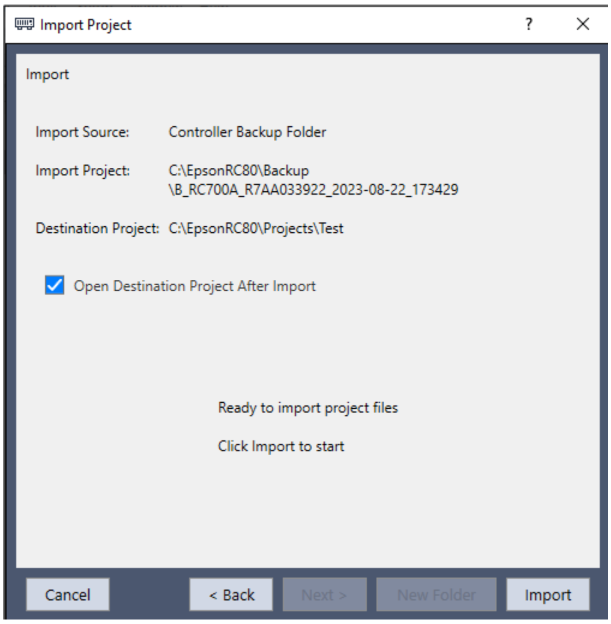
2. Select a controller status folder and click the [OK] button.



3. The new project name is set to the project found in the controller backup folder. You can modify the new project name if desired. Select the project save destination and folder, then click the [Next] button.



4. Verify the import source, import project, and destination project. Check [Open Destination Project After Import] if you want the project to open after import. If the import destination already exists, you will be asked if you want to overwrite it.

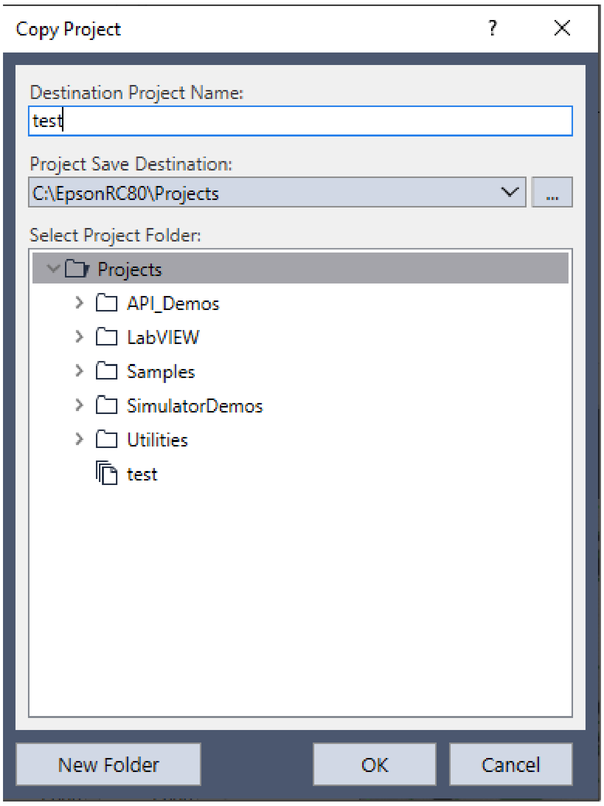


5. Click the [Import] button.

6.10.12 [Copy] Command (Project Menu)

The [Copy] command copies all files to a specified folder or project. You can use the current project name if you select a new project save destination or folder. You can also specify a new name for the destination project.

You should use the [Copy] command to make backup copies of your project on a regular basis.



Item	Description
Destination Project Name	Type in a name for the new copy of the project. Project names can contain alphanumeric characters along with underscores. The maximum number of characters is 24. You can use the same name as the current project if you select a different project save destination to the current project save destination and folder.
Project save destination	Select the project save destination to copy. When specifying a new destination, click the [...] button to set the save destination. Set the destination in advance in the Epson RC+ menu, [Setup]-[Preferences]-[Workspace]-[Project Save Destination].
OK	Performs the copy process.
Cancel	Cancels the operation and closes the dialog.

6.10.13 [Build] Command (Project Menu)

: Ctrl+B

This command builds the current project so that it can be executed.

The Build command does the minimum amount of work required to bring the project up to date in the robot controller. For example, if a change was made to one program file in the project, then build will compile the changed file only, link it with the remaining object files, and the new file will be sent to the controller.

KEY POINTS

When sending the required files to the compact vision, make sure to rebuild not build.

During the build process, the status window displays each step of the build. If there are any errors, they will be displayed on the status window.

6.10.14 [Rebuild] Command (Project Menu)

Ctrl+Shift+B

Rebuilds the entire current projects. All program files are re-compiled, linked, and sent to the controller. All point files in the project are sent to the controller.

> **KEY POINTS**
>
> If using the camera of the compact vision, rebuild to send the required files to the compact vision.

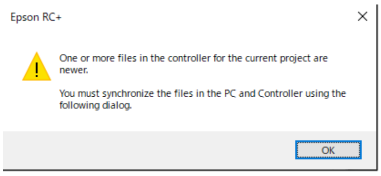
6.10.15 [Synchronize Project] Command (Project Menu)

If following conditions are fulfilled when PC and Controller are connected, [Synchronize Project] can be selected.

- When the project name on the PC and on the Controller was the same.
- When the file in the Controller has been changed after the last project build on the PC.

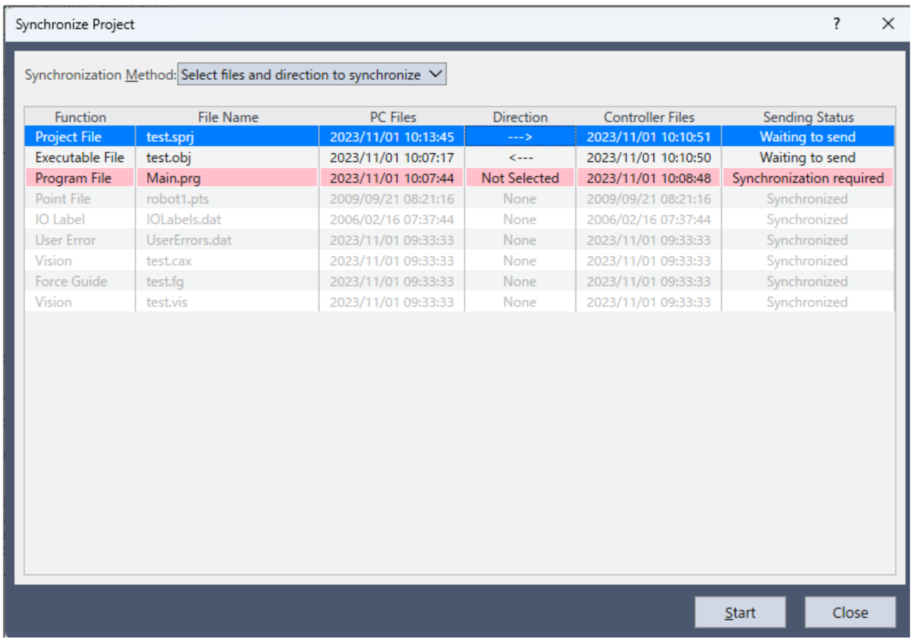
How to set the synchronization

1. Select Epson RC+ 8.0 Menu-[Project]-[Synchronize Project]. The message shown below will appear.



2. Click the [OK] button.

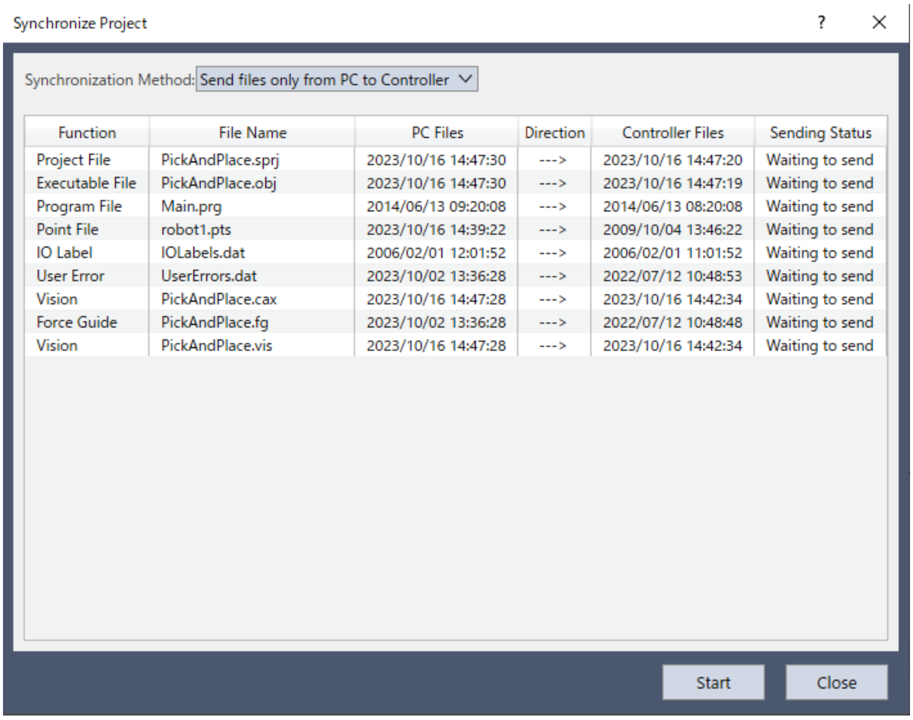
3. The dialog shown below appears. Select the method of synchronization and click the [Start] button.



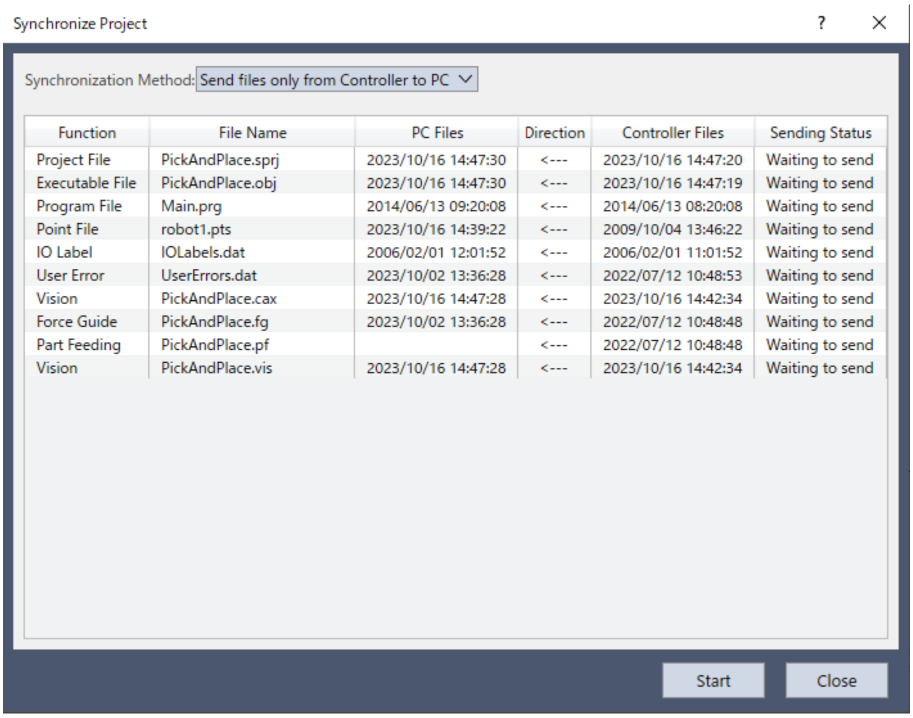
4. The project on the PC and on the Controller synchronized.

Item	Description
Synchronization Method	Select the method of synchronization. ▪ When the "Send files only from PC to Controller" is selected, the file on the PC is copied to the Controller. ▪ When the "Send files only from Controller to PC" is selected, the file on the Controller is copied to the PC.
Function	Displays the file type.
File Name	List of files in the project.
PC Files	The update timestamps for PC files.
Direction	Indicates the direction of file transfer. ▪ -->: PC to Controller. ▪ <---: Controller to PC. ▪ None: It cannot be synchronized because the files are the same. ▪ Select: It is indicated in pink. The direction of file transfer needs to be selected by user because the file was changed after the last project build on PC.
Controller Files	The update timestamps for Controller files.
Sending Status	Indicates the status of the file. ▪ Waiting to send: Waits for synchronization method to be selected. ▪ Synchronized: The files are the same.
Start	Starts the synchronization.
Close	Closes [Synchronize Project] dialog. When [Close] button clicked without project synchronization, error 10015 "Project could not be synchronized." is displayed. The synchronization will not be done and connection to the Controller will disconnected.

When "Send files only from PC to Controller" is selected for Synchronization Method:

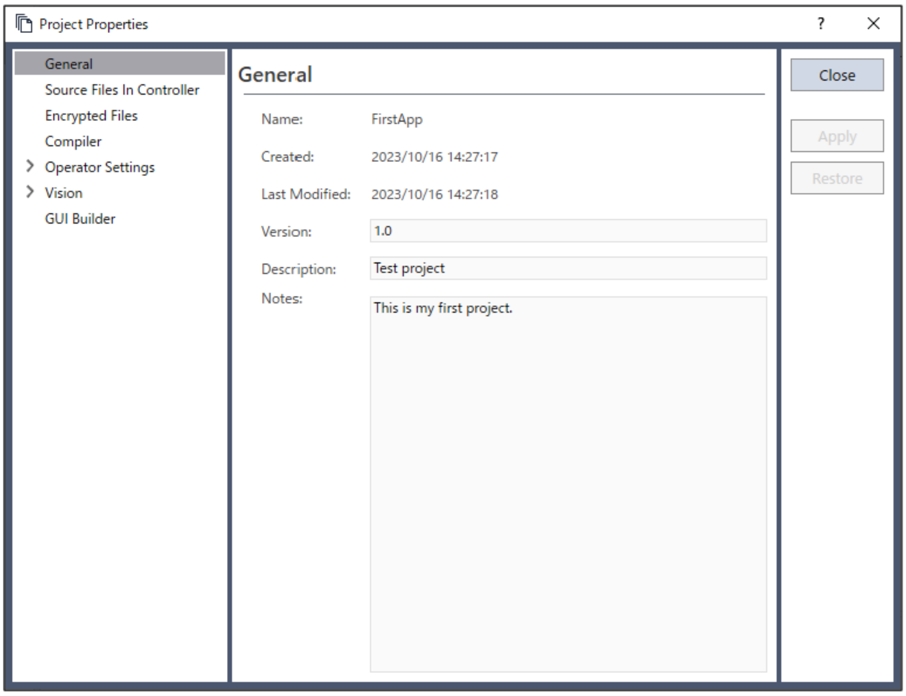


When "Send files only from Controller to PC" is selected for Synchronization Method:



6.10.16 [Properties] Command (Project Menu)

In the Epson RC+ 8.0 menu, select[Project]-[Properties] menu to open the [Project Properties] dialog.

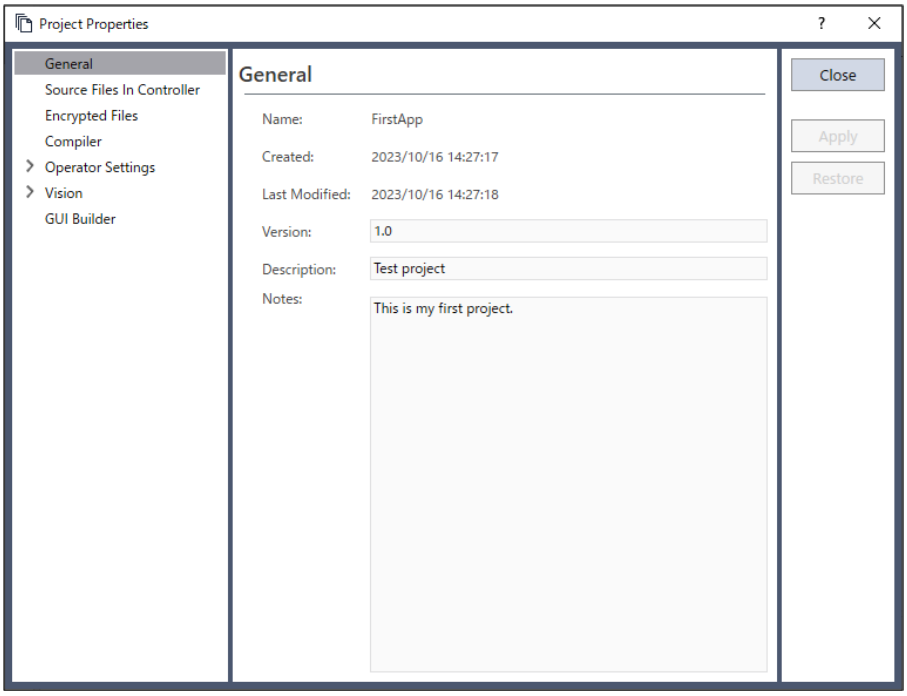


Common buttons

Item	Description
Close	Close the [Project Properties] dialog.
Apply	Sets current values after changes have been made.
Restore	Revert back to previous values.

6.10.16.1 [Project]-[Properties]-[General] Page

View general properties for the current project. You can also edit general properties. All project property settings are stored in the project file. They are also stored in the controller during project build



Item	Description
Name	The name of the current project.
Created	Date and time when the project was created.
Last Modified	Date and time when the project was last modified.
Version	User version number of the project. You can type any text here.
Overview	A description of the project. You can type any text here.
Notes	Any project notes can be entered into this section.

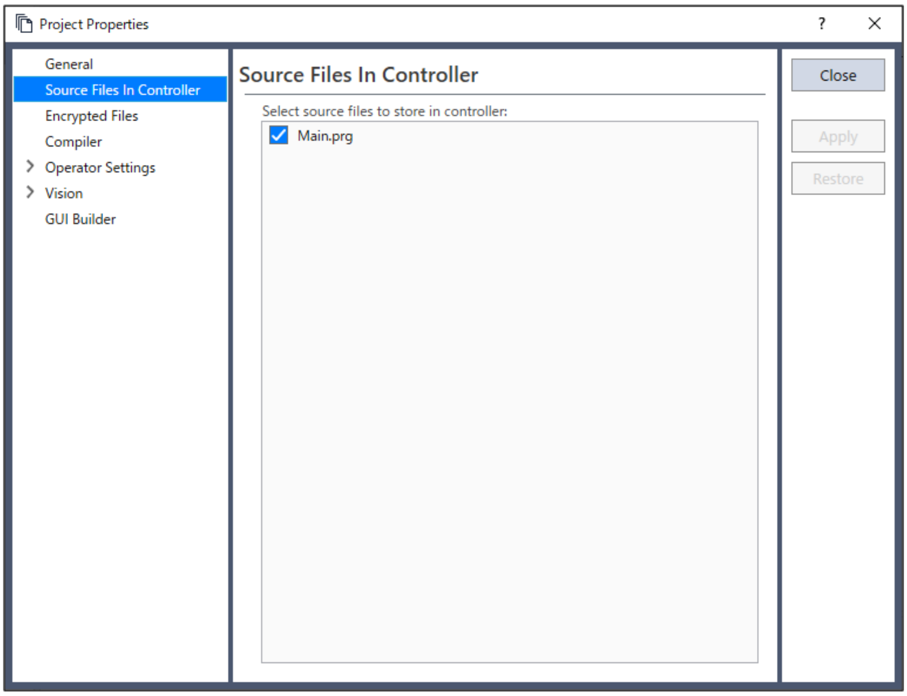
TIP

When the [Open Project] dialog is used, clicking the [Project Info] button will open a dialog that contains the general project properties entered on this page.

6.10.16.2 [Project]-[Properties]-[Source Files In Controller] Page

This page allows you to select which source files will be stored in the controller during project build.

After changes are applied, the next project build will clear the project in the controller and perform a rebuild.



Item	Description
Select source files to store in controller	This is a list of the source files in the project. Select which source files you want to have stored in the controller.

6.10.16.3 [Project]-[Properties]-[Encrypted Files] Page

This page allows you to encrypt files in the current project.

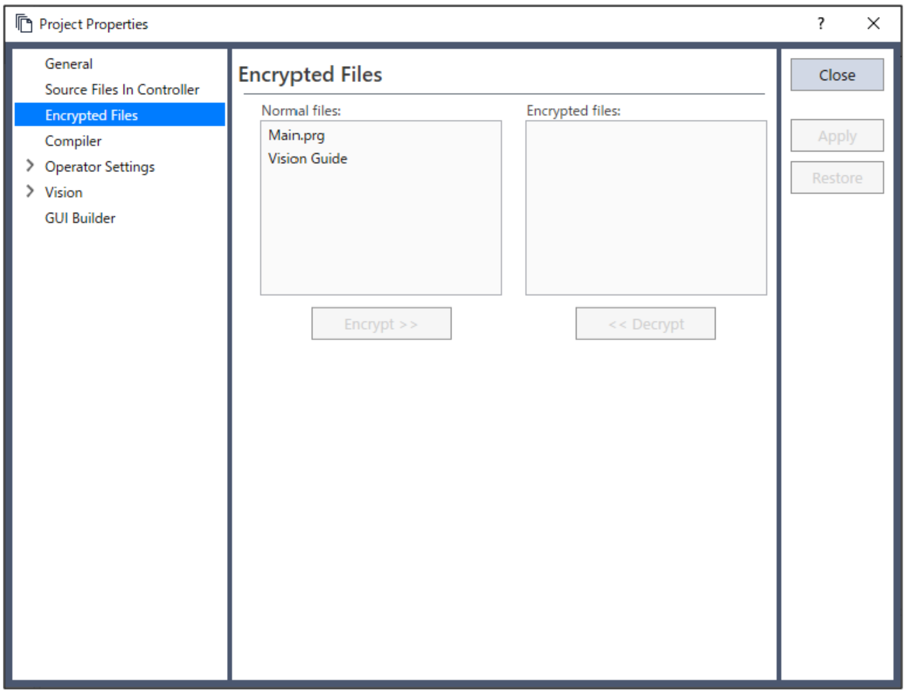
For details on using encrypted files, see below.

Using Encrypt Files

⚠ CAUTION

USE EXTREME CAUTION!

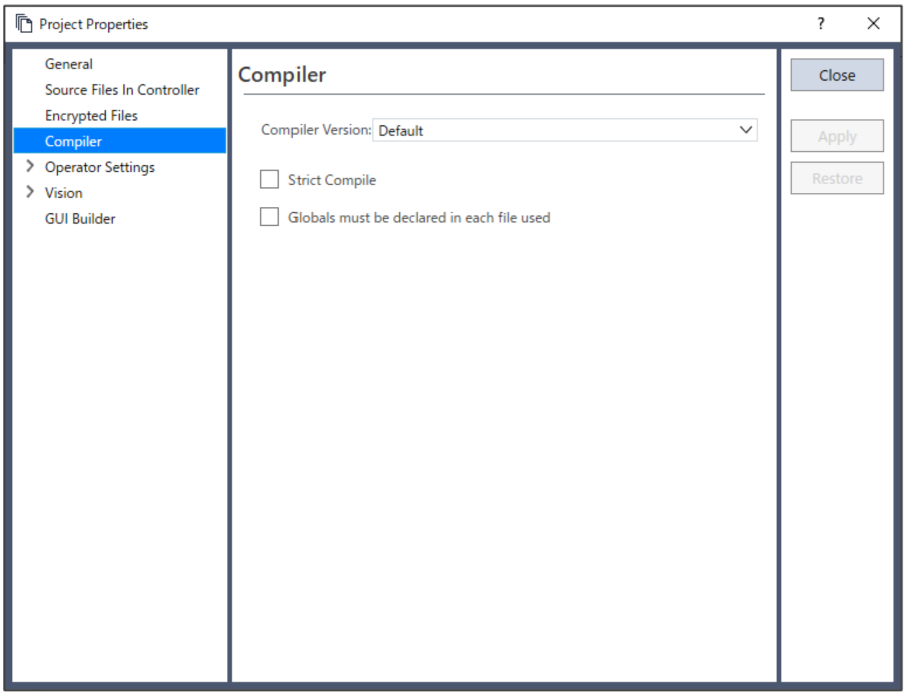
Keep a record of the password(s) used for encryption in a safe place. Once a file is encrypted, it can only be opened with the password you enter. If you forget the password, the file contents CANNOT BE RECOVERED.



Item	Description
Normal files	This is a list of the source files in the project that are not encrypted.
Encrypted files	This is a list of the source files in the project that are encrypted. Select which source files you want to decrypt.
Encrypt >>	Encrypts the files selected in the [Normal files] list. When this button is clicked, you will be prompted for a password that will be used to access these encrypted files.
<<Decrypt	Decrypts the files selected in the [Encrypted files] list. When this button is clicked, you will be prompted for the password that was used to encrypt the files.

6.10.16.4 [Project]-[Properties]-[Compiler] Page

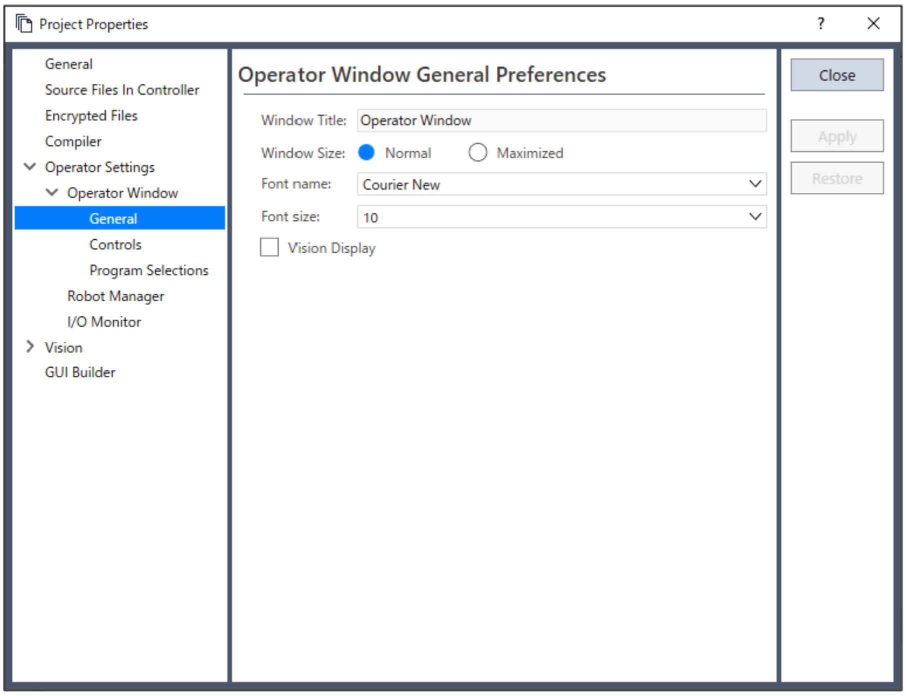
This page allows you to configure the compiler settings.



Item	Description
Compiler Version	[Default] is the normal setting. When the projects cannot be built because new SPEL+ language keywords have been added that conflict with your variable names, you can select a previous version to build the projects. Specify the controller version that compiles the project. Epson RC+ 8.0 is compatible with EPSON RC+ 7.0. For details on compatibility, refer to the following manual. "SPEL+ Language Reference - Appendix B: Precaution of Compatibility"
Strict Compile	Checks the Boolean type strictly. If the program contains following descriptions, an error will occur. ▪ Boolean variables are assigned to other numerical types ▪ Specifies a wait time to Wait ▪ Compares Boolean types
Globals must be declared in each file used	Checks the Global variables (including Global Preserve variables) for each file. When this item is checked, you must declare Global variables in each file in which they are used, otherwise an error will occur at build time. Enabling this item reduces a build time of a project which uses many Global variables.

6.10.16.5 [Project]-[Properties]-[Operator Settings]-[Operator Window]-[General] Page

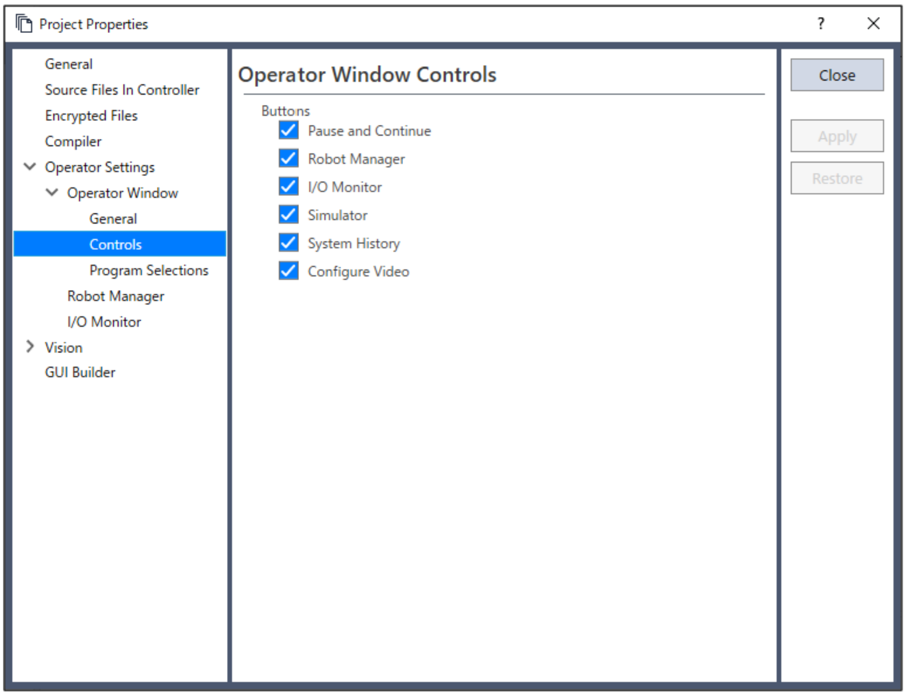
This page allows you to configure the general settings for the Operator Window.



Item	Description
Window Title	Type in the title that you want to appear at the top of the operator window. Up to 64 characters can be entered for the title.
Window Size	Choose Normal or Maximized.
Font Name	Choose the font you desire for the operator window.
Font Size	Choose the size of the font you desire for the operator window.
Vision Display	If this checkbox is set, the Vision Guide image will be displayed in the operator window.

6.10.16.6 [Project]-[Properties]-[Operator Settings]-[Operator Window]-[Controls] Page

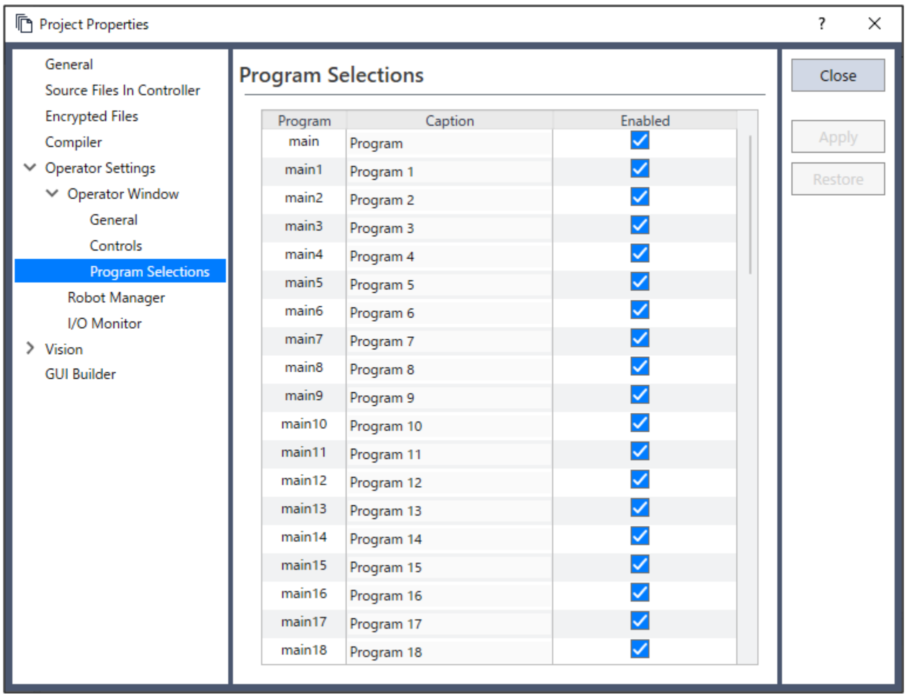
This page allows you to configure the controls for the Operator Window.



Item	Description
Pause and Continue	Check this box if you want the [Pause] and [Continue] buttons to be displayed. This will allow the operator to pause and continue from the operator window.
Robot Manager	Check this box if you want the [Robot Manager] button to be displayed. This will allow the operator to open the Robot Manager from the operator window.
I/O Monitor	Check this box if you want the [I/O Monitor] button to be displayed. This will allow the operator to view input and output status.
Simulator	Check this box if you want the [Simulator] button to be displayed. This will allow the operator to open the Robot Manager from the operator window.
System History	If this checkbox is set, the [System History] button will appear. You can check the system history.
Configure Video	Check [Project Properties]-[Operator Settings]-[General]-[Vision Display]. Check this box if you want the [Configure] button to be displayed. You can configure video displays from the operator window.

6.10.16.7 [Project]-[Properties]-[Operator Settings]-[Operator Window]-[Program Selections]

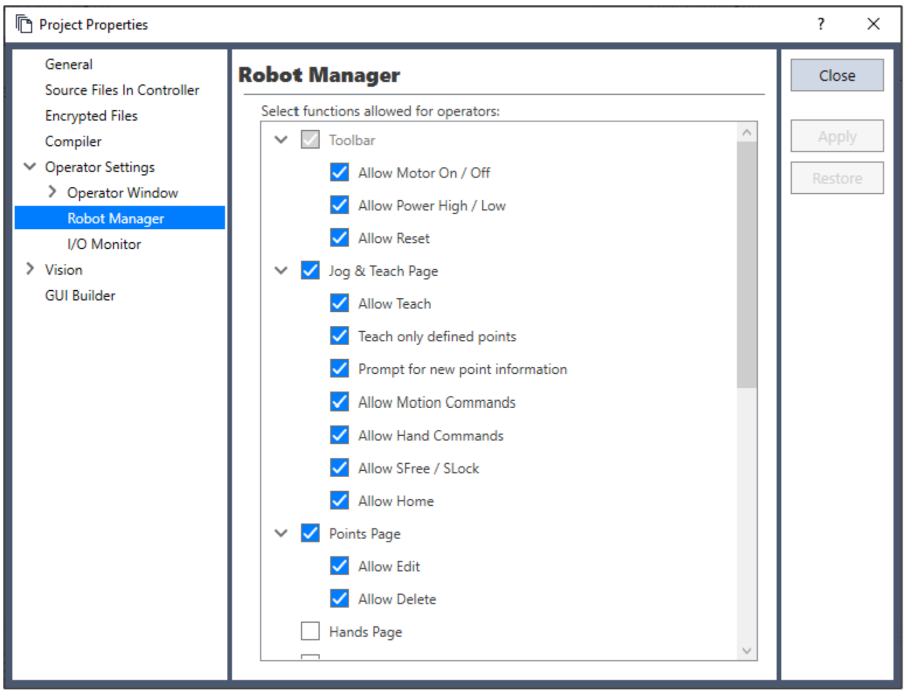
This page allows you to configure programs operated in the operator window.



Item	Description
Program Selections	Choose the programs operated in the operator window and title. Each project can have up to 64 programs that can be started from the Operator Window. The programs are named main, main1 main2, … main63. Each program has an associated startup function using the same name as the program (main, main1, main2…main63). In the program selections grid, you can define a friendly name for each of the 64 programs. You can also select the programs to be displayed in the program list of the operator window by checking the [Enabled] checkbox.

6.10.16.8 [Project]-[Properties]-[Operator Settings]-[Robot Manager]-[Function selection]

Use this page to configure the Robot Manager for operators.

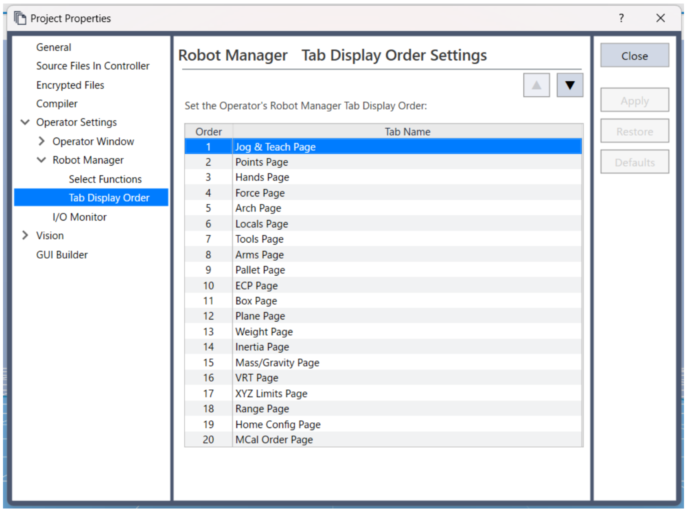


Item	Description
Select functions allowed for operators	Check the pages that you want the operator to have access to when the Robot Manager is displayed from the operator window. In some pages, there are additional options.
Tool Bar	Sets items shown in the toolbar. This checkbox is selected by default and cannot be changed.
- Allow Motor On/Off	Allows the operator to switch the motor on or off.
- Allow Power High/Low	Allows the operator to switch the power high or low.
- Allow Reset	Allows the operator to reset the Emergency Stop status.
Jog & Teach Tab	Sets items shown in the [Jog & Teach] panel.
- Allow Teach	Allows the operator to teach points from the [Jog & Teach] page.
- Teach only defined points	Only defined points are shown in the point list on the [Jog & Teach] page.
- Prompt for new point information	When the operator teaches a new point, a dialog will be displayed for entering the point label and description.
- Allow Motion Commands	Allows the operator to execute motion commands from the [Jog & Teach] page.
- Allow Hand Operation	Allows the operator to execute hand commands from the [Jog & Teach] page.
- Allow SFree / SLock	Allows the operator to free or lock joints from the [Jog & Teach] page.
- Allow Home	Allows the operator to home the robot from the [Jog & Teach] page.
Points Page	Sets items shown in the Points Page.
- Allow Edit	Allows the operator to edit point data on the [Points] page.
- Allow Delete	Allows the operator to delete points on the [Points] page.
Hand settings tab	Allows the operator to view and edit the Hand Settings tab.
Force data tab	Allows the operator to view and edit the Force Data tab.
Arch settings tab	Allows the operator to view and edit the Arch Settings tab.
Local settings tab	Allows the operator to view and edit the Local Settings tab.
Tool settings tab	Allows the operator to view and edit the Tool Settings tab.
Arms tab	Allows the operator to view and edit the Arms tab.
Pallets tab	Allows the operator to view and edit the Pallets tab.
External control point settings	Allows the operator to view and edit the External Control Point Settings tab.
Boxes tab	Allows the operator to view and edit the Boxes tab.
Planes tab	Allows the operator to view and edit the Planes tab.

Item	Description
Weight tab	Allows the operator to view and edit the Weight tab.
Inertia tab	Allows the operator to view and edit the Inertia tab.
Mass setting tab	Allows the operator to view and edit the Mass Setting tab.
VRT tab	Allows the operator to view and edit the VRT tab.
XYZ Limits tab	Allows the operator to view and edit the XYZ Limits tab.
Range tab	Allows the operator to view and edit the Range tab.
Home Config tab	Allows the operator to view and edit the Home Config tab.
MCORDR setting tab	Allows the operator to view and edit the MCORDR Setting tab.

6.10.16.9 [Project]-[Properties]-[Operator Settings]-[Robot Manager]-[Tab Order]

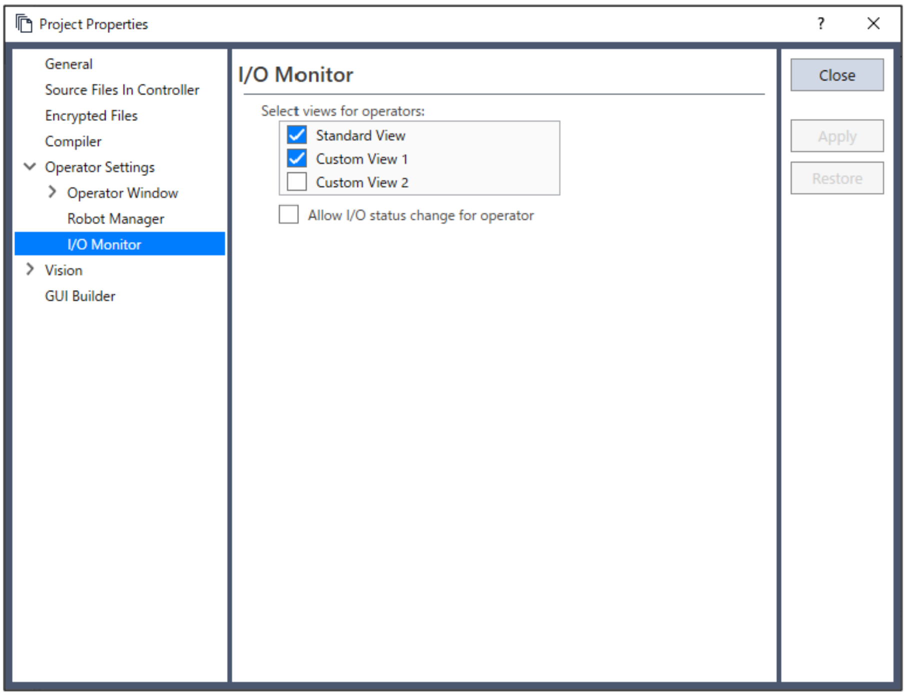
Sets the Robot Manager tab display order.



Item	Description
▼▲	Sorts the tab display order list.
Set the Robot Manager Tab Display Order:	Sets the Robot Manager tab display order that the operator can use when displaying the operator window.
Order	Shows the display order of the tabs.
Tab name	Shows the Robot Manager tab name.

6.10.16.10 [Project]-[Properties]-[Operator Settings]-[I/O Monitor] Page

Use this page to configure the I/O Monitor for operators.

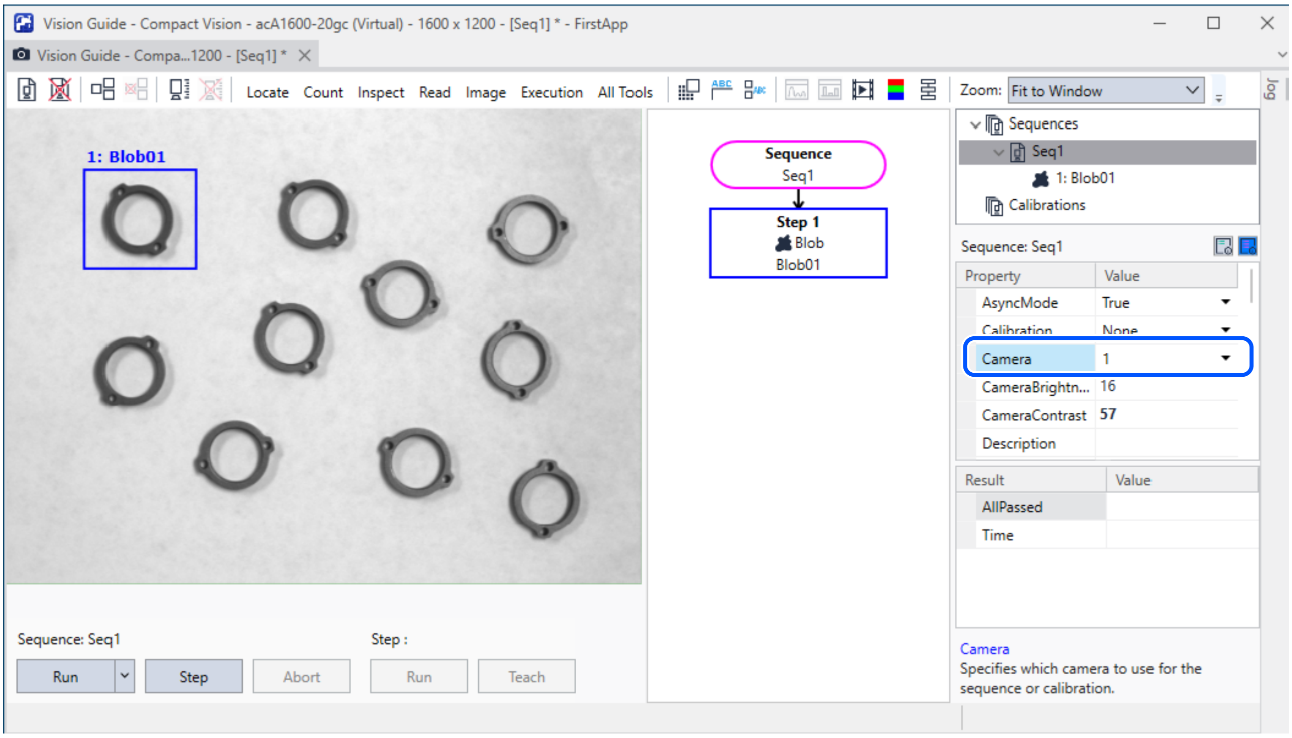


Item	Description
Select views for operators	Configures the I/O views that operators use when opening the [I/O Monitor] from the [Operator Window]. You can configure the custom views.
Allow I/O status change for operator	Check this box if you want to allow operators to turn on or off the inputs and outputs.

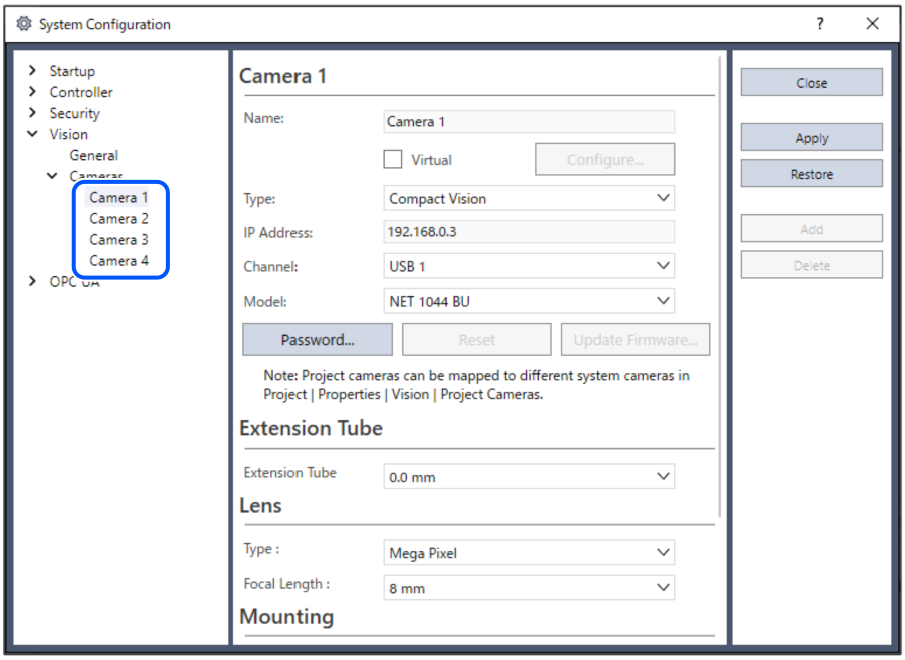
6.10.16.11 [Project]-[Properties]-[Vision]-[Project Cameras]

You can map project cameras to system cameras.

Project camera is a camera specified by Camera property of vision sequence or calibration.



System camera is a camera configured on System Configuration of RC+.



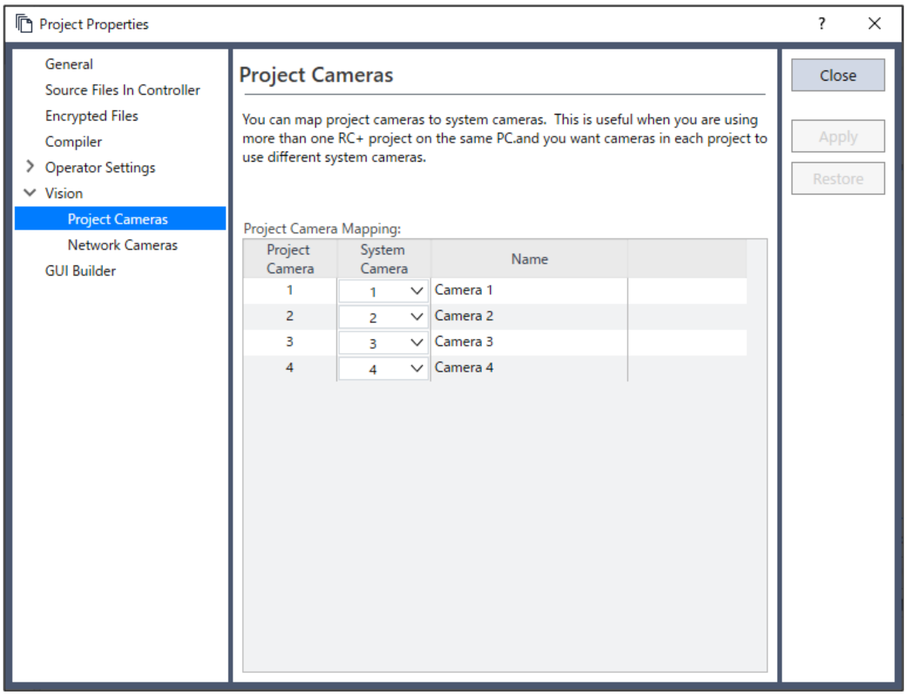
This function is useful when you use multiple projects on the same PC and refer to the different system camera numbers from the each project camera.

By default, project cameras are mapped one to one with system cameras.

When a project is opened, the cameras used in the project are automatically mapped to system cameras. If one or more cameras cannot be mapped, the [Resolve Camera Configuration] dialog box is displayed.

For more details, refer to the following manual:

"Setup System Cameras and Project Cameras" in "Vision Guide 8.0 Hardware & Setup"



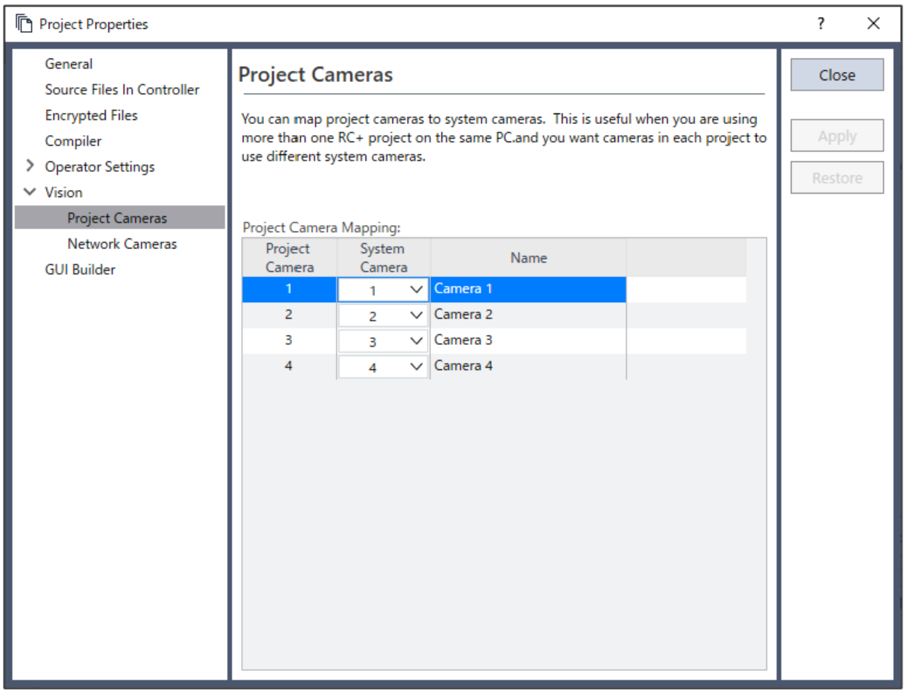
Item	Description
Project Camera	The project camera number to be mapped.
System Camera	Select the system camera to use for the project camera.

Mapping of a project camera

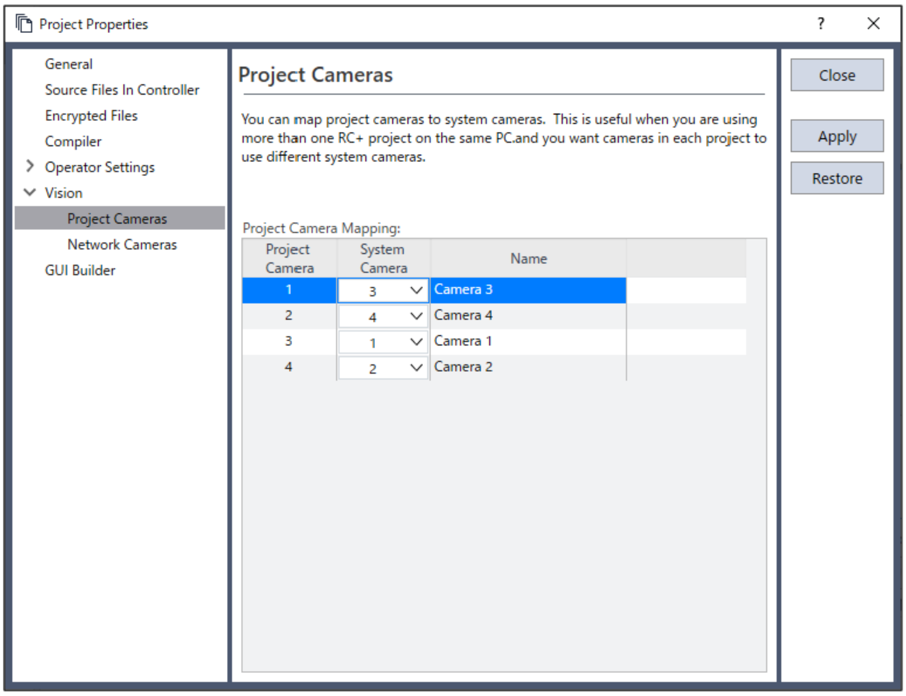
For example, let's assume four system cameras are set in RC+ that has two projects; A and B as shown in the table below.

System camera	Model
1	NS1044BU
2	NS4133CU
3	acA1600-60gm
4	acA2500-14gm

To map a camera for the project A, select [Project]-[Properties]-[Vision]-[Project Camera] with opening the project A. When setting NS1044BU as a camera 1 that is used in this project, map Project Camera 1 to System Camera 1. (set Project Camera 2, 3, and 4 in the same step.)



To map a camera for the project B, select [Project]-[Properties]-[Vision]-[Project Camera] with opening the project B. When setting acA1600-60gm as camera 1 that is used in this project, map Project Camera 1 to System Camera 3. (set Project Camera 2, 3, and 4 in the same step.)



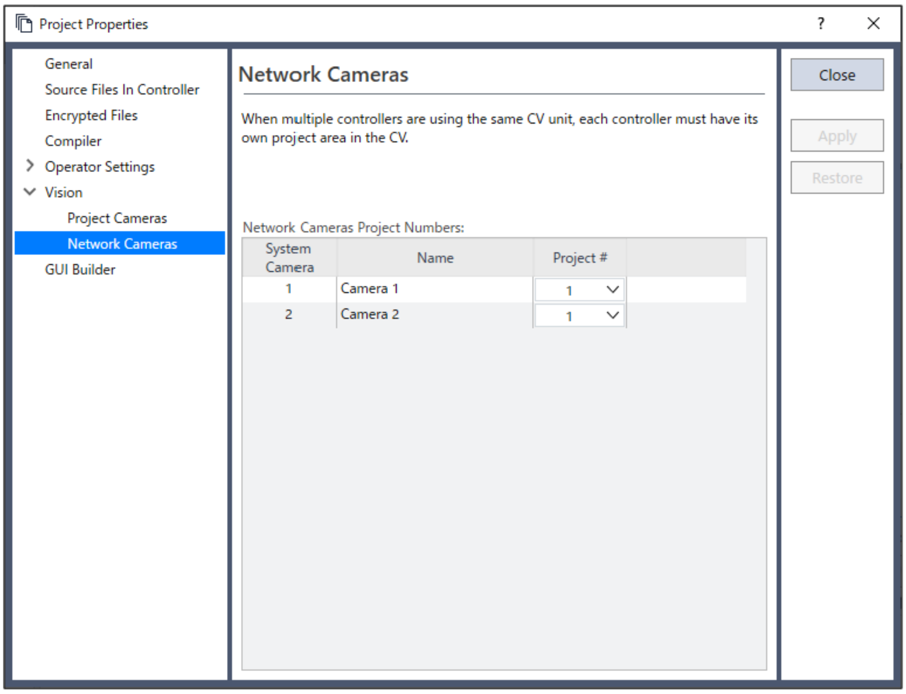
6.10.16.12 [Project]-[Properties]-[Vision]-[Network Cameras]

The Compact Vision systems can manage two vision projects simultaneously.

Each vision project can be used by one Controller, so two Controllers can use the same Compact Vision unit.

In this page, you can configure the vision project number of the Compact Vision cameras used for this project.

"Project 1" is used by default.



Item	Description
System Camera	Select the system camera to use for the project camera.

Item	Description
Project # (Network Camera Project Numbers)	Select the vision project number.

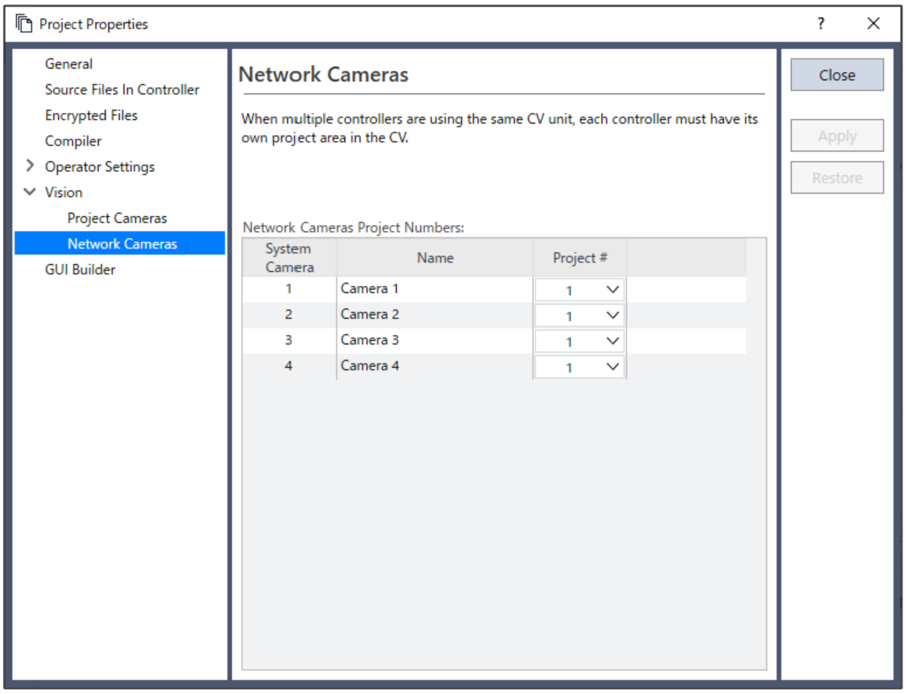
Setting of a network camera project

For example, let's assume there are two controllers, two projects, and a CV. Four cameras are connected to the CV and they are set as four system cameras to RC+ as shown in the table below.

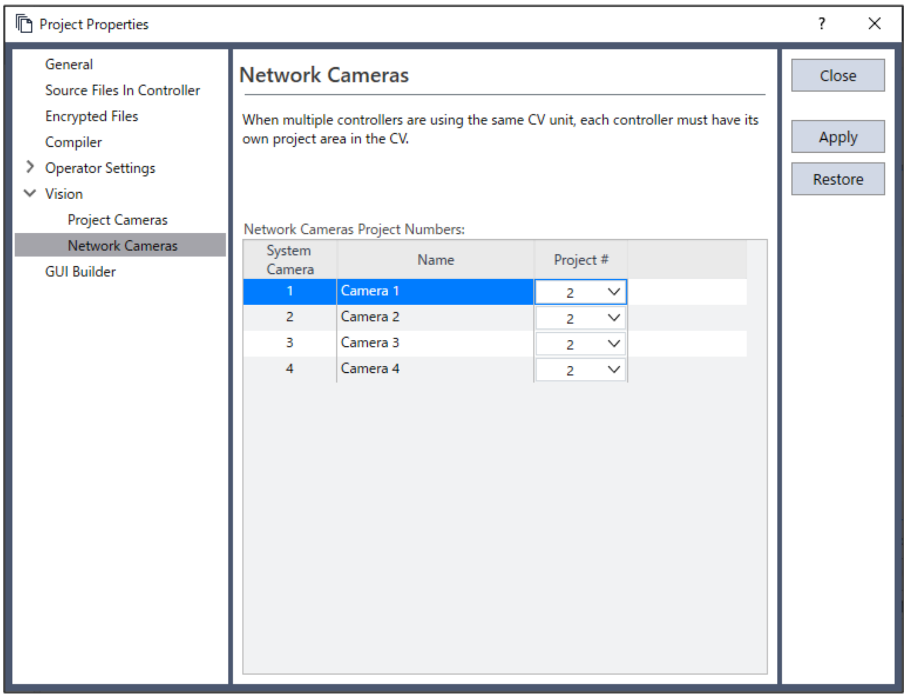
System camera	Model
1	NS1044BU
2	NS4133CU
3	acA1600-60gm
4	acA2500-14gm

The two projects controlled by the compact vision must use different cameras. In here, the project A uses the system camera 1 and 2, and the project B uses the system camera 3 and 4.

To set network camera project for the project A, select [Project]-[Properties]-[Vision]-[Network Cameras] with project A open. Set Project # for the system cameras 1 and 2 that is used in this project to 1 as shown below. (Project # for the system cameras 3 and 4 is set to 1, however, Project # for a camera that is not used in a project is optional.)



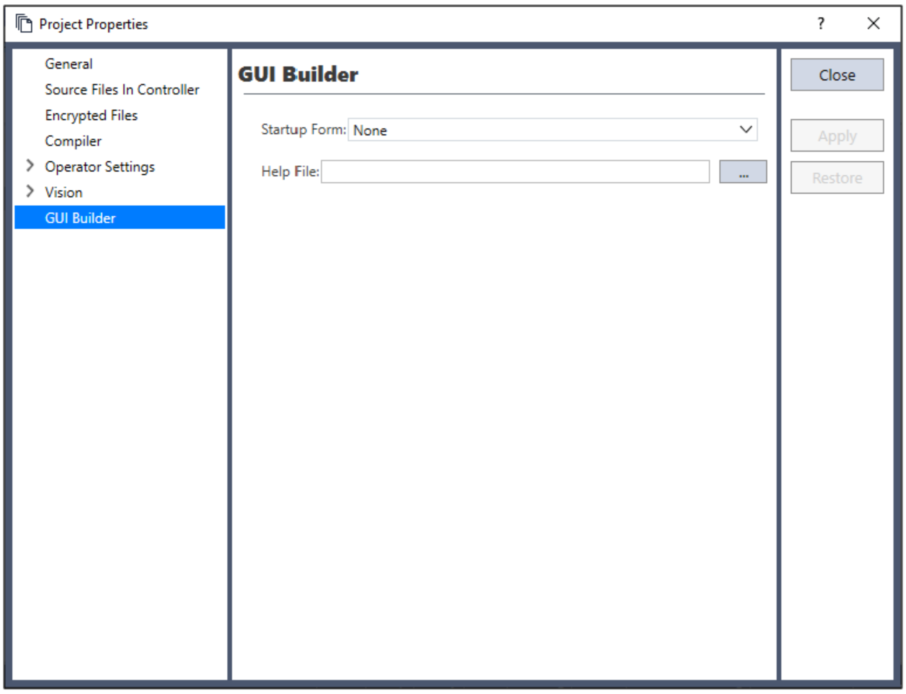
To set network camera project for the project B, select [Project]-[Properties]-[Vision]-[Network Cameras] with project B open. Set Project # for the system cameras 3 and 4 that is used in this project to 2 as shown below. (Project # for the system cameras 1 and 2 is set to 2, however, Project # for a camera that is not used in a project is optional.)



Now, the two projects; A and B can be used by the CV simultaneously .

6.10.16.13 [Project]-[Properties]-[GUI Builder]

On this page, you can specify the startup form for GUI Builder and also set the value of the help file used in your project.



Item	Description
Startup Form	Select the startup form for the current project. If no forms have been created in GUI Builder, then there will be no forms in the list.
Help File	Set help file that will be used by forms in GUI Builder.

6.11 [Run] menu

The Epson RC+ 8.0 [Run] menu includes commands for running and debugging programs.

6.11.1 [Run Window] Command (Run Menu)

: F5

Opens the [Run] window to run a program.

Before opening the [Run] window, all files will be saved automatically if there are any unsaved files and then the project will be built. If there are any errors during build, the [Run] window will not be opened.

(If the [Auto file save] preference is off in [Setup]-[Preferences]-[Workspace], you will be prompted to save all files if there are any unsaved files.)

After the [Run] window opens, you must click the [Start] button to initialize program execution.

See details below.

Run window

6.11.2 [Test in Auto Mode] (Run Menu)

Shift+F5

Opens the operator window.

Before running [Test in Auto Mode], all files will be saved automatically if there are any unsaved files and then the project will be built. If there are any errors during build, the [Operator Window] will not be opened.

(If the [Auto file save] preference is off in [Setup]-[Preferences]-[Workspace], you will be prompted to save all files if there are any unsaved files.)

See details below.

Operator Window

6.11.3 [Step Into] Command (Run Menu)

: F11

Execute the current source line. If the current line is a function, the next step will be the first line in the function.

6.11.4 [Step Over] Command (Run Menu)

: F10

Execute the current source line. If the current line is a function, the entire function will be executed.

6.11.5 [Walk] Command (Run Menu)

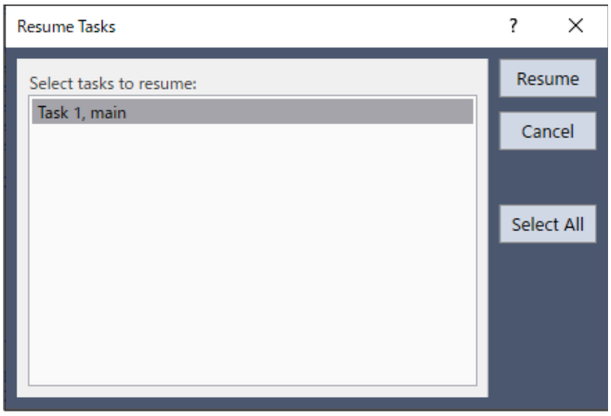
: F12

Execute lines until after the next motion command or output command, and stop at the next line. Configure whether to stop at an output command by checking the [Walk stops for output commands] checkbox in the [Setup]-[System Configuration]-[Controller]-[Preferences] tab.

6.11.6 [Resume] Command (Run Menu)

: F7

Opens the [Resume Tasks] dialog box. Use this command to resume one or more halted tasks. This command is available only when one or more tasks are in halt mode.



Item	Description
Select tasks to resume	A list of all currently halted tasks. Click on one or more tasks to resume.
Resume	Click to resume.
Select All	Click to select all of the tasks in the list.
Cancel	Cancels the operation and closes the dialog.

6.11.7 [Stop] Command (Run Menu)

Stops all tasks. This command is disabled when no tasks are running.

6.11.8 [Toggle Breakpoint] Command (Run Menu)

: F9

Sets the selected line as a breakpoint or returns it to normal.

When a line is a breakpoint, a breakpoint icon is displayed in the program window left margin.

You can set breakpoints while tasks are running.

If a line cannot be a breakpoint (such as a blank line), then the breakpoint icon will not appear for that line.

6.11.9 [Clear All Breakpoints] Command (Run Menu)

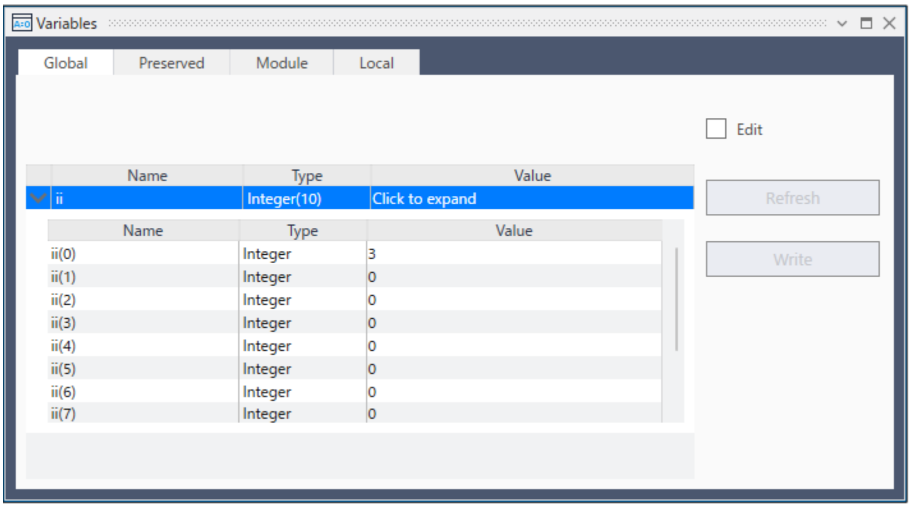
Ctrl+Shift+F9

Clears all breakpoints.

6.11.10 [Display Variables] Command (Run Menu)

: F4

Displays a dialog box that shows the values for all variables in robot controller memory.

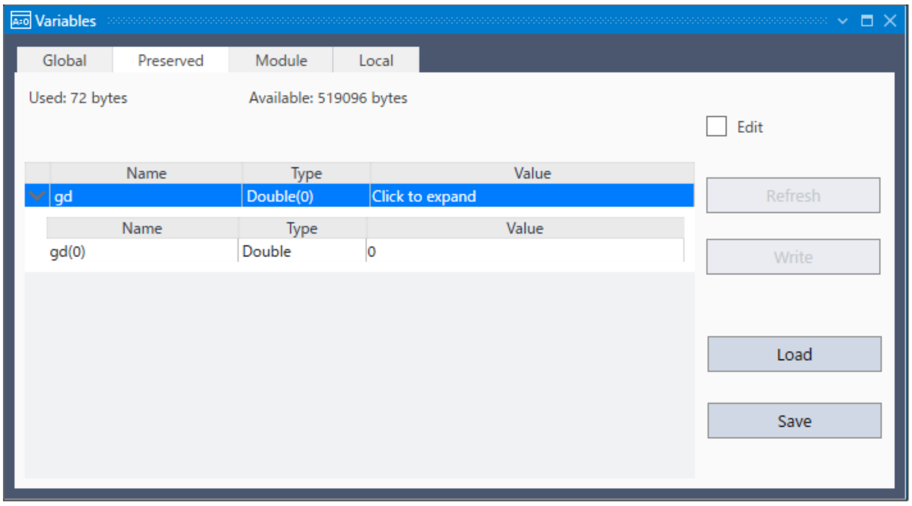


To change a variable value

1. Check the [Edit] checkbox.
2. Enter the new value in the [Value] column. As you type in new values, the text color changes to red, indicating that the value is new and as not been written.
3. Click the [Write] button to save the changes to the controller.

   Click the [Read] button or uncheck [Edit] to cancel changes and restore the previous values.

When an array is displayed, the first element is shown. You can change which element to view by selecting the desired array subscript from the drop-down list and then clicking the [Read] button.



The Preserved page displays the Global Preserve variables. The numbers of used and available bytes for preserved variables are also displayed.

You can save the values of Global Preserve variables in the controller to a file on the PC by clicking the [Save] button. The default file name is "GlobalPreserves.dat".

KEY POINTS

A “GlobalPreserves.dat” file can also be saved from [Controller Backup] in the Epson RC+ 8.0 menu-[Tools]-[Maintenance] dialog.

You can load the global preserve variables that are stored in the file on the PC by clicking the [Load] button.

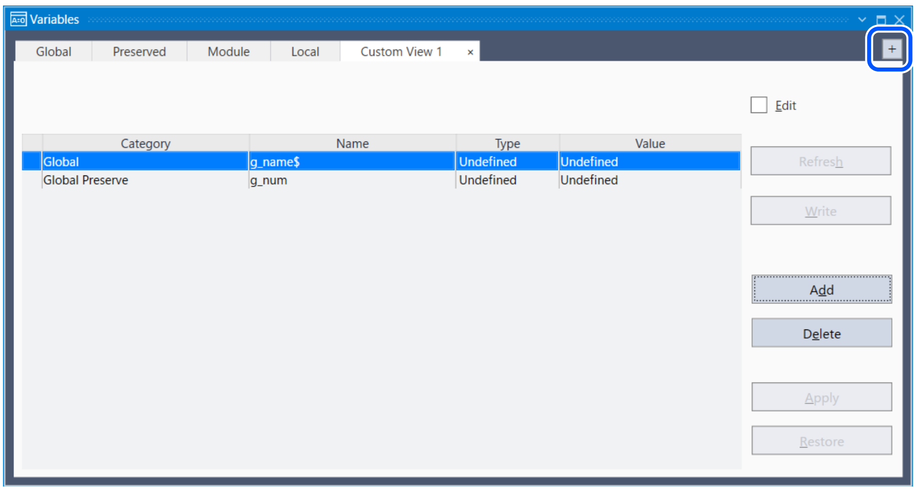
For module variables, you must select the desired program.

Local variables are not displayed unless one or more tasks have reached a breakpoint or have been halted from the Task Manager.

You can view local variables for each function in the call stack for each halted task.

Using Custom View

1. Click the [Custom View] tab. If a [Custom View] tab does not appear, click the [+] button on the top right of the window to select it. You can display up to 2 custom view.



TIP

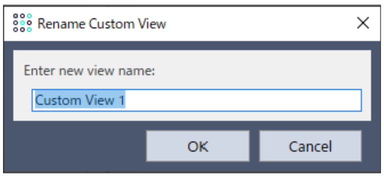
Right-click on a custom view tab to rename the tab.

2. Click the [Add] button to add a new row to the list.
3. Click the category column and select a variable. (You can choose from Global Preserve or Global)
4. Click the variable name and select the variable.
5. Add more rows as needed by repeating steps 2 to 4.
   - [Apply]: Save the changes.
   - [Delete]: Delete a row.
   - [Restore]: Cancel changes.

To rename a view

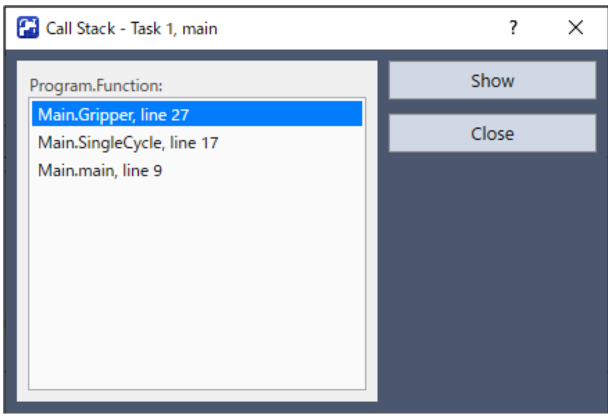
You can rename custom view tabs.

1. Select the [Custom View] tab. If no custom views appear, click the [+] button on the top right of the window to select it.
2. Right click on the view tab and select [Rename].
3. Rename Custom View will be displayed. Type in a new name, and then click [OK].



6.11.11 [Call Stack] Command (Run Menu)

The Call Stack dialog displays the function call stack for one task.



The Call Stack command is available when a program window is clicked which contains a function that is currently halted.

The most recent function is at the top of the list, and parent functions are listed afterwards in descending order. The last function is the task function.

Each row in the list shows a program, function, and line number.

You can view the code for any of the function calls in the list by selecting a function, then clicking the [Show] button. The program window for the function you selected is then displayed, and then move the cursor to the line of the function call.

6.12 [Tools] Menu

Epson RC+ 8.0 has several GUI tools to support the system development. All tools can be accessed from the [Tools] menu. Many also have tool bar buttons and hot keys.

The Tools Menu includes the following selections:

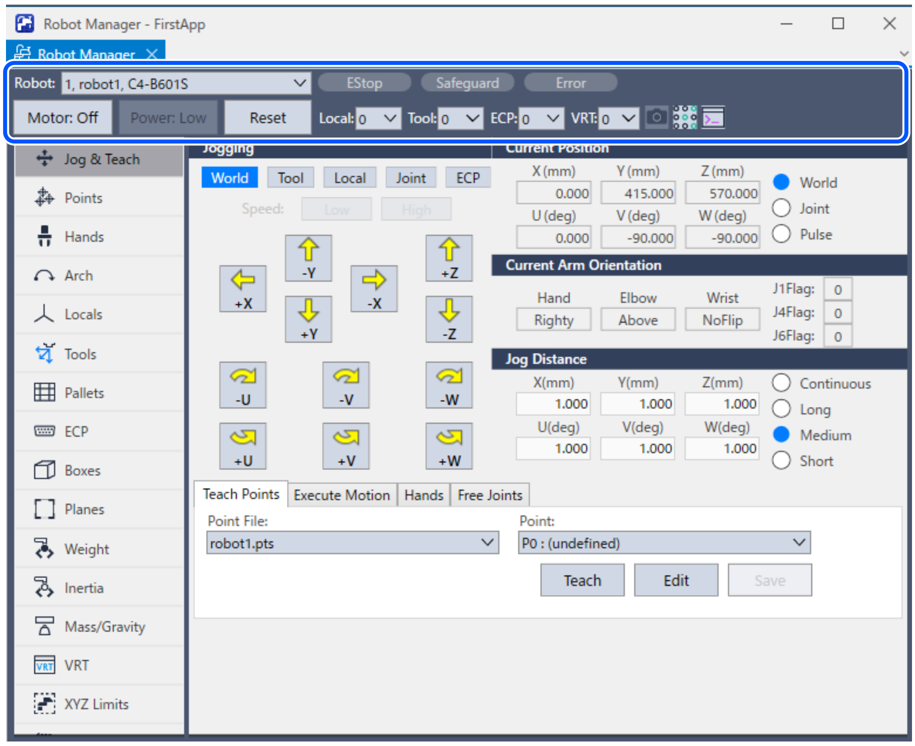
- Robot Manager: Motor control, Jog & Teach, change robot parameters.
- Command Window: Execute SPEL+ commands directly.
- I/O Monitor: Monitor and change I/O status.
- Task Manager: Monitor and control task status.
- Macros: Opens the Macro Window.
- I/O Label Editor: Edit I/O labels.
- User Error Editor: Edit user errors.
- Controller: Do maintenance on the controller, such as backup, restore, and export status.
- Simulator: Check the robot orientation and motion in a 3D display.
- GUI Builder: You can create a GUI (graphical user interface). This product is a paid add-on that requires a separate license.
- Conveyor Tracking: This is used for conveyor tracking calibration. This is not supported by virtual controllers. This product is a paid add-on that requires a separate license.
- Part Feeding: This sets and calibrates part feeding parameters. This is not supported by virtual controllers. This product is a paid add-on that requires a separate license.
- Vision: This uses Vision Guide for image processing.
- Force Guide: This uses Force Guide to create a force sensor application.
- Force Monitor: This displays the current force value, and provides analysis and comparisons of historical values.

6.12.1 [Robot Manager] Command (Tools Menu)

: F6

This command opens the [Robot Manager] window. This window contains several tabs that are used to control the robot motors and power, jog the robot and teach points, and view/edit several parameters for the robot.

The [Robot Manager] window contains buttons for basic robot operations, such as turning motors on/off. It shows the emergency stop, safeguard, and error status.



Item	Description
Robot	Select a robot.
Emergency Stop	Indicates if Emergency Stop has occurred. The display will show red if an Emergency Stop has occurred. To clear the Emergency Stop status, click the [Reset] button.
Safeguard	Indicates whether the Safeguard input is on or off. When on, the display is yellow.
Error	Indicates whether an error status has occurred. When an error occurs, the display is red.
Motor: Off / On	Turns all robot motors on or off.
Power: Low / High	Puts the robot servo system in high power mode or low power mode.
Reset	Resets the robot servo system and Emergency Stop condition.
Local	This drop down list is used to select the local coordinate system for jogging and teaching.
Tool	This drop down list is used to select the tool for jogging and teaching.
Arm	This drop down list is used to select the arm for jogging and teaching. Vertical 6-axis robot (including the N series) are not shown.
ECP	This drop down list is used to select the ECP for jogging.
VRT	This drop down list is used to select the number set for VRT parameters.

When you switch to the [Robot Manager] window, the robot's speed setting will be set to the speed (high, low) on the Jog & Teach window.

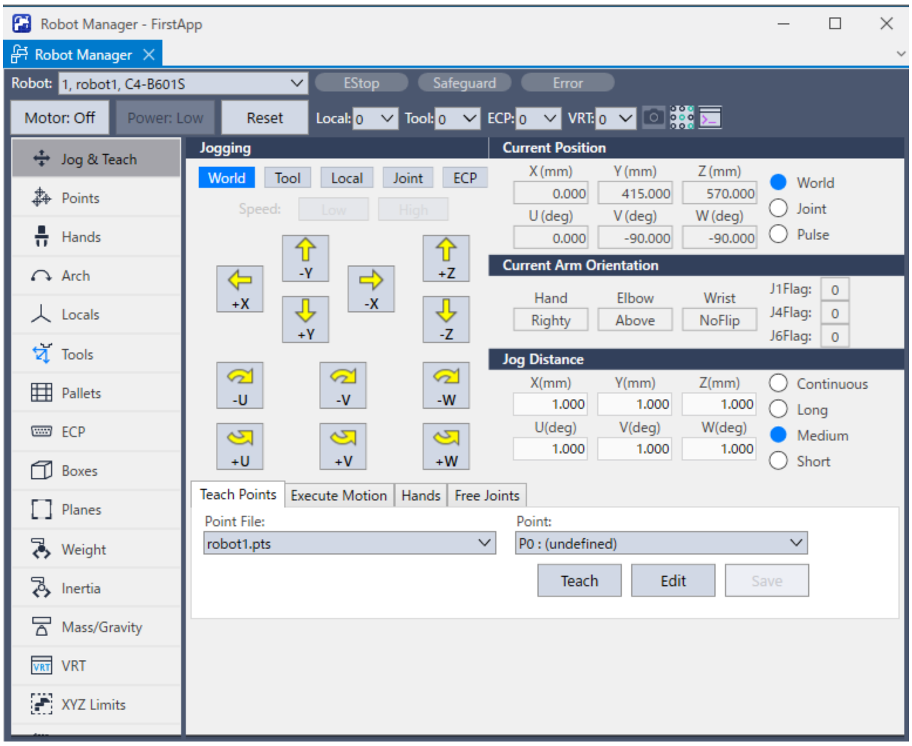
Motion command after the above operation will be executed at this speed. Set the speed again by the commands such as Motor, Speed, and Accel.

> ✎ KEY POINTS
>
> The order of the Robot Manager panels can be changed from [Setup] - [Preferences] - [Robot Manager].
>
> See details below.
>
> [Setup]-[Preferences]-[Robot Manager]

6.12.1.1 [Tools]-[Robot Manager]-[Jog & Teach] Panel

The [Jog & Teach] page is primarily used for jogging the robot to a desired position and teaching a point using the current coordinates and orientation.

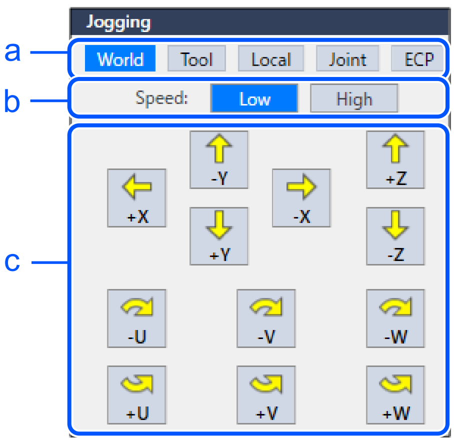
You can jog the robot in World, Tool, Local, Joint, or ECP modes. You can also execute motion commands.



The [Robot Manager]-[Jog & Teach] page contains several controls, described below.

[Jogging] Group

This group contains controls for setting jog mode, speed, and jog buttons.



Symbol	Item
a	Jog Mode
b	Speed
c	Jog Buttons

Jog Mode

This dropdown list contains the following choices jog mode.

- World: Jogs the robot along the X, Y, Z axes in the current local, tool, arm, and ECP.
  - For robots with 4 DOF (Cartesian coordinate or SCARA), you can also jog U (roll).
  - For robots with 6 DOF (vertical 6-axis (including N series)), you can jog U (Z axis rotation of the base coordinate system), V (Y axis rotation of the base coordinate system), and W (X axis rotation of the base coordinate system). When the [Jog & Teach] panel is displayed, Jog & Teach is set to "World".
- Tool: Jogs the robot in the coordinate system defined by the current tool.
- Local: Jogs the robot in the coordinate system defined by the current local.
- Joint: Jogs each joint of the robot. A separate set of jog buttons will appear when using joint mode when using non-Cartesian robots.
- ECP: Jogs the robot along the axes of the coordinate system defined by the current external control point. Coordinates are World coordinates.

Speed

The speed for jogging and motion commands can be changed by selecting Low or High. When starting RC+, the speed is set to "Low speed" when the [Robot Manager] window is displayed. Jogging is always in low power mode. The speeds and accelerations associated with the jog speed settings are shown below.

SCARA robot RS series

Speed	Jog Method	Speed	Accel	Decel
Low	Continuous World/Tool/ECP XYZ	10 mm/sec	100 mm/sec^2	200 mm/sec^2
	Continuous World/Tool/ECP UVW	2 deg/sec	20 deg/sec^2	40 deg/sec^2
	Continuous Joint	*	10 deg/sec^2	20 deg/sec^2
	Step	1/5 of default PTP speed	Default PTP acceleration	Default PTP deceleration
High	Continuous World/Tool/ECP XYZ	50 mm/sec	100 mm/sec^2	200 mm/sec^2
	Continuous World/Tool/ECP UVW	10 deg/sec	20 deg/sec^2	40 deg/sec^2
	Continuous Joint	*	10 deg/sec^2	20 deg/sec^2
	Step	Default PTP speed	Default PTP acceleration	Default PTP deceleration

* Speed of Continuous Joint depends on the robot model

Vertical 6-axis robot, N series

Speed	Jog Method	Speed	Accel	Decel
Low	Continuous World/Tool/ECP XYZ	10 mm/sec	200 mm/sec^2	400 mm/sec^2
	Continuous World/Tool/ECP UVW	2 deg/sec	20 deg/sec^2	40 deg/sec^2
	Continuous Joint	*	20 deg/sec^2	40 deg/sec^2
	Step	1/5 of default PTP speed	Default PTP acceleration	Default PTP deceleration
High	Continuous World/Tool/ECP XYZ	*	200 mm/sec^2	400 mm/sec^2
	Continuous World/Tool/ECP UVW	15 deg/sec	20 deg/sec^2	40 deg/sec^2
	Continuous Joint	*	20 deg/sec^2	40 deg/sec^2
	Step	Default PTP speed	Default PTP acceleration	Default PTP deceleration

* Speed of Continuous Joint and High speed Continuous XYZ depends on the robot model.

Jog Buttons

Use the jog buttons to jog the robot throughout the work envelope.

They can be controlled only by the mouse.

The robot jogs one step at a time as you click the button in either "Long", "Medium", or "Short" mode of the Jog Distance. The robot jogs continuously by holding the button down. To jog continuously without stepping, set the Jog Distance to Continuous. For details, refer to How to jog robot.

You can change the orientation of the jog buttons to align your PC monitor with the robot from the Epson RC+ 8.0 menu-[Setup]-[Preferences]-[Jog & Teach].

The jog buttons are displayed differently depending on the Jog mode.

- For World, Local, Tool, and ECP jogging, the X, Y, Z, U, V, W buttons appear. (V and W are only for 6-axis robots (including N series))
- For Joint jogging, the joint buttons labeled J1 - J6 appear.
- The X, Y, and Z buttons jog the robot in the Cartesian axis.
- The U buttons rotate the Tool coordinate system of the Z axis. (roll)
- For 6-axis robots (including N series), the V buttons rotate the Tool coordinate system of the Y axis. (pitch)
- The W buttons rotate the Tool coordinate system of the X axis. (yaw)

- Local

  In the [Local] box, select the local coordinate system for jogging or teaching.

  Only Locals that have been defined are shown in the list. When you teach a point, the Local point attribute defaults to the current local number.

- Tool

  In the [Tool] box, select the tool for jogging or teaching. Only Tools that have been defined are shown in the list.

- Arm

  In the [Arm] box, select the arm for jogging or teaching.

  Only Arms that have been defined are shown in the list.

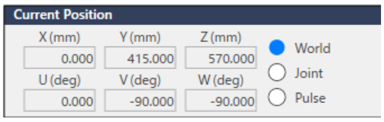
> **KEY POINTS**
>
> Arms are not used with 6-axis robots (including the N series). Arms cannot be change from the [Arm] box with a PG axis.

- ECP

  In the [ECP] box, select the ECP to perform jog operation.

  Only ECPs that have been defined are shown in the list. ECPs are only allowed if the External Control Point option has been activated.

[Current Position] Group



This group displays the current position of the robot. There are three ways to display position.

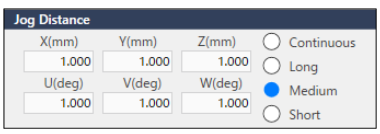
- World: Displays the current position and tool orientation in the selected local coordinate system
- Joint: Displays the current joint values
- Pulse: Displays the current encoder pulse count for each joint

[Current Arm Orientation] Group

This group displays the current arm orientation.

- 6-axis robot: Hand orientation, Elbow orientation, wrist orientation, J1Flag value, J4Flag value, J6Flag value
- N: Hand orientation, Elbow orientation, wrist orientation, J4Flag value, J6Flag value
- RS series: Hand orientation, J1Flag value, J2Flag value
- Others: Hand orientation

[Jog Distance] Group



There are radio buttons for selecting Continuous, Long, Medium, and Short jog distances.

When "Continuous" is selected, the robot operates in continuous jog mode and the [Jog Distance] box is grayed out and cannot be selected. If anything other than "Continuous" is selected, the distance specified in the [Jog Distance] box is considered as one step (step mode).

To change a jog distance, first select the distance to be changed, then type in the new value.

Distance	Set Value *	Default Value
Short	More than 0 to 10	0.1
Medium	More than 0 to 30	1
Long	More than 0 to 180	10

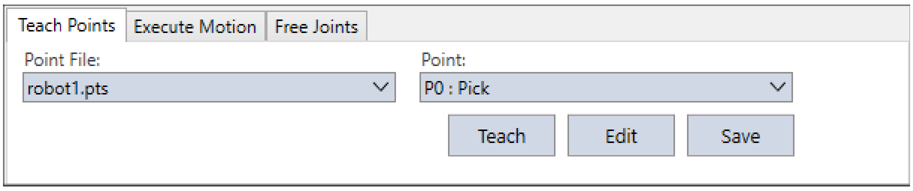
* If you enter a too large value, an error message appears when you attempt to jog.

When the jog mode is changed, the jog distance units change appropriately between millimeters (mm) and degrees (deg).

KEY POINTS

When the jog distance is longer than the default, jog distance is reset to default status by rebooting Epson RC+.

[Teach Points] Tab

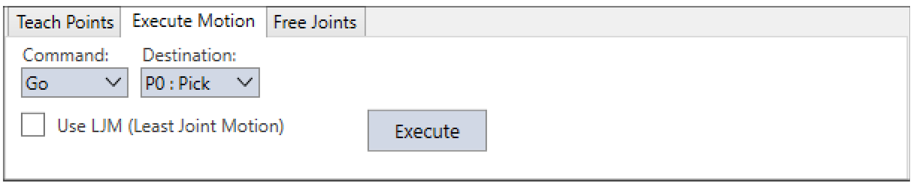


Select the [Teach] tab. This tab shows the current point file name and point number.

- Use the [Teach] button to register the current robot position.
- Use the [Edit] button to select and view the current point in the [Points] tab.
- Click the [Save] button to save teach point data.

See How to teach points for more information.

[Execute Motion] Group



Select the [Execute Motion] tab.

This executes motion commands. You can execute commands such as Go, Move, Jump, Arc, Home, and Align (6-axis robots only).

- Click the [Execute] button from this group to execute the motion.
- When [USE LJM (Least Joint Motion)] checkbox is checked, posture of the manipulator is automatically adjusted to reduce the motion distance. The default setting is unchecked.

In the Epson RC+ 8.0 menu, clear the [Setup]-[Preferences]-[Jog & Teach]-[Enable motion commands] checkbox to disable motion command execution.

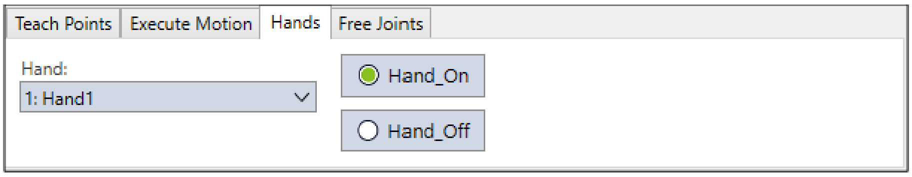
[Free Joints] Group

You can free one or more joints using the checkboxes. PG axis not available for 6-axis robots (including the N series).

- Click the [Free All] button to free all joints from the servo control.
- Click the [Lock All] button to lock all joints under the servo control.

[Hands] group

When hands are set, the [Hands] tab is displayed.



Item	Description
Hand:	Select the hand to operate from pull-down menu. In the pull-down menu, the registered hands are displayed for the robot selected in [Robot:] on the upper left of the [Robot Manager] window.
Hand_On button	When this button is pressed, the Hand_On command is immediately executed for the hand selected above. Furthermore, when the return value of the Hand_On function is retrieved and the result is True, the LED display on the left of the button lights up. ▪ : The return value of Hand_On function is True ▪ : The return value of Hand_On function is False
Hand_Off button	When this button is pressed, the Hand_Off command is immediately executed for the hand selected above. Furthermore, when the return value of the Hand_Off function is retrieved and the result is True, the LED display on the left of the button lights up. ▪ : The return value of Hand_Off function is True ▪ : The return value of Hand_Off function is False

For details of hand settings, refer to the following manual:.

"Hand Function Manual"

How to jog

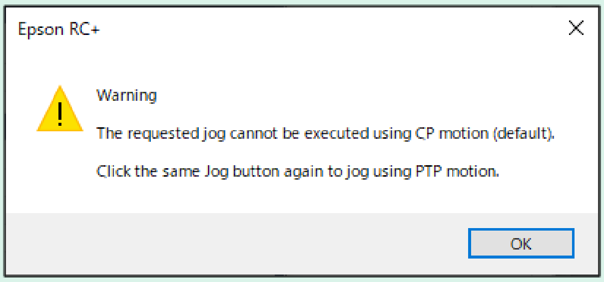
In the upper left hand corner of the [Jog & Teach] page, you will see a control group called Jogging that contains jog buttons. In the World, Local, Tool, and ECP jog modes, the robot is jogged in the Cartesian coordinate system (X, Y, Z). In the Joint jog mode, each robot joint can be jogged separately.

The jog speed is determined by the Speed setting.

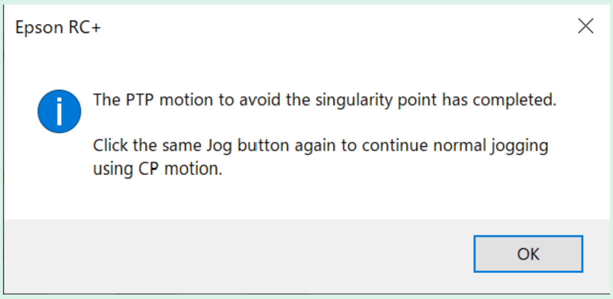
- In step mode, each time you click a jog button, the robot moves along the appropriate axis by the amount specified in the [Jog Distance] control group.
- If "Continuous" is selected in the [Jog Distance] group, the movement continues using linear interpolation while the jog key is held down. This is called continuous jog operation.

KEY POINTS

- For robots other than the 6-axis robots, the jog motion in step mode is PTP (point to point) motion. It is difficult to predict exact jog motion trajectory in the PTP motion. Therefore, be careful that the robot doesn't collide with peripheral equipment and that the robot arms don't collide with the robot itself during jogging.
- For the 6-axis robots, the jog motion is CP (Continuous Path) motion. Note that when jogging near the singularity, if you try to pass through the singularity, a warning dialog below will appear.

  Click the [OK] button and click the same Jog button again to jog using PTP motion and pass the singularity. It is difficult to predict exact jog motion trajectory in the PTP motion. Therefore, be careful that the robot doesn't collide with peripheral equipment and that the robot arms don't collide with the robot itself during jogging. Also, if you attempt the other jogs or operations, it cancels the switching to PTP motion. So when jogging near the singularity again, the same warning dialog will appear.
- If passing the singularity in the continuous jog motion, the following warning message will appear.

  When jogging in continuous mode, if an out of range condition occurs, the robot motors will turn off and an error will be displayed. In this case you must execute a Reset and Motor On from the Control Panel page to continue the jog.

To jog

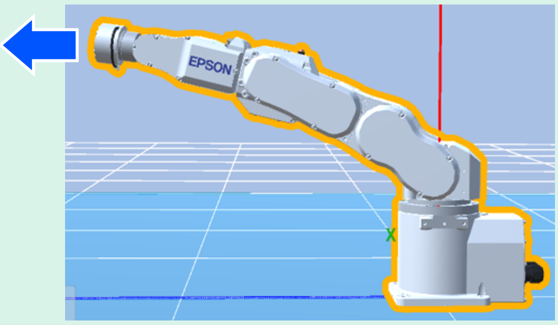
1. Select the jog mode: World, Tool, Local, Joint, or ECP.
2. Select the jog speed: “Low” or “High”.
3. Select the jog distance (Continuous, Long, Medium, or Short) in the [Jog Distance] group. You can type in the desired jog distance when "Continuous" is not selected.
4. Click on one of the jog buttons with the mouse button. If you hold the mouse button down, the robot will continue to jog.

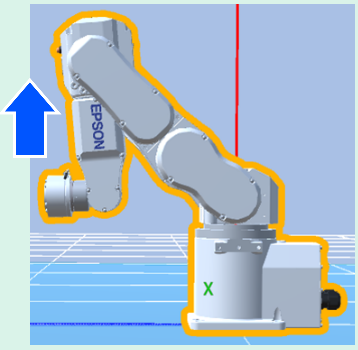
   When jogging is started, the jog button color changes from yellow to cyan. After jogging is completed, the jog button color returns to yellow.

   If you click any jog button during a step jog, the robot will stop.

KEY POINTS

- You can change the orientation of the jog buttons to in the Epson RC+ 8.0 menu-[Setup]-[Preferences]-[Jog & Teach]. This allows you to align the orientation of the jog buttons with the orientation of the robot motion.
- As shown in the illustrations below, when the robot reaches to the limit of the motion range during Continuous Jog motion, the robot stops before the limit of the motion range. Use Step Jog if you want to move the robot to the limit of the motion range. The robot motion stops when the following conditions are satisfied.
    - When the robot's current position becomes "approx. 5 mm or less from the limit of the motion range".
    - When the robot operates Continuous Jog motion in the direction reaching to the limit of motion range as shown in the illustrations below.





Jogging in Teach Mode

You can jog and move the robot at slow speed with the safeguard open by using the Teach Pendant. For more details, refer to the following manual:

- "Robot Controller option Teach Pendant TP1 manual"
- "Robot Controller option Teach Pendant TP2 manual"
- "Robot Controller option Teach Pendant TP3 manual"
- "Robot Controller option Teach Pendant TP4 manual"

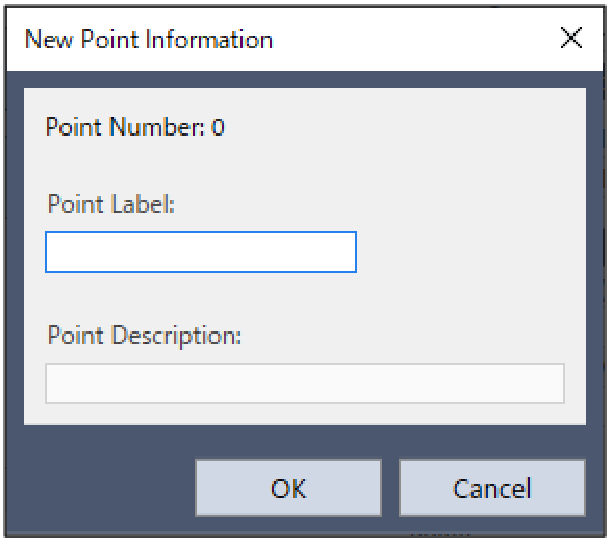
How to teach points

To move the robot to the target point, the point data indicating the robot position is necessary.

Follow these steps to teach points from the [Robot Manager] :

1. Select the point file you are teaching points for from the [Point File] dropdown list box on the [Teach] page.
2. Select the point number you want to teach in the [Points] box.
3. Jog the robot to the desired position. Alternatively, free the robot joints in the [Free Joints] tab, and manually move the robot into position. (Direct teaching)
4. Click the [Teach] button. This will save the robot's current position data.

   In the Epson RC+ 8.0 menu, if the [Setup]-[Preferences]-[Jog & Teach]-[Prompt for new point information] check box is selected, you can enter point labels and comments. Point labels can include up to 32 alphanumeric characters and underscores. Only alphabets can be used for the first letter. Characters can be upper case or lower case.



KEY POINTS

As an alternative to clicking the [Teach] button, on the [Points] tab you can type in the coordinates of the point.

Saving your work

You can save your work in the following three ways.

- In the [Teach...] tab, click the [Save] button.
- In the Epson RC+ 8.0, click [File]-[Save], or click the [Save] on the toolbar.
- In the Epson RC+ 8.0 menu, execute [Project]-[Save], or click the [Save all files] button on the toolbar.

When you want to restore the data without saving the point files, select [Restore] from the [File] menu.

When you close the [Robot Manager], you will be prompted if you want to save your changes. Click the [Yes] button to save changes. Click the [No] button to discard changes without saving.

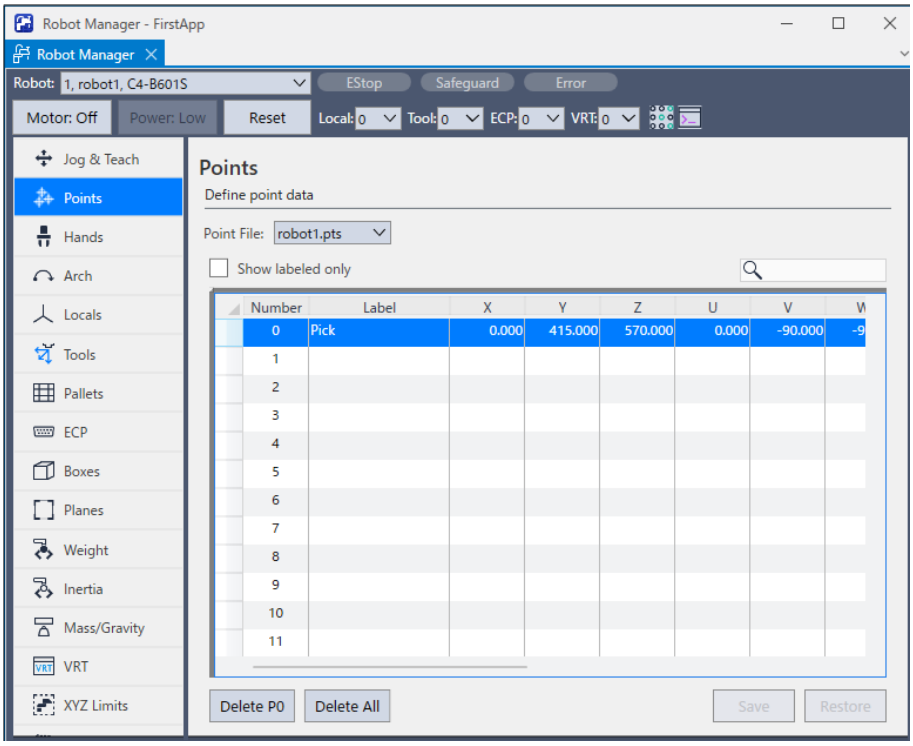
6.12.1.2 [Tools]-[Robot Manager]-[Points] Page

You can input/delete the point data.

When a point file is selected, the robot controller loads the file into memory.

As points are taught on the [Robot Manager]-[Jog & Teach] page, the spreadsheet on the Points page is updated.

For details on how to save points, refer to Saving your work.



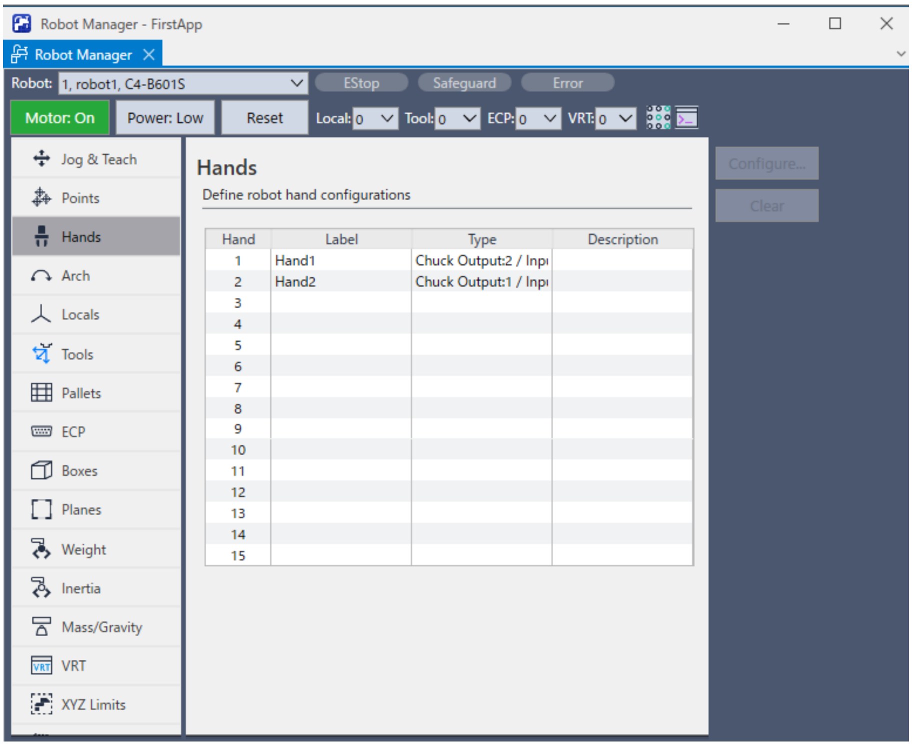
Navigating the grid

Press the [Tab] key to move to the next cell. Press the [Shift] + [Tab] key to move to the previous cell. Press the arrow [↑] or [↓] key to move to the cell above or below.

Item	Description
Point File	Select a point file.
Show only registered	Only shows registered point data.
	Find label.
Delete Pxxx	Deletes the selected point. You will be prompted to confirm the operation.
Delete All	Deletes all points in the file. You will be prompted to confirm the operation.
Save	Saves the current values.
Restore	Reverts to the previous values. You will be prompted to confirm the operation.

6.12.1.3 [Tools]-[Robot Manager]-[Hands] Page

You can check a list of registered hands, register new hands, and change registration information.

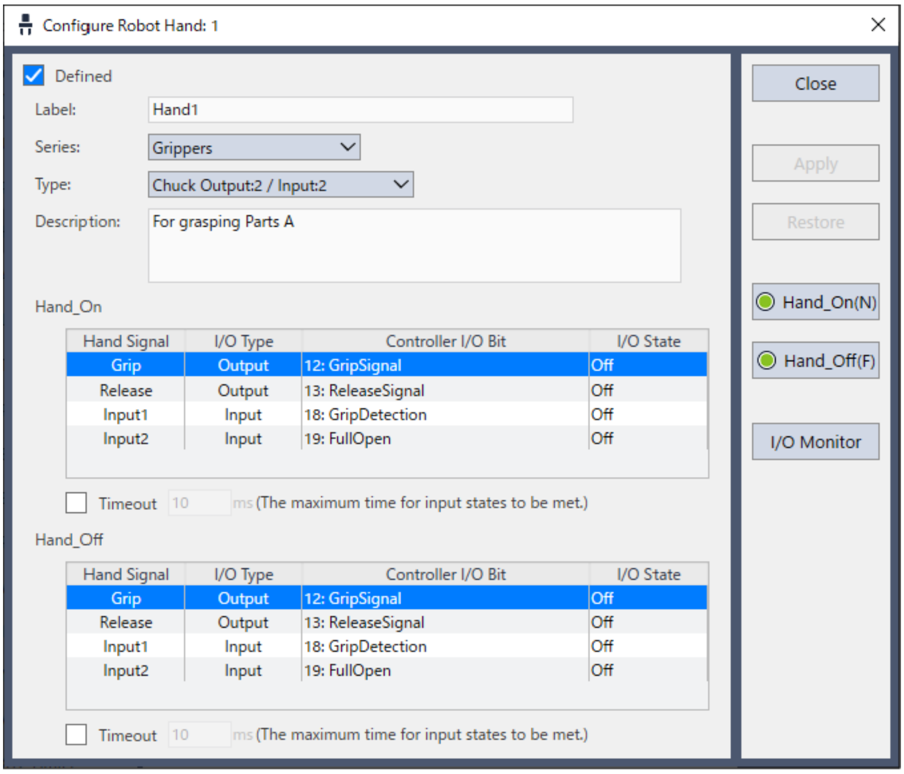


Item	Description
Hand	The number of hand. Up to 15 hands can be set for each robot 1 to 4.
Label	Indicates the label name given to the hand number.
Type	Indicates the type of hand.
Description	A description of the hand.
Configure	Select one hand and click this button to display the [Configure Robot Hand: *] window where you can register a new hand and change or delete the registration information.
Clear	When you select a registered hand and press this button, the hand deletion confirmation dialog will be displayed. If you select the [Yes] button here, the registered hand information will be deleted.

Select one hand from hands 1 to 15 and press the [Configure...] button to display the [Configure Robot Hand:] screen.

For more details on hand settings, refer to the following manual:

"Hand Function Manual"

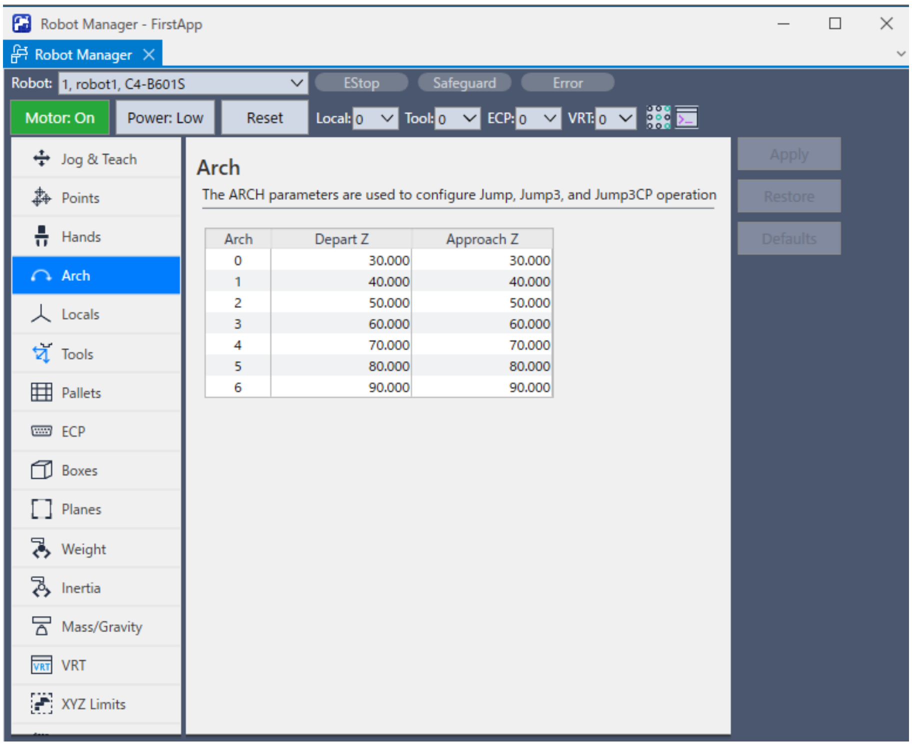


6.12.1.4 [Tools]-[Robot Manager]-[Arch] Page

This page allows you to configure the depart Z and approach Z settings in the robot's Arch table. Arch is used for the Jump, Jump3, and Jump3CP motion commands.

For more details on Arch settings, refer to the following manual:

"SPEL+ Language Reference - Arch Statement"



Navigating the grid

Press the [Tab] key to move to the next cell. Press the [Shift] + [Tab] key to move to the previous cell. Press the arrow [↑] or [↓] key to move to the cell above or below.

Item	Description
Arch	The Arch number. There are up to seven different setting pairs in the Arch table.
Depart Z	Specifies the vertical rise distance of the arch motion.
Approach Z	Specifies the vertical descent distance of the arch motion.
Apply	Sets the current values.
Restore	Reverts to the previous values.
Defaults	Displays factory default settings.

To change Arch settings

1. Put the cursor in the Depart Z or Approach Z cell of the row you want to change.
2. Type in the new value.

6.12.1.5 [Tools]-[Robot Manager]-[Locals] Page

This page allows you to define local coordinate systems for a robot. When the tab is selected, the values of all local tools you can define are displayed. A grid is used to display all of the values for the locals you can define. Local "0" is the base coordinate system and cannot be changed from this page.

To change the base coordinate system, use the Base command from the command window. For more details, refer to the following manual:

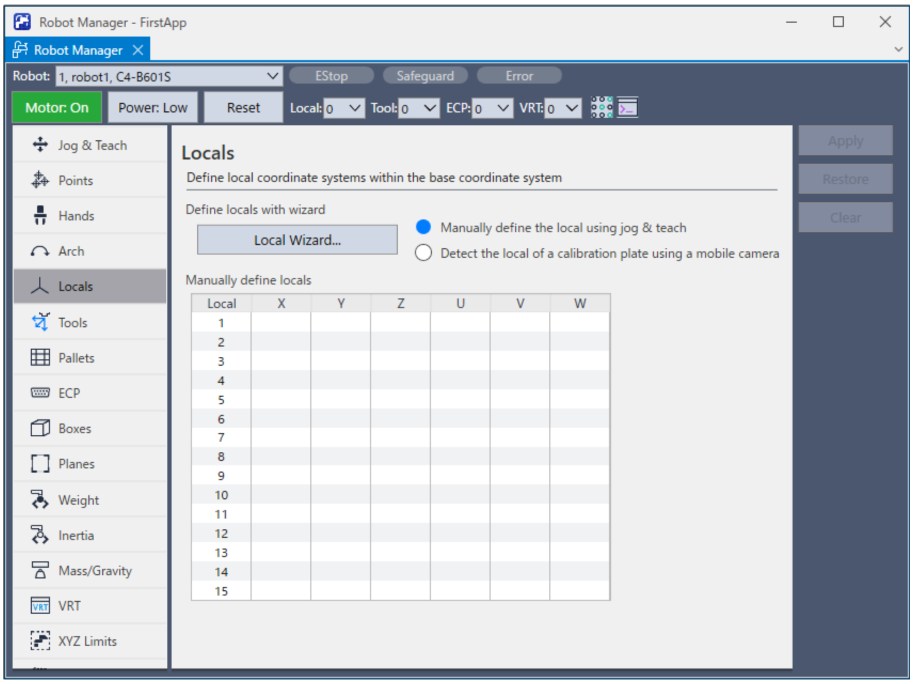
"SPEL+ Language Reference"

When a local is undefined, then all fields for that local will be blank. When you enter a value in any of the fields for an undefined local, then the remaining fields will be set to zero.

Click the [Apply] button to define the local coordinate system.

For more details on local settings, refer to the following manual:

"SPEL+ Language Reference - Local Statement"



Navigating the grid

Press the [Tab] key to move to the next cell. Press the [Shift] + [Tab] key to move to the previous cell. Press the arrow [↑] or [↓] key to move to the cell above or below.

Item	Description
Local Wizard	Click this button to start the Local Wizard. Follow the instructions for each step to define a local. See details in the next section.
X	The X coordinate of the local origin in the base coordinate system.
Y	The Y coordinate of the local origin in the base coordinate system.
Z	The Z coordinate of the local origin in the base coordinate system.
U	Rotation angle of the local about the base Z axis. (roll)
V	Rotation angle of the local about the base Y axis. (pitch)
W	Rotation angle of the local about the base X axis. (yaw)
Apply	Sets the current values.
Restore	Reverts to the previous values.
Clear	Clears all selected values.

Using the Local Wizard

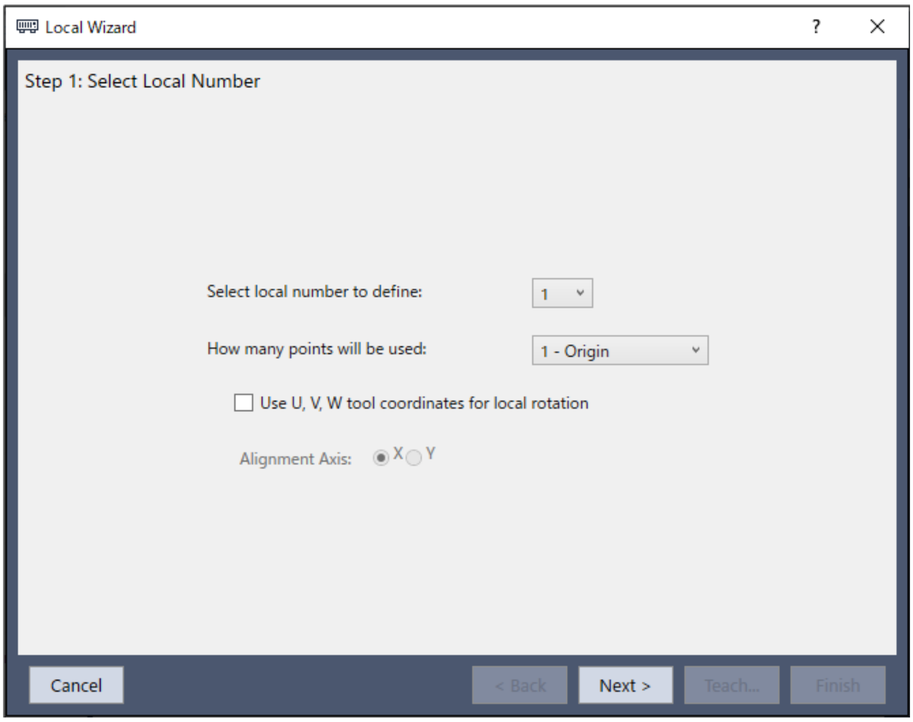
A wizard is provided for defining a local coordinate system. You can define a local using a single point or three points, as described in the following sections. This section describes the procedure used when selecting [Manually define the local using jog & teach]. For details of the Local setting, refer to the following manual:

"Vision Calibration" in "Vision Guide 8.0 Software"

Using the Local Wizard to teach a single point local

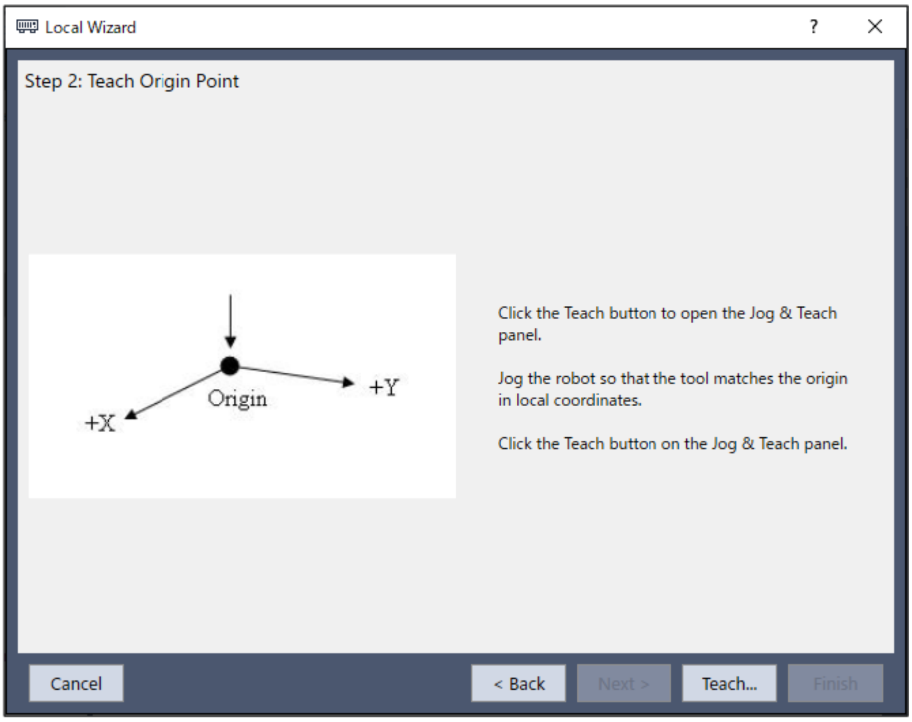
1. Open the [Robot Manager] and click on [Locals] to show the [Locals] page.

2. Select [Manually define the local using jog & teach], and then click the [Local Wizard] button. The dialog shown below appears.



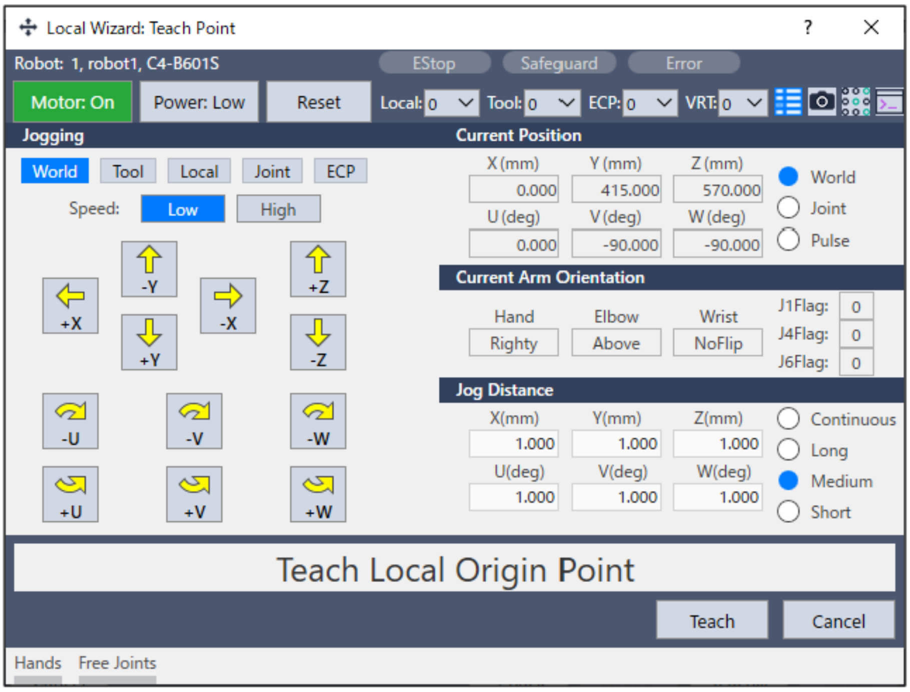
3. Select the local number you want to define.

For [How many points will be used], select [1 – Origin]. Since this is a single point local, you will just teach the origin of the new coordinate system. If you want to use the U, V, or W axes for the orientation of the coordinate system, check the [Use U, V, W tool coordinates for local rotation] checkbox. If this checkbox is unchecked, the new coordinate system is offset from local 0 in X and Y, but is not rotated about any axis. Click the [Next] button.



4. Teach the local origin point.

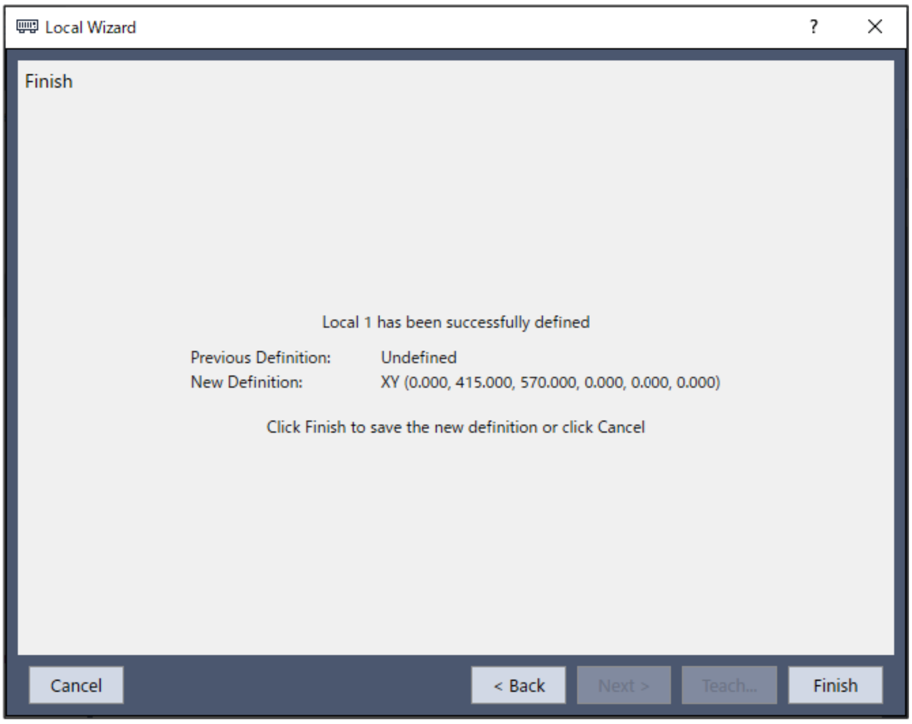
Click the [Teach] button to open the [Local Wizard Teach Point] dialog box.



5. Jog the robot until the end effector is aligned with the local origin point.

6. Click the [Teach] button.

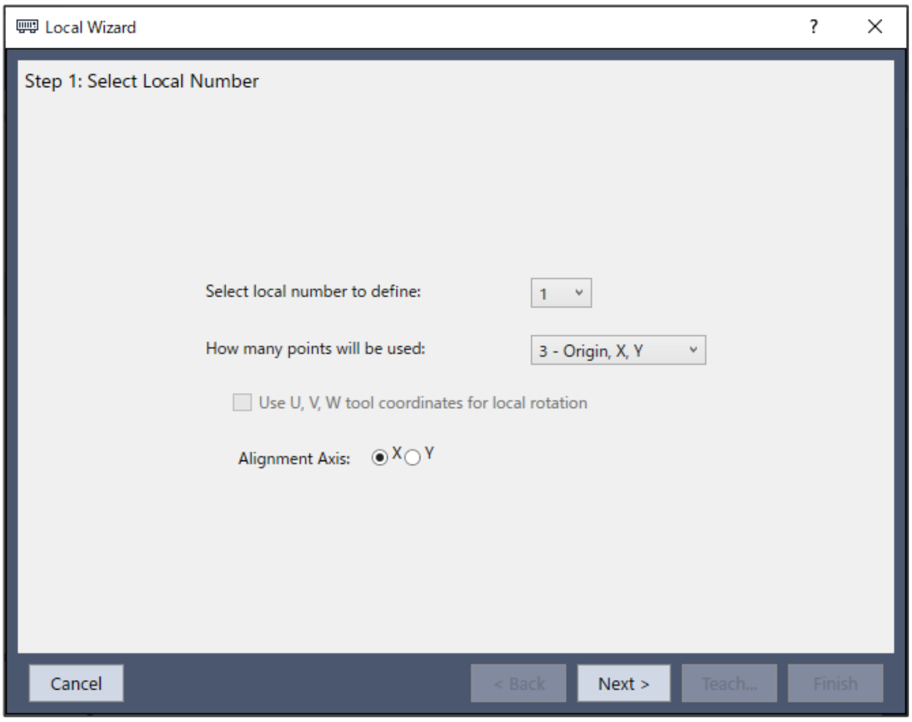
7. The new local definition is displayed as shown below. Click the [Finish] button to accept the new definition.



Using the Local Wizard to teach a three point local

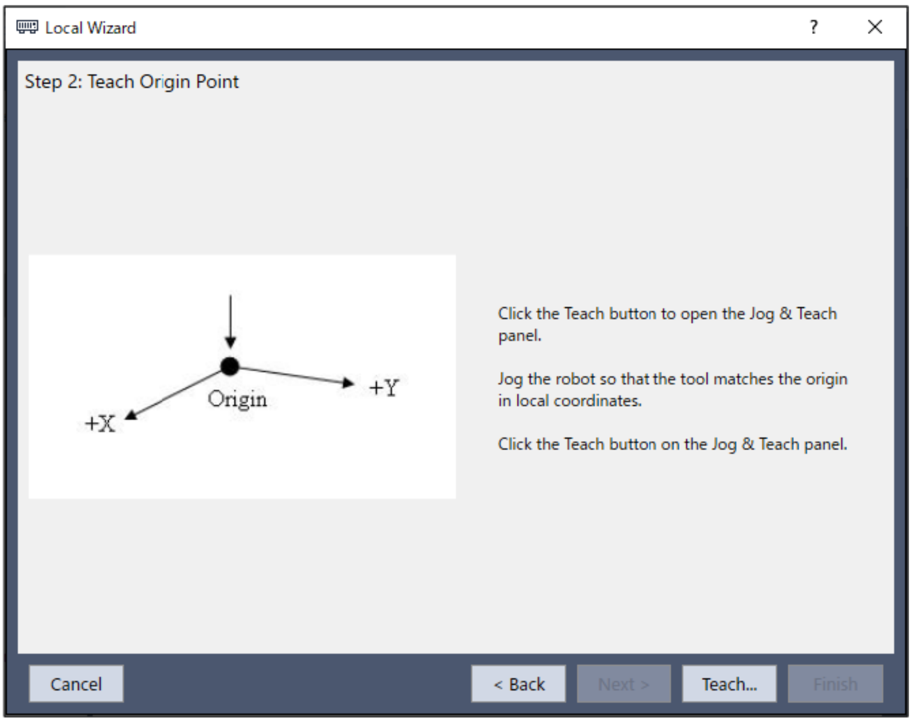
1. Open the [Robot Manager] and click on [Locals] to show the [Locals] page.

2. Select [Manually define the local using jog & teach], and then click the [Local Wizard] button. The dialog shown below appears.



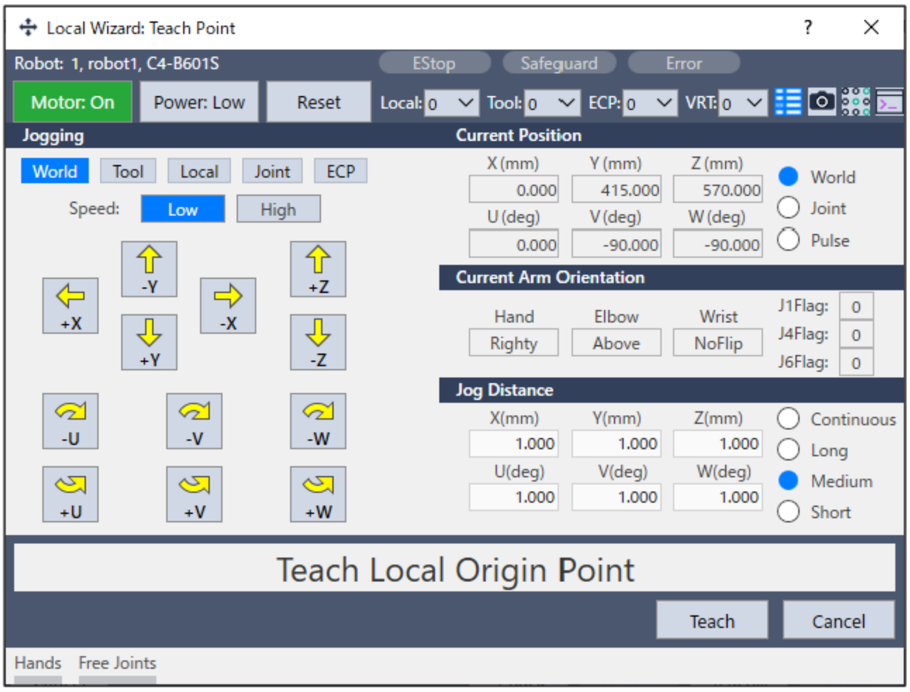
3. Select the local number you want to define.

   For [How many points will be used], select [3 - Origin, X, Y]. Since this is a three point local, you will teach the origin of the new coordinate system, and then teach one point anywhere along the X axis and one point anywhere along the Y axis. Select which axis will be used to align the coordinate system. For example, if you select X, then the new coordinate system X axis will be aligned to the X axis point that you will teach in a later step. The Y axis point will be used to determine tilt. Click the [Next] button.



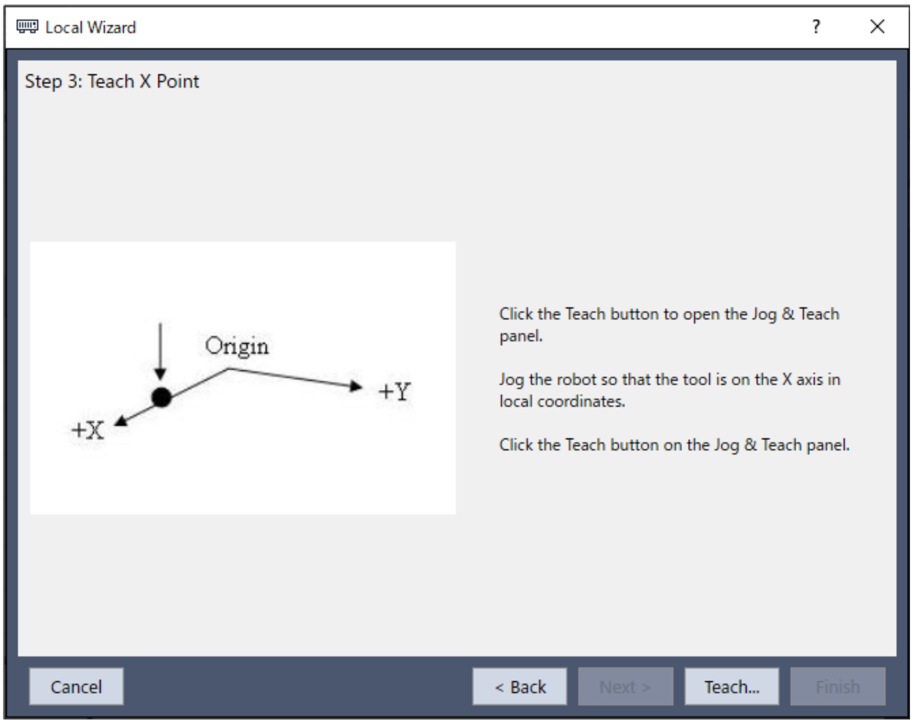
4. Teach the local origin point.

   Click the [Teach] button to open the [Local Wizard Teach Point] dialog box.



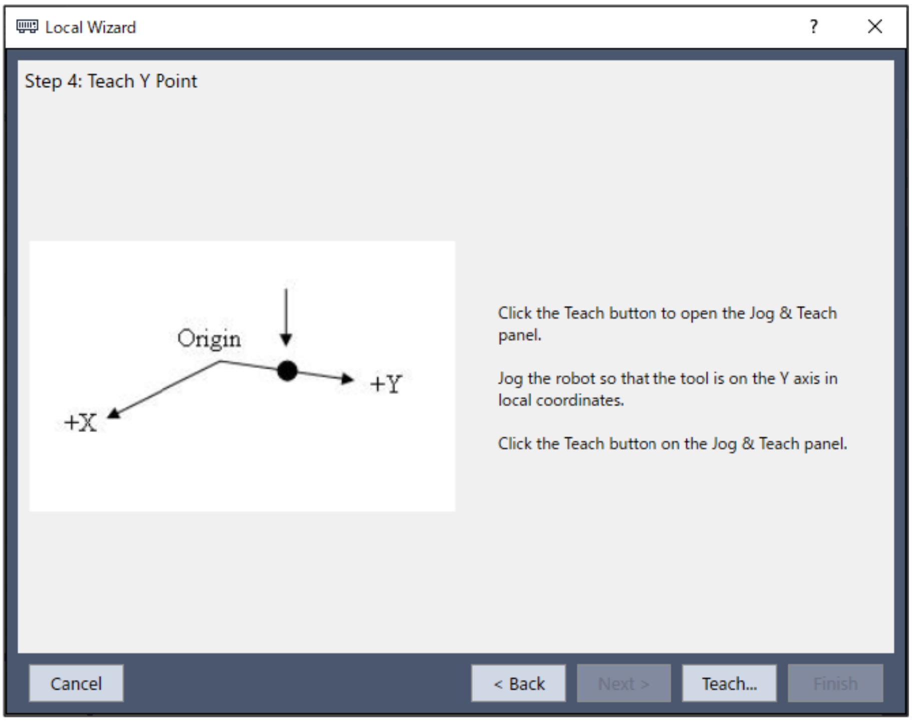
5. Jog the robot until the end effector is aligned with the origin point.

   Click the [Teach] button. The following dialog appears.



6. Teach a point on the local X axis.

   Click the [Teach] button and jog the robot until the end effector is aligned with a point anywhere along the X axis of the new coordinate system. Click the [Teach] button on the [Local Wizard Teach Point] dialog box to continue.

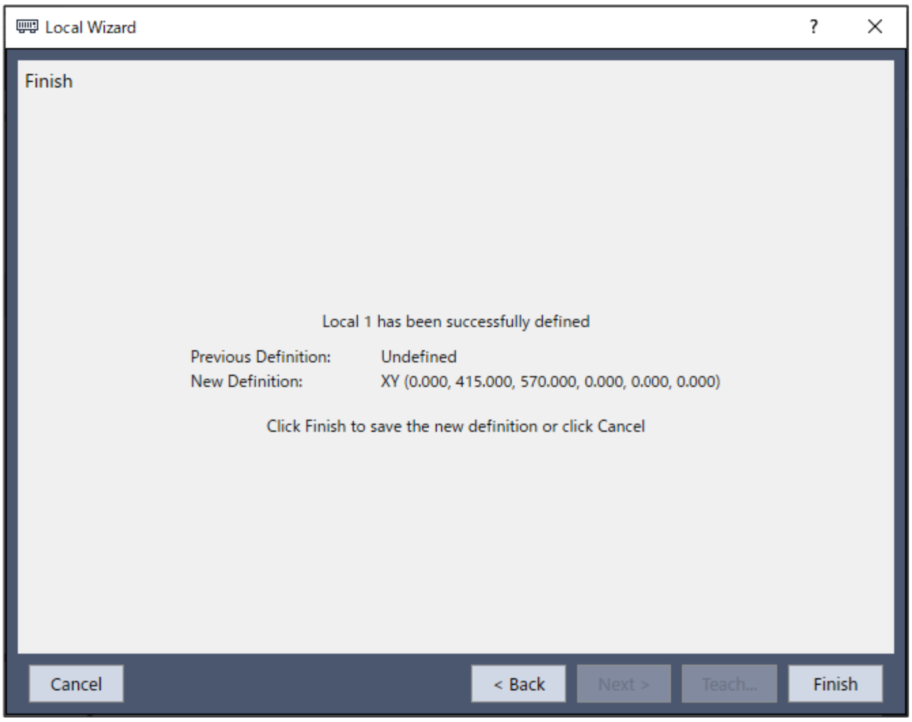


7. Teach a point on the local Y axis.

   Click the [Teach] button and jog the robot until the end effector is aligned with a point anywhere along the Y axis of the new coordinate system. Click the [Teach] button on the [Local Wizard Teach Point] dialog box to continue.

8. The new local definition is displayed as shown below.

   Click the [Finish] button to accept the new definition.



6.12.1.6 [Tools]-[Robot Manager]-[Tools] Page

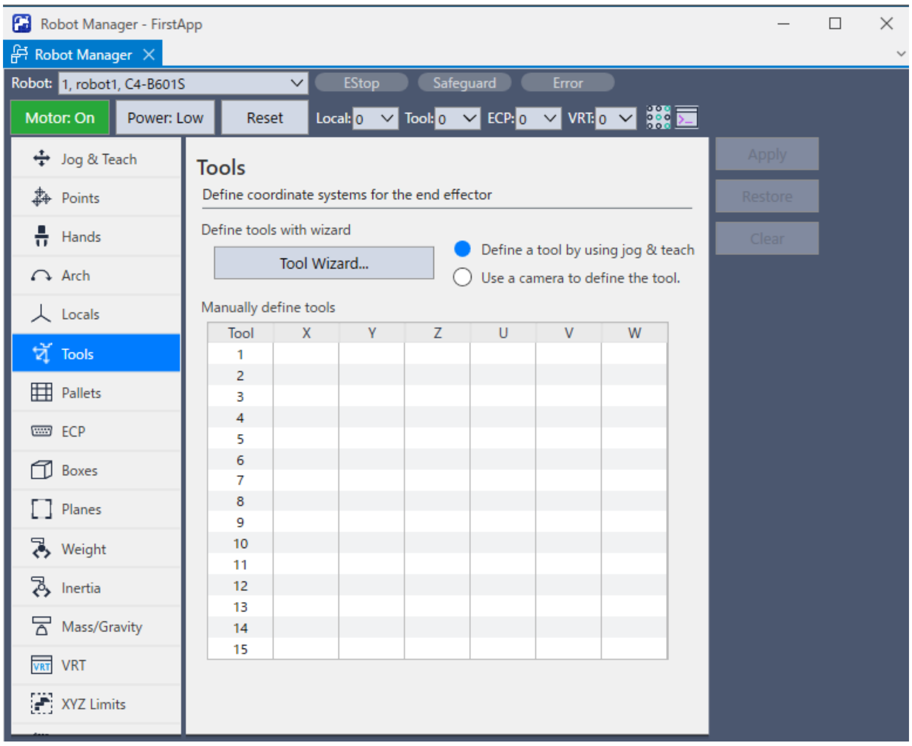
This page allows you to define tool settings for a robot. When the tab is selected, the values of 15 tools you can define are displayed.

When a tool is undefined, then all fields for that tool will be blank. When you enter a value in any of the fields for an undefined robot tool, then the remaining fields will be set to zero.

Click the [Apply] button to define the tool coordinate system.

For more details on tool settings, refer to the following manual:

"SPEL+ Language Reference - TLSet Statement"



Navigating the grid

Press the [Tab] key to move to the next cell. Press the [Shift] + [Tab] key to move to the previous cell. Press the arrow [↑] or [↓] key to move to the cell above or below.

Item	Description
Tool Wizard	This button starts the Tool Wizard. Follow the instructions for each step of the wizard to define a tool. See details in the next section.
X	The X coordinate of the tool.
Y	The Y coordinate of the tool.
Z	Z offset of tool.
U	Rotation angle of the tool about the Z axis. (roll)
V	Rotation angle of the tool about the Y axis. (pitch)
W	Rotation angle of the tool about the X axis. (yaw)
Apply	Sets the current values.

Item	Description
Restore	Reverts to the previous values.
Clear	Clears all selected values.

Define tools with wizard

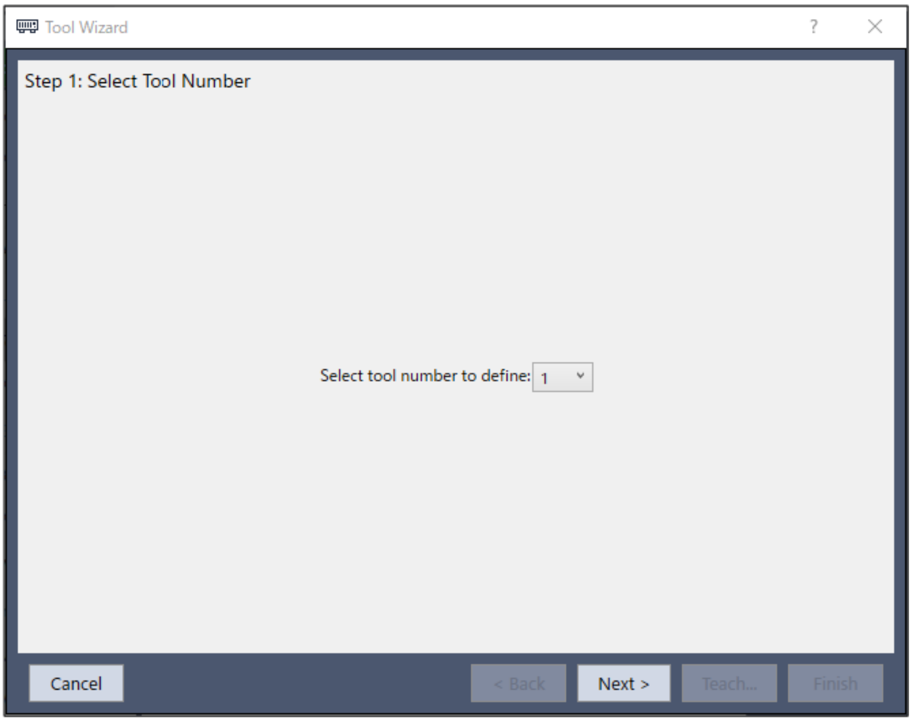
A wizard is provided for defining a tools coordinate system. This section describes the procedure for SCARA and 6-axis robots when [Define a tool by using jog & teach] is selected. For details of the Tool definition, refer to the following manual:

"Vision Calibration" in "Vision Guide 8.0 Software"

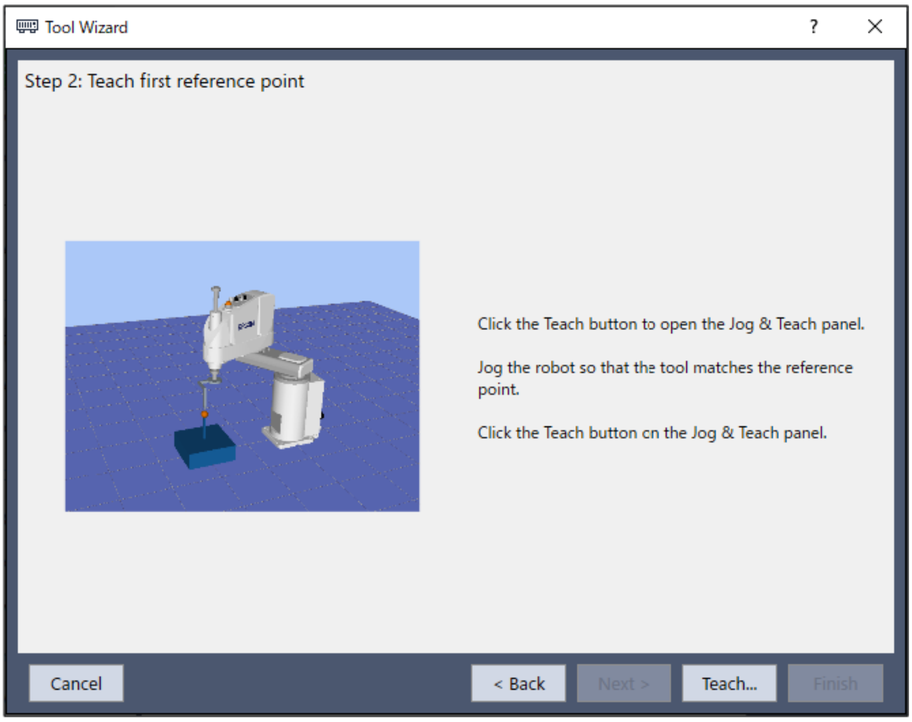
Using the Tool Wizard for SCARA Robots

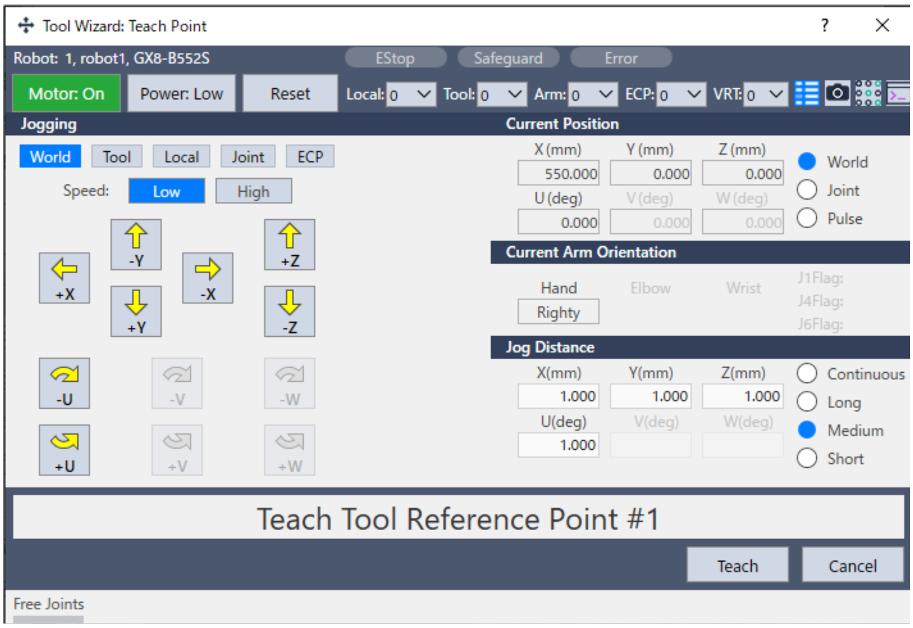
1. Select [Robot Manager]-[Tools] tab to show the [Tools] page.
2. Select [Define a tool by using jog & teach], and then click the [Tool Wizard] button. The dialog shown below appears.

   Select the tool number to define and click the [Next] button.



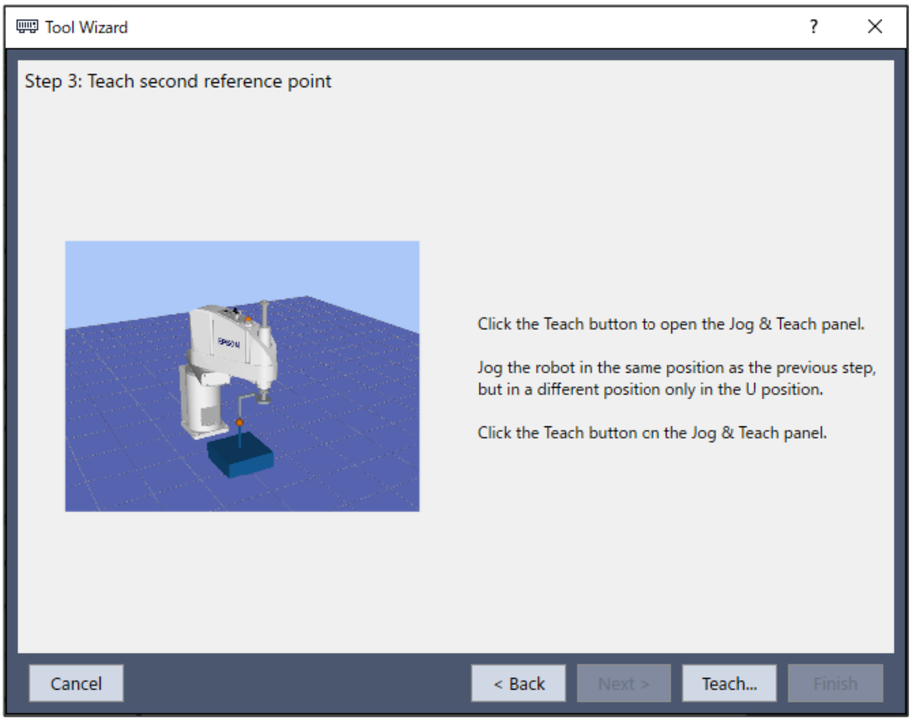
3. Jog the robot until the tool is aligned with the reference point. Then click the [Teach] button to show the [Jog & Teach] dialog box. Match the tool and the reference point.

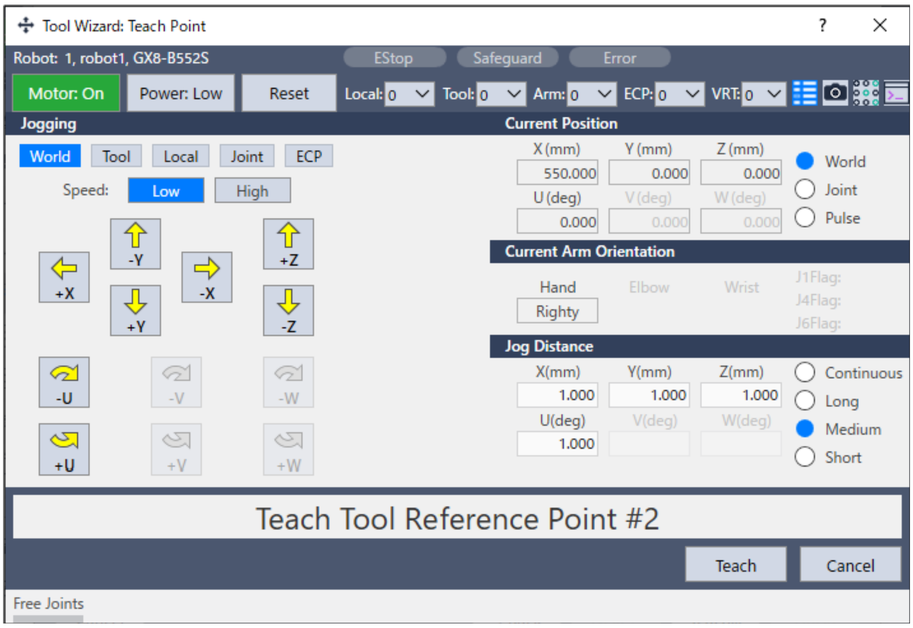




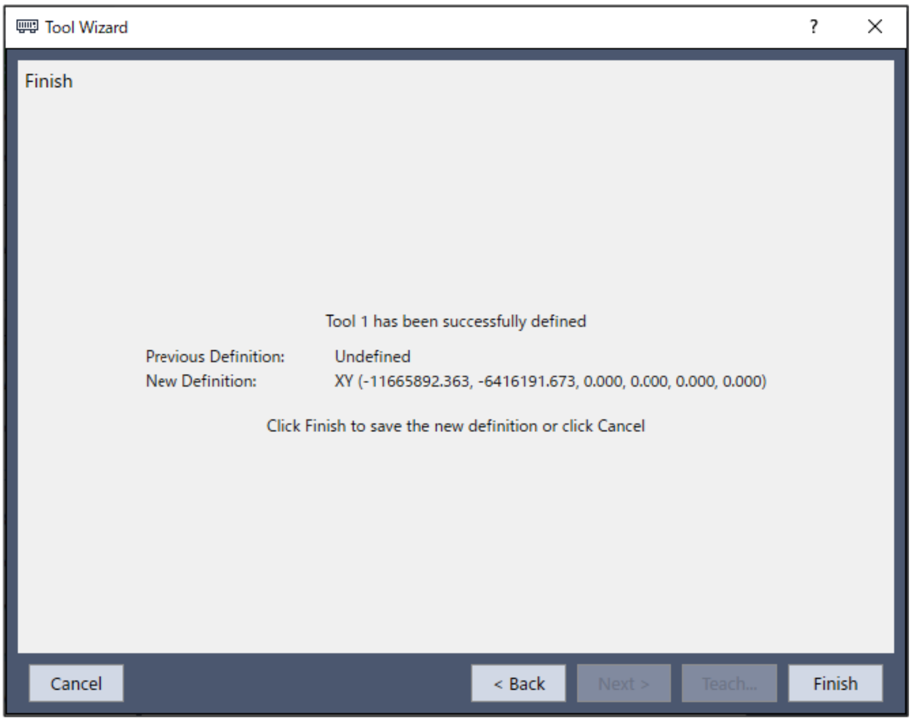
4. Click the [Teach] button. The following dialog appears.

After rotating the U axis as shown below to change the angle, jog the X and Y axes until the tool is aligned with the reference point. Then click the [Teach] button to show the [Jog & Teach] dialog box. Match the tool and the reference point.





5. Click the [Teach] button. The new tool definition is displayed as shown below. Click the [Finish] button to apply the new definition.



KEY POINTS

The robot can be calibrated with a different posture from the wizard.

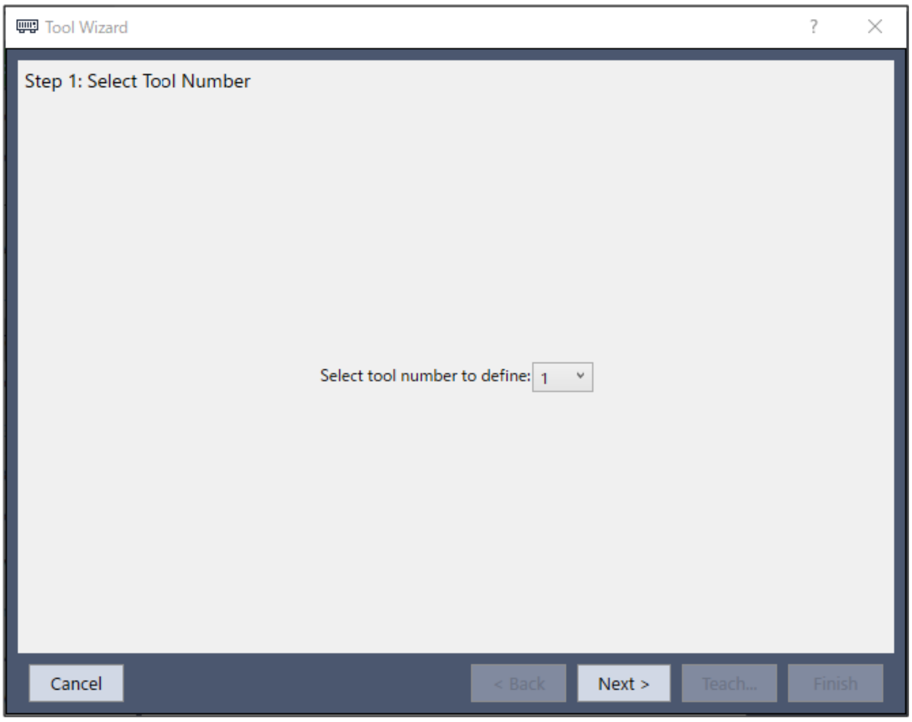
For 6-axis robots (including N series)

KEY POINTS

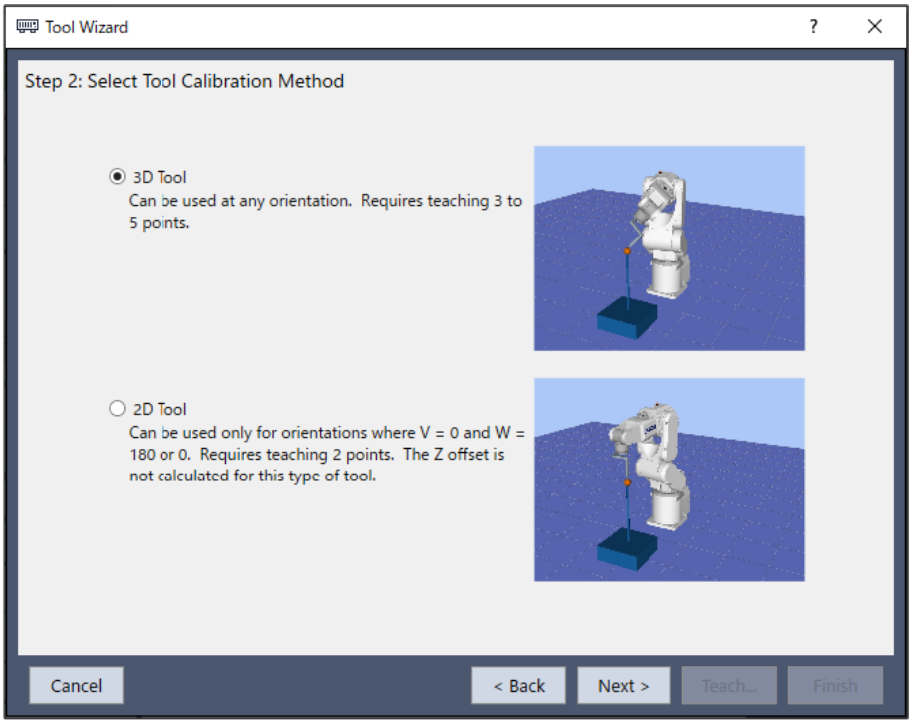
- There are two calibration methods for 6-axis robots. 3D Tool moves the robot in X, Y, Z, U, V, and W directions to calibrate, while 2D Tool moves the robot in X, Y, Z, and U directions. The robot can be calibrated with 2D Tool only when the robot posture is “V=0°, W=0°”, or “V=0°, W=180° (-180°).
- When comparing 2D Tool and 3D Tool, 2D Tool has following advantages and disadvantages. Choose the suitable method according to the intended use.
  - Advantages:
    - Shorter calibration time than 3D Tool
    - Since V and W axes are not moved, peripherals and cables are less likely to interfere the calibration
  - Disadvantages:
    - Calibration accuracy may be worse than 3D Tool
    - The Z-axis direction offset is not performed automatically (*1)

*1: If the Z-axis direction offset is required, enter the offset value in the following dialog box after calibration.

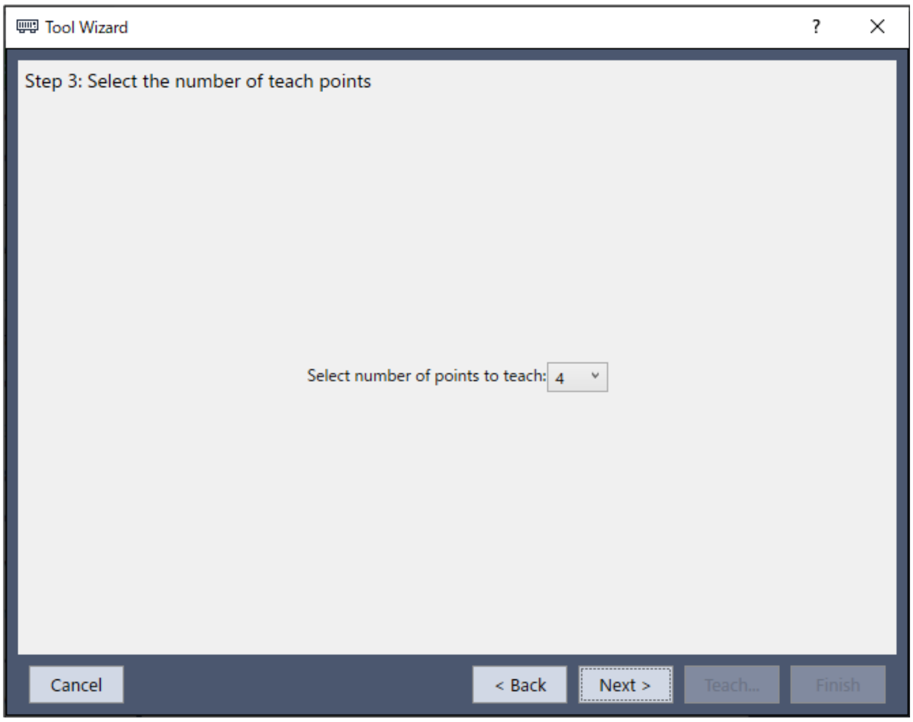
1. Select [Robot Manager]-[Tools] tab to show the [Tools] page.
2. Select [Define a tool by using jog & teach], and then click the [Tool Wizard] button. The dialog shown below appears. Select the tool number to define and click the [Next] button.



3. Click the [Next] button. Select either 3D Tool or 2D Tool.



4. If using 3D Tool, select the number of points to teach, and click the [Next] button.

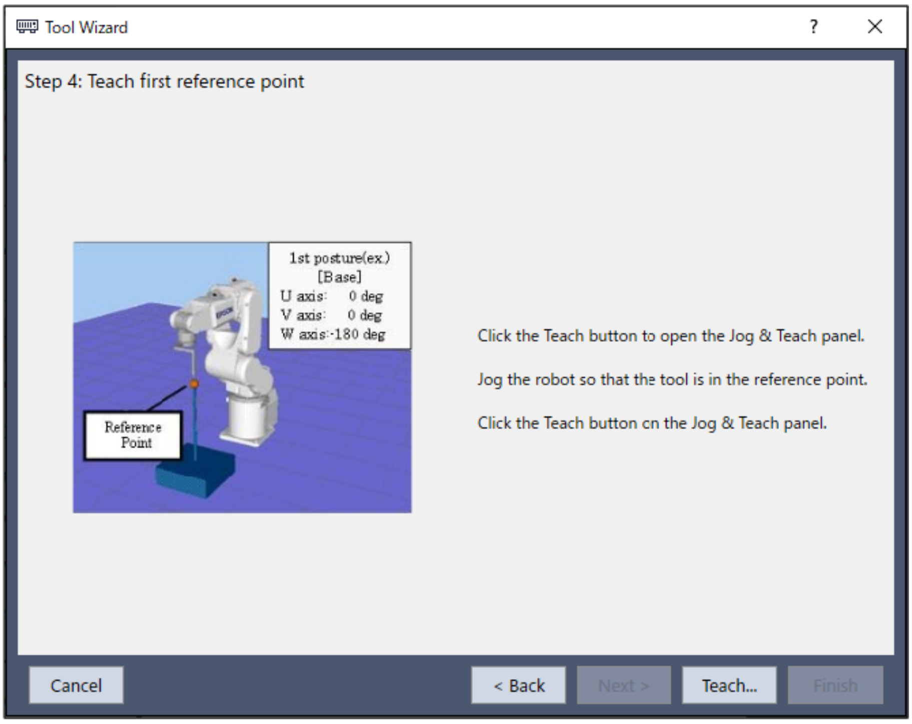


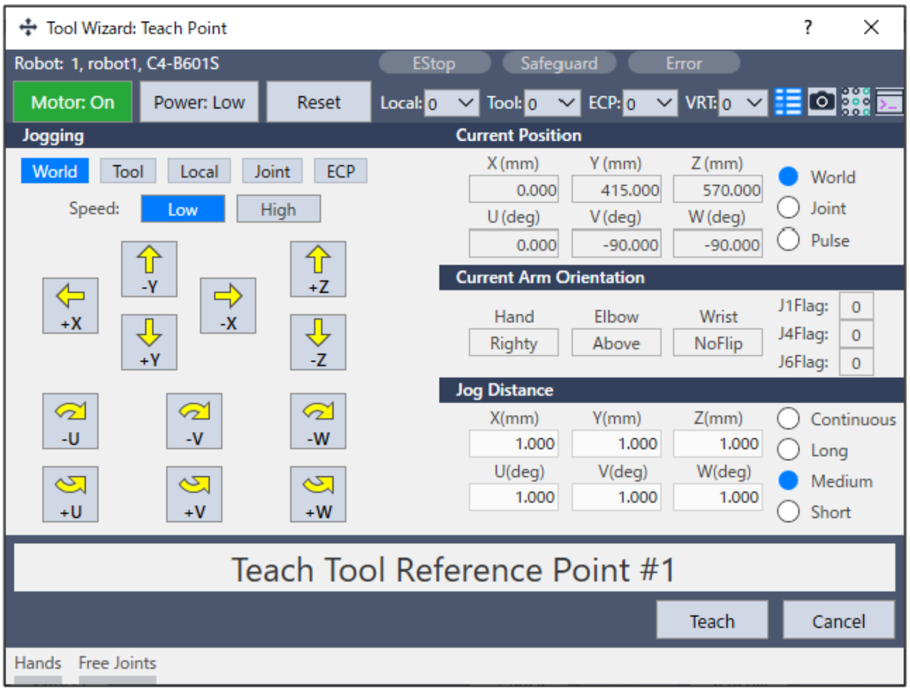
KEY POINTS

The "number of points to teach" is the amount of times to teach the same point (reference point) in the robot motion range while changing only the tool direction. The number to teach should be at least three. Although it depends on the teaching accuracy of each point, more accurate tool setting can be set by increasing the number. To increase the tool setting accuracy, set the angle of approximately 10° or more for J5 pulse in order to avoid singularity near 0° when teaching the reference point.

5. Jog the robot until the tool is aligned with the reference point.

   Then click the [Teach] button to show the [Jog & Teach] dialog box. Match the tool and the reference point.





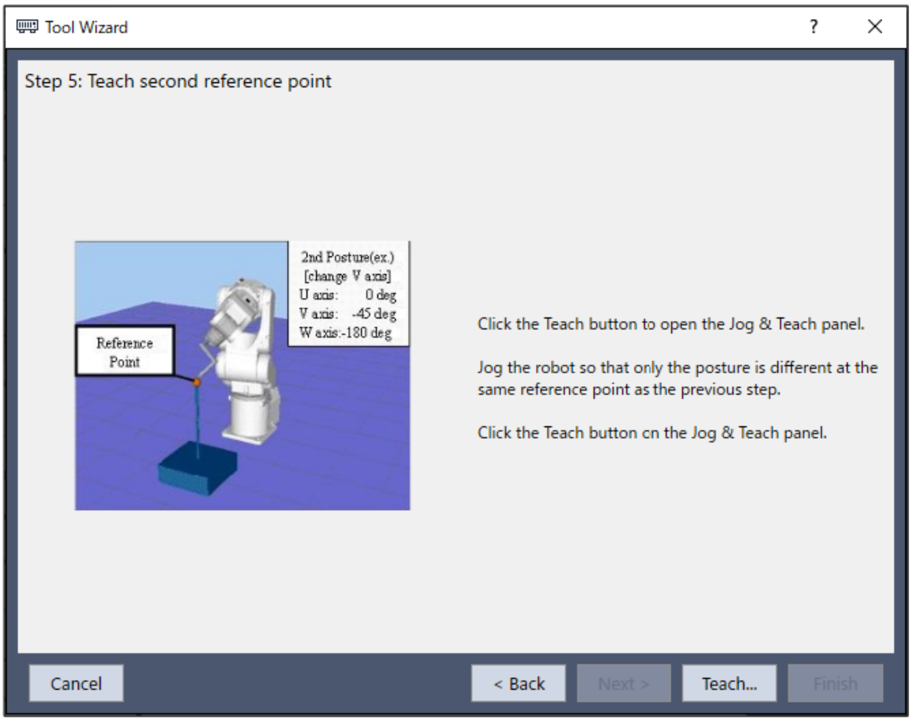
6. Click the [Teach] button. The following dialog appears.
   - If using 3D Tool, rotate the U, V, and W axes as shown below, and then jog the X, Y, and Z axes until the tool is aligned with the reference point. Repeat teaching until the robot can reach the reference point from other tool orientation as often as you specified in (3).
   - If using 2D Tool, rotate only the U axis as shown below, and then jog the X, Y, and Z axes until the tool is aligned with the reference point.

Clicking the [Teach] button displays the [Jog & Teach] dialog box for both 3D Tool and 2D Tool. Match the tool and the reference point.

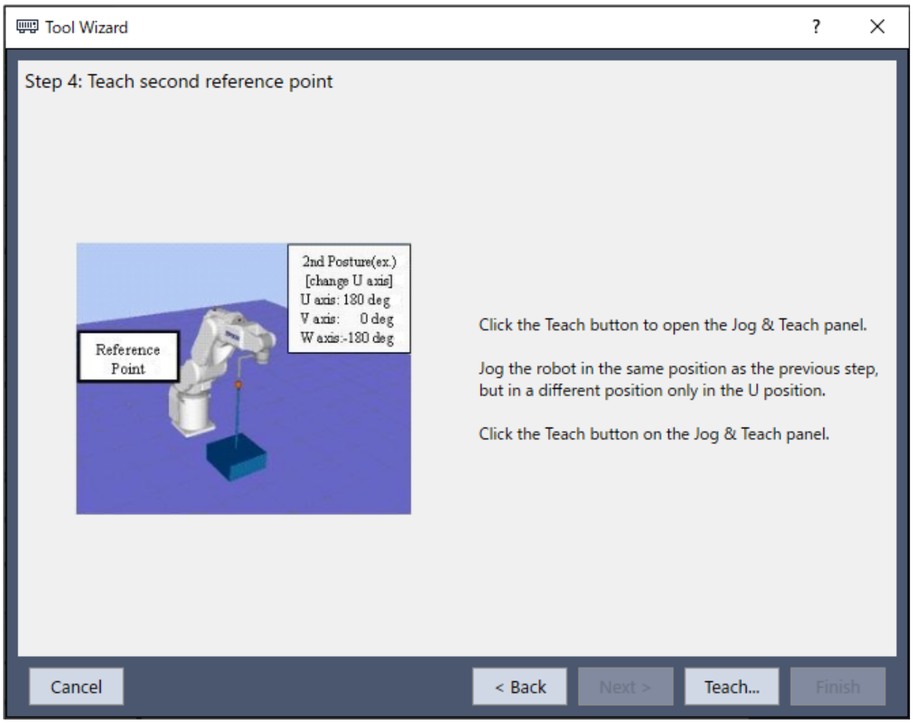
KEY POINTS

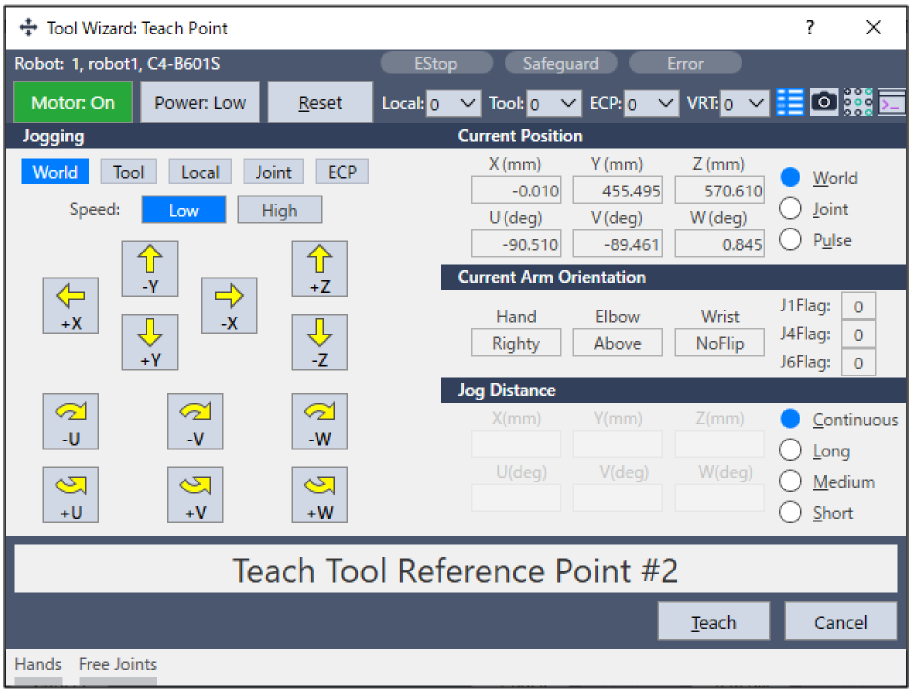
When moving the U, V, and W axes, move the arm upward in order to avoid collision of the tool and the reference point.

For 3D Tool:

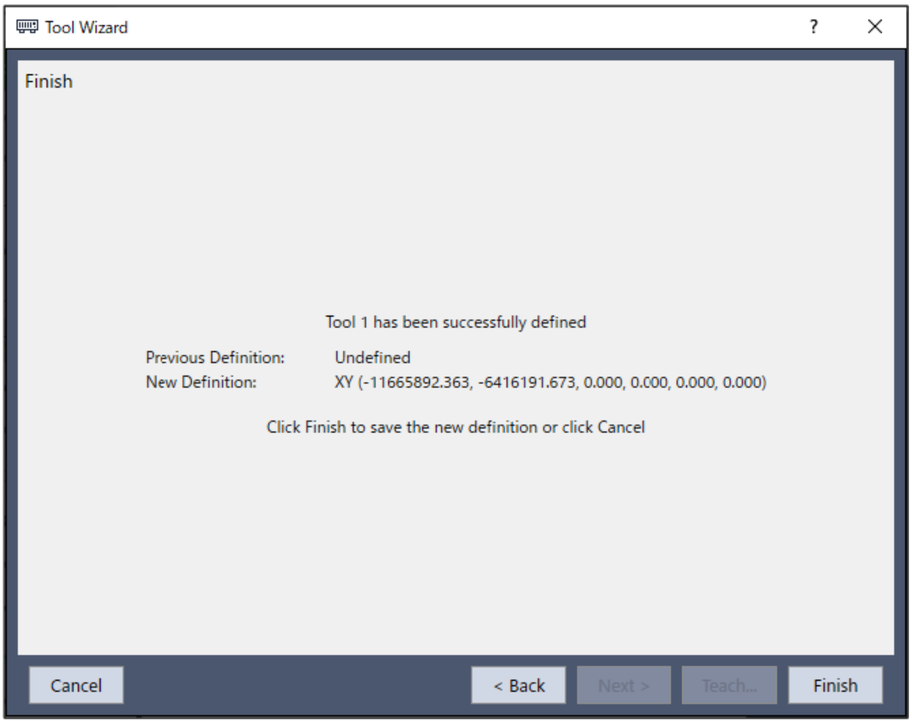

Tool Wizard
Step 5: Teach second reference point
2nd Posture(ex.)
[change V axis]
U axis: 0 deg
V axis: -45 deg
W axis:-180 deg
Reference Point
Click the Teach button to open the Jog & Teach panel.
Jog the robot so that only the posture is different at the same reference point as the previous step.
Click the Teach button cn the Jog & Teach panel.
Cancel
< Back
Next >
Teach...
Finish

For 2D Tool:


Tool Wizard
Step 4: Teach second reference point
2nd Posture(ex.)
[change U axis]
U axis: 180 deg
V axis: 0 deg
W axis:-180 deg
Reference Point
Click the Teach button to open the Jog & Teach panel.
Jog the robot in the same position as the previous step, but in a different position only in the U position.
Click the Teach button on the Jog & Teach panel.
Cancel
< Back
Next >
Teach...
Finish



7. The new tool definition is displayed as shown below. Click the [Finish] button to apply the new definition.



KEY POINTS

Although it is recommended to calibrate the robot with the same posture as the wizard, it is possible to calibrate the robot with the different robot posture from wizard. When you don't use the robot with the same posture as the wizard, change the robot posture for five degrees or more. The bigger the robot posture change, the more accurate the tool setting.

6.12.1.7 [Tools]-[Robot Manager]-[Arms] Page

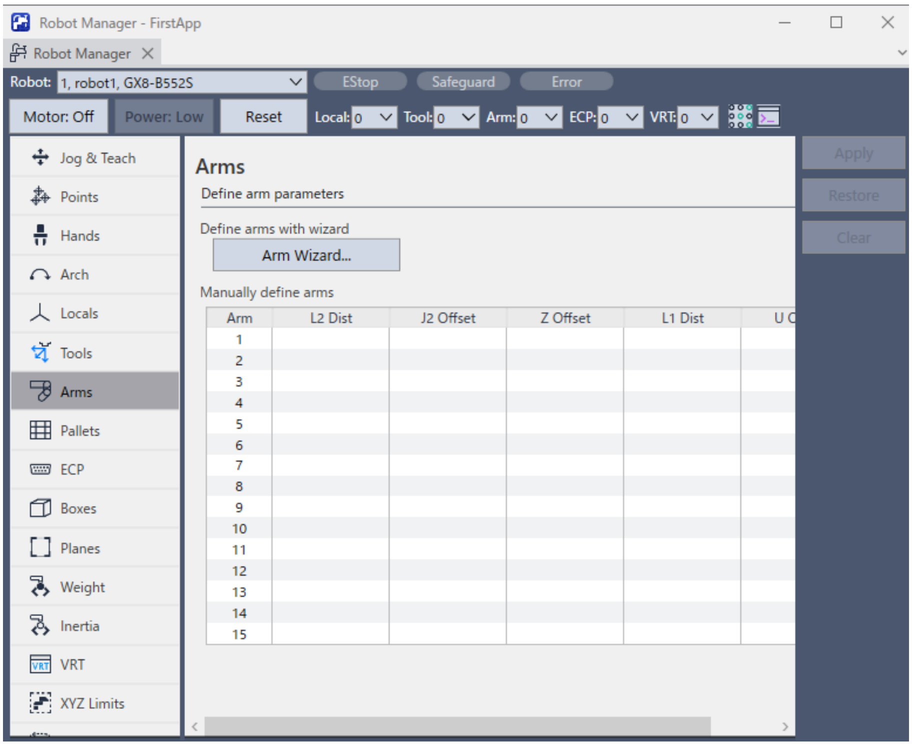
This page allows you to define Arm settings for a robot. When the tab is selected, the current Arm values are displayed. The tab is disabled if the current robot does not support the Arm command.

A grid is used to display all the values for all 15 arm configurations you can define.

When an arm is undefined, then all fields for that arm will be blank. When you enter a value in any of the fields for an undefined arm, then the remaining fields will be set to zero and the tool will be defined when you click the [Apply] button.

For more details on arm settings, refer to the following manual:

"SPEL+ Language Reference - ArmSet Statement"



Navigating the grid

Press the [Tab] key to move to the next cell. Press the [Shift] + [Tab] key to move to the previous cell. Press the arrow [↑] or [↓] key to move to the cell above or below.

Item	Description
Arm Wizard	Open the wizard for configuring the additional arm using the camera. Define the tool by following the instructions. For more details, refer to the following manual: "Vision Calibration" in "Vision Guide 8.0 Software"
L2 Dist	Distance between the center of joint 2 and the center of the orientation joint in millimeters.
J2 Offset	Angle of the line from the center of joint 2 to the center of the orientation joint in "°".
Z Offset	The Z offset between the new orientation axis and the standard orientation axis.
L1 Dist	Distance between the center of the shoulder joint and the center of the elbow joint in millimeters.
U Offset	The angle offset between the standard orientation zero position and the new orientation axis zero position in "°".

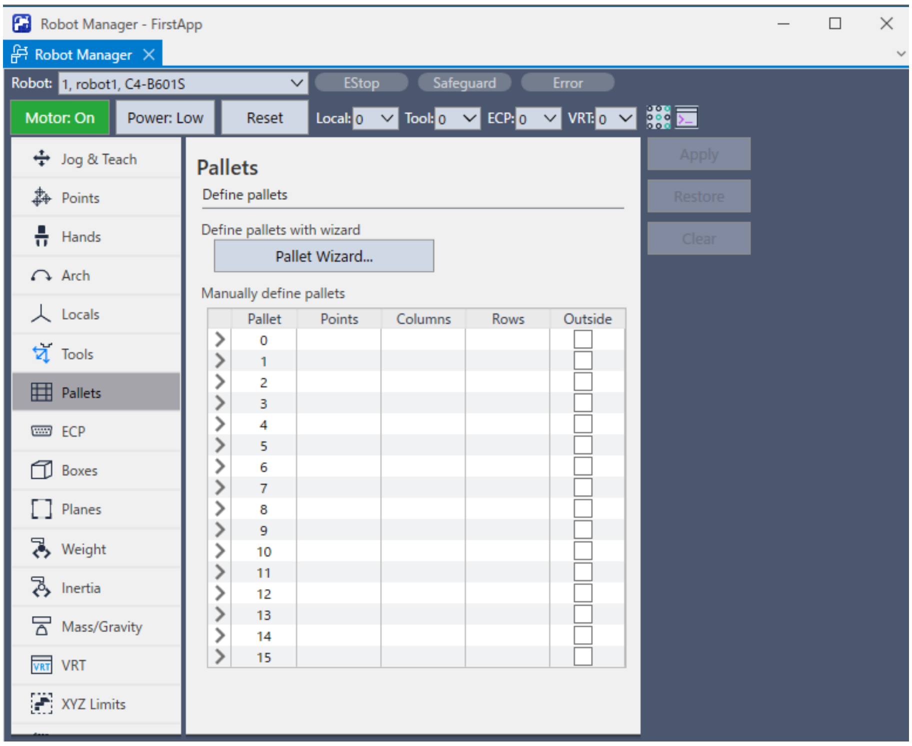
Item	Description
Apply	Sets the current values.
Restore	Reverts to the previous values.
Clear	Clears all selected values.

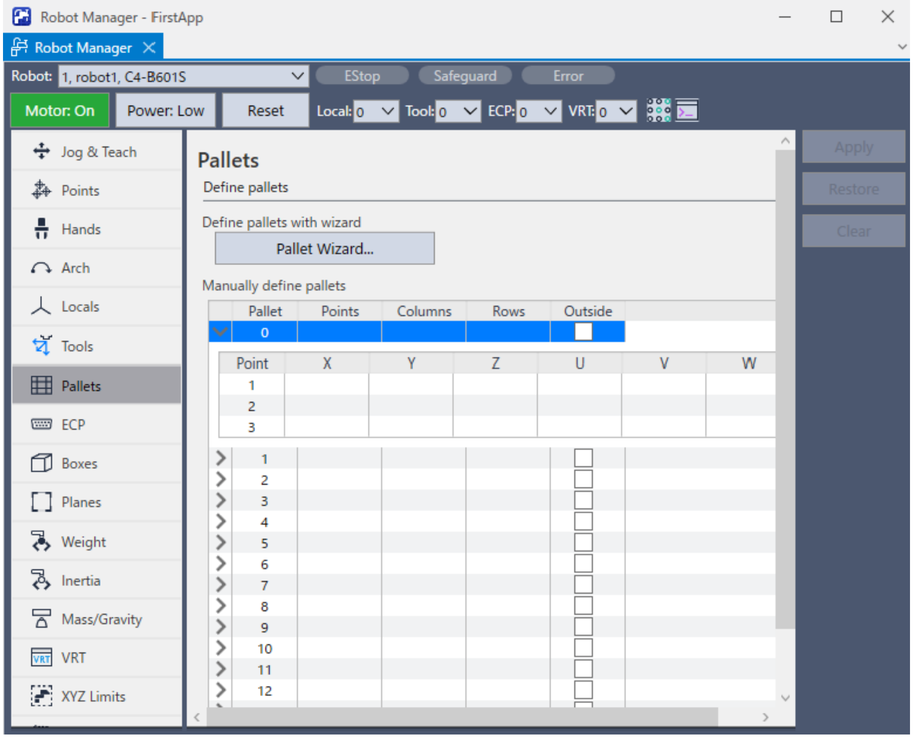
6.12.1.8 [Tools]-[Robot Manager]-[Pallets] Page

This page allows you to define the pallet (Pallet). When the page is selected, values for the available pallet are displayed. When a Pallet is undefined, then all fields for the Pallet will be blank. The Pallet will be defined when you press the [Apply] button.

For more details on Pallet settings, refer to the following manual:

"SPEL+ Language Reference - Pallet Statement"





Navigating the grid

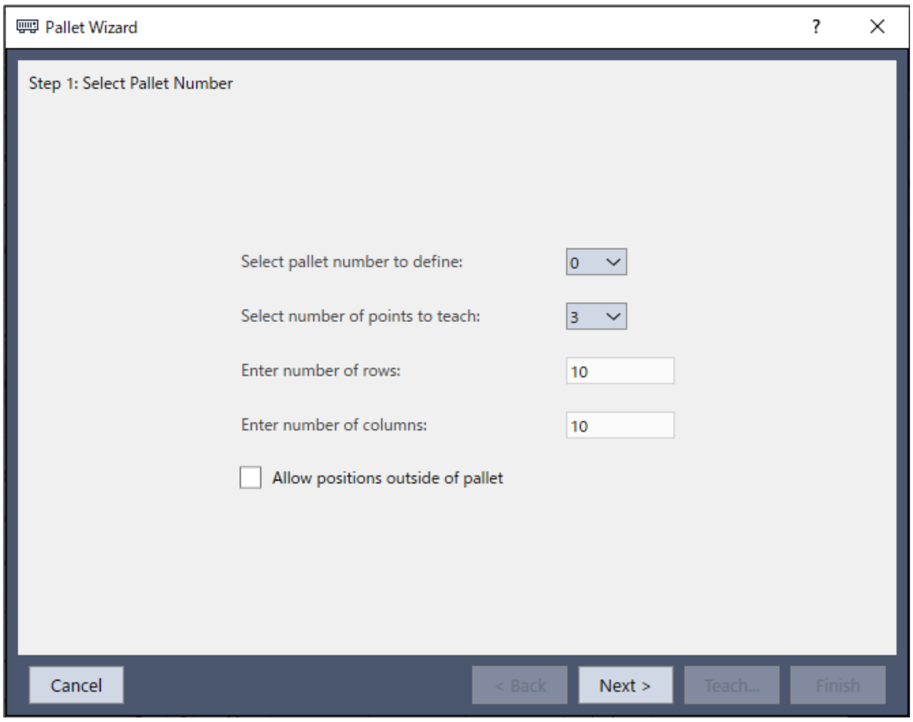
Press the [Tab] key to move to the next cell. Press the [Shift] + [Tab] key to move to the previous cell. Press the arrow [↑] or [↓] key to move to the cell above or below.

Item	Description
Points	Specify the point variable to use for pallet definition. Select either 3 or 4.
Columns	Specify the division number of Point number 1(coordinate system data 1) and Point number 2 (coordinate system data 2) by an integer. The range is from 1 to 32767. (Division× Division Division <= 32767)
Rows	Specify the division number of Point number 1(coordinate system data 1) and Point number 3 (coordinate system data 3) by an integer. The range is from 1 to 32767. (Division× Division Division <= 32767)
Outside	Creates an accessible pallet outside the specified columns and rows. Optional.
X	Set the X coordinate in millimeters.
Y	Set the Y coordinate in millimeters.
Z	Set the Z coordinate in millimeters.
U	Set the U coordinate in degrees.
V	Set the V coordinate in degrees.
W	Set the W coordinate in degrees.
Apply	Sets the current values.
Restore	Reverts to the previous values.
Clear	Clears all values.

Wizard

1. Select [Robot Manager]-[Pallets] tab to show the [Pallets] page.

2. Click the [Pallet Wizard] button. The following dialog appears.

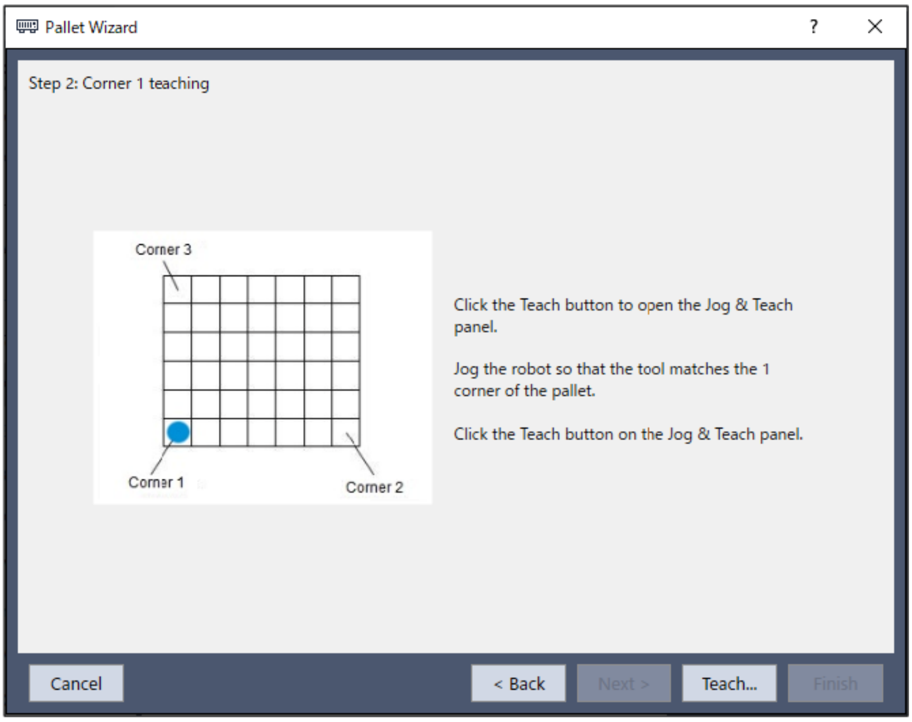


3. Select the pallet number to define, the number of points to teach, the number of rows and columns, and whether to use "Outside". Then, click the [Next] button.

KEY POINTS

If a pallet is a well ordered rectangular shape, only 3 of the 4 corner points should be specified. However, in most situations, it is recommended to use 4 corner points to define a pallet.

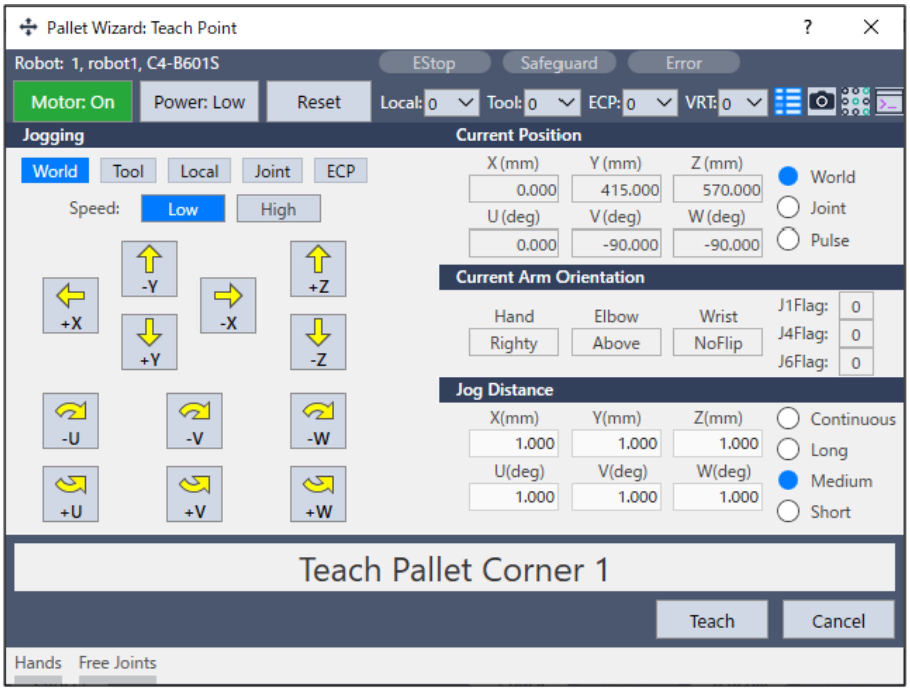
4. Click the [Teach] button to show the [Teach first point] page.



KEY POINTS

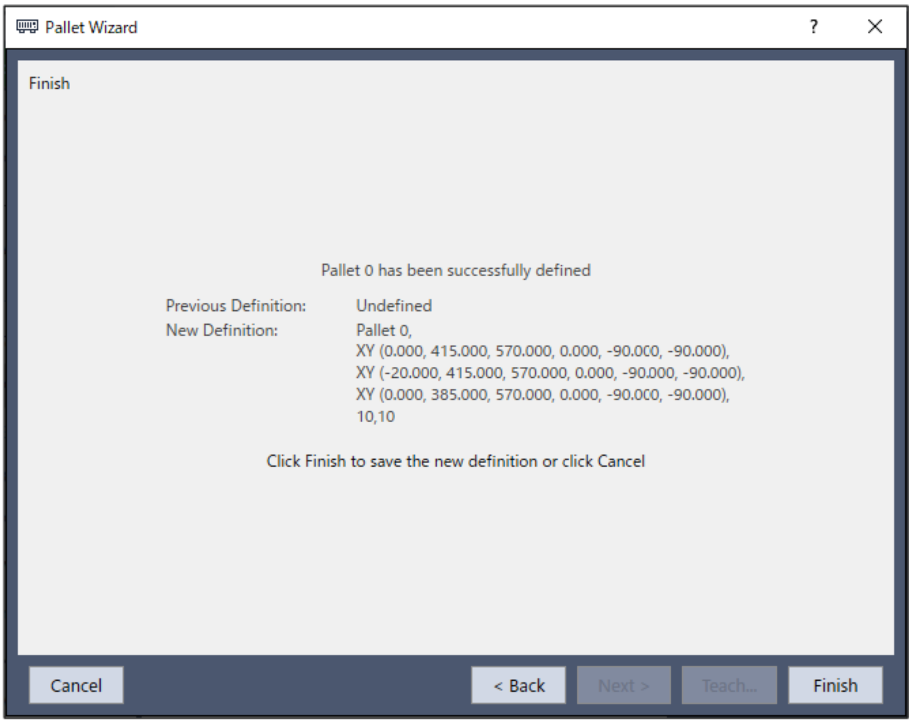
You can proceed to the next step without teaching if a predefined pallet number is selected. When doing so, only teach the points that require re-teaching.

5. Jog the robot to the first corner to teach the position of it. Click the [Teach] button. The following dialog appears.



6. Teach the second to forth corners by following the steps (4) and (5).

7. The new pallet definition is displayed as shown below. Click [Finish] to apply the new definition.



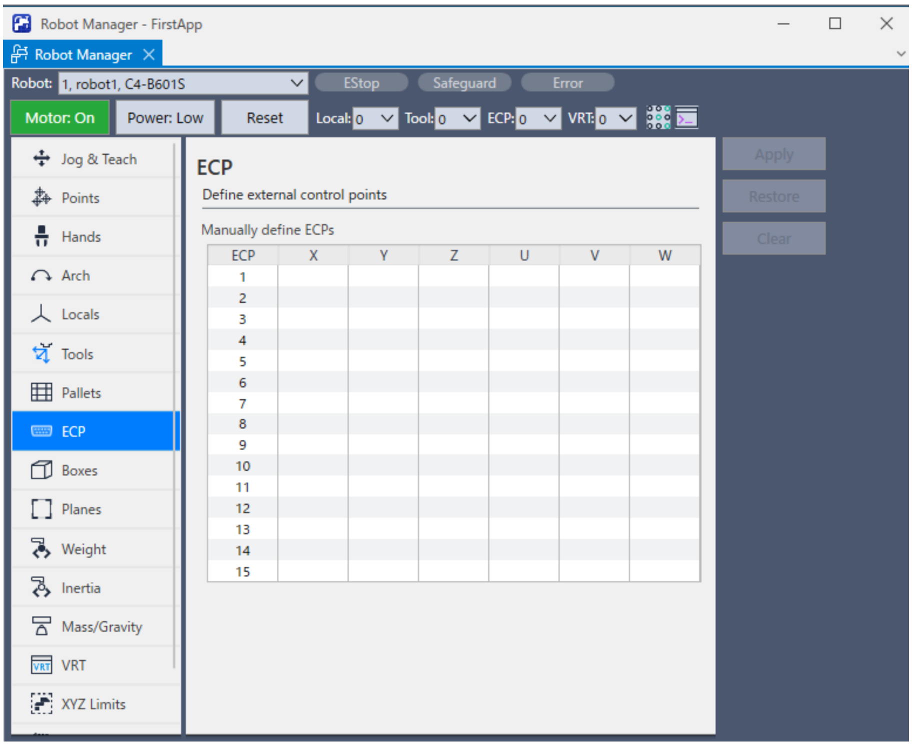
6.12.1.9 [Tools]-[Robot Manager]-[ECP] Page

This page allows you to define ECP (external control point) settings for a robot. When the page is selected, the current values are displayed. When an ECP is undefined, then all fields for that ECP will be blank. When you enter a value in any of the fields for an undefined ECP setting, then the remaining fields will be set to zero. The ECP will be defined when you press the [Apply] button.

If the ECP option is not enabled in the controller, this page will not be visible.

For detailed information on using external control points in your application, see below.

ECP Coordinate Systems (Option)



Navigating the grid

Press the [Tab] key to move to the next cell. Press the [Shift] + [Tab] key to move to the previous cell. Press the arrow [↑] or [↓] key to move to the cell above or below.

Item	Description
X	The X coordinate of the ECP.
Y	The Y coordinate of the ECP.
Z	The Z coordinate of the ECP.
U	Rotation angle of the ECP about the Z axis. (roll)
V	Rotation angle of the ECP about the Y axis. (pitch)
W	Rotation angle of the ECP about the X axis. (yaw)
Apply	Sets the current values.
Restore	Reverts to the previous values.

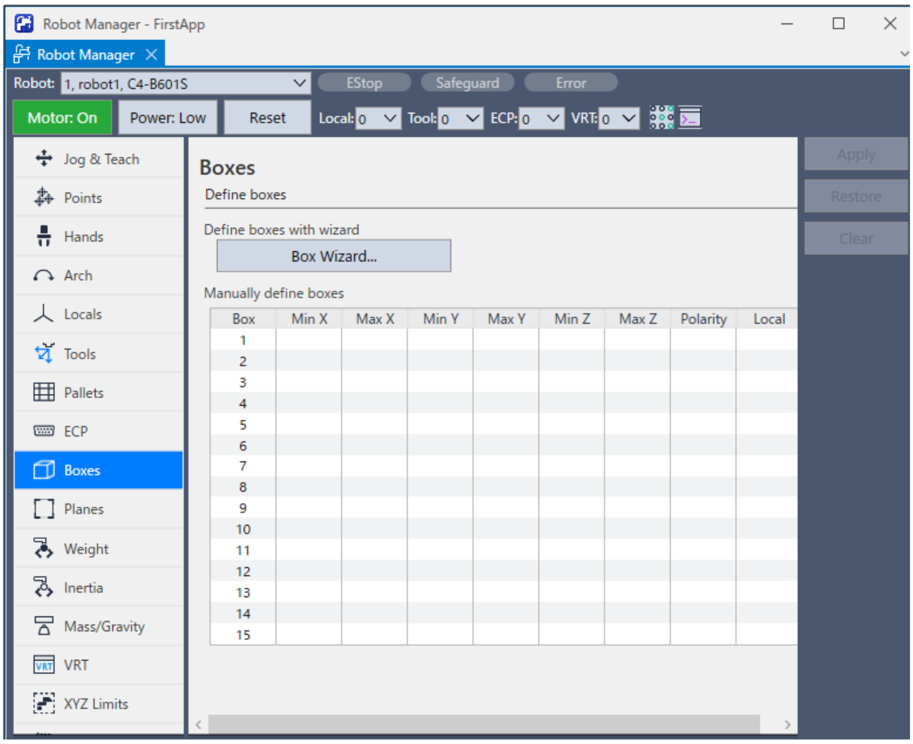
Item	Description
Clear	Clears all values.

6.12.1.10 [Tools]-[Robot Manager]-[Boxes] Page

This page allows you to define Box (approach check area) settings for a robot. When the page is selected, the current values are displayed. When a Box is undefined, then all fields for that Box will be blank. When you enter a value in any of the fields for an undefined Box setting, then the remaining fields will be set to zero. The Box will be defined when you press the [Apply] button.

For more details on Box Page settings, refer to the following manual:

"SPEL+ Language Reference - Box Statement"



Navigating the grid

Press the [Tab] key to move to the next cell. Press the [Shift] + [Tab] key to move to the previous cell. Press the arrow [↑] or [↓] key to move to the cell above or below.

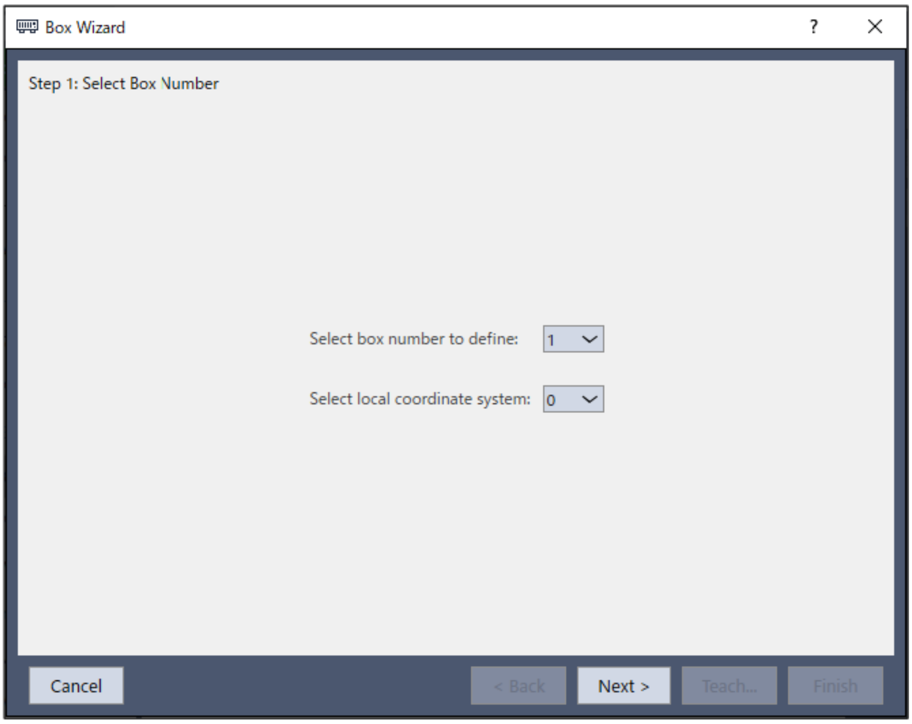
Item	Description
Min X	Type in the minimum X limit value in millimeters.
Max X	Type in the maximum X limit value in millimeters.
Min Y	Type in the minimum Y limit value in millimeters.
Max Y	Type in the maximum Y limit value in millimeters.
Min Z	Type in the minimum Z limit value in millimeters.
Max Z	Type in the maximum Z limit value in millimeters.
Polarity	Sets the polarity to output I/O at approach check.

Item	Description
Local	Selects the local coordinate system.
Apply	Sets the current values.
Restore	Reverts to the previous values.
Clear	Clears all values.

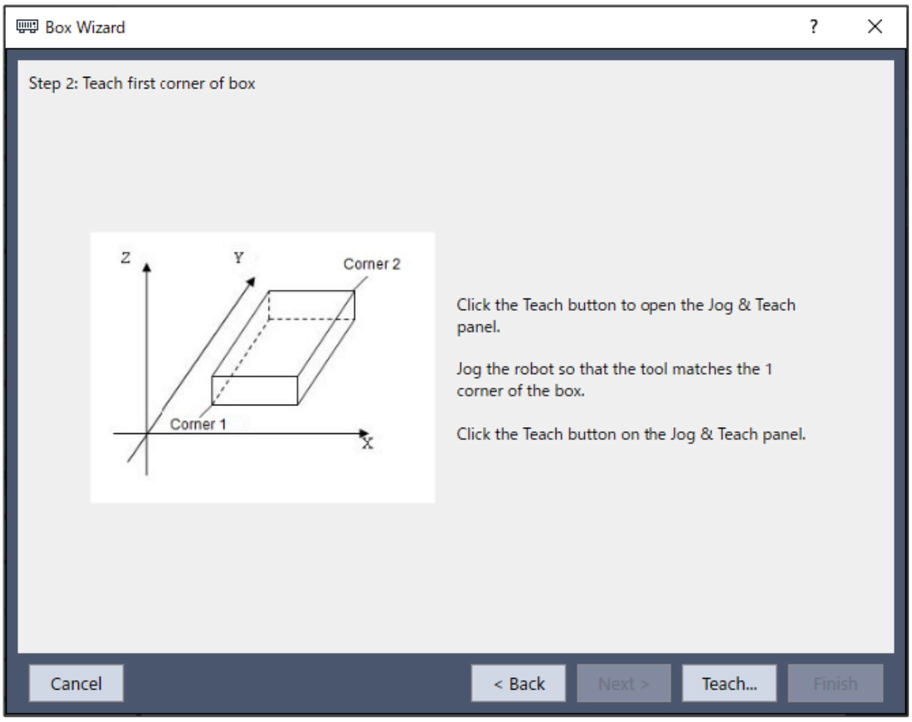
Setting both values to zero disables the limits.

Using the Box Wizard

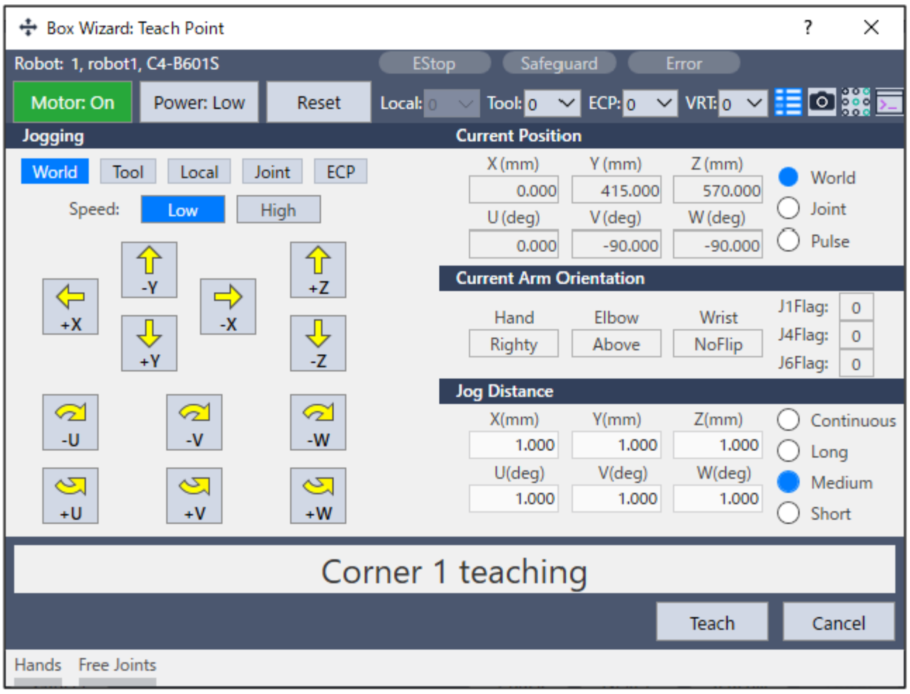
1. Select [Robot Manager]-[Boxes] tab to show the [Boxes] page.
2. Click the [Box Wizard] button. The following dialog appears.



3. Select the Box number and local coordinate system to define and click the [Next] button.
4. Click the [Teach] button to show the [Teach first corner of box] page.

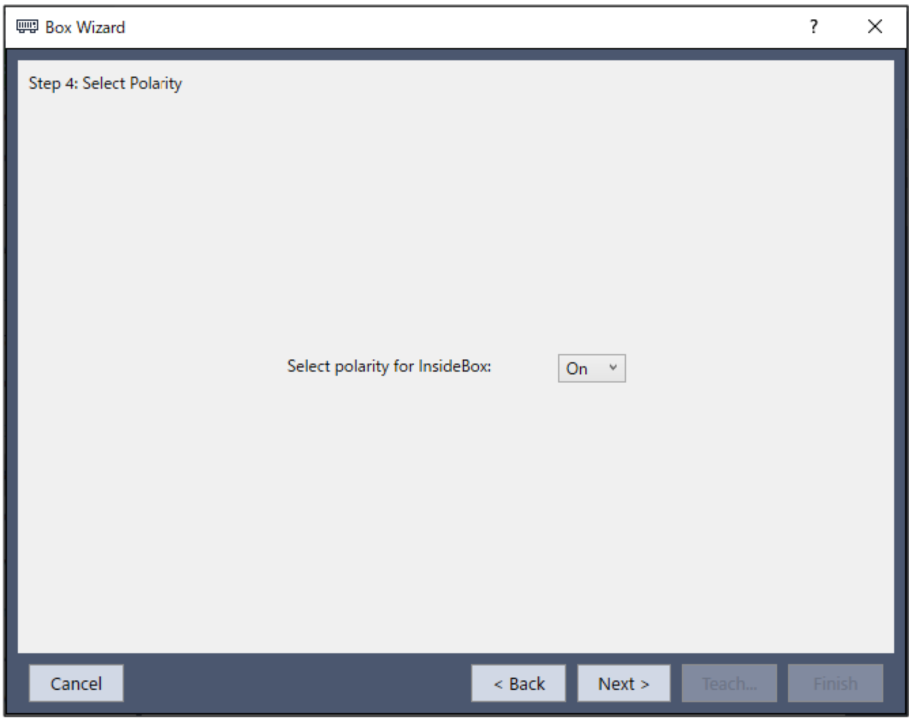


5. Jog the robot to the first corner to teach the position of it. Click the [Teach] button. The following dialog appears.

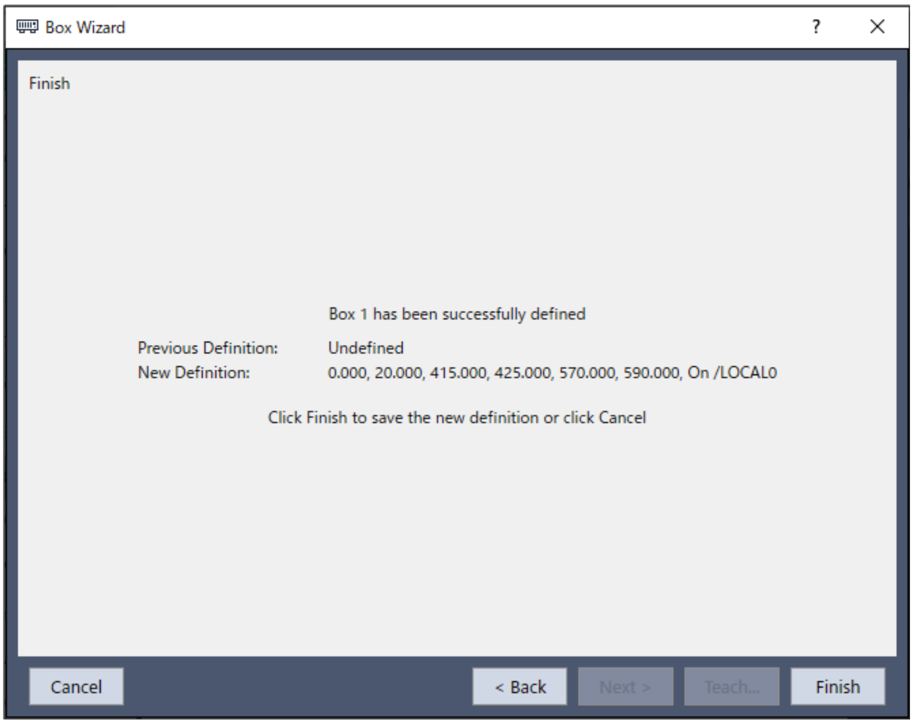


6. Teach the second corner by following the steps (4) and (5).

7. Sets the polarity to output I/O at approach check.



8. The new box definition is displayed as shown below. Click [Finish] to apply the new definition.

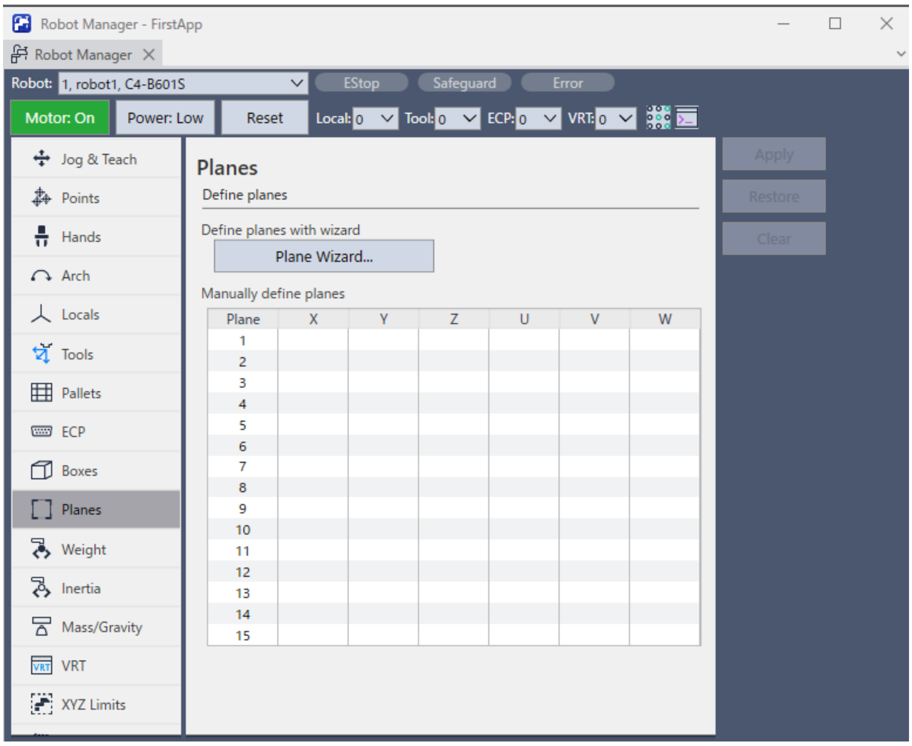


6.12.1.11 [Tools]-[Robot Manager]-[Planes] Page

This page allows you to define Plane (approach check plane) settings for a robot. When the page is selected, the current values are displayed. When a Plane is undefined, then all fields for that Plane will be blank. When you enter a value in any of the fields for an undefined Plane setting, then the remaining fields will be set to zero. The Plane will be defined when you press the [Apply] button.

For more details on Planes settings, refer to the following manual:

"SPEL+ Language Reference - Plane Statement"



Navigating the grid

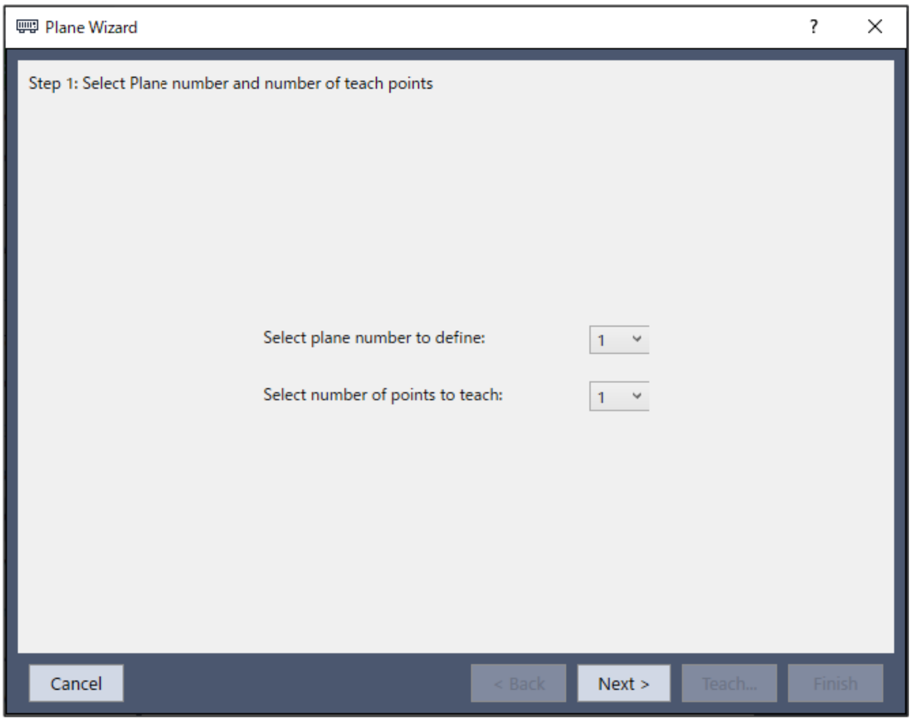
Press the [Tab] key to move to the next cell. Press the [Shift] + [Tab] key to move to the previous cell. Press the arrow [↑] or [↓] key to move to the cell above or below.

Item	Description
X	Sets the X origin of the coordinate for approach check plane.
Y	Sets the Y origin of the coordinate for approach check plane.
Z	Sets the Z origin of the coordinate for approach check plane.
U	Sets the U origin of the coordinate for approach check plane.
V	Sets the V origin of the coordinate for approach check plane.
W	Sets the W origin of the coordinate for approach check plane.
Apply	Sets the current values.
Restore	Reverts to the previous values.
Clear	Clears all values.

Wizard

1. Select [Robot Manager]-[Planes] tab to show the [Planes] page.

2. Click the [Plane Wizard] button. The dialog shown below appears.



3. Select the plane number to define and the number of points to teach, and then click the [Next] button.

KEY POINTS

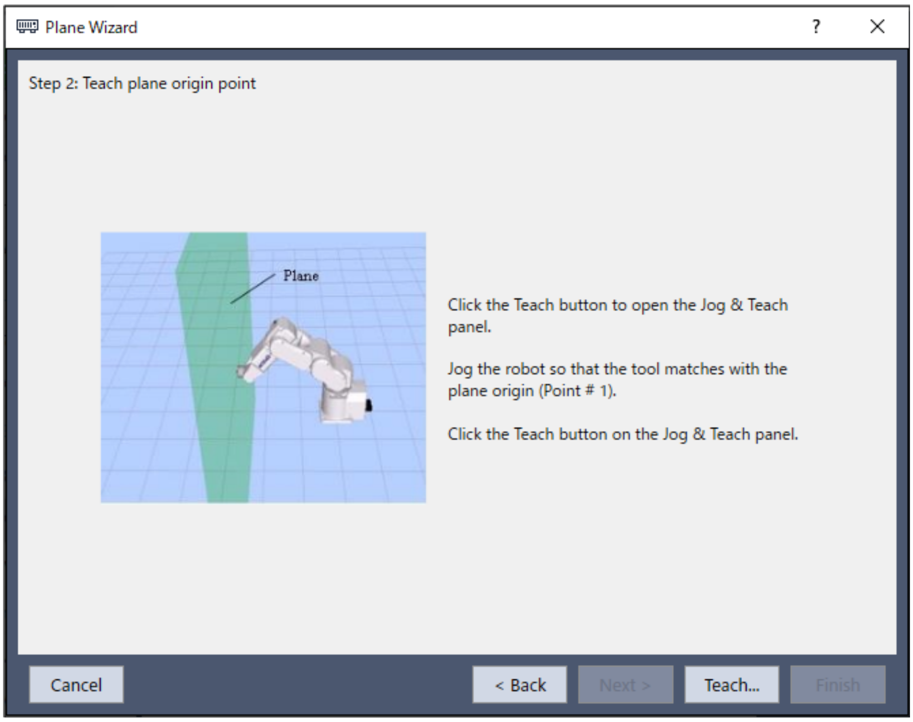
You can select either "1" or "3" for the number of points to teach. If you select "1", the robot posture at teaching will be reflected. If you select "3", the robot posture will not be reflected.

For details, refer to the following manual.

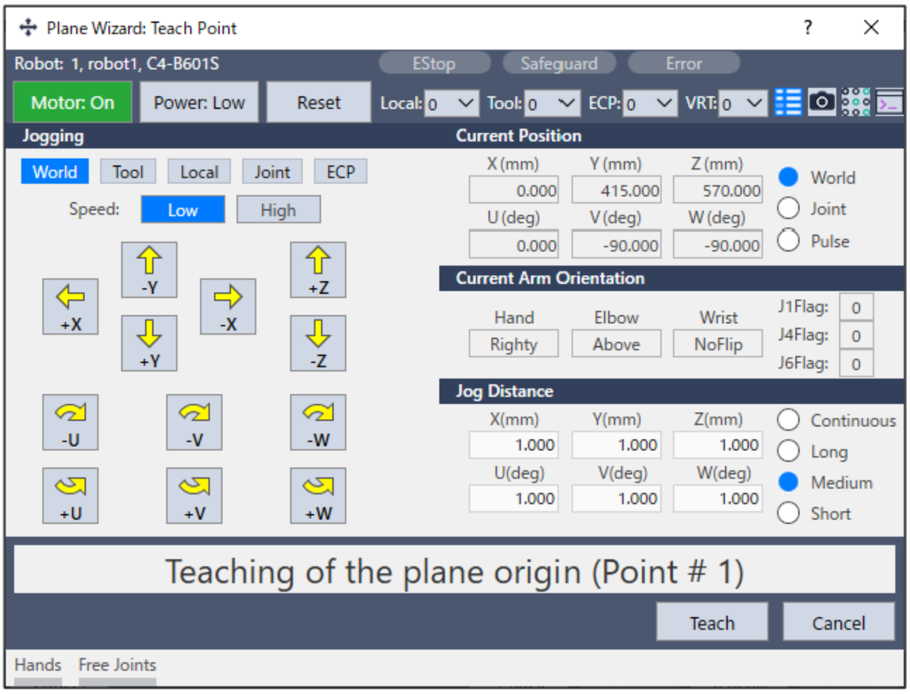
"SPEL+ Language Reference - Plane Statement"

4. Click the [Teach] button to show the [Teach plane origin point] page.

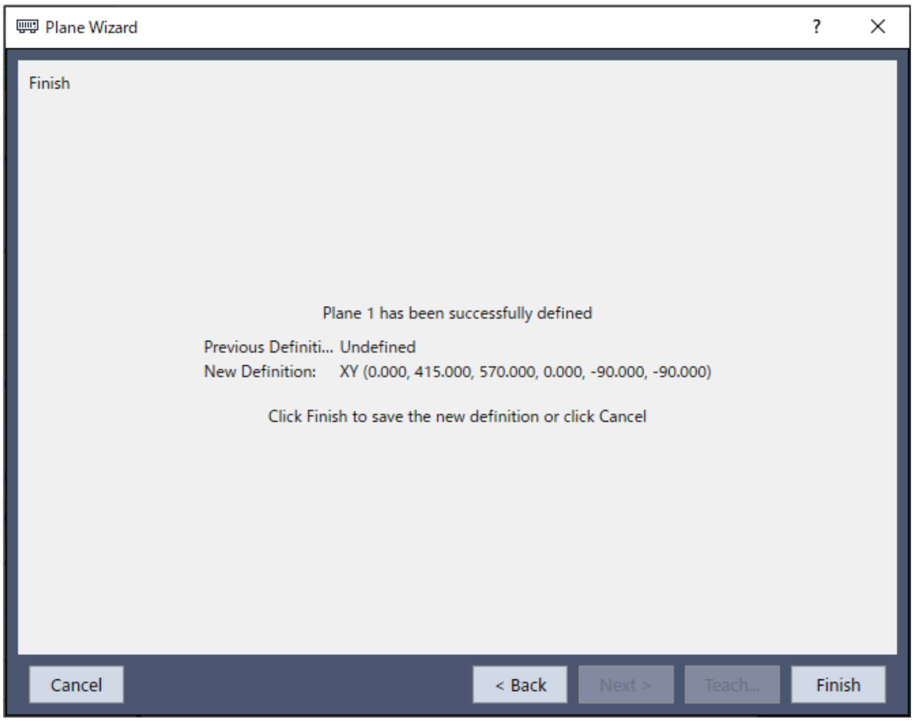
   - If the number of point to teach is "1":



i. Jog the robot to the reference point to teach the position of it. Click the [Teach] button. The following dialog appears.

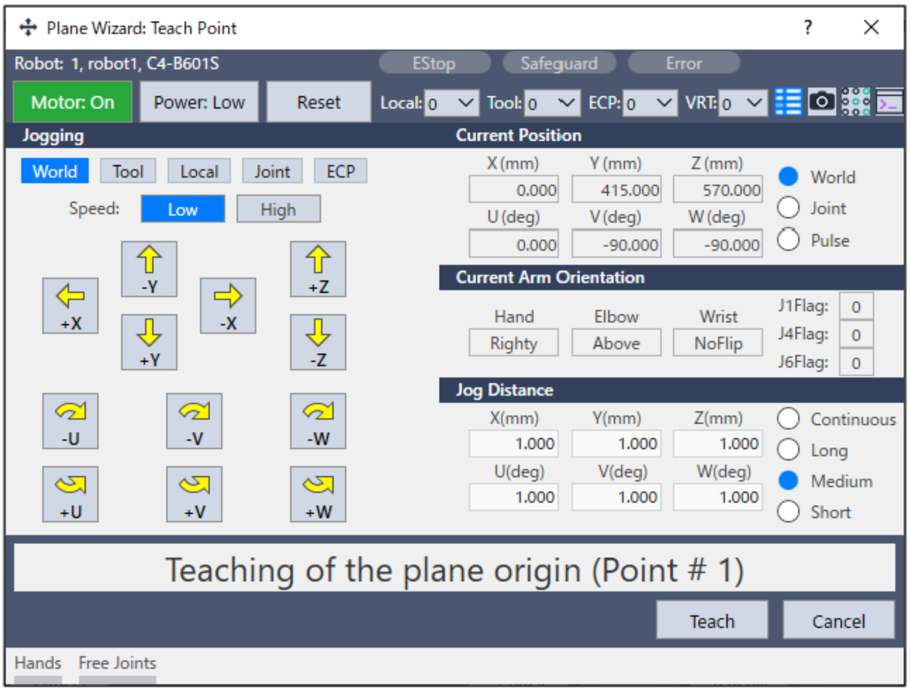


ii. The new plane definition is displayed as shown below. Click [Finish] to apply the new definition.



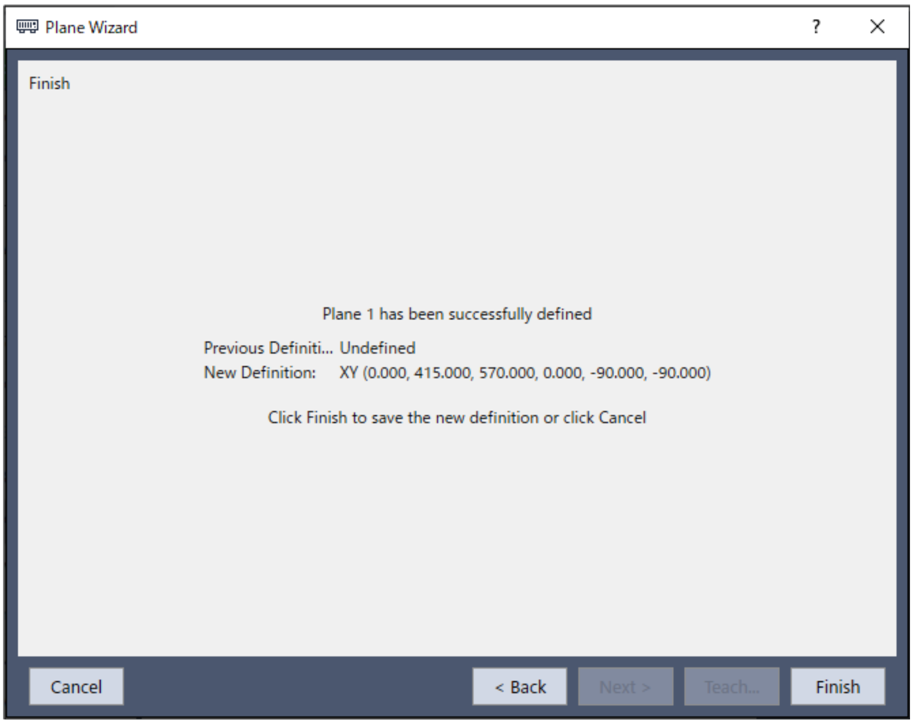
- If the number of point to teach is “3”:

i. Jog the robot to the reference point to teach the position of it (Point #1). Click the [Teach] button. The following dialog appears.



ii. Teach the X axis specified point (Point #2) and the Y axis specified point (Point #3) in the same way as the step 1).

5. The new plane definition is displayed as shown below. Click [Finish] to apply the new definition.



6.12.1.12 [Tools]-[Robot Manager]-[Weight] Panel

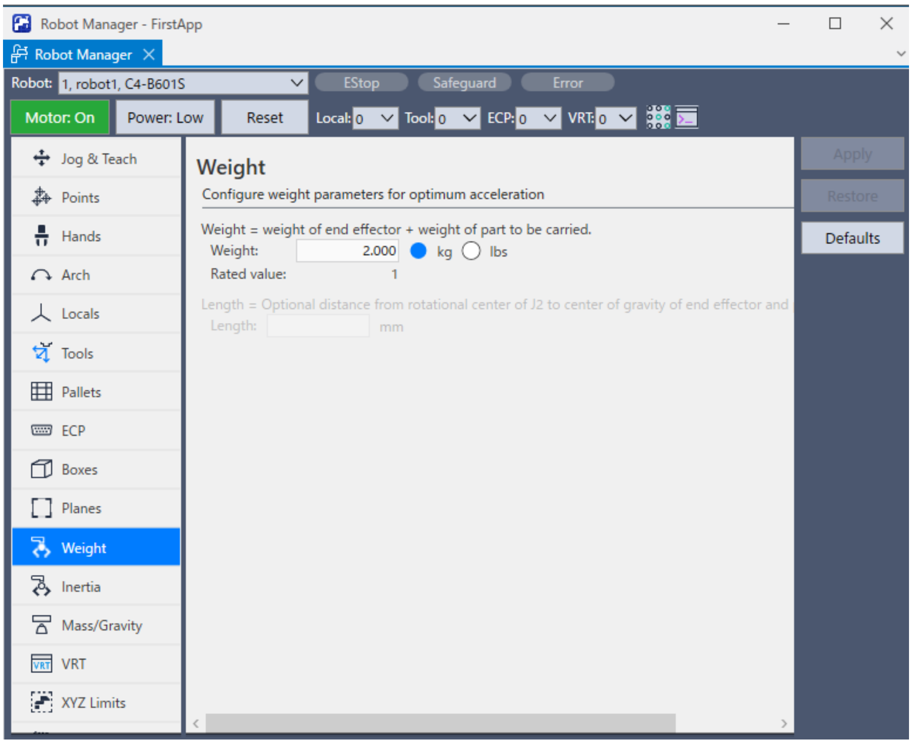
This page is for changing the Weight parameters for the robot.

For more details on Weight settings, refer to the following manual:

"SPEL+ Language Reference - Weight Statement"

You can also set by following "Weight, Inertia, and Eccentricity/offset Measurement Utility". See details below.

Weight, Inertia, and Eccentricity / Offset Measurement Utility



Item	Description
Weight	Type in the new total weight of the payload on the robot.
kg/lbs	Select which unit you would like to view. (kg/lbs)
Length	Change it when you use SCARA robots with special specifications which length of Arm #2 has been changed. The arm length is the distance from the center of the second axis to the center of the third axis.
Apply	Sets the current values.
Restore	Reverts to the previous values.
Defaults	Displays factory default settings.

6.12.1.13 [Tools]-[Robot Manager]-[Inertia] Page

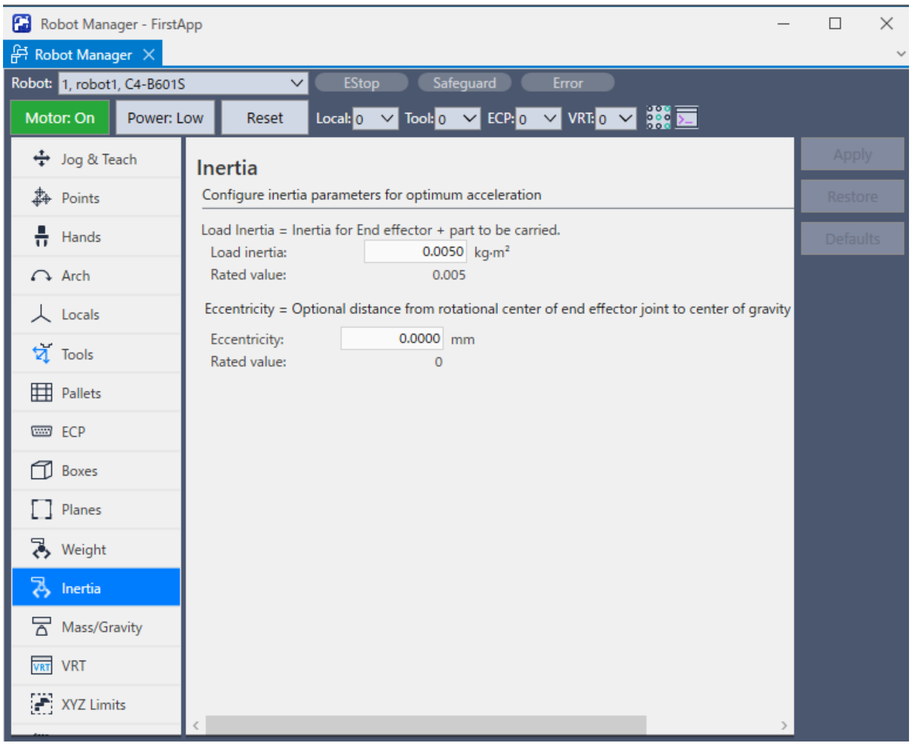
This page is for changing the Inertia parameters.

For more details on Inertia settings, refer to the following manual:

"SPEL+ Language Reference - Inertia Statement"

You can also set by following "Weight, Inertia, and Eccentricity/offset Measurement Utility". See details below.

Weight, Inertia, and Eccentricity / Offset Measurement Utility



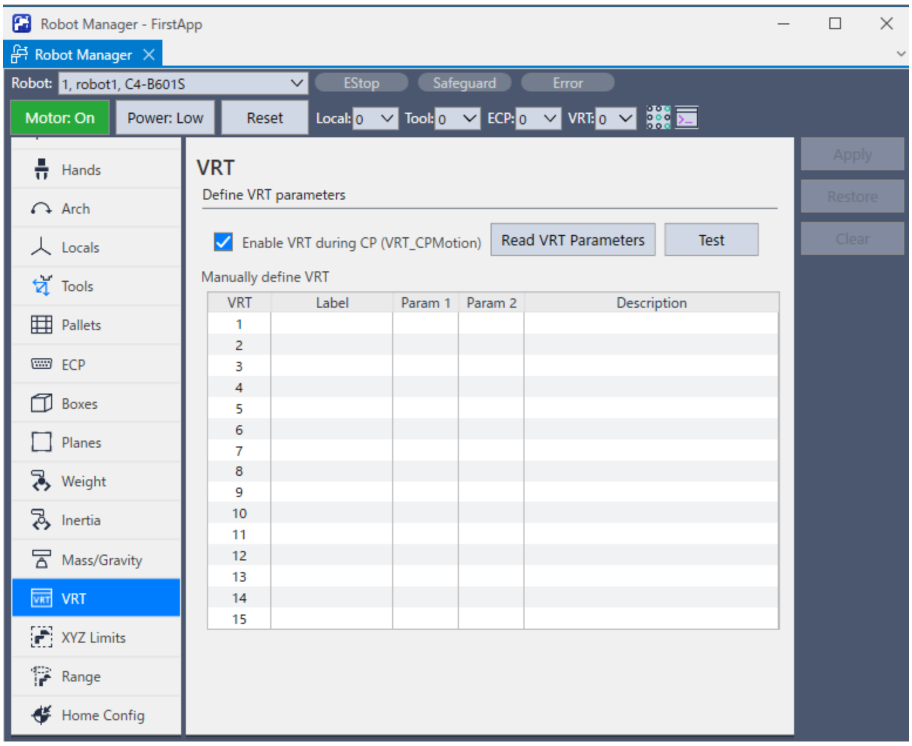
Item	Description
Load inertia	Type in the new load inertia of the payload on the robot in kg· m^2. This includes the inertia of end effector plus the part to be carried.
Eccentricity	Type in the new eccentricity value in millimeters. This is the distance from rotational center of joint 4 to the center of gravity of end effector and part.
Apply	Sets the current values.
Restore	Reverts to the previous values.
Defaults	Displays factory default settings.

6.12.1.14 [Tools]-[Robot Manager]-[VRT] page

This page is for changing the VRT parameters for the robot.

For details, refer to the following manual.

"Vibration Reduction Technology"



If the [VRT] tab is not displayed, the VRT Option is not set. Please refer to the following to configure Option settings.

Installing Controller License

Values set in the Robot Manager are saved even when the controller is turned off.

Navigating the grid

Press the [Tab] key to move to the next cell. Press the [Shift] + [Tab] key to move to the previous cell. Press the arrow [↑] or [↓] key to move to the cell above or below.

Item	Description
Enable VRT during CP operation (VRT_CPMotion)	Specify whether to enable or disable the VRT function. (When CP is in action) Default: Disabled
Read VRT Parameters	Start the VR software. The VR software must already be installed.
Test	Start VR Unit Check (software). VR Unit Check can be used to check for VR unit malfunctions. For details, refer to the following manual. ""Vibration Reduction Technology - Checking the VR Unit" The VR software must already be installed.
Label	Sets the label for the selected VRT number. (Optional)
Param1	Sets the VRTParam1 value. Setting range: 100 to 500
Param2	Sets the VRTParam2 value. Setting range: 100 to 500 Set VRTParam2 to a value that varies by over ±10% from VRTParam1.
Description	Sets a comment for the selected VRT number. (Optional)
Apply	Sets the current values. Allow several seconds for processing.
Restore	Reverts to the previous values.

Item	Description
Clear	Clears all selected values.
Select VRT number VRT: 0	Select the number set for the VRT parameter to apply the VRT function for this number to all subsequent robot actions. However, this setting will take precedence if a VRT command is executed during operations. 0: VRT OFF

6.12.1.15 [Tools]-[Robot Manager]-[XYZ Limits] Page

This page allows you to configure limits for X, Y and Z motion in the robot envelope.

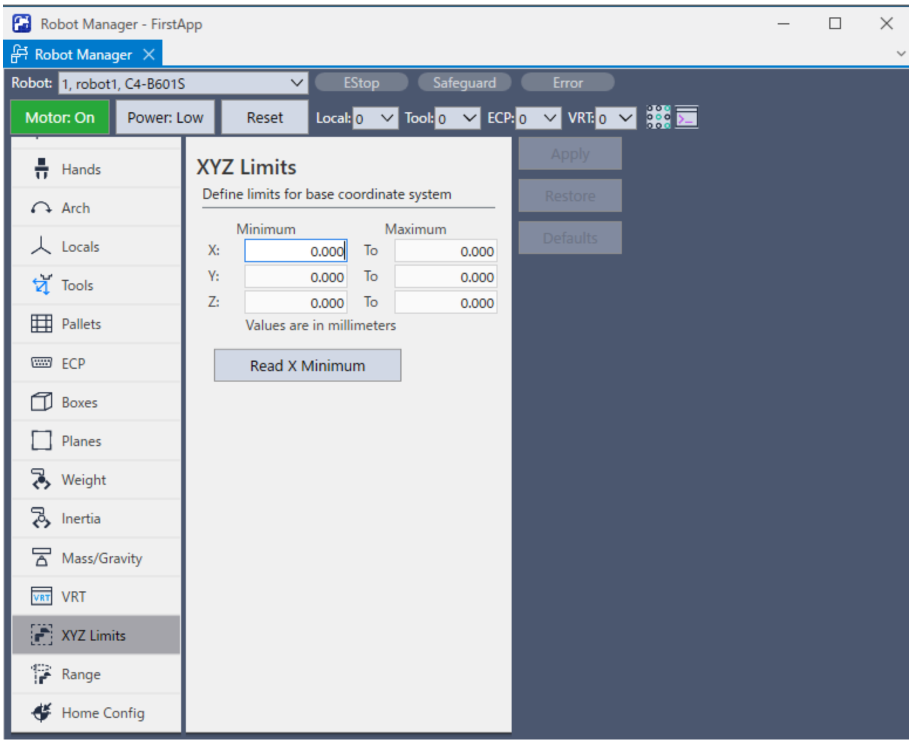
For more details on XYZ limits, refer to the following manual:

"SPEL+ Language Reference - XYLim Statement"

KEY POINTS

When using Safety Function (the Controller with Safety Board), the Safety Limited Position (SLP) of the Safety Function can be used. Set them using the Safety Function Manager. For details, refer to the following manual.

"Robot Controller Safety Function Manual Setting Safety Limited Position (SLP)"



Item	Description
X, Y, Z	Type in the minimum and maximum X, Y, and Z limit values. Setting both values to zero disables the limits.
Read Current	Click this button to read the value from the current robot position. The button text shows the axis and minimum or maximum depending on which text field has the current focus.
Apply	Sets the current values.

Item	Description
Restore	Reverts to the previous values.
Defaults	Reverts back to the default settings.

6.12.1.16 [Tools]-[Robot Manager]-[Range] Page

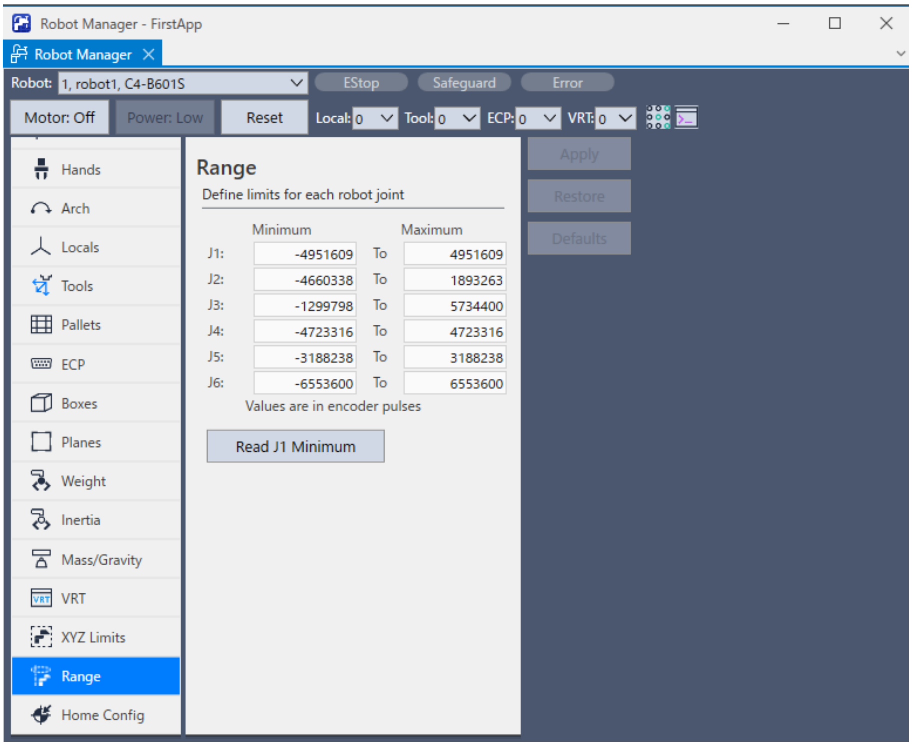
This page allows you to configure the robot joint software limits.

For more details on Range settings, refer to the following manual:

"SPEL+ Language Reference - Range Statement"

When using Safety Function (the Controller with Safety Board), the Safety Limited Position (SLP) of the Safety Function can be used. Set them using the Safety Function Manager. For details, refer to the following manual.

"Robot Controller Safety Function Manual Setting Safety Limited Position (SLP)"



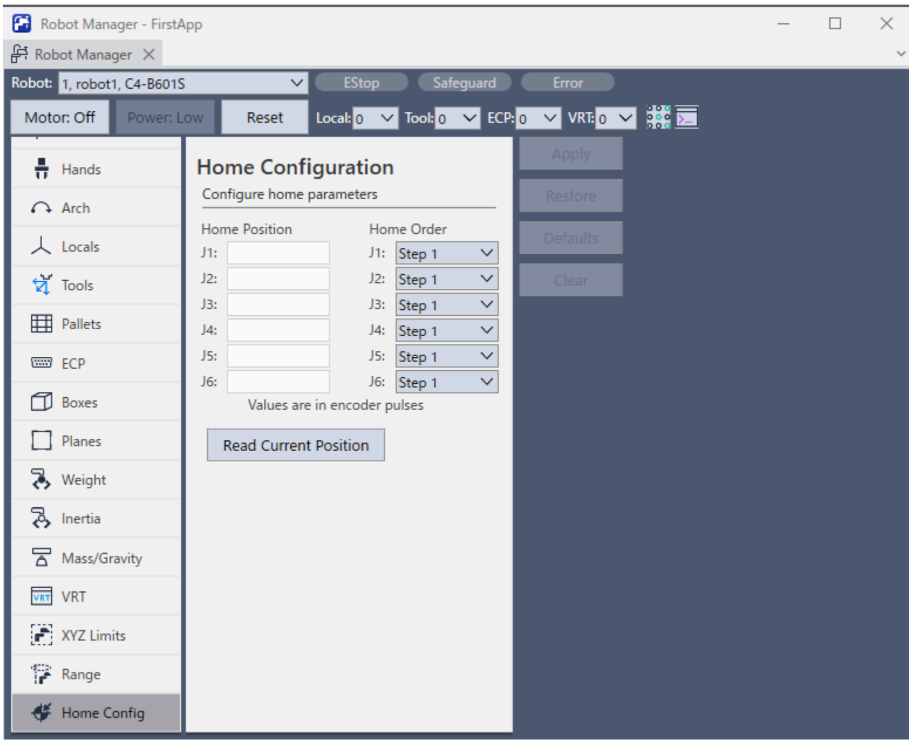
Item	Description
J1 - J6	Type in the minimum and maximum encoder pulse values for each joint.
Read Current	Click this button to read the current joint value of the robot into the current field. The button text will change depending on which text field has focus.
Apply	Sets the current values.
Restore	Reverts to the previous values.
Defaults	Reverts back to the default settings.

6.12.1.17 [Tools]-[Robot Manager]-[Home Config] Page

Home Config allows you to configure the optional user home position.

For more details on the Home Config settings, refer to the following manual:

"SPEL+ Language Reference - HomeSet Statement., Hordr Statement"



Changing home position

When you select the [Home Config] tab, the current home position is read from the robot controller and displayed in the text boxes. If the home position has never been defined, then the text boxes will be blank.

To define the home position, you can enter an encoder position value for each of the four robot joints in the text boxes. Alternatively, you can select the [Jog & Teach] page to jog the robot to the desired home position, then select the [Home Config] page and click the [Read Current Position] button to read the current encoder position values.

Changing home order

[Home Order] specifies the order in which each joint moves to the home position. More than one joint can be homed in the same step.

Testing home

After making changes to the home position, open the [Robot Manager]-[Jog & Teach]-[Execute Motion] tab, and then click the [Home] button from the motion command.

Item	Description
Read Current	Click this button to read the current position encoder pulse value into the currently selected text field. The button text will change according which text field is selected.
Apply	Sets the current values.
Restore	Reverts to the previous values.
Defaults	Set the value of the [Home order] group box to the default value.
Clear	Clears all values in the [Home Position] group box.

6.12.2 [Command Window] Command (Tools Menu)

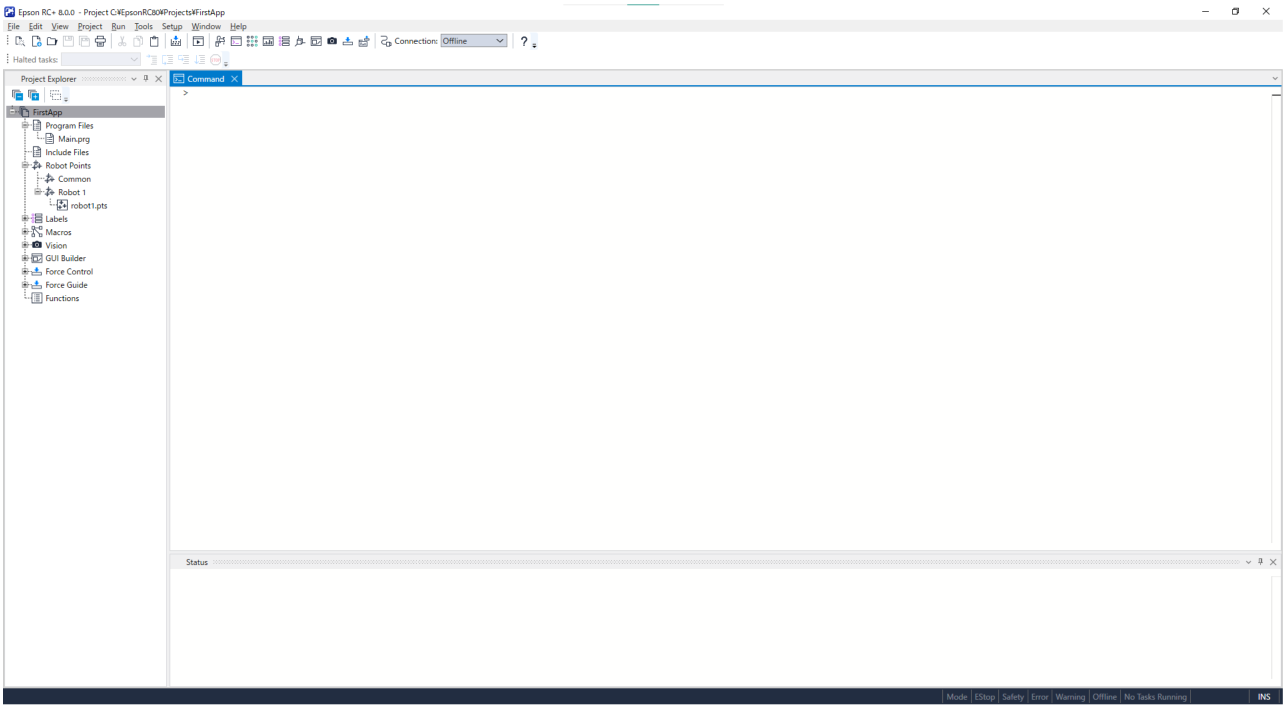
: Ctrl+M

You can execute SPEL+ commands from the robot controller and view the results.

Open command window

To open the Command window by either of the following methods.

- Select the Epson RC+ 8.0 menu-[Tools]-[Command Window]
- Or, click on the  [Command Window] button on the toolbar.
- Or, press the [Ctrl]+[M] keys



Set the Command window input format

In the Epson RC+ 8.0 menu, select [Setup]-[Preferences]-[Command Window] to change input format setting. See details below.

[Setup]-[Preferences]-[Command Window] Page

To execute SPEL+ commands from the Command window

1. Type in the desired command after the prompt (>). Commands can be entered in upper or lower case.
2. Press the [Enter] key to execute the command.
3. Wait for the prompt to return before typing in a new command.

   When an error occurs, an error number will be displayed along with an error message.

If the input format is set to the terminal, you can use the arrow keys or the mouse to move the cursor to any line in the window that starts with a prompt (>) character and execute it by pressing the [Enter] key.

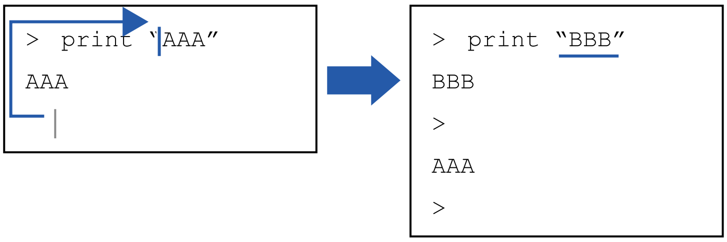
If the input format is set to command line, press the [↑] or [↓] arrow keys to scroll through the command history. Press the arrow keys to show the executed commands in sequence. Press the [Enter] key to execute.

Example: When executing the print "BBB" command after executing the print "AAA" command When you execute the print "AAA" command, the execution result will be displayed on the command window as shown below.

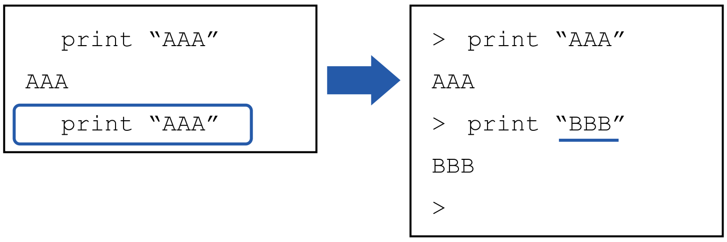
```
> print "AAA"
AAA
```

You can execute the print "BBB" command by diverting the executed command.

- When the input format is set to terminal, move the insertion point by the arrow keys to rewrite "AAA" of the executed command to "BBB."



- When the input format is set to command line, press the [↑] or [↓] arrow keys to display the previously executed commands in order. You can rewrite "AAA" of the displayed command to "BBB."



Command Window Keys

Key	Action
Ctrl+A	Select entire window.
Ctrl+C	Stop the program and initialize robot controller. If a robot motion command is in progress, the prompt will return when the command has been completed.
Ctrl+V	Execute Paste command. Paste from Clipboard to current selection.
Ctrl+X	Execute Cut command. Cut current selection and put in Clipboard.
Ctrl+Z	Undo last change.
Ctrl+W	Re-display last command line after the prompt. Only enabled in terminal format.
Ctrl+Home	Go to the top of the window. Only enabled in terminal format.
Ctrl+End	Go to last prompt at end of the window. Only enabled in terminal format.
?	Translates to "PRINT" when used as the first character of a command. This can be used to display variables or any statement that requires a PRINT command.
[↑] Key	Calls command history. Press this key to display the previous command. This is only enabled in command line format.

Key	Action
[↓] Key	Calls command history. Press this key to display the next command. This is only enabled in command line format.

6.12.3 [I/O Monitor] Command (Tools Menu)

: Ctrl+I

The I/O Monitor window lets you monitor all controller hardware inputs and outputs and also memory I/O. You can display up to five screens (three screens in standard view, and two screens in user view) side by side.

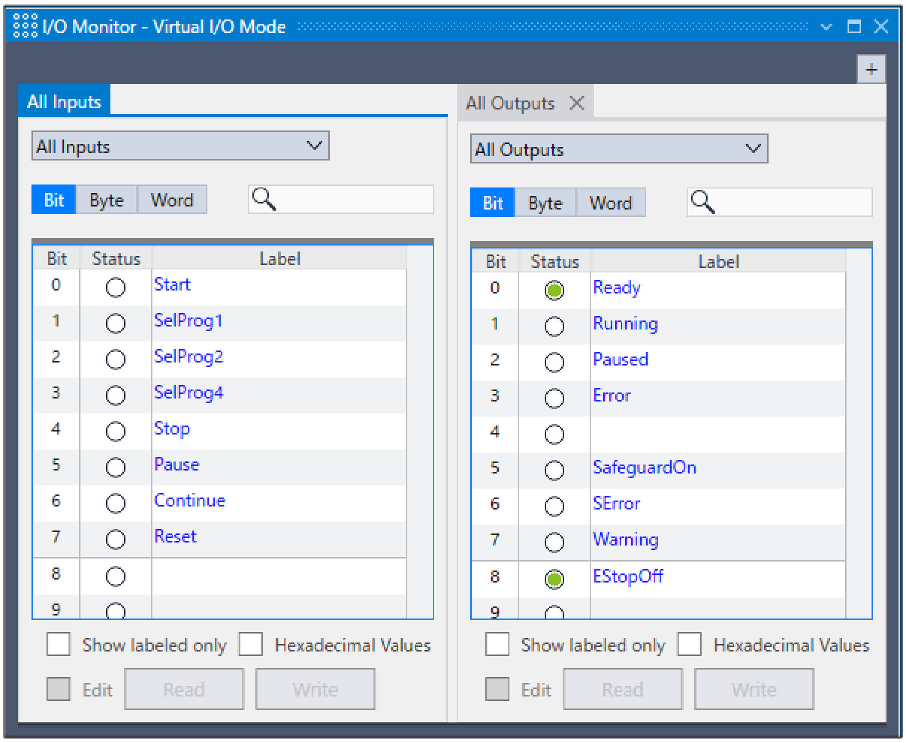
A custom view can have any combination of input, output, or memory elements you want to monitor.

- Labels that have been defined using the [I/O Label Editor] are displayed next to each bit, byte, or word.
- After the [I/O Monitor] window has been opened, the input and output status for the current view is constantly updated.
- Mouse over any comment entered in the I/O Label Editor to display it as a tooltip.
- You can turn outputs on and off by double clicking on the output LED images in the Status column.
- Specify byte and word values to change the status of multiple bits at the same time.

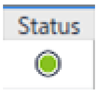
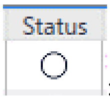
Open the I/O Monitor

Open the I/O Monitor by either of the following methods.

- Select the Epson RC+ 8.0 menu-[Tools]-[I/O Monitor]
- Or, click on the [I/ O Monitor] button on the toolbar.
- Or, type the [Ctrl] + [I] keys on the keyboard.



Item	Description
+	Adds a standard or user view screen.

Item	Description
	Searches for labels, and moves the cursor to the row of the label found.
Status (when [Bit] is selected)	Double-click the LED images to turn outputs on and off. When the virtual I/O is working, you can turn outputs on and off by double-clicking the output LED images in the Status column.  : on  : off
Value (when [Byte], [Word] are selected)	Displays the status of each bit summarized in bytes and words.
Show labeled only	Only shows I/Os with preset labels.
Hexadecimal Values	To view bytes and words in hexadecimal format, check the [Hexadecimal Values] checkbox.
Edit	Select the [Edit] checkbox to change the value. Only bytes and words can be used.
Read (when [Edit] is selected)	Reads the current I/O status and value.
Write (when [Edit] is selected)	Sets current values after changes have been made.

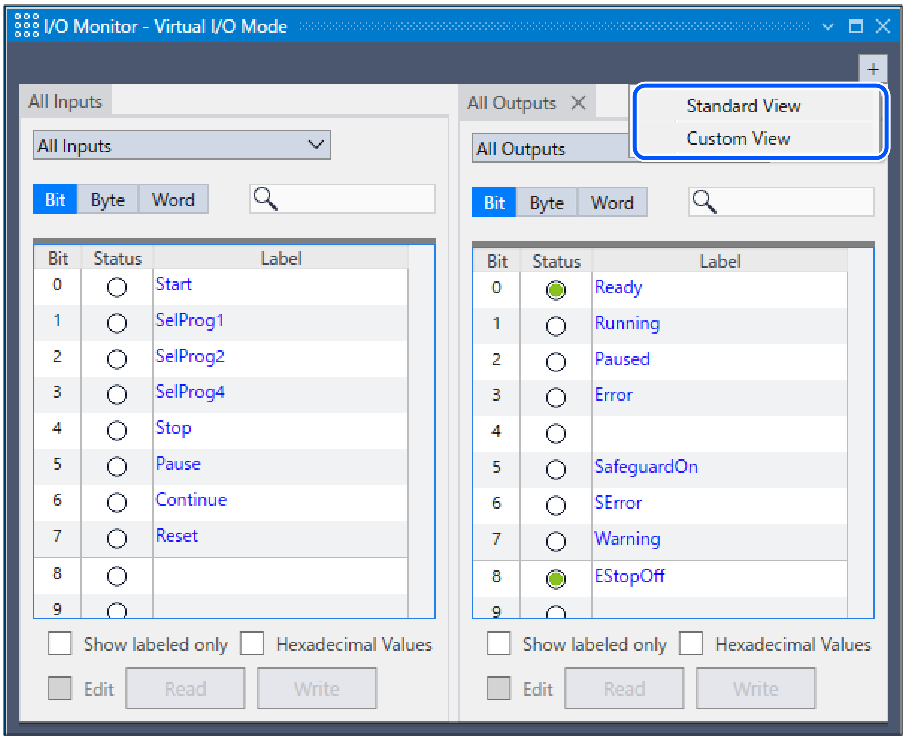
TIP

Enter a bit, byte, or word number on the active window to move the cursor to the corresponding row.

Using the I/O Monitor

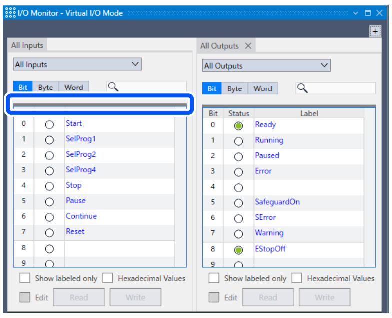
- Add view screen

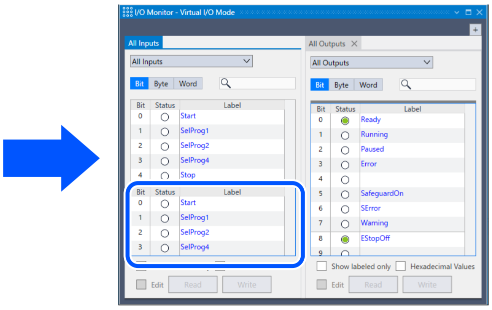
  The I/O Monitor window can display up to five screens (three screens in standard view, and two screens in custom view) side by side. There are two standard views available by default. To add a view, click the [+] button on the top right of the window.



- Split screen

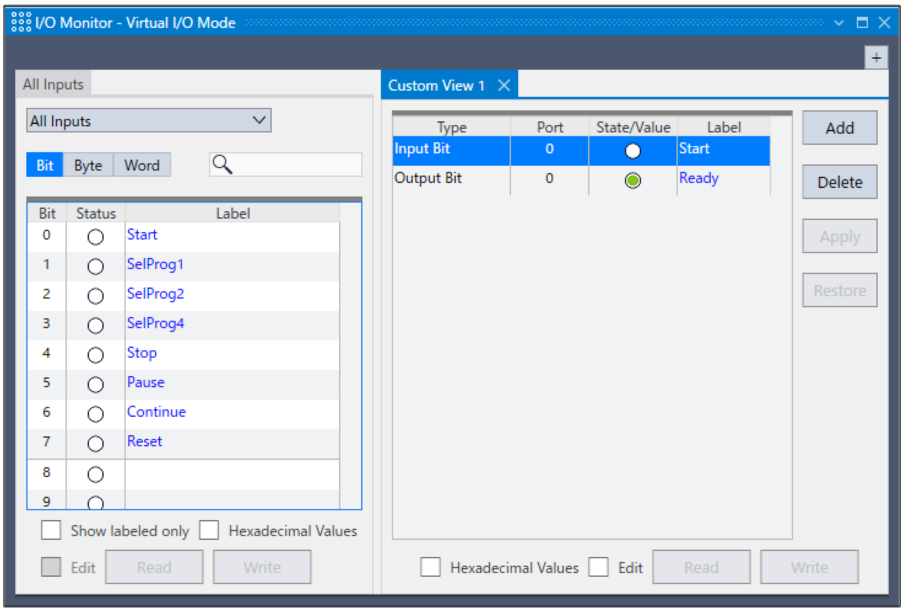
Drag the split bar at the top of the grid down to split each grid into two scroll regions.





Using Custom View

1. Select the [Custom View] tab.



TIP

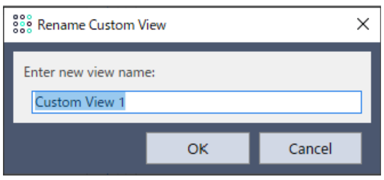
- If a Custom View does not appear, click the [+] button on the top right of the window to select it.
- Right-click on a custom view tab to rename the tab.

2. Click the [Add] button to add a new row to the list.
3. Click the Type column, and then select an I/O type.
4. Click on the Port column, and then select a port number.
5. Add more rows as needed by repeating steps 2 to 4.
   - [Apply]: Save changes.
   - [Delete]: Delete row.
   - [Restore]: Cancel changes.

To rename a view

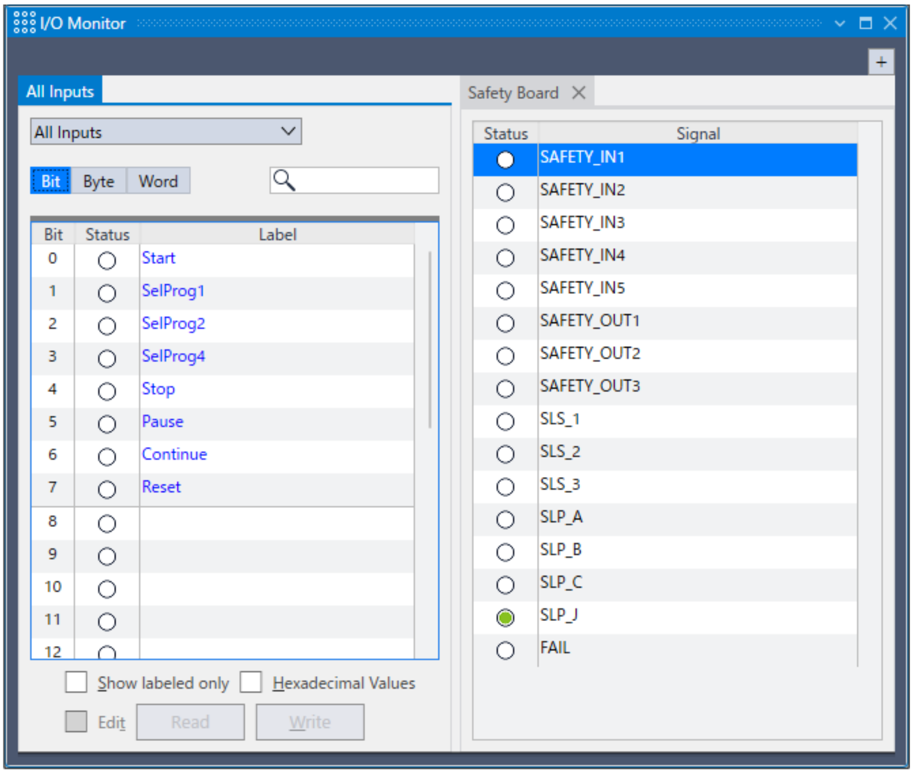
You can rename custom view tabs.

1. Select the [Custom View] tab. If no custom views appear, click the [+] button on the top right of the window to select it.
2. Right click on the view tab and select [Rename].
3. Rename Custom View will be displayed. Type in a new name, and then click [OK].



Using Monitor of the Safety Board

Select the [Safety Board] tab. I/O status of the safety board will be shown.



KEY POINTS

- To show the monitor of the safety board, connect Epson RC+ to the Controller installed the safety board.

  For the setting of safety functions, refer to the following manual.

  "Robot Controller Safety Function Manual"
- The following cannot be operated with I/O monitor of the safety board.
  - Output on/off
  - Displaying custom view
  - Renaming the tab name

Signal for the safety board and the status

	Signal	State	Notes
Inputs and Outputs	SAFETY_IN1, SAFETY_IN2, SAFETY_IN3, SAFETY_IN4, SAFETY_IN5	Signal level for safety input is ▪ High: turning LED on ▪ Low: turning LED off	Safety input signal is negative logic (Active Low).
	SAFETY_OUT1, SAFETY_OUT2, SAFETY_OUT3	Signal level for safety output is ▪ High: turning LED on ▪ Low: turning LED off	Safety output signal is negative logic (Active Low).

	Signal	State	Notes
State	SLS_1, SLS_2, SLS_3	Safety Limited Speed is ▪ Enabled: turning LED on ▪ Disabled turning LED off	For the violation of the limited speed, refer to the system history.
	SLP_A, SLP_B, SLP_C	Safety Limited Position is ▪ Enabled: turning LED on ▪ Disabled turning LED off	For the violation of the limited position, refer to the system history.
	SLP_J	Soft Axis Limiting ▪ Enabled: turning LED on ▪ Disabled turning LED off	Soft axis limiting is always enabled outside of TEACH mode.
	FAIL	Fault detection for safety board ▪ Abnormal: turning LED on ▪ Normal: turning LED off	For the information on the fault, refer to the system history.

KEY POINTS

For more details on the PG board monitor, refer to the following manual:

"PG Motion System - Show PG Signal Status"

6.12.4 [Task Manager] (Tools Menu)

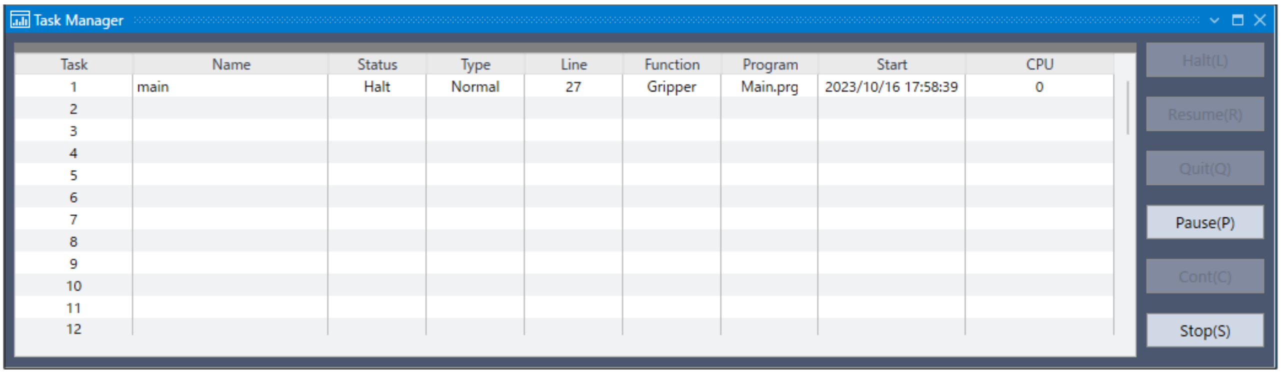
: Ctrl+T

The Task Manager window allows you to Halt (suspend), Resume (continue), and Quit (abort) tasks.

Start Task Manager

Open Task Manager using either of the following methods.

- Select the Epson RC+ 8.0 menu-[Tools]-[Task Manager]
- Or, type Ctrl + T.
- Or, click on the [Task Manager] button on the toolbar.



Item	Description
Task	Task number from 1 to 32, background tasks from 65 to 80, and 11 trap tasks.
Name	Name of the function that was started as a task.
Status	Current task status: Run, Wait, Halt, Pause, Aborted, Finished.
Type	Task types ■ Normal: This task is a normal task ■ NoPause: This task does not pause with Pause statement or when Pause input or Safety Door open occur. ■ NoEmgAbort: This task continuously processes during the Emergency Stop or error occurrence.
Line	Current task line number.
Function	Current task function name.
Program	Current task program name.
Start	The date and time that the task was started.
End time	The date and time that the task ended.
CPU	CPU load factor of each task. This function assists problem detection of the user created tasks.
Halt	Suspends the selected task. The halted task can be resumed by the [Resume] button. The [Halt] button is only available when a task is running (status is Run). When Halt is executed, the [Resume] button will be enabled. If a motion command associated with Halt is executed, the motion will be completed before the task reaches the Halt state. The task also temporarily stops when the task is NoPause type or NoEmgAbort type.
Resume	After one or more tasks are suspended with the [Halt] button, clicking the [Resume] button to make the halted tasks continue where they left off. First, a confirmation dialog is displayed.
Quit	This button stops the selected task permanently. You cannot resume a task once you have executed Quit. To restart the task, you must start it from within a program or from the Run window. The task also stops when the task is NoPause type or NoEmgAbort type.
Pause	This button pauses tasks that can be paused. After pause, you must use either [Cont] or [Stop]. The task does not pause when the task is NoPause type or NoEmgAbort type.
Continue	This button continues all tasks that were paused with the [Pause] button.
Stop	Stops all tasks.

Operation

The Task Manager is used for suspending, resuming, stepping, and stopping tasks. When Task Manager is started, you will see a grid containing status information for 32 tasks standard tasks and 11 trap tasks. Also, you will see the status information of 16 background tasks if the background task is enabled. There are 8 items shown for each task. To view all of the columns, use the scroll bar or resize the window.

In the following Example 1, the function repeats until the standard input I/O bit port 1 turns ON.

Since Sw() is the command in which the tasks are not switched, this task occupies the process. It may affect other user tasks or the whole system of the Controller. In order to specify such tasks, use the CPU load factor display.

Restrictions

Displayed values do not guarantee the accuracy. Due to limitations of measuring method, some differences are included. The load factor of the properly created program is minimal. Also, in a program like Example 2, commands are executed by other system tasks. Therefore, the load factor is displayed as "0".

Example 1)

```
Function main
  Do
    Do
      If  Sw(1) = On Then Exit Do
    Loop
    Go P(0)
  Loop
Fend
```

Example 2)

```
Function main
  Do
    Print "TEST"
  Loop
Fend
```

To Halt, Step, Walk, and Resume a task

The [Halt] button will become active after you select a running task.

Click the [Halt] button to stop the task you selected for a moment.

After a task has been halted, the source code will be displayed and the next step will be indicated. You can click on the [Resume] button to resume execution. (You can also execute [Step Into], [Step Over], or [Walk] from the [Run] Menu.)

To Pause and Continue tasks

Pause allows you to "suspend" all tasks that can be suspended.

Click on the [Pause] button to pause available tasks. The robot will decelerate to a stop immediately.

After executing Pause, click on the [Cont] button to resume all suspended tasks.

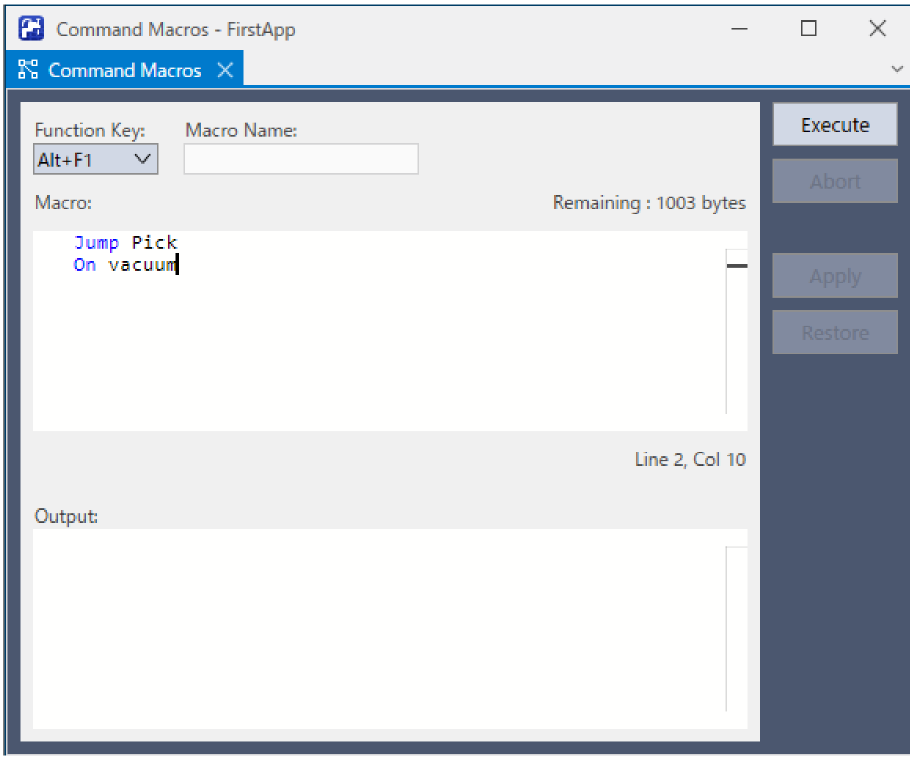
To view source code at the current execution line

Select a task row. Then right click and select [Go To Line]. The program editor will be opened at the current execution line

6.12.5 [Macros] (Tools Menu)

You can create SPEL+ command macros using the Macro Editor. Macros consist of one or more SPEL+ statements that can be executed from the command window. A macro statement may use global variables, I/O labels, and point labels. You can assign a macro to each of the Alt function keys except for Alt+F4, which is a Windows shortcut to close the application.

1. Select [Tools]-[Macros] to open the [Command Macros] dialog box.



2. Type the macro statements in the [Macro] text box.
    - Up to 512 characters can be entered for a macro statement. Any further characters after 512 characters will be deleted when pressing the [Enter] key.
    - Your macro entry can be up to 1023 bytes in size. Macro entries after 1024 bytes will be deleted when clicking the [Apply] button.
3. Click the [Apply] button to save changes.
4. Click [Execute] to run the macro.

You can use macros to move the robot and control I/O. The separate execute step is provided for safety.

To open a macro and execute it, type [Alt] + function key. Then click [Execute] to run it. Macros never execute by pressing the function key.

6.12.6 [I/O Label Editor] Command (Tools Menu)

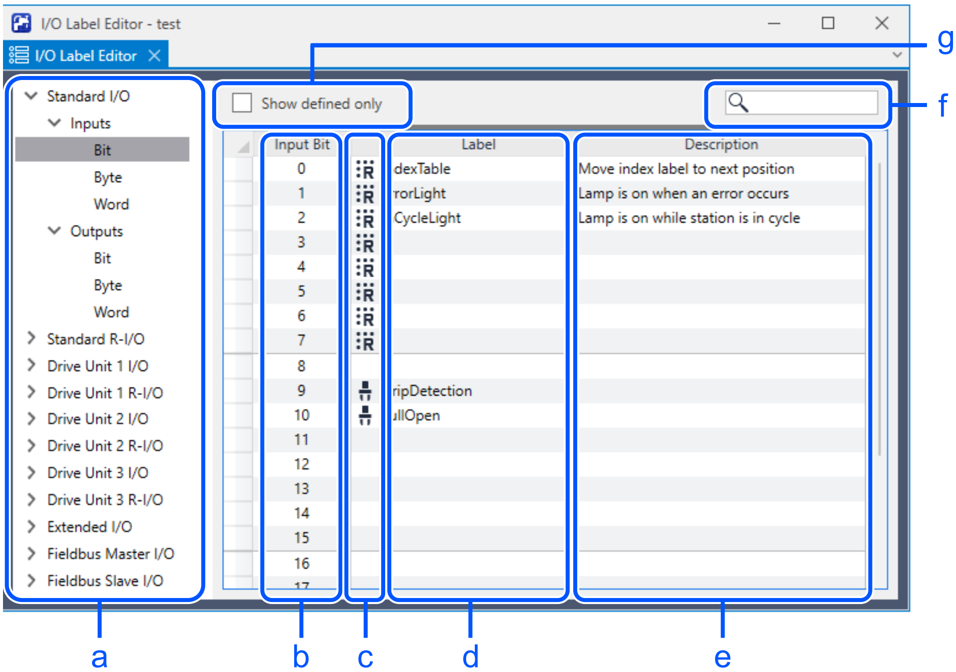
: Ctrl+L

The I/O label editor lets you define meaningful names for inputs, outputs, and memory I/O for each project. The labels can be used in your programs, from the Command window, or in macros. They are also displayed in the I/O Monitor window.

Open the I/O Label Editor

Open the I/O Label Editor using either of the following methods.

- Select the Epson RC+ 8.0 menu-[Tools]-[I/O Label Editor]
- Ctrl + L
- Or, click on the [I/O Label Editor] button on the toolbar.



Navigating the grid

Press the [Tab] key to move to the next cell. Press the [Shift] + [Tab] key to move to the previous cell. Press the arrow [↑] or [↓] key to move to the cell above or below.

> **TIP**
>
> Enter a bit, byte, or word number on the active window to move the cursor to the corresponding row.

Symbol	Description
a	Shows the types of I/O. For each type of I/O you can view and edit labels for bits, bytes (8 bits), and words (16 bits).
b	This shows the bit, byte, or word number for the I/O displayed.
c (when Bit is selected)	Shows the types of defined I/O. Hovering the mouse over the icon will show Label as a tool tip. ▪ : Remote I/O ▪ : Hand I/O
d	Sets the label. You can type in up to 32 characters for a label. Label characters can be alphanumeric or underscore.
e	Enter a description associated with the label. If you add a description to a comment, then the description will be displayed as a tool tip on the I/O Monitor.
f	Searches for labels, and moves the cursor to the row of the label found.

Symbol	Description
g	Only shows registered I/O labels.

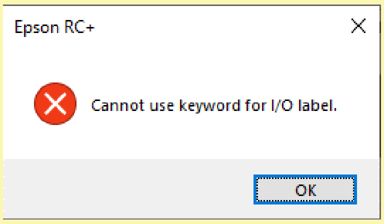
KEY POINTS

- The I/O Label Editor shows all available I/O types on your controller.
- If the Virtual I/O in [Setup]-[System Configuration]-[Controller]-[Preferences] is enabled, the I/O Label Editor will display all I/Os. For example, you can edit Fieldbus I/O labels, but you may not have a Fieldbus board installed in the controller.

CAUTION

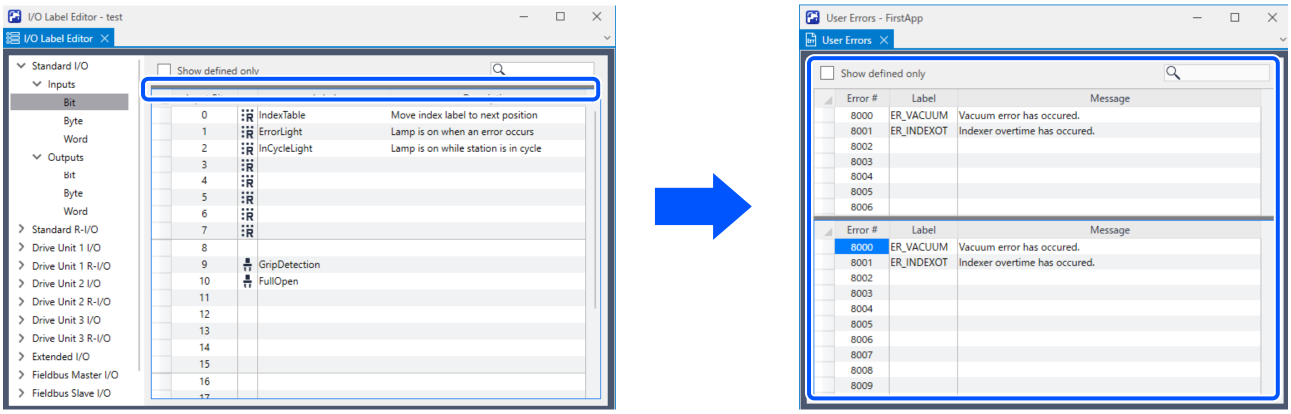
When you specified the string (SPEL+ commands and so on) that has other roles in Epson RC+ 8.0 as a label, it may have unexpected consequences. Specify a unique label name so that it does not duplicate with those strings.

If you set a keyword, the following dialog will be shown.



Split screen

Drag the split bar at the top of the grid down to split each grid into two scroll regions.



To add or edit a label

1. Select the type of I/O label. Shows a label in the spreadsheet. The number of rows in the spreadsheet equals the number of bits, bytes, or words available for the type you have selected.
2. Select the row to add or edit, and then enter a label. Type in the label, which can be up to 32 alphanumeric characters without any spaces. Optionally, you can type a description for the label in the [Description] field.
3. Save the label.

KEY POINTS

After adding or editing labels, save the changes in the Epson RC+ 8.0 menu by executing [Save] from the [File] menu or by clicking on the  [Save all files] toolbar button. If any duplicate labels are detected, an error message will be displayed and the save operation will be aborted. You must correct the duplication before you can save the labels successfully.

Cut and paste labels and descriptions

You can cut and paste labels and descriptions by selecting them with the mouse, then executing [Copy], [Cut], and [Paste] from the Epson RC+ 8.0 [Edit] menu.

You can also cut and paste entire rows using the following steps:

1. Select one or more rows by using the row selectors.

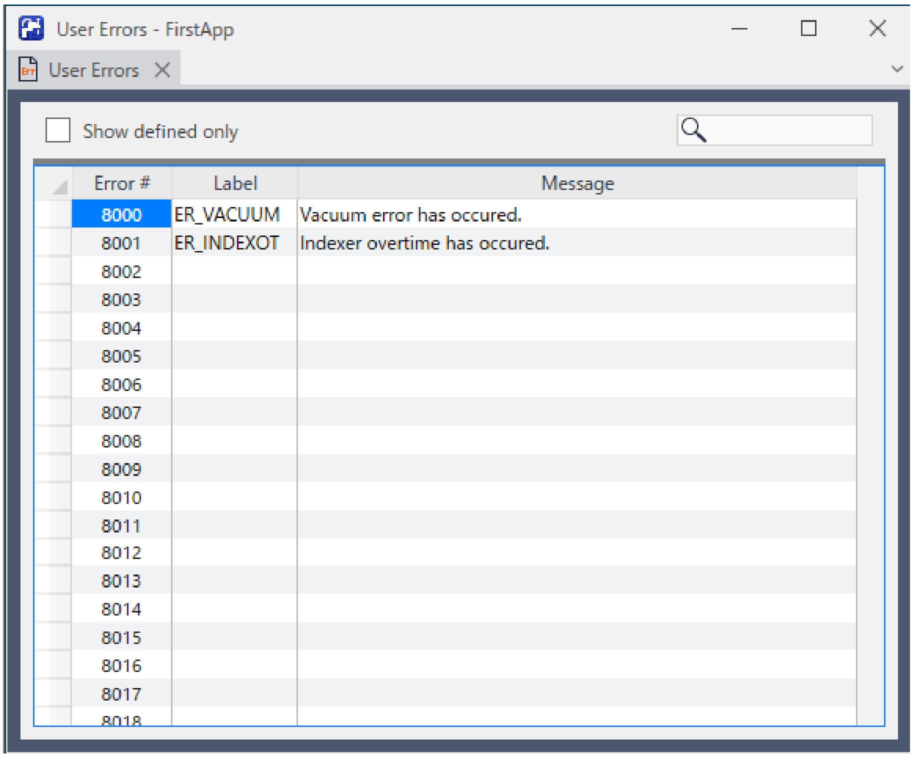
   Execute either the [Cut] or [Copy] command from the Epson RC+ 8.0 [Edit] menu. When selecting multiple rows, hold down the [Shift] or [Ctrl] key while selecting rows with the mouse.

2. Select the row where you want to start the paste by clicking the row selector on the left of the row.

3. In the Epson RC+ 8.0 menu, select [Edit]-[Paste].

6.12.7 [User Error Editor] (Tools Menu)

: Ctrl+U The User Error Editor allows you to define user errors.



Navigating the grid

Press the [Tab] key to move to the next cell. Press the [Shift] + [Tab] key to move to the previous cell. Press the arrow [↑] or [↓] key to move to the cell above or below.

> **TIP**
>
> Enter the error number to move the cursor to the corresponding row.

Item	Description
Show defined only	Only shows registered user errors.
	Searches for labels, and moves the cursor to the row of the label found.
Error	The user error number. User error numbers can be from 8000 to 8999.
Label	Sets the label. Labels can be up to 32 characters in length, containing both half-width alphanumeric characters and underscores.
Message	Enter the message that appears when an error occurs.

> **TIP**
>
> It is recommended that you use the ER prefix for each error label and use all caps for the label. This makes it easy to see error labels in your code.
>
> For example:

Error Number #	Label	Message
8000	ER_VACUUM	Vacuum error has occurred.
8001	ER_INDEXOT	Indexer overtime has occurred.

In your program code, use the Error statement to generate a user error.

For example:

```
Function main
  On Vacuum
  Wait Sw(VacOn), 1
  If TW = 1 Then
    Error ER_VACUUM
  EndIf
Fend
```

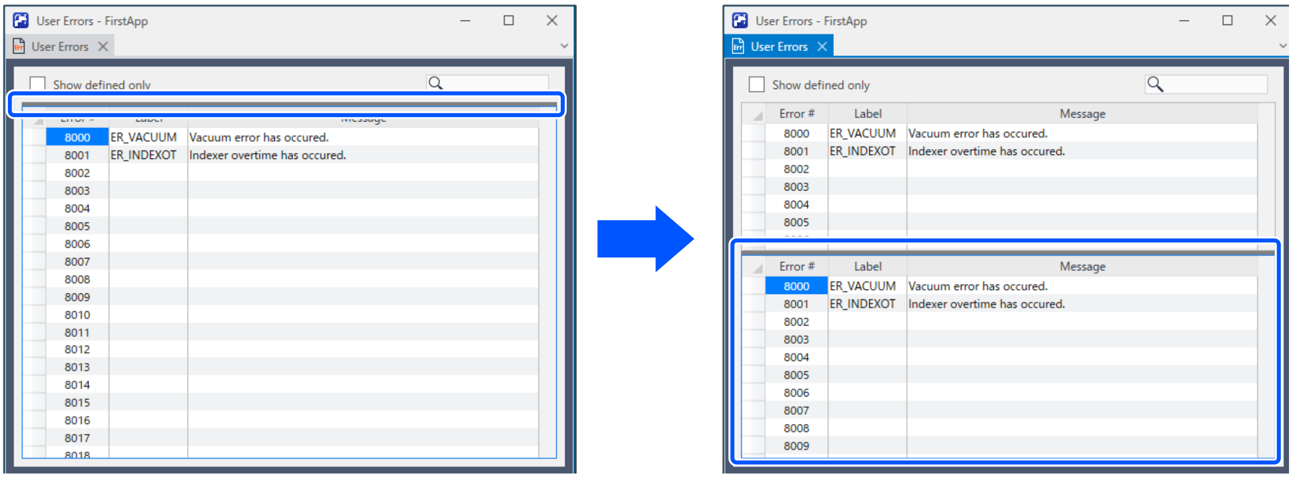
The user error information is stored in the current project directory in a file called UserErrors.dat.

You can use [Import File] from the [File] menu to import user errors from other projects.

After adding new error definitions, save the changes by executing Save from the [File] menu or by clicking on the  [Save all files] on the toolbar.

Split screen

Drag the split bar at the top of the grid down to split each grid into two scroll regions.



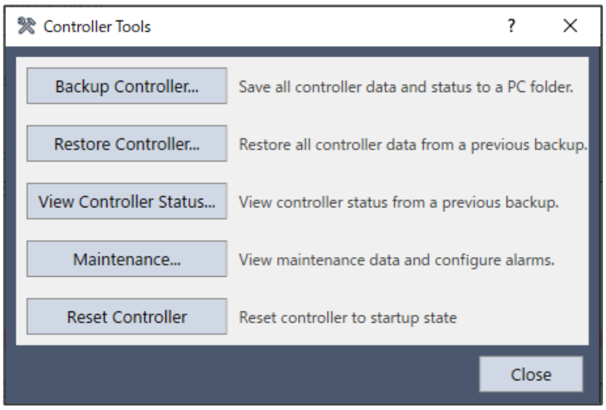
6.12.8 [Controller] Command (Tools Menu)

Select Controller from the [Tools] menu to open the [Controller Tools] dialog box.

From the [Controller Tools] dialog box, you can save and restore the complete controller configuration and the project using the [Backup Controller] and [Restore Controller] commands. You can also save and view controller status, and reset the controller.

Before servicing the system, you should execute [Backup Controller] and store the system configuration on an external media such as a USB memory key.

If required, you can use [Restore Controller] to restore previously stored data.



Backup Controller

Use Backup Controller to save controller configuration data on your PC.

The current status is saved in a folder containing several files. The controller configuration settings, task status, I/O status, robot status etc. are saved in these files. This is useful for users to send a snapshot of the controller status to the supplier of your region or to Epson technical support, should the need arise.

KEY POINTS

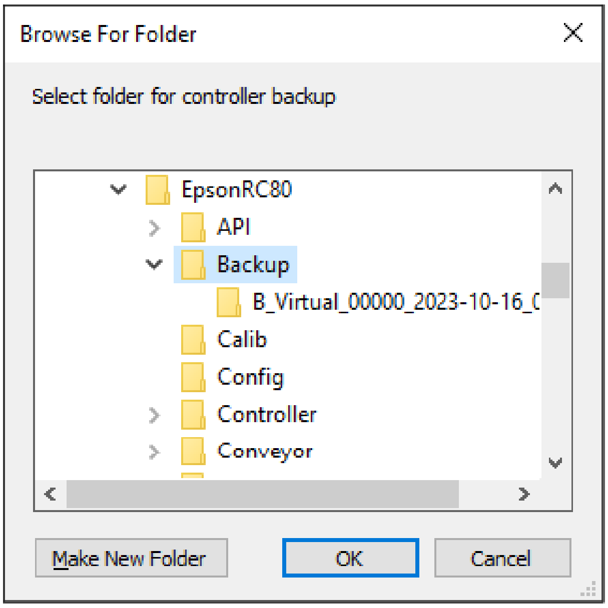
- Backup Controller is the same as Backup Controller Function and Controller Status Storage Function that saves the state to the USB memory connected to the controller.

  Each is saved in the following folder.

  - Epson RC +: B_Controller type_Serial number_Date and time
  - Controller: BU_Controller type_Serial number_Date and time

- This preference allows you to configure whether project files are saved or not when the controller status is exported. For details, refer to [Setup]-[System Configuration]-[Controller]-[Preferences].
- With RC800 series Controllers, you can select whether to save Controller data when an error occurs. Select [Include controller data in backup when error occurs] from the pull-down menu to the right of the [Backup Controller] button.

1. Select Controller from the [Tools] menu to open the [Controller Tools] dialog box.
2. Click on the [Backup Controller] button to open the [Browse For Folder] dialog box.



3. Select the disk drive and parent folder where you want to save the information.

   You can create a new parent folder by clicking the [Make New Folder] button.

4. Click the [OK] button.

   A new folder containing the backup files will be created. The selected folder will be named “B_” followed by the controller type, the controller serial number, and the date / time.

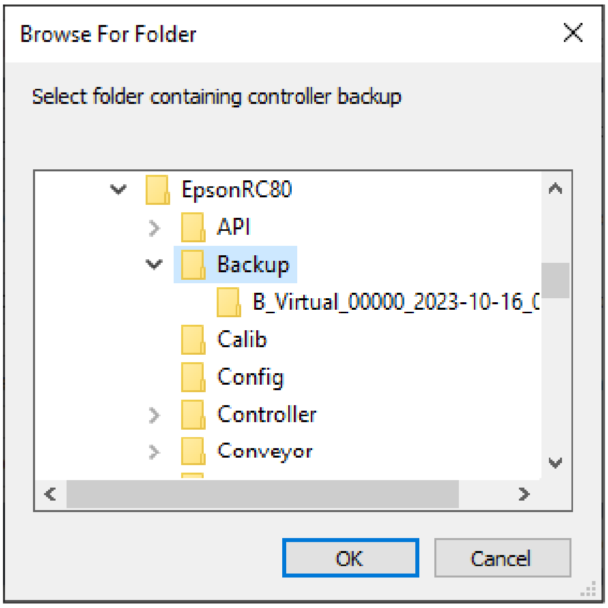
Restore Controller

Use Restore Controller to load controller settings from previously saved backup data. You cannot restore the controller data while tasks are running. If you attempt to do so, an error message will be displayed.

To restore controller configuration:

1. Select Controller from the [Tools] menu to open the [Controller Tools] dialog box.

2. Click on the [Restore Controller] button to open the [Browse For Folder] dialog box.



3. Select the drive and folder where the information is stored.

B_Controller type_Serial number_Date and time

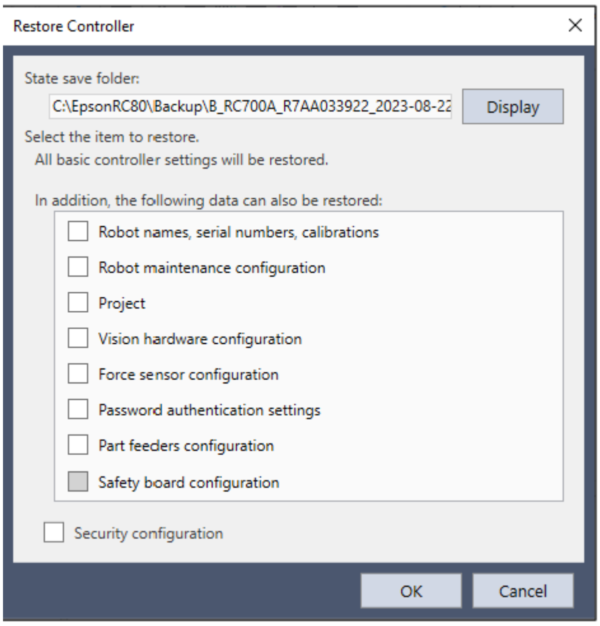
KEY POINTS

You can also select a folder containing export controller status information.

BU_Controller type_Serial number_Date and time

4. Click the [OK] button to display the [Restore Controller] dialog.

Click the [Display] button to display the [Controller Status Viewer] dialog box to check saved file information.



- Robot names, serial numbers, calibrations checkbox

This checkbox allows you to restore the robot name, robot serial number, Hofs data, and CalPls data. If the wrong Hofs data is restored, the robot may move to wrong positions. Be careful when you set the alternative models. This is turned off by default.

A Safety Board password is necessary when using Controller with Safety Board.

- Robot maintenance configuration checkbox

  This checkbox allows you to restore the parts consumption data. For details, refer to the following manual.

  - For RC700-D, RC700-E, RC800-A: "Robot Controller Manual - Alarm"
  - For RC700, RC90 series: "Robot Controller Maintenance Manual - Alarm"
  - For T, VT series: "Robot Maintenance Manual - Alarm"

  This is turned off by default.

  Check this checkbox when restoring a backup data which is retrieved while [Setup]-[System Configuration]-[Controller]-[Preferences]-[Enable robot maintenance data] is enabled. If not checked, the maintenance data will not be reflected.

- Project checkbox

  This checkbox allows you to restore project files. Project files are restored to the PC and controller. On a PC, project files are restored to the project path used when building the project. However, if the project path is not included in the project save destination in Preferences, project files will be saved to the default project save location. This is turned off by default.

  When the project is restored, all the values of Global Preserve variables are restored. For details about Global Preserve variable backup, refer to 5.12.10 [Display Variables] Command (Run Menu).

- Vision hardware configuration checkbox

  This checkbox allows you to restore the vision hardware configuration. This is turned off by default. For details, refer to the following manual.

  "Epson RC+ 8.0 option: Vision Guide 8.0"

- Force sensor configuration checkbox

  This checkbox allows you to restore the force sensor configuration. This is turned off by default. For details, refer to the following manual.

  "Epson RC+ 8.0 option Force Guide 8.0"

- Password authentication settings checkbox

  This checkbox allows you to restore the connection password that was stored in the controller.

- Part feeders configuration

  This checkbox allows you to restore the setting of part feeding communication. This is turned off by default. For details, refer to the following manual.

  "Epson RC+ 8.0 Option Part Feeding 8.0 Introduction & Software"

- Safety Board Settings

  Allows you to restore the setting of the safety function. This is turned off by default. When the Controller is equipped with a Safety Board, a Safety Board password is necessary when checking. For details, refer to the following manual.

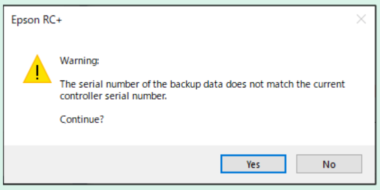
"Robot Controller Safety Function Manual - Restoring Saved (Backed-up) Settings"

- Security configuration checkbox

  This checkbox allows you to restore the security configuration. This is turned off by default. See details below.

  Security

5. Click the [OK] button to restore the system information.

> **KEY POINTS**
>
> - Restore the system configuration saved using Backup Controller only for the same system.
>
>   When different system information is restored, the following warning message appears.
>
>   
>
>
>   Click the [No] button to cancel restoration of data except for special situations such as controller replacement.
>
> - When restoring the backup which includes the data of the robot configured to the Drive Unit, be sure to restore the data while the Drive Unit is connected and turned on.
> - When restoring the backup including the unsupported robot information to the target controller, an error occurs.
> - If the file "IOLabels.dat" which saves I/O labels exceed 400 kB, a parser communication error may occur. If an error occurs, adjust the number of characters for I/O labels not to exceed 400 kB.
> - You cannot restore the backup including PG to the virtual controller.
> - You cannot restore the data that is backed up by the virtual controller to T series and VT series manipulators.
> - You can select [Safety board configuration] when all of the following conditions are met.
>   - Controller with Safety Board,
>   - Backup Controller has the safety board information
> - For the Controller with the safety board installed, selecting one or more of the items below will start Safety Function Manager after restarting the Controller.
>   - Robot names, serial numbers, calibrations checkbox
>   - Safety Board Settings
>
>   For details, refer to the following manual.
>
>   "Robot Controller Safety Function Manual"

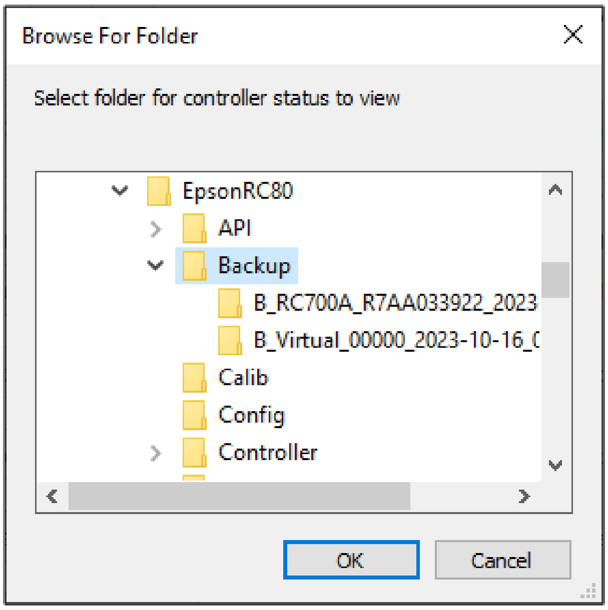
View Controller Status

Click the [View Controller Status] button to view the status data stored from a previous status export. See details in the previous section “Backup Controller”.

To view controller status:

1. Select Controller from the [Tools] menu to open the [Controller Tools] dialog box.

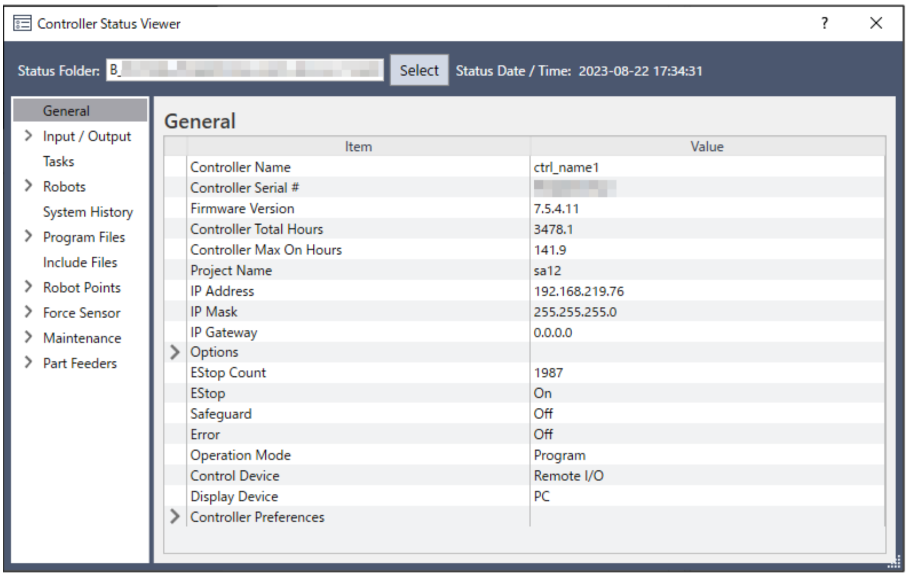
2. Click on the [View Controller Status] button to open the [Browse For Folder] dialog.



3. Select the drive and folder where the information is stored.
   - RC+: B_Controller type_Serial number_Date and time
   - Controller: BU_Controller type_Serial number_Date and time

4. Click the [OK] button to view the selected controller status.

5. The [Controller Status Viewer] dialog will be displayed.



6. Select items to view from the tree on the left side of the dialog.

7. To view another controller status, click the [Select] button next to the [Status Folder] name, and then select a new status folder.

Reset Controller

Use the [Reset Controller] button to reset the SPEL controller.

Maintenance

It shows the parts consumption data for the Controller or Manipulator parts. For details, refer to the following manual.

- For RC700-D, RC700-E, RC800-A: "Robot Controller Manual - Alarm"
- For RC700, RC90 series: "Robot Controller Maintenance Manual - Alarm"
- For T, VT series: "Robot Maintenance Manual - Alarm"

6.12.9 [Simulator] (Tools Menu)

: Ctrl+F5

Opens the [Simulator] window. See details below.

Simulator

6.12.10 [GUI Builder] (Tools Menu)

: Ctrl+F7

Opens the [GUI Builder] window. For details, refer to the following manual.

"Epson RC+ 8.0 Option GUI Builder 8.0 GUI Builder Environment"

6.12.11 [Conveyor Tracking] (Tools Menu)

: Ctrl+F8

Opens the [Conveyor Tracking] window. See details below.

Conveyor Tracking

6.12.12 [Part Feeding] (Tools menu)

: Ctrl+F12

Opens the [Part Feeding] window. For details, refer to the following manual.

"Epson RC+ 8.0 Option Part Feeding 8.0 Introduction & Software - Software - Part Wizard"

6.12.13 [Vision] (Tools Menu)

: Ctrl+F9

For details, refer to the following manual.

"Vision Guide 8.0 Hardware Manual - Setup - Software Configuration"

6.12.14 [Force Guide] (Tools Menu)

: Ctrl+F11

Opens the [Force Guide] window. For details, refer to the following manual.

"Epson RC+ 8.0 Option Force Guide 8.0 - Software - [Force Guide] (Tools Menu)"

6.12.15 [Force Monitor] (Tools Menu)

: Ctrl+F10

Opens the [Force Monitor] window. For details, refer to the following manual.

"Epson RC+ 8.0 Option Force Guide 8.0 - Software - [Force Monitor] (Tools Menu)"

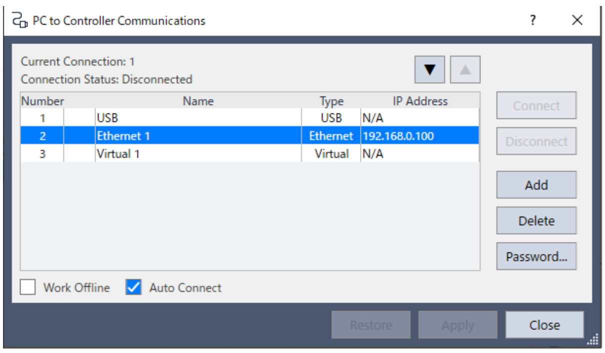
6.13 [Setup] Menu

The [Setup] menu contains the following commands:

- [PC to Controller Communications] Command (Setup Menu)
- [System Configuration] Command (Setup Menu)
- [Preferences] Command (Setup Menu)
- [License Configuration] Command (Setup Menu)

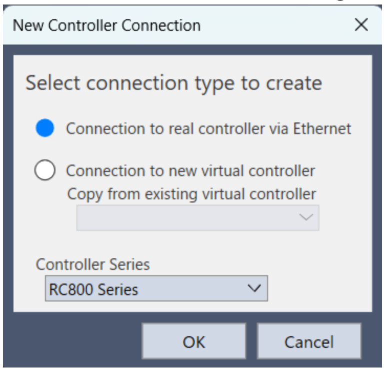
6.13.1 [PC to Controller Communications] Command (Setup Menu)

To configure communications with the Controller, select [PC to Controller Communications] from the [Setup] menu. The [PC to Controller Communications] dialog will be displayed as shown below:



Item	Description
▼▲	Sorts the connection list. Connection #1 “USB” cannot be changed.
No.	Sets the connection number. You can set up to 99 connections. Connection #1 “USB” cannot be changed.
Name	Renames the connection. Connection #1 “USB” cannot be changed.

<table>
<tr><th>Item</th><th>Description</th></tr>
<tr><td>Controller series</td><td>Configures the connecting Controller series. If the connection type is Ethernet, you can change the Controller.
RC800: Connects to RC800 series Controllers
RC700: Connects to RC700, RC90, T, and VT series Controllers</td></tr>
<tr><td>Connection type</td><td>Shows the connection type.
USB: Communicates with the Controller by USB cable
Ethernet: Communicates with the Controller by Ethernet
Virtual: Communicates with a virtual controller</td></tr>
<tr><td>IP Address</td><td>Sets the IP address of the "Ethernet" connection.</td></tr>
<tr><td>Connect</td><td>Connect the selected communication.</td></tr>
<tr><td>Disconnect</td><td>Disconnect the communication.</td></tr>
<tr><td>Add</td><td>Add communication information of Ethernet or a virtual controller. "No." is automatically assigned to the lowest unused number. Clicking this button opens the dialog to specify the communication type.



</td></tr>
<tr><td>Delete</td><td>Deletes selected communication information. Connection #1 “USB” cannot be deleted. The deleted "No." becomes a missing number, and the "No." of other connections will not change.</td></tr>
<tr><td>Password</td><td>Sets authentication passwords for the Ethernet connecting Controller and PC connection destination.</td></tr>
<tr><td>Work Offline</td><td>You can build a project without connecting to the controller in Offline mode. Some functions such as Robot Manager are not available in this mode.</td></tr>
<tr><td>Auto Connect</td><td>Performing an operation that requires a Controller connection (execute build, start Robot Manager) while a Controller is not connected, will automatically connect to the last Controller connection.</td></tr>
<tr><td>Restore</td><td>Revert back to previous values.</td></tr>
<tr><td>Close</td><td>Closes the dialog box.</td></tr>
<tr><td>Apply</td><td>Saves changes.</td></tr>
</table>

KEY POINTS

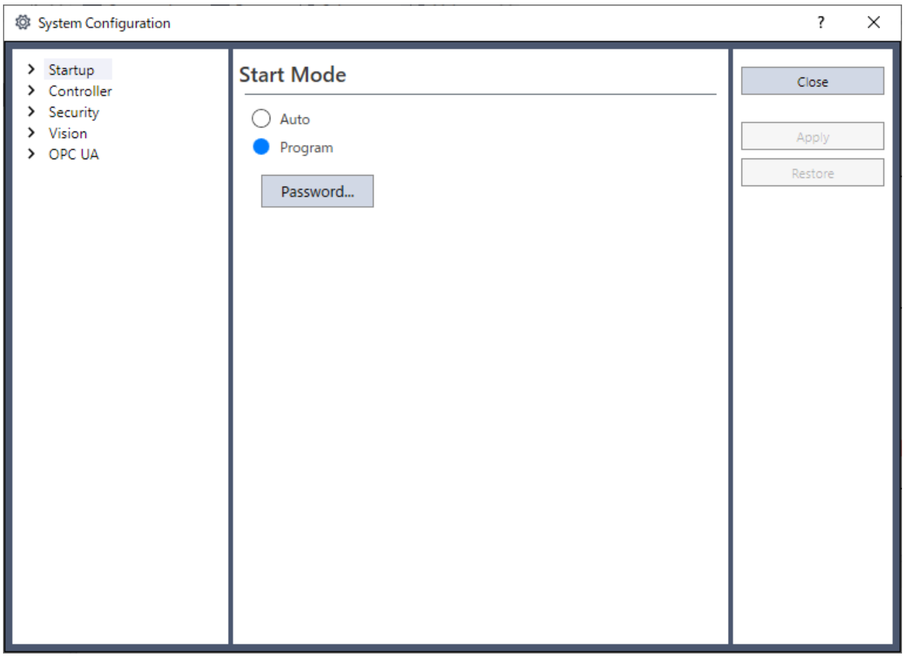
Changing the connection number (by using the ▼▲, deleting connections, or other operations) may affect the RC+ API application. Be careful when you set the alternative models.

KEY POINTS

Program total execution time: In the virtual controller, the total execution time of programs is limited to one hour. If total execution is over one hour, a warning message appears. You can execute the program again after the warning is displayed. The total execution timer will be reset.

6.13.2 [System Configuration] Command (Setup Menu)

The [System Configuration] command opens a dialog that contains several pages that are used to configure the system for the Epson RC+ 8.0 environment. To open the [System Configuration] dialog box, select [Setup]-[Preferences].



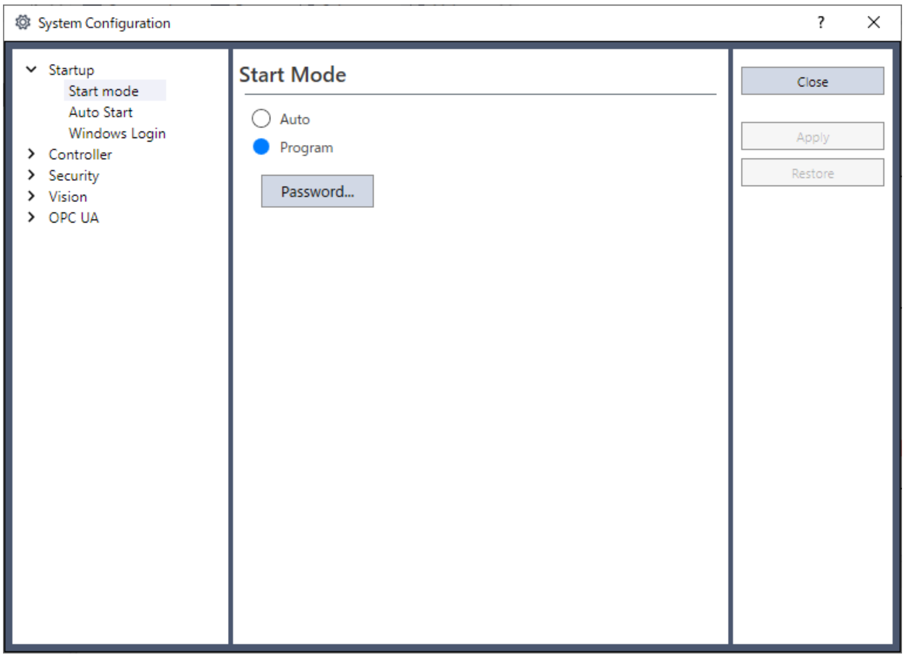
Common buttons

Item	Description
Close	Close the [System Configuration] dialog. Pressing the [Close] button after changing the system configuration will reset the Controller to a startup state.
Apply	Saves changes. The Controller may reset to enable the new settings.
Restore	Revert back to previous values.

6.13.2.1 [Setup]-[System Configuration]-[Startup]

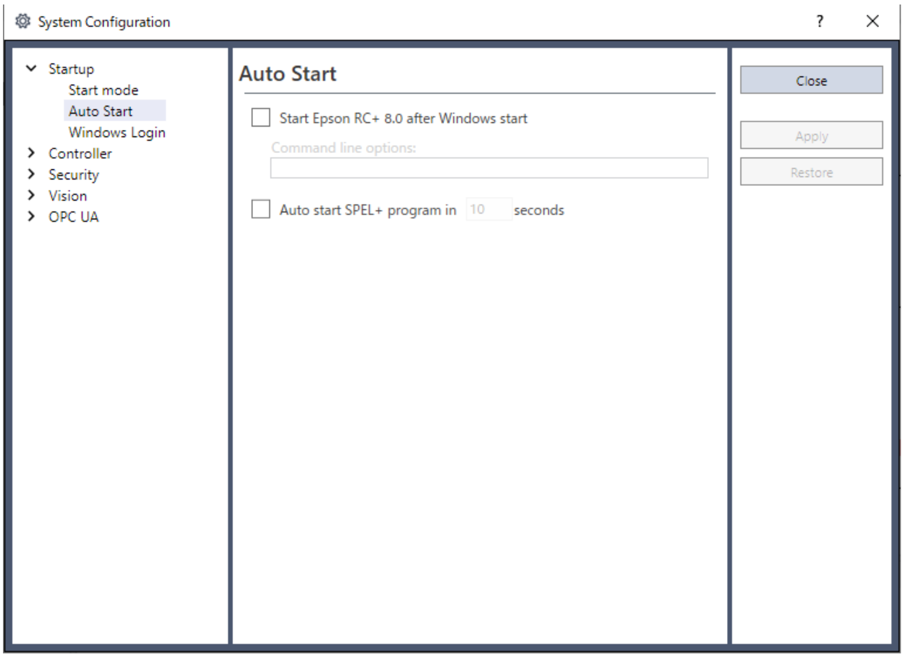
6.13.2.1.1 [Setup]-[System Configuration]-[Startup]-[Start Mode] Page

From the Start Mode page, you can choose whether Epson RC+ 8.0 starts in Auto mode or Program mode.



Item	Description
Auto	Select Auto to start Epson RC+ 8.0 in Auto mode. See details below. **Operation**
Program	Select Program to start Epson RC+ 8.0 in Program mode. See details below. **Operation**
Password	Click this button to change the password required to enter Program mode from Operator mode when Epson RC+ 8.0 starts.

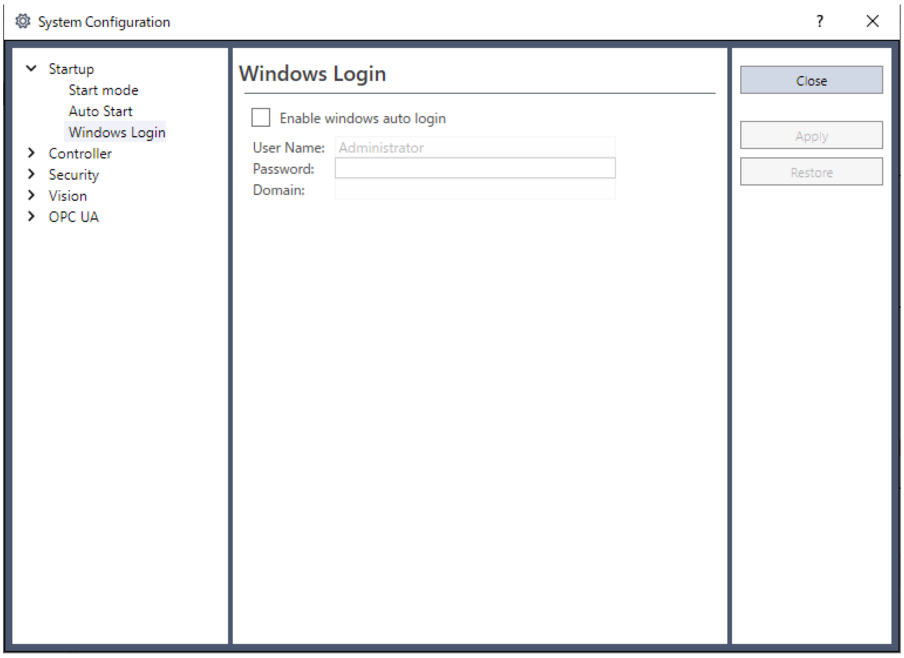
6.13.2.1.2 [Setup]-[System Configuration]-[Startup]-[Auto Start] Page



Item	Description
Auto Start Epson RC+ after Windows start	Check this box if you want Epson RC+ 8.0 to automatically start after Windows starts.
Command line options	Enter the command line options used when Epson RC+ 8.0 is automatically started. This is enable only when the [Start Epson RC+ 8.0 after Windows start] checkbox is checked. Up to 1024 characters can be entered.
Auto start SPEL+ program	In Auto Mode, check this box if you want to execute the main program after a delay. This is active only when starting in Operator mode and the control device is "PC".

6.13.2.1.3 [Setup]-[System Configuration]-[Startup]-[Windows Login] Page

The Windows Login page allows you to configure the automatic login when Windows starts.

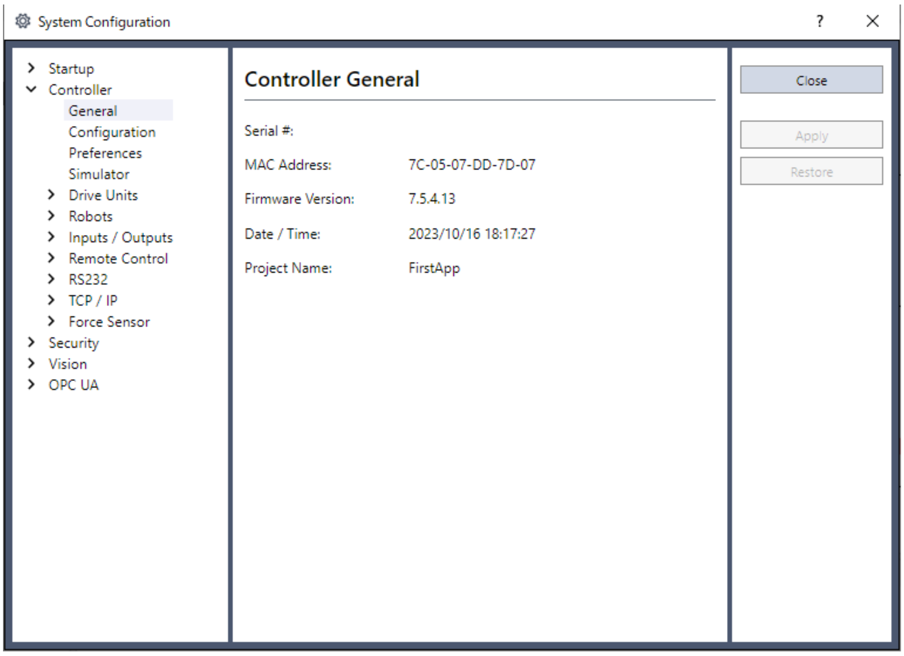


Item	Description
Enable windows auto login	Check this box if you want to automatically login to Windows when it starts. You must supply a valid user name, password, and domain.
User Name	Enter the name of a valid Windows user on the system.
Password	Enter the login password for the user.
Domain	If the PC is the member of a domain, enter the name here. Optional.

6.13.2.2 [Setup]-[System Configuration]-[Controller]

6.13.2.2.1 [Setup]-[System Configuration]-[Controller]-[General] Page

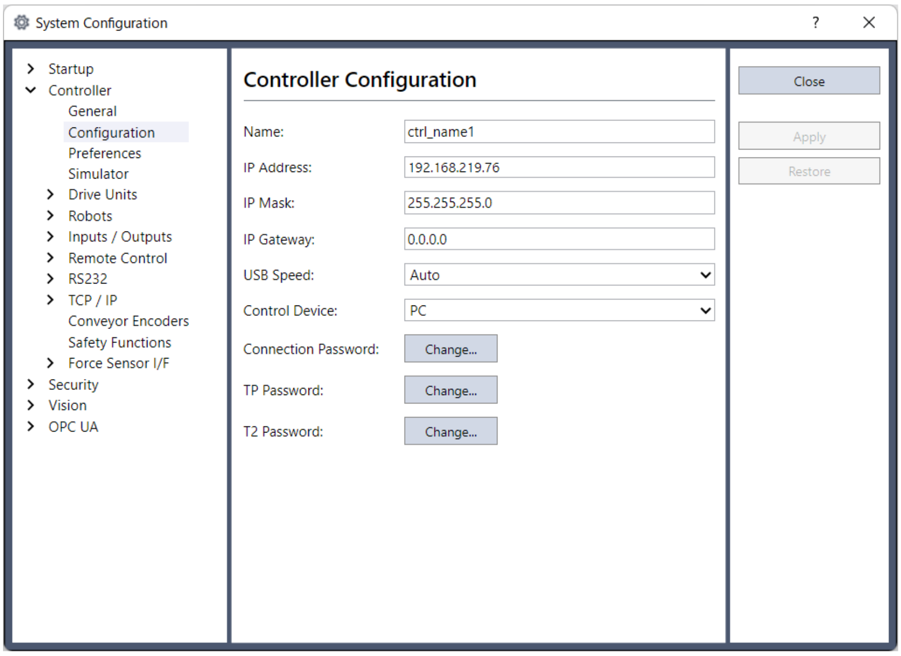
This page allows the user to view general information about the controller.



Item	Description
Serial #	Displays the serial number of the current controller.
MAC Address	Displays the MAC Address of the controller.
Firmware Version	Displays the firmware version used in the current controller.
Date / Time	Displays the current date and time in the controller.
Project Name	Displays the name of the project in the controller.

6.13.2.2.2 [Setup]-[System Configuration]-[Controller]-[Configuration] Page

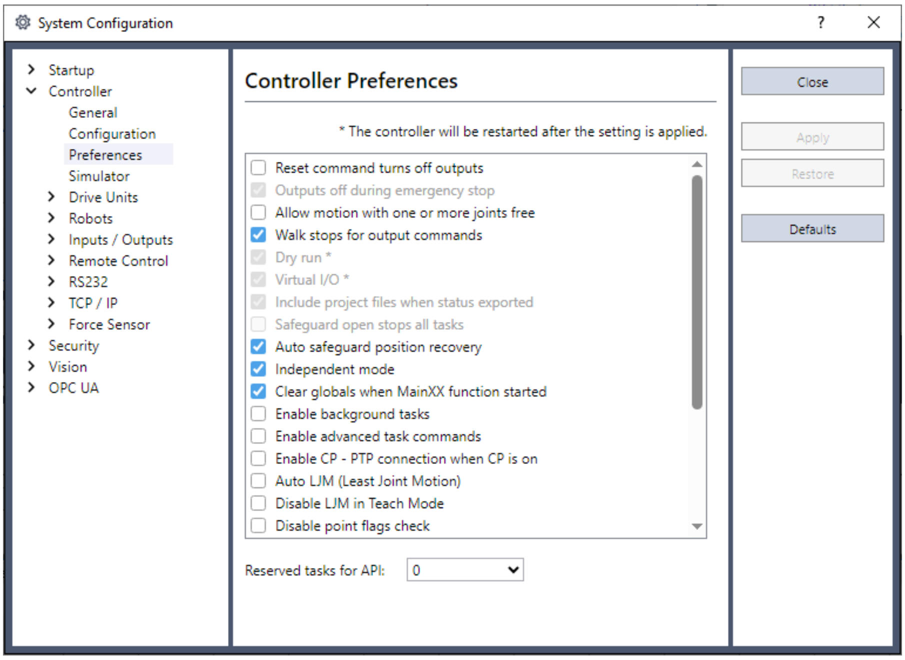
This page allows the user to view and change the controller configuration settings.



Item	Description
Name	Use this text box to change the controller name. You may use any name up to 16 characters long using alphanumeric characters and underscore.
IP Address	Use this text box to set current IP address of the LAN-1 port. The IP Address must be on the same subnet as the PC. (Default IP address is 192.168.0.1.)
IP Mask	Use this text box to set the IP mask of the LAN-1 port. Note that the IP Mask must match the IP mask used for your network.
IP Gateway	Use this text box to set the IP gateway of the LAN-1 port. This is only required if you will be accessing the controller from outside of the local network.
Control Device	Allows you to select the Control Device.
Connection Password	Allows you to change authentication passwords for the Ethernet connecting Controller and PC connection destination.
TP Password	Allows you to change the TP password.
T2 Password	Allows you to change the TP2 password.

6.13.2.2.3 [Setup]-[System Configuration]-[Controller]-[Preferences] Page

This page contains controller preference settings.



Item	Description
Default	Reverts back to the default settings.

* The controller will be restarted after the setting is applied.

When changing an item marked with "*", click the [Apply] button to immediately reboot the Controller.

When changing an item not marked with “*”, close the [System Configuration] dialog to reboot the Controller (Excluding the “Auto safeguard position recovery” setting that is not included in the Controller Status).

RESET command turns off outputs

When this preference is turned on, all outputs other than remote control outputs will be turned off when a Reset instruction is executed. This is turned off by default.

> **KEY POINTS**
>
> - The outputs of the standard I/O, expansion I/O, and Fieldbus I/O are included in the "outputs" mentioned in the above preferences [RESET command turns off outputs] and [Outputs off during Emergency Stop]. Memory I/O is not affected by these preferences. Therefore, memory I/O bits are not turned off by the RESET command execution or during Emergency Stop.
> - Regardless of whether it is checked or not, the output set for the hand will not be turned off even if the Reset command is executed. This is to prevent the workpiece from being unintentionally released by executing the Reset command. For the hand function, refer to the following manual:
>
>   "Hand Function Manual"

Outputs off during Emergency Stop

When this preference is turned on, all outputs other than remote control outputs will be turned off when emergency stop occurs. Also, no outputs can be turned on until the emergency stop condition is cleared. This is turned on by default.

Uncheck this preference to execute I/O On/Off using the NoEmgAbort task or background task after Emergency Stop. If it remains checked, the execution order of turn off by this preference and turn on using the task are not guaranteed.

> **KEY POINTS**
>
> - You should design your system to always remove all power to output devices when emergency stop occurs. Even if the controller turns off outputs, the I/O hardware could malfunction.
> - Regardless of whether it is checked or not, the output set for the hand will not be turned off even if emergency stop is performed. This is to prevent the workpiece from being unintentionally released by performing emergency stop. For the hand function, refer to the following manual:
>
>   "Hand Function Manual"

Allow motion with one or more joints free

When this preference is turned on, motion commands can be executed after SFree has been used to free one or more joints. This is turned off by default.

Walk stops for output commands

When checked, the Walk command from the Run Menu will execute lines until after the next motion or output statement (whichever comes first). When unchecked, the Walk command will execute lines until after the next motion statement and will not stop for output statements. This is turned on by default.

Dry run

This preference allows you to run programs without a robot connected to the controller. All program statements will work. Motion statements will execute approximately the same amount of time as when connected to a robot. This is turned off by default.

KEY POINTS

When using safety function (the Controller with Safety Board), the setting cannot be changed at this window. Use the safety function manager and change it. For details, refer to the following manual.

"Robot Controller Safety Function Manual - Making Dry Run Settings"

Virtual I/O

This preference allows you to run programs using virtual I/O. When Virtual I/O is enabled, I/O commands do not affect the hardware I/O. There are also several commands available for turning on inputs from within a program. This is turned off by default.

KEY POINTS

Remote function is also available when virtual I/O is enabled.

Include project files when status exported

This preference allows you to configure whether project files are included or not when the controller status is exported. For information on saving controller status, see below. This is turned on by default.

[Controller] Command (Tools Menu)

Safeguard open stops all tasks

Check this option to cause all normal tasks and NoPause task to stop when the safeguard is open. Only NoEmgAbort task and background tasks will continue.

This option can be used in applications where pause / continue are not required.

This is turned off by default.

Auto safeguard position recovery

This preference allows you to move a robot back to the position where it was at the safeguard opened when continuing the program execution. This is turned on by default.

- Auto recover ON

  Automatically turns ON a motor and moves a robot in low power status to the position where it was when the safeguard opened. Continues the usual cycle. (Default)

- Auto recover OFF

  In the Run Window and Operator Window, when an operator clicks the [Resume] button, a dialog with a [Recover] button will be shown.

  The operator needs to hold down the [Recover] button until the motor is ON and the robot's return is finished. Otherwise the robot will stop before reaching final position. After verifying that the robot's return is finished, the operator clicks the [Continue] button to continue the usual cycle.

KEY POINTS

The peak current is generated in each manipulator when the motor is turned ON. If multiple manipulators are connected using a Drive Unit or a PG Unit, the motor on timing at recovery is changed intentionally for each manipulator to avoid generating the peak current simultaneously. In this case, each manipulator requires approx. 1.5 seconds to turn on the motor.

Independent mode

This preference allows you to use the controller without interfacing with the Windows (Independent mode).

Use this option when you want to use the controller through the external device using Remote I/O. This is turned on by default.

Initialize global variables when function starts

This preference allows you to initialize the global variables as the function becomes active.

Turn off this preference when you sue the global variables from the background task. Otherwise, the variables will be initialized by the controller and the variable-access conflict from tasks will occur. This is turned on by default.

Enable background tasks

This preference allows you to execute background tasks. This is turned off by default.

Enable advanced task commands

This preference allows you to execute StartMain, Cont, Recover, Reset Error commands. This is turned off by default.

CAUTION

Before you execute StartMain, Cont, Recover, Reset Error commands, you should understand each command's specification and verify that the system has the appropriate condition to execute these commands. Improper use, such as executing commands continuously in a loop, can reduce the security of system. Be careful.

Enable CP – PTP connection when CP is ON

This preference allows you to overlap the trajectories of CP motion and PTP motion during CP ON. This is turned off by default.

KEY POINTS

Over-speed error or Over-acceleration-speed error may occur according to the motion acceleration / deceleration speed setting. If the error occurs, adjust the acceleration / deceleration speed setting or uncheck this checkbox.

Auto LJM (Least Joint Motion)

This preference allows you to enable Auto LJM at the controller start up. This is turned off by default. To disable Auto LJM temporarily, use AutoLJM Off command.

KEY POINTS

If Auto LJM is enabled at all times, this function automatically adjusts the posture of the robot to reduce the motion distance, even when you intended to move the joint widely. Therefore, it is recommended to disable Auto LJM at the controller start up and operate the robot as you desired using AutoLJM On command or LJM function.

Disable LJM in Teach Mode

This preference allows you to invalidate LJM in the TEACH mode. The LJM function becomes invalid regardless of the command of AutoLJM. This is turned off by default.

Disable Point flag check

This preference allows you to continue operation even when point flags, one was specified as a target point and the other one after the motion completion, do not match in a CP motion. However, if the flags do not match at the transferring point while CP On is used, the robot will stop at the point and the motion will not become a path motion. This is turned off by default.

Motor off when Enable switch off in Teach Mode

This preference is read-only. It shows whether motors will be turned off when the Enable switch is off during Teach Mode. This is turned on by default.

Enable robot maintenance data

This preference allows you to enable the parts consumption management for the Controller and robot parts. This is turned on by default.

Motor power low when ForcePowerLow signal OFF

This preference allows you to specify whether or not to invert the logic of the ForcePowerLow signal input value.

When this checkbox is selected, the ForcePowerLow signal will work as the forced low power function which operates the robot in the low power mode when remote I/O input signal is Low.

When this checkbox is unchecked, the ForcePowerLow signal will work as the forced low power function which operates the robot in the low power mode when the remote I/O input signal is High.

This is turned off by default.

For details of the ForcePowerLow signal, see below.

Remote Input

ForcePowerLow signal change pauses all tasks

This preference allows you to specify whether to stop or temporarily stop the tasks when the input of the ForcePowerLow (forced low power) signal is changed.

When this checkbox is selected, all tasks and commands will be temporarily stopped when the remote I/O input signal is changed. The program execution can be continued.

When this checkbox is unchecked, all tasks and commands will be stopped when the remote I/O input signal is changed. The program needs to be restarted.

This is turned off by default.

For details of the ForcePowerLow signal, see below.

Remote Input

Disable Test (T2)

This preference is read-only. It shows whether the execution of Test (T2) of TP3 and TP4 is prohibited. This is turned off by default.

Disable connection password

This preference allows you to disable the PC (Ethernet) authentication for PC (Ethernet) connections. This is turned off by default.

Disable connection from Epson RC+ Express Edition

When this checkbox is checked, the connection from Epson RC+ Express Edition is restricted. Security features do not apply to Epson RC + Express Edition. Check this checkbox if you want to limit unintended connections. This is turned off by default.

Applying XYLim to motion trajectory and pulse motion

When this checkbox is checked, XYLim is applied to not only the motion commands of target coordinate, but also motion trajectory from starting point of motion to target coordinate. Moreover, XYim is applied to pulse motion. This is turned on by default.

⚠ CAUTION

When this checkbox is not checked, robot may pass outside of XYlim area. Be careful.

Include error controller data when saving the state to a USB flash drive.

Check this checkbox to include Controller data at time of error in the backup data from the RC800 series Controller to the USB flash drive. This is turned off by default.

Limit motion speed by orientation rotation speed

Check this checkbox to enable SpeedRLimitation at Controller startup. When SpeedRLimitation is enabled, the tool orientation change speed during CP operation is limited so that the motion speed does not exceed set SpeedR. To temporarily disable, use the SpeedRLimitation Off command.

This is turned off by default.

KEY POINTS

The SpeedR default value is set to low, so if SpeedR is not set properly, CP movements will be slow with orientation changes. If SpeedRLimitation is always enabled, set SpeedR (upper limit of tool orientation change speed) appropriately along with SpeedS.

Reserved tasks for API

This setting is used to execute more than one Spel class methods of the RC+ API. You can set up to 16 tasks. The default is 0.

KEY POINTS

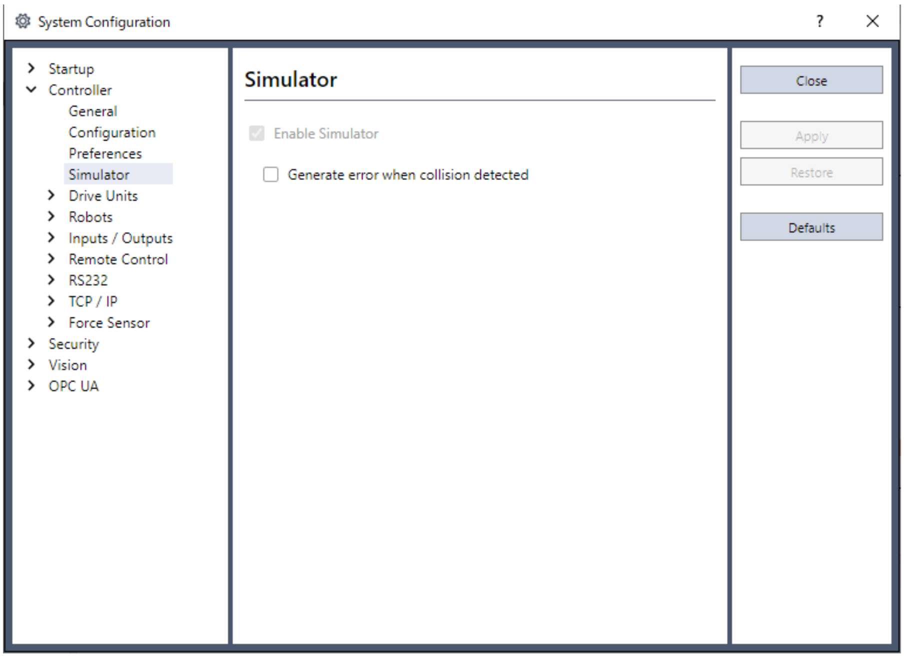
The RC+ API tasks use some of the normal tasks. Therefore, if this setting is used, the number of normal tasks available for the Spel+ programs will be as follows:

(Normal tasks) = 32 - (RC+ API tasks)

6.13.2.2.4 [Setup]-[System Configuration]-[Controller]-[Simulator]

This page contains simulator preference settings. For details on simulators, see below.

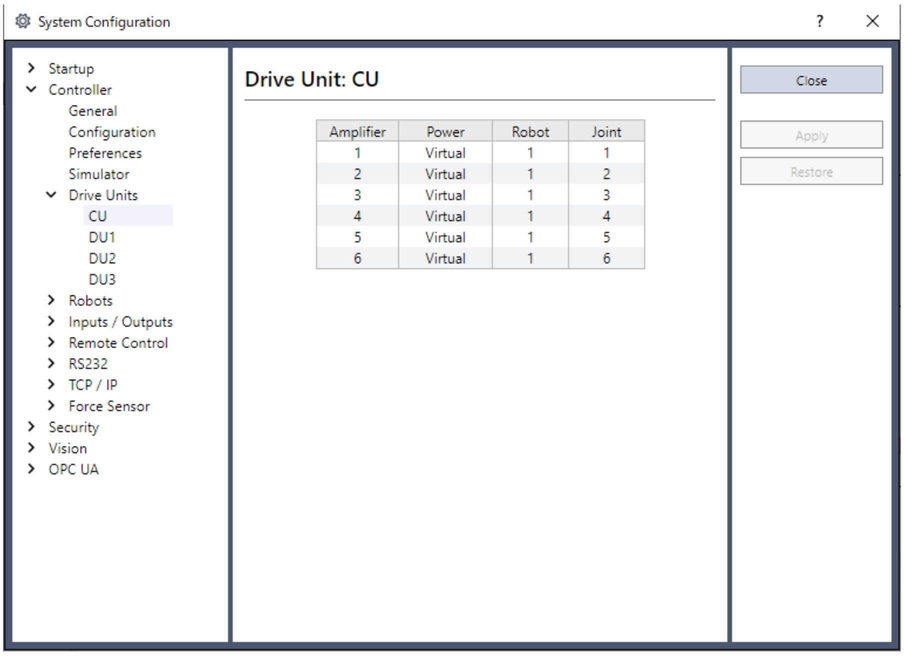
Simulator



Item	Description
Enable Simulator	This enables the simulator function. This setting cannot be changed if a virtual controller is connected.
Generate error when collision is detected	If a collision is detected while the SPEL+ program is running, an error will occur with the Controller and program execution will stop.
Default	Reverts back to the default settings.

6.13.2.2.5 [Setup]-[System Configuration]-[Controller]-[Drive Units]

This page displays the status of the Drive Unit. It shows Output, Robot, and Axis settings of each Drive Unit.

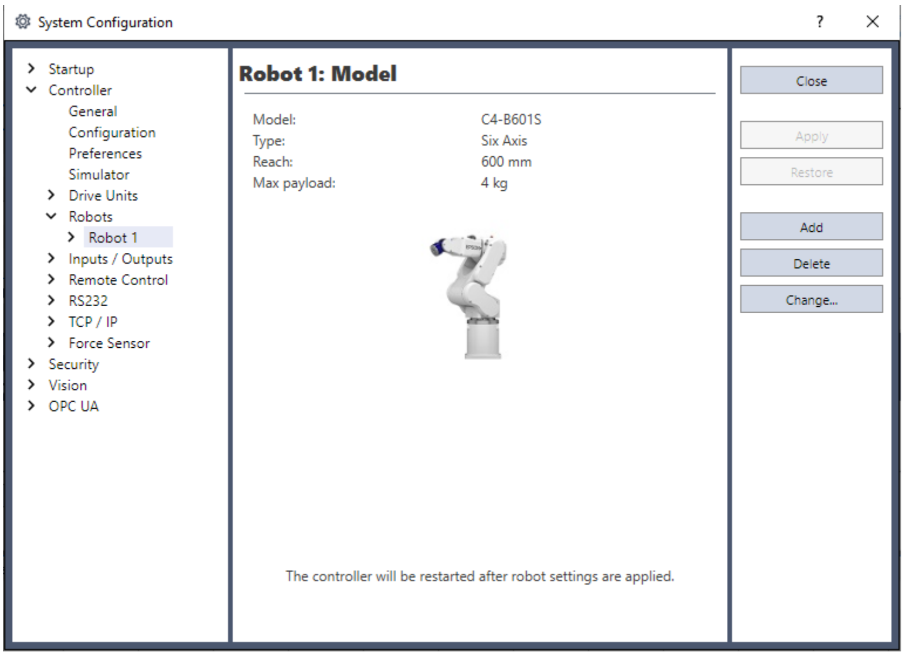


6.13.2.2.6 [Setup]-[System Configuration]-[Controller]-[Robots]

6.13.2.2.6.1 [Setup]-[System Configuration]-[Controller]-[Robots]-[Robot**]-[Model] page

This displays robot information. For details on adding, deleting, or changing robots, see below.

Robot Configuration



Item	Description
Model	Displays the robot model.
Type	Displays the robot type.
Reach	Displays the robot arm length.
Max payload	Displays maximum payload of the robot.

Item	Description
Add	Adds a robot.
Remove	Deletes a robot.
Change	Changes the robot.

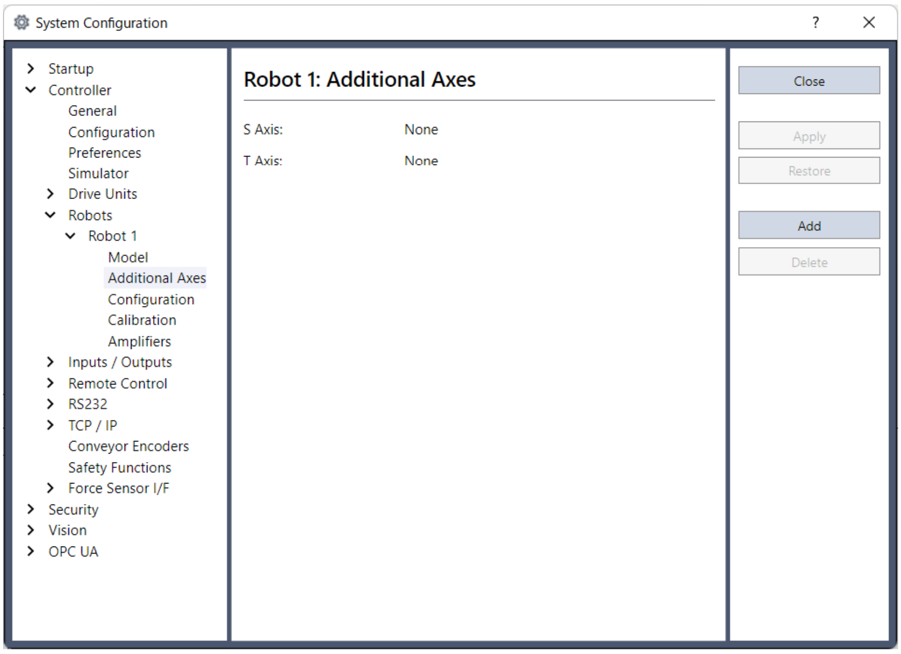
KEY POINTS

- When using safety function (the Controller with Safety Board), do not change the robot model of the time of shipment. It cannot be complied with Safety Function.
- For the Controller with Safety Board, assign a robot using the safety function to Robot 1.

6.13.2.2.6.2 [Setup]-[System Configuration]-[Controller]-[Robots]-[Robot**]-[Additional Axes]

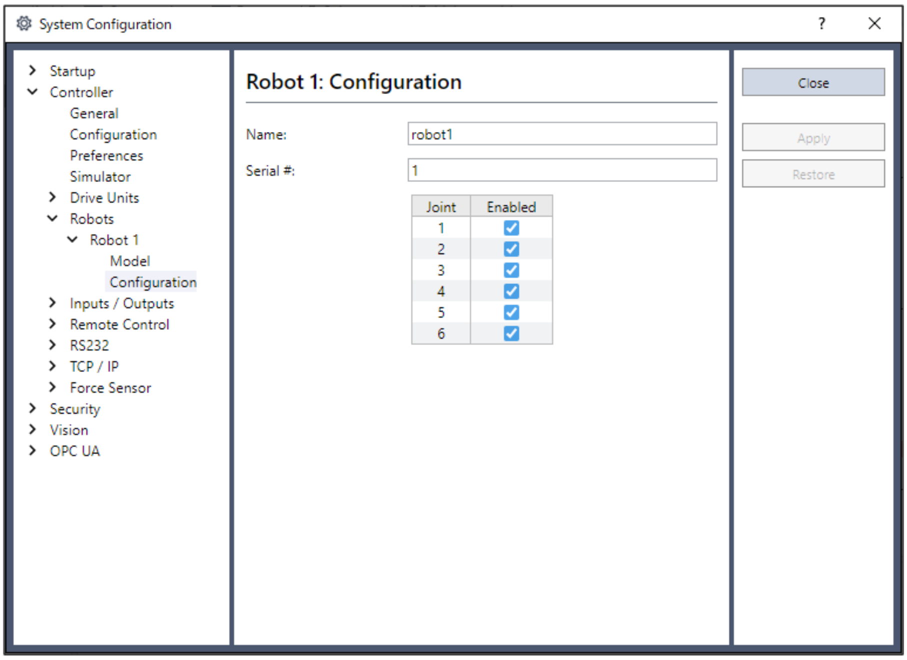
For details of the additional axes, see below.

Configuration of Additional Axes



Item	Description
S Axis	Displays the configuration of additional S axis.
T Axis	Displays the configuration of additional T axis.
Add	Adds an additional axis.
Remove	Deletes an additional axis.

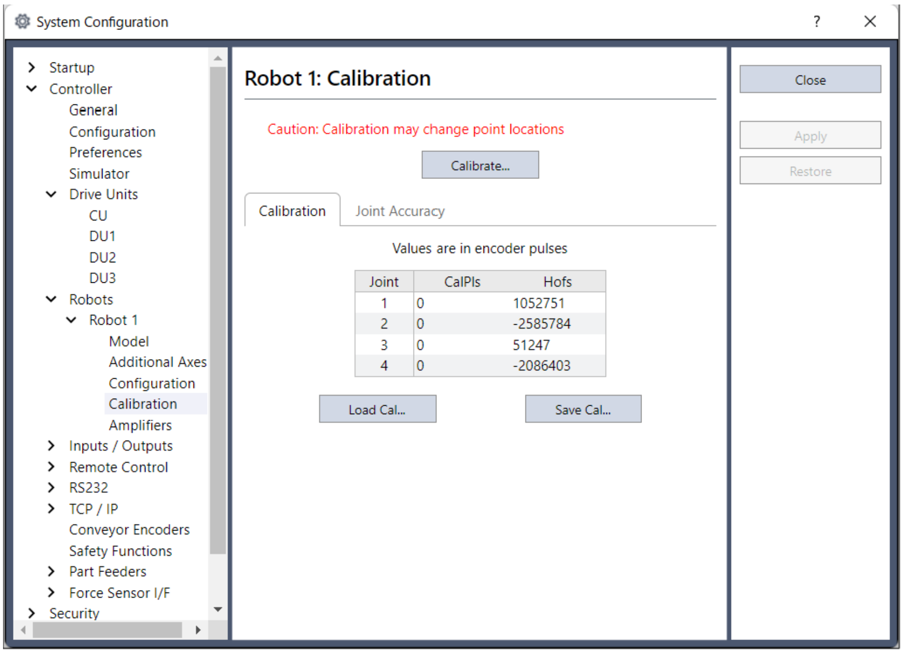
6.13.2.2.6.3 [Setup]-[System Configuration]-[Controller]-[Robots]-[Robot**]-[Configuration] page



Item	Description
Name	Enter a Name for the robot.
Serial #	Enter the Serial number of the robot.
Joint, Enabled	These checkboxes determine if the respective joint is enabled or disabled.

6.13.2.2.6.4 [Setup]-[System Configuration]-[Controller]-[Robots]-[Robot**]-[Calibration] Page

You can calibrate each joint of the robot from this page.



Item	Description
Calibrate	Display the [Calibration Wizard] dialog for calibration.
CalPls	These are the Calpls settings for each joint. Normally, the calibration wizard will calculate these values.
Hofs	These are the Hofs settings for each joint. Normally, the calibration wizard will calculate these values.
Load Cal	Use this button to load data from a previously save calibration file. After the data is loaded, the grid will be refreshed to show the values.
Save Cal	Use this button to save the calibration data to a calibration file.

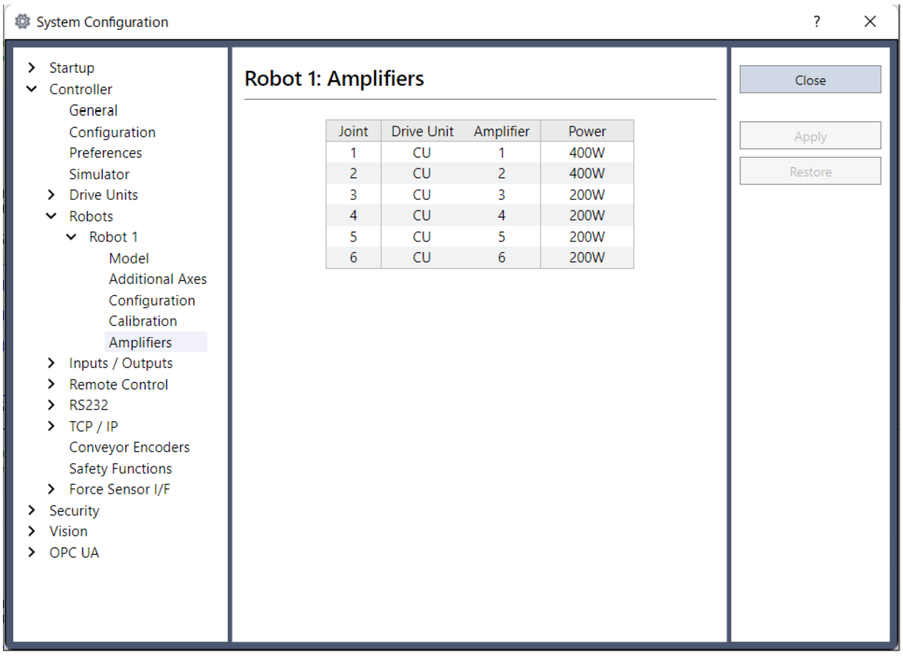
KEY POINTS

When using safety function (the Controller with Safety Board), changing Hofs requires start the Safety Function Manager to update the Hofs setting on the safety board. For details, refer to the following manual.

"Robot Controller Safety Function Manual - Checking Settings When Starting up Safety Function Manager"

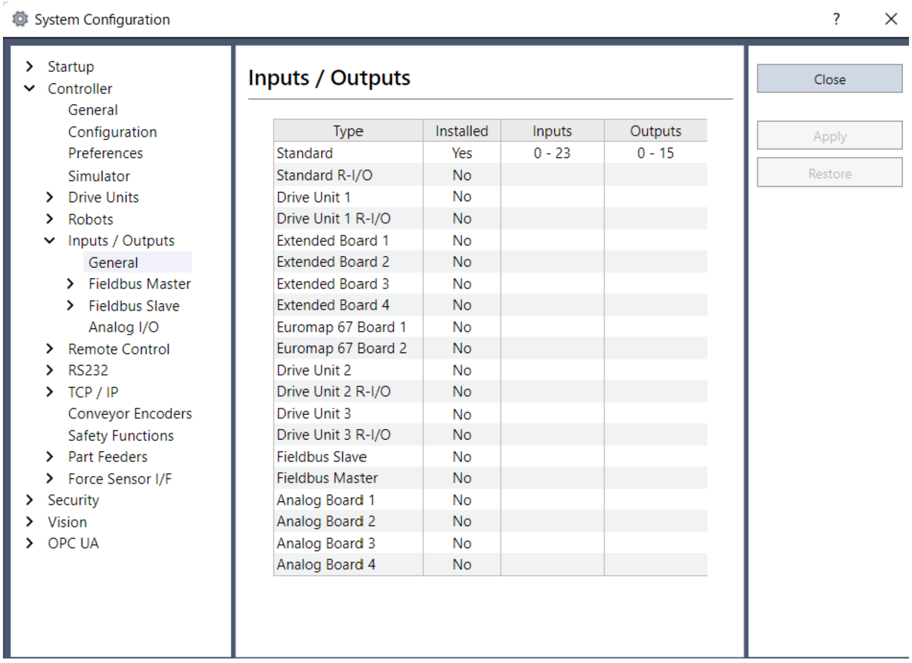
6.13.2.2.6.5 [Setup]-[System Configuration]-[Controller]-[Robots]-[Robot**]-[Amplifiers] Page

This shows the power for each robot amplifier currently in the controller.



6.13.2.2.7 [Setup]-[System Configuration]-[Controller]-[Inputs / Outputs] Page

This page shows the I/O hardware installed in the controller. There are no settings to configure.



6.13.2.2.7.1 [Setup]-[System Configuration]-[Controller]-[Inputs / Outputs]-[Fieldbus Master]

For details of Fieldbus master, refer to the following manual:

"Robot Controller Option: Fieldbus I/O"

6.13.2.2.7.2 [Setup]-[System Configuration]-[Controller]-[Inputs / Outputs]-[Fieldbus Slave]

For details of Fieldbus slave, refer to the following manual:

"Robot Controller Option: Fieldbus I/O"

6.13.2.2.7.3 [Setup]-[System Configuration]-[Controller]-[Inputs / Outputs]-[Analog I/O] Page

To configure, add, or confirm the analog I/O boards, refer to the following manual.

"Robot Controller Manual - Analog I/O Board"

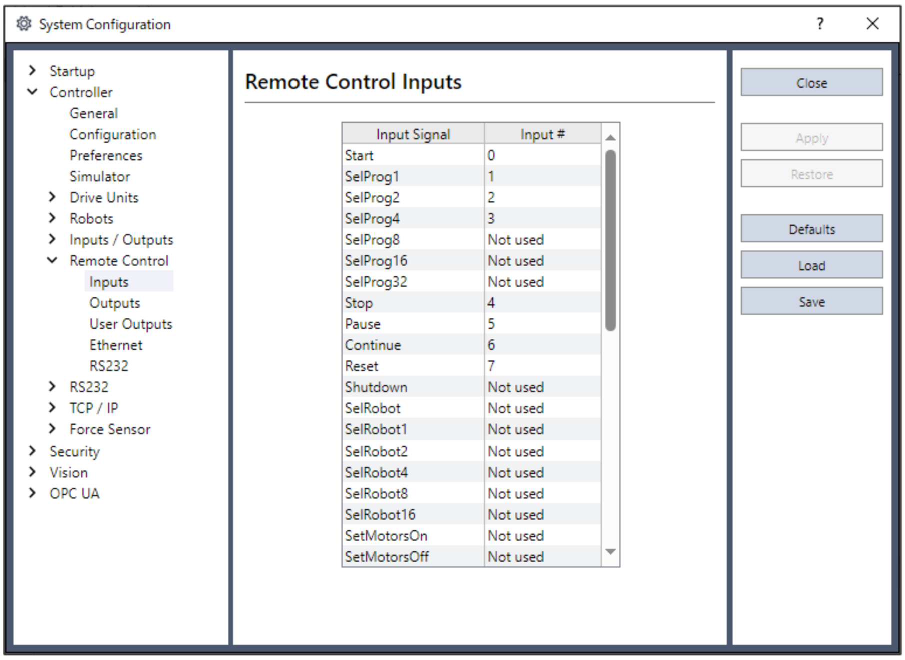
6.13.2.2.8 [Setup]-[System Configuration]-[Controller]-[Remote Control]

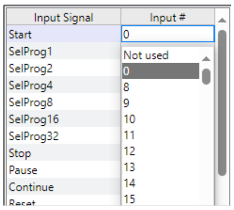
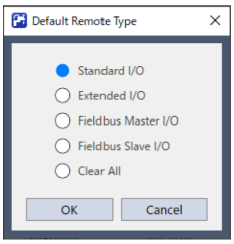
For details of Remote function, see below.

Remote Control

6.13.2.2.8.1 [Setup]-[System Configuration]-[Controller]-[Remote Control]-[Inputs] Page

Use this page to configure the controller remote control inputs.



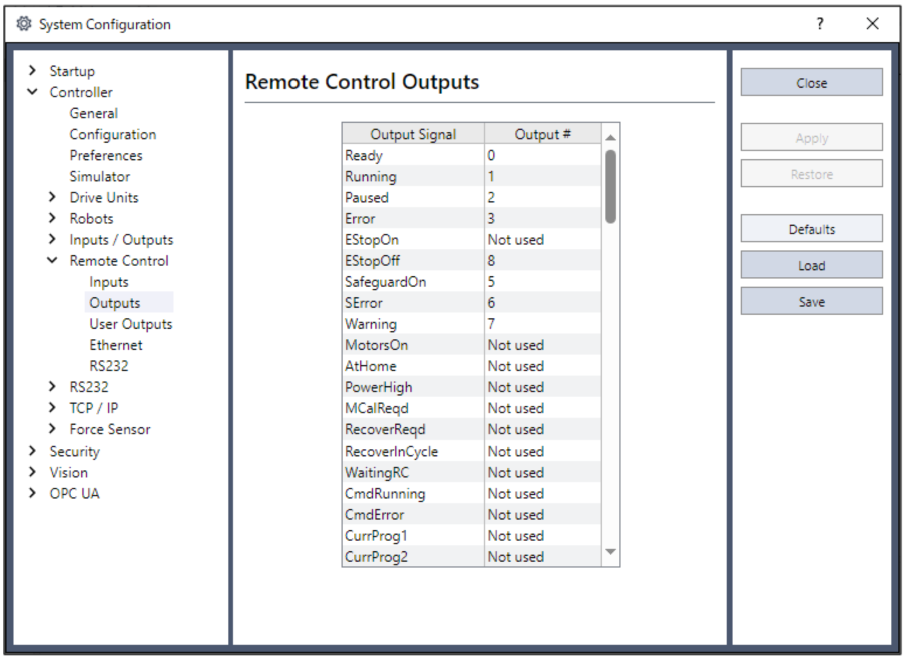
Item	Description
Input Signal	Displays the input signal. See details below. **Remote Input**
Input #	 Select an input bit to use for the corresponding input signal." Select Not used" to disable the remote output. For example, "Start" is assigned to I/O input bit 0 in the above dialog. Select "Not used" to use this as a normal I/O input.
Default	 Click this button to set the default remote inputs. First, a dialog box will be displayed asking you which type of inputs to use for defaults: Standard, Extended I/O, Fieldbus master, or Fieldbus slave I/O. You can also select the [Clear All] button to set all remote inputs to Not used.
Load	Reads the assigned remote inputs and outputs from a file on the PC and save it in the controller.
Save	Saves the assigned remote inputs and outputs shown in the dialog to a file on the PC.

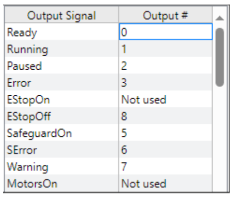
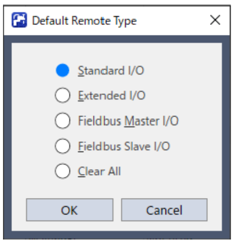
KEY POINTS

Both the remote inputs and outputs are loaded or saved together when using [Load] or [Save].

6.13.2.2.8.2 [Setup]-[System Configuration]-[Controller]-[Remote Control]-[Outputs] Page

Use this page to configure the controller remote control outputs.



Item	Description
Output Signal	Displays the output signal. See details below. **Remote Outputs**
Output #	 Select an output bit to use for the corresponding output signal." Select Not used" to disable the remote output. For example, "Ready" is assigned to I/O output bit 0 in the above dialog. Select "Not used" to use this as a normal I/O output.
Default	 Click this button to set the default remote outputs. First, a dialog box will be displayed asking you which type of outputs to use for defaults: Standard, Extended I/O, Fieldbus master, or Fieldbus slave I/O. You can also select the [Clear All] button to set all remote outputs to Not used.
Load	Reads the assigned remote inputs and outputs from a file on the PC and save it in the controller.
Save	Saves the assigned remote inputs and outputs shown in the dialog to a file on the PC.

> **KEY POINTS**
>
> Both the remote inputs and outputs are loaded or saved together when using [Load] or [Save].

6.13.2.2.8.3 [Setup]-[System Configuration]-[Controller]-[Remote Control]-[User Outputs] Page

Adds a user-defined output remote I/O. See details below.

User-defined Remote Output I/O

6.13.2.2.8.4 [Setup]-[System Configuration]-[Controller]-[Remote Control]-[Ethernet]

Enables the Remote Ethernet function. See details below.

Remote Ethernet

6.13.2.2.8.5 [Setup]-[System Configuration]-[Controller]-[Remote Control]-[RS232] Page

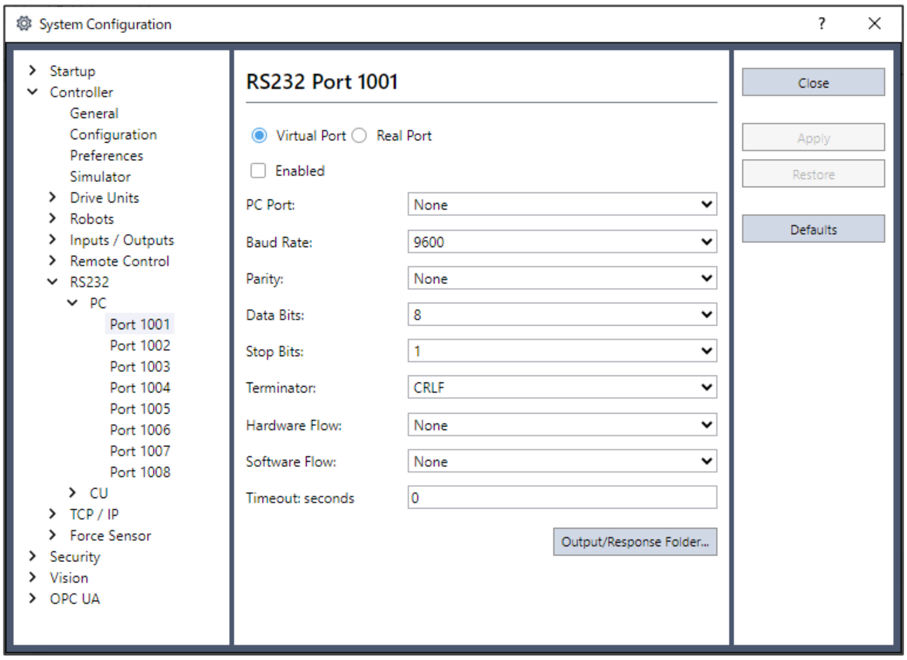
Enables the remote RS232 function. See details below.

Remote RS232

6.13.2.2.9 [Setup]-[System Configuration]-[Controller]-[RS232]

6.13.2.2.9.1 [Setup]-[System Configuration]-[Controller]-[RS232]-[PC] Page

Use this page to configure the RS232 ports on PC.



To configure an RS-232 port

1. In the Epson RC+ 8.0 menu, select [Setup]-[System Configuration]-[RS232]-[PC]. Select a port from the tree.
2. Select the [PC port] and change the settings as desired.

3. Set the [Enabled] checkbox.
4. Click the [Apply] button to save the new settings.
5. Click the [Close] button to close the dialog.

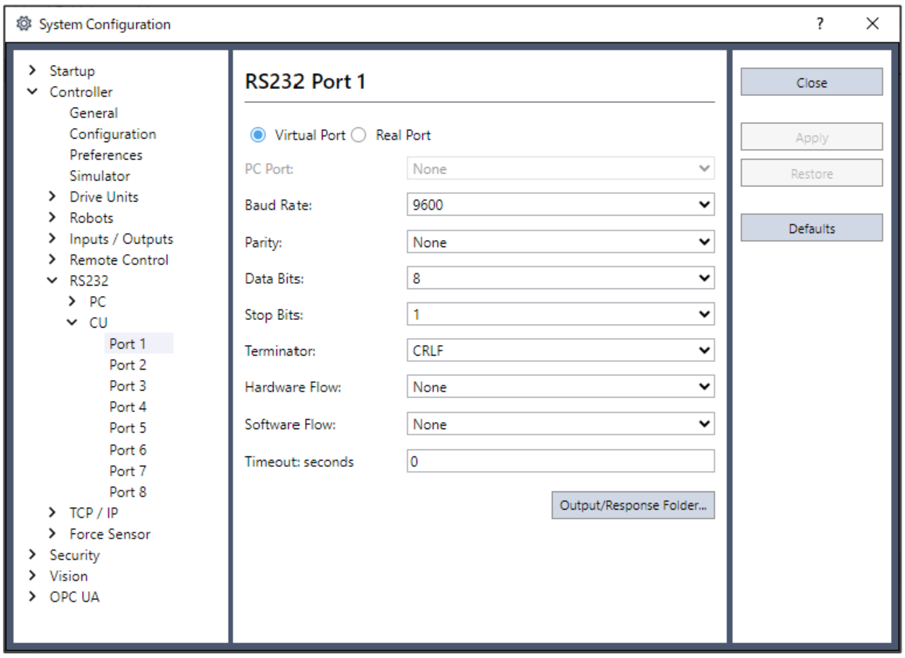
KEY POINTS

If set to a virtual port, the [Communication response / Output folder...] button appears. This opens the communication input/output file folder. See details below.

Restriction on SPEL+ command execution

6.13.2.2.9.2 [Setup]-[System Configuration]-[Controller]-[RS232]-[CU] Page

There is one page for each RS-232C port. If there are no RS232C ports installed in the special slot, then no selections are visible in the tree.



To configure an RS-232 port

1. In the Epson RC+ 8.0 menu, select [Setup]-[System Configuration]-[RS232]-[Controller]. Select a port from the tree.
2. Change the settings as desired.
3. Click the [Apply] button to save the new settings.
4. Click the [Close] button to close the dialog.

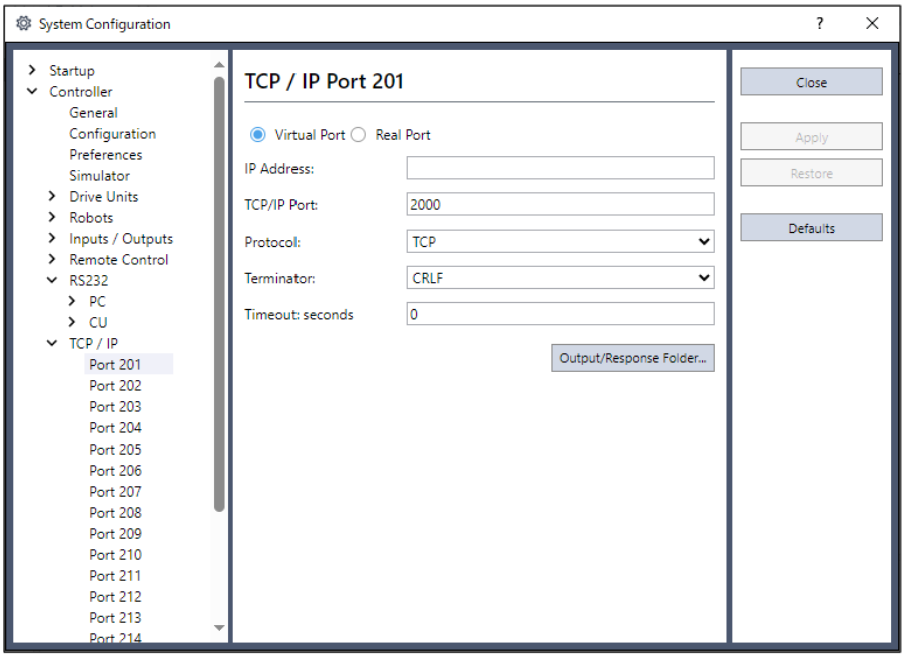
KEY POINTS

If set to a virtual port, the [Communication response / Output folder...] button appears. This opens the communication input/output file folder. See details below.

Restriction on SPEL+ command execution

6.13.2.2.10 [Setup]-[System Configuration]-[Controller]-[TCP/IP] Pages

There is one page for each TCP / IP port in the controller.



To configure a TCP/IP port

1. In the Epson RC+ 8.0 menu, select [Setup]-[System Configuration]-[TCP/IP]. Select a port from the tree.
2. Enter the IP address in the [IP Address] dialog box.
3. Enter the TCP/IP port number in the [TCP/IP Port] box. This must be the same port number that is used on the host device. It must be different from any of the other TCP/IP port numbers used for the other TCP/IP ports.
4. Change the other settings as desired.
5. Click the [Apply] button to save the new settings.
6. Click the [Close] button to close the dialog.

> **KEY POINTS**
>
> If set to a virtual port, the [Communication response / Output folder...] button appears. This opens the communication input/output file folder. See details below.
>
> **Restriction on SPEL+ command execution**

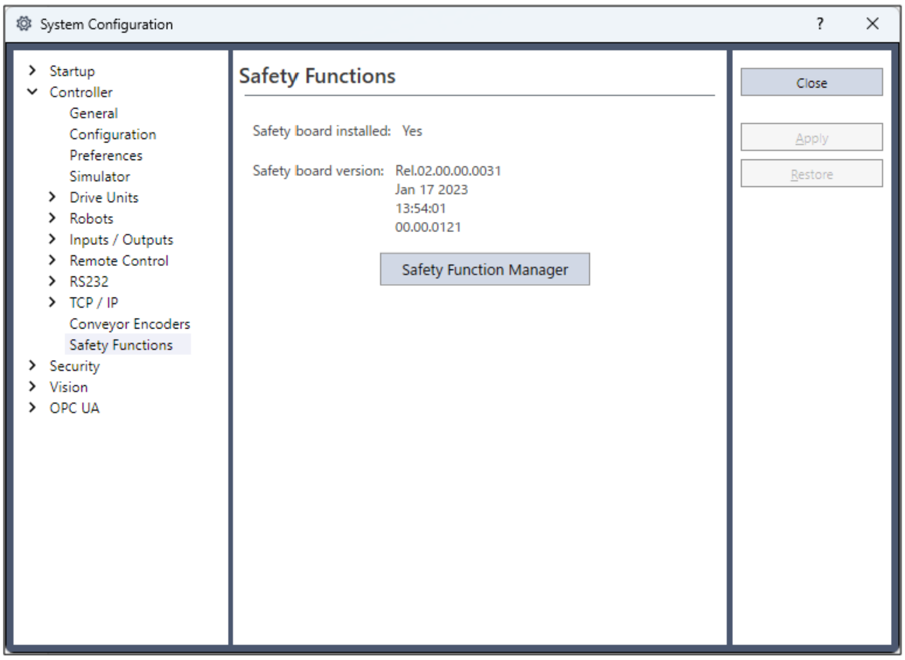
6.13.2.2.11 [Setup]-[System Configuration]-[Controller]-[Conveyor Encoders]

For more information on conveyor encoders, see below.

Conveyor Tracking

6.13.2.2.12 [Setup]-[System Configuration]-[Controller]-[Safety Functions]

Displays information of safety board when using safety function (the Controller with Safety Board). Also, it is possible to start the safety function manager and change the parameter. It won't be displayed if the Controller is not with the safety board, or the safety board is not installed in the dedicated slot.



Item	Description
Safety board installed	Displays whether the safety board is installed or not.
Safety board version	Displays the version of the safety board.
Safety Function Manager	Starts the safety function manager. For instructions, refer to the following manual. "Robot Controller Safety Function Manual - Setting Safety Functions (Setting Software: Safety Function Manager)"

6.13.2.2.13 [Setup]-[System Configuration]-[Controller]-[Force Sensor]

For more details on the force sensor, refer to the following manual:

"Epson RC+ 8.0 option Force Guide 8.0"

6.13.2.3 [Setup]-[System Configuration]-[Security]

For more information on security, see below.

Security

6.13.2.4 [Setup]-[System Configuration]-[Vision]

For instructions, refer to the following manual.

"Vision Guide 8.0 Hardware & Setup - Software Configuration"

6.13.2.5 [Setup]-[System Configuration]-[OPC/UA]

For more details on OPC UA, refer to the following manual:

"Robot Controller Option OPC UA Server"

6.13.3 [Preferences] Command (Setup Menu)

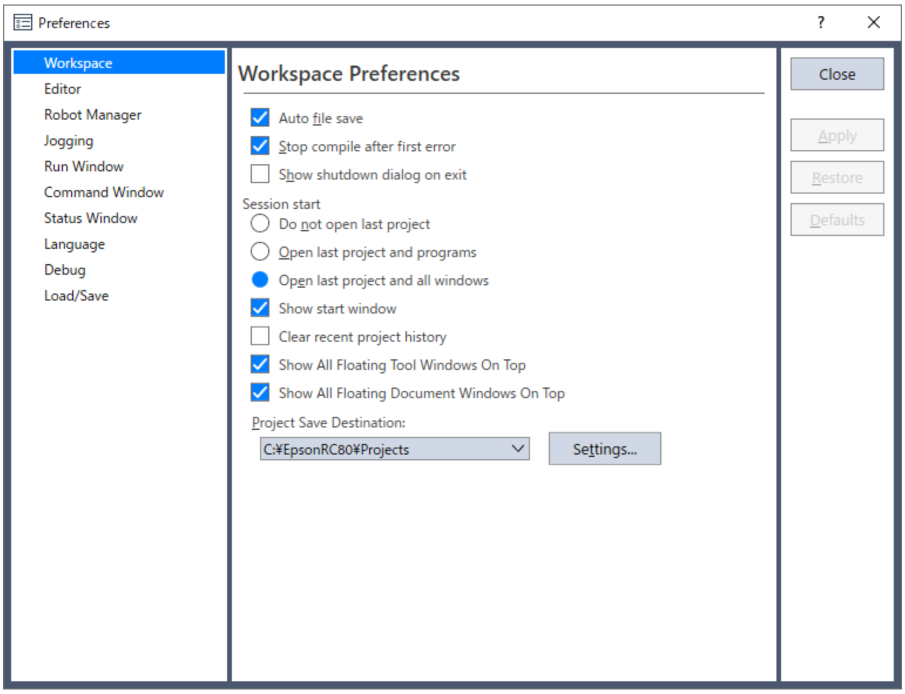
Select the Epson RC+ 8.0 menu-[Setup]-[Preferences] to display the [Preferences] dialog. Select the tree on the left of the dialog to change the display on the right.

Common buttons

Item	Description
Close	Closes the [Preferences] dialog.
Apply	Saves changes.
Restore	Revert back to previous values.
Defaults	Reverts back to the default settings.

6.13.3.1 [Setup]-[Preferences]-[Workspace] Page

From this page, you can configure your workspace preferences.

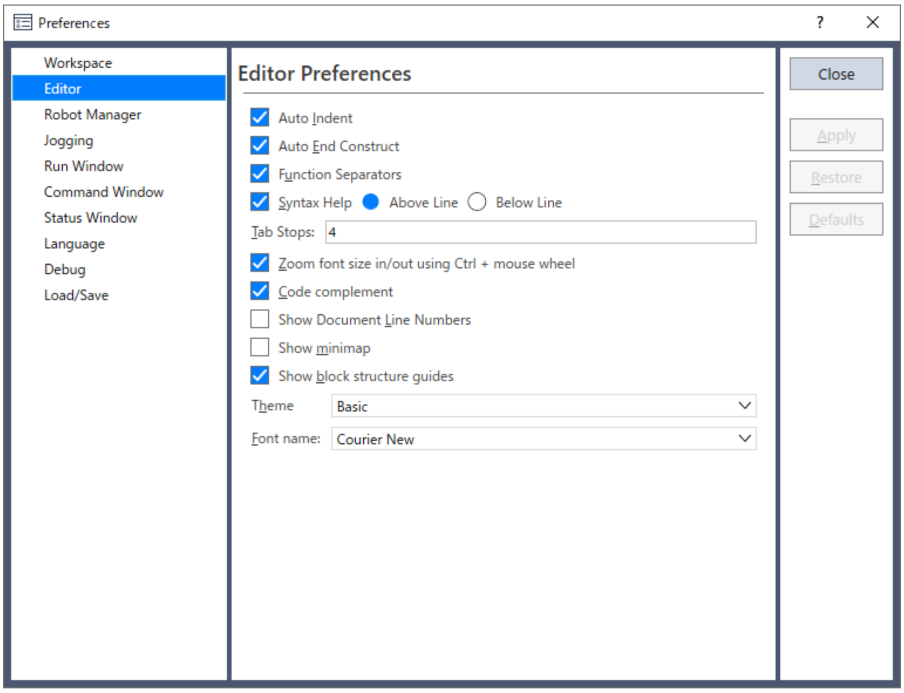


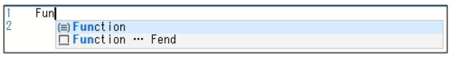
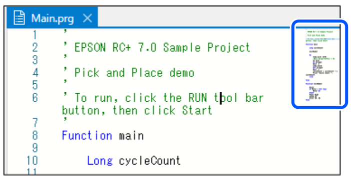
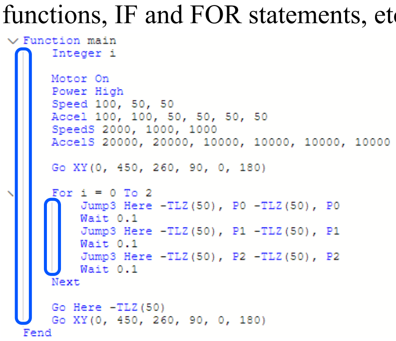
Item	Description
Auto file save	Checking this box will cause Epson RC+ 8.0 to automatically save any open files before executing a command that requires the file to be saved. For example, if a file needs to be saved before executing a project build, the file will automatically be saved before running the build. However, a message will appear to check whether you want to save changes, regardless of the setting, when closing Epson RC+ 8.0. This is turned on by default.
Stop compile after first error	Stops compile after first error occurs. This will make it easier to spot the first error on the [Project Build Status] window. You can fix errors one by one. This is turned on by default.
Display the shutdown dialog on exit	Displays the shutdown dialog when closing the Epson RC+ 8.0. See details below. This is turned off by default. **Exit Command (File Menu)**

Item	Description
Do not open last project	If this radio button is selected, the last project will not be opened when Epson RC+ 8.0 is started. This is turned off by default.
Open last project and program file	If this radio button is selected, the last project will be opened and any program windows that were previously opened will be opened. This is turned off by default.
Open last project and all windows	If this radio button is selected, the last project will be opened and all windows will be restored to their previous locations. This is the default setting.
Display Start Window.	If this checkbox is on, Start Window will be displayed when Epson RC+ 8.0 is started. This is turned on by default.
Clear recent project history	If this checkbox is selected, the [Start Window]-[Recent Projects] history, and in the RC+ Menu, [Projects]-[Recently Projects] history will be deleted when Epson RC+ 8.0 is closed. This is turned off by default.
Show All Floating Tool Window On Top	You can display floating Tool windows at the front of the main window. This is turned on by default.
Show All Floating Document Window On Top	You can display floating Document windows at the front of the main window. This is turned on by default.
Project save destination	Select the project save destination from the drop down list.
Configure	Sets the project save destination. Click the [Preferences] to display the [Manage project save destination] dialog box. Adds or deletes a project save destination.

6.13.3.2 [Setup]-[Preferences]-[Editor] Page

This page is used to configure your preferences for the program editor windows.

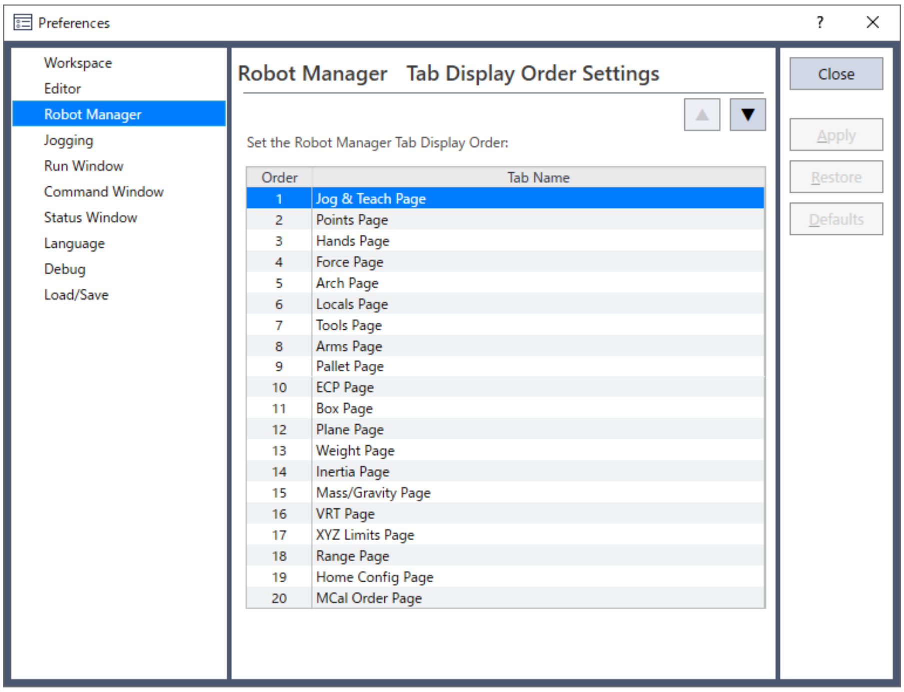


Item	Description
Auto Indent	Check this box if you want new lines to follow the indentation for the previous line. Also, lines will automatically be indented after Do, If, Else, For, Select, and Case statements. This is turned on by default.
Auto end-of-block input	Automatically generates structured instruction terminations in Epson RC+. For example, entering a "for" statement will automatically add "Next". This is turned on by default.
Function Separators	Check this box to display a line after each Fend statement. This is turned on by default.
Syntax Help	Check this box to enable the Syntax Help keyword list and Syntax Help window. The Syntax Help keyword list appears when typing a keyword. The Syntax Help window displays syntax for a keyword after it has been typed, as shown below.  This is turned on by default.
Above Line	Select this button to display the syntax help window above the input line.
Below Line	Select this button to display the syntax help window below the input line.
Tab Stops	Type in the number of columns to move for the TAB key. The default is 4.
Zoom font size in/out Ctrl + mouse wheel	Move the mouse wheel up or down while holding down the Ctrl key to zoom the font size in and out on the Editor window. This is turned on by default. This is turned on by default.
Code complement	Input suggestions are shown while entering keywords, as shown below.  Select with the [↑] or [↓] keys on the keyboard, and then press [Enter] or click the mouse to enter the code. This is turned on by default.
Show Line Number	Shows the line on the left side of the editor. This is turned off by default.
Show minimap	Shows the mini map on the right side of the editor, as shown below.  This is turned off by default.
Show block structure guide	As shown below, guide lines are displayed in the control block structure and indented sections of functions, IF and FOR statements, etc.  This is turned on by default.

Item	Description
Theme	Select a theme, and change the editor background and text color. Choose from Basic, Dark, or Dark (High Contrast). "Basic" is selected by default.
Font Name	Changes the font displayed in the editor.

6.13.3.3 [Setup]-[Preferences]-[Robot Manager]

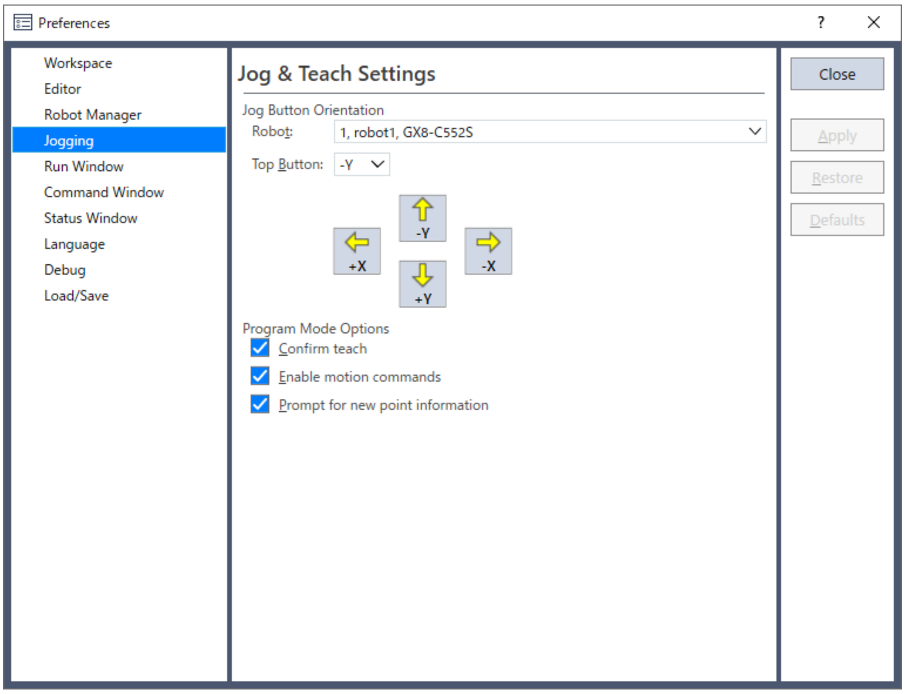
Sets the Robot Manager tab display order.



Item	Description
▼▲	Sorts the tab display order list.
Setting the Robot Manager tab display order	Sets the Robot Manager tab display order.
Order	Shows the display order of the tabs.
Tab name	Shows the Robot Manager tab name.

6.13.3.4 [Setup]-[Preferences]-[Jogging] Page

Set [Jog & Teach].



Setting Jog Button Orientation

Item	Description
Robot	Select a robot.
Top Button	Change the Jog button shown at the top. You can change the orientation of the jogging buttons and arrow keys for the X and Y axes. You can also click on one of the buttons to change it to the top button position. The jog button orientations are useful for "aligning" your PC monitor with the robot's Cartesian coordinate system. Align the buttons so that the robot moves in the direction of the arrows.

Program Mode Options

This is an option when using [Robot Manager]-[Jog & Teach] in program mode.

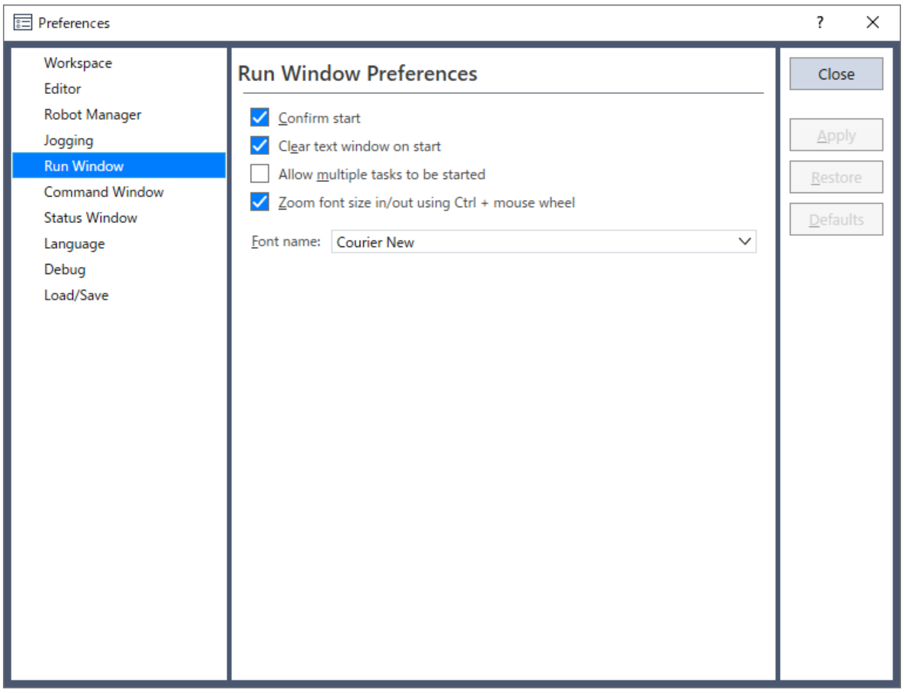
Item	Description
Confirm teach	Displays a confirmation dialog box prompting the user to register the current position when clicking the [Teach] button on the [Jogging]-[Teach] tab. This is turned on by default.
Enable motion commands	Check this box if you want to execute motion commands (Go, Jump, etc.) from the [Jog & Teach]-[Execute Motion] tab. This is turned on by default.
Prompt for new point information	Display the [New Point Information] dialog when an undefined point is taught. This is turned on by default.

KEY POINTS

[Robot Manager] is disabled when used from the operator window in operator mode or from RC+ API. To configure the Robot Manager for operators, see [Project]-[Properties]-[Operator Settings]-[Robot Manager].

6.13.3.5 [Setup]-[Preferences]-[Run Window] Page

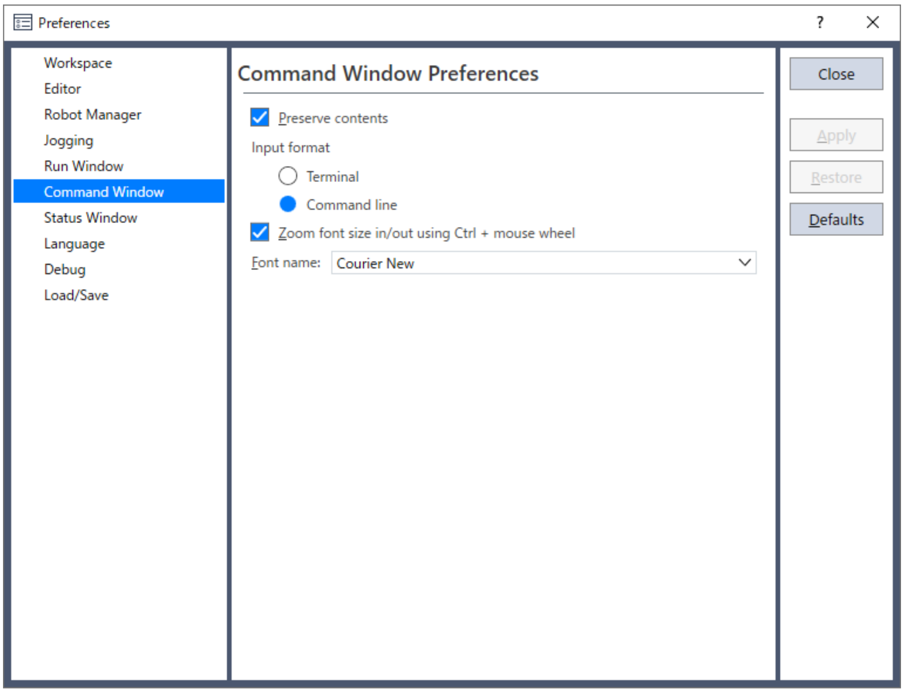
This page allows you to change preferences for the Run Window.



Item	Description
Confirm Start	This checkbox allows you to select if you want to see a confirmation message box before a program is started. This is turned on by default.
Clear text window on start	Checking this will cause the Run Window text pane to be cleared each time the [Start] button is clicked. This is turned on by default.
Allow multiple tasks to be started	Checking this allows you to start a task from the Run window while other tasks are running. The [Start] button will not be disabled after starting a task. This is turned off by default.
Zoom font size in/out Ctrl + mouse wheel	Move the mouse wheel up or down while holding down the Ctrl key to zoom the font size in and out on the Run window. This is turned on by default.
Font Name	Click on the Font button to change the font for the Run window.

6.13.3.6 [Setup]-[Preferences]-[Command Window] Page

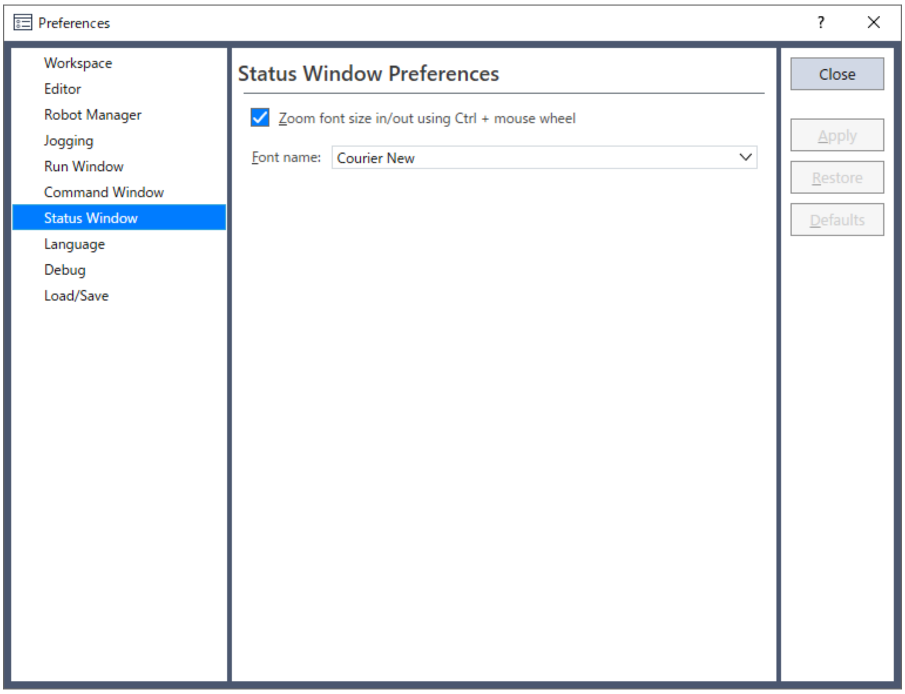
This page allows you to change preferences for the Command Window.



Item	Description
Preserve contents	Checking this option will cause the command window to preserve its contents between sessions. Deselect this checkbox to clear contents when closing the Command window. This is turned on by default.
Input format	Set the command input format. ▪ Terminal: The same input method as EPSON RC+ 7.0 and earlier. Move the cursor to a previously executed command row, and execute the command again. You can also rewrite some commands to execute them. ▪ Command line: An input method that resembles the Windows Command Prompt. Enter the command on the final row of the Command window to output results. Executed commands are displayed in sequence when pressing the [↑] or [↓] keys. Default: Terminal
Zoom font size in/out Ctrl + mouse wheel	Move the mouse wheel up or down while holding down the Ctrl key to zoom the font size in and out on the Command window. This is turned on by default.
Font Name	Click on the Font button to change the font for the Command window.

6.13.3.7 [Setup]-[Preferences]-[Status Window] Page

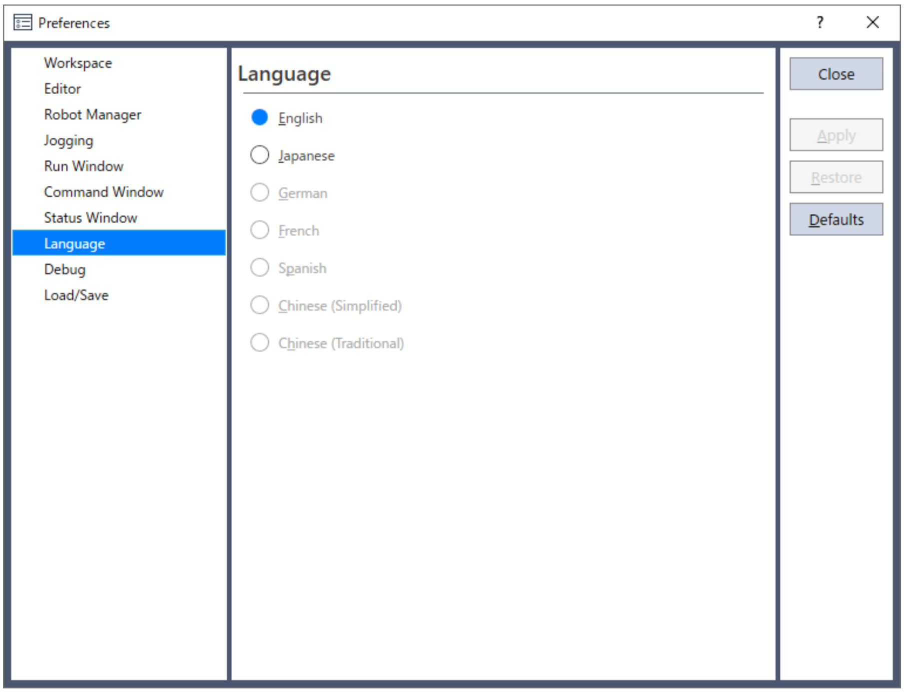
This page allows you to change preferences for the Status window.



Item	Description
Zoom font size in/out Ctrl + mouse wheel	Move the mouse wheel up or down while holding down the Ctrl key to zoom the font size in and out on the Status window. This is turned on by default.
Font Name	Click on the Font button to change the font for the Status window.

6.13.3.8 [Setup]-[Preferences]-[Language] Page

This page allows you to change the Epson RC+ 8.0 GUI language.



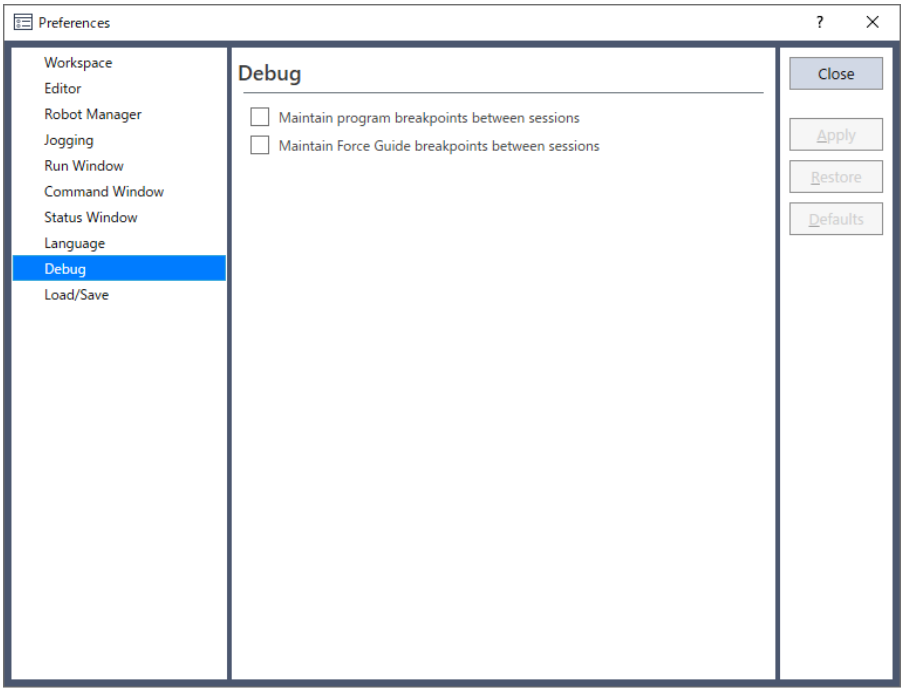
- When Epson RC+ 8.0 is installed on a Windows system using a Western language, then the English, German, French, and Spanish selections are available.
- When it is installed on a Windows system using Japanese, then English and Japanese are available.

- When it is installed on a Windows system using Chinese, then English, Chinese (Simplified), and Chinese (Traditional) are available.

Select the desired language and click the [Apply] button to change languages.

Item	Description
Language	This set of option buttons allows you to choose which language to use for the Epson RC+ 8.0 GUI.

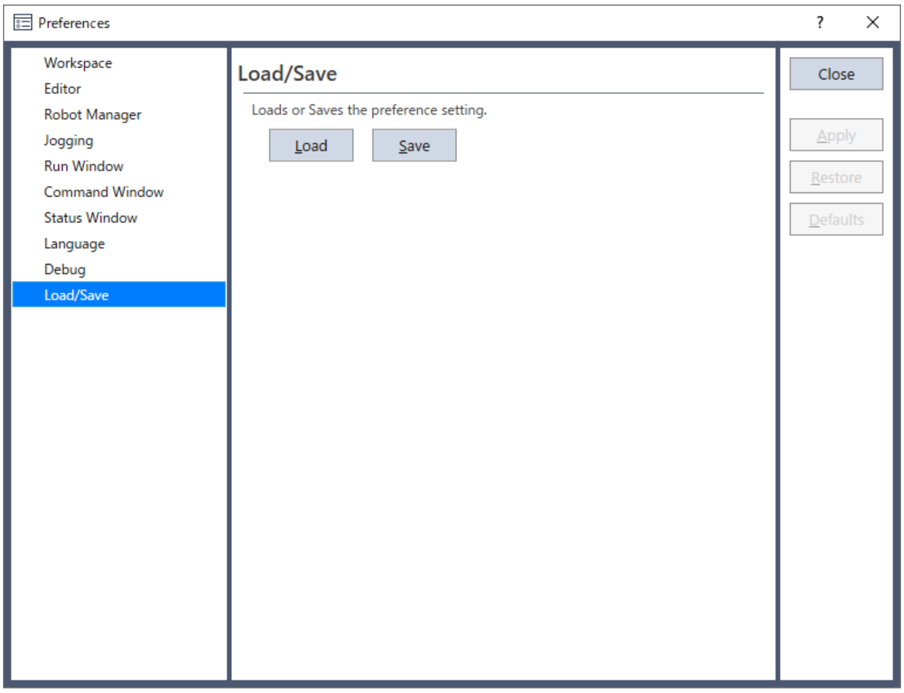
6.13.3.9 [Setup]-[Preferences]-[Debug] Page



Item	Description
Maintain program breakpoint between sessions	Saves breakpoints set in the program file, and automatically restores breakpoints when next opening projects. This is turned off by default.
Maintain Force Guide breakpoint between sessions	Saves breakpoints set in Force Guide, and automatically restores breakpoints when next opening projects. This is turned off by default.

6.13.3.10 [Setup]-[Preferences]-[Load/Save] Page

This page allows you to load or save preferences.



Item	Description
Load	Reads the preferences previously saved on the PC.
Save	Saves the preferences to a file on the PC.

6.13.4 [License Configuration] Command (Setup Menu)

This dialog allows you to view and enable licenses corresponding to the Controller.

For details on enabling a license, see below.

Installing Controller License

6.14 [Window] menu

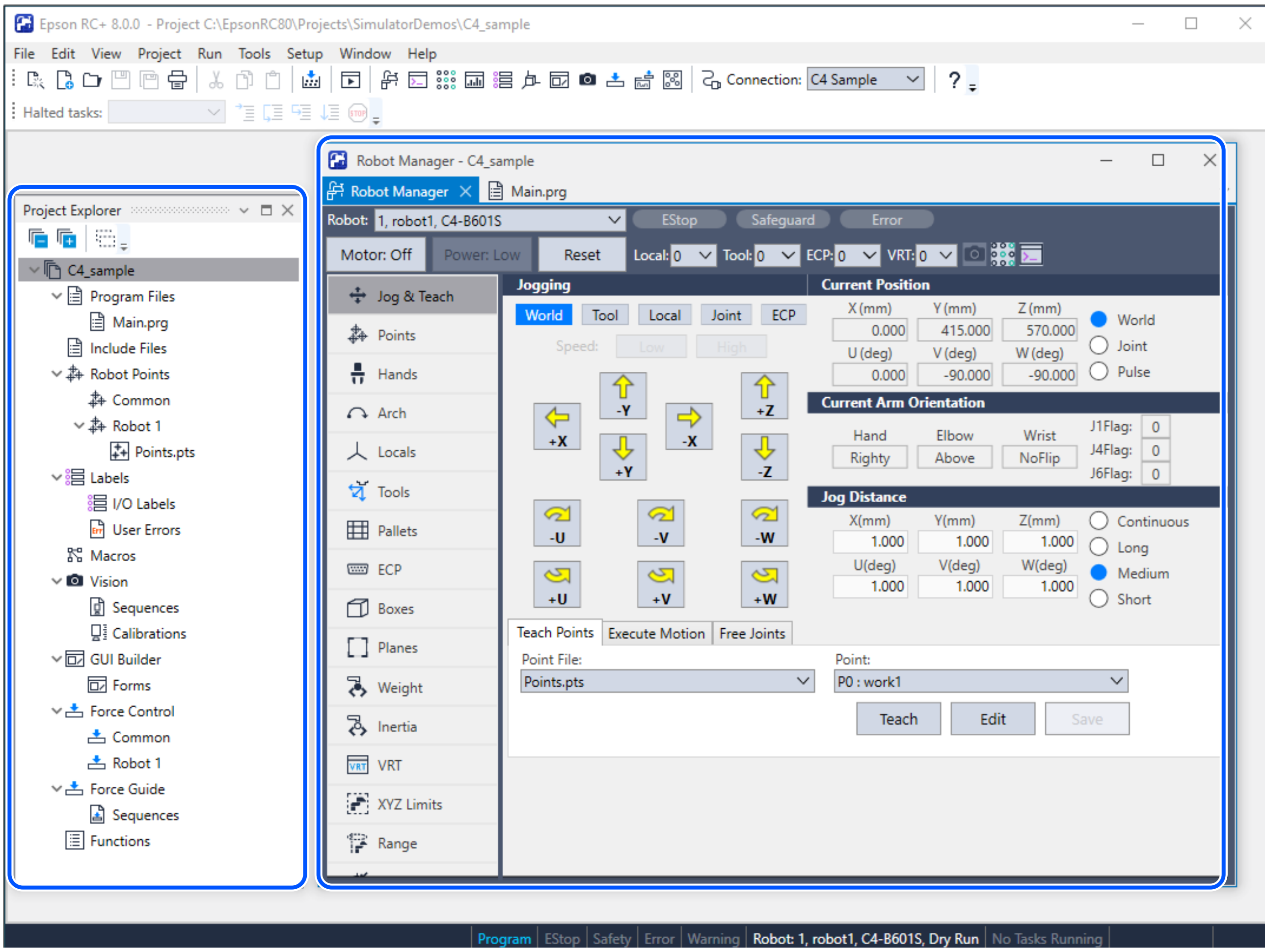
The [Window] menu contains selections for managing the currently open Epson RC+ 8.0 child windows.

6.14.1 [Floating] (Window Menu)

Ctrl + Shift + F10

The selected subwindow can be put into a floating state (displayed as a separate window). Docking windows are floated in either of the following ways.

- Move the window by the title bar or tab to the docking position.
- Right-click on the title bar or tab of the window, and then select [Floating].

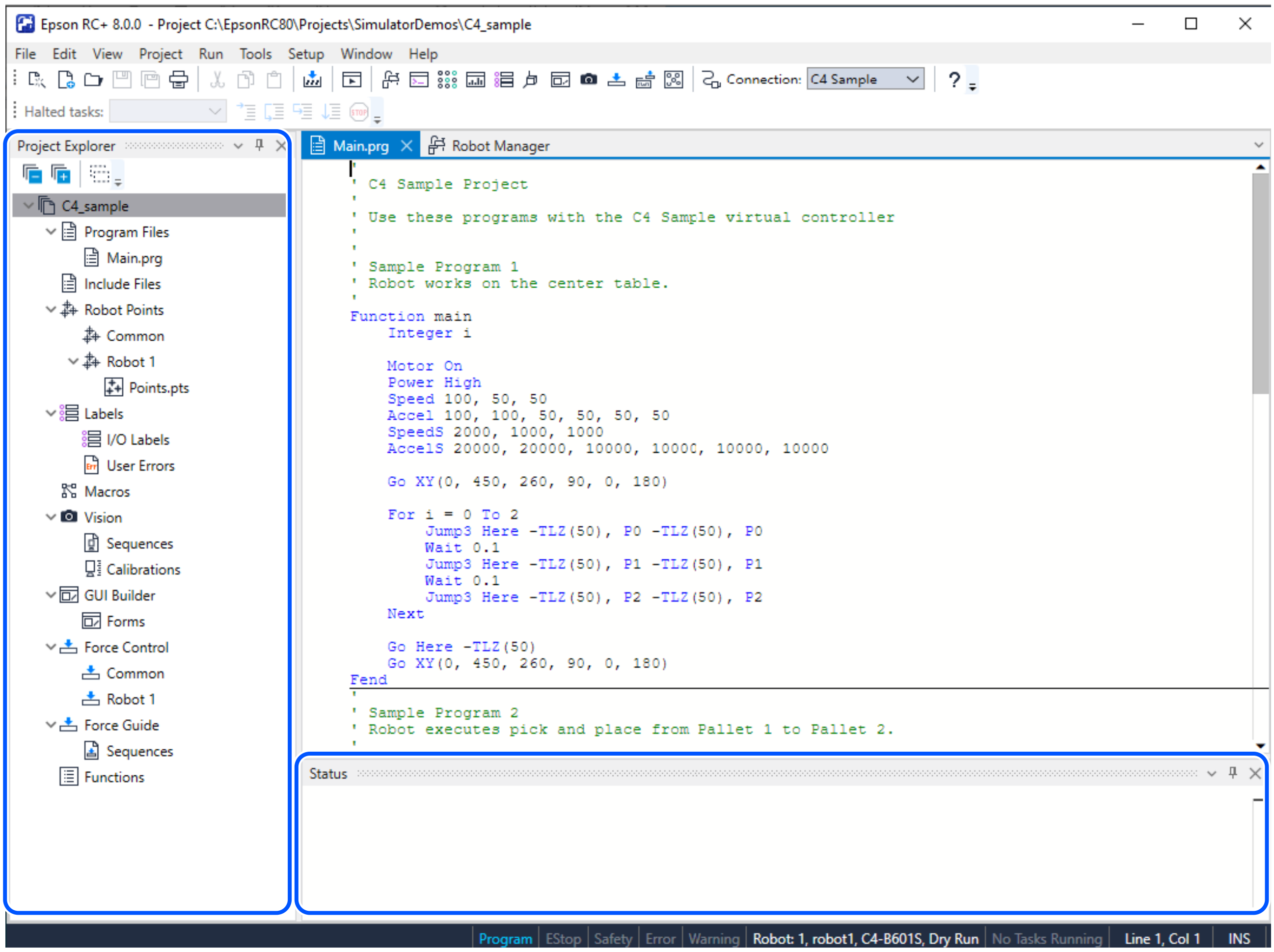


6.14.2 [Docking] (Window Menu)

Ctrl + Shift + F11

This docks the floated Tool window in its original position. Floating windows are docked in either of the following ways.

- Drag the window to the docking position
- Right-click on the title bar of the window, and then select [Docking].

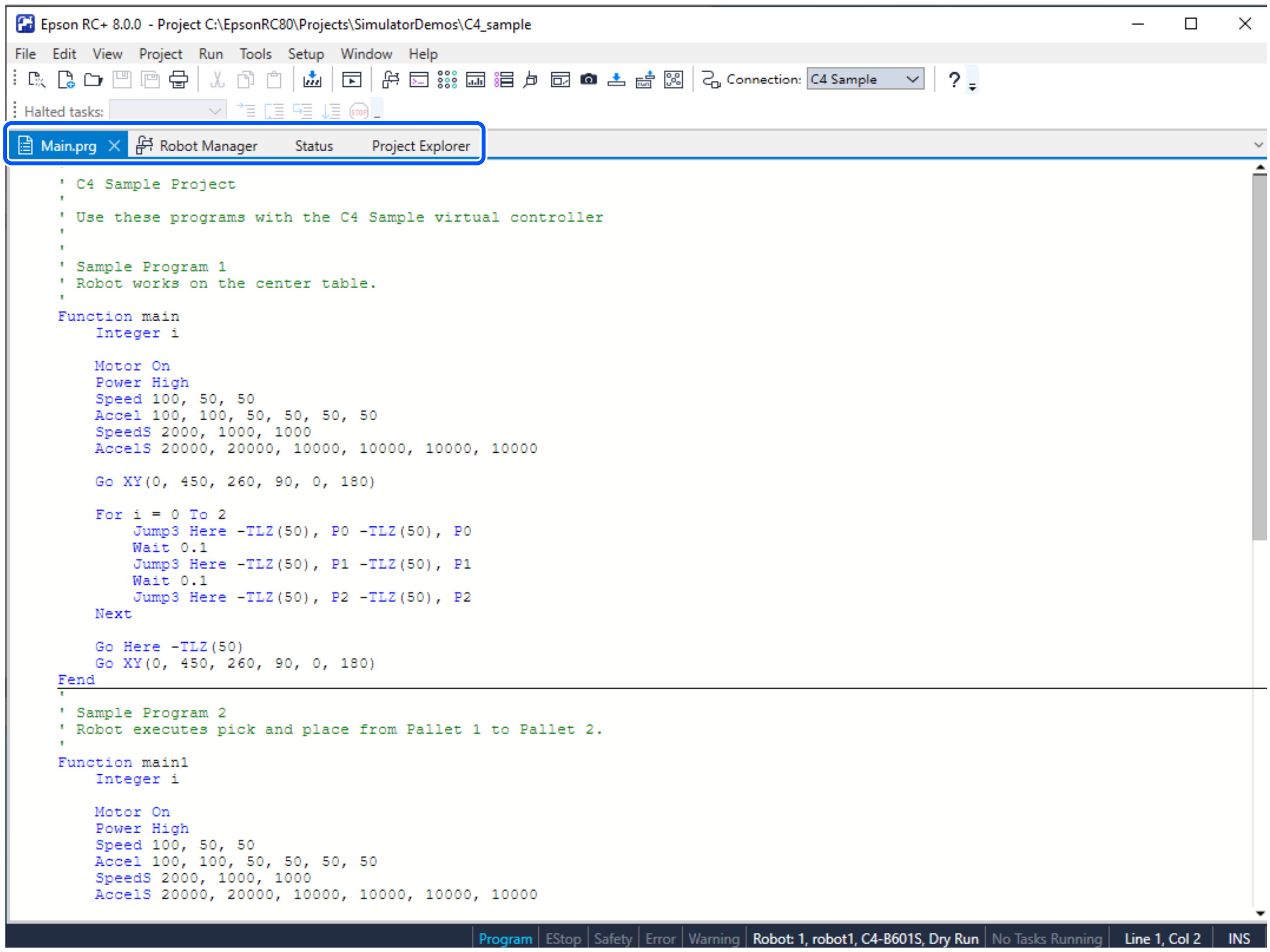


6.14.3 [Docking As Tabbed Document] (Window Menu)

Ctrl + Shift + F12

This docks the floated subwindow in the main area. Floating windows are docked in either of the following ways.

- Drag the window to the main area
- Right-click on the title bar of the window, and then select [Docking As Tabbed Document].



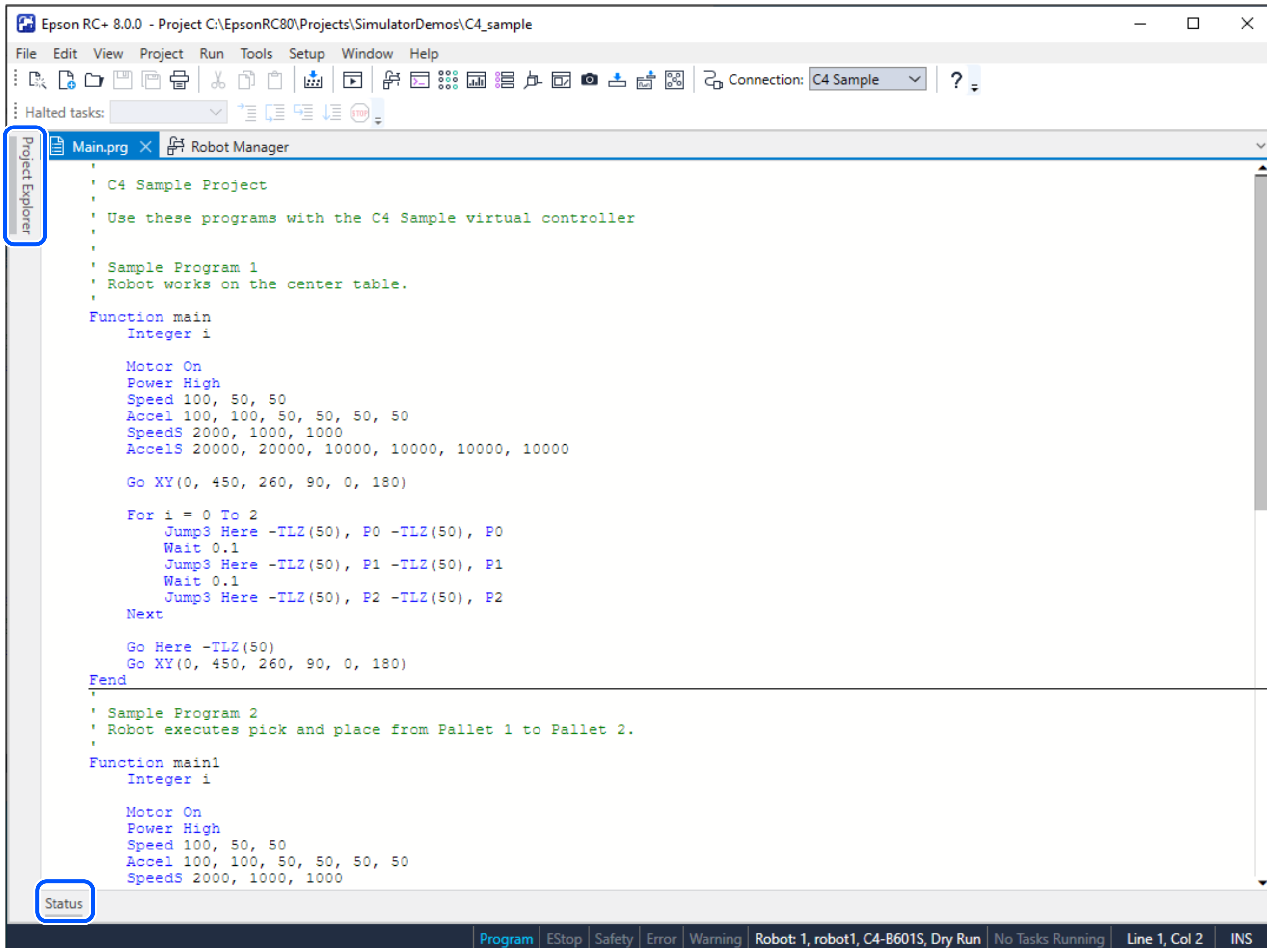
6.14.4 [Auto Hide] (Window Menu)

This minimizes unused Tool windows as a tab. To display hidden windows again, click on the minimized tab. Tool windows displayed from tabs overlap on the document window, making this useful when working with a small monitor. Tool windows are automatically hidden in either of the following ways.

- Right-click on the title bar of the Tool window, and select [Auto Hide].
- Click on the [Auto-Hide] icon at the top of the Tool window

TIP

When displaying the window, click the [Auto-Hide] icon to disable [Auto Hide].



6.14.5 [Hide] Command (Window Menu)

Hides the selected window.

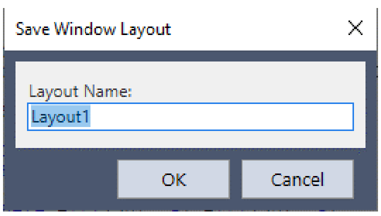
In the Epson RC+ 8.0 menu, select [Show] to redisplay the window.

6.14.6 [Save Window Layout] Command (Window Menu)

Names and saves the current window layout information. Execute [Apply Window Layout] to read the saved layout information and restore the window layout.

KEY POINTS

You can save up to eight layouts.

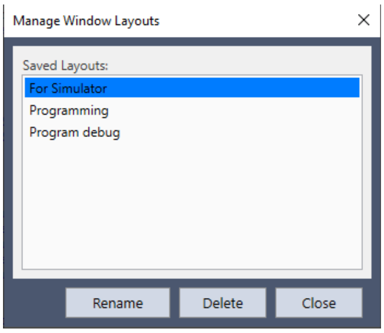


6.14.7 [Apply Window Layout] Command (Window Menu)

Reads the window layout information saved in [Save Window Layout] to restore the window layout.

6.14.8 [Manage Window Layouts] Command (Window Menu)

Renames or deletes saved window layouts.



Item	Description
Rename	Renames the window layout.
Delete	Deletes the window layout.
Close	Closes the [Manage Window Layouts] dialog.

6.14.9 [Reset Window Layout] Command (Window Menu)

You can reset the window layout to the default setting. You can choose from [Standard] or [Layout for multiple displays].

See details below.

Reset Window Layout

6.14.10 [Close all windows] Command (Window Menu)

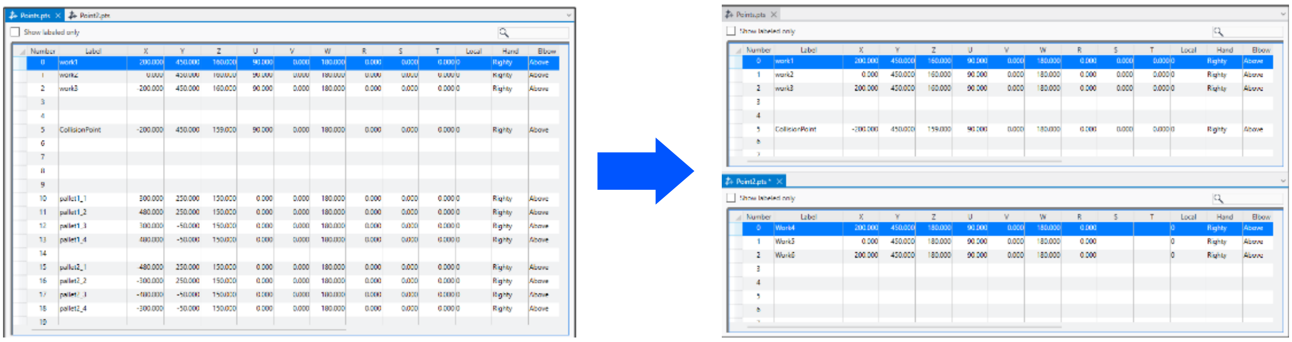
This command closes all windows.

KEY POINTS

Project Explorer and Status Window are not included.

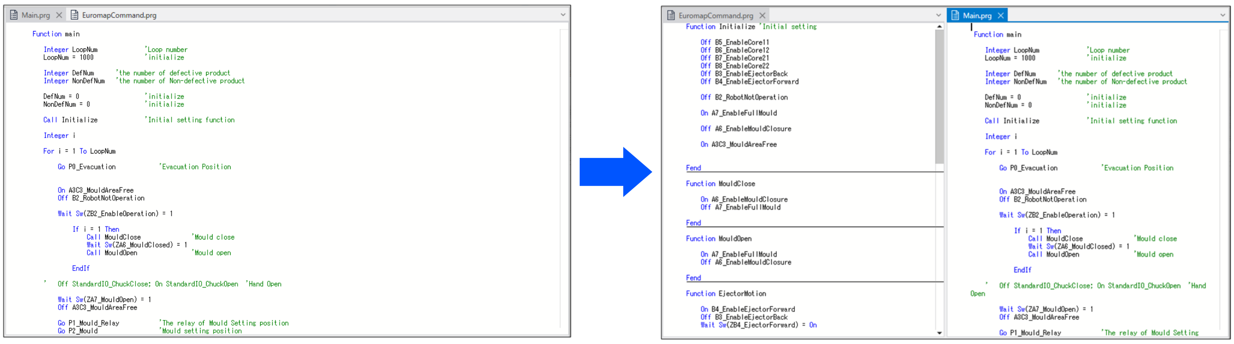
6.14.11 [New Horizontal Tab Group] (Window Menu)

You can split selected subwindows on the top and bottom to create a new tab group. This is useful when simultaneously referencing point data and other tabular data.



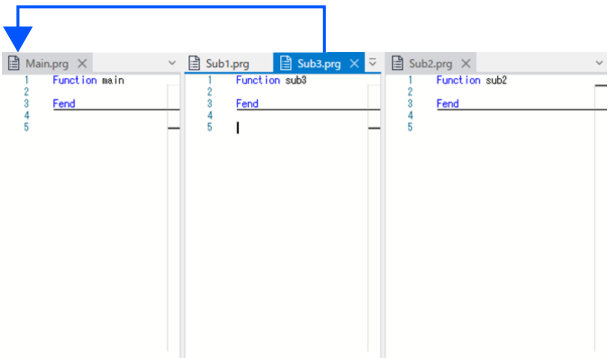
6.14.12 [New Vertical Tab Group] (Window Menu)

You can split selected subwindows on the left and right to create a new tab group. This is useful when simultaneously referencing programs and other vertically formatted data.



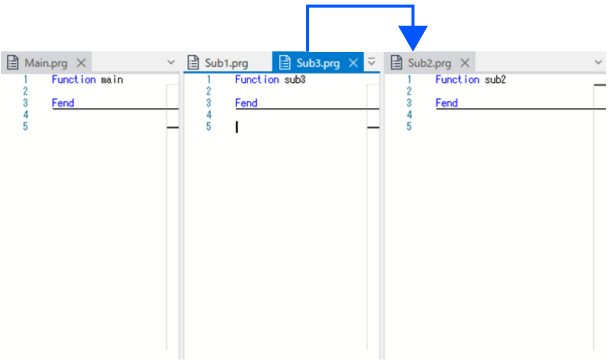
6.14.13 [Move To Previous Tab] (Window Menu)

Moves the selected subwindow to the previous tab group.



6.14.14 [Move To Next Tab Group] (Window Menu)

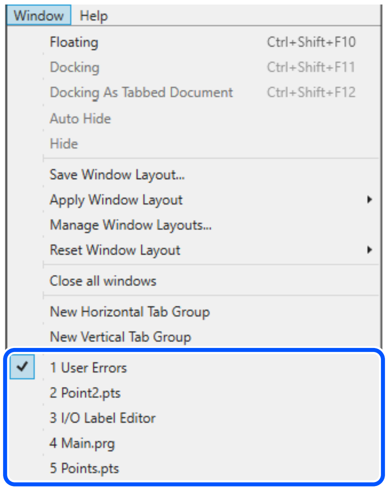
Moves the selected subwindow to the next tab group.



6.14.15 List of Open Windows (Window Menu)

A list of currently open document windows is displayed at the bottom of the [Window] menu.

When you choose an open window from the listing, you make that document active. A check mark appears in front of the document name of the currently active window.



6.15 [Help] Menu

Shows help, manuals, and version information.

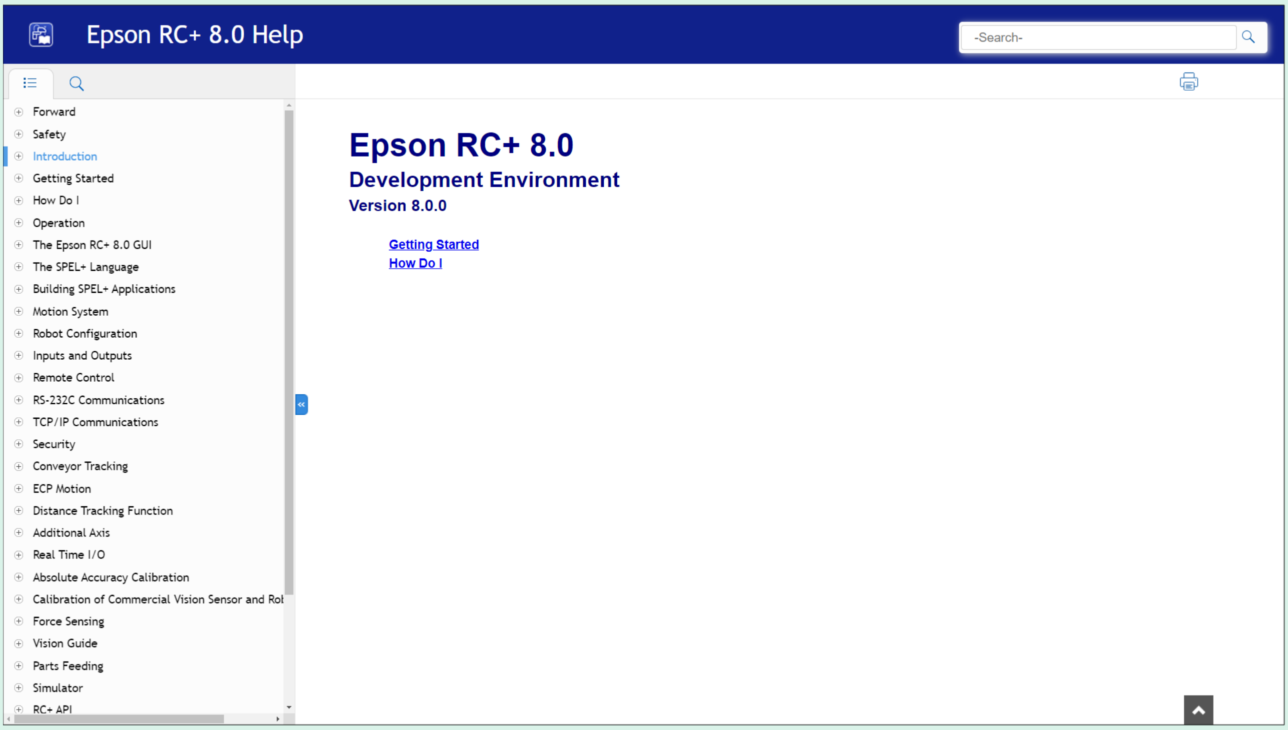
6.15.1 [Help] (Help Menu)

Ctrl+F1

Shows help in the browser.

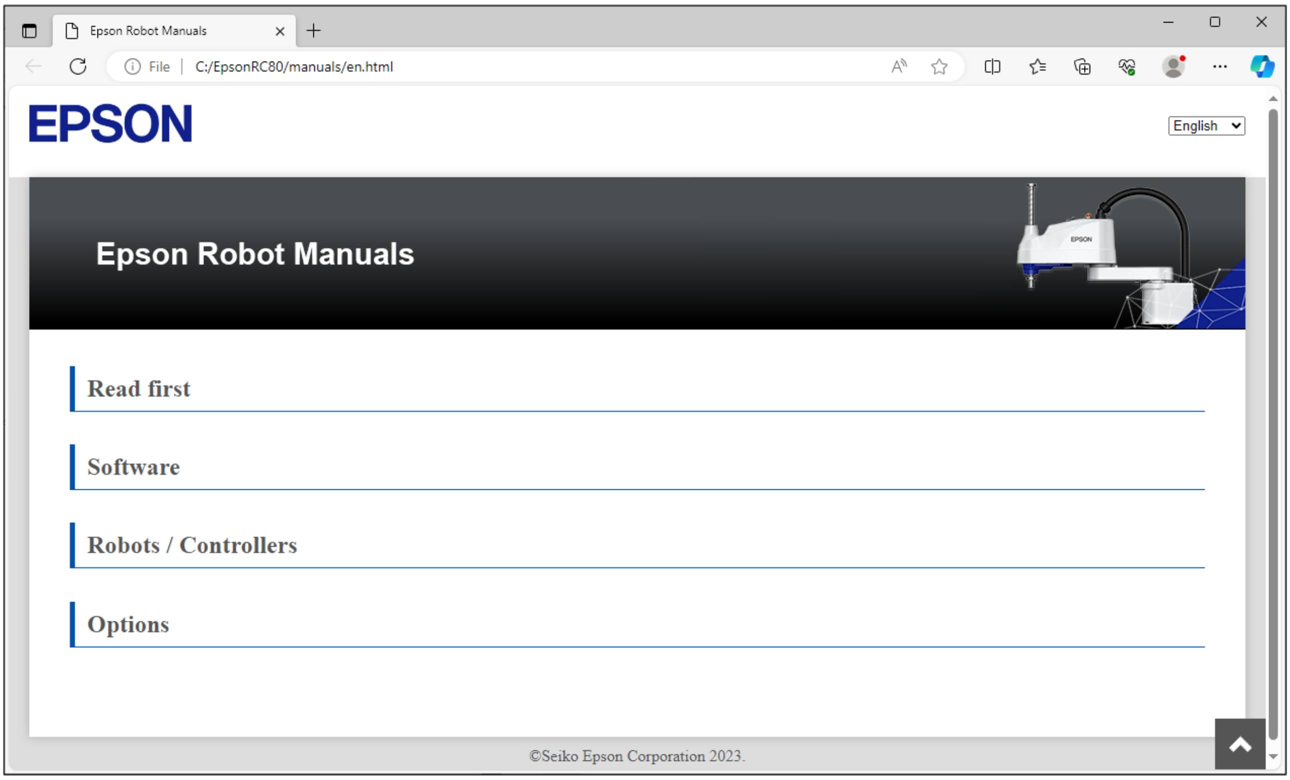
KEY POINTS

Recommended browser: Browsers which is compatible with HTML5 (latest Google Chrome, Microsoft Edge, etc.)



6.15.2 [Manuals] (Help Menu)

Shows a list of manuals in the browser. The Help Menu Manuals submenu contains selections for each of the manuals in PDF format. These include, the Safety Manual, and manuals for Epson RC+ 8.0, SPEL+ Language Reference, Controller, Robot, and the Options.



KEY POINTS

Manuals are provided in PDF. To view the manuals, use the PDF viewer included with Windows. You can also install Adobe Acrobat Reader or other PDF viewer.

6.15.3 [About Epson RC+ 8.0] Command (Help Menu)

The About command displays a dialog box showing the current version of the Epson RC+ 8.0 software, along with copyright and license information. When calling technical support about Epson RC+ 8.0, you should report the version you are using from this dialog.

Item	Description
Close	Closes the [About Epson RC+ 8.0] dialog.
RC+ 8.0 Release Notes	Shows a list of RC+ 8.0 release notes in the browser.
OSS License	Shows the [OSS License] dialog.

7. The SPEL+ Language

This chapter contains information about the SPEL+ Language.

- Overview
- Program structure
- Commands and statements
- Function and Variable Names (Naming restriction)
- Date types
- Operators
- Working with variables
- Working with strings
- Files
- Multi-statements
- Labels
- Comments
- Error handling
- Multi-tasking
- Multiple robots
- Coordinate systems
- Robot arm orientations
- Robot motion commands
- Working with robot points
- Input and output control
- Using Traps
- Special tasks
- Background task
- Constants

- Calling native functions in dynamic link libraries

7.1 Overview

SPEL+ is a BASIC-like programming language that runs in the controller. It supports multitasking, motion control, I/O control.

Programs are written in ASCII text and then compiled into executable object files.

Several language instructions can also be executed in immediate mode from the Command window.

7.2 Program Structure

7.2.1 What is a SPEL+ program?

A SPEL+ program is a collection of functions, variables, and macros. You can put one or more statement in each line of a program. Every program file has a ".prg" extension and is stored in the project folder.

Each project must include at least one program and define the function called "main". "Function main" is the default definition. If "Function main" is not found, an error occurs.

In addition, you can define other 63 main functions in the same project. Each program has own start function: main1, main2… main63. Each of the main functions can be started from the [Operator window], the remote console, or RC+ API.

A function definition begins with the Function statement and ends with the Fend statement.

The following program file contains two function definitions. Function Main calls function "Func1".

```
MAIN.PRG
Function Main
  Call Func1
  ...
Fend
Function Func1
  Jump pickpnt
  ...
Fend
```

7.2.2 Calling functions

You can execute a user function by using the Call statement. The function can reside in any program file in the current project. A return value is returned, with the function used as an argument. You can also omit the Call statement if you don't need the return value. When Call is omitted, then parentheses for the arguments must not be supplied.

Here are some examples:

```
Call MyFunc(1, 2)
MyFunc 1, 2
Print MyFunc(1, 2)
```

7.3 Commands and Statements

Commands and statements consist of a SPEL+ instruction followed by the parameters for that instruction.

Commands are executed immediately. You can execute commands from the Command window or from the Macros dialog box.

Statements can be used only in programs.

Statements can include more than one SPEL+ instruction. When you put several statements in a line of a program (Multi-Statement), use a semi-colon (;) to separate instructions.

The maximum length for a line is 512 characters.

7.4 Function and Variable Names (Naming restriction)

The function name can include up to 64 characters. The variable name can include up to 32 alphanumeric, Japanese, Chinese, or underscore characters. Characters can be upper case or lower case.

The following names are valid:

- `Function main`
- `Real real_var`
- `Integer IntVar`

Function and variable names cannot begin with an underscore.

SPEL+ keywords cannot be used as function or variable names. (Example: Go, On)

String variables must have an additional dollar sign ('$') suffix. An example is shown below.

```
Function Test
  String modname$
  Print "Enter model name:"
  Line Input modname$
  Print "model is ", modname$
Fend
```

Restrictions for naming in SPEL+ language

- Characters can be alphanumeric, Japanese, or underscore character.
- Use alphabets for the first letter.
- Characters can be upper case or lower case.
- No keywords can be used.
- Maximum limits of names are as follows. (For one –byte character)

Name	Max. limit
Point label	32
I/O label	32
User error label	32
Function name	64
Variable name	32
Line label	32

7.5 Date Types

You can declare different types of data in your program. All variables must be declared. The following table shows the different data types for the SPEL+ language.

Data Type	Size	Range
Boolean	2 byte	True or False
Byte	2 byte	-128 to +127
Double	8 bytes	-1.79E+308 to 1.79E+308 Number of significant figure is 14
Int32	4 bytes	-2147483648 to +2147483647
Int64	8 bytes	-9223372036854775808 to +9223372036854775807
Integer	2 byte	-32768 to +32767
Long	4 bytes	-2147483648 to +2147483647
Real	4 bytes	-3.40E+38 to 3.40E+38 Number of significant figure is 6
Short	2 bytes	-32768 to +32767
String	256 bytes	All ASCII characters Up to 255 characters
UByte	2 bytes	0 to +255
UInt32	4 bytes	0 to 4294967295
UInt64	8 bytes	0 to 18446744073709551615
UShort	2 bytes	0 to 65535

7.6 Operators

The following table shows the operators for the SPEL+ language.

Keyword or Symbol	Example	Description
+	A+B	Addition
-	A-B	Subtraction
*	A*B	Multiplication
/	A/B	Division
**	A**B	Exponentiation
=	A=B	Equal
>	A>B	Greater than
<	A<B	Less than
>=	A>=B	Greater than or equal
<=	A<=B	Less or than equal
<>	A<>B	Not equal

Keyword or Symbol	Example	Description
And	A And B	Performs logical and bitwise AND operation.
Mod	A Mod B	Returns the remainder obtained by dividing a numeric expression by another numeric expression.
Not	Not A	Performs logical or bitwise negation of the operand.
Or	A Or B	Performs the bitwise Or operation on the values of the operands.
Xor	A Xor B	Performs the bitwise Xor operation on the values of the operand.

7.7 Working with Variables

7.7.1 Variable scopes

SPEL+ has variables with different scopes:

- Local
- Module
- Global

7.7.2 Local variables

Local variables are available to enabled statements in the same function. Functions using local variable names cannot refer to the same local variables in other functions. This is why they are called locals, because they are local to the function they are being used in.

To declare local variables in a function, use one of the variable declaration instructions at the beginning of the function after the Function statement:

Boolean, Byte, UByte, Integer, Short, UShort, Long, Int32, UInt32, Int64, UInt64, Real, Double, String

For example, the following function declares several local variables:

```
Function test
  Integer intVar1, intVar2
  Real realVar
  String dataStr$
  Integer array(10)
Fend
```

7.7.3 Module variables

Module variables are available to all functions in the same program file. To declare module variables in a program, use one of the variable declaration instructions at the beginning of the program before any Function statements:

Boolean, Byte, UByte, Integer, Short, UShort, Long, Int32, UInt32, Int64, UInt64, Real, Double, String

> ✎ TIP
>
> In order to indicate that a variable is module level, precede the name with “m_”, as shown in the example below. With this, you can improve the program readability.

For example, the following function declares several module level variables:

```
' Module level vars, used by all functions in this file
Integer m_IntVar1, m_IntVar2
Real m_RealVar
String m_DataStr$
Integer m_Array(10)
Function main
  m_IntVar1 = 25
  Call test
Fend
Function test
  Print mm_IntVar1
Fend
```

7.7.4 Global variables

Global variables can be shared between all functions in a project. The Global instruction is used to declare a global variable.

To declare global variables in a program, use the Global instruction with the desired variable type (Boolean, Byte, UByte, Integer, Short, UShort, Long, Int32, UInt32, Int64, UInt64, Real, Double, String) at the beginning of the program before any Function statements:

Boolean, Byte, UByte, Integer, Short, UShort, Long, Int32, UInt32, Int64, UInt64, Real, Double, String

For details, see Data Types.

> ✎ TIP
>
> To indicate that variables are global, precede the name with “g_” as shown in the example below. With this, you can improve the program readability.

Program: MAIN.PRG

```
Global Integer g_TotalCycles
Function main
  Call LoadPart
  ...
  ...
Fend
```

Program: LOADPART.PRG

```
Function LoadPart
  Jump pick
  On gripper
  Wait .1
  Jump place
  Off gripper
  Wait .1
```

```
  g_TotalCycles = g_TotalCycles + 1
Fend
```

7.7.5 Global Preserve variables

You can preserve global variable values by using the optional Preserve parameter when you declare global variables.

Preserved variables are stored in the controller's SRAM.

If the data type of a preserved variable or the number of dimensions is changed, the variable values will be cleared.

KEY POINTS

Be careful about the backup battery power, because you will lose the data of global preserve variables stored in SRAM if the battery is weak.

7.7.6 Arrays

You can declare local, module, and global variables with up to three dimensions as arrays for all data types.

To declare an array, use this syntax:

```
dataType name ( ubound1 [ , ubound2 [ , ubound3] ] )
```

SPEL+ arrays are zero based. The first element is referenced with a value of zero.

The total available number of array elements for local variables is 200 for strings and 2000 for all other types.

The total available number of array elements for global preserve variables is 400 for strings and 4000 for all other types.

The total available number of array elements for global and module variables is 10,000 for strings and 100,000 for all other types.

To calculate the total elements used in an array, use the following formula. (If a dimension is not used, substitute 0 for the ubound values.)

total elements = (ubound1 + 1) * (ubound2 + 1) * (ubound3 + 1)

Array declaration examples:

```
' Global string array
Global String gData$(10)
Function main
' Arrays local to this function
Integer intArray(10)
Real coords(20, 10)
```

Use Redim to change the bounds of an array at run time.

```
Integer a(10)
Redim a(20)
```

To preserve values when using Redim, add the Preserve optional argument.

```
Integer a(10)
Redim Preserve a(20)
```

Use UBound to get the maximum element number.

```
Integer i, a(10)
For i = 1 to UBound(a)
  a(i) = i
Next i
```

7.7.7 Initial values

All variables are initialized when first used except for Global Preserve variables. Strings are set to empty, and all other variables are set to zero.

7.7.8 Clearing arrays

Execute Redim (without Preserve) to clear all of the elements of array variables.

7.8 Working with Strings

A string in SPEL+ is a set of ASCII characters (Code &h01 to &hff) with a maximum length of 255.

You must declare strings in your programs with the String instruction.

All string variable names must end with a dollar sign ($) suffix.

The following table shows the string commands available in SPEL+.

Keyword	Description
Asc	Returns the decimal ASCII value of the first character in a string.
Chr$	Converts an ASCII value into a one character string.
FmtStr	Formats a numerical or date/time expression.
FmtStr$	Formats a numerical or date/time expression.
Hex$	Returns a string containing the hexadecimal value of a number.
InStr	Returns the position of a substring within a string.
LCase$	Returns the specified string in lower case characters.
Left$	Returns a substring beginning with the first character of a string.
Len	Returns the length (number of characters) of a string.
LTrim$	Returns the specified string with left spaces removed.
Mid$	Returns a substring of a string.
ParseStr	Parses a string into an array of tokens.
Right$	Returns a substring from the end of a string.
RTrim$	Returns the specified string with right spaces removed.

Keyword	Description
Space$	Returns a string containing a specified number of space (ASCII 32) characters.
Str$	Converts a number to a string.
String	Declare a string variable in a program.
Tab$	Returns a tab string.
Trim$	Returns the specified string without spaces before or after.
UCase$	Returns the specified string in upper case characters.
Val	Converts a string to a number.

7.9 Files

SPEL+ has several commands for handling files.

Keyword	Description
AOpen	Opens a file for append.
BOpen	Opens a file for binary access.
Close	Closes a file.
FileExists	Checks if a file exists.
FolderExists	Check if a folder exists.
FreeFile	Returns an unused file handle.
Input	Inputs one or more variables from a file
Del	Deletes a file.
Line Input	Inputs line from a file.
Read	Reads a specified number of bytes into a string variable.
ReadBin	Reads binary data.
ROpen	Opens a file with the read only mode.
Seek	Sets the current file pointer.
Flush	Writes data buffer to disk.
WOpen	Opens a file with the write mode.
Write	Writes the string to the file. Line terminator is not added.
WriteBin	Writes binary data.

Before using a file you must open it with one of the following commands: AOpen, Bopen, ROpen, and WOpen. And specify a file number in the Open statement. File number can be 30 to 63.

Here is an example to save a text file and read it.

```
String data$(10)
```

```
Function SaveData()
  Integer fNum, i

  fNum = FreeFile
  WOpen "c:\mydata\data.txt" As #fNum  ' Store the count
  Print #fNum, UBound(data$)
  For i = 0 To UBound(data$)
    Print #fNum, data$(i)
  Next i
  Close #fNum
Fend

Function LoadData()
  Integer fNum, i, maxNum
  fNum = FreeFile
  ROpen "c:\mydata\data.txt" As #fNum
  Input #fNum, maxNum
  Redim data$(maxNum)
  For i = 0 To UBound(data$)
    Input #fNum, data$(i)
  Next i
  Close #fNum
Fend
```

7.10 Multi-statements

A program statement can contain several statements separated by semi-colons.

The total length of a multi-statement program line cannot exceed 255 characters.

For example:

```
Function Test
  Pass P1; Pass P2; Go P3   ' Multi-statement
Fend
```

It is not recommended to use multi-statements. Multi-statements can make your code more difficult to read and debug.

7.11 Labels

A program line is an alphanumeric name followed by a colon (":") that marks a location in a program for a GoTo or GoSub statement. The name may be up to 32 characters long and can include alphanumeric characters and the underscore ("_") character if it is not the first character. You cannot use any SPEL+ keywords as labels.

For example:

```
Function Main
  Do
    Jump P1
    Jump P2
    If Sw(1) Then GoTo MainAbort
  Loop
  MainAbort:                       ' Program label
  Print "Program aborted"
Fend
```

7.12 Comments

Use comments to add notes to your programs. An apostrophe character (') starts a comment.

Example:

```
Function Main
' ****** Main Demo Program ******
  Xqt conveyor                            ' Start the conveyor task
  Do
    Print "Press ENTER to run demo cycle"
    Print "Press CTRL+C to quit"
    Input dumy
    Call demo                             ' Execute the demo function
  Loop                                    ' Return to start of main loop
```

7.13 Error Handling

When an error occurs in a SPEL+ function, you can cause execution to be transferred to an error handling routine for processing the error. The routine must be inside a function definition.

The table on the next page shows the program instructions that are used for error handling.

Item	Description
OnErr	Use the OnErr statement to define the location of the error handling routine.
Err	Use Err to retrieve the number for the current error status. Use this in the error handling routine to determine which error has occurred.
Error	Generate a user defined error which can be caught by an error handler.
Era	Use Era to retrieve the axis number for which the error occurred. This is normally used in the error handling routine.
Erl	Use Erl to retrieve the line number in which the error occurred. This is normally used in the error handling routine.
Ert	Use Ert to retrieve the task number in which the error occurred. This is normally used in the error handling routine.
ErrMsg$	Use ErrMsg$ to retrieve the error message associated with a specified error number.
Errb	Use Errb to retrieve the robot number in which the error occurred. This is normally used in the error handling routine.

User Errors

You can define your own error messages by using the User Error Editor which is available from the Tools Menu. See details below.

[User Error Editor] (Tools Menu)

Example:

The following example shows a simple error handling routine. When an error occurs, program execution goes to the ErrHandler label, where the error handler starts. The error number is displayed and the operator is asked to continue or not. If the operator enters "N" then the program executes the Quit All statement to end the program.

```
Function Main
  String cont$
  Integer i
  OnErr Goto Errhandler
  For i = 1 To 10
    Jump P(i)
  Next i
  Exit Function

  ' *** Error handler ***
  Errhandler:
  enum = Err
  Print "Error #", enum, " occurred"
  Print "Continue (Y or N)?"
  Line Input cont$
  Select cont$
    Case "y", "Y"
      EResume Next
    Default
      Quit All
  Send
Fend
```

7.14 Multi-tasking

For some applications, you may want to control other equipment besides the robot, such as conveyors, pick and place units, etc. By using multi-tasking, you can control this other equipment with their own tasks.

SPEL+ supports up to 32 normal tasks and 16 background tasks (48 tasks in total) running simultaneously. A task is a function that has been started by the system or by the Xqt statement.

Use the Xqt statement to start another task from within a function. You can optionally specify a task number from 1 to 32 in the Xqt statement.

A task started from a background task is started as a background task. You can execute up to 16 background tasks simultaneously.

The table below shows the program instructions that are used for multitasking.

Statement	Description
Xqt	Starts a function as a task.
Halt	Temporarily suspends execution of a task.
Resume	Resumes a task that has been halted.
Quit	Stops a task.
Signal	Sends a signal to one or more tasks that are waiting for the signal using WaitSig.
SyncLock	Locks a resource for use by the current task and blocks other tasks from using the resource until SyncUnlock is executed.
WaitSig	Waits for a signal from another task.
Pause	Pause all tasks.

One example for starting another task is to run a conveyor system for the robot work cell.

Program: MAINTASK.PRG

```
Function Main
  Xqt Conveyor    ' Start the conveyor task
  Do
    ...
    ...
  Loop
Fend
```

Program: CONVTASK.PRG

```
Function Conveyor
  Do
    Select True
      Case Sw(10) = On
        Off convCtrl
      Case Sw(11) = On
        On convCtrl
    Send
  Loop
Fend
```

7.15 Using Multiple Robots

You can control more than one robot in the same project. Use the Robot statement to switch the current robot for the current task. For most applications, you should use a separate task for each robot.

Each robot has its own set of point files. You can configure which point files to use in the Project Editor. The default point file you configure for each robot is automatically loaded into memory when the main task is started.

The following program is an example where two robots run simultaneously, each with its own task.

```
Function main
  Xqt RB1
  Xqt RB2
Fend

Function RB1
  Robot 1
  Speed 50
  Do
    Jump pick
    On gripper1
    Wait .1
    Jump place
    Off gripper1
    Wait .1
  Loop
Fend

Function RB2
  Robot 2
  Speed 50
  Do
    Jump pick
    On gripper2
    Wait .1
    Jump place
    Off gripper2
    Wait .1
```

```
    Loop
Fend
```

7.16 Coordinate Systems

7.16.1 Overview

This section describes the coordinate systems for different types of robots supported in SPEL+.

Right-handed rule is used for all coordinate systems. The following coordinate systems are used in SPEL+:

- Robot Coordinate System: This is the native coordinate system of the robot. This is also known as the default base coordinate system or world coordinate system.
- Local Coordinate System: This is a user defined coordinate system located somewhere within the working envelop. (Local)
- Tool Coordinate System: This is the coordinate system of the tool mounted on the robot end-effector. This is also known as the end effector coordinate system.

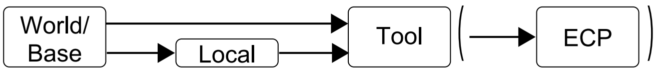
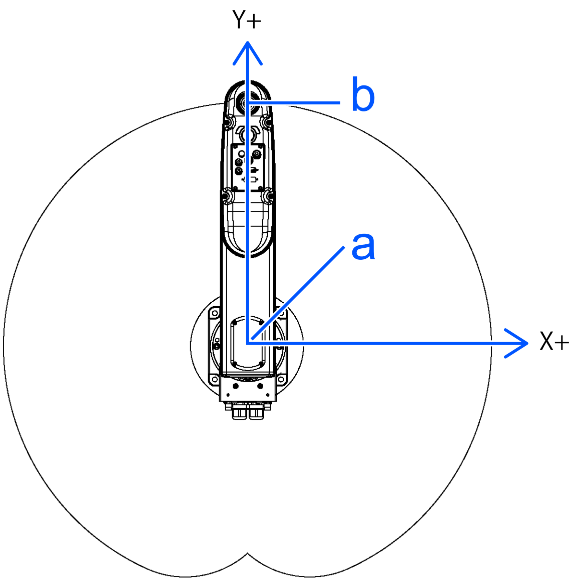


Figure: Transform order of the position/orientation from origin to tool.

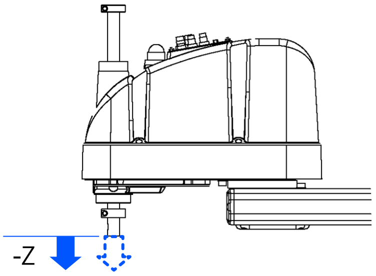
7.16.2 Robot Coordinate Systems

Robot Coordinate System of SCARA Robot

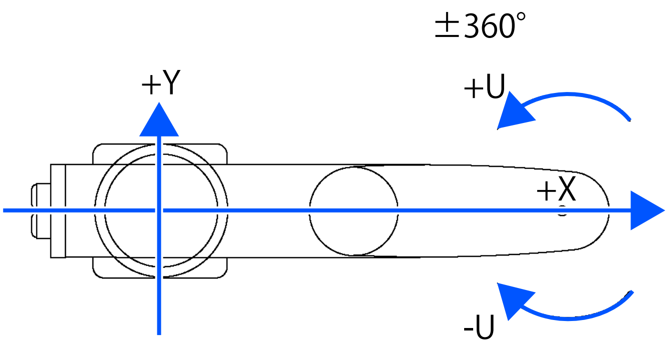


Symbol	Description
a	Origin point
b	Center of Joint #3

- Robot coordinate system Z axis

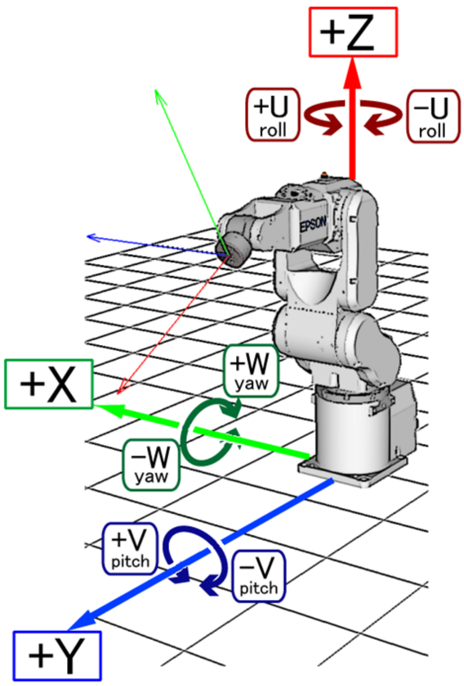


- Robot coordinate system U axis

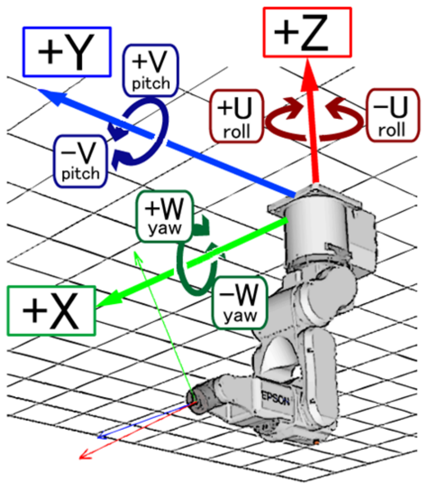


Robot Coordinate Systems of 6-Axis Robot

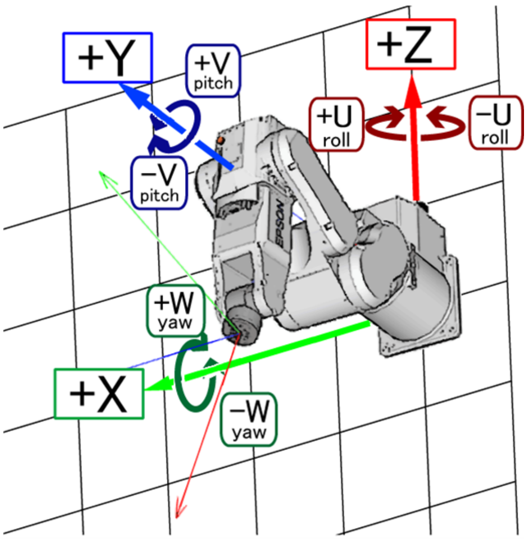
- Table Top Mounting



- Ceiling Mounting



- Wall Mounting



In the robot coordinate system, +Z axis is defined in the opposite direction of gravity. X and Y axes are defined in horizontal plane as shown in the figures above.

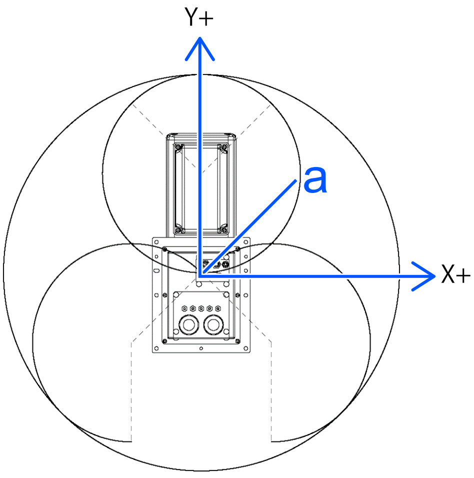
The position and orientation are designated by the position data (X, Y, Z) and the orientation data (U, V, W).

Roll-Pitch-Yaw angles are used for the orientation data.

U corresponds to roll (Z-axis rotation), V corresponds to pitch (Y-axis rotation), and W corresponds to yaw (X-axis rotation).

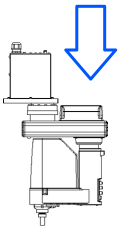
The orientation is designated by rotating the coordinate axis of U, V, and W, in that order (movable axis expression).

Robot Coordinate System of Ceiling Mounting SCARA Robot (RS series)

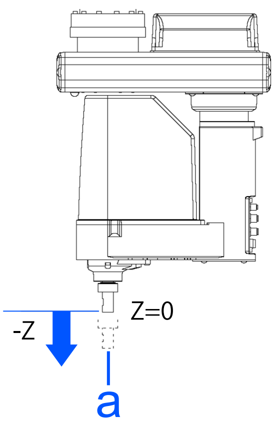


a: Origin point

* Illustration is from this direction.

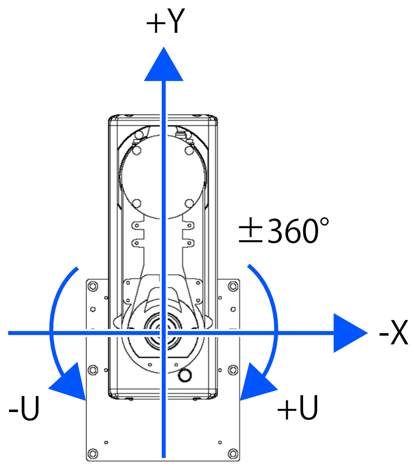


- Robot coordinate system Z axis

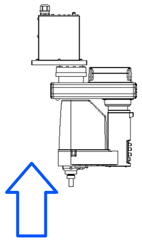


a: Center of Joint #3

- Robot coordinate system U axis

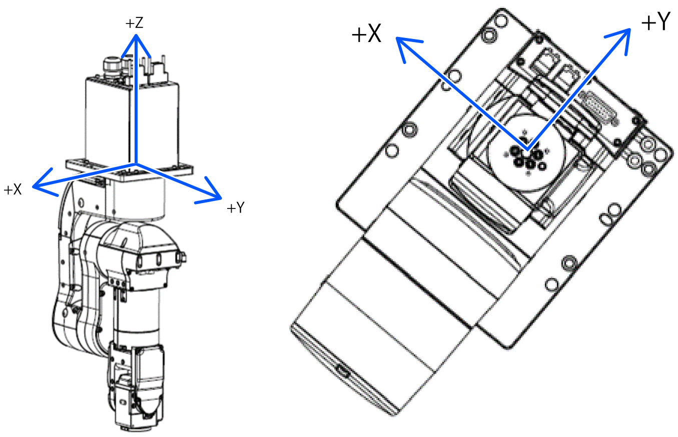


* Illustration is from this direction.

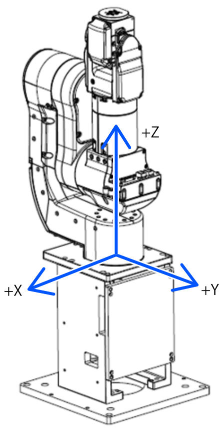


Robot Coordinate Systems of N Series Robot

- Ceiling Mounting

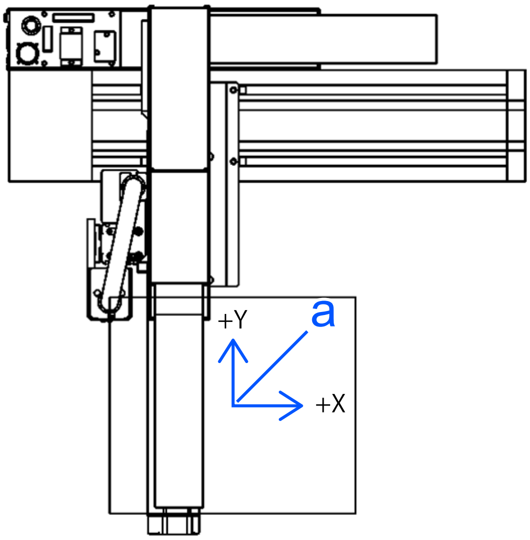


- Table Top Mounting



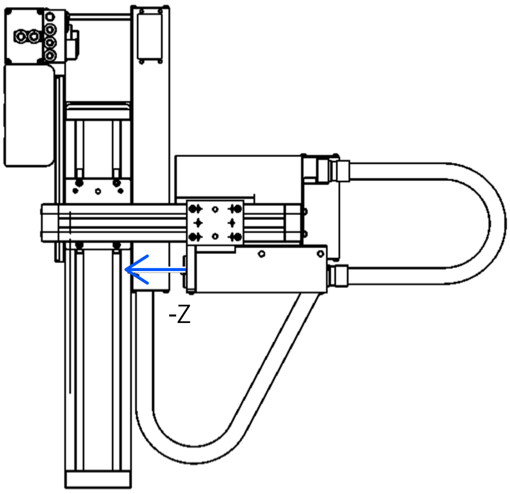
Robot Coordinate Systems for Cartesian Robot

- X axis / Y axis in Robot Coordinate System of Cartesian Robot

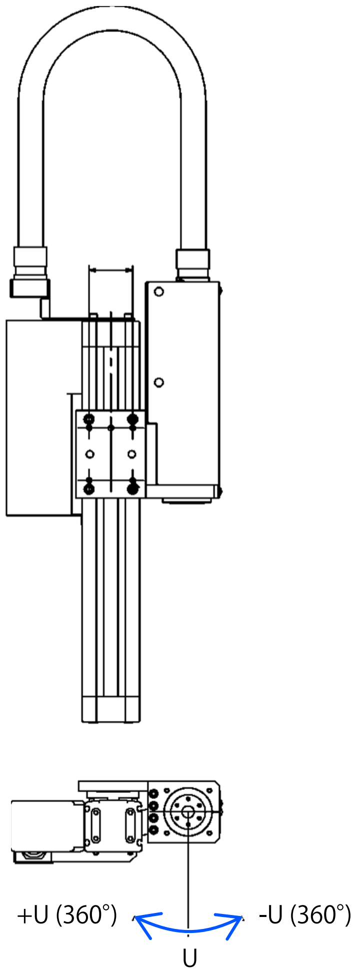


a: Origin point

- Z axis in Robot Coordinate System of Cartesian Robot



- U axis in Robot Coordinate System of Cartesian Robot



7.16.3 Local Coordinate Systems

This is a user defined coordinate system.

With SPEL+, up to 15 relative positional relationships from the robot coordinate system can be defined as local coordinate systems.

Point data is assigned with a local number from 1 to 15 as the local coordinate system, and the numbers can be used for point data attributes.

For example, program change can be minimized by using the local coordinate system, even when the robot orientation and position are changed.

To define a local coordinate system, use the Local statement or Robot Manager of the Epson RC+.

The local coordinate system "0" matches the robot coordinate system (Base). Therefore, when "0" is used for the local number in the point editor or simulator, it is same as specifying the robot coordinate system.

7.16.4 Tool Coordinate Systems

This is the coordinate system of the tool mounted on the Joint #6 flange. It is also known as the end effector coordinate system.

Point data is defined by the position and orientation of the tool coordinate system with respect to a robot coordinate system or a local coordinate system. The position is specified by the position data (X, Y, Z) and the orientation is specified by the orientation data (U, V, W) that correspond with roll, pitch, and yaw.

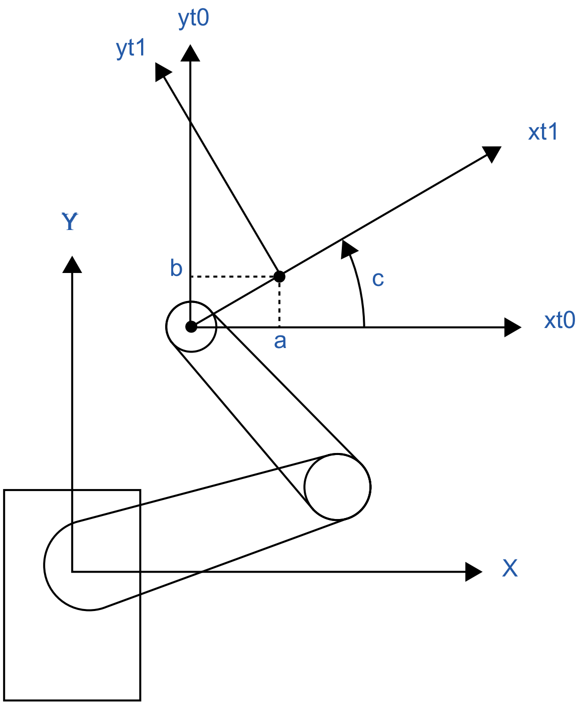
You can also define and use your own tool coordinate systems. To define the tool coordinate systems, use Tlset or Robot Manager of the Epson RC+.

The default TOOL 0 coordinate systems are defined as follows according to the robot type.

SCARA Tool 0 coordinate system

The origin of tool 0 for SCARA robots is the center of the fourth joint (rotation joint). When the fourth joint is adjusted to the position of 0 degrees, the tool 0 coordinate system axes are parallel to the robot coordinate system axes. (See figure below.)

The tool 0 coordinate system rotates as the fourth joint rotates since the origin of the tool 0 coordinate system is fixed at the fourth joint (rotation joint).



Symbol	Description
X, Y	Robot Coordinate Systems
xt0, yt0	Tool 0 Coordinate Systems
xt1, yt1	Tool 1 Coordinate Systems

6-axis Tool 0 coordinate system

- Table and ceiling mounting robots:

  The origin of tool 0 is the center of the flange on the sixth joint. When all joint angles are 0 degree, the vertically upward direction is the tool X axis, the tool Y is the same direction of X axis in the base coordinate system, and the tool Z axis is perpendicular to the sixth joint flange. (See the figure below).

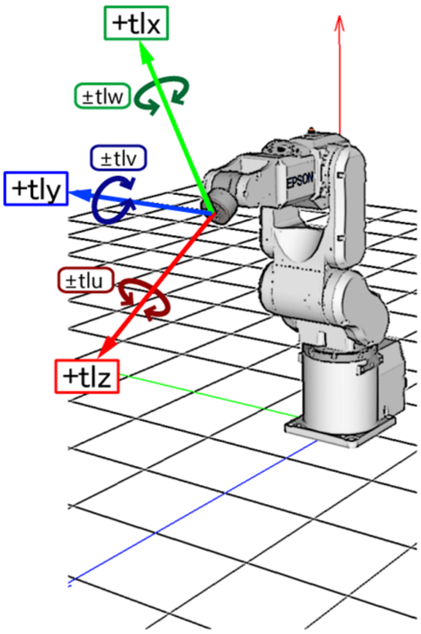
  The tool 0 coordinate system moves as the 6-axis robot changes its orientation since the origin of the tool 0 coordinate system is fixed at the sixth joint.

- Wall mounting robots:

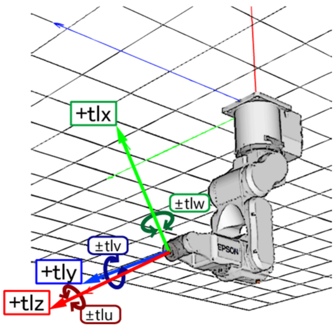
The tool 0 coordinate system is defined as below.

(tl: abbreviation of Tool)

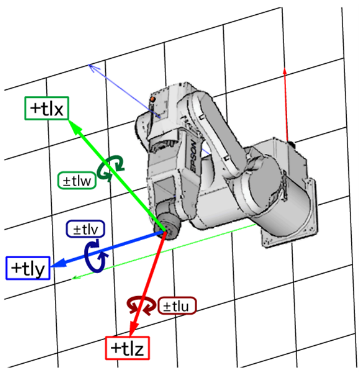
- Table Top Mounting



- Ceiling Mounting



- Wall Mounting



N series Tool 0 coordinate system

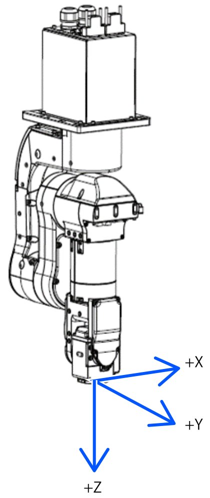
- Ceiling mounting robots:

When all joint angles are 0 degree, the tool 0 coordinate system has the X axis in –X axis direction, Y axis in the Y axis direction, and the Z axis in the –Z axis direction on the robot coordinate system. (See the figure below)

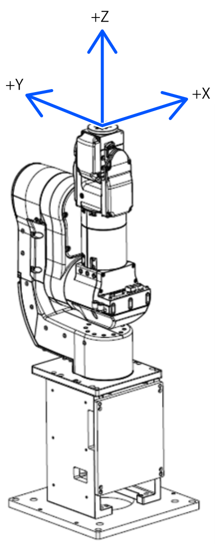
- Table top mounting:

  When all joint angles are 0 degree, the tool 0 coordinate system has the X axis in –X axis direction, Y axis in the Y axis direction, and the Z axis in the Z axis direction on the robot coordinate system. (See the figure below)

- Ceiling Mounting



- Table Top Mounting



7.16.5 ECP Coordinate Systems (Option)

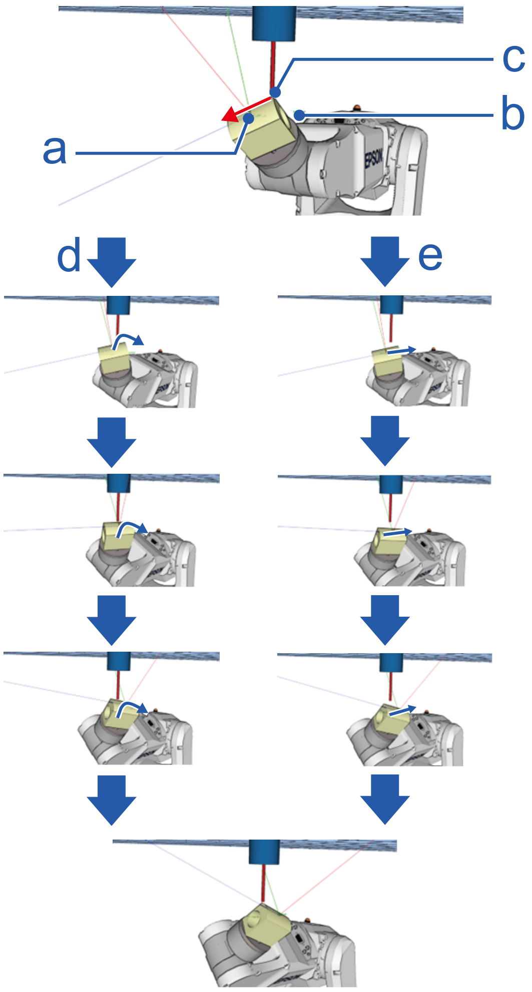
Specify a coordinate system whose origin point is on the tip of the outside fixed tool (hereafter referred to as the external control point or ECP) to move the robot arm holding a part in the trajectory made on the external control point along with the part's edges.

The following figures give a concrete example.

An ordinal Move statement controls the moving speed and orientation change of the tool center point (TCP). In the case of Move statement with the ECP argument, the part's edge is controlled to take a straight and constant-speed trajectory instead of TCP.

In the following example of no ECP, TCP takes a straight trajectory but the part's edge is distant from ECP.

If there is no orientation change, the trajectory is the same as normal operation of Move command.



Symbol	Description
a	Start teaching point
b	End teaching point
c	ECP control...
Red Arrow	When you want to move at constant speed/constant orientation change in this direction
d	ECP coordinate system: Path along external control points
e	Robot Coordinate Systems
Blue Arrow	TCP path

The following commands are available for optional ECP:

- Move command
- Arc3 command
- Curve and CVMove commands
- ECP jog motion in Robot Manager

Use the ECPSet statement for defining an ECP coordinate system. A maximum of 15 ECP coordinate systems can be defined. See details below.

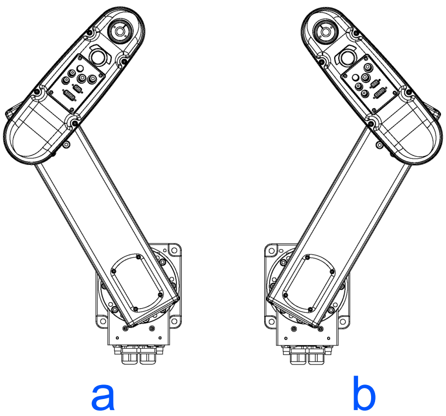
ECP Motion

7.17 Robot Arm Orientations

When developing a robot program, it is necessary to specify the point data taught for a particular arm orientation. If you fail to do so, the position can deviate slightly depending on the arm orientation, which in turn can cause the arm to follow an unexpected path, resulting in interference with peripheral equipment. This can be dangerous! To prevent this from happening, the orientation that the arm will be in when moved to the given point should be specified ahead of time in the point data. Such information can also be changed from the program.

7.17.1 SCARA robot arm orientations

With two types of arm orientation, a SCARA robot can move to nearly any position and orientation within a given work envelope. Examples are shown in the figures on the next page.

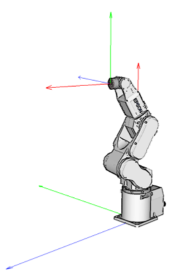
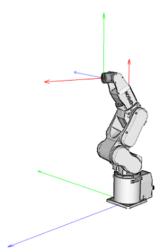


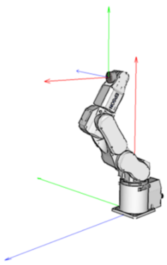
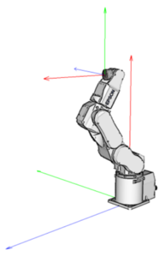
Examples of moving to the same point using Lefty and Righty arm orientations

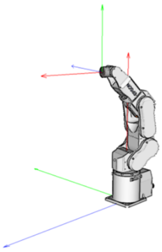
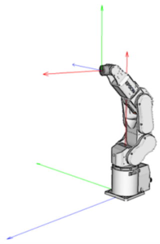
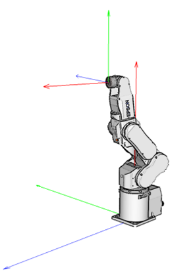
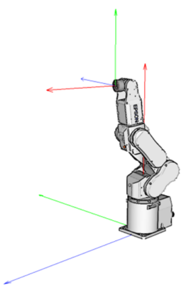
- a: Lefty arm orientation
- b: Righty arm orientation

7.17.2 6-axis robot arm orientations

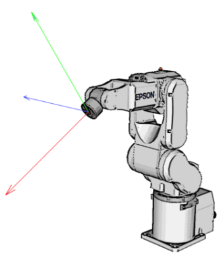
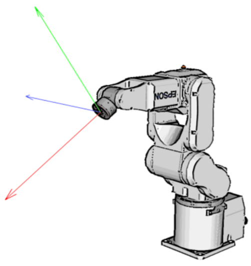
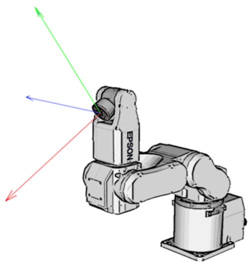
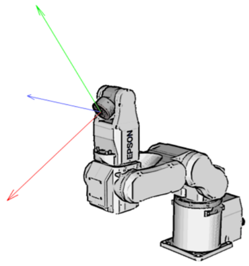
The 6-axis robot can be operated in different arm orientations within a given work envelope as shown below:

	Righty hand orientation (Arm #1)	
	NoFlip wrist orientation	Flip wrist orientation
Above elbow orientation		

	Righty hand orientation (Arm #1)	
	NoFlip wrist orientation	Flip wrist orientation
Below elbow orientation		

	Lefty hand orientation (Arm #1)	
Above elbow orientation		
Below elbow orientation		

Enlarged examples of right hand orientation are shown below.

	NoFlip wrist orientation	Flip wrist orientation
Above elbow orientation		
Below elbow orientation		

KEY POINTS

The simulator function allows you to check the robot's movements on your computer. See details below.

Jog Operation on the Robot Operation Panel

To specify the arm orientation of the 6-axis robot, add a forward slash (/) followed by:

- L (for Lefty hand orientation) or R (Righty hand orientation)
- A (Above elbow orientation) or B (Below elbow orientation)
- NF (NoFlip wrist orientation) or F (Flip wrist orientation)

There are eight available orientations as shown below, however, some combinations are not available depending on the point.

Available Orientation

- 1: /R /A /NF
- 2: /L /A /NF
- 3: /R /B /NF
- 4: /L /B /NF
- 5: /R /A /F
- 6: /L /A /F
- 7: /R /B /F
- 8: /L /B /F

At some points in the work envelope, the 6-axis robot can have the same position and orientation even if the fourth joint or the sixth joint is rotated 360 degrees. To distinguish these points, the J4Flag and J6Flag point attributes are provided.

To specify the J4Flag, add a forward slash (/) followed by:

- J4F0 (-180 < fourth joint angle <= 180), or
- J4F1 (fourth joint angle <= -180 or 180 < fourth joint angle)

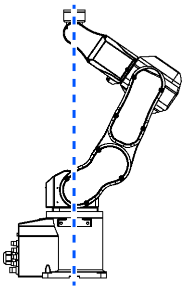
To specify the J6Flag, add a forward slash (/) followed by:

- J6F0 (-180 < sixth joint angle <= 180), or
- J6F1 (-360 < sixth joint angle <= -180 or 180 < sixth joint angle <= 360), or
- J6Fn (-180*(n+1) < sixth joint angle <= -180*n or 180*n < sixth joint angle <= 180*(n+1))

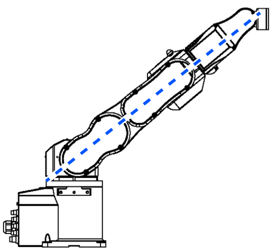
Singularity

The orientation in the boundary where the arm orientation switches to the other.

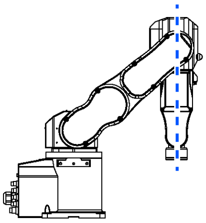
- Hand singularity: The boundary where Righty hand orientation and Lefty hand orientation switch



- Elbow singularity: The boundary where Above elbow orientation and Below elbow orientation switch



- Wrist singularity: The boundary where NoFlip wrist orientation and Flip wrist orientation switch



For the 6-axis robot, Hand / Wrist singularities exist also inside the motion range. When jogging near the singularity, follow the directions below.

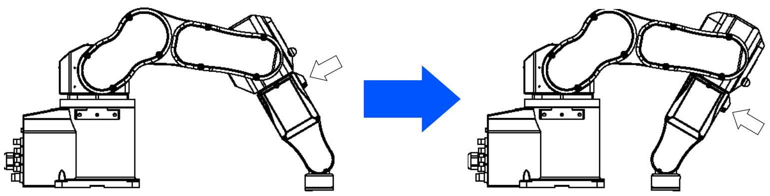
PTP motion near the singularity

When jogging a robot from point P1 near the singularity to a point calculated by point operations such as P1+X(10), the robot may move to unintended direction because the arm orientation is not properly specified.

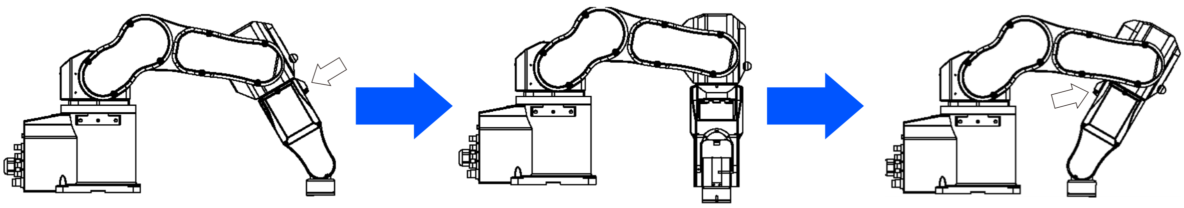
For example, when jogging from a point where the wrist is NoFlip to another point calculated by point operations, if the wrist keeps the NoFlip orientation while jogging, Joints #4 and #6 may rotate widely (by approx. 180 °). In this case, switch to the Flip wrist orientation to jog smoothly through the wrist singularity.

This phenomenon occurs not only with the point operations but also when creating points automatically with Pallet command or the result values that run from vision sequence.

- Proper motion



- Unintended motion (Joints #4 and #6 rotate 180 °)



However in the cases, it is difficult for users to specify the proper arm orientations by a program. For this LJM function is a useful command. LJM function switches the arm orientations to enable the least motion of the joints. For LJM function details, refer to the following manual:

"SPEL+ Language Reference"

Also, AutoLJM command can automatically apply LJM function to the motion commands which are included in a particular section of the program without using LJM function.

For details on the AutoLJM command, refer to the following manual:

"SPEL+ Language Reference"

In addition, you can set AutoLJM function to be enabled at the controller start up by setting preferences of the controller. However, if Auto LJM is enabled in preferences, this function automatically adjusts the posture of the manipulator to reduce the motion distance, even when you intend to move the joint widely. Therefore, it is recommended to build a program using AutoLJM command or LJM function to operate the manipulator as you desired.

If you specify all points by teaching, the arm orientations are also recorded. Therefore, the manipulator moves to the taught position without using LJM function or AutoLJM. Instead, the manipulator may move differently from the taught position by the use of LJM and AutoLJM.

LJM function for CP motion command

LJM function and AutoLJM command described above are also available for CP motion commands. However, since CP motion commands give priority to operate based on specified trajectories, the manipulator sometimes reach to the point Flag with a different posture from the specified one. At this time, if CP motion command is used with CP On, an error from 4274 to 4278 will occur according to the mismatched point flag. To avoid the error, operate the manipulator with CP Off, or match the point flag of a target point and the one after motion completion. If operated with CP Off, the error does not occur and the manipulator can continue operation from the point where the mismatch happened.

Also, you can set the controller's preference so that the mismatches of flags are not considered as an error at the controller startup. However, path motions which use CP On will be disabled.

CP motion near the singularity (singularity avoiding function in CP motion)

When executing Move or CP motion near the singularity, the joint speed may increase rapidly. The acceleration error will occur and the joints will move widely and interfere with peripherals. In particular, the position of Joint #1 near the hand singularity and Joints #2 - #6 near the wrist singularity change greatly.

Epson RC+ 8.0 has a singularity avoiding function to prevent acceleration errors during the execution of CP motion commands that pass the wrist singularity described above. With this function, the manipulator takes a detour to avoid an acceleration error by passing a different trajectory and returns to the original trajectory after passing the singularity.

Since it takes a different trajectory from the original one, it may reach an orientation that does not match the orientation specified for the target point. At this time, if CP motion command is used with CP On, an error from 4274 to 4278 will occur according to the mismatched point flag. To avoid the error, operate the manipulator with CP Off, or match the point flag of a target point and the one after motion completion. If operated with CP Off, the error does not occur and the manipulator can continue operation from the point where the mismatch happened.

For details on the singularity avoiding function, refer to the following manual:

"SPEL+ Language Reference - AvoidSingularity Statement"

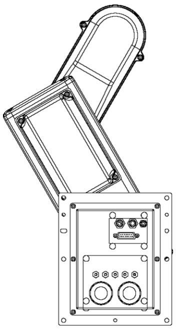
Singularity avoiding function is enabled as default. If you want to avoid the error by reducing the motion speed in order to maintain the trajectory accuracy, you can enable the variable-speed CP motion function by setting the AvoidSingularity command to "3". With the variable-speed CP motion function, vertical 6-axis robots (including N-series) and RS-series robots that are approaching a singularity while performing CP motion can automatically suppress speed while maintaining trajectory to avoid acceleration and overspeed errors, and then return to normal speed instructions after leaving the singularity. Due to maintaining the trajectory and passing near the singularity, the first, second, fourth, and sixth joints may move significantly. If the AvoidSingularity command is set to "SING_VSD", the arm orientation does not change from before the motion.

If you cannot avoid errors even if you use the singularity avoiding function, use PTP motion to enable the least motion of the joints or arrange the manipulator installation position and hand offset volume to prevent the CP motion near the singularity.

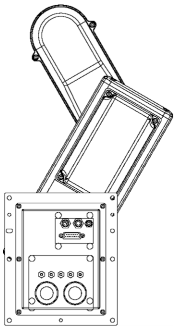
7.17.3 RS series arm orientations

The RS series can be operated in various arm orientations within a given work envelope as shown below:

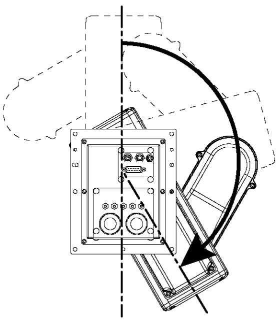
- Lefty arm orientation



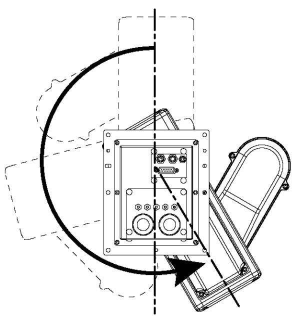
- Righty arm orientation



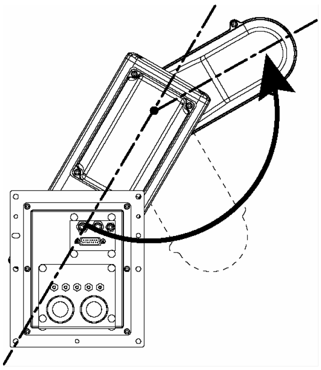
- J1 F0 arm orientation



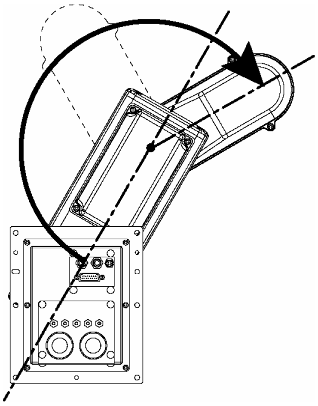
- J1 F1 arm orientation



- J2 F0 arm orientation



- J2 F1 arm orientation



To specify the arm orientation of the RS series, add a forward slash (/) followed by:

- L (for Lefty hand orientation) or R (Righty hand orientation)
- J1F0 or J1F1
- J2F0 or J2F1.

For the RS series robots, some points in the work envelope can have the same position and orientation even if J1 or J2 is rotated 360 degrees. To distinguish these points, the J1Flag and J2Flag point attributes are provided.

To specify the J1Flag, add a forward slash (/) followed by:

- J1F0 (-90 < first joint angle <= 270), or
- J1F1 (-270 < the first joint angle <= -90 or 270 < the first joint angle <= 450)

To specify the J2Flag, add a forward slash (/) followed by:

- J2F0 (-180 < second joint angle <= 180), or
- J2F1 (-360 < second joint angle <= -180 or 180 < second joint angle <= 360), or

There are eight available orientations as shown below, however, some combinations are not available depending on the point.

Available Orientation

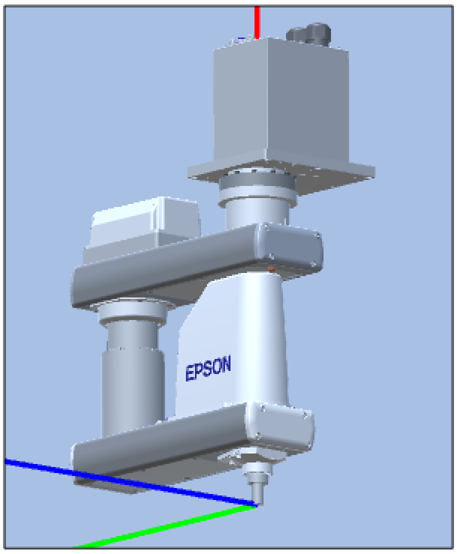
- 1: /R /J1F0 /J2F0
- 2: /L /J1F0 /J2F0
- 3: /R /J1F1 /J2F0
- 4: /L /J1F1 /J2F0
- 5: /R /J1F0 /J2F1
- 6: /L /J1F0 /J2F1
- 7: /R /J1F1 /J2F1

- 8: /L /J1F1 /J2F1

Singularity

The orientation in the boundary where the arm orientation switches to the other.

- Hand singularity: The boundary where Righty hand orientation and Lefty hand orientation switch (X=0, Y=0)



When jogging near the singularity, follow the directions below.

PTP motion near the singularity

When jogging a robot from point P1 near the singularity to a point calculated by point operations such as P1+X(10), the robot may move to unintended direction because the arm orientation is not properly specified.

For example, when jogging from a point where the hand is Righty to another point calculated by point operations, if the hand keeps the Righty orientation while jogging, Joints #1 may rotate widely (by approx. 180 °). In this case, switch to the Lefty hand orientation to jog smoothly through the wrist singularity.

This phenomenon occurs not only with the point operations but also when creating points automatically with Pallet command or the result values that run from vision sequence.

However in the cases, it is difficult for users to specify the proper arm orientations by a program. For this LJM function is a useful command. LJM function switches the arm orientations to enable the least motion of the joints. For LJM function details, refer to the following manual:

"SPEL+ Language Reference"

Also, AutoLJM command can automatically apply LJM function to the motion commands which are included in a particular section of the program without using LJM function.

For details on the AutoLJM command, refer to the following manual:

"SPEL+ Language Reference"

In addition, you can set AutoLJM function to be enabled at the controller start up by setting preferences of the controller. However, if Auto LJM is enabled in preferences, this function automatically adjusts the posture of the manipulator to reduce the motion distance, even when you intend to move the joint widely. Therefore, it is recommended to build a program using AutoLJM command or LJM function to operate the manipulator as you desired.

If you specify all points by teaching, the arm orientations are also recorded. Therefore, the manipulator moves to the taught position without using LJM function or AutoLJM. Instead, the manipulator may move differently from the taught position by the use of LJM and AutoLJM.

CP motion near the singularity (singularity avoiding function in CP motion)

When executing Move or CP motion near the singularity, the joint speed may increase rapidly. The acceleration error will occur and the joints will move widely and interfere with peripherals. In particular, the position of Joint #1 near the hand singularity changes greatly.

Epson RC+ 8.0 has a singularity avoiding function to prevent acceleration errors during the execution of CP motion commands that pass the hand singularity described above. With this function, the manipulator takes a detour to avoid an acceleration error by passing a different trajectory and returns to the original trajectory after passing the singularity.

Since it takes a different trajectory from the original one, it may reach an orientation that does not match the orientation specified for the target point. At this time, if CP motion command is used with CP On, an error from 4274 to 7278 will occur according to the mismatched point flag. To avoid the error, operate the manipulator with CP Off, or match the point flag of a target point and the one after motion completion. If operated with CP Off, the error does not occur and the manipulator can continue operation from the point where the mismatch happened.

For details on the singularity avoiding function, refer to the following manual:

"SPEL+ Language Reference - AvoidSingularity Statement"

Singularity avoiding function is enabled as default. If you want to avoid the error by reducing the motion speed in order to maintain the trajectory accuracy, you can enable the variable-speed CP motion function by setting the AvoidSingularity command to "SING_VSD". With the variable-speed CP motion function, vertical 6-axis robots (including N-series) and RS-series robots that are approaching a singularity while performing CP motion can automatically suppress speed while maintaining trajectory to avoid acceleration and overspeed errors, and then return to normal speed instructions after leaving the singularity. Due to maintaining the trajectory and passing near the singularity, the first, second, fourth, and sixth joints may move significantly. If the AvoidSingularity command is set to "3", the arm orientation does not change from before the motion.

If you cannot avoid errors even if you use the singularity avoiding function, use PTP motion to enable the least motion of the joints or arrange the manipulator installation position and hand offset volume to prevent the CP motion near the singularity.

7.17.4 N series arm orientations

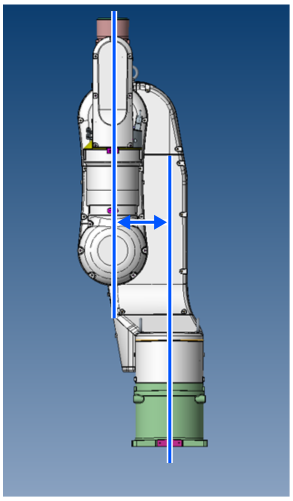
The N series can be operated in various arm orientations within a given work envelope as shown below:

Orientation of N series is different depending on "with" and "without" offset.

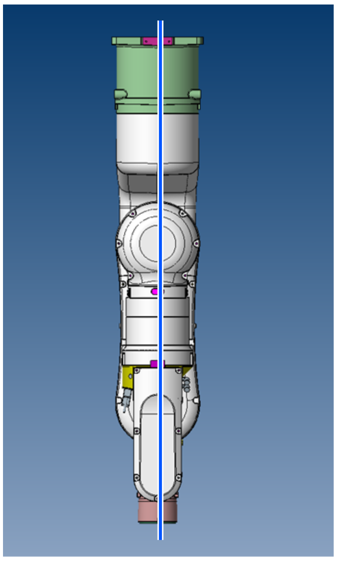
Offset is a distance between Joint #2 and Joint #1.

Examples of orientation "with" and "without" offset are shown below.

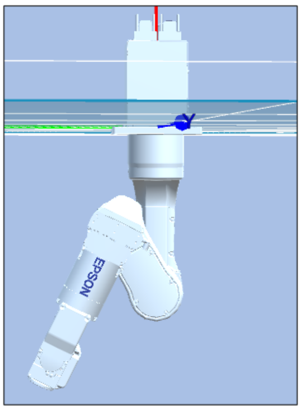
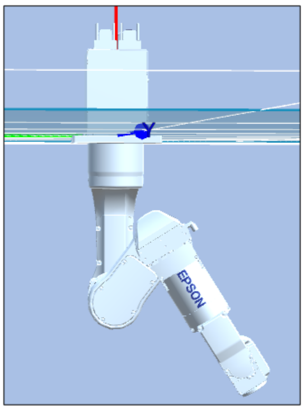
- With offset: Distance from Joint #2 and Joint #1 is not 0mm

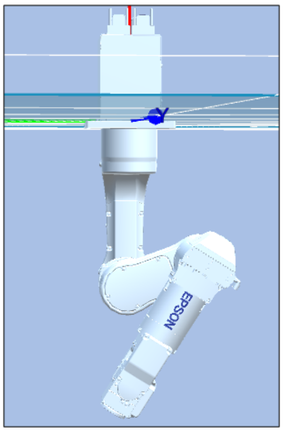
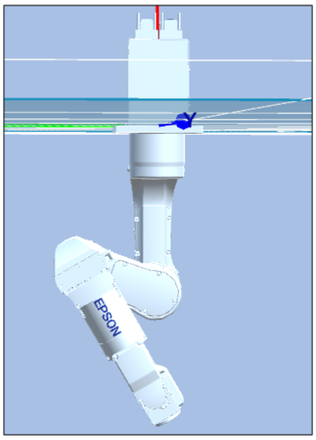


- Without offset: Distance from Joint #2 and Joint #1 is 0mm

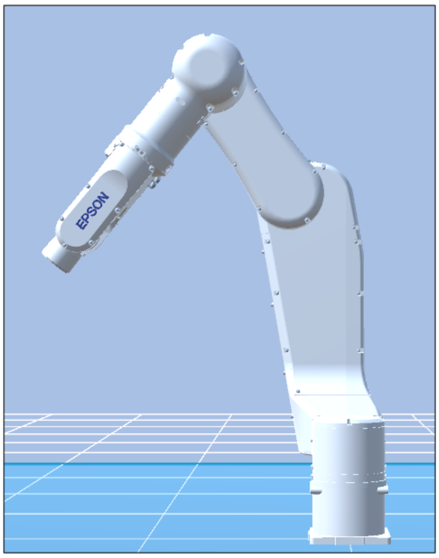
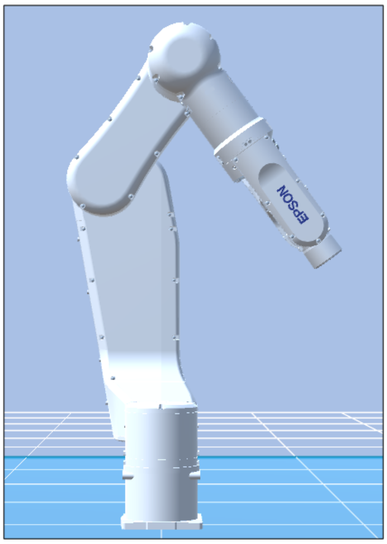
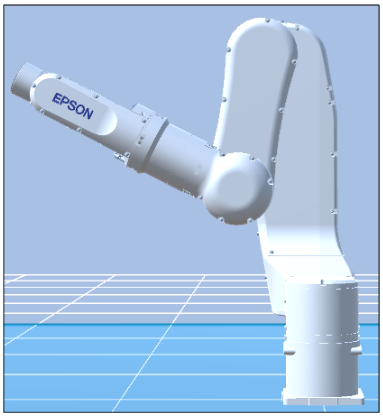
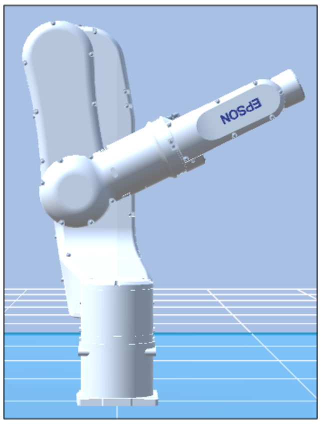


Without offset (Illustrations: N2-A450SR)

	Arm orientation	
	Righty arm orientation	Lefty arm orientation
Above elbow orientation		

	Arm orientation	
	Righty arm orientation	Lefty arm orientation
Below elbow orientation		

With offset (Illustration: N6-A1000S)

	Arm orientation	
	Righty arm orientation	Lefty arm orientation
Above elbow orientation		
Below elbow orientation		

To specify the arm orientation of the N series, add a forward slash (/) followed by:

- L (for Lefty hand orientation) or R (Righty hand orientation)
- A (Above elbow orientation) or B (Below elbow orientation)

- NF (NoFlip wrist orientation) or F (Flip wrist orientation)

There are eight available orientations as shown below, however, some combinations are not available depending on the point.

Available Orientation

- 1: /R /A /NF
- 2: /L /A /NF
- 3: /R /B /NF
- 4: /L /B /NF
- 5: /R /A /F
- 6: /L /A /F
- 7: /R /B /F
- 8: /L /B /F

At some points in the work envelope, the robot can have the same position and orientation even if the fourth joint or the sixth joint is rotated 360 degrees. To distinguish these points, the J4Flag and J6Flag point attributes are provided.

To specify the J4Flag, add a forward slash (/) followed by:

- J4F0 (-180 < fourth joint angle <= 180), or
- J4F1 (fourth joint angle <= -180 or 180 < fourth joint angle)

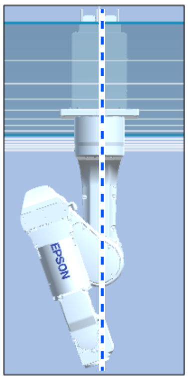
To specify the J6Flag, add a forward slash (/) followed by:

- J6F0 (-180 < sixth joint angle <= 180), or
- J6F1 (-360 < sixth joint angle <= -180 or 180 < sixth joint angle <= 360), or
- J6Fn (-180*(n+1) < sixth joint angle <= -180*n or 180*n < sixth joint angle <= 180*(n+1))

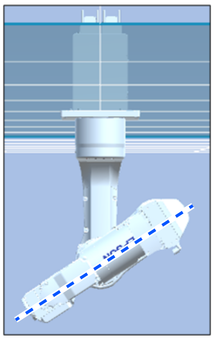
Singularity

The orientation in the boundary where the arm orientation switches to the other.

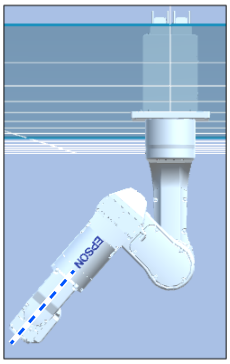
- Hand singularity: The boundary where Righty hand orientation and Lefty hand orientation switch



- Elbow singularity: The boundary where Above elbow orientation and Below elbow orientation switch



- Wrist singularity: The boundary where NoFlip wrist orientation and Flip wrist orientation switch



For the N series robot, Hand / Wrist singularities exist also inside the motion range like the 6-axis robot. When jogging near the singularity, pay attention to the same points as the 6-axis robot.

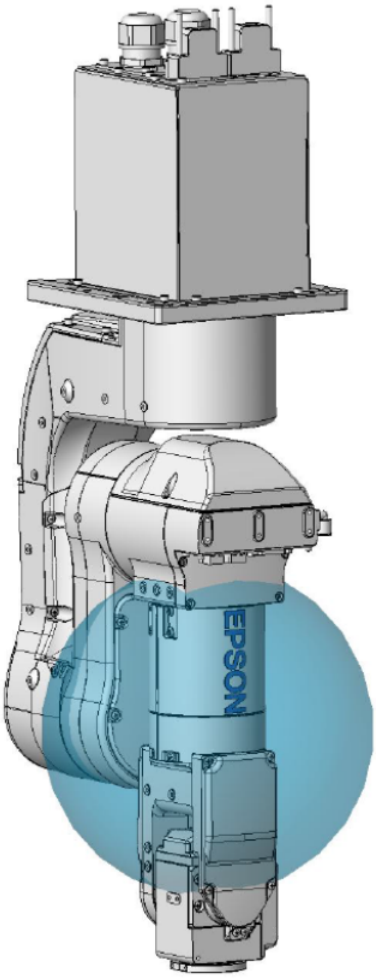
For common considerations, see below.

6-axis robot arm orientations

The following describes the elbow singularity area that is unique for 6-axis robots.

Elbow singularity area

For the N series robot, the singularity exists where the P point is on the sphere shown in the figure below. The P point cannot be inside the sphere. Therefore, CP motion to pass inside the sphere is not available.



Elbow singularity area avoidance motion

When the robot passes through the sphere as shown in the figure below, the robot behaves differently depending on the singularity avoiding function (AvoidSingularity) mode. The operation is as follows.

Mode: SING_AVOID

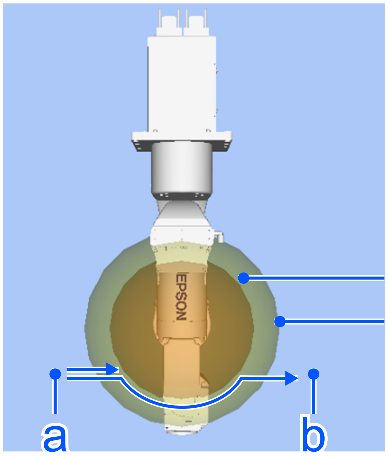
The robot moves to the end point while avoiding the elbow singularity area as indicated with a red line (P-point trajectory) in the figure below. Also, an error occurs in the following cases:

- If SpeedS setting value is too large, error 4242, 4243, 4255, or 5044 occurs. The errors can be prevented by setting SpeedS lower.
- If the motion is stopped/paused, or the safety door is opened during the singularity avoiding motion (PTP motion), error 4242, 4250, 4252, or 4256 occurs. Do not stop the operation or open the safety door during the singularity avoiding motion.
- If the singularity avoiding motion mode (SING_AVOID) is selected for N series, an error 4255 or 4256 occurs.

Mode: Other than SING_AVOID

The error 4252 occurs when the robot touches the elbow singularity area as indicated with a blue line (P-point trajectory) in the figure below.

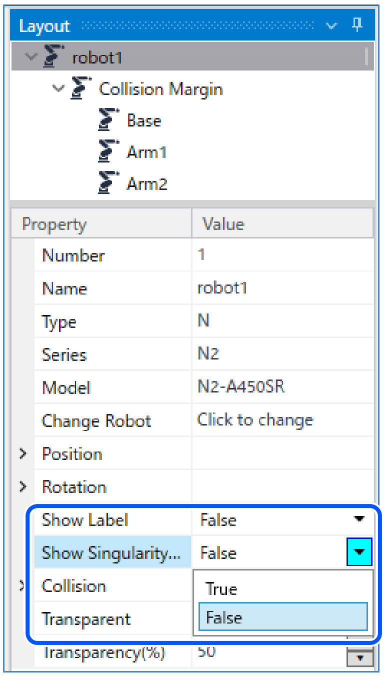
The elbow singularity area (Flange) is an area when Joint #5 is 0°.



Symbol	Description
a	Start point
b	End point
c	Elbow singularity area (Flange)
d	Elbow singularity area (P point)

Note:

- The pass motion can be confirmed by the sample simulator program “N2_sample”.
- With Jump3, Jump3CP, and JumpTLZ, the motion to pass the elbow singularity area is not available. (Shoulder and wrist singularity pass motions are possible.)
- In the singularity avoiding motion, the Joint #4 and #6 may rotate largely.
- In the singularity avoiding motion, onward and backward paths may differ.
- To display the elbow singularity area and elbow singularity neighborhood in the simulator, set the robot's Show Singularity Area property to "True". The Show Singularity Area property is only displayed for the N series.



7.18 Robot Motion Commands

SPEL+ includes several commands for controlling the robot from your programs.

7.18.1 Homing the robot

The Home command moves the robot to a user defined "park" or "idle" position. This command works for all robots. It is mainly used for absolute encoder robots that normally do not need to be mechanically homed. Use the HomeSet command to set the home position and the Hordr command to set the home order.

7.18.2 Point to point motion

Point to point (PTP) commands move the tool center point of the robot from its current position to a specified point. Motion of the tool center point may not be in a straight line.

To set the speed for point to point commands, use the Speed command. To set acceleration and deceleration, use the Accel command.

Command	Description
Go	Move directly to a point using point to point motion.
Jump	Jump to a point. First move up to the current LimZ setting, the move over the destination point, then move to the point. (From the current position, the third axis is moved up to the uppermost point (Z=0). A lateral orientation move is performed to above the specified target position, and the third axis is then lowered to the target position.) The Arch table settings determine the Jump profile.
Jump3	Jump to a point in 3 dimensions. Move in a straight line with the same orientation until the recede point. The motion between the recede points is PTP motion.
Pass	Move near one or more points.
TGo	Move directly to a point in a tool coordinate system.
BGo	Move in a PTP motion to the relative specified point in Base / Local coordinate system

7.18.3 Linear motion

Linear motion commands move the tool center point of robot from its current position to a specified point in a straight line. Liner motion is a CP (Continuous Path) motion.

To set velocity (speed) for straight motion, use the SpeedS command. To set acceleration and deceleration, use the AccelS command.

Command	Description
Move	Move in a straight line to the specified point.
TMove	Move in a straight line to the specified point in a tool coordinate system.
Jump3CP	Jump to a point in 3 dimensions using CP motion. Move in a straight line until the recede point. The motion between the recede points is also a straight line motion.
BMove	Move in a straight line to the relative specified point in Base / Local coordinate system

7.18.4 Curves

Curves commands move the robot in a circular arc. Curves is a CP (Continuous Path) motion.

To set velocity (speed) for Curves, use the SpeedS command. To set acceleration and deceleration, use the AccelS command.

Command	Description
Arc	Move the robot through one point to another point using circular interpolation.
Arc3	Move the robot in 3D using circular interpolation.
Curve	Defines the CP motion control of robots.
CVMove	Executes a path specified by Curve.

7.18.5 Joint motion

Command	Description
JTran	The JTran command can be used to move one joint of the robot a specified distance to a specified position. (Rotational joint: degrees; linear joint: mm) Speed and acceleration, as with point to point motion commands, are specified with Speed and Accel commands.
PTran	The PTran command can be used to move one joint of the robot to an encoder pulse position. (Pulse: integer units) Speed and acceleration, as with point to point motion commands, are specified with Speed and Accel commands.
Pulse	The Pulse command can be used to move all joints of the robot to encoder pulse positions. (Pulse: integer units) Speed and acceleration, as with point to point motion commands, are specified with Speed and Accel commands.
PG_Scan	The PG_Scan command can be used to rotate a pulse generator axis of a Joint-type single axis PG robot continuously in CW/CCW directions. (To rotate it continuously, you need to enable the continuous rotation parameter.) Speed and acceleration, as with point to point motion commands, are specified with Speed and Accel commands.

7.18.6 Controlling position accuracy

Use the Fine command to adjust position accuracy for the end of a motion command. Fine specifies, for each joint, the allowable positioning error for detecting completion of any given move. The lower the Fine settings, the more accurate the final position of the joint, which can cause slower performance. Conversely, large Fine settings can speed up motion commands, but position accuracy will decrease. For many applications, the default settings can be used.

7.18.7 CP Motion Speed / Acceleration and Tool Orientation

When you attempt to change only the tool orientation while keeping the tool tip of the robot arm at the specified coordinate point or when the tool orientation variation is larger than the travel distance of the tool tip, the speed of tool orientation change may greatly accelerate. To prevent this, there is a function that automatically limits the motion speed when the speed of tool orientation change is large.

If you want to manually set the upper limit of the tool orientation change speed during CP motion, turn on SpeedRLimitation. If, during CP operation, the tool orientation change speed then exceeds the set SpeedR, the motion speed is limited so that the tool orientation change speed becomes SpeedR. If the tool orientation change speed does not exceed the set SpeedR, it operates at the set SpeedS. The upper limit of the tool orientation change speed should be set in advance with SpeedR.

For example:

```
SpeedR 50              ' deg/sec
SpeedRLimitation On
Move P1
```

For more details, refer to the following manuals:

"SPEL+ Language Reference - SpeedRLimitation, SpeedR"

KEY POINTS

- The SpeedR default value is set to low, so if SpeedR is not set properly, CP movements will be slow with orientation changes. If SpeedRLimitation is enabled, set SpeedR (upper limit of tool orientation change speed) appropriately along with SpeedS.

  Also, by adding the ROT parameter to the CP motion commands, you can perform motions based on the angular velocity and angular acceleration/deceleration of the main axis for the specified orientation changes.

  For example:

  ```
  SpeedR 50         ' deg/sec
  AccelR 200, 200   ' deg/sec2
  Move P1 ROT
  ```

- The tool orientation variation is normally comprised of orientation variations of more than one rotation axis.

  The SpeedR and AccelR parameters specify the angular velocity and acceleration/deceleration of the main axis regarding the orientation variation. Therefore, actual angular velocity and acceleration/deceleration of the orientation variation are different from the parameters except for the case where the rotation axis of the orientation is only one.

  While the motion command with the ROT parameter is executed, the specified SpeedS and AccelS parameters are invalid.

  The ROT parameter can be used with the following motion commands:

  - Move
  - Arc
  - Arc3
  - BMove
  - TMove
  - Jump3CP

7.18.8 PTP Speed / Acceleration for Small Distances

You can change the speed and acceleration for small distances using PTPBoost and PTPBoostOK. Normally, PTPBoost is not required. In certain cases, you may want to shorten the cycle time even if vibration becomes larger, or conversely you may want to reduce vibration even if cycle time becomes longer. PTPBoost is a robot parameter with values from 0 – 100 that affects the speed and acceleration for small distances. Normally, for small distance motion, the desired speed cannot be attained using the current acceleration. By increasing PTPBoost, acceleration, deceleration, and speed are increased for small distance motion. To check if a motion command will be affected by PTPBoost, use the PTPBoostOK function. For details, refer to the following manual:

"SPEL+ Language Reference - PTPBoost, PTPBoostOK"

7.18.9 Pressing Motion

To use the pressing motion, use the following torque control mode commands.

- TC (Returns the torque control mode setting and current mode.)
- TCSpeed (Specifies / returns the speed limit in the torque control.)
- TCLim (Specifies the torque limit of each joint for the torque control mode.)

The low power mode is limited by a low power upper limit. Therefore, normally use the High power mode. For details and command usage, refer to the following manual:

"SPEL+ Language Reference - TC, TCSpeed, TCLim"

7.18.10 Collision Detection Function (Detection Function of Robot Motion Error)

Detect the robot motion error from differentiation between desired speed and the actual speed (speed deviation value). Errors can be detected by this function is classified into A and B.

- A: Collision or contact of robot arm or hand occurs
- B: Robot motion errors other than collision or contact

Also, error B is classified into below according to the power condition.

- Error in high power
    - B1: Torque saturation due to a low Weight or Inertia setting.
    - B2: Torque saturation due to combined motion of multiple joints and movement of a long object.
    - B3: Torque saturation due to supply voltage reduction.
    - B4: Error motion due to hardware error or software malfunction.
- Error in low power
    - B4: Error motion due to hardware error or software malfunction.
    - B5: Torque saturation in low power due to a hand or a long object that exceeds the weight described in the specifications.

When an A or B error is detected, one of the messages below will be displayed and the robot will stop. Reduce the damage of the robot or equipment.

- Error 5057: detect the collision in high power. (Detect the robot motion error.)
- Error 5058: detect the collision in low power. (Detect the robot motion error.)

The following errors have previously existed; however, this function can detect the above errors quickly.

- Error 5042, 5043: Position error.

Error is not detected by torque saturation in short time. Detect a state with high risk that causes a malfunction and stop the robot. The following phenomena may occur if you continue to use the robot in B1 or B2 status. Make a state that errors do not occur.

- Loose binding parts such as screws.

- Reduction gear is damaged.
- Increase a risk of robot damage

Turn ON CollsionDetect command and detection is enabled. (Default: ON)

Default is different depending on the firmware version.
- Ver.7.2.1.x or later: default: ON
- Before Ver.7.2.0.x : default: OFF
- When upgrading before Ver.7.2.0.x or Ver7.2.1.x or later: default: OFF

Reboot a controller to return to the default.

The following describes the detail of error B when error 5057 or 5058 is detected without a collision or contact of the robot or arm.

In high power mode

Check the torque saturation by using PTRQ command. Torque saturation is occurred if the joint outputs "1" in PTRQ command. In that case, make sure that the Weight setting is properly and in accord with the hand weight. Also, make sure that Inertia setting is properly for joint #4 of SCARA robot and joint #6 of 6-axis robot.

Next, make sure that there is no torque saturation by using PTRQ command by combined motion that multiple joints (#2, #3 and #5 joints of 6-axis robot) operate in the same direction and throwing around the long object.

If torque saturation occurs, reduce acceleration/deceleration of Accel command until there is no torque saturation (the value: 1.0 or less is displayed in PTRQ).

Also, torque saturation may occur due to reduction of supply voltage that inputs to the controller. Check the power supply voltage is within the specifications.

You can turn ON/OFF the collision detection function per equipment if you want to use without performing those error detection due to equipment compatibility securement or similar reasons.

If other error occurs at the same time, take a countermeasure for that first.

In low power mode

Make sure that hand weight is within the specifications.

Also, check the torque saturation when errors occur on the joint #4 and 5 of 6-axis robot. When torque saturation is occurred, it is long object that cannot be hold by low power mode. Hold in high power mode.

If other error occurs at the same time, take a countermeasure for that first.

Immediately stop result of the torque saturation by combination of the following motion and command. Error of A and B can be detected faster.
- HP motion: LimitTorqueStop Command
- LP motion: LimitTorqueStopLP Command

The following describes details of collision of the robot arm A and contact detection.

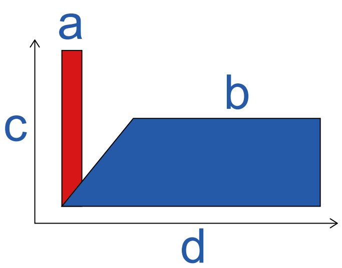
For reduction of damage on the arms and the end effectors due to the collision with peripherals, there are two functions: Collision detection function, and torque restriction function described in the next section.
- The collision detection function detects the collision and stops the robot immediately.
- The torque restriction function restricts torque at the collision and also stops the robot immediately.

These are functions to reduce damage on the robot at the collision, but cannot avoid damage completely. Also, the functions cannot be used for the purpose of human safety.

As shown following, the force applied to the robot during a collision can be roughly divided into two types:

the impact of speed right before the collision, and the pressing force by motor torque after the collision.

The collision detection function and the torque restriction function reduce damage caused by the pressing force right after the collision. These functions do not have any effect on damage caused by the impact of speed.



Symbol	Description
a	Impact force at speed
b	Pressing force with torque
c	Force
d	Time

The collision detection function detects collision by the speed deviation value for robot motion control (differentiation between the desired speed and the actual speed) showing an abnormal value which is greatly different from normal motion due to the collision.

Turn ON CollsionDetect command and the detection is enabled. (Default: ON)

Default is different depending on the firmware version.

- Ver.7.2.1.x or later: default: ON
- Before Ver.7.2.0.x : default: OFF

Reboot a controller to return to the default.

When enabled, this function reduces the time of pressing force by the motor torque, by detecting the collision and stopping the robot immediately. This reduces the pressing force by about 20%. To reduce the damage more, use this function together with the torque restriction function.

This function is automatically disabled during the pressing motion and the force sensing operation described in the following.

Pressing Motion

Also, the function may have false detection in cases of powerful contact motion and significant acceleration and deceleration which may have consecutive torque saturation. To confirm if there is a risk of false detection, use PTRQ.

If PTRQ is less than 1 for all axes, there is no risk of false detection.

If PTRQ is one, torque saturation is occurring on the axis. This means excessive acceleration and deceleration are applied, and it is not preferable for motor control. It also has a risk of damage on the manipulator. In such a situation, take the following countermeasures.

For contacting operation,

- Lower acceleration and deceleration at a contact
- Set a contacting depth shallow

If you want to operate the manipulator without taking the above countermeasures, you can enable and disable the function for each axis. Set the function off for the axis which you want to disable the function.

For details of the command and function, refer to the following manual.

"SPEL+ Language Reference - CollisionDetect, CollisionDetect Function"

7.18.11 Torque Restriction Function

As with the collision detection function, the torque restriction function can reduce damage during a collision.

Collision Detection Function (Detection Function of Robot Motion Error)

The torque restriction value used for this function is defined by adding the margin to the upper limit torque value used in the program in order to avoid malfunction. By using the torque restriction function, the pressing force can be reduced.

For example, if the torque is restricted at 30%, the pressing force can also be reduced to 30%. Also, the robot immediately stops when the torque reaches the upper limit value. By stopping the robot immediately, a further 20-30% reduction effect can be obtained.

When the torque is restricted at 30% and the robot is stopped immediately, the total of less than 25% or equivalent reduction effect can be obtained.

For SCARA robots, the end of the extended shaft may get caught and bent. To reduce occurrence of the bent shaft, it is recommended to use this function to reduce the pressing force to the maximum degree.

If malfunction occurs, take any of the following measures for the axis of malfunction.

- Set LimitTorqueStop or LimitTorqueStopLp off
- Increase the threshold value of LimitTorque or LimitTorqueLp

To use the torque restriction function for jogging motion, follow the steps below.

1. Execute PTCLR and start torque measurement.
2. Execute the jogging motion.
3. Measure the maximum torque value by PTRQ, and then add the margin to it.
4. Set LimitTorqeLP and LimitTorqeLPStop.

If the robot is temporarily stopped in the low power motion, the value larger than the normal program operation or jogging motion may be obtained. In such case, execute the temporary stop while measuring PTRQ and include it into measurement.

For details of the command and function, refer to the following manual.

"SPEL+ Language Reference"

- LimitTorque Statement, LimitTorque Function
- LimitTorqueLP Statement, LimitTorqueLP Function
- LimitTorqueStop Statement, LimitTorqueStop Function
- LimitTorqueStopLP Statement, LimitTorqueStopLP Function

The following is a sample program which automatically configures the collision detection function and the torque restriction function.

The program repeats the motion called “all_ax_move”.

The program enables the collision detection function, measures the maximum torque in the first five moves, adds the margin to the measured value (1.2 times if HighPower, 1.4 times if LowPower), and sets the upper limit torque value to stop the robot at the upper limit torque.

This is the example of automatic setting to repeat motion with the above settings from the sixth time.

When the upper limit torque value is changed, the changed value will be considered as "1.0" for the subsequent PTRQ measurement. If the margin of 1.2 times is set, PTRQ will be slightly larger than 0.8, and if the margin of 1.4 rimes is set, PTRQ will be slightly smaller than 0.7.

Setting example)

```
Function main
  Integer icnt
  Real rtrq(6)

  Motor On
  Power High
'   Power Low
  Weight 2
  Speed 50
  Accel 80, 80

  icnt = 1
  PTCLR
  LimitTorque 100         'init HighPower limit torque
  LimitTorqueLP 100       'init LowPower  limit torque
  CollisionDetect On
  Do
    Call all_ax_move
    Print PTRQ(1), PTRQ(2), PTRQ(3), PTRQ(4), PTRQ(5), PTRQ(6)
    icnt = icnt + 1
    If icnt = 5 Then
      If Power = 1 Then   'High power case
        Print "LimitTorque set"
        rtrq(1) = PTRQ(1) * 1.2 * LimitTorque(1) + 1.0
        rtrq(2) = PTRQ(2) * 1.2 * LimitTorque(2) + 1.0
        rtrq(3) = PTRQ(3) * 1.2 * LimitTorque(3) + 1.0
        rtrq(4) = PTRQ(4) * 1.2 * LimitTorque(4) + 1.0
        rtrq(5) = PTRQ(5) * 1.2 * LimitTorque(5) + 1.0
        rtrq(6) = PTRQ(6) * 1.2 * LimitTorque(6) + 1.0
        Print LimitTorque(1), LimitTorque(2), LimitTorque(3), LimitTorque(4),
LimitTorque(5), LimitTorque(6)
        LimitTorque rtrq(1), rtrq(2), rtrq(3), rtrq(4), rtrq(5), rtrq(6)
        Print LimitTorque(1), LimitTorque(2), LimitTorque(3), LimitTorque(4),
LimitTorque(5), LimitTorque(6)
        LimitTorqueStop On
      Else               'Low poser case
        Print "LimitTorqueLP set"
        rtrq(1) = PTRQ(1) * 1.4 * LimitTorqueLP(1) + 1.0
        rtrq(2) = PTRQ(2) * 1.4 * LimitTorqueLP(2) + 1.0
        rtrq(3) = PTRQ(3) * 1.4 * LimitTorqueLP(3) + 1.0
        rtrq(4) = PTRQ(4) * 1.4 * LimitTorqueLP(4) + 1.0
        rtrq(5) = PTRQ(5) * 1.4 * LimitTorqueLP(5) + 1.0
        rtrq(6) = PTRQ(6) * 1.4 * LimitTorqueLP(6) + 1.0
        Print LimitTorqueLP(1), LimitTorqueLP(2), LimitTorqueLP(3),
LimitTorqueLP(4), LimitTorqueLP(5), LimitTorqueLP(6)
        LimitTorqueLP rtrq(1), rtrq(2), rtrq(3), rtrq(4), rtrq(5), rtrq(6)
        Print LimitTorqueLP(1), LimitTorqueLP(2), LimitTorqueLP(3),
LimitTorqueLP(4), LimitTorqueLP(5), LimitTorqueLP(6)
        LimitTorqueStopLP On
      EndIf
    EndIf
    If icnt > 5 Then
```

```
      icnt = 6
    Endif
  Loop While icnt > 0

Fend

Function all_ax_move
  Integer icount
  Go JA(10, 10, 10, 10, 10, 10)
  Go JA(-10, -10, -10, -10, -10, -10)
Fend
```

7.18.12 Weight, Inertia, and Eccentricity / Offset Measurement Utility

Functional overview

Epson RC+ 8.0 supports "Weight, Inertia, and Eccentricity/Offset Measurement Utility". It is a function measures and sets the following three parameters with the customer's end effector attached to the robot.

- Load weight: Specify with the weight command
- Weight inertia: Specify with the inertia command
- Eccentricity (offset of weight center of gravity from J6 flange surface for the 6-axis robot)

Measurement methods and compatible models

Two methods to measure:

- Static (measured at low speed)

  Measures and calculates the parameter value as accurately as possible.

- Iteration (measured at high speed)

  Sets the target parameter value to use the motor torque appropriately, not to measure the exact value. In particular, for large weight, large inertia, and eccentric end effectors, places importance on setting appropriate values that do not break the robot and are well balanced with speed.

The combination of Static and iteration measurements is determined by the robot model. Table 1 and 2 describe it.

Table 1: Measurement methods and compatible models for the 6-axis robot (Static & iteration)

			Measurement posture (angle for each axis), motion area (motion range angle for each axis)					
Measurement parameters	Measurement methods	Measurement speed	J1	J2	J3	J4	J5	J6
WEIGHT/OFFSET	Static	Low speed	0 deg	0 deg	-3 to 3 deg	0 deg	-3 to 3 deg	0 deg
INERTIA	Iteration	High speed	0 deg	0 deg	0 deg	0 deg	0 deg	270 to -360 deg

			version 1.3 compatible models *1
Measurement parameters	Measurement methods	Measurement speed	C4, C8, C12, C4-B, C8-B, C12-B, VT6 *2, *3
WEIGHT/OFFSET	Static	Low speed	✓

			version 1.3 compatible models *1
Measurement parameters	Measurement methods	Measurement speed	C4, C8, C12, C4-B, C8-B, C12-B, VT6 *2, *3
INERTIA	Iteration	High speed	✓
A combination of measurements			Staic & iteration

*1: Refer to readme in the following project folder for the supported models in the latest version.
`C:\EpsonRC80\projects\Utilities\WeightInertiaMeasurement`

*2: N2 and N6 are not compatible.

*3: The wall mounting type is not compatible.

Table 2: Measurement methods and compatible models for the SCARA robot (Static & iteration, or iteration only)

			Measurement posture (angle for each axis), motion area (motion range angle for each axis)			
Measurement parameters	Measurement methods	Measurement speed	J1	J2	J3	J4
WEIGHT	Static	Low speed	0 deg	0 deg	0 to -50mm	0 deg
WEIGHT	Iteration	High speed	0 deg	0 to 90 deg	0 mm	0, 180 deg
INERTIA	Iteration	High speed	0 deg	0 deg	0 mm	-180 to 180 deg
Eccentricity	Iteration	High speed	0 to 90 deg	-75 to 90 deg	0 mm	-360 to 360 deg
Eccentricity (for RS3 and RS4)	Iteration	High speed	0 to 90 deg	55 to 220 deg	0 mm	-360 to 360 deg

			version 1.3 compatible models *1		
Measurement parameters	Measurement methods	Measurement speed	GX4, GX8, GX4-B, GX8-B, GX4-C, GX8-C, GX10-B, GX20-B, GX10-C, GX20-C, LS3-B, LS6-B, LS10-B, LS20-B*2	RS3, RS4	G3, G6, G10, G20 *3, T3, T6, T3-B, T6-B, LS3, LS6, LS20, LS3-B*V1, LS6-B*V1
WEIGHT	Static	Low speed	✓	-	-
WEIGHT	Iteration	High speed	-	✓	✓
INERTIA	Iteration	High speed	✓	✓	✓
Eccentricity	Iteration	High speed	✓	-	✓
Eccentricity (for RS3 and RS4)	Iteration	High speed	-	✓	-
A combination of measurements			Staic & iteration	Iteration only	

*1: Refer to readme in the following project folder for the compatible models in the latest version.
`C:\EpsonRC80\projects\Utilities\WeightInertiaMeasurement`

*2: Excludes LS3-B*V1, LS6-B*V1.

*3: G1 is not compatible.

Preparation for measurement

When measuring, secure the necessary space for the operation and attach an end effector to the robot to measure. The motion range depends on the model and the corresponding measurement method. Refer to table 1 and 2. This measurement utility also operates in the simulator. Check the motion range on the simulator beforehand. You can also check the approximate measurement time on the simulator. For high-speed measurements, operates with speed 100, accel 100, 100. It cannot be measured with an end effector which strength cannot withstand high speed. In addition, to move a wide motion range, measure without connecting wiring or piping.

Measurement and measurement time

Run a program written in the SPEL+ language to measure.

- Destination to save: `C:\EpsonRC80\projects\Utilities` (for default installation)
- Project name: WeightInertiaMeasurement

Execute the main function first.

- For the 6-axis robot: Measures in order of weight, offset, and inertia.
- For the SCARA robot: Measures in order of weight, inertia, and offset.

The measurement time takes 4 to 13 minutes. A high-payload (20kg) SCARA robot takes the longest time.

Check for the start of the measurement and the operation at low speed

Before starting the measurement, the following message appears:

```
Start Measurement: [y: yes, n: no]:
?
```

To measure, enter "y" or "Y". If you enter any other characters, the measurement exits.

When you enter "y" or "Y", after checking the motion range of all measurements at low speed, measures. Make sure that there are no problems with the motion range. The measurement time takes 2 to 4 minutes.

Display of the setting parameters before starting the measurement

The three setting parameters before starting the measurement appear as follows. It changes when you start the measurement. To return the parameters after measuring, modify them manually.

An example for the SCARA robot:

```
Current Weight: 1 kg, Current Inertia: 0.016 kgm2, Current Eccentricity: 0 mm.
```

An example for the 6-axis robot:

```
Current Weight: 1 kg, Current Inertia: 0.03 kgm2, Current Offset: 0 mm.
```

Parameter settings by the measurement

After the measurement starts, the above three parameters are changed, and the parameter is confirmed in the order of measurements and set to the Controller. All three parameters are set and the measurement ends. If the measurement is quit in the middle, the setting of the parameter values is not ensured. It is not possible to resume the measurement from the middle. If you quit, start the measurement again from the beginning.

Measurement details and what appears in the Run window

An example of the measurement is shown below. "<<" is a supplementary explanation.

An example of the 6-axis robot measurement (C8, combination of measurement: Static & Iteration)

```
Weight,Inertia,Offset/Eccentricity Measurement Utility  ver. 1.0.0. << version
display
2022/9/7  10:39:52
Model: C8-A701S, PerformMode  0
Max Weight: 8 kg, Max Inertia: 0.15 kgm2, Max Offset: 300 mm.
Current Weight: 1 kg, Current Inertia: 0.03 kgm2, Current Offset: 0 mm. <<current
set value
ROBOT MOVEMENT AREA
WEIGHT,OFFSET Measurement Movement Area: J1, J2, J4, J6 [0 deg.]; J3, J5 [-3 to 3
deg.]
INERTIA Measurement Movement Area: J1, J2, J3, J4, J5 [0 deg.]; J6 [270 to -360
deg.]
Start Measurement: [y: yes, n: no]:
?y << check for the start of the measurement
WEIGHT,OFFSET Measurement Movement Area: J1, J2, J4, J6 [0 deg.]; J3, J5 [-3 to 3
deg.]
Area Movement Check [Low Power Mode] << low-speed operation for the weight and the
offset measurement motion range
INERTIA Measurement Movement Area: J1, J2, J3, J4, J5 [0 deg.]; J6 [270 to -360
deg.]
Area Movement Check [Low Power Mode] << low-speed operation for the inertia
measurement motion range
------------------------------------------------------------
Start of WEIGHT,OFFSET Measurement for 6axis [Static Method]
------------------------------------------------------------
Warm up Movement: J3, J5 (Repeats 10 times)[High Power Mode] << warm-up operation
Start Measurement J3, J5 (Repeats 6 times)
Measurement 1. << starts measurements for the weight and offset (six measurements)
Measurement 2.
Measurement 3.
Measurement 4.
Measurement 5.
Measurement 6.
---------------------------------------
WEIGHT 5.7 kg, OFFSET 35 mm << the measurements and set values for the weight and
the offset
---------------------------------------
WEIGHT,OFFSET Measurement and Settings Completed.
------------------------------------------------------------
Start of INERTIA Measurement for 6axis [Iteration Method]
------------------------------------------------------------
Current weight : 5.7 kg, Current offset : 35 mm
Warm up Movement: J6 (Repeats 5 times)[High Power Mode] << warm-up operation
Start INERTIA Measurement: J6
Measurement 1. << starts the measurement for the inertia (one to twelve
measurements)
Measurement 2.
Measurement 3.
-----------------------
INERTIA : 0.13 kg*m2 << measured value for the inertia
-----------------------
INERTIA Measurement and Settings Completed.
-----------------------------------------------------------
WEIGHT : 5.7 kg, INERTIA : 0.13 kg*m2, OFFSET : 35 mm << final result and the set
value
-----------------------------------------------------------
motor off
2022/9/7  10:43:19
-------------- COMPLETE ------------------
```

An example of the SCARA robot measurement (GX8, combination of measurement: Static & Iteration)

```
Weight,Inertia,Offset/Eccentricity Measurement Program ver. 1.0.0. << version
display
2022/9/7  10:52:40
Model: GX8-A553S, PerformMode  0
Max Weight: 8 kg, Max Inertia: 0.16 kgm2, Max Eccentricity: 150 mm.
Current Weight: 4 kg, Current Inertia: 0.01 kgm2, Current Eccentricity: 0 mm.
<<current set value
ROBOT MOVEMENT AREA
WEIGHT Measurement Movement Area: J1, J2 [0 deg.]; J3 [0 to -50 mm.]; J4 [0 deg.]
INERTIA Measurement Movement Area: J1 [0 deg]; J2 [90 deg]; J3 [0mm]; J4 [-180 to
180 deg.]
ECCENTRICITY Measurement Movement Area: J1 [0 to 90 deg]; J2 [-75 to 90 deg.]; J3
[0mm]; J4 [-360 to 360 deg.]
Start Measurement: [y: yes, n: no]:
?y
WEIGHT Measurement Movement Area: J1, J2 [0 deg.]; J3 [0 to -50 mm.]; J4 [0 deg.]
Area Movement Check [Low Power Mode] << low-speed operation for the weight
measurement motion range
INERTIA Measurement Movement Area: J1 [0 deg]; J2 [90 deg]; J3 [0mm]; J4 [-180 to
180 deg.]
Area Movement Check [Low Power Mode] << low-speed operation for the inertia
measurement motion range
ECCENTRICITY Measurement Movement Area: J1 [0 to 90 deg]; J2 [-75 to 90 deg.]; J3
[0mm]; J4 [-360 to 360 deg.]
Area Movement Check [Low Power Mode] << moves the eccentricity measurement motion
range at low speed
--------------------------------------------------------
Start of WEIGHT Measurement for SCARA [Static Method]
--------------------------------------------------------
Warm up Movement: (Repeats 10 times)[High Power Mode] << warm-up operation at high
speed
Start WEIGHT: J3 (Repeats 5 times)
Measurement 1. << starts the measurement for WEIGHT (five measurements)
Measurement 2.
Measurement 3.
Measurement 4.
Measurement 5.
-----------------------
WEIGHT : 5.1 kg << measured value for the weight
-----------------------
WEIGHT Measurement and Settings Completed.
------------------------------------------------------------
Start of INERTIA Measurement for SCARA [Iteration method]
------------------------------------------------------------
Current Weight: 4.2 kg
Warm up Movement: (Repeats 5 times)[High Power Mode] << warm-up operation
Start Inertia Measurement: J4
Measurement 1. << starts the measurement for the inertia (one to twelve
measurements)
Measurement 2.
Measurement 3.
Measurement 4.
Measurement 5.
Measurement 6.
-----------------------
INERTIA : 0.07 kg*m2 << measured value for the inertia
-----------------------
INERTIA Measurement and Settings Completed.
------------------------------------------------------------
Start of ECCENTRICITY Measurement for SCARA [Iteration Method]
------------------------------------------------------------
Current weight : 5.1 kg, Current inertia : 0.07kgm2
Warm up Movement: (4 movements x 1 set)[High Power Mode] << warm-up operation at
```

```
high speed
Start ECCENTRICITY Measurement: J1-J4
Measurement 1. << starts the measurement for the eccentricity (one to thirteen
measurements)
Measurement 2.
Measurement 3.
Measurement 4.
Measurement 5.
-----------------------
ECCENTRICITY : 90 mm << ends the measurement and sets the value
-----------------------
ECCENTRICITY Measurement and Settings Completed.
------------------------------------------------------------
WEIGHT : 5.1 kg, INERTIA : 0.07 kg*m2, ECCENTRICITY : 90 mm
------------------------------------------------------------
motor off
2022/9/7  10:57:54
------------- COMPLETE -----------------
```

An Iteration method example of the measurement for the SCARA robot WEIGHT (LS6-B, combination of measurement: Static & Iteration Iteration only)

```
The measurement for Inertia and eccentricity is omitted because it is the same
operation as the GX8 example above.
-----------------------------------------------------------
Start of WEIGHT Measurement for SCARA [Iteration Method]
-----------------------------------------------------------
Warm up Movement: (Repeats 6 times)[High Power Mode] << warm-up operation at high
speed
Start WEIGHT Measurement: J2
Measurement 1. << starts the measurement for WEIGHT (one to thirteen measurements)
Measurement 2.
Measurement 3.
Measurement 4.
Measurement 5.
Measurement 6.
-----------------------
WEIGHT : 1 kg << ends the measurement and sets the value
-----------------------
WEIGHT Measurement and Settings Completed.
```

Precautions

- For the 6-axis robot, this function measures the tip weight of Arm 6. For the SCARA robot, it measures the weight attached to the shaft. For the weight of a workpiece, measure by attaching an equivalent object or set by adding the weight separately. The weight attached to the 6-axis arm 3 or 5 and the SCARA arm 2 should be converted to equivalent weight and added separately.

- The weight set value and the inertia set value have the minimum value (about 10% to 20% of the maximum value). 0kg, 0kgm ^{2}, Values close to these are not set. Light end effectors less the minimum value are rounded up to the minimum value. When operating with the simulator, the minimum value is set.

- It is recommended to use CollisionDetect with the default "ON". You can use the CollisionDetect command to check the current setting.

  ```
  > CollisionDetect
  ON,ON,ON,ON (always OFF in the simulator)
  ```

- The settings for this measurement are as follows: performmode: 0 (normal) accel: 100
  Use this setting with the upper limit of performmode "Normal" and accel "100". If you use performmode other than

"Normal", change the mode of this measurement as well. When changing, modify the following Gperformmode variables manually and rebuild it before executing it.

```
"Function main
GPerformMode = 0  '0:normal, 1:boost, 2:low vibration
main2
Fend"
```

- If you use other than "0" with a robot not compatible with the perform mode, the following message will appear, and exits.

```
PerfomMode 1 is not supported in this robot.
-- end --"
```

- This measurement result can be used to set the same model uses the same end effector. It cannot be used for setting to different models (even if the arm length is different). Repeat the measurement with the compatible model.
- If an error occurs in the middle of the program, force-quit the program, eliminate the cause (overload of the end effector, collision of the robot, etc.), and run it again.
- The following warning message may be output at each measurement and after all measurements.

```
"Warning: XXXXX over specification, please check the endeffector."(XXXXX is
Weight, Inertia, Offset, and Eccentricity)
```

The measured value is too large for the parameter. This measurement program ends after setting the maximum value, but check the design to make sure that there is no problem with the end effector specification.

- For not compatible model, the following message appears and the measurement is quit.

```
N2-A450SR is not supported.
-- end --"
```

- For the 6-axis wall mounting type, the following message appears and exits.

```
Wall mounted type manipulators are not supported.
C8-A701SW is not supported.
-- end --"
```

7.19 Working with Robot Points

A robot point is a set of coordinates that define a position in the robot work envelope. For SCARA and Cartesian robots, a point is defined by the position data (X, Y, Z) within the reference rectangular coordinate space and the orientation data (U) which is the rotation about the Z axis of the rectangular coordinate.

For 6-axis robots, a point is defined by the position and orientation of the tool coordinate system with respect to a reference rectangular coordinate system. The point is specified by the position data (X, Y, Z) and the orientation is specified by the orientation data (U, V, W) which correspond with roll (rotation about the Z axis), pitch (rotation about the Y axis), and yaw (rotation about the X axis).

When the additional ST axis is installed, the point is specified by the position data of each additional axis (S, T).

The X, Y, and Z coordinates of a point are specified in millimeters.

The U, V, and W coordinates are specified in degrees.

The S and T coordinates of a point are specified in millimeters or degrees, according to the type of axis.

Points are referenced using the letter P followed by an integer number or integer expression or by a label defined in the point file editor or [Robot Manager]-[Jog & Teach] page.

7.19.1 Defining points

You can define points in a program statement, points editor window, [Robot Manager]-[Jog & Teach] page, or at the [Command] window.

In a program statement or at the [Command] window, you can assign coordinates to a point you want to register or define the current position of the robot.

```
P1 = XY(200, 100, -25, 0)     'Assign coordinates to point P1
Pick = XY(300, 200, -45, 0)  'Assign coordinates to point pick
P10 = Here                    'Assign a point to current position
```

7.19.2 Referencing points by point label

You can assign names to point numbers so you can refer to points by name in a program.

Assign names on [Robot Manager]-[Point Data] tab on the toolbar or from the editor of the point data in the Epson RC+ 8.0 menu-[File]-[Open File]. Names must be unique for each point number when used in the same point file.

Point labels can include up to 32 bytes of alphanumeric, Japanese, Chinese, and underscore characters. However, the first character must be an English, Japanese, or Chinese character. Only alphabets and Japanese can be used for the first letter.

```
For i = 0 To 10
  Go pick
  Jump place
Next i
```

7.19.3 Referencing points with variables

Use the letter P followed by a variable name within parentheses that represents the point number you are referencing.

```
For i = 0 To 10
  Go P(i)
Next i
```

KEY POINTS

Although you can define points at the [Command] window for test purposes, it is recommended that all points be defined in a program, point editor, or with the [Robot Manager]-[Jog & Teach] page. Points defined at the [Command] window will be cleared from memory when you build a project or run a program unless you execute "SavePoints".

7.19.4 Using points in a program

When starting programs, the default point file for the robot is loaded. You can also load other points in the program using the LoadPoints statement.

```
Function main
  Integer i

  LoadPoints "model1.pts"
  For i = 0 To 10
    Jump pick
    Jump place
  Next i
Fend
```

7.19.5 Importing points into program

You can import points into the current project while the program is running using the ImportPoints statement.

```
Function main
    Integer i

    ImportPoints "c:\models\model1.pts", "robot1.pts "
    LoadPoints "robot1.pts"
    For i = 0 To 10
      Jump pick
      Jump place
    Next i
Fend
```

7.19.6 Saving and loading points

Use "LoadPoints" to load a point file in the current project. You can optionally specify the Merge parameter to combine points in a file with points that have already been loaded.

Use "SavePoints" to save the points in a point file. If the point file is in the current project, it will be updated on the PC when it is connected and the same project is open.

If the point file is not the current project, it will not be automatically updated on the PC. Use Project Synchronize to copy the file to the PC if desired.

⚠ CAUTION

Rebuild the program if Project Synchronize is performed.

7.19.7 Point attributes

Each point definition can optionally specify a local number and various arm orientations, depending on the robot type.

You can specify point attributes in point assignment statements or use individual statements and functions to change the attributes of a previously defined point.

Local point attribute

To specify a local coordinate system number for a point in an assignment statement, add a forward slash (/) followed by the local number after the coordinates of the point.

```
P1 = XY(300, -125.54, -42.3, 0) /1  ' P1 is in local 1
```

The local number can also be an expression enclosed in parentheses.

```
P2 = P3 /(mylocal)
```

Use the PLocal function and statement to read and set the local attribute of a point.

Hand point attribute

To specify orientation for the SCARA or 6-axis robot, add a forward slash (/) followed by L (for Lefty hand orientation) or R (for Righty hand orientation).

```
P2 = XY(200, 100, -20, -45) /L  ' Hand orientation is Lefty
P3 = XY(50, 0, 0, 0) /2 /R      ' Lefty in Local 2
```

You can read and set point hand orientation using the Hand statement and function.

```
Hand P1, Righty
```

Elbow point attribute

To specify elbow orientation for the 6-axis robot in a point assignment statement, add a forward slash (/) followed by A (Above elbow orientation) or B (Below elbow orientation),

Elbow orientation is Below.

```
P1 = XY(0, 600, 400, 90, 0, 180) /B
```

You can read and set point elbow orientation using the Elbow statement and function.

Wrist point attribute

To specify wrist orientation for the 6-axis robot in a point assignment statement, add a forward slash (/) followed by NF (NoFlip wrist orientation) or F (Flip wrist orientation).

Wrist orientation is Flip.

```
P2 = XY(0, 600, 400, 90, 0, 180) /F
```

You can read and set point wrist orientation using the Wrist statement and function.

J4Flag and J6Flag point attributes

At some points in the work envelope, the 6-axis robot can have the same position and orientation even if the fourth joint or the sixth joint is rotated 360 degrees. To distinguish these points, the J4Flag and J6Flag point attributes are provided.

To specify the J4Flag in a point assignment statement, add a forward slash (/) followed by J4F0 (-180 < the forth joint angle <= 180) or J4F1 (the forth joint angle <= -180 or 180 < the forth joint angle).

```
P2 = XY(0, 600, 400, 90, 0, 180) /J4F1
```

To specify the J6Flag in a point assignment statement, add a forward slash (/) followed by J6F0 (-180 < the sixth joint angle <= 180), J6F1 (-360 < the sixth joint angle <= -180 or 180 < the sixth joint angle <= 360), or J6Fn (-180*(n+1) < the sixth joint angle <= -180*n or 180*n < the sixth joint angle <= 180*(n+1)).

```
P2 = XY(50, 400, 400, 90, 0, 180) /J6F2
```

J1Flag and J2Flag point attributes

At some points in the work envelope, the RS series can have the same position and orientation even if the first joint or the second joint is rotated 360 degrees. To distinguish these points, the J1Flag and J2Flag point attributes are provided.

To specify the J1Flag in a point assignment statement, add a forward slash (/) followed by J1F0 (-90 < the first joint angle <= 270) or J1F1 (-270 <= the first joint angle <= -90 or 270 < the first joint angle <= 450).

```
P2 = XY(-175, -175, 0, 90) /J1F1
```

To specify the J2Flag in a point assignment statement, add a forward slash (/) followed by J2F0 (-180 < the second joint angle <= 180), J2F1 (-360 < the second joint angle <= -180 or 180 < the second joint angle <= 360).

```
P2 = XY(300, 175, 40, 90) /J2F1
```

J1Ang and J2Flag point attributes

At the origin of the robot coordinate system, the RS series can have the same position and orientation even if the first joint is rotated. To distinguish these points, the J1Ang point attributes are provided.

7.19.8 Extracting and setting point coordinates

Use the CX, CY, CZ, CU, CV, CW, CS, and CT commands to get a coordinate of a point and set it.

```
xcoord = CX(P1)
P2 = XY(xcoord, 200, -20, 0)
ycoord = CY(P*)  ' Gets current Y position coordinate

CX(pick) = 25.5
CY(pick) = CY(pick) + 2.3
```

7.19.9 Alteration of points

There are several ways of modifying a point without re-teaching it. You can change one or more coordinate values with relative offsets or absolute values.

To set an absolute value for a coordinate, use a colon followed by the axis letter and the value.

To add a relative offset to a coordinate, use an axis letter followed by the offset value or expression in parentheses.

If the offset is negative, then precede the axis letter with the minus sign. If parentheses are omitted, they will be automatically added.

```
Go P1 -Z(20)
```

Move to P1 with a z offset of 20mm.

```
Go P1 :Z(-25)
```

Move to P1 with a z absolute position of -25mm

```
Go P1 -X(20) +Y(50) :Z(-25)
```

Move to P1 with offsets for X and Y relative offsets and an absolute position for Z

Point alternation of 6-axis robot

When changing the orientation by roll (U), pitch (V), and yaw (W) in the SPEL+ program, adding the angles to V and W axes (e.g. +V(10), +W(10)) does not mean the rotation of Y and X axes in the robot coordinate system. To change the orientation (U, V, and W) after teaching the points, set the robot to actual posture by Jog & Teach in the Robot Manager.

7.20 Input and output control

7.20.1 Hardware I/O

There are 24 DC inputs and 16 DC outputs on a standard controller. By purchasing I/O boards, you can add additional inputs and outputs. You can expand the I/O by using the Fieldbus I/O master option and Fieldbus I/O slave option. Also, you can input/output the Analog signal by using the Analog I/O board option. See details below.

Inputs and Outputs

KEY POINTS

I/O boards cannot be added to T series and VT series manipulators.

7.20.2 Memory I/O

There are 128 bytes (1024 bits) of memory I/O. Memory I/O is especially useful for synchronizing multi-tasking. Each memory bit can be treated as both an input and an output.

Use the commands with the "Mem" prefix for memory I/O.

7.20.3 I/O Commands

Command	Description
In	Reads one byte (eight bits) of input data.
InW	Reads one word (sixteen bits) of input data.
MemIn	Reads one byte (eight bits) of Memory I/O.
MemInW	Reads one word (sixteen bits) of Memory I/O.
MemOff	Turns off one Memory I/O bit.
MemOn	Turns on one Memory I/O bit.
MemSw	Read status of one bit of memory I/O.
Off	Turns off one output bit.
On	Turns on one output bit.
Out	Sets/reads one byte (eight bits) of output data.

Command	Description
OutW	Sets/reads one word (sixteen bits) of output data.
Oport	Reads the status of one output bit.
InBCD	Reads input data in BCD (binary coded decimal) format.
OpBCD	Outputs one byte of output data in BCD format.
Sw	Read status of one bit of hardware inputs or memory inputs.

7.21 Using Traps

Traps enable a program to jump to a label or enable a function to be called when a certain event occurs.

Traps are divided into the following two types:

- 4 Traps are fired by user defined input
- 7 Traps are fired by system

You should keep trap functions short and avoid continuous loops. According to the type, some Traps must be re-armed. Also, some motion commands are limited to execute in trap functions.

For details on Trap commands, refer to the following manual:

"SPEL+ Language Reference"

Here is a simple example for a trap. In this example, when input 1 turns on, it executes the Sw1Trap function.

```
Function main
  ' Sets the trap
  Trap 1 Sw(1) = On Xqt Sw1Trap
  Do
    RunCycle
  Loop
Fend

Function Sw1Trap
  ' Turn on output 1 for 2 seconds
  On 1, 2
  ' Wait for trap condition to clear
  Wait Sw(1) = Off
  ' Re-arm the trap
  Trap 1 Sw(1) = On Xqt Sw1Trap
Fend
```

Trap	Description
Trap 1 - 4 Goto	Triggered by an input condition specified by the user. User traps can use GoTo, Call, or Xqt.
Trap 1 - 4 Call	
Trap 1 - 4 Xqt	
Trap Emergency Xqt	When Emergency Stop occurs, a specified function is executed.

Trap	Description
Trap Error Xqt	When an error occurs, a specified function is executed.
Trap SgOpen Xqt	When the Safeguard circuit is open, a specified function is executed.
Trap SgClose Xqt	When the Safeguard circuit is closed, a specified function is executed.
Trap Pause Xqt	When the system enters the Pause state, a specified function is executed.
Trap Abort Xqt	When all tasks (except background tasks) have been stopped by user or system, such as when a command corresponding to Abort All is executed, a specified function is executed.
Trap Finish Xqt	When all tasks (except background tasks) have been finished, a specified function is executed. However, the function will not be executed under the condition that executes Trap Abort.

7.21.1 Cautions of Trap when it triggers the system condition

⚠ CAUTION

Forced Flag

Specify Forced flags in I/O output commands such as On or Off to enable On/Off of I/O outputs at the following times:

During emergency stop, when safety door is open, in teach mode, or when an error occurs

Do not connect external equipment that operates mechanically, such as actuators, to an I/O output that specifies a Forced flag. Doing so may cause external equipment to operate at the following times, which can be extremely dangerous:

During emergency stop, when safety door is open, in teach mode, or when an error occurs

Forced flag is designed to be specified for I/O outputs connected to external equipment without mechanical motion such as status display LEDs.

[Outputs off during emergency stop] setting

Uncheck the [Outputs off during emergency stop] checkbox in the [Preferences] page of [Setup]-[System Configuration]-[Controller] to execute I/O On/Off using the Trap Emergency Xqt task after Emergency Stop. If this checkbox is checked, the execution order of turning Off by the controller and turning On using the task are not guaranteed.

7.22 Special Tasks

Each task of SPEL+ pauses by Pause input or Safety Door open and stops by Emergency Stop or Error. Therefore you cannot create a system that monitors the whole system.

To enable the Robot Controller to monitor the whole system, the following special tasks are provided:

- NoPause/NoEmgAbort task

You can create a task that continues a processing even when the Pause is input or safeguard is open by specifying NoPause or NoEmgAbort as a task type when creating task by Xqt.

- Background task

  You can create a task that starts as the controller power is turned ON and continues a processing even when the Pause is input or safeguard is open.

These special tasks are useful tasks but may reduce the safety of the system by using them improperly.

Be sure to understand the following items when using these tasks.

7.22.1 Precautions to Use the Special Tasks

⚠ CAUTION

Forced Flag

Specify Forced flags in I/O output commands such as On or Off to enable On/Off of I/O outputs at the following times:

During emergency stop, when safety door is open, in teach mode, or when an error occurs

Do not connect external equipment that operates mechanically such as actuator to the I/O output that specifies Forced flag. Doing so may cause external equipment to operate at the following times, which can be extremely dangerous:

During emergency stop, when safety door is open, in teach mode, or when an error occurs

Forced flag is designed to be specified for I/O outputs connected to external equipment without mechanical motion such as status display LEDs.

NoEmgAbort Task

When Emergency Stop or errors occur, finish the task promptly after completing the error handling.

If you do not complete the NoEmgAbort task, the controller does not change to Ready status and you cannot cancel the Emergency Stop or the error. You cannot execute Reset command from the NoEmgAbort task to cancel the Emergency Stop or the error automatically.

NoEmgAbort task is designed for I/O process without motion and communication with external device using the Ethernet. Therefore there are commands such as robot motion commands that cannot be executed in the NoEmgAbort task. An error occurs if you use these commands. The list of these commands is in the next section.

For details, refer to Help or the following manual:

"SPEL+ Language Reference - Xqt"

NoPause Task

NoPause task continues the operation during the Pause or Safety Door open condition. However, when a robot is operating NoPause task, the task pauses as the robot pauses.

Background task

Background task always exists while the controller is working, and it is designed for monitor of the entire system and communication with external device. Therefore there are commands such as robot motion commands that cannot be executed in the background task. An error occurs if you use these commands. The list of these commands is in the next section.

In addition, the background task continues processing even when Pause is input or safeguard is open, so it doesn't affect the controller state transition.

See details below.

Background Task

[Outputs off during emergency stop] setting

Uncheck the [Outputs off during emergency stop] checkbox in the [Preferences] page of [Setup]-[System Configuration]-[Controller] to execute I/O On/Off using the NoEmgAbort task or background task after Emergency Stop. If this checkbox is checked, the execution order of turn Off by the controller and turn On using the task are not guaranteed.

[Safeguard open stops all tasks] setting

When the [Outputs off during emergency stop] checkbox in the [Preferences] page of [Setup]-[System Configuration]-[Controller] is checked, NoPause task stops by Safety Door open. NoEmgAbort task or background task continues the task.

Setting of [Enable the Background task]

Check the [Enable background tasks] checkbox in the [Preferences] page of [Setup]-[System Configuration]-[Controller] when you use the background task.

Setting of [Initialize global variables when function starts]

Uncheck the [Clear globals when MainXX function started] checkbox in the [Preferences] page of [Setup]-[System Configuration]-[Controller] when you use the global variables from the background task. When this checkbox is checked, the controller will initialize the variables and the variable-access conflict from tasks will occur.

⚠ CAUTION

Setting of [Enable advanced task commands]

Check the [Enable advanced task commands] checkbox in the [Preferences] page of [Setup]-[System Configuration]-[Controller] when you execute the following commands from a background task:

StartMain, Cont, Recover, Reset Error, Reset

When you execute these commands from a task, you should understand each command specification and verify that the system has the appropriate conditions. Note that improper use, such as executing commands continuously in a loop, can reduce the security of system.

7.22.2 NoPause/NoEmgAbort task specification

Status by Event and Task

Event	Task Type		
	Normal	NoPause	NoEmgAbort
Pause Statement, Pause Input, Pause Button	Pause	Continue *1	Continue
Safety Door Open	Pause *2	Continue *1 *2	Continue
Error during Auto Mode	Stop	Stop	Continue
Error during Program Mode	Pause	Pause	Continue

Event	Task Type		
	Normal	NoPause	NoEmgAbort
Emergency Stop	Stop	Stop	Continue
Stop Button, Stop Input	Stop	Stop	Stop
Halt Statement, Halt Button	Pause	Pause	Pause
Brake Point	Pause	Pause	Pause
Switching to Teach Mode	Stop	Stop	Stop

*1 When the robot is operating, the task pauses as the robot pauses.

*2 When [Outputs off during Emergency Stop] is checked in the [Preferences] page of [Setup]-[System Configuration]-[Controller], normal tasks and NoPause tasks stop when the Safety Door is opened.

Task Execution

- Normal: Omit the task type in Xqt statement, or specify Normal for the task type.

```
Xqt NormalTask
Xqt NormalTask, Normal
```

- NoPause: Specify NoPause in Xqt statement.

```
Xqt NoPauseTask, NoPause
```

- NoEmgAbort: Specify NoEmgAbort in Xqt statement.

```
Xqt NoEmgAbortTask, NoEmgAbort
```

You cannot change the task type after executing a task.

main to main63 that are executed at the beginning of the program are executed as normal tasks.

Type of a task executed in Trap Xqt is determined by the event type.

For details, refer to Help or the following manual:

"SPEL+ Language Reference - Trap"

Restricted Commands by Task Types

- Normal: No restriction
- NoPause: No restriction
- NoEmgAbort: Cannot execute the following commands.
  - Command for robot motion
  - Commands for vision
  - Reset, Xqt, Trap, etc.

  For details, refer to Help or the following manual:

"SPEL+ Language Reference - Xqt"

7.22.3 NoPause/NoEmgAbort task example

The following example shows a program that monitors the error of the controller and switches the I/O On/Off when error occurs according to the error number.

Program examples of ErrOn, EStopOn, and SafetyOn are indicated in the SPEL+ Language Reference.

```
Function main
  Xqt ErrorMonitor, NoEmgAbort
  :
  :
Fend

Function ErrorMonitor
  Wait ErrorOn
  If 4000 < SysErr And Syserr < 5999 Then
    Print "Mortion Error = ", SysErr
    Off 10, Forced
    On 12, Forced
  Else
    Print "Other Error = ", SysErr
    Off 11, Forced
    On 13, Forced
  EndIf
Fend
```

7.23 Background Task

7.23.1 Primary features of background task

The purpose of the background task is to monitor the status of the cell as a whole and to communicate with external devices.

Function BgMain, a function specified as the "Background task" will be automatically activated as task 65 when the controller starts and loads the project.

If another task is created within the background task using the XQT command, that created task will be assigned to task No.65 (and onward in the ascending order) and will also function as a background task. In addition, specifying a task type for an XQT command in a background task has no meaning.

An operator is not necessarily aware of the operating Background task which does not stop at the input of emergency stop or safeguard signal. The Background task will not stop when an operator inputs "PAUSE" or "ABORT".

In this sense, the background task functions for the application program to work as a part of the system.

On the other hand, the execution commands to operate the Manipulator, set-up commands for the Manipulator or the commands for image processing cannot be executed within the background task.

⚠ CAUTION

- Specify Forced flags in I/O output commands operated from background tasks to enable On/Off of I/O outputs at the following times:

  During emergency stop, when safety door is open, in teach mode, or when an error occurs

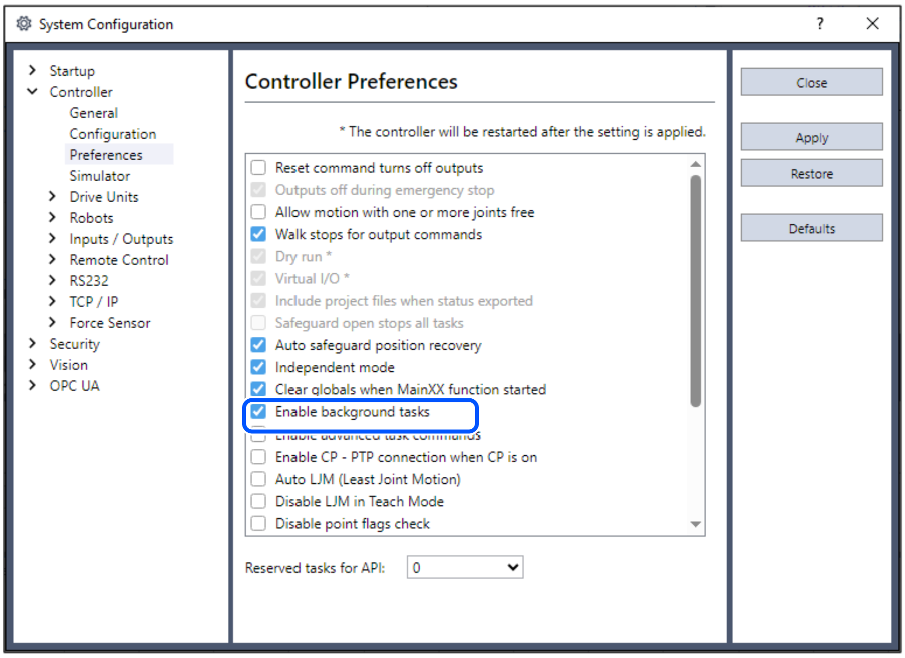
- Do not connect external equipment that operates mechanically, such as actuators, to an I/O output that specifies a Forced flag. Doing so may cause external equipment to operate at the following times, which can be extremely dangerous:

  During emergency stop, when safety door is open, in teach mode, or when an error occurs

  Forced flag is designed to be specified for I/O outputs connected to external equipment without mechanical motion such as status display LEDs.

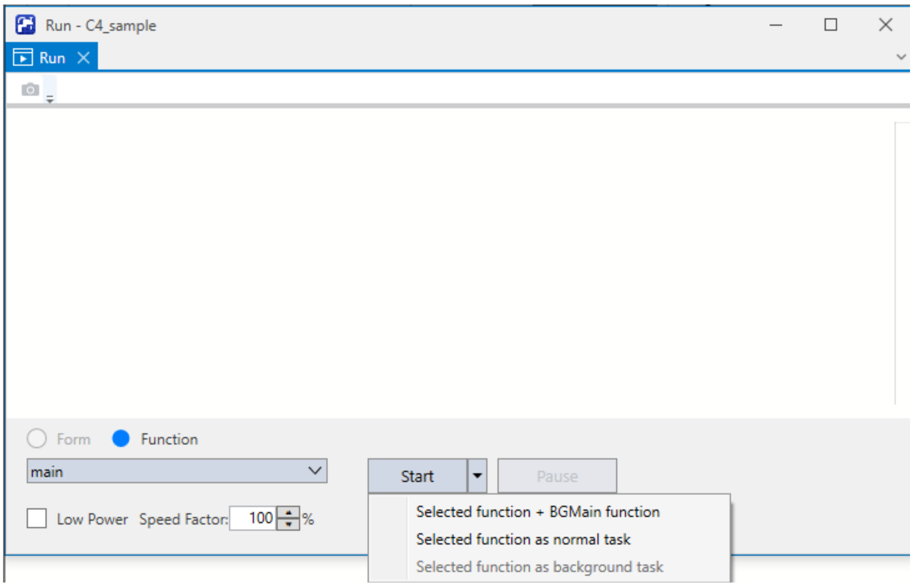
7.23.2 Setup and start the background task

When you use the background task, first check the [Enable background tasks] in the [Preferences] page of [Setup]-[System Configuration]-[Controller].



When you have already checked the box above and the Function BgMain exists in your program, it will automatically start as Task 65 as the controller starts and loads the project, it executes as a "Background task".

However in PROGRAM mode, the Function BgMain will not start automatically. Start from the [Start] button in the [Run] window. The PROGRAM mode is for creating programs and debugging. It may be more efficient if the BgMain function doesn't start.

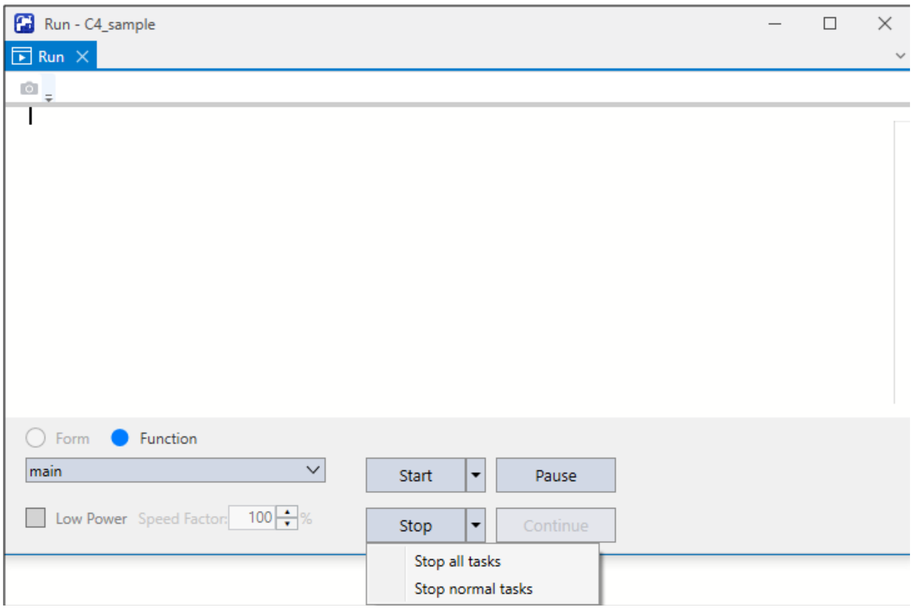


When the controller operating mode shifts from PROGRAM to AUTO mode, the Function BgMain will start automatically.

7.23.3 Holding background task (from being activated)

The purpose of the background task is to monitor the status of the cell as a whole and to communicate with external devices. It is activated before a non-background task is activated and continues to function when the non-background task either generates an error or is aborted by an operator. In this sense, the background task can be a program that never stops functioning.

The background task can be debugged in PROGAM mode. By clicking the [Stop] button in the [Run] window, you can choose whether to abort the background task as well.



In the [Task Manager] window, the background tasks can be managed in the same way as the non-background tasks except for the [Pause/Cont] button. You can set a break point in a background task and step through the code.

As a rule, the background task cannot be controlled in AUTO mode. It is by design that any error that occurs in the background task cannot be recovered in AUTO mode. Therefore, thorough debugging in PROGRAM mode is recommended. Be particularly careful that the communication errors are handled properly without fail before using the background task in AUTO mode.

The following tables show how the background will (or will not) be affected by operation from the console.

Operator Window

Button	Background task
Start	It will not be affected.
Abort	It will not be affected.
Pause	It will not be affected.
Continue	It will not be affected.

Remote Input

Button	Background task
Start / Stop	It will not be affected.
Pause / Continue	It will not be affected.
Reset	It will not be affected.
Shutdown	It will be stopped.

Run Window (PROGRAM mode)

Button	Background task
Start	You can select how to start the task.
Abort	You can select how to abort the task: abort only non-background task or abort all tasks including the background task.
Pause	It will not be affected.
Continue	It will not be affected.

Task Manager (PROPGRAM mode)

Button	Background task
Halt / Resume	When the background task is selected, you cannot execute Halt/Resume.
Quit	When the background task is selected, you can execute Quit.
Pause/Cont	It will not be affected.
Stop	All tasks including the background task will stop.

Break point (PROGRAM mode)

Switch name	Background task
Set a break point	You can set a breakpoint to the background task. It will pause at the break point.
Step Into	Available
Step Over	Available
Continue	Available

Switch name	Background task
Walk	Available However, motion commands cannot be executed from background tasks.

7.23.4 Commands that will cause error in background task

The following commands are prohibited in background tasks and execution will result in error: commands that relate to the Manipulator operation or operation settings commands that relate to the Vision relation instruction TRAP commands If a program that is to be executed as the background task includes any of the following commands, it will result in error when executed.

However, using the command related to the Manipulator operation settings or the Manipulator settings to gain the current setting values or refer to them will not result in error:

Commands that will cause error are almost the same as with NoEmgAbort, but there are some commands such as Xqt that can be executed in a background task. For details, refer to Help or the following manual:

"SPEL+ Language Reference - Xqt"

7.23.5 Background task and Remote control

No matter whether the background task is being executed or not, it doesn't affect the remote I/O outputs Ready, Running, and Pause. For example, even if the background task is being executed, when no non-background tasks (Task No. 1 - 32) are being executed, the READY output will be ON.

7.24 Constants

There are several predefined constants for use in SPEL+ program. A project build time, the values for these constants are substituted for the constant name.

For the predefined constants list, refer to the following manual:

"SPEL+ Language Reference"

7.25 Calling native functions in dynamic link libraries

Epson RC+ 8.0 allows you to call native functions in Dynamic Link Libraries (DLLs).

This is used for complicated arithmetic processing and call for a native function of an external device.

To call the native DLL function, use a Declare statement which is a function definition command from the SPEL+ program and write a function call as normal.

For details, refer to the following manual.

"SPEL+ Language Reference - Declare"

Sample of calling a native DLL

By using a development tool such as Microsoft Visual Studio 2019, you can create a native DLL that can be called from SPEL+. Here, it uses Visual Studio 2019 as a sample to create a function that executes the arithmetic operator.

Step 1: Decide on variable type for a native DLL

You need to plan the data type to use for transferring with the native DLL in the Epson RC+ 8.0. Correspondence table for the Epson RC+ 8.0 data type and the C/C++ variable type is shown below. You cannot use the C/C++ byte type and structure because the Epson RC+ 8.0 has no correspond data for them.

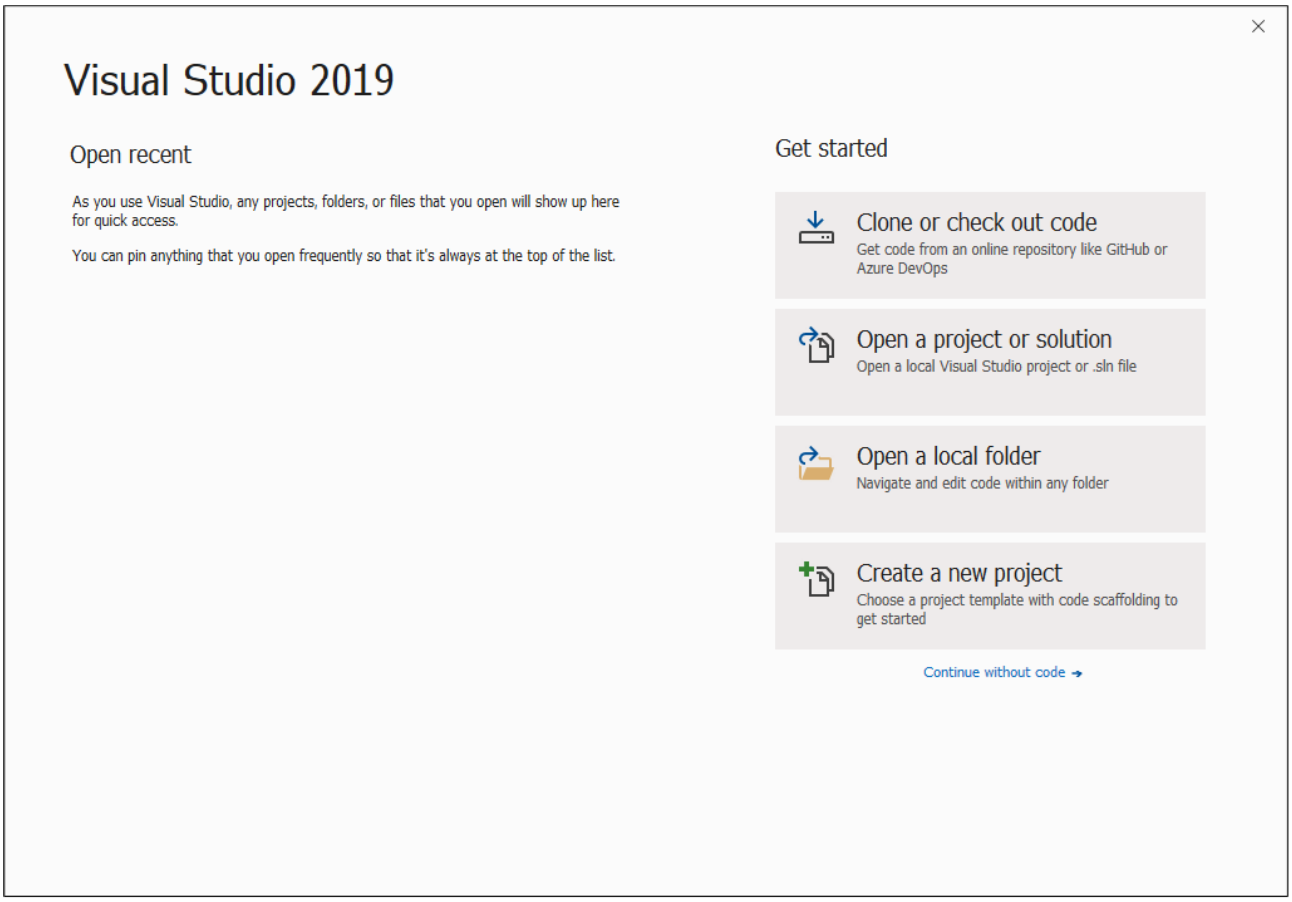
Data correspondence

Epson RC+ 8.0	C/C++
Boolean	short
Byte	short
Short	short
Integer	short
Long	int
Real	float
Double	double
String	char [256] * Null included

Step 2: Create a native DLL

1. Start Visual Studio 2019.

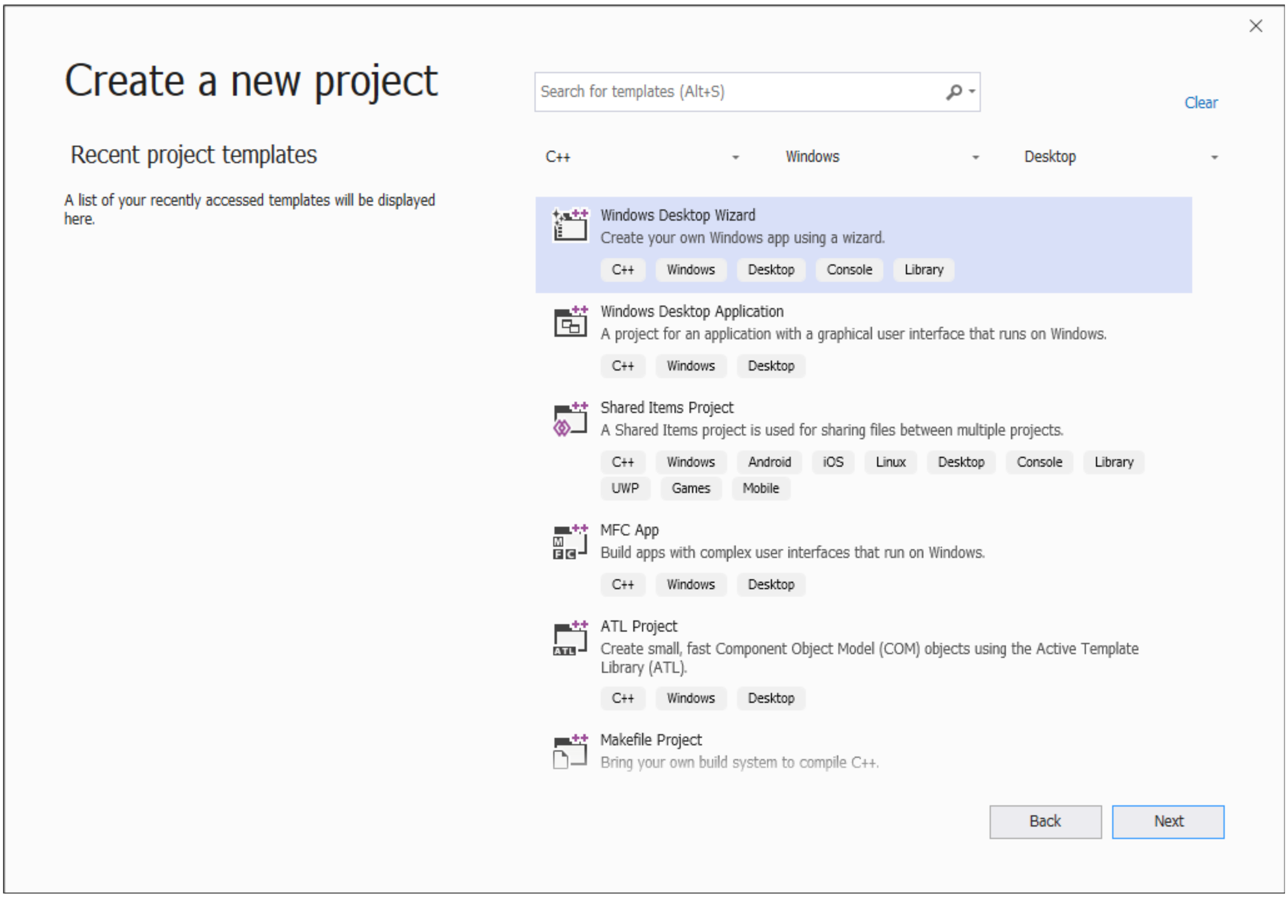
   Select the "Create a new project" on the start window.



2. [Create a new project] dialog box will appear.

   i. Select “Windows Desktop Wizard” from the project template list shown on the right of dialog box.

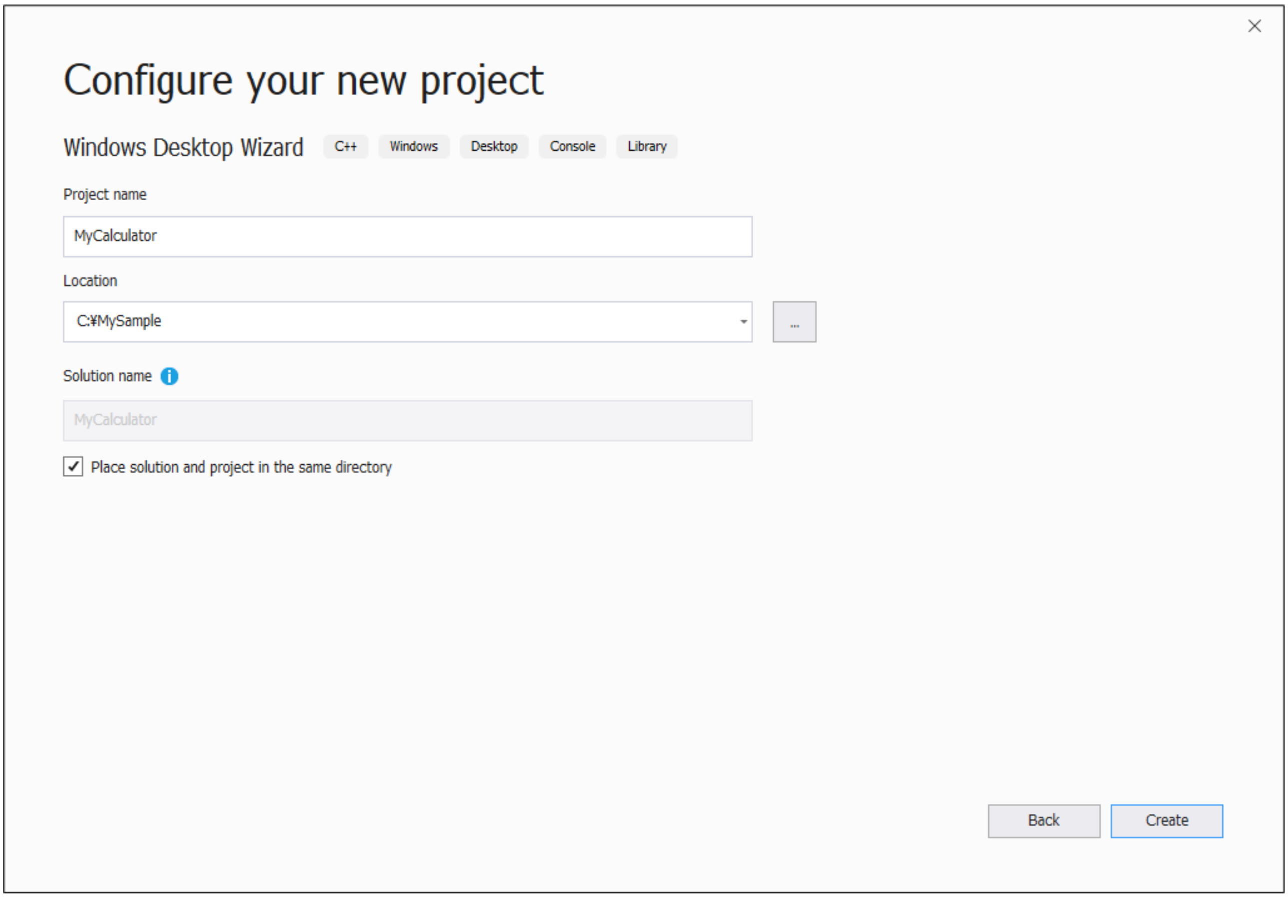
ii. Click the [Next] button.



3. Start the Windows Desktop Wizard.

i. Type in a name for a project in the [Project name] box. (Here types in "MyCalculator".)

ii. Click the [Create] button.

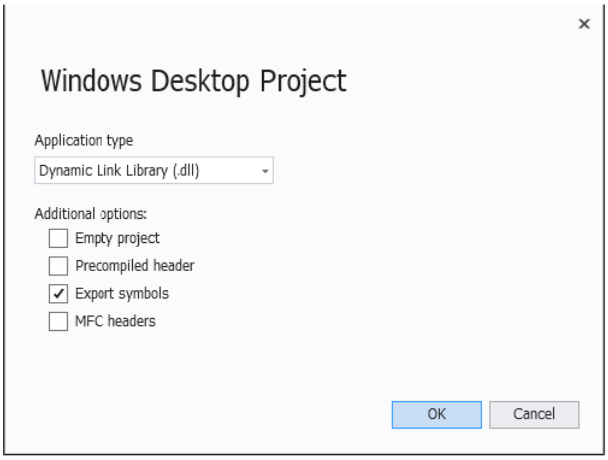


4. Set the project options.

   i. Select the “Dynamic Link Library (.dll)” in the [Application type].

   ii. Check the [Export symbols] box in the [Additional options:].

   iii. Click the [OK] button.



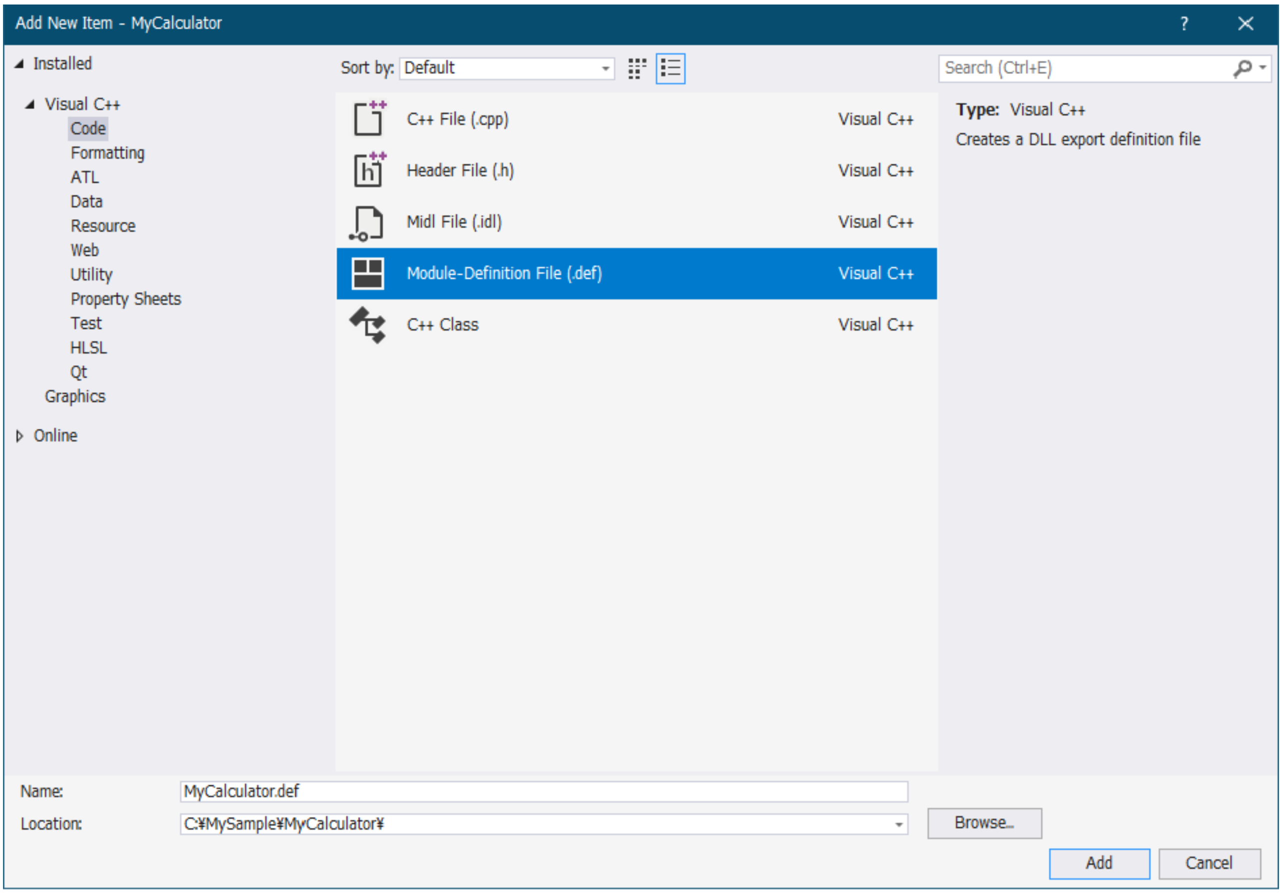
5. A simple example of function “fnMyCalculator” will be created in MyCalculator.cpp. Add a function MyArithmetic which executes the arithmetic operator to this file.

```
MYCALCULATOR_API float MyArithmetic(short value1, short value2, char * kind )
{
  if ( !strcmp(kind, "add") )
  {
    return (float)(value1 + value2);
```

```
    }
    else if ( !strcmp(kind, "sub") )
    {
      return (float)(value1 - value2);
    }
    else if ( !strcmp(kind, "mul") )
    {
      return (float)(value1 * value2);
    }
    else if ( !strcmp(kind, "div") )
    {
      return (float)(value1) / (float)(value2);
    }
    else
    {
      strcat_s(kind, 10, " NG");
      return 0;
    }
}
```

6. Export a function to enable it to be called from SPEL+.

i. Select Visual Studio 2019 menu-[Project]-[Add New Item]. The [Add New Item] dialog box will appear.



ii. Select the [Visual C ++]-[Code] in the tree on the left.

iii. Select “Module-Definition File (.def)” from the project template list shown on the center of dialog box.

iv. Type in a file name in the [Name:]. (Here types in "MyCalculator.def" as a file name.)

v. Click the [Add (A)] button.

vi. Register “fnMyCalculator function” and “MyArithmetic function” to the created “MyCalculator.def” file.

```
LIBRARY "MyCalculator"
EXPORTS
fnMyCalculator
MyArithmetic
```

7. Build the project and create the DLL. Select [Win32] as a solution platform for Visual Studio 2019. Then, select Visual Studio 2019 menu-[Build]-[Build MyCalculator]. DLL will be successfully created if no error occurs.

> **KEY POINTS**
>
> In Epson RC+ 8.0, 64 bit native DLL is not available. When using a DLL created with a version older than Visual Studio 2015, it is necessary to install the runtime corresponding to that version in advance.

Step 3: Call the DLL function from SPEL+

> **KEY POINTS**
>
> You can now try your DLL function from SPEL+. Before you call your function from the Epson RC+ 8.0, you must debug it and check thoroughly if it can work without errors. In case that error occurs (such as system error) in the native function, the Epson RC+ 8.0 will not work normally.

1. Copy the created MyCalculator.dll to the Epson RC+ 8.0 project folder (e.g. `C:\EpsonRC80\projects\dllcall`).

2. Define a DLL function which executes the arithmetic operator in the SPEL+ program and write a function call for MyArithmetic in Function main.

```
Declare MyArithmetic, "MyCalculator.dll"(value1 As Integer, value2 As Integer,
ByRef calc$ As String) As Real
Function main
  Real result;
  String calc$

  calc$ = "add"
  result = MyArithmetic(1, 2, ByRef calc$);
  Print "1+2=", Str$(result)
  calc$ = "sub"
  result = MyArithmetic(1, 2, ByRef calc$);
  Print "1-2=", Str$(result)
  calc$ = "mul"
  result = MyArithmetic(1, 2, ByRef calc$);
  Print "1*2=", Str$(result)
  calc$ = "div"
  result = MyArithmetic(1, 2, ByRef calc$);
  Print "1/2=", Str$(result)
Fend
```

3. Build and execute the project. The following result will be displayed.

```
1+2=3
1-2=-1
1*2=2
1/2=0.5
```

KEY POINTS

Before you build the project, be sure to copy the native DLL to the project folder without fail. If you fail, a warning or error will occur.

Please note the dependency of the DLL when using a third party DLL as a native DLL. If the dependent DLL does not exist in the project folder or the folder set in the environment variable PATH of Windows, a warning or error will occur.

8. Building SPEL+ Applications

8.1 Designing Applications

8.1.1 Designing Simple Applications

The simplest SPEL+ application has one program and one point file. This is what is automatically defined for you when you create a new project. A blank program named "Main.prg" and a blank point file named "Points.pts" are created.

To write and run a simple application

1. Select [New Project] from the [Project] menu to create a new project.
2. Write your program source code in the file that was created for you called “Main.prg”.
3. Teach the robot points using the [Robot Manager]-[Jog & Teach] page.
4. Run the program by selecting [Run] Window from the [Run] menu or by pressing [F5] (the shortcut key for the [Start] command).

8.1.2 Application layout

Before writing your application, you need to decide what your application will accomplish and how the project will be structured. Here are some general guide lines.

Programs

Each project contains up to 64 programs. Each program that can be started from the Operator Window, Remote Control, RC+ API, or GUI Builder.

Each program has a start function, as shown in the table below.

Programs #	Program Name	Start Function
0	main	main
1	main1	main1
2	main2	main2
3	main3	main3
4	main4	main4
5	main5	main5
6	main6	main6
7	main7	main7
⋮	⋮	⋮
63	main63	main63

Your project must always define function main so that the main program can be started. The other programs are optional. If you use the Operator Window for your operator interface, you can define meaningful names for each of the programs used in your project in [Project]-[Properties]-[Operator setting]-[Operator Window].

Operator interface

- Operator Window

To use [Operator Window] in Epson RC+ 8.0, set Epson RC+ 8.0 to start in operator mode and display the operator window automatically after starting Windows.

Operators can select up to 64 programs. They can also optionally use the Pause/Continue buttons, I/O Monitor, Robot Manager, and System History viewer.

Use the operator window from the PC. The Control Device must be set to PC from [Setup]-[System Configuration]-[Controller]-[Configuration].

For details on configuring Epson RC+ 8.0 for auto start, see below.

Start Mode

- Remote Control

  Use remote control to turn motors on/off, home the robot, start programs, etc. A simple push button box can be used, or a PLC can be connected.

  When using Remote Control, the Control Device must be set to Remote from [Setup]-[System Configuration]-[SPEL Controller Board]-[Configuration].

- Windows Applications using RC+ API

  Use the RC+ API Option along with a Windows development tool such as Visual Basic, Visual C#, or Visual C++. For details, refer to the following manual.

  "RC+ API option"

- GUI Builder

  Use the GUI Builder option. For details, refer to the following manual.

  "GUI Builder"

Safety interface

Use guard doors, safety mats, light curtains, etc. to protect the operator from injury.

Robot Points, Pallets, Tools, Locals

Decide on which points you need for the work cell. In many cases you will only need one point file per robot.

Take advantage of Pallets, Tools, and Locals. Time spent on using these can save hours later on the production line. For example, if your cell has many points that take a lot of time to train, consider using Locals so that if the end effecter is damaged or replaced, you only need to redefine the Locals, not retrain all of the points.

Try to design in automatic or semi-automatic procedures for calibrating tools and locals. Even if you define them manually, write instructions on how to define them so the process can be repeated easily.

Inputs and outputs

Layout your I/O early in the design stages. Use I/O labels in your programs. You must purchase additional I/O boards if you need more than 24 inputs or 16 outputs. You can also use the Fieldbus option so the controller can be a Fieldbus slave.

Peripherals

The Robot Controller has one RS-232C port as standard. Available ports can be added (up to 5 ports) by installing an optional RS-232C board.

See details below.

RS-232C Communications

> ✎ KEY POINTS
>
> T series and VT series manipulator do not have RS-232C port on the controller. Optional RS-232C board cannot be added.

You can use TCP/IP to connect peripheral equipment. See details below.

TCP/IP Communications

8.1.3 Auto start at power up

Your application can automatically log in a Windows user and start you SPEL+ project after Windows boots.

See details below.

Auto Start

8.2 Managing Projects

8.2.1 Overview

What is an Epson RC+ 8.0 Project?

An Epson RC+ 8.0 project is a collection of program files, include files, point files, force files, I/O labels, user errors, Vision settings, and conveyor settings used to run a SPEL+ application.

Why do you need projects?

Projects are a safe and convenient way to manage your SPEL+ applications. All the information for each application is kept in one project. By keeping all of your application code and point definitions in one project, it is easy to open a project and begin running or editing. Also, it is easy to create new versions of an application and run older versions.

Projects make it easier to maintain your application code with less chance of a program being lost.

Previously, multiple commands had to be executed to maintain an application. Creating the application also required a good understanding of SPEL. Without a good understanding, there was a possibility of overwriting and erasing important data during the work.

There are also functions for copying and renaming projects, making it easy to create new projects from previous versions and for backing up projects to an external media such as a USB memory key.

What does an Epson RC+ 8.0 project consist of?

Each project is saved in a folder specified by the customer.

Example: `C:\EpsonRC80\Projects`

✎ KEY POINTS

You can set the save location from the Epson RC+ menu: [Setup]-[Development Preferences]-[Workspace]-[Project Save Location].

The following paragraphs describe the components of a project.

- Project file

  This file contains all of the information that describes the project. This file is automatically created by Epson RC+ 8.0. You should never edit this file. Doing so may cause errors when you open the project. The file extension is ".sprj".

- Program files

  A program file is an ASCII text file that contains one or more SPEL+ functions. Each function in SPEL+ can be run as a separate task (thread) in the controller or called from other functions. The file extension is ".prg".

- Point File

  A point file contains a list of robot points. The file extension is ".pts".

- Include files

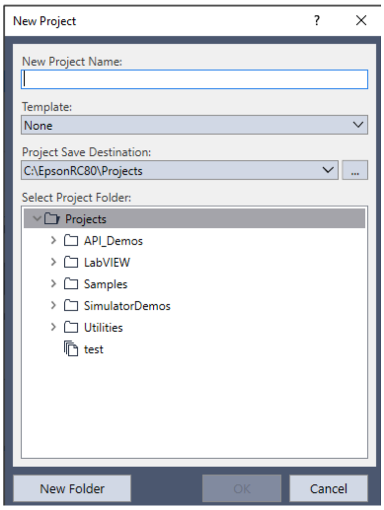
  In the include file, you can declare variables and macros. The file extension is ".inc".

- Force files

  Force files are where force objects are saved. The file name extension is "frc".

8.2.2 Creating a new project

Projects always reside in a folder specified by the customer (example: `\EpsonRC80\Projects`). Also you can create a sub-folder to systematize the projects of different types.



To create a new project

1. In the Epson RC+ 8.0 menu, select [Project]-[New Project]. The [New Project] dialog box will appear.
2. Optionally, select a template to base the project on.

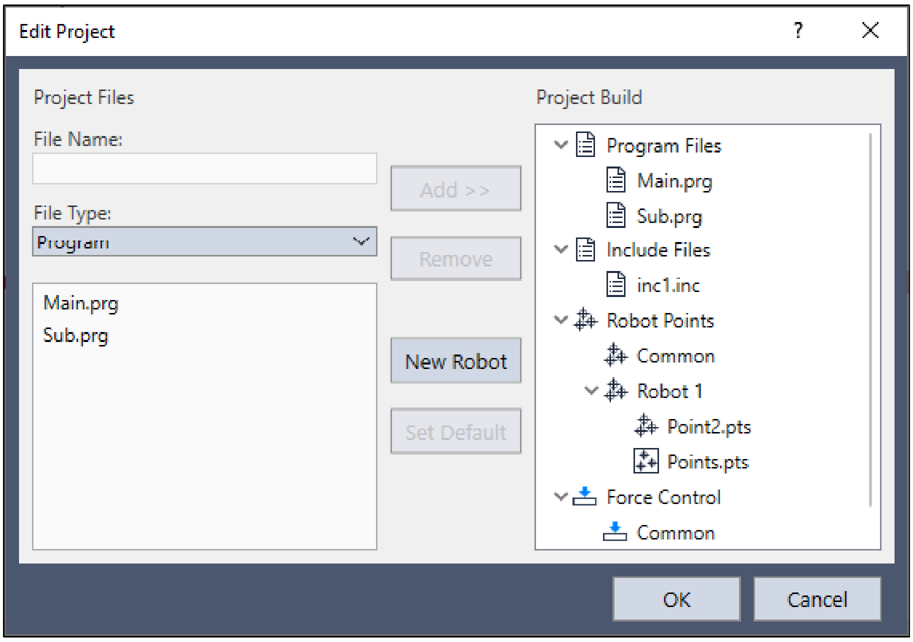
3. Select the location where the project is saved. Or select the parent folder and click the [New Folder] button to create a new folder.
4. Type in the name for the new project.
5. Click the [OK] button to create the project.

8.2.3 Configuring a project

Each application project you create must be configured properly before you can run the program. There are two commands in the [Project] menu that allow you to configure a project: [Edit and Properties].

Editing a project

Select [Edit] from the [Project] Menu to open the [Edit Project] dialog. From this dialog, you can configure program files, include files, point files, and force files used in the current project.



See details below.

[Edit] Command (Project Menu)

8.2.4 Building a project

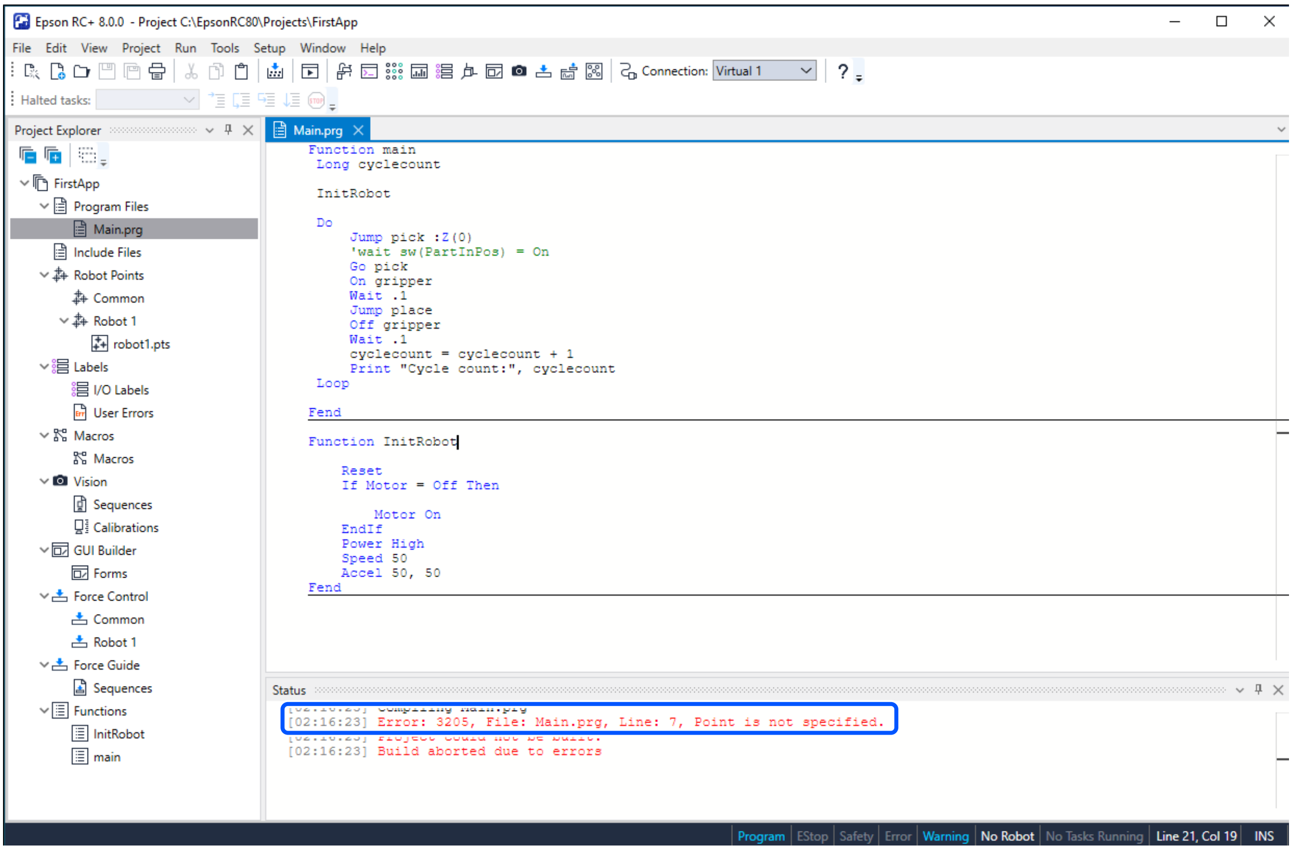
Before you run any program in your application, you must build the project. To build your application project

- Select [Build] from the [Project] menu. Or click the  [Build] button on the toolbar.
- Or Select [Rebuild] from the [Project] menu. This will rebuild the entire project.
- Select [Run] Window from the [Run] menu. Or from the tool bar, click the  [Run Window] button. The project will be built before the [Run] Window appears.
- Select [AUTO mode test] from the [Run] menu. The project will be built before the [Operator Window] appears.

After the files have been compiled and linked, the project files are sent to the controller.

Status Pane

This window shows progress messages and error messages during project build. When errors occur during the build process, a message is displayed that includes the error number, program file name, and line number. Double click on the line with the error to go directly to the source code that caused the error.



8.2.5 Backing up a project

To make a backup copy of the current project, use the [Copy Project] command in the [Project] menu to copy the project to another folder. You can also save the project under a different name.

This is useful for transferring a project to an external medium such as USB memory.

8.3 Editing Programs

Before you can edit a program, it must be in the current project and opened in a program window.

To open a program for editing

1. Select [Open] from the [File] menu.
2. Select the file(s) you want to open.
3. Click the [Open] button to display the file.

8.3.1 Program rules

A program contains one or more SPEL+ function definitions.

Lines can be blank. You can insert any number of blank lines to separate subroutines and functions, if desired.

The maximum length for each line is 512 characters.

8.3.2 Typing in program code

You can enter program statements in upper or lower case. Whenever you leave a line that has been changed, the line will be formatted. SPEL+ keywords are case formatted and spaces are inserted around operators and after semicolons.

Consider using mixed case or lower case for variables and function names instead of all CAPs. This will make your code easier to read.

Use indentation for statements within loops. The "Auto Indent" feature automatically moves the cursor under the start of the previous line. It also indents lines after If, Else, For, Select, Case, and Do statements.

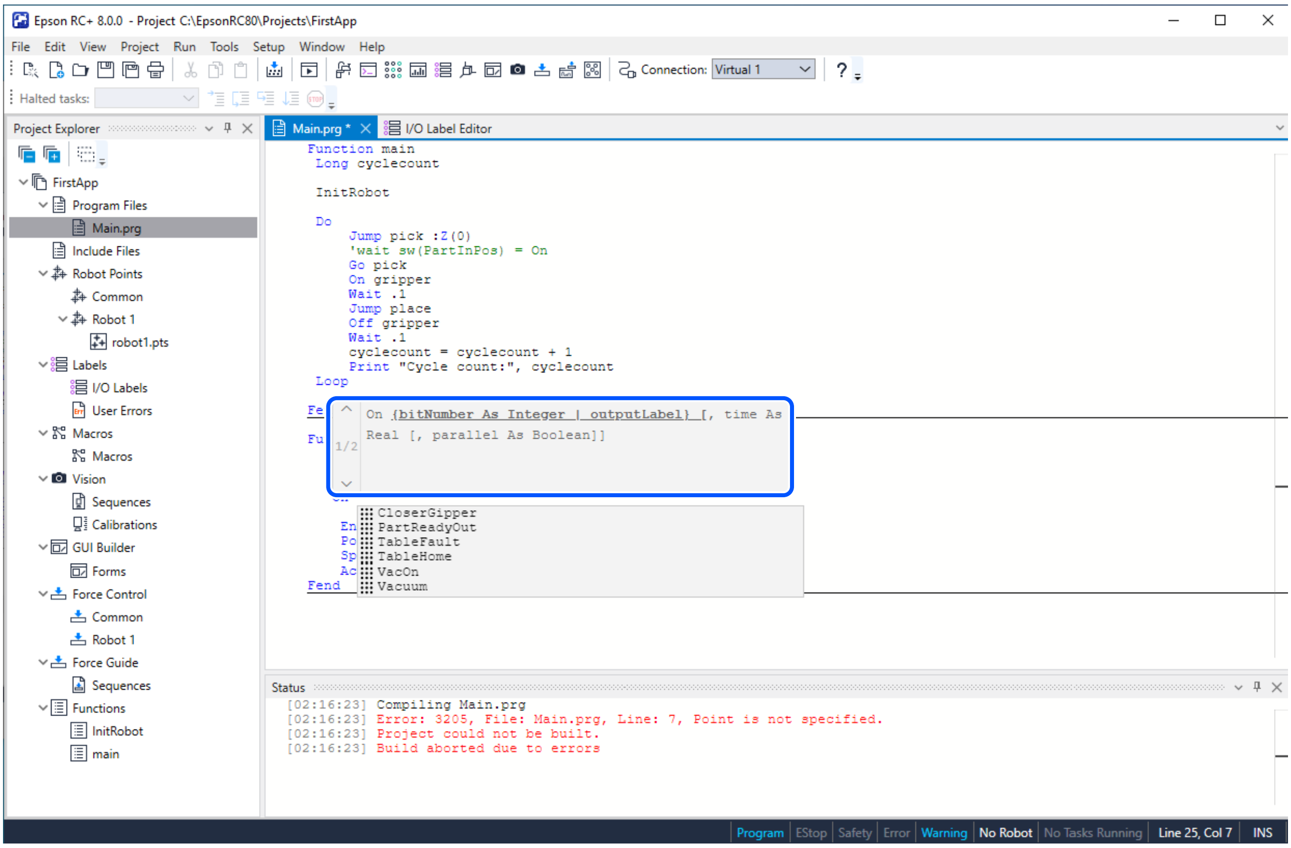
```
For i = 1 To 10
  Jump P(i)
  Jump P0
Next i
```

8.3.3 Syntax helpand code completion

Epson RC+ 8.0 includes a feature to assist in entering program code.

Syntax Help

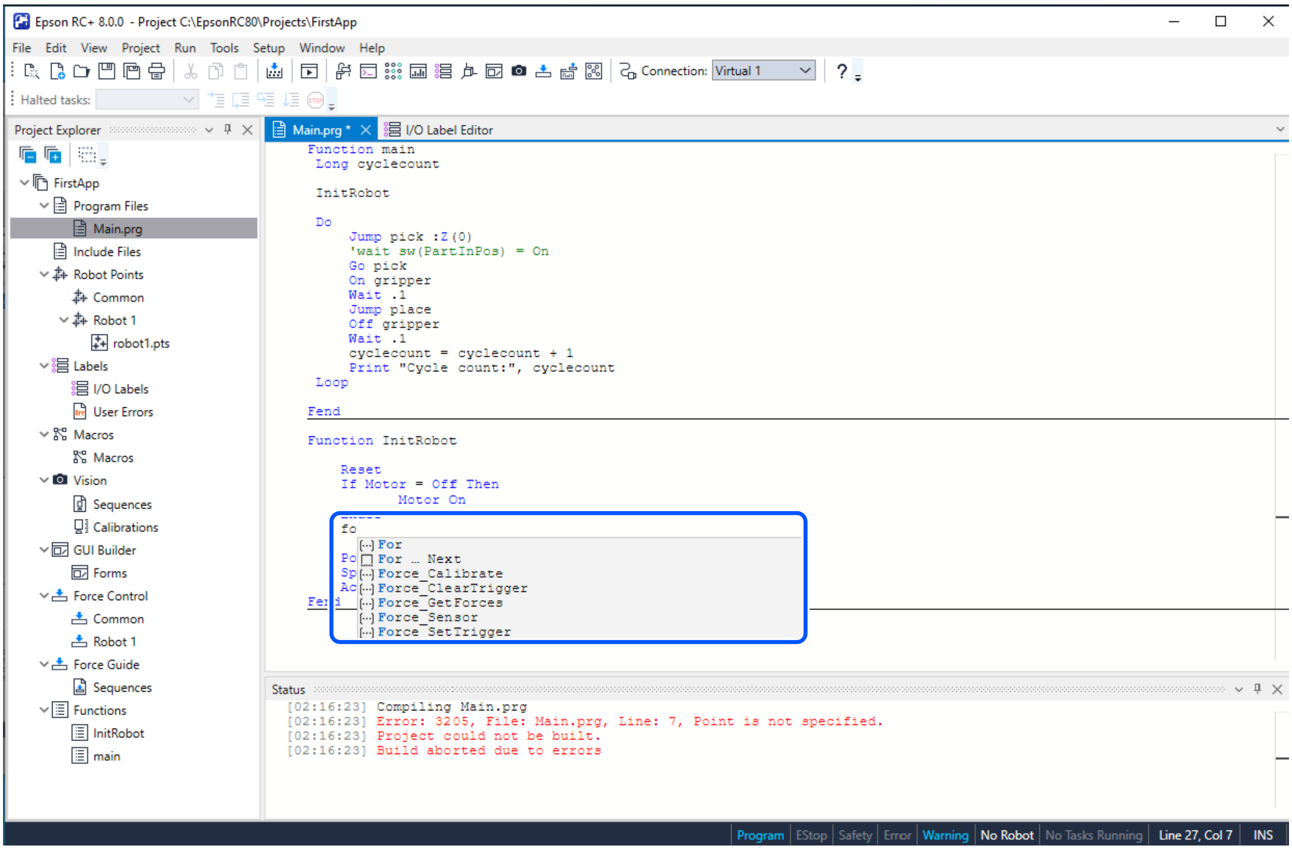
When you enter a SPEL+ keyword, the Syntax Help window will show the syntax of the statement or function. After a statement is entered, Syntax Help will automatically close. Or press the [Esc] key to close it. You can enable / disable Syntax Help from the [Setup]-[Preferences]-[Editor] tab.



Code Completion

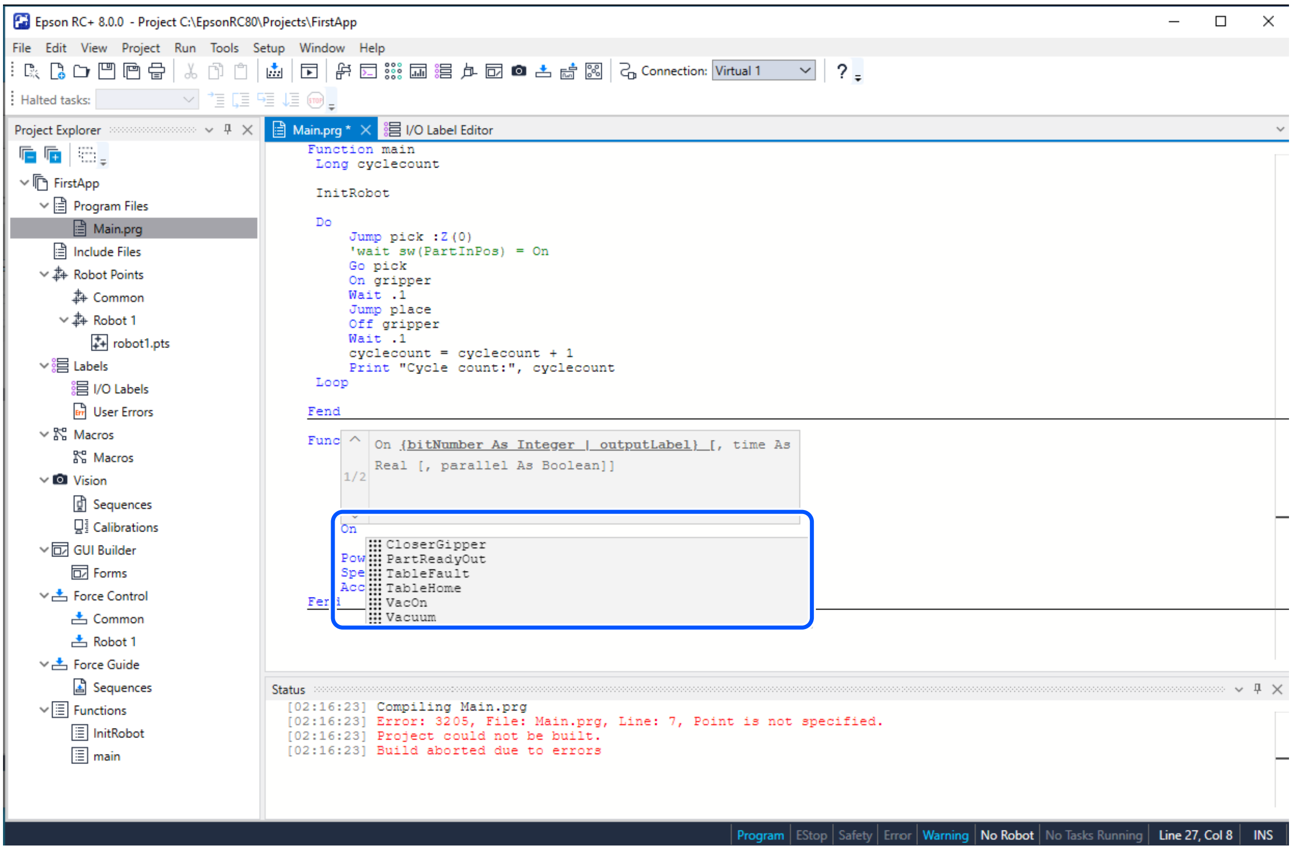
When you are entering a keyword for a statement, a list box appears, showing the available SPEL+ statement keywords, constants, and user functions that start with the text you have typed. To use a keyword in the list box, select the keyword.

For example, typing "fo" will display a list of candidates. If you select [For...Next], a Next statement is automatically created with an indented blank line above it.



If you enter a command that requires parameters to be specified, a parameter list box will be displayed. To use a parameter in the list box, select the parameter. You can also type in a value not shown in the list, such as a variable or literal constant. Press [Esc] to hide the list box.

In the example shown below, the first parameter of the On statement is an output label. A list of output labels in the current project is displayed.

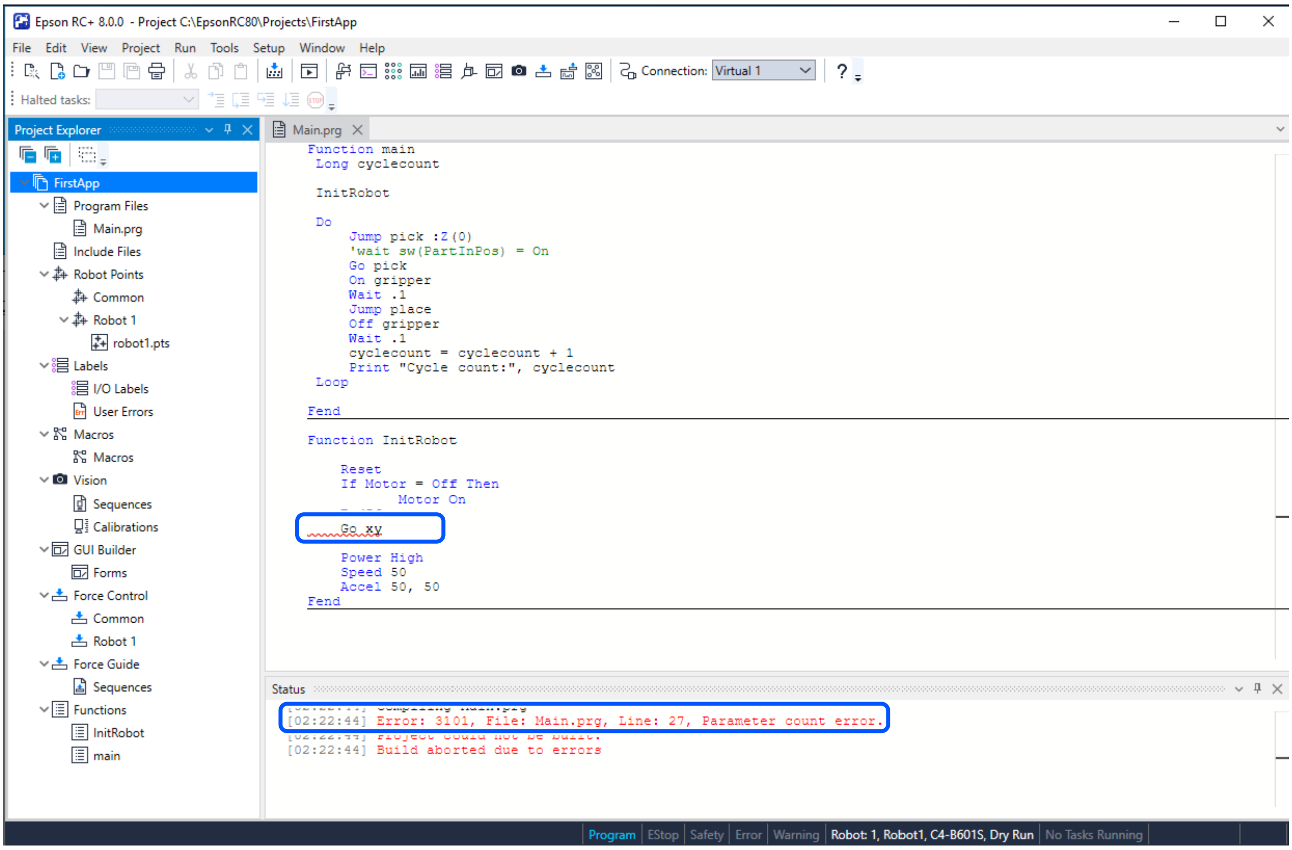


You can enable/disable Code Completion from the [Setup]-[Preferences]-[Editor] tab.

8.3.4 Syntax errors

When a syntax error is detected, a red wavy line will appear under the line containing the error. If a build is executed, a message will appear on the status bar.

For example, in the program shown below, the message "Parameter count error" is displayed on the status bar.



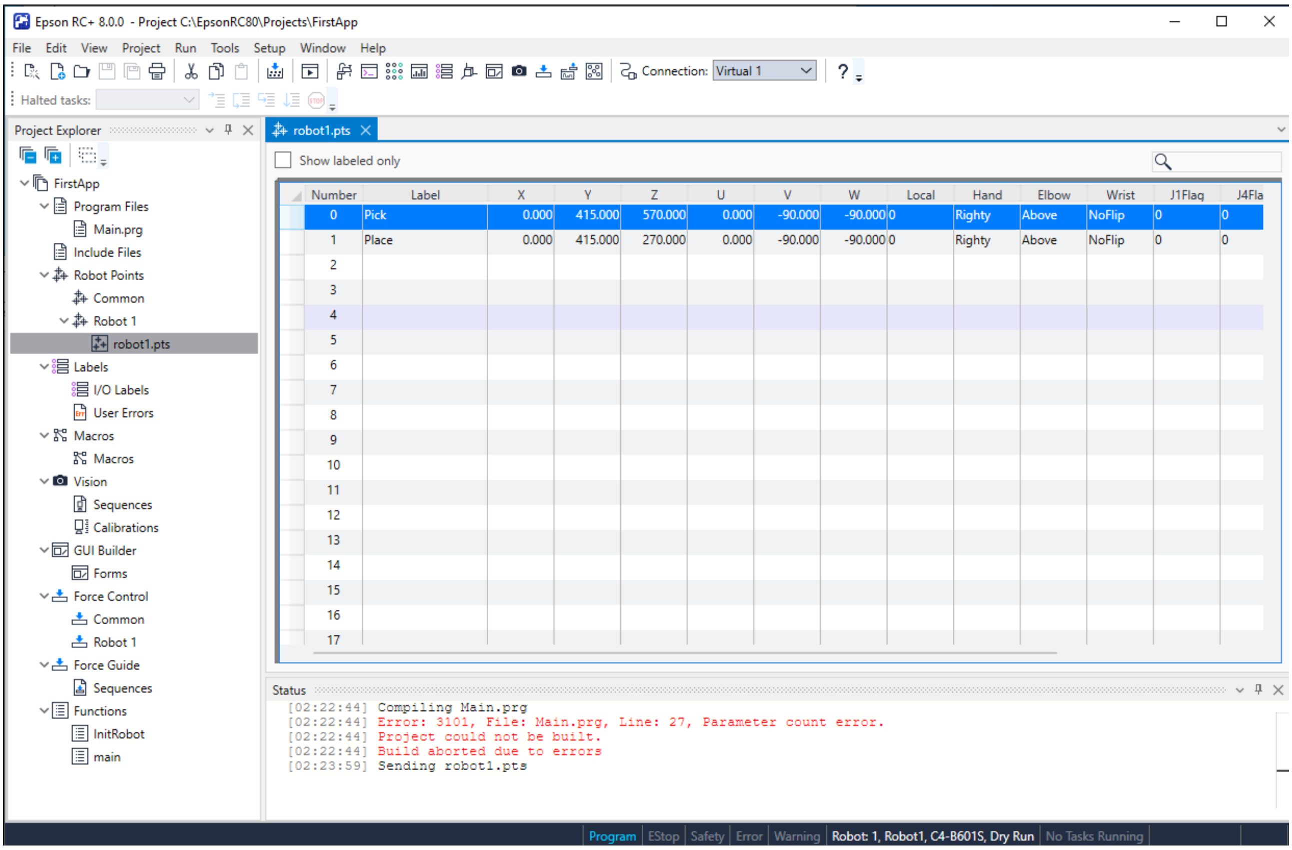
8.4 Editing Points

You can edit the robot points from the robot point file. You can define new points or cut, copy, and paste points from one point file to another, including between projects.

To open a point file for editing

1. Select [Open] from the [File] menu to show the Open dialog box.
2. Select the [Points] option button. A list of files is displayed in the [Select file to open] box.
3. Select the point file you want to edit by clicking on the name.
4. Click [Open] to open the file. You will see a spread sheet window for the point file you selected.

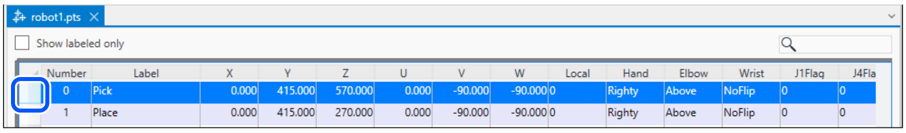
A list of each point in the file is displayed. By default, the spreadsheet always contains rows for all points, even if they are not defined. The cells for an undefined point are blank.



Item	Description
Show defined only	Only I/Os with defined labels are displayed.
Select row	Click the left edge of row to select a row.
No.	Point number. Range is from 0 to a maximum point number.
Label	Point label name.
Coordinates	Coordinates of X, Y, Z are in millimeters, and U, V, and W in degrees.
Local	Local number. Range is from "0" to "15".
Hand	Select hand orientation. Lefty and Righty.
Elbow	Select elbow orientation (Above, Below). This column is shown only for 6-axis robots.
Wrist	Select wrist orientation (Flip, NoFlip). This column is shown only for 6-axis robots.
J4Flag	Select J4Flag (0, 1). This column is shown only for 6-axis robots.
J6Flag	Select J6Flag (0 to 127). This column is shown only for 6-axis robots.
J1Flag	Select J1Flag (0, 1). This column is shown only for RS series and 6-axis robots.
J2Flag	Select J2Flag (0, 1). This column is shown only for RS series.
J1Angle	J1Angle coordinates are in units of degrees. This column is shown only for RS series and N series.
J4Angle	J4Angle coordinates are in units of degrees. This column is shown only for N series.
Comment	A description of the point.

To select a row

Click the left edge of row to select a row.



To select one or more rows

To select more than one row, point to the row select column of the first row you want to select. Hold down the left mouse button and drag the mouse down or up to select more rows. While holding down the [Shift] or [Ctrl] key, you can select multiple rows with the mouse.

To select all rows

Select [Select All] from the [Edit] menu or press the [Ctrl] + [A] keys.

To define a new point

Move the cursor anywhere in the row of the point you want to define using the mouse, and then click a cell you want to type in. Enter information for the point. This automatically defines the point, which means it will be sent to the robot controller at the next project build or Jog and Teach command.

For example, click on the Name column and type in a name of the point. Press the [Tab] key to move to the X coordinate column. Type a coordinate value and then press [Enter]. You will see zeros automatically entered in all of the other coordinates. This means that the point is defined.

To delete a point

1. Select one or more rows.
2. Delete a point using one of the following methods.
   - In the Epson RC+ 8.0 menu, select [Cut] from the [Edit] menu.
   - Right-click on the left end of the row and select [Cut].
   - Press [Ctrl] + [X] or [Delete].

To [Cut] and [Paste] points

1. Select one or more rows.
2. Cut or copy using one of the following methods.
   - In the Epson RC+ 8.0 menu, select [Cut] or [Copy] from the [Edit] menu.
   - Right-click on the left end of the row and select [Cut] or [Copy].
   - Press [Ctrl] + [X] or [Ctrl] + [C].
3. Select the row where you want to start the paste.
4. Paste a point using one of the following methods.
   - In the Epson RC+ 8.0 menu, select [Paste] from the [Edit] menu.
   - Right-click on the left end of the row and select [Paste].
   - Press [Ctrl] + [V].

8.5 Running and Debugging Programs

You can run programs from the Run Window or from the Operator Window. The Run Window is used primarily for testing and debugging. The Operator Window is used as an operator interface for simple applications or demos.

You can also run programs using the RC+ API option or GUI Builder option.

To run a program

Select [Run] Window from the [Run] menu. This command will build the project (if required) and open the [Run] Window. The [Run] Window allows you to choose which function to execute. Select a function and then click the [Start] button.

8.5.1 Run window

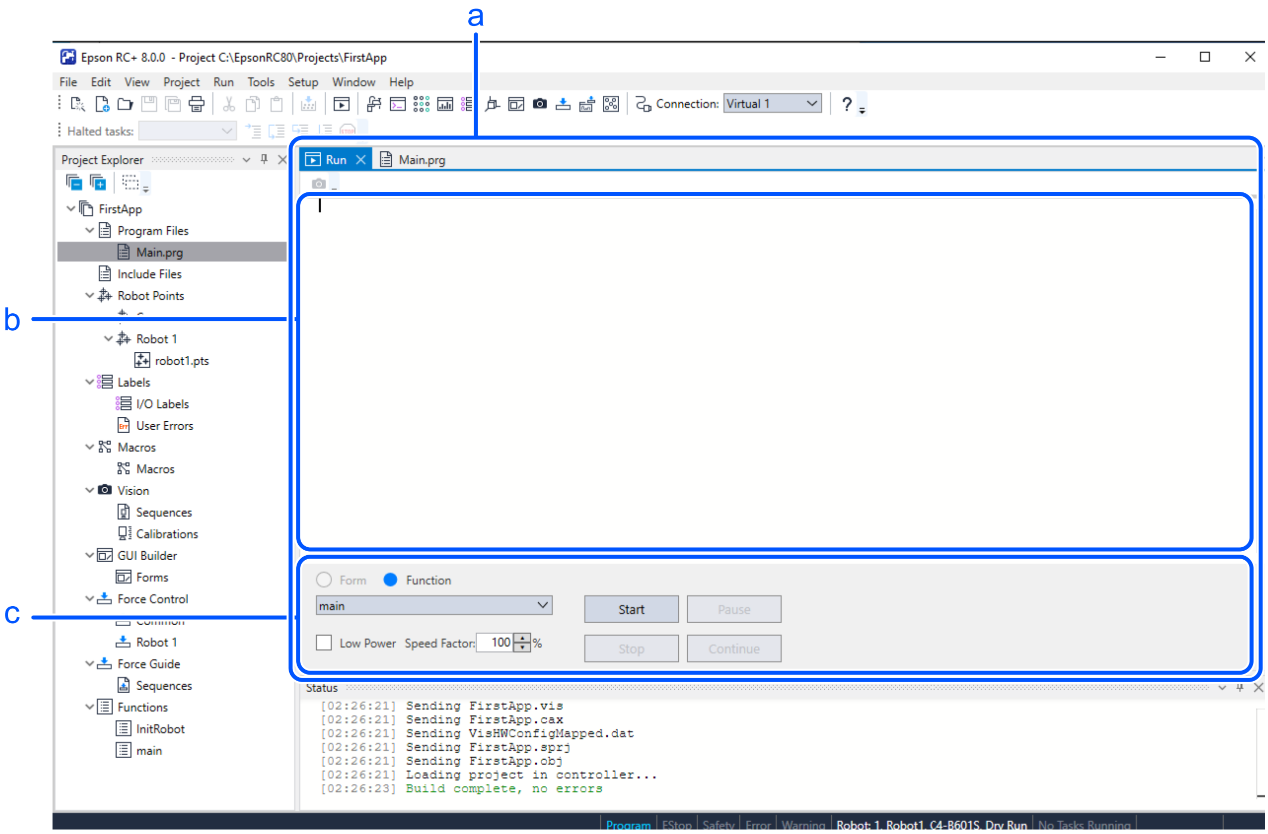
The Run window includes controls for running the programs in the current project.

To open the Run window

Select [Run Window] from the [Run] menu. Or from the tool bar, click the  [Run Window] button. If necessary, all changed open files will be saved and the project will be built. If the build is successful, the Run window will appear.

To close the Run window

Choose [Close] from the [File] menu or click on the [×] button in the upper right hand corner of the window.



Symbol	Item
a	Run window
b	Text area
c	Operation area

Item	Description
Text area	This is the area that takes up most of the run window. Output from your programs is displayed here. When your program uses an Input statement, you can type in the requested input from this text box. You can use the scroll bars to view the entire text buffer. If an error occurs while running a program, the error number, program file name, line number and function name will be displayed in this text area. You can double click on the line where the error is displayed to directly go to the source line that caused the problem.
Function	Select a function to start. Functions are sorted alphabetically. Function main is selected by default.
Low Power	During this box is checked, SPEL+ ignores the Power High command by force. This allows you to run your program in low power mode to verify operation without having to change the program.
Speed Factor	Specifies the robot motion speed factor. The speed factor is a percentage of maximum point to point speed and linear interpolated speed. For example, if you program executes Speed 80 and the speed factor is 50%, the robot will move at speed 40.
Start	Starts the function displayed in the [Function] list.
Stop	Stops all tasks. If the robot is executing a motion command when this button is pressed, the robot will decelerate to a stop. The shortcut key is [Ctrl] + [C].
Pause	Pauses all tasks with pause enabled. Activates the [Continue] button. If the robot is executing a motion command when this button is pressed, the robot will decelerate to a stop.
Continue	Continues paused tasks.

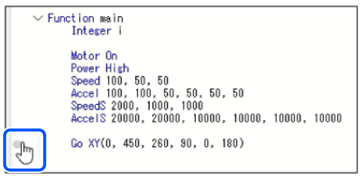
8.5.2 Debugging

Epson RC+ 8.0 supports source level debugging. You can set breakpoints and step through your source code. You can also pause / continue a program or halt a task using the Task Manager.

Setting and clearing breakpoints

Open the program where you want to set a breakpoint, and then click on the line where you want to stop. Use one of the following methods to set a breakpoint:

- When the mouse hovers over the left edge of a line (row), a gray circle appears. When clicked, it changes to a breakpoint icon.



- Press the [F9] key.

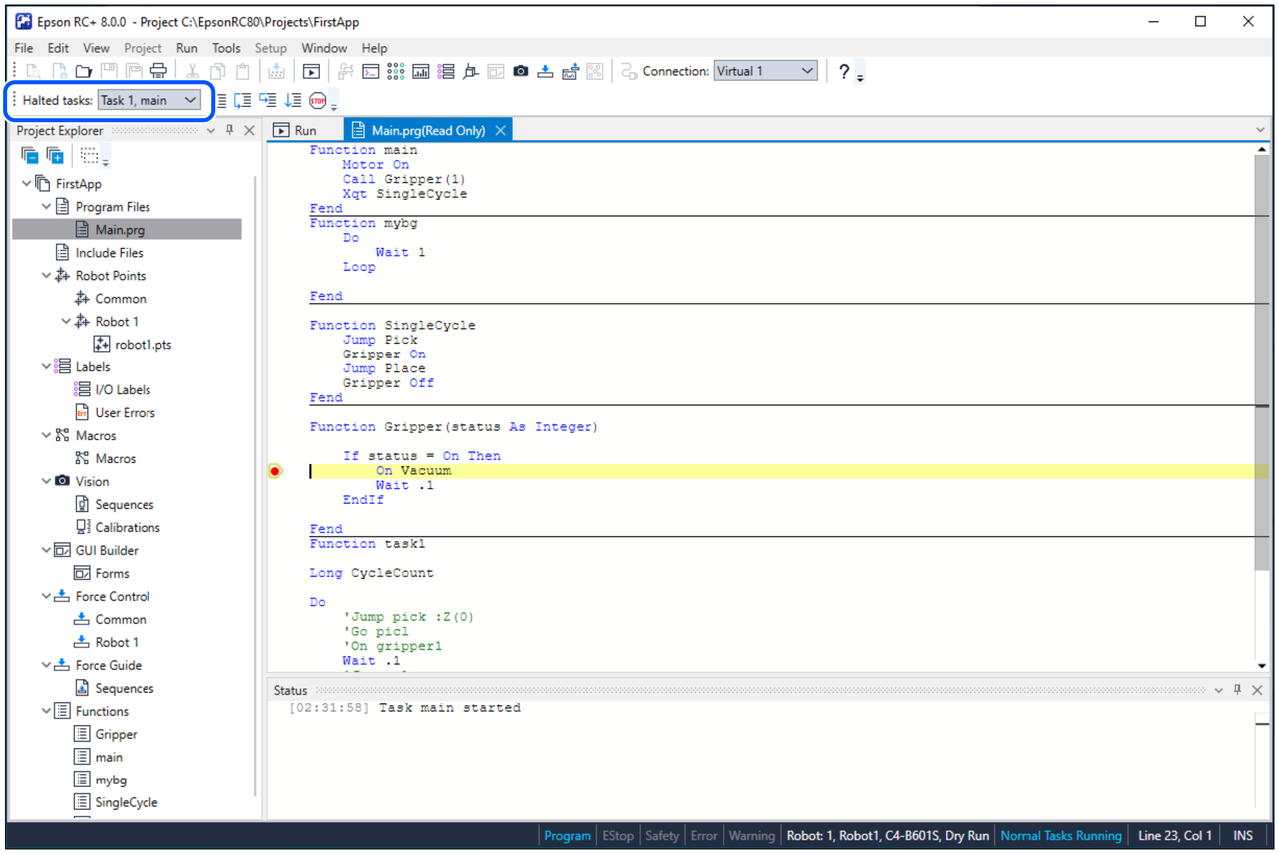
- Epson RC+ 8.0 menu - [Run] - [Toggle Breakpoint]

Breakpoints are cleared when setup is performed again on lines where the breakpoints were set. To clear all breakpoints, select [Clear All Breakpoints] from the [Run] menu.

You cannot set a breakpoint on non-executing statements, such as #define, #include, or blank lines.

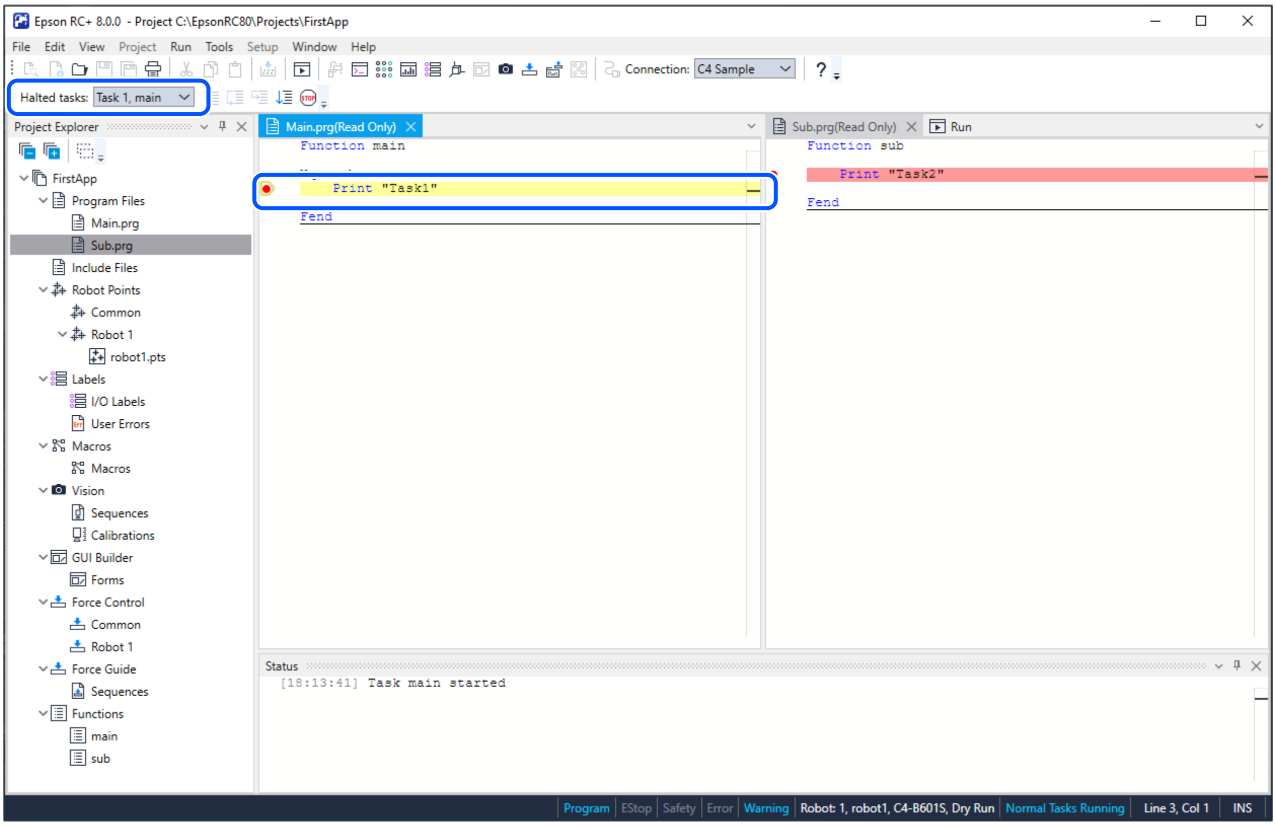
After setting a breakpoint, the task will halt when the execution line is reached the breakpoint. You can set or clear a breakpoint while a task is running.

When reached a breakpoint, the program window containing the program source line at the breakpoint is opened and the line is highlighted in yellow. Task numbers are displayed in the [Halted tasks] menu.

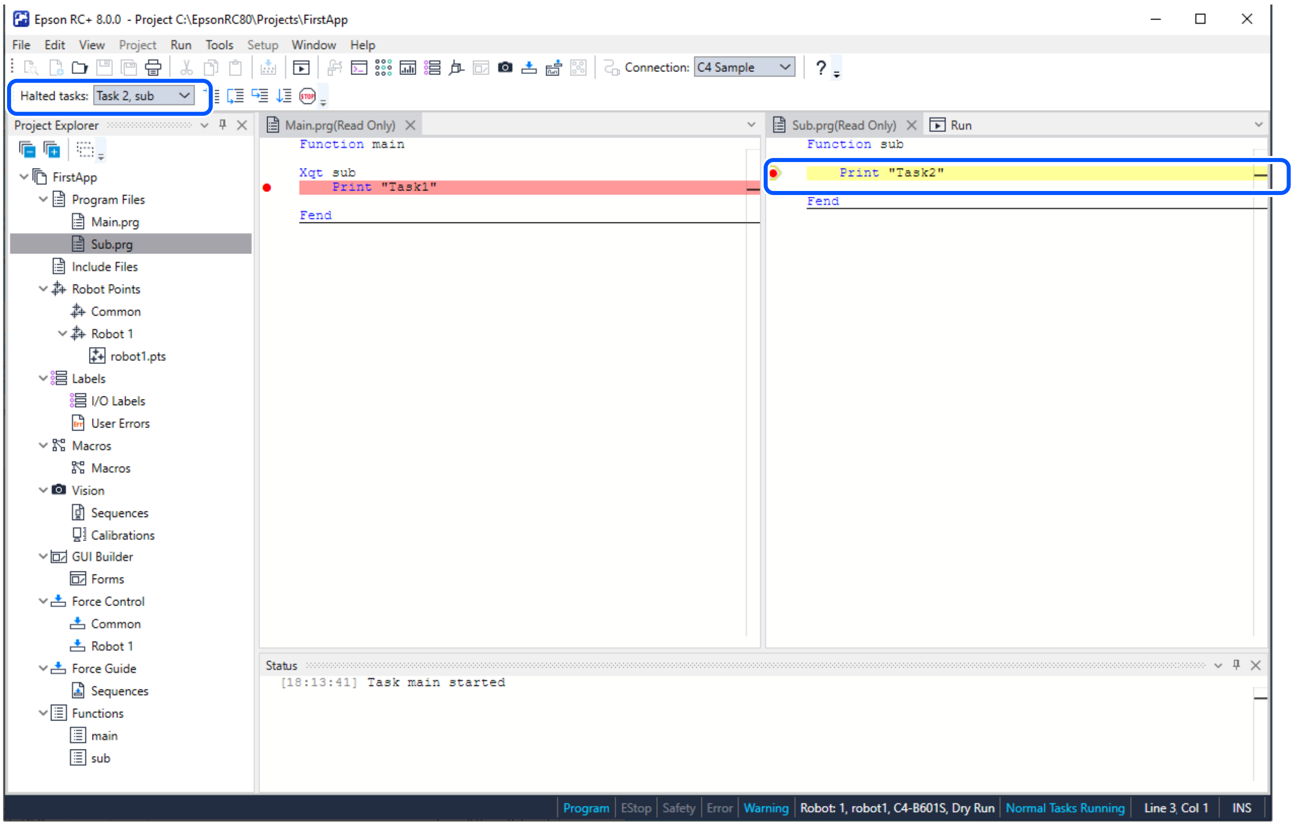


When multiple tasks have reached a breakpoint, you can confirm each task from [Halted tasks]. When you switch tasks in [Halted tasks], the location where the breakpoint stopped is displayed.

If you select task 1 from [Halted tasks] :



If you select task 2 from [Halted tasks] :



Stepping through a program

There are three commands on the [Run] Menu that are used for stepping through code.

- [Step Into] steps through each line and also steps into functions when a step is executed on a Call statement.

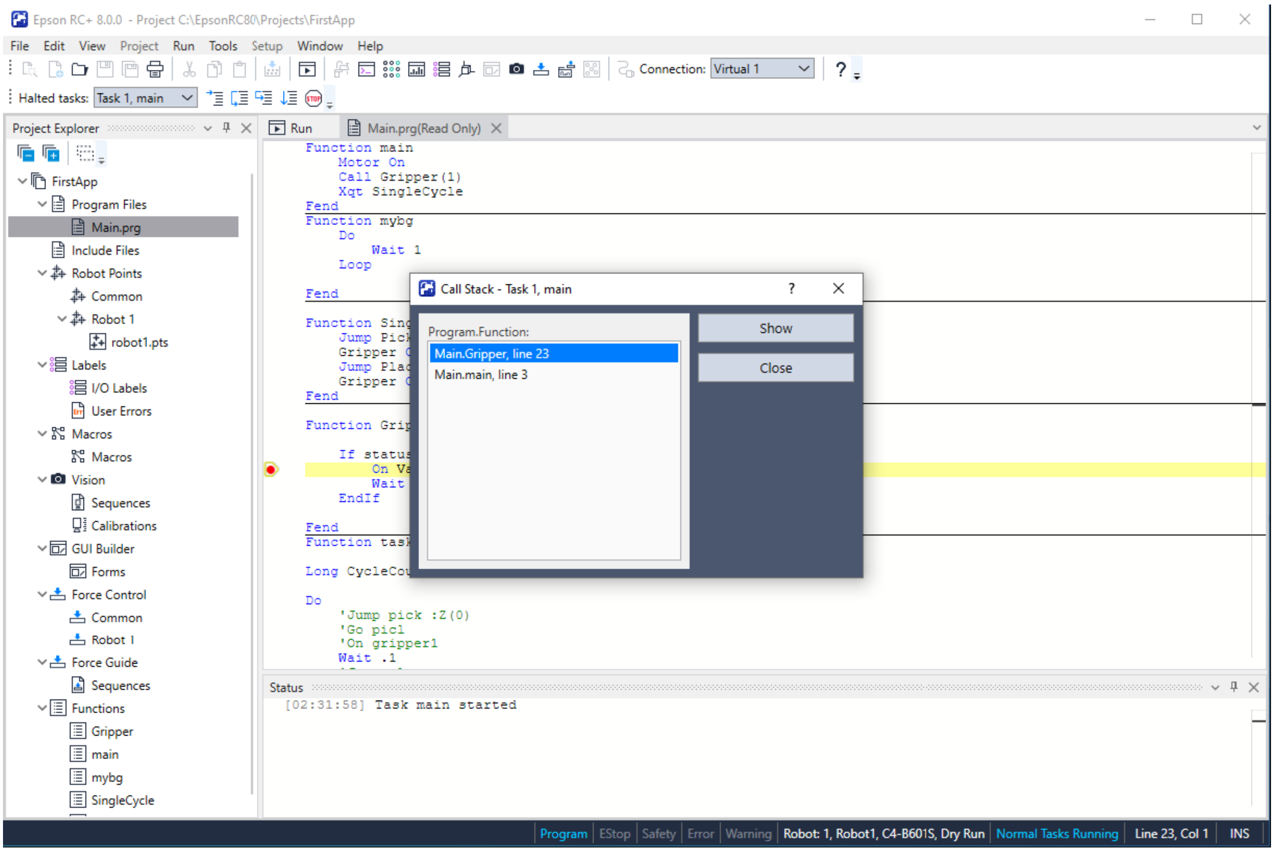
- [Step Over] steps through each line but when a Call statement is encountered, the function in the statement is executed completely.
- Walk executes lines until after the next motion command and then halts the task. It will halt after the next output command if the [Setup]-[System Configuration]-[Controller]-[Preferences]-[Walk stops for outputs] checkbox is checked.

To step through code, you must set a breakpoint and run until the breakpoint is reached, or suspend a task from the Task Manager using the [Halt] button.

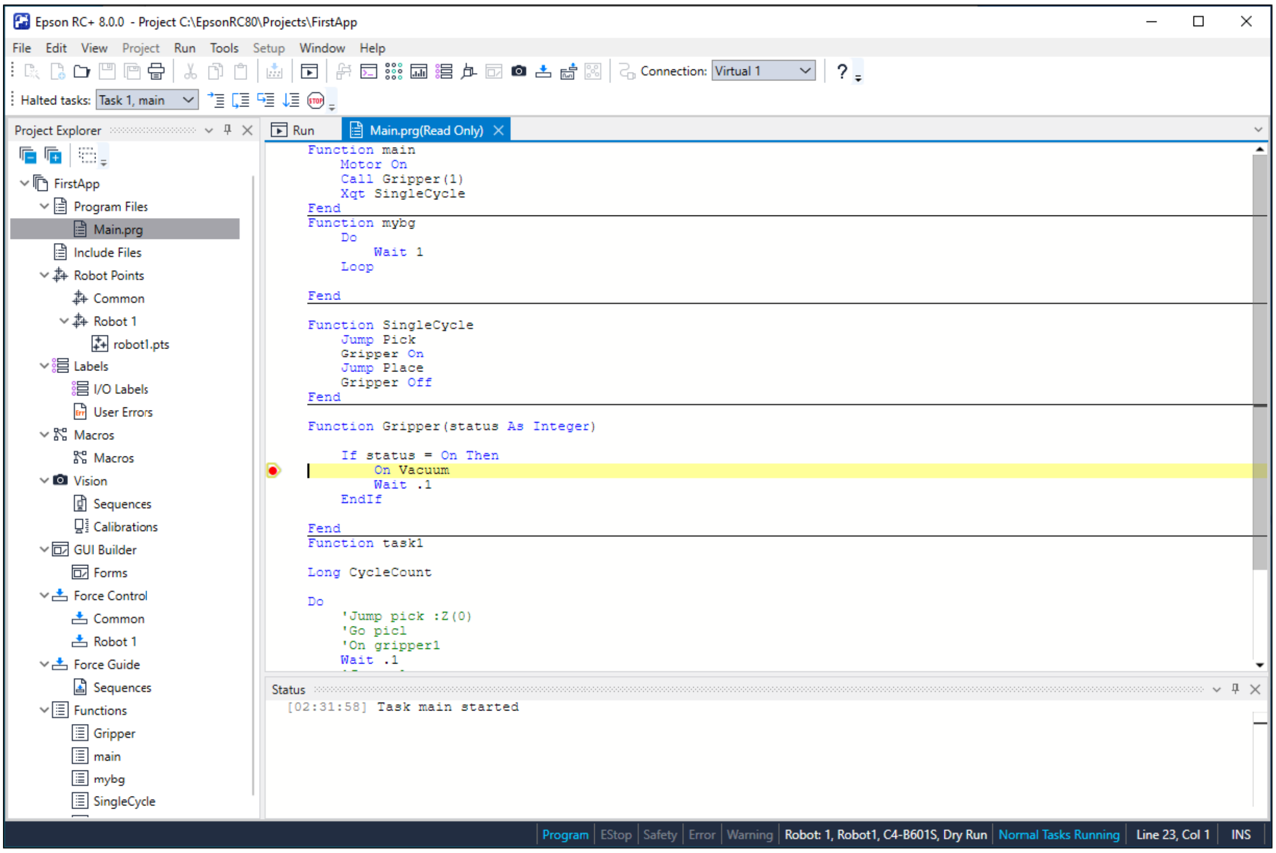
Viewing the Call Stack

Sometimes you may want to examine the call stack for the current task after you halt the task from the task manager, or reach a break point.

To view the call stack, select [Call Stack] from the [Run] Menu. The [Call Stack] list will be displayed, as shown below.



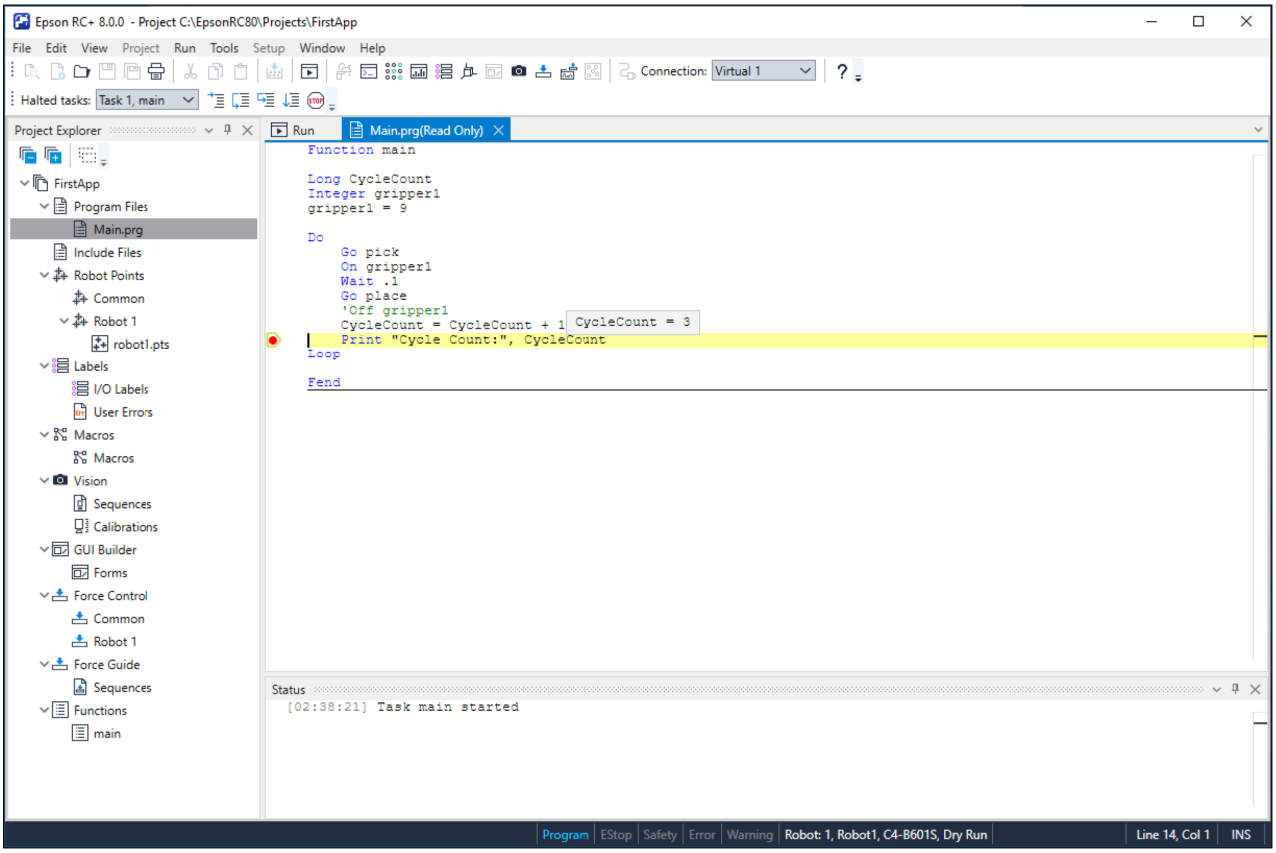
After you double click a function in the Call Stack list, the function will be displayed in a program window. The cursor moves to the line in the call stack where the next called function is located. In the example below, the SingleCycle function points to the Gripper On statement, indicating that Gripper was called from SingleCycle.



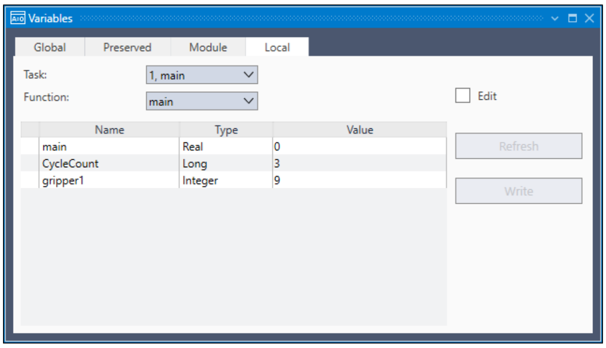
Displaying variables

To view variable values, you can do one of the following:

1. When a task is halted by halt or breakpoint, you can view the value for a variable by moving the mouse cursor over the variable name. The value will be displayed in a tool tip type window above the variable name.



2. Select [Display Variables] from the [Run] menu to display the variable display dialog. This dialog has the following tabs: [Global], [Preserved], [Module], and [Local].



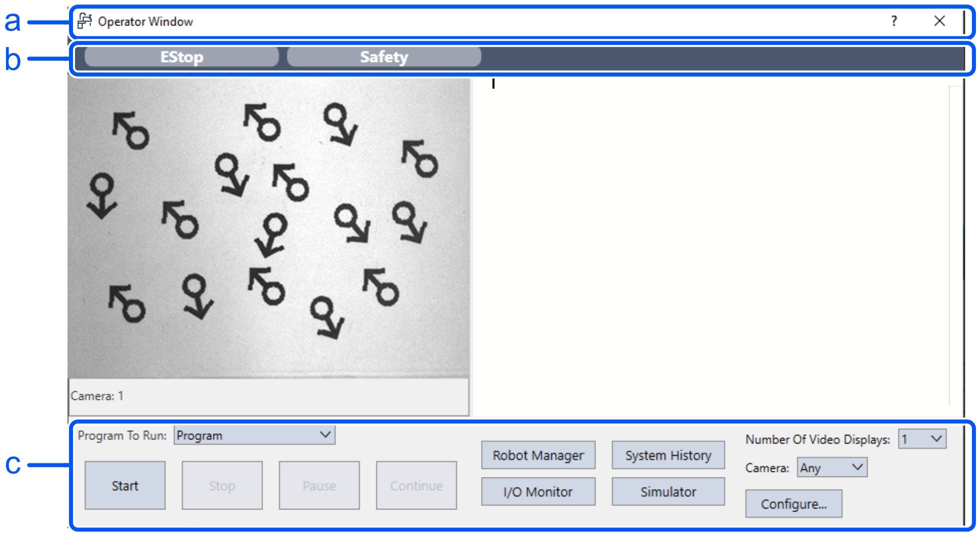
KEY POINTS

Up to 600 variables can be displayed on each tab.

You can change the value of a variable by checking the [Edit] checkbox. Enter the new value in the [Value] column. Next, click the [Write] button to change the variable. When the [Edit] box is checked, the variable values are not automatically updated. You can update all values by clicking the [Refresh] button.

8.6 Operator Window

The Operator Window can be used as a simple interface for operators. You can configure Epson RC+ 8.0 to open only the Operator Window when started. In addition, when Remote Control is being used, the Operator Window can be displayed for monitoring purposes.

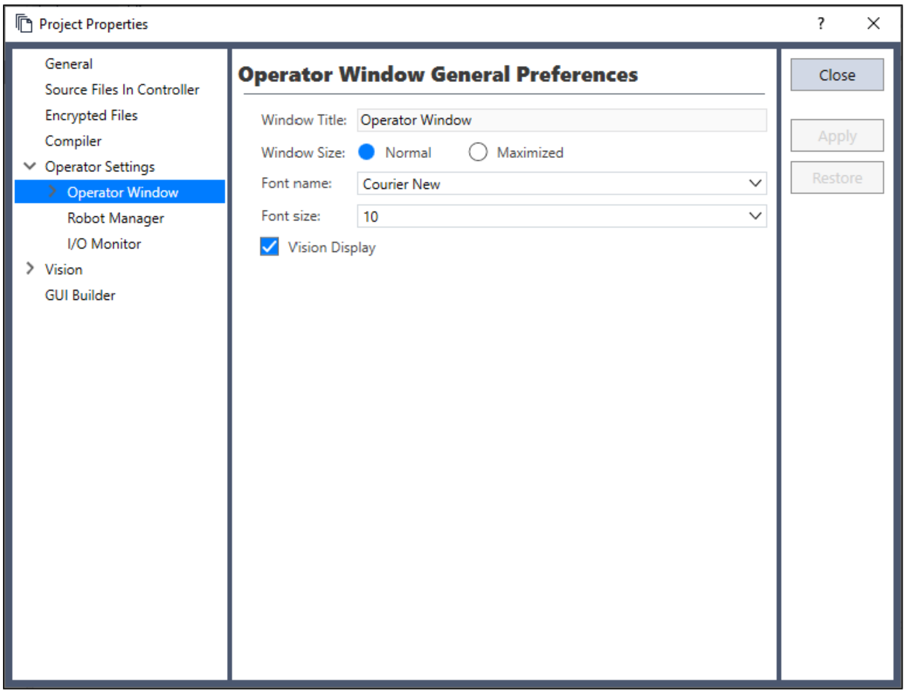


Symbol	Item
a	Title
b	Status Bar
c	Operator Button

Item	Description
Status Bar	The status bar is located at the top of the window and shows emergency stop and safeguard status. In addition, if a warning is detected from the controller (such as low encoder battery), a warning label will be displayed on the right side of the status bar. If the mouse is over this label, you can see the warning error message. When there is no warning, the warning label is hidden.
Program to Run	Select a program to run.
Start	Starts the selected program.
Stop	Stops all tasks.
Pause	Pauses all tasks that are enabled for pause.
Continue	Continues paused tasks.
Robot Manager	Display the [Robot Manager] dialog in operator mode. It cannot be shown while the program is running.
I/O Monitor	Opens the I/O Monitor in operator mode. This window can remain open while programs are running.
System History	Display the [System History] window. This window can remain open while programs are running.
Simulator	Opens the [Simulator] window. This window can remain open while programs are running.
Number of Displays	Sets the number of displays to be shown.
Camera	Project cameras appear in a drop-down list.
Configure	Opens the [Configure Video Displays] window. You can set the window to be displayed as the main window.

8.6.1 Operator window configuration

You can configure the Operator Window from the Operator Window pages in [Project]-[Properties].



There are several settings for operator Robot Manager and I/O Monitor.

See details below.

[Properties] Command (Project Menu)

8.6.2 Auto start configuration

You can configure the system to let it log into Windows automatically. Also you can configure a program to start automatically from the [Operator] window.

See details below.

Auto Start

8.7 Using Remote Control

You can design your application to be run from external equipment using hardware I/O control. This includes push button boxes, PLCs, and other PC systems.

See details below.

Remote Control

8.8 Using Encrypt Files

Encrypted files allow you to prevent end users from viewing your source code. When a file is encrypted, you must supply a password to open the file. Other users cannot view the file contents, even with an external editor, such as Notepad.

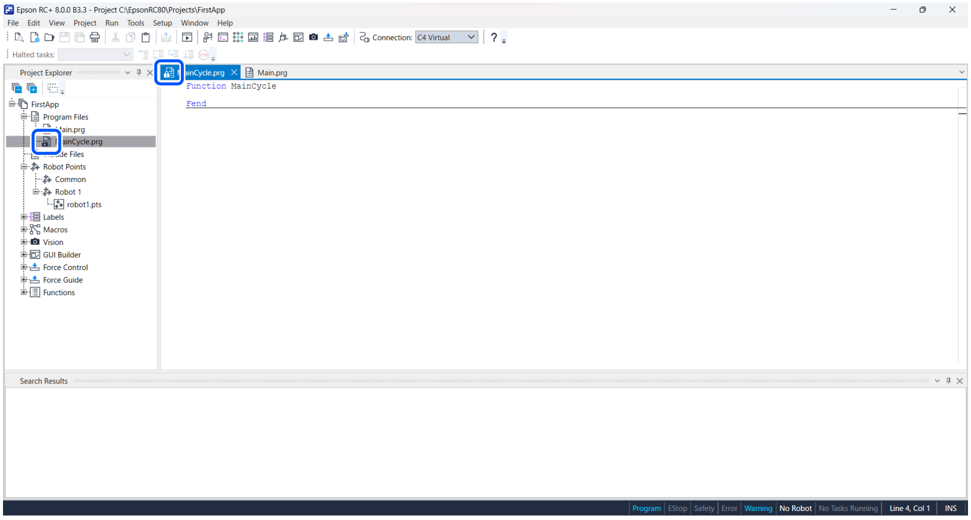
Each encrypted file can have its own password, or you can choose to encrypt multiple files with one password. You can encrypt program files, include files, Vision Guide, and GUI Builder.

If an encrypted file is imported from another project, it will remain encrypted in the current project.

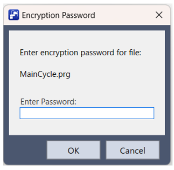
As an example, assume you have some special SPEL+ programming code that you do not want your end users to view. But you want to allow end users to change some of the code in the project. To do this, put all of the functions you want to be hidden in one or more encrypted program and include files. Portions that can be modified by the end user should be saved as a separate program file and not encrypted. As a result, the end user will only be able to modify some programs that are not encrypted.

When files are encrypted, their icons are shown with a lock in the Project Explorer and also in the title bar of the program window.

In the screenshot below, the file MainCycle.prg is encrypted, so its icons include a lock image.



When you open an encrypted file, you will be prompted for the password.



KEY POINTS

Up to 64 characters can be entered for password.

CAUTION

USE EXTREME CAUTION!

Keep a record of the password(s) used for encryption in a safe place. Once a file is encrypted, it can only be opened with the password you enter. If you forget the password, the file contents CANNOT BE RECOVERED.

To configure encrypted files in your project, select [Project Properties] from the project menu, then select [Encrypted Files] in the tree on the left.

See details below.

[Properties] Command (Project Menu)」

9. Simulator

9.1 Simulator Functions

Simulator functions enable easy robot motion checking on your PC, which gives you flexibility to consider the system layout, measure the operation time, and create robot programs.

They are useful from the introduction stage to the launch of robot system.

9.1.1 Overview

The following are the major simulator functions: Robot motion 3D display

Shows robot orientation and motion in a 3D display from various viewpoints. Offers accurate display data based on design data.

This function is not available for the manipulator series (models) as described below. If a simplified layout consideration or measurement of motion time is needed, set an alternative model. However, external dimensions and motion range may differ. Be careful when you set the alternative models.

For details on unsupported models, see below.

Appendix C: Simulator Functions List of Unsupported Manipulator Models

Series	Model	Alternative model (When using a Virtual Controller)	
X5	All models	No alternative model	-
G6-II	Protected-model G6-***D*-II, G6-***P*-II	Standard model, Cleanroom model G6-***S*-II, G6-***C*-II	*
G10-II	Protected-model G10-***D*-II, G10-***P*-II	Standard model, Cleanroom model G10-***S*-II, G10-***C*-II	*
G20-II	Protected-model G20-***D*-II, G20-***P*-II	Standard model, Cleanroom model G20-***S*-II, G20-***C*-II	*

*: External dimensions and motion range differ.

Interference check

Checks whether the robot (including the hand and devices installed on the robot) interferes with itself or its peripherals. (This function is not available for the robots that do not support 3D display.)

Robot operation time prediction

Predicts the robot operation time for a program.

Considers the speed setting (Speed, etc.) and acceleration / deceleration setting (Accel, etc.) when predicting the robot motion time.

SPEL+ program execution

Allows you to create, execute, and debug SPEL+ programs.

For restrictions on the simulator functions, see below.

Simulator Specifications and Restrictions

9.2 Using the Simulator

You can try the simulation functions using the provided sample virtual controllers and projects.

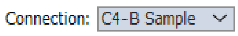
9.2.1 Working with the samples

You can operate a robot easily using the samples provided.

Follow the steps below:

- Connect with a sample virtual controller (robot)
- Open a project
- Display the [Robot Simulator] window
- Operate the robot by executing the program
- Next step

1. Connect with the virtual controller

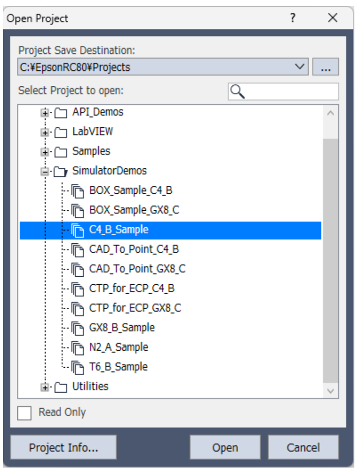


Select "C4-B Sample" from the Epson RC+ 8.0 toolbar [Connection] list box. When the connection is complete, the [Connection] list box shows "C4-B Sample".

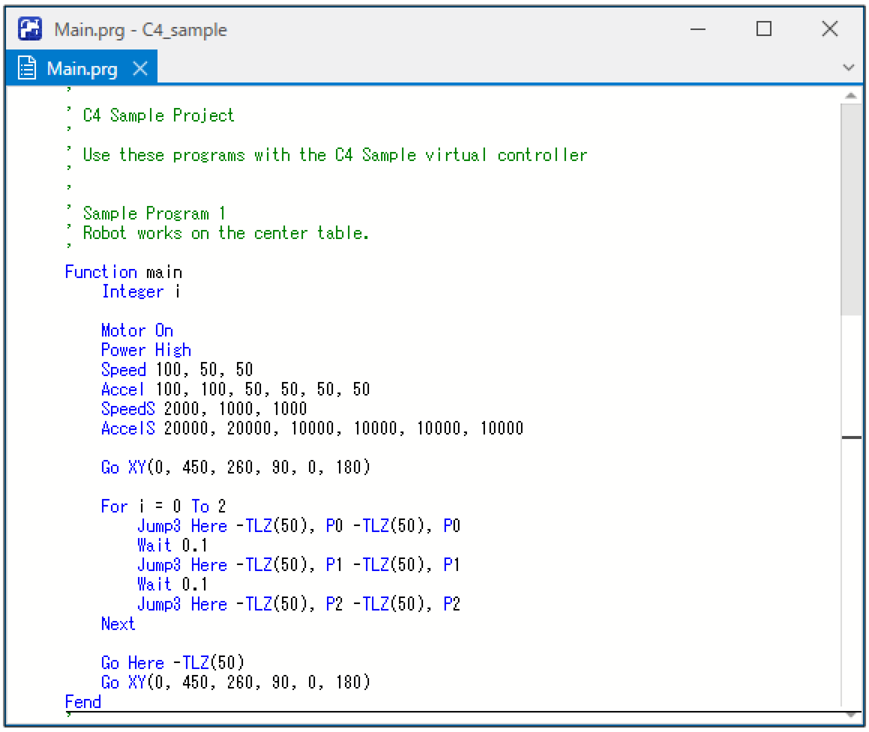
2. Open a project

   i. Click [Open...] from [Project] on Epson RC+ 8.0 menu.

   ii. Select [Projects]-[SimulatorDemos]-[C4_B_Sample].

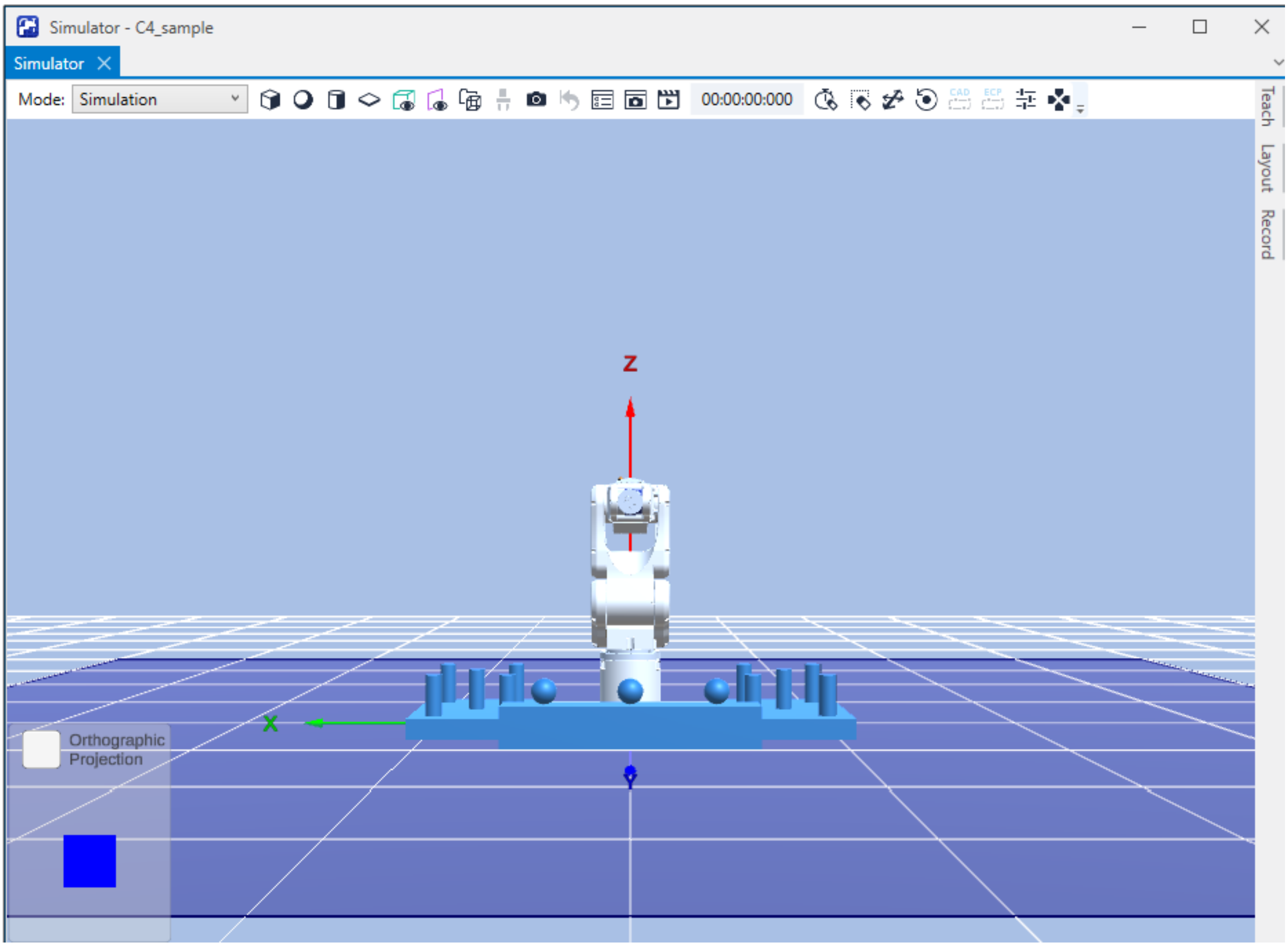


   iii. Click the [Open] button. The [Robot Simulator] window appears.



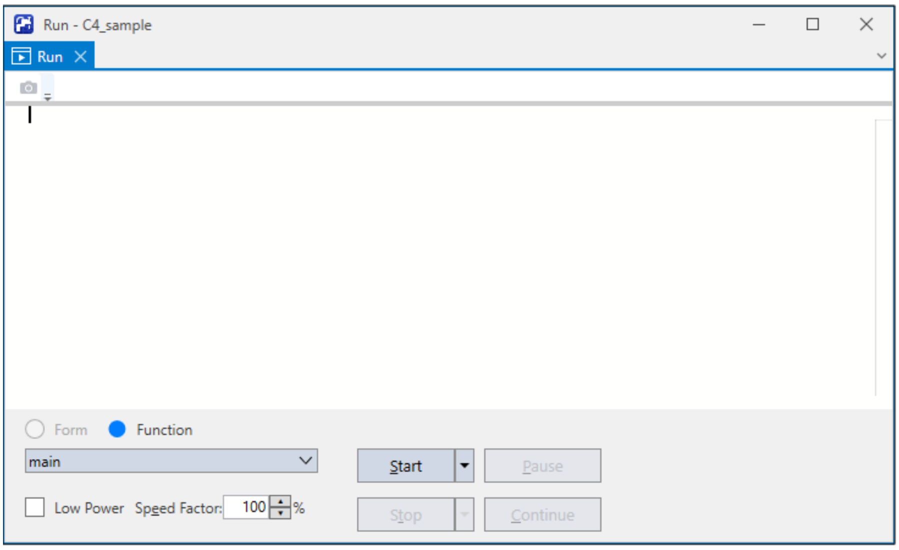
3. Display the [Robot Simulator] window

Click the  [Simulator] button on the toolbar. The [Robot Simulator] window appears.



4. Operate the robot by executing the program

i. From the tool bar, click the [Run Window] button. The project will be built and the [Run] window will appear.



ii. Click the [Start] button. When the message "Are you ready to start?" appears, click [Yes]. The program starts and the robot moves in the 3D display.

5. Next step

If you want to change the sample, follow the Steps 5 to 7 below. If you want to create your own system, start from Step 1.

Working with the user created system

If you want to change the sample virtual controller, see below and change the copied sample.

Virtual controller - "Copy the sample or configured virtual controller"

9.2.2 Working with the user created system

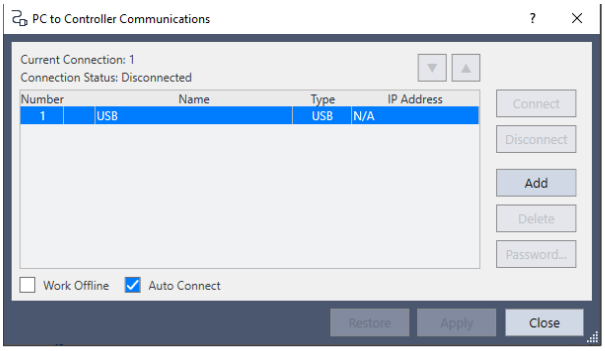
You can create your own system and simulate the robot operation on your PC.

Follow the steps below:

- Create a new virtual controller (Connection setting)
- Connect with the virtual controller
- Configure a robot
- Display the [Robot Simulator] window
- Place the objects
- Create a program
- Operate the robot by executing the program
- Measure the robot operation time
- Test the collision detection

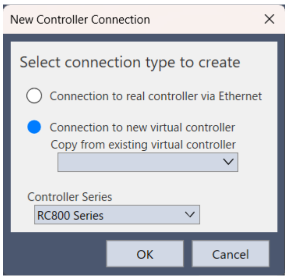
1. Create a new virtual controller (Connection setting)

i. Click the [PC to Controller Communications] button on the Epson RC+ 8.0 toolbar. The [PC to Controller Communications] dialog box is displayed.



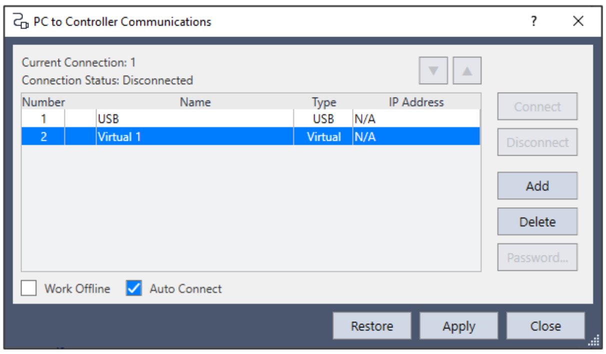
ii. Click the [Add] button. The [New Controller Connection] dialog appears.

iii. Select the [Connection to new virtual controller], and then select [RC800 Series] as the [Controller series].



iv. Click the [OK] button.

v. A new virtual controller named "Virtual 1" is created. Click the [Apply] button.



KEY POINTS

Program total execution time

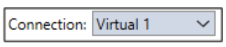
In the virtual controller, the total execution time of programs is limited to one hour.

If total execution is over one hour, a warning message appears.

You can execute the program again after the warning is displayed. The total execution timer will be reset.

vi. Click the [Close] button to return to the Epson RC+ 8.0 main window.

2. Connect with the virtual controller



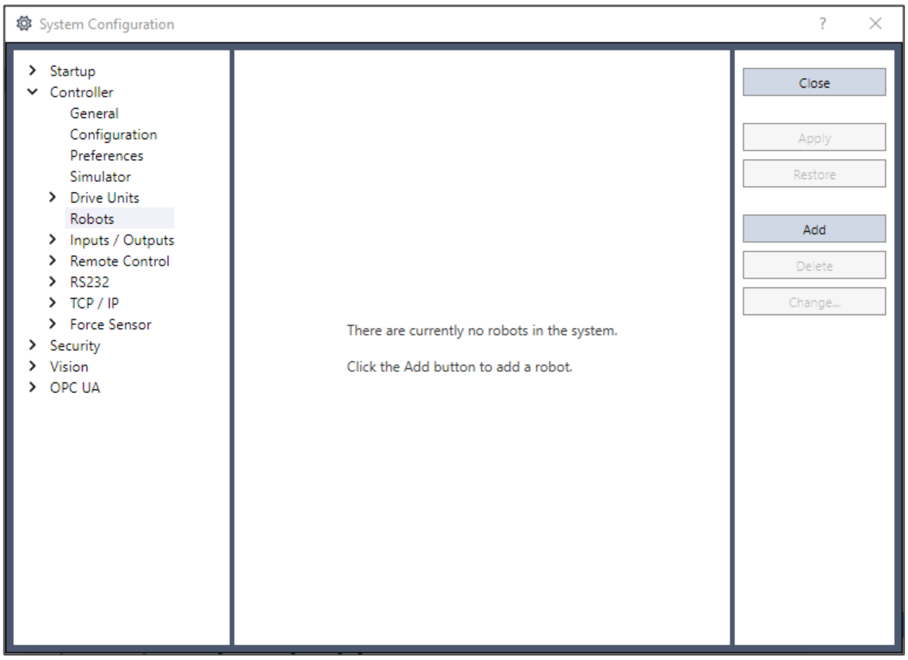
Select the created "Virtual 1" from the Epson RC+ 8.0 toolbar [Connection] list box. When the connection is complete, the [Connection] list box shows "Virtual 1".

3. Configure a robot

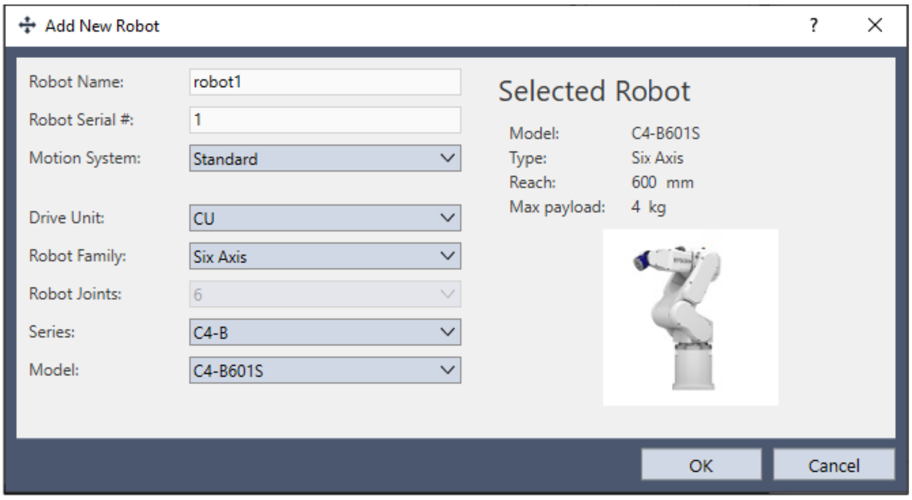
In this tutorial, a C4-B601S robot model is used.

i. In the Epson RC+ 8.0 menu, select [Setup]-[System Configuration].

ii. Select [Controller]-[Robots] from the dialog tree. The message “There are currently no robots in the system. Please register your robot" will appear.



iii. Click the [Add] button to open the [Add New Robot] dialog box. Input the robot information as follows:
[Robot Name]: robot1
[Robot Serial #]: 1
[Motion System]: Standard
[Drive Unit]: CU
[Robot Family]: Six Axis
[Series]: C4-B
[Robot]: C4-B601S

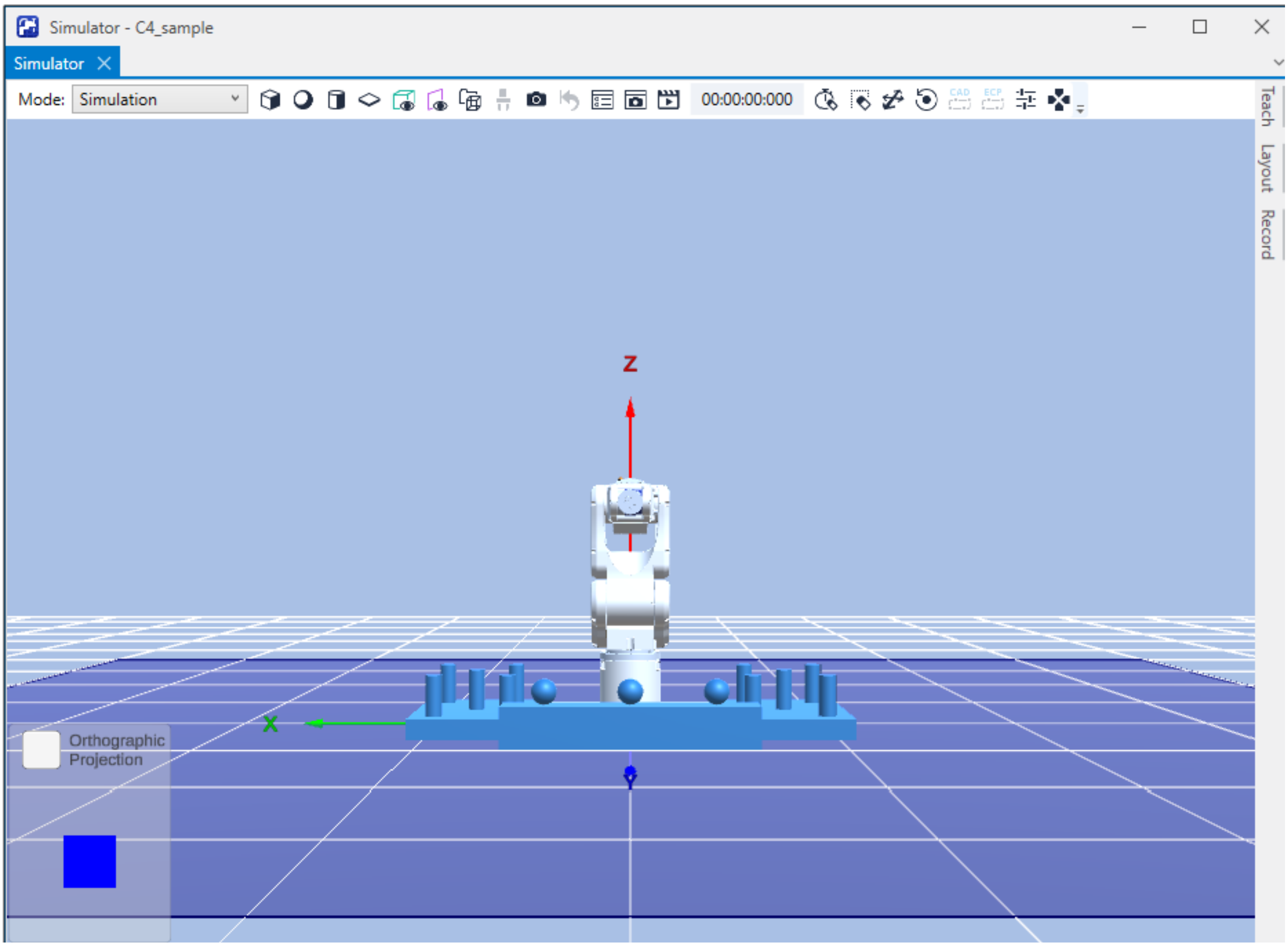


iv. Click the [OK] button. The message "Restarting Controller" appears.

v. When the message disappears, click the [Close] button to return to the Epson RC+ 8.0 main window.

4. Display the [Robot Simulator] window

Click the toolbar- [Simulator] button. The following window appears.

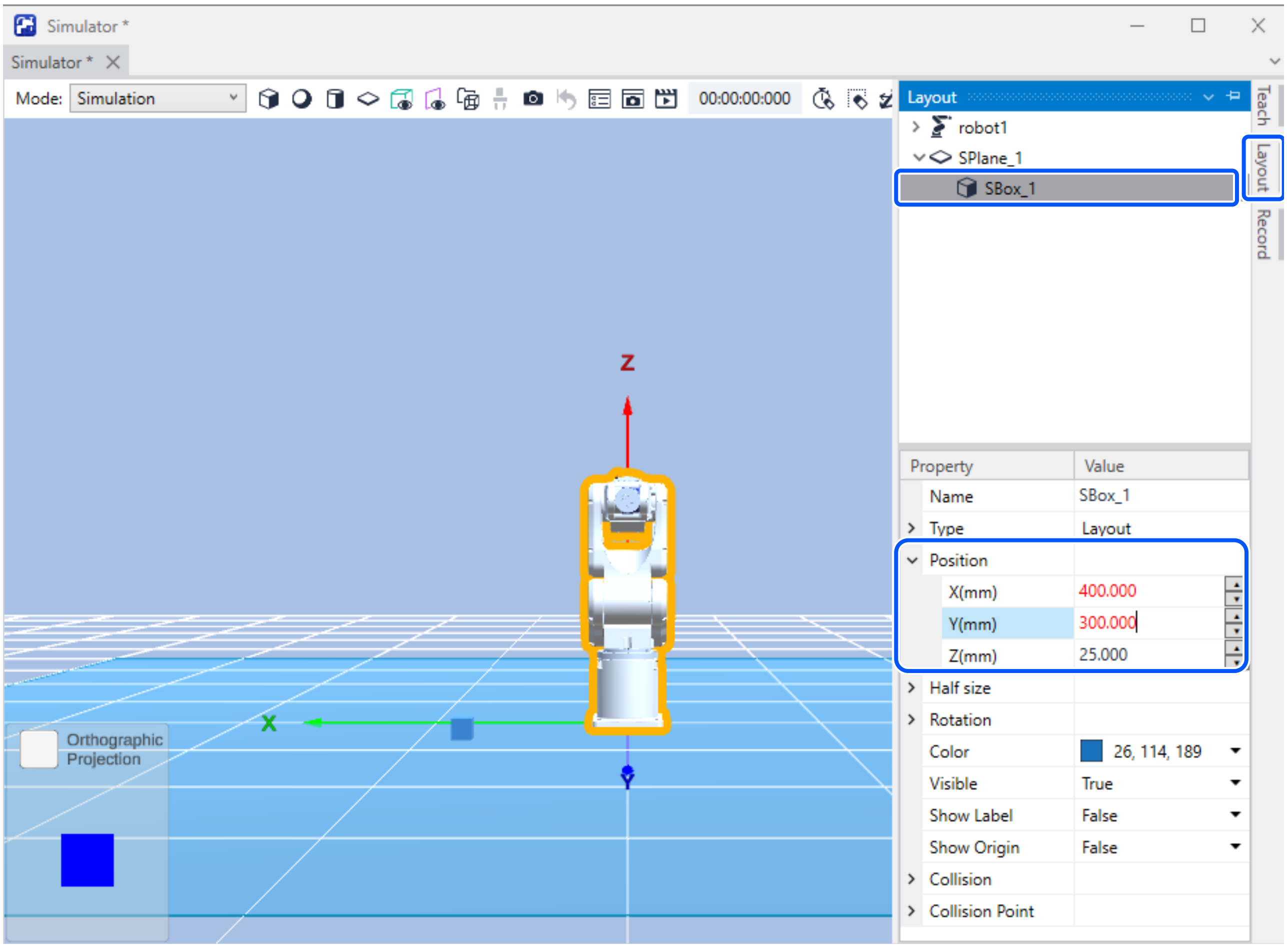


5. Place the objects

For this tutorial, we will add a box to the layout.

i. Click the [Box] button on the toolbar.

ii. Click the [Layout] tab and select "SBox_1" from the [Layout] tree. Change the [Position] of the property grid. For this tutorial, enter X = 400, Y = 300.



TIP

To save the layout change, execute the Epson RC+ 8.0 menu-[File]-[Save].

6. Create a program

   i. Create a new project

      a. Click the Epson RC+ 8.0 menu-[Project]-[New Project].

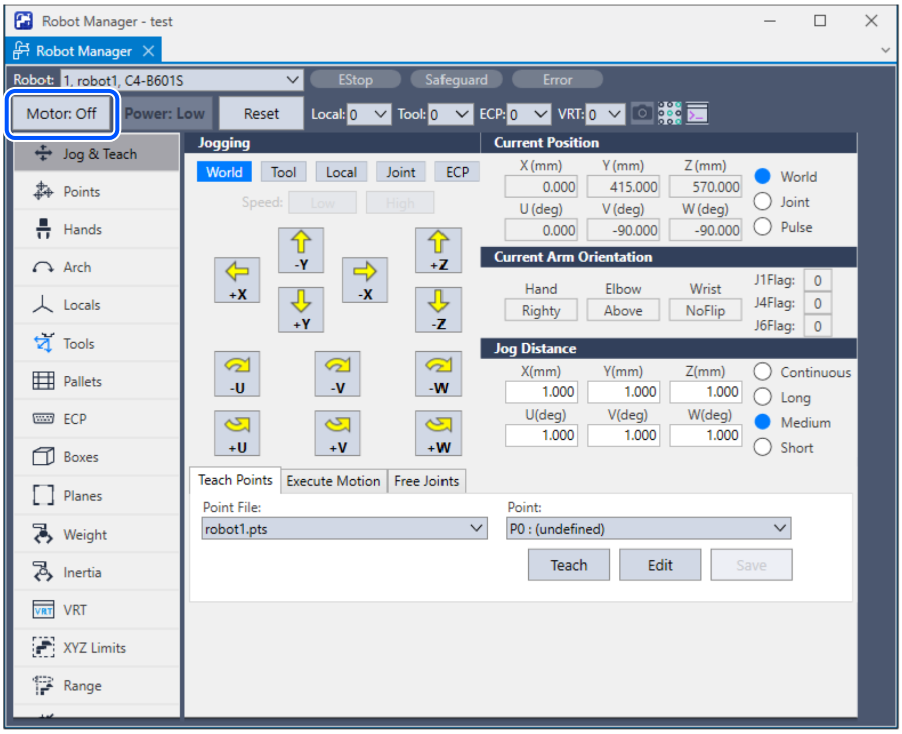
      b. Enter a new project name. For this tutorial, enter "Test".

      c. Click the [OK] button. Then, the project "Test" is created.

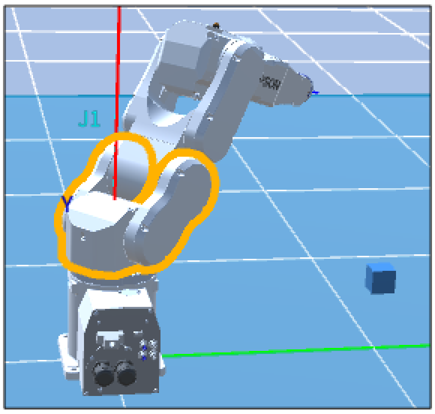
   ii. Operate the robot and teach points.

      a. Click the Epson RC+ 8.0 toolbar-  [Robot Manager] button to display the [Robot Manager].

      b. Click the [Motor: Off] button. The message to confirm the operation appears. Click the [Yes] button.

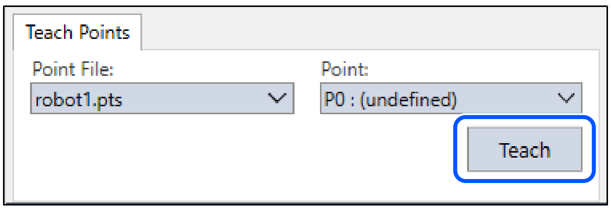


c. In the [Robot Simulator] window, move the robot joint to a point where it does not interfere with the box. A robot joint can be moved by clicking the  [Rotate/Jog] button on the toolbar and dragging the joint.



d. Click the [Teach] tab in the upper right corner of the simulator window to display the [Teach] panel.

Click the [Teach] button. The message to confirm the operation appears. Click the [Yes] button.



e. The [New Point Information] dialog appears. Click the [OK] button.

f. Select “P1 - (undefined)” from the [Point] list box on the lower right.

g. Click the toolbar  [Rotate/Jog] button and drag the joint to a position where it does not collide with the box.

h. Click the [Teach] button from the [Teach] tab. The message to confirm the operation appears. Click the [Yes] button.

i. The [New Point Information] dialog appears. Click the [OK] button.

j. Click the Epson RC+ 8.0 toolbar [Save all files] button to save the P0 and P1 data.

> **TIP**
>
> You can also use the [Jog & Teach] window to move the robot.

iii. Create and execute a program with robot motion.

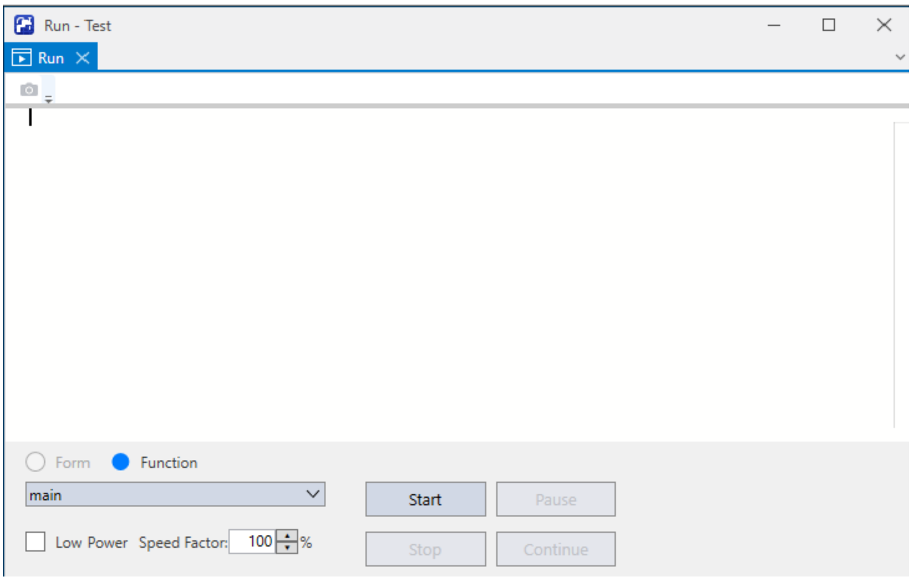
a. Create the following program in Main.prg program.

```
Function main
  Go P0
  Go P1
Fend
```

b. Click the Epson RC+ 8.0 toolbar [Build project] button. The program will be built. When the build is complete, the message "Build complete, no errors" appears in the status window.

7. Operate the robot by executing the program

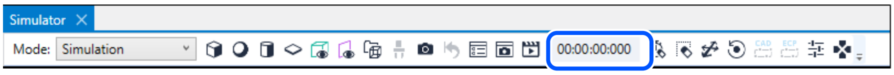
i. Click the Epson RC+ 8.0 toolbar  [Run Window] button. The following window appears.



ii. Click the [Start] button. When the message "Are you ready to start?" appears, click [Yes]. The program starts and the robot moves in the 3D display.

8. Measure the robot operation time

The elapsed program run time (cycle time) is displayed in the Tool bar of the [Robot Simulator] window. The displayed time is an estimated time from when the program is executed using the [Start] button in the [Run] window until the program ends.



The following describes how to measure the operation time between two points (P0 → P1).

i. Change the program in the “Main.prg” file to the following program.

```
Function main
  Motor On
  Power High
  Speed 100
  Accel 100,100
  Go P0
Fend

Function main2
  Go P1
Fend
```

ii. Click the Epson RC+ 8.0 toolbar [Build project] button.

The program will be built. When the build is complete, the message "Build complete, no errors" appears in the status window.

iii. Click the Epson RC+ 8.0 toolbar  [Run Window] button.

iv. Confirm that “main” is selected in the [Function] drop-down list and click the [Start] button. When the message "Are you ready to start?" appears, click [Yes].

The program starts and the robot goes to P0, the point to start the time measurement, in the 3D display.

v. Select “main2” in the [Function] dropdown list.

vi. Click the [Start] button. When the message "Are you ready to start?" appears, click [Yes].

The program starts and the robot moves in the 3D display. At this time, the elapsed time displayed in the upper right of the [Simulator] window serves as a guide for the operation time from P0 to P1.

KEY POINTS

When you operate the real robot, the actual cycle time will be longer than the simulated cycle time according to the model, Fine, load settings. See details below.

Simulator Specifications and Restrictions

Also, when Speed, Accel values in the program are changed, the cycle time will reflect it.

TIP

Motion command includes Move and Jump as well as Go. For information on how to use these commands, refer to the following manuals:

- Help
- "SPEL+ Language Reference"

9. Test the collision detection

   i. Go back to the [Robot Simulator] window.

   ii. Click the toolbar [Rotate/Jog] button and drag the joint to a position where it collides with the box.

   When the robot joint hits the box, the display turns to red.

   iii. Select “P2 - (undefined)” from the [Point] list box in the [Teach] tab, and click the [Teach] button.

   The message to confirm the operation appears. Click the [Yes] button.

   iv. The [New Point Information] dialog appears. Click the [OK] button.

   v. Click the Epson RC+ 8.0 toolbar [Save all files] button to save the P2 data.

   vi. Click the toolbar [Rotate/Jog] button and drag the joint to a position where it does not collide with the box.

   vii. Click the Tool bar- [Reset Collision] button. Then, the red display returns to normal.

   viii. Add the following function to the “Main.prg” program file.

```
Function main3
  Go P2
Fend
```

   ix. Click the Epson RC+ 8.0 toolbar [Build project] button. The program will be built.

   When the build is complete, the message "Build complete, no errors" appears in the status window.

   x. Click the Epson RC+ 8.0 toolbar [Run Window] button.

   xi. Select “main3” in the [Function] dropdown list.

   xii. Click the [Start] button.

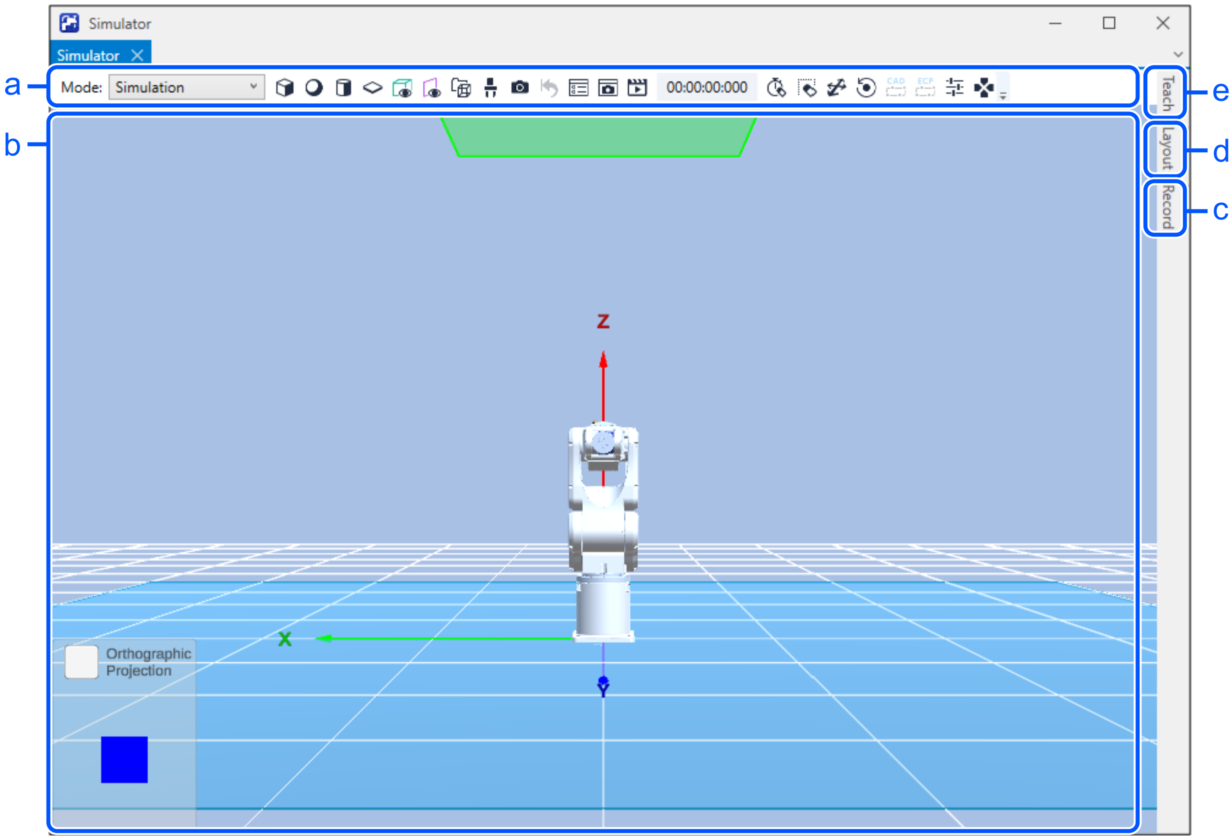
   The message "Are you ready to start?" appears. Click the [Yes] button. The program starts and the robot moves in the 3D display. When the robot joint hits the box, the display turns to red.

> **TIP**
>
> When a collision happens, the users can stop the controller program execution with an error. See below.
>
> **Collision detection**

9.3 Description of Functions

This section describes how to use the [Robot Simulator] window and functions.

9.3.1 [Robot Simulator] window layout



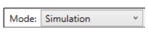
Symbol	Item	Description
a	Tool Bar	Buttons for commands commonly used in the simulator are lined up.
b	3D Display	Allows you to check the robot orientation and motion from various points of view.
c	Record Tab	Allows you to record simulation results.
d	Layout Tab	Displays lists of robot and layout objects, and sets properties.
e	Teach Tab	Used to check current position and perform point teaching.

9.3.1.1 Tool Bar

The configuration of the toolbar will change according to the mode. There are two modes: simulation mode and playback mode. Simulation mode is described here.

For more information on the Playback mode toolbar, see below.

Record / Playback

Button	Description
	Simulator operating mode. It switches between [Simulation Mode] and [Playback Mode].
 Layout Box	Adds a box object.

Button	Description
Layout Sphere	Adds a sphere object.
Layout Cylinder	Adds a cylinder object.
Layout Plane	Adds a floor / wall object.
Surveillance Area	Adds a surveillance area object.
Surveillance Plane	Adds a surveillance plane object.
CAD	Adds a CAD object. When you click this button, the [Open CAD Data] dialog appears.
Hand	Adds a hand object. When you click this button, the [Open Hand Data] dialog appears. Hand sample data is provided in the Epson RC+ 8.0 directory (`C:\EpsonRC80\Simulator\Hand Samples`).
Camera	Add a virtual camera. When you click this button, the [System Configuration] dialog box opens and the Add Camera window appears. You can select a camera and lens.
Reset Collision	Resets the collision detection status. When you click this button while the robot is not interfering with any layout objects, the red display turns to normal.
Simulator Settings	Shows the [Simulator Settings] dialog. In this dialog, the 3D [Render Options] can be configured.
Screenshot	Saves the current 3D display as an image file. The [Save As] dialog box appears, allowing you to specify a file name and file type (JPEG or PNG format).
Create Movie	In Simulation mode, a project can be executed. In Playback mode, simulation results (log files) can be played back and movie files can be saved in MP4 format. A dialog for naming and saving a file appears.
00:00:00:000 Elapsed Time	Shows the program execution time as if you ran the same program with a real controller. When a program starts, the elapsed time counter counts from "0:" and stops when the program finished. It pauses counting when the program is paused and resumes when the program continues execution.
Clear Elapsed Time	Resets the elapsed time to "0".
Clear TCP path	Clear the TCP path (including Render Singularity Avoiding path) which the robot displays.
Move	Displays the guides. The objects can be moved by dragging the guides. For SCARA robots, the third joint can be dragged to move up and down.

Button	Description
Rotate/Jog	Displays the guides. The objects can be rotated by dragging the guides. The guides for the Manipulator are shown only at the Manipulator base. The joint angle can be changed by dragging the arm. For SCARA robots, the fourth joint can be rotated.
CAD CAD to Point	Switches to the mode for outputting the point data from the CAD data.
ECP CAD to Point for ECP	Switches to the mode for outputting the point data for external control point (ECP) motion from the CAD data.
Robot Operation Panel	Displays the robot operation panel. Jog operation is available.
Direct Teaching	By dragging the robot, you can operate jog operating with virtual direct teaching.

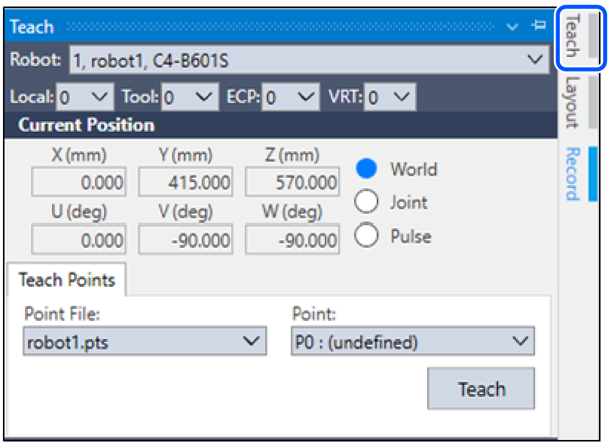
9.3.1.2 Teach

Select [Teach] from the right tab of the simulator window to display the Teach panel. You can teach the robot while checking the 3D display.

The Teach panel includes some functions of the [Jog & Teach] panel on Robot Manager.

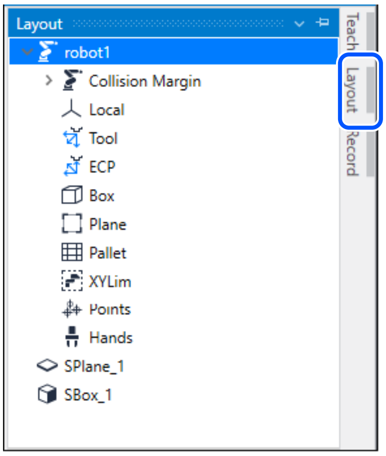
See details below.

[Tools]-[Robot Manager]-[Jog & Teach] Panel

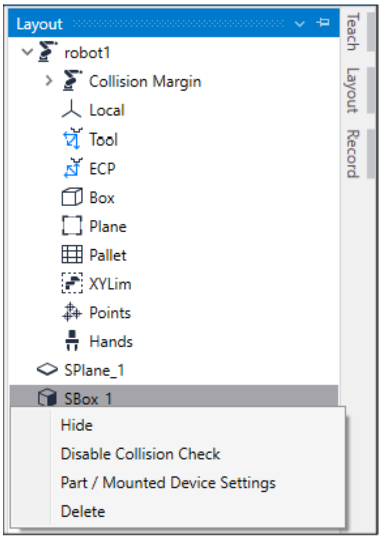


9.3.1.3 Layout Tree

Select [Layout] from the right tab of the simulator window to display a tree view of robot and layout objects.



The context menu appears by right-clicking the layout object. The frequently used functions can be used without operating by the property grid. The displayed items vary depending on objects.



[Cut], [Copy], and [Paste] commands in the [Edit] menu are available for layout objects other than CAD objects.

Also, the hierarchy of the layout objects can be changed by dragging and dropping.

What is an Object?

The objects in the simulator are either a "Robot Object" or a "Layout Object":

A "Robot Object" includes the robot itself, its hand, local coordinates, point information, etc. A "Layout Object" includes objects to be placed around the robot to simulate the robot peripheral environment in the 3D display.

Robot Object

- Robot: The robot itself. The display data is handled by the simulator.
- Hand: The hand is created by loading the CAD data (XVL(.xv3), VRML2.0, STEP, IGES, and DXF) from a file.
- Force Sensor: Set force sensor can be displayed.
- Safety Function: When using option of safety function, it displays robot monitored range and monitored areas.
- Object to reflect force control data: Force Control, Force Guide
- Object to reflect collision detection margin: Collision Margin

- Object to reflect a robot parameter : Local, Tool, Box, Plane, Pallet, XYLim
- Object to reflect robot point data: Point
- Objetct to reflect safety function parameter: Monitored Range, Monitored Areas

Layout Objects

- Simple object: Box, Sphere, Cylinder, Floor / Wall
  The display data for these objects is handled by the simulator. The object size can be changed as desired by editing properties.
- CAD object: These objects are created by loading CAD data (XVL(.xv3), VRML2.0, STEP, IGES, and DXF) from a file.

Camera Objects

The following devices can be displayed. Devices that are supported in Epson RC+ 8.0 Option Vision Guide 8.0 Hardware & Setup can be selected.

- Camera: USB and GigE camera can be selected.
- Lens: Each model of standard camera lens, megapixel camera lens, mega-pixel camera lens (HF), 1-inch lens can be selected.
- Extension tube: Each length of tubes can be selected.

Surveillance objects

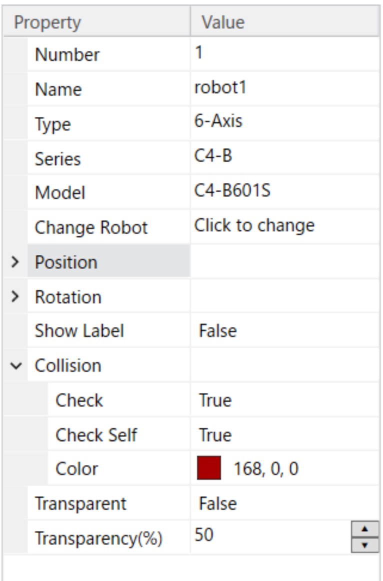
Layout object that detects a contact or a collision with a robot. There are two types; Surveillance Area and Surveillance Plane. Similar to the layout objects, the display data is prepared in advance. The object size can be changed as desired by editing properties.

9.3.1.4 Property Grid Pane

In the Property Grid pane, you can view and change the settings of the robot objects and layout objects in the Layout Object pane.

9.3.1.4.1 Robot Object Properties

Robot



Property	Value
Number	Robot number

Property	Value
Name	Robot name. You can specify any name for a robot.
Type	Robot type The robot type (SCARA and 6-Axis) is shown. This property is read-only.
Series	Robot series The robot series is shown. This property is read-only.
Model	Robot model name The robot model is shown. This property is read-only.
Change Robot	If you want to change the robot model, click on [Change Robot] button. When you click on this button, a dialog appears to change the robot. For details, see Changing the robot model described later in this chapter.
Position	Robot position Specifies the base center in the simulator World coordinates.
Rotation	Robot angle
Show Label	▪ Displays the label: True ▪ Not display the label: False (default)
Show Singularity Area	Shows/hides elbow singularity areas for N series robots. ▪ Show area: True ▪ Not show area: False (Default) This can be set for N series only.

Collision Property	Value
Check	Enables / disables the collision detection for layout objects. ▪ Enable: True (default) ▪ Disable: False Even if this is enabled, it does not detect collision between the robot base and layout objects.
Check Self	Enables / disables the collision detection for a robot itself. ▪ Enable: True (default) ▪ Disable: False
Color	Specify color to be used when collision of the arms is detected. Default: 168,0,0

Property	Value
Transparent	▪ Semi-transparent: True ▪ Not semi-transparent: False (default) The front-back relation of the objects may be incorrect depending on the viewing angle. See details below. **Simulator Specifications and Restrictions**
Transparency	Specifies the transparency in the range of 1 to 90%. Transparency increases as the setting value becomes larger.

Changing the robot model

When you want to change the displayed Robot click the [Change Robot] button. The [System Configuration]-[Controller]-[Robots]-[Robot**]-[Change]-[Change Robot] dialog will appear.

[Change Robot] button is not displayed, increase the property grid width and click the [Value] column of the grid once.

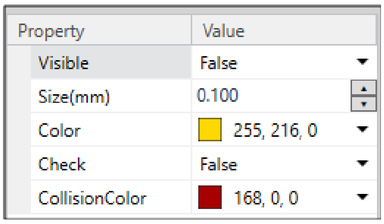
KEY POINTS

When you change the displayed robot model, all the settings for the robot (Local coordinates, Tool coordinates, etc.) will be initialized to the default values.

Collision Margin

Sets a collision detection margin for each robot arm collectively or separately.

When the robot type is changed from SCARA robot to 6-Axis robot or 6-Axis robot to SCARA robot, the Value will be reset. When changing to the same type robot, the value will be remained.



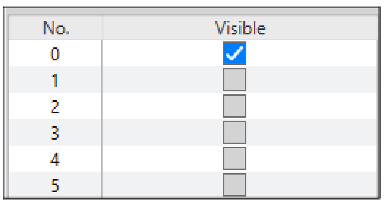
Property	Value
Visible	▪ Display: True ▪ Not display: False (Default)
Size	Margin size
Color	Margin color Default: 255,216,0
Check	Enables / disables the collision detection. ▪ Enable: True ▪ Disable: False (default)
CollisionColor	Specifies a color to be displayed when detecting a collision. Default: 168,0,0

Local, Tool, ECP, Box, Pallet

If the local coordinate system of the corresponding number is not defined yet, the check box is grayed.

You can set up from the [Tools]-[Robot Manager] items.

- Local : Local settings
- Tool : Tool settings
- ECP : External control point settings
- Box : Entry detection area
- Pallet : Pallet settings

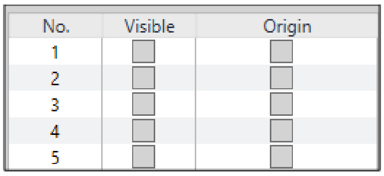


Item	Description
Visible	Shows / Not show corresponding settings. ▪ Not visible: Uncheck (default) ▪ Visible: Check

For Local 0 (Base), "Visible" is the default.

Plane

If the each setting of the corresponding number is not defined yet, the check box is grayed. You can set up from [Tools]-[Robot Manager]-[Planes].

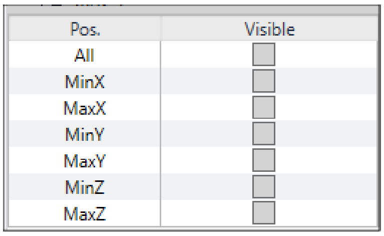


Item	Description
Visible	Shows / Not show corresponding settings. ▪ Not visible: Uncheck (default) ▪ Visible: Check
Origin	Shows / Not show the origin of corresponding settings. When the checkmark is not placed in [Visible], the [Origin] checkbox is grayed. ▪ Not visible: Uncheck (default) ▪ Visible: Check

XYLim

If the XYLim is not defined yet, the check box is grayed.

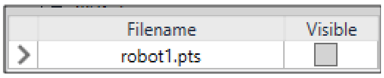
You can set up from [Tools]-[Robot Manager]-[XYZ Limits].



Item	Description
Pos.	Shows the position of parameter that configures XYLim.
Visible	Shows / Not show the plane in corresponding position. ▪ Not visible: Uncheck (default) ▪ Visible: Check If Pos. All is checked, it displays all plane that configures XYLim. Not put in the check, they will be hidden. If it set to show some planes, the check box shows indeterminate state.

Points

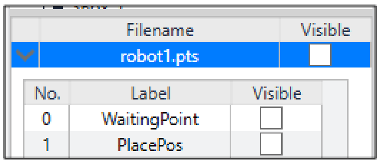
Display the point display setting status in the point file. Switch to show/non-show all points.



Item	Description
File Name	Show a point file name.
Visible	Shows / Not show all points ▪ Not visible: Uncheck ▪ Visible: Check If it is set to show some points, the check box shows indeterminate state.

Point

If the point of the corresponding number is not defined yet, the check box is grayed.

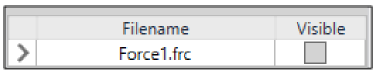


Item	Description
Label	Shows a point label In the dialog, point labels cannot be configured or edited.
Visible	Shows / Not show a point ▪ Not visible: Uncheck (default) ▪ Visible: Check

> **TIP**
>
> If you cannot see the [Visible] column, increase the property grid display width.

Force Control

Show the display method of the force objects in the force file. Switch visible and not visible of the all force objects.

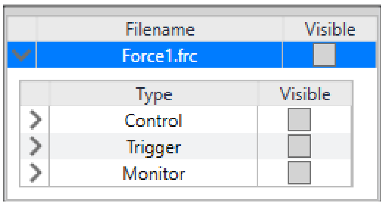


Item	Description
File Name	Show a force file name.
Visible	Shows / Not Show all force objects. ▪ Not visible: Uncheck ▪ Visible: Check If it is set to show some force objects, the check box shows indeterminate state.

Force Object

Show the display method of the force control objects, force trigger objects, and force monitor objects in the force file. Switch visible and not visible of the all force objects of the specified type.

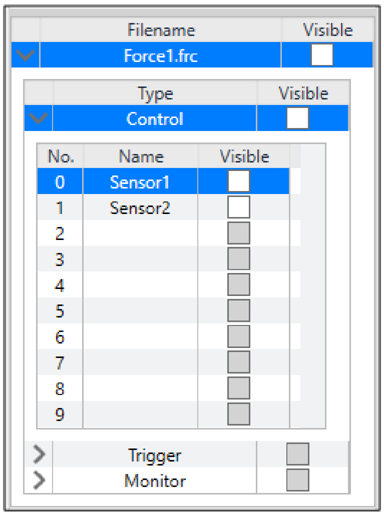
If there is no force object of the specified type, the checkbox is grayed.



Item	Description
Type	Control (force control), Trigger (force trigger), and Monitor (force monitor) are shown.
Visible	Shows / Not Show all force objects of the specified type. ▪ Not visible: Uncheck ▪ Visible: Check If it is set to show some force objects, the check box shows indeterminate state.

Force Control, Force Trigger, Force Monitor

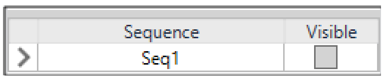
When the force object of the corresponding number is not defined yet, the checkbox is grayed.



Item	Description
Name	Show a force label. In the dialog, force label cannot be configured or edited.
Visible	Shows / Not Show the force object. ▪ Not visible: Uncheck ▪ Visible: Check

Force Guide

Show the display method of the force guide object in the sequence which is set by the force guide, and switch visible and not visible of the all force objects.

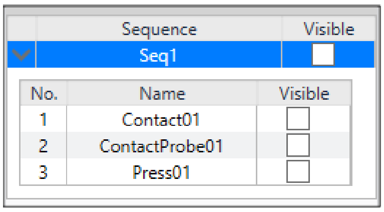


Item	Description
Sequence	Show a force guide sequence name.

Item	Description
Visible	Shows / Not Show the all force objects. ▪ Not visible: Uncheck ▪ Visible: Check If it is set to "Visible" for some force objects, the check box shows indeterminate state.

Force Guide Object

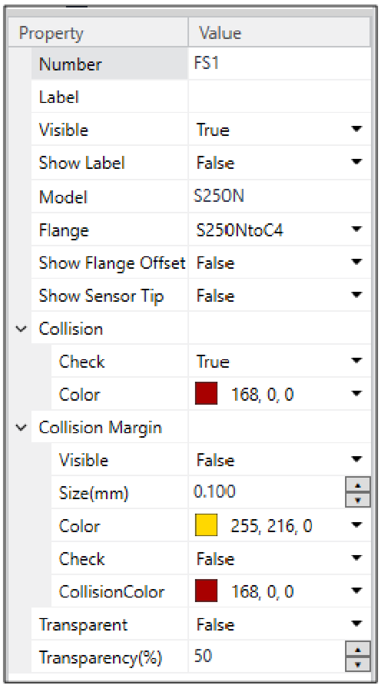
When the force guide object type of the corresponding number (a step number in the sequence) is "Decision" or "SPELFunc", the checkbox is grayed. When the [Enabled] property of the object is set to "False", the [Name] is also grayed.



Item	Description
Name	Show a force guide object name.
Visible	Shows / Not Show the force guide object. ▪ Not visible: Uncheck ▪ Visible: Check

Force Sensor

By registering Force Sensor, "Force Sensor" is displayed in the layout object.



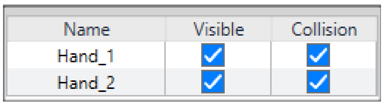
Property	Value
Number	Displays a sensor number registered in the controller.
Label	Displays a sensor name registered in the controller.

Property	Value
Visible	▪ Display: True (default) ▪ Not display: False
Show Label	▪ Displays the label: True ▪ Not display the label: False (default)
Model	Displays a model registered in the controller.
Flange	Displays a flange determined by combination of the robot and Force Sensor (default). Select "None" to hide.
Show Flange Offset	▪ Display: True ▪ Not display: False (Default) Specifies whether to display the flange offset position in coordinate system.
Show Sensor Tip	▪ Display: True ▪ Not display: False (Default) Specifies whether to display the tip position of Force Sensor in coordinate system.

For properties such as "Collision", "Collision Margin" or "Transparent", refer to the hand or layout objects properties.

Hands

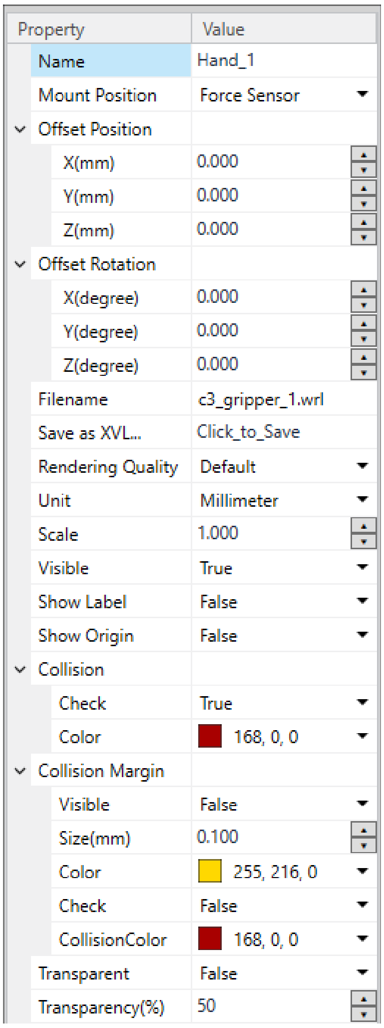
Show a status of Visible and Collision of the configured hand.



Item	Description
Name	Shows a hand name.
Visible	Shows / Not Show the hand. ▪ Not visible: Uncheck ▪ Visible: Check
Collision	Enable / Disable the collision detection. ▪ Disable: Uncheck ▪ Enable: Check

Hand

When a hand is registered with a robot, "Hand" is added in the Layout Objects tree.



Property	Value
Name	Hand name You can specify any name for a hand. (Default: Hand_1)
Mounted Position	Hand mounting position It can be attached to Force Sensor or Tool coordinate system.
Offset Position	Mounting offset from the robot end effector position.
Offset Rotation	Hand mounting direction
File name	CAD data file name of the hand. It cannot be changed.
Save as XVL…	The loaded hand object can be saved in the XVL format. Click [...] and specify the destination. When XVL-format hand data is loaded, this item is grayed out and cannot be used.
Rendering Quality	Set the rendering quality. ▪ Standard: Default ▪ Quality-preferred: Fine ▪ Speed-preferred: Fast
Unit	Set the unit of length for the CAD data.
Scale	Set the scale ratio of the CAD data.

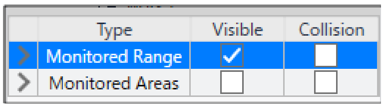
Property	Value
Visible	▪ Display: True (default) ▪ Not display: False
Show Label	▪ Displays the label: True ▪ Not display the label: False (default)
Show Origin	▪ Displays the origin coordinate system: True ▪ Not displays the origin coordinate system: False (default)

Collision Property	Value
Check	Enables / disables the collision detection. ▪ Enable: True (default) ▪ Disable: False Collision with the robot flange is not detected even when this property is set "True".
Color	Specifies a color to be displayed when detecting a collision. Default: 168,0,0

Collision Margin Property	Value
Visible	▪ Display: True ▪ Not display: False (Default)
Size	Margin size
Color	Margin color Default: 255,216,0
Check	Enables / disables the collision detection. ▪ Enable: True ▪ Disable: False (default)
CollisionColor	Specifies a color to be displayed when detecting a collision. Default: 168,0,0
Transparent	▪ Semi-transparent: True ▪ Not semi-transparent: False (default) The front-back relation of the objects may be incorrect depending on the viewing angle. See details below. **Simulator Specifications and Restrictions**
Transparency	Specifies the transparency in the range of 1 to 90%. Transparency increases as the setting value becomes larger.

Safety Function

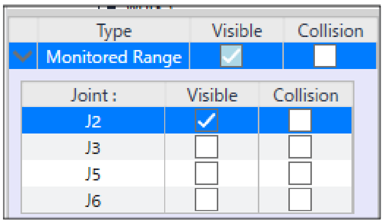
If option of safety function is enabled, "Safety Function" is displayed in the layout object. Shows robot monitored range and monitored areas are displayed or hidden, and also shows status of collision. Moreover, it switches visible and not visible of robot monitored range and monitored areas, and switched enabled and disabled of collision detection.



Item	Description
Type	Displays Range of robot monitored range, and Areas of robot monitored areas.
Visible	Shows / Not Show the robot monitored range and monitored areas. ▪ Not visible: Uncheck ▪ Visible: Check If it set to show some robot monitored range and monitored areas, the check box shows indeterminate state.
Collision	Enable / Disable the collision detection for the robot monitored range and monitored areas. ▪ Disable: Uncheck ▪ Enable: Check If collision detection of some of robot monitored range and monitored areas enabled, the check box shows indeterminate state.

Monitored Range

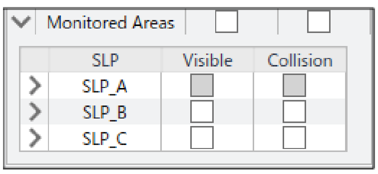
Shows robot monitored range are displayed or hidden, and shows the enabled and disabled of collision detection.



Item	Description
Joint	Display joints that able to be set of robot monitored range. If it is SCARA robot, displays J2 or J3. If it is 6-Axis robot, displays J2, J3, J5, and J6.
Visible	Shows / Not Show the robot monitored range. ▪ Not visible: Uncheck ▪ Visible: Check
Collision	Enable / Disable the collision detection for the robot monitored range. ▪ Disable: Uncheck ▪ Enable: Check

Monitored Areas

Shows monitored areas are displayed or hidden, and shows status of collision detection. Also, it switches visible and not visible of monitored areas, and switches enabled and disabled of collision detection.

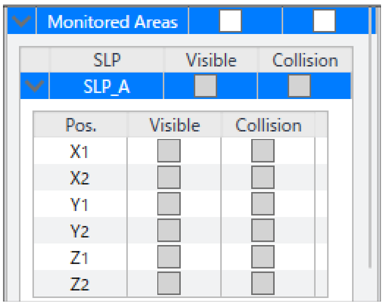


Item	Description
SLP	Displays monitored areas SSLP_A, SLP_B, and SLP_C.

Item	Description
Visible	Shows / Not Show the robot monitored range. ▪ Not visible: Uncheck ▪ Visible: Check If it set to show some robot monitored areas, the check box shows indeterminate state.
Collision	Enable / Disable the collision detection for the robot monitored areas. ▪ Disable: Uncheck ▪ Enable: Check If collision detection of some of robot monitored areas enabled, the check box shows indeterminate state.

SLP_A, SLP_B, SLP_C

Shows each plane that configures monitored areas are displayed or hidden, and shows status of collision detection. If the plane in corresponding position is not defined yet, the checkbox is grayed.

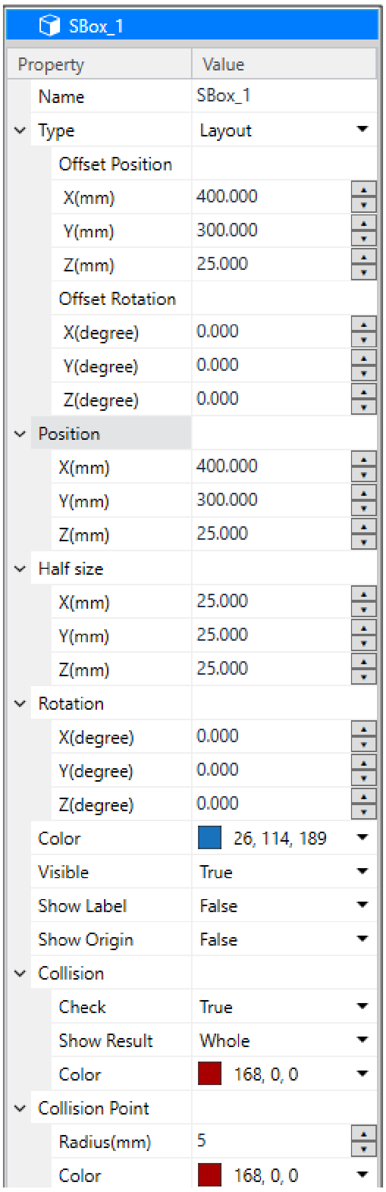


Item	Description
Pos.	Display X1, X2, Y1, Y2, Z1 and Z2 as a plane that configures monitored areas.
Visible	Shows / Not Show the plane that configures monitored areas. ▪ Not visible: Uncheck ▪ Visible: Check
Collision	Enable / Disable the collision detection for the plane that configures monitored areas. ▪ Disable: Uncheck ▪ Enable: Check

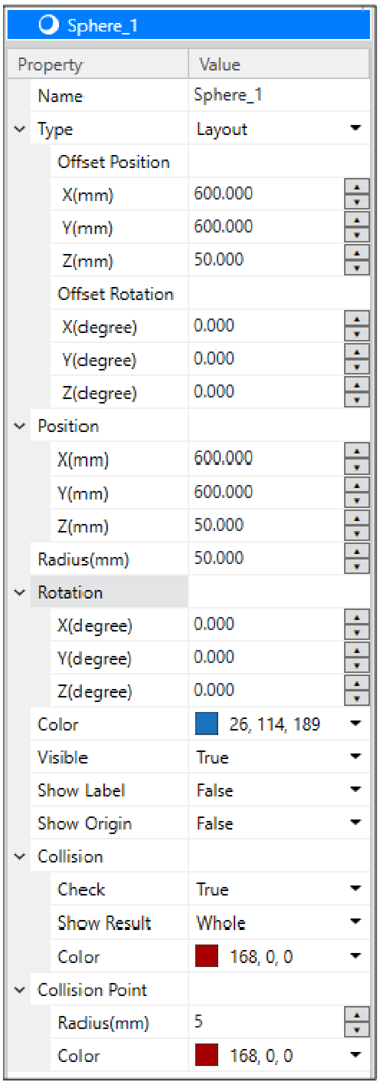
9.3.1.4.2 Layout Objects

Layout Box, Layout Sphere, Layout Cylinder, Layout Plane, CAD

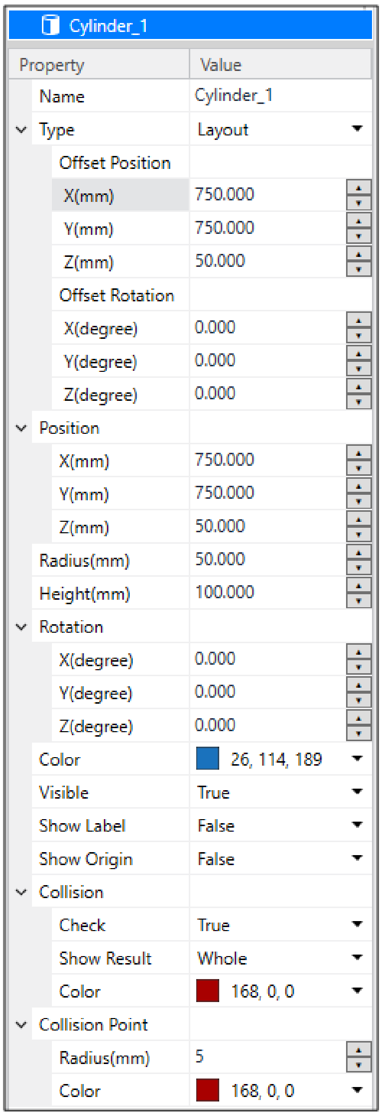
There are common attributes for all objects and others are for particular objects.



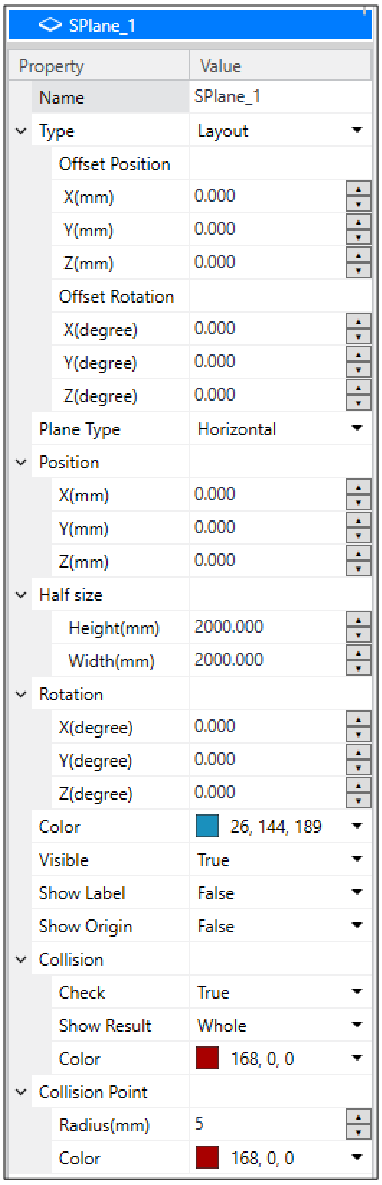
Property	Value
SBox_1	
Name	SBox_1
Type	Layout
Offset Position	
X(mm)	400.000
Y(mm)	300.000
Z(mm)	25.000
Offset Rotation	
X(degree)	0.000
Y(degree)	0.000
Z(degree)	0.000
Position	
X(mm)	400.000
Y(mm)	300.000
Z(mm)	25.000
Half size	
X(mm)	25.000
Y(mm)	25.000
Z(mm)	25.000
Rotation	
X(degree)	0.000
Y(degree)	0.000
Z(degree)	0.000
Color	26, 114, 189
Visible	True
Show Label	False
Show Origin	False
Collision	
Check	True
Show Result	Whole
Color	168, 0, 0
Collision Point	
Radius(mm)	5
Color	168, 0, 0

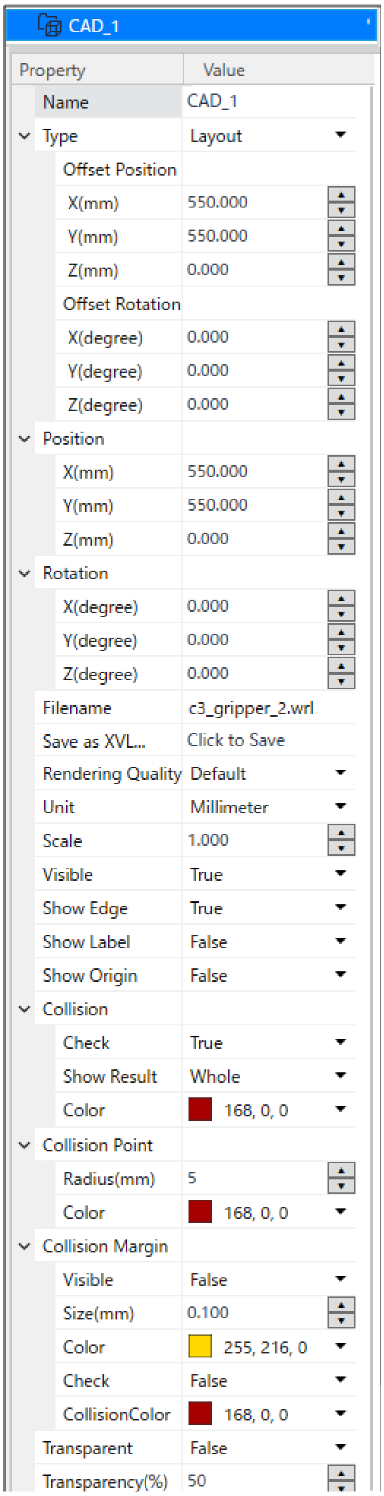


Property	Value
Sphere_1	
Name	Sphere_1
Type	Layout
Offset Position	
X(mm)	600.000
Y(mm)	600.000
Z(mm)	50.000
Offset Rotation	
X(degree)	0.000
Y(degree)	0.000
Z(degree)	0.000
Position	
X(mm)	600.000
Y(mm)	600.000
Z(mm)	50.000
Radius(mm)	50.000
Rotation	
X(degree)	0.000
Y(degree)	0.000
Z(degree)	0.000
Color	26, 114, 189
Visible	True
Show Label	False
Show Origin	False
Collision	
Check	True
Show Result	Whole
Color	168, 0, 0
Collision Point	
Radius(mm)	5
Color	168, 0, 0



Property	Value
Cylinder_1	
Name	Cylinder_1
Type	Layout
Offset Position	
X(mm)	750.000
Y(mm)	750.000
Z(mm)	50.000
Offset Rotation	
X(degree)	0.000
Y(degree)	0.000
Z(degree)	0.000
Position	
X(mm)	750.000
Y(mm)	750.000
Z(mm)	50.000
Radius(mm)	50.000
Height(mm)	100.000
Rotation	
X(degree)	0.000
Y(degree)	0.000
Z(degree)	0.000
Color	26, 114, 189
Visible	True
Show Label	False
Show Origin	False
Collision	
Check	True
Show Result	Whole
Color	168, 0, 0
Collision Point	
Radius(mm)	5
Color	168, 0, 0





Property	Object	Value
Name	All	You can specify any name.
Plane Type	Plane	▪ Floor: Horizontal (default) ▪ Wall: Vertical

Property	Object	Value
Type	All	Sets the object type. ▪ Layout: Layout object (default) ▪ Part: Part object ▪ Mounted Device: Mounted device When Part or Mounted Device is selected, the [Part/Mounting Device Settings] dialog box appears. See details below. **Part/Mounted Device Settings**
Offset Position	All	You can view/set the position of an object as an offset from its parent object. If the parent object does not exist, it will display/set the same as the Position property.
Offset Rotation	All	You can view/set the object's placement angle as an offset from the parent object. If the parent object does not exist, it will display/set the same as the Rotation property.
Position	All	Specifies a center point in simulator World coordinates. Layout Cylinder: Bottom surface center
Half size	Box, Plane	Box is specified by length from the center. Specified by X, Y, Z. The length is double this length. Plane is specified by height (length of floor, height of wall) and width (width of floor and wall). The length, height, and width are double the specified size.
Radius	Sphere, Cylinder	Sphere radius, cylinder radius
Height	Cylinder	Cylinder height
Rotation	All	Object angle (Z-axis centering)
Filename	CAD	CAD data file name. It cannot be changed.
Save as XVL…	CAD	The loaded hand object can be saved in the XVL format. Click [...] and specify the destination. When XVL-format hand data is loaded, this item is grayed out and cannot be used.
Rendering Quality	CAD	Specify the rendering quality. ▪ Standard: Default ▪ Quality-preferred: Fine ▪ Speed-preferred: Fast
Unit	CAD	Set the unit of length for the CAD data.
Scale	CAD	Set the scale ratio of the CAD data.
Color	Box, Sphere, Cylinder, Plane	Display color Refer to Change layout object color for the details.
Visible	All	▪ Display: True (default) ▪ Not display: False
Show Edge	CAD	Displays Edge (edge line) of the CAD data. ▪ Display: True (default) ▪ Not display: False When loading a CAD file, you can specify whether to show or hide the Edge.

Property	Object	Value
Show Label	All	▪ Displays the label: True ▪ Not display the label: False (default)
Show Origin	All	▪ Displays the origin coordinate system: True ▪ Not displays the origin coordinate system: False (default)
Transparent	CAD	▪ Semi-transparent: True ▪ Not semi-transparent: False (default) The front-back relation of the objects may be incorrect depending on the viewing angle. See details below. **Simulator Specifications and Restrictions**
Transparency	CAD	Specifies the transparency in the range of 1 to 90%. Transparency increases as the setting value becomes larger.

Collision

Property	Object	Value
Check	All	Enables / disables the collision detection. ▪ Enable: True (default) ▪ Disable: False Collision with the robot flange is not detected even when this property is set "True".
Show result	All	Specify how to display the color configured in Color property when the a collision is detected. ▪ Entirely: Whole (default) ▪ Collision point: Point ▪ Entire object and collision point: WholeAndPoint
Color	All	Specify the color to be used when collision of the arms is detected. Default: 168,0,0

Collision Point

Property	Object	Value
Radius(mm)	All	Specify the radius of collision point displayed when the collision is detected.
Color	All	Specify the color to be used when collision of the arms is detected. Default: 168,0,0

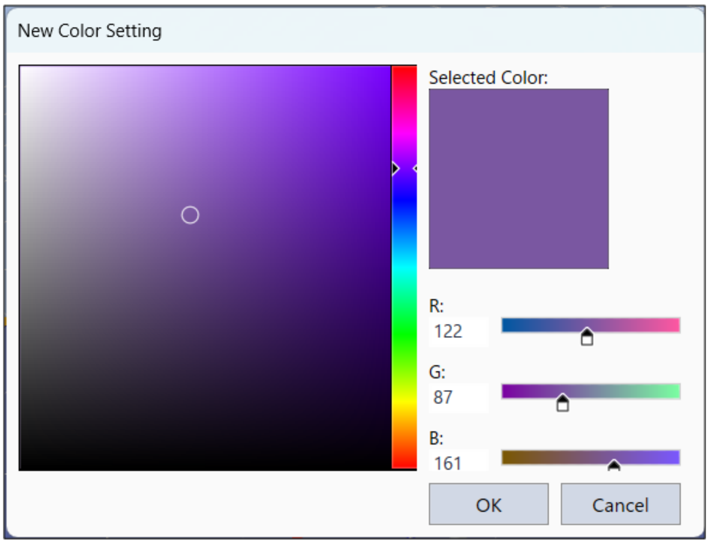
Change layout object color

When you want to change layout object color, click on the drop-down ▾ in the Color property and the dialog shown below will be displayed. If you cannot see the drop-down ▾, increase the property grid width.

Click color you want to display. Layout object color will be changed.

If you do not want to change color, click anywhere other than the display color setting dialog. The dialog will be closed.

If you create a custom color, right-click any color in the bottom two rows (16 colors) in the [Custom] tab, and the color setting dialog will be displayed.



Create a custom color and click the [OK] button.

Created color will be displayed in the display color setting dialog.

9.3.1.4.3 Camera Objects

Some properties are common for the both fixed camera and mobile camera, and other properties are only enabled for either one of them.

Camera_1

Property	Value
Name	Camera_1
Camera Number	1
Type	PC Vision
Connection Type	GigE
Model	acA640-120gm
Resolution	640 x 480
Extension Tube	0.0 mm
Lens Type	Mega Pixel
Focal Length	8 mm
Camera View	
Margin(mm)	5
Camera Tip	
X(mm)	1664.500
Y(mm)	1650.000
Z(mm)	66.900
Visible	False
Show View Ray	True
Show View Cent...	True
Near Plane	
Width(mm)	46
Height(mm)	34
Distance(mm)	100
Visible	True
Color	255, 255, 0
Fill	False
Far Plane	
Width(mm)	650
Height(mm)	486
Distance(mm)	1500
Visible	True
Color	0, 255, 0
Fill	True
Pixel Resolution	
Near X(mm)	0.072
Near Y(mm)	0.071
Far X(mm)	1.016
Far Y(mm)	1.013
Mount Type	Fixed
Position	
X(mm)	1650.000
Y(mm)	1650.000
Z(mm)	0.000
Rotation	
X(degree)	0.000
Y(degree)	0.000
Z(degree)	0.000
Visible	True
Show Label	False
Show Origin	False
Collision	
Check	True
Show Result	Whole
Color	168, 0, 0
Collision Point	
Radius(mm)	5
Color	168, 0, 0
Collision Margin	
Visible	False
Size(mm)	0.100
Color	255, 216, 0
Check	False
CollisionColor	168, 0, 0
Transparent	False
Transparency(%)	50

Camera_2(robot1: J6)

Property	Value
Name	Camera_2
Camera Number	2
Type	PC Vision
Connection Type	GigE
Model	acA640-120gm
Resolution	640 x 480
Extension Tube	0.0 mm
Lens Type	Mega Pixel
Focal Length	8 mm
Camera View	
Margin(mm)	5
Camera Tip	
X(mm)	0.000
Y(mm)	501.900
Z(mm)	644.500
Visible	False
Show View Ray	True
Show View Cent...	True
Near Plane	
Width(mm)	46
Height(mm)	34
Distance(mm)	100
Visible	True
Color	255, 255, 0
Fill	False
Far Plane	
Width(mm)	650
Height(mm)	486
Distance(mm)	1500
Visible	True
Color	0, 255, 0
Fill	True
Pixel Resolution	
Near X(mm)	0.072
Near Y(mm)	0.071
Far X(mm)	1.016
Far Y(mm)	1.013
Mount Type	Mobile
Robot	1
Joint	6
Offset Position	
X(mm)	0.000
Y(mm)	20.000
Z(mm)	60.000
Offset Rotation	
X(degree)	0.000
Y(degree)	-90.000
Z(degree)	-90.000
Position	
X(mm)	0.000
Y(mm)	435.000
Z(mm)	630.000
Rotation	
X(degree)	0.000
Y(degree)	-90.000
Z(degree)	-90.000
Visible	True
Show Label	False
Show Origin	False
Collision	
Collision Point	
Collision Margin	
Transparent	False
Transparency(%)	50

Property	Object	Value
Name	All	Displays the camera name.
Camera Number	All	Displays the system camera number.
Type	All	The camera type (Compact Vision, PC Vision) is displayed.
Connection Type	All	Displays camera connection type (GigE, USB).
Model	All	Displays a camera model.
Resolution	All	Displays a camera resolution.
Extension Tube	All	Displays a length of extension tube.
Lens Type	All	Displays a lens type.
Focal Length	All	Displays focal length of lens.
Show View Ray	All	Show / not show the view ray. ▪ Display: True (default) ▪ Not display: False
Show View Center	All	Show / not show the view center. ▪ Display: True (default) ▪ Not display: False

Camera Tip

Property	Object	Value
X, Y, Z	All	Displays the world coordinates on the edge of the camera lens. Change the value to change the camera position.
Visible	All	▪ Display: True (default) ▪ Not display: False

Near Plane/ Far Plane

Property	Object	Value
Width	All	Displays the camera view width.
Height	All	Displays the camera view height.
Distance	All	Displays distance of Camera Tip and Near Plane/ Far Plane.
Visible	All	Visible / not visible the depth of field. ▪ Display: True (default) ▪ Not display: False
Color	All	Sets the camera view color.

Property	Object	Value
Fill	All	Sets a fill of the camera view. In case of Near Plane: ▪ Display: True ▪ Not display: False (Default) In case of Far Plane: ▪ Display: True (default) ▪ Not display: False

Pixel Resolution

Property	Object	Value
Near X, Y	All	Displays the size of Near Plane in pixel.
Far X, Y	All	Displays the size of Far Plane in pixel.

Mount type

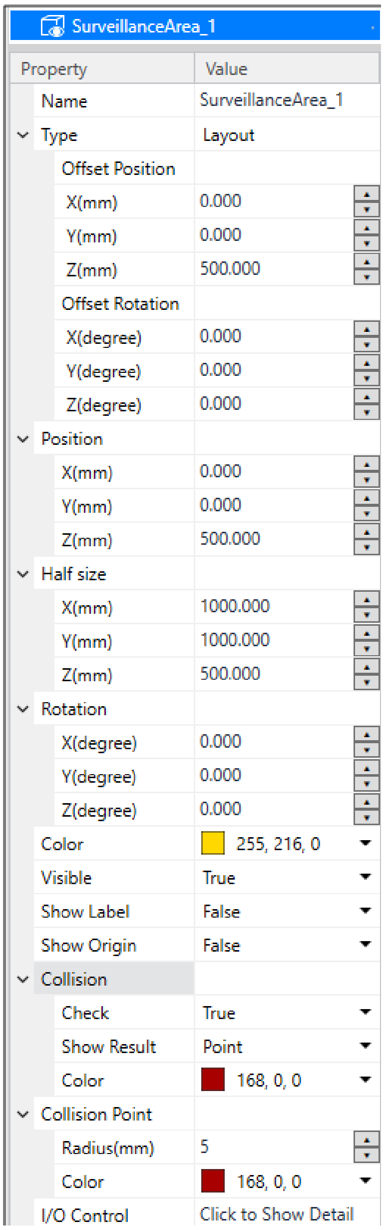
Property	Object	Value
Mount type	All	Displays the mount type of camera. The mount type (fixed camera, mobile camera) can be changed from [Setup]-[System Configuration]-[Vision]-[Camera] mounting settings.
Robot	Mobile	Displays the mounted robot number. The robot can be changed from the [Setup]-[System Configuration]-[Vision]-[Camera] mounting settings.
Joint	Mobile	Displays the mounted joint number. Joints (5, 6) can be changed from the [Setup]-[System Configuration]-[Vision]-[Camera] mounting settings.
Offset Position	Mobile	Display the relative position from the mounted joint. It is linked to the setting value of the camera's Position property or Joint property. Therefore, changes to these properties may cause the values to be updated automatically. Even in such cases, the values are correct as settings and do not affect the position of the camera in 3D.
Offset Rotation	Mobile	Display the relative orientation from the mounted joint. It is linked to the setting value of the camera's Rotation property or Joint property. Therefore, changes to these properties may cause the setting values to be updated automatically as follows: When Y is set at +90 degrees or -90 degrees, the X and Z settings are swapped. Even in such cases, the values are correct as settings and do not affect the orientation of the camera in 3D.

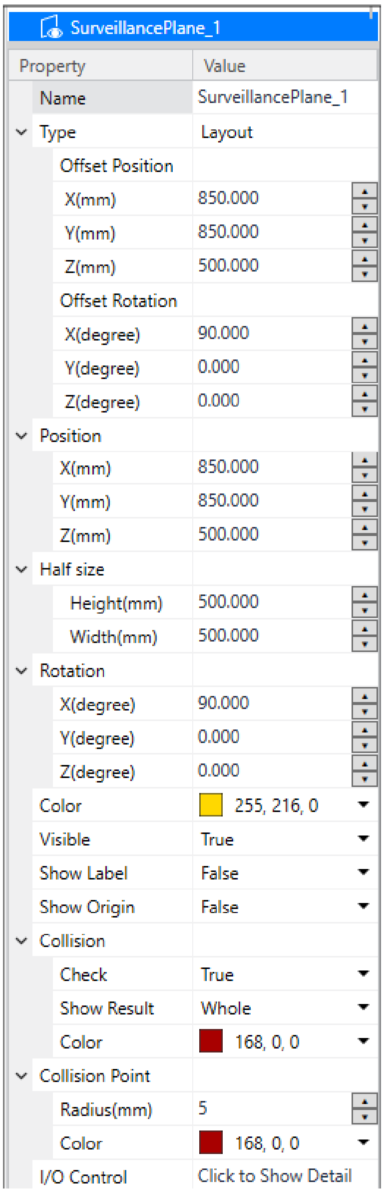
For properties such as "Collision", "Collision Margin", or "Transparent", refer to the layout objects properties.

9.3.1.4.4 Surveillance Objects

SurveillanceArea, SurveillancePlane

There are attributes that are enabled SurveillanceArea or SurveillancePlane or both.





Property	Value
Name	You can specify any name.
Type	Set to Layout. Cannot change the setting.
Offset Position	You can view/set the position of an object as an offset from its parent object. If the parent object does not exist, it will display/set the same as the Position property.
Offset Rotation	You can view/set the object's placement angle as an offset from the parent object. If the parent object does not exist, it will display/set the same as the Rotation property.
Position	Specifies a center point in simulator World coordinates.
Half size	Specifies a length from the center. The length is double this length.
Rotation	Specify the placement angle.
Color	You can change the display color. Click the drop-down ▾ to change it. The display color setting dialog will be displayed. Refer to Change layout object color for the details.

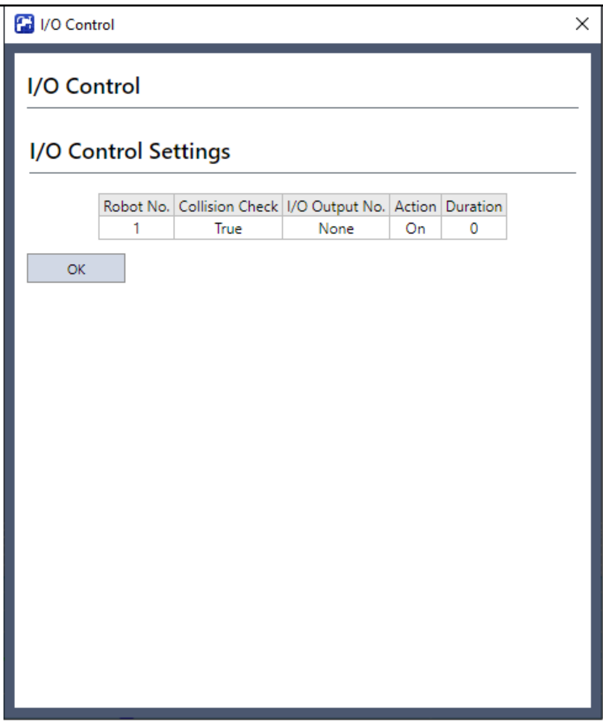
Property	Value
Visible	▪ Display: True (default) ▪ Not display: False
Show Label	▪ Displays the label: True ▪ Not display the label: False (default)
Show Origin	▪ Displays the origin coordinate system: True ▪ Not displays the origin coordinate system: False (default)

Collision

Property	Value
Check	Enables/disables the collision detection and I/O control setting. ▪ Enable: True (default) ▪ Disable: False
Show result	Specify how to display the color configured in Color property when the a collision is detected. ▪ Entirely: Whole (default) ▪ Collision point: Point ▪ Entire object and collision point: WholeAndPoint
Color	Specifies a color to be displayed when detecting a collision. Default: 168,0,0

Collision Point

Property	Value
Radius(mm)	Specify the radius of collision point displayed when the collision is detected.
Color	Specifies a color to be displayed when detecting a collision. Default: 168,0,0
I/O Control	Click [...].to display the [I/O Control Setting] dialog. You can select the following settings for each configured robot. ▪ Enables / disables the collision detection: Enabled (True), Disabled (False) ▪ I/O Output # ▪ On/Off Setting ▪ Duration (Time to continue On/Off)

<table>
<tr><th>Property</th><th>Value</th></tr>
<tr><td></td><td>


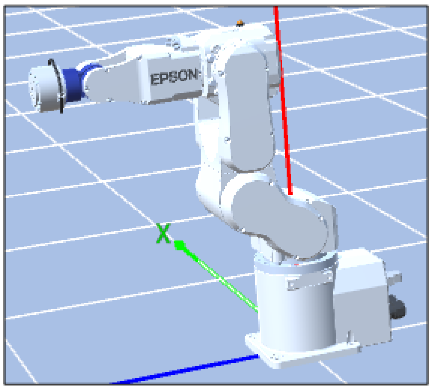
When a robot collides or contacts with an object, Output No. of the specified I/O is turned On or Off.
I/O output bit numbers already used in remote output signals cannot be used (are not displayed.) If a bit that has already been set is assigned to a remote output setting, the setting value is reset to None.
When setting the duration (On/Off duration), disable the setting that causes an error stop if a collision is detected during program execution (*). If set to Enable, it will not function even if a collision is detected, because it will be in Emergency Stop status.
</td></tr>
</table>

* See details below.

Collision detection

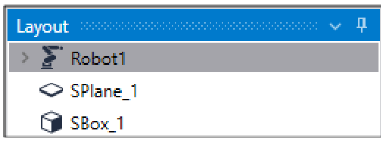
9.3.1.5 3D Display

In the 3D display, you can check the robot orientation and motion from various points of view.

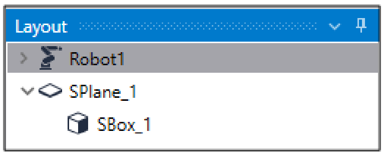


Adding a layout object

When a layout object is added while the robot object is selected in [Layout Objects], it will be added as independent object.



When a layout object is added while the layout object is selected in [Layout Objects], it will be added as a grouped object of the selected object.



Grouped objects move together when the parent object moves. RightTable/CenterTable/LeftTable of the sample virtual controller "C4 Sample" is an example of grouping.

Editing a layout object

[Cut], [Copy], and [Paste] commands in the [Edit] menu are available for layout objects other than CAD objects.

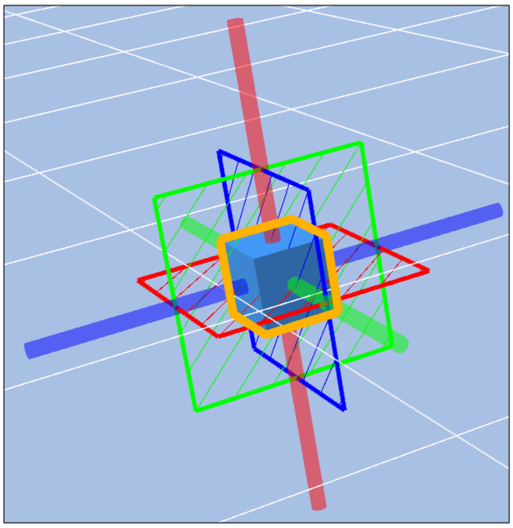
Changing a hierarchy of layout objects

To change a hierarchy of layout objects, drug and drop a layout object in the layout object.

Change the robot / layout object position

Guides that show the direction of movement can be displayed by clicking the  [Move] button on the toolbar and clicking an object such as a robot or box.

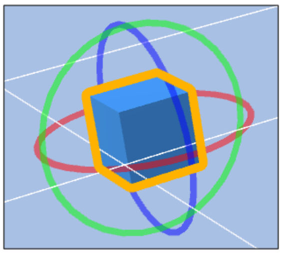
To move the object, drag the grid corresponding to the axis.



Rotate the robot / layout object

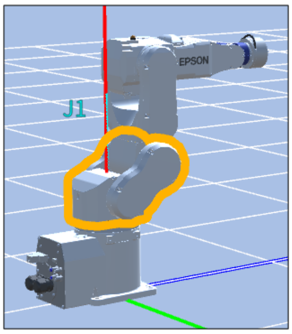
Guides that show the direction of rotation can be displayed by clicking the  [Rotate/Jog] button on the toolbar and clicking an object such as a robot base or box.

To rotate the object, drug the guide corresponding to the direction you want to rotate the object.



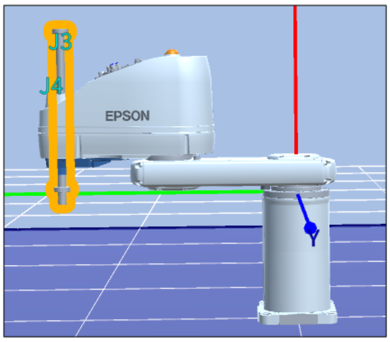
Move the robot joint

A robot joint can be moved by clicking the -  [Rotate/Jog] button on the toolbar and dragging the joint. The selected joint is displayed in yellow.



For SCARA robot:

- To move the shaft (J3): Click the toolbar  [Move] button and drag the robot shaft up or down.
- To rotate the shaft (J4): Click the toolbar  [Rotate/Jog] button and drag the robot shaft left or right.



Change the view point

To rotate the view point, press the mouse left button and drag the 3D display. To move the view point left, right, down, or up, press the right mouse button and drag the 3D display. You can also use the [W], [A], [S], [D], and cursor keys to move the view point. You can reset the view point from the menu opened by right click.

Zoom the layout

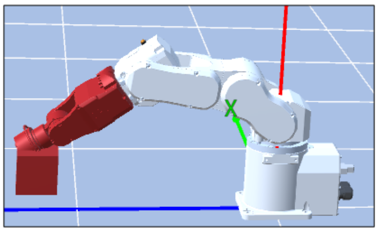
To zoom the 3D display, use the mouse wheel to scroll. You can change the zoom level from the menu opened by right clicking with the mouse.

Check for collisions

When a collision between a robot and layout object is detected, the collided robot joint and layout object are displayed in red.

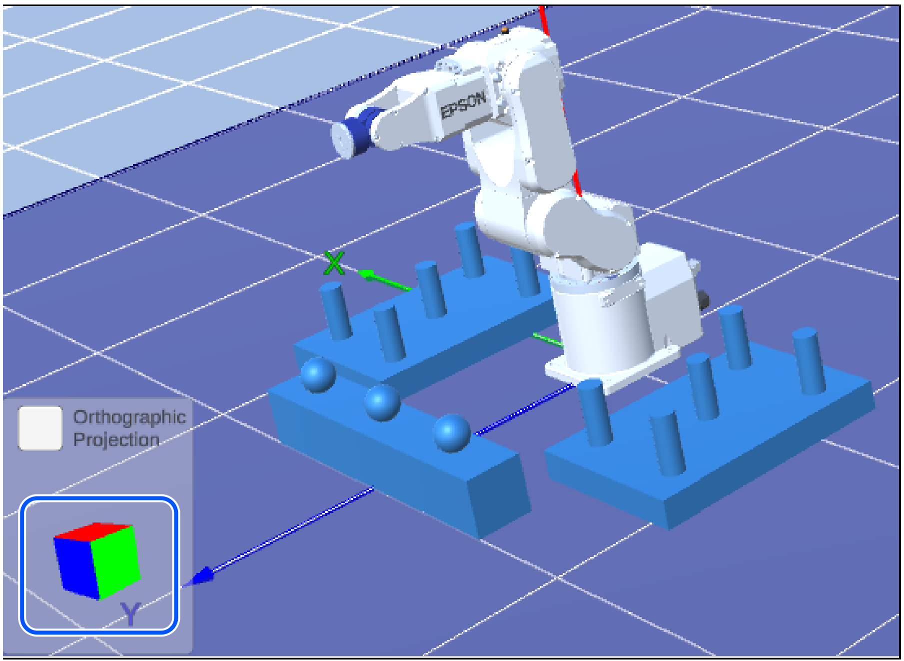
For details on collision detection, see below.

Collision detection

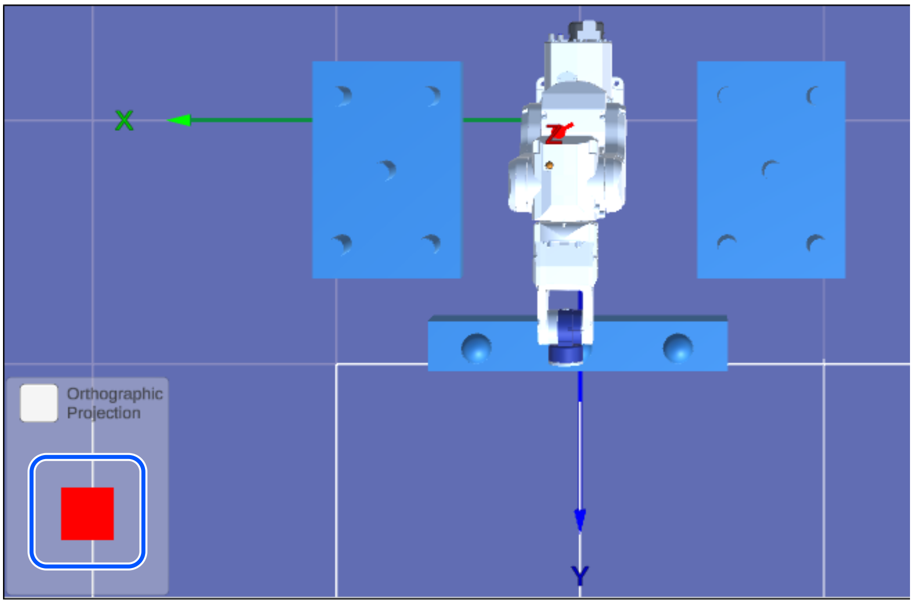


3D view control

The cube is located in the lower left corner of the 3D display. Dragging the cube with the mouse rotates the cube, which in turn rotates the viewpoint.

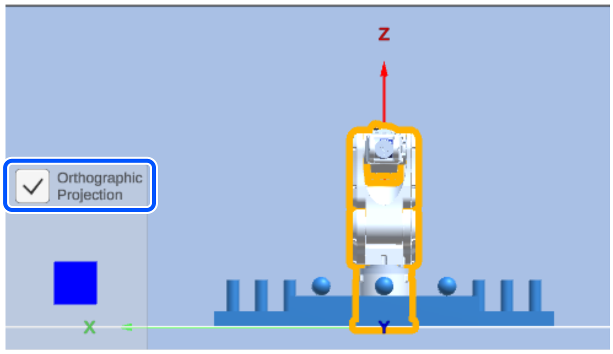


You can also click on a plane on the cube to reset the viewpoint to that plane.



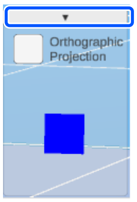
Orthographic projection

Check [Orthographic Projection] in the lower left corner of the 3D display to switch the 3D view to an orthographic projection (object size does not depend on perspective).



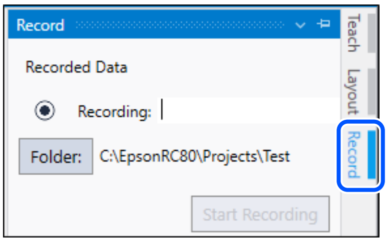
Minimization

You can minimize the display area of the 3D view operation cube and [Orthographic Projection] checkbox.

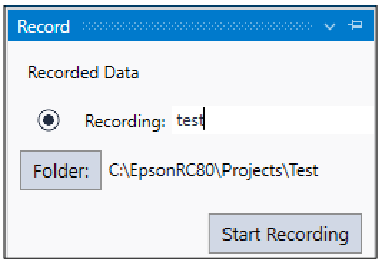


9.3.1.6 Record / Playback

Select [Record] from the right tab of the simulator window to display the Record panel. In Playback mode, you can record and produce simulation results.



Record (simulation mode)



Selecting the [Record] tab in the upper right corner of the simulator window allows you to set up simulation results for recording.

Item	Description
Log	Specifies a log file for recording.
Folder	Specifies a folder to save the log file.

Item	Description
Start Recording/Stop Recording	Starts and stops recording. Once the log file is specified and the [Start Recording] button is pressed, recording is ready.

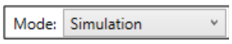
Playback (playback mode)

Item	Description
Reset Collision	Resets the collision detection status. When you click this button while the robot is not interfering with any layout objects, the red display turns to normal.
Simulator Settings	Shows the [Simulator Settings] dialog. In this dialog, the 3D [Render Options] can be configured.
Screenshot	Saves the current 3D display as an image file. The [Save As] dialog box appears, allowing you to specify a file name and file type (JPEG or PNG format).
Create Movie	Plays a simulation result (log file) and saves into a movie file in MP4 format. The [Save As] dialog box appears, allowing you to specify a file name.
00:00:00:000 Elapsed Time	Displays the run time of the log file being played.
Clear TCP path	Clear the TCP path (including Render Singularity Avoiding path) which the robot displays.
PLAY	Plays a simulation result of a specified log file.
STOP	Stops the simulation playback.

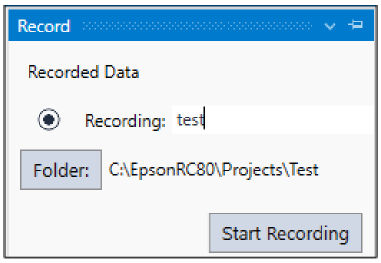
Outputting robot motion to log file

To reproduce the results of a simulation later, the robot's motion (action) must be output to a log file.

1. Confirm that the mode is “Simulation” mode on the Simulator toolbar.



2. Select the [Record] tab in the upper right corner of the simulator window to display the [Record] window.



3. Enter a name for [Log].

4. Click [Folder] to set the folder where the log will be saved.

5. Click [Start Recording].

   The icon next to the log file name changes from  to , and recording is ready. A message prompting program execution appears.

6. Run the program.

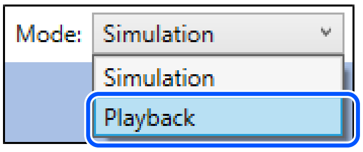
   When program execution is complete, recording will automatically end. When recording has ended, the icon next to the log file name changes from  to  and a log file with the specified name is generated in the specified folder.

KEY POINTS

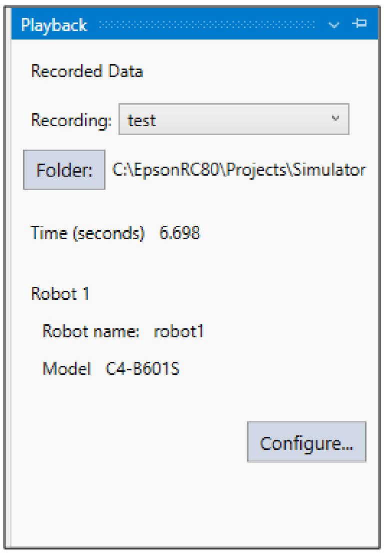
During recording, the [Start Recording] button changes to the [Stop Recording] button. To stop recording, press the [Stop Recording] button.

Reproducing robot motion from log file

1. From the simulator toolbar, change the mode from “Simulation” to “Playback”.

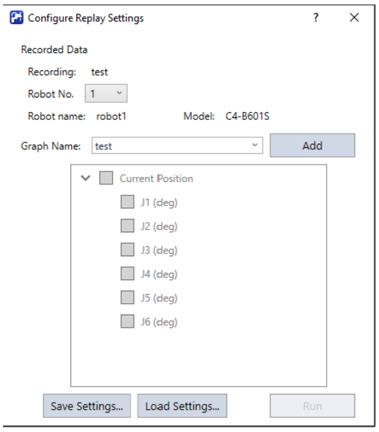


2. Select the [Playback] tab in the upper right corner of the simulator window to display the [Playback] window.



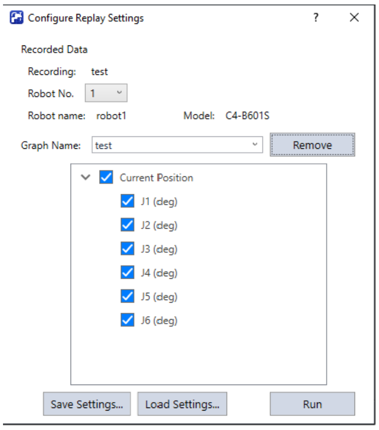
3. From the [Log] drop-down list, select the log you want to reproduce.

4. Click the [Configure] button. The configuration window will appear.

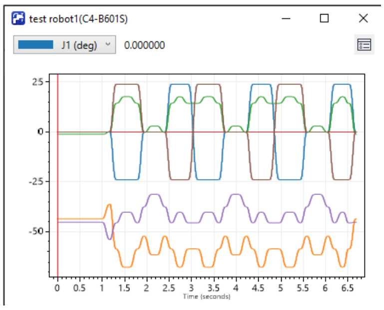


5. Select the target robot from [Robot No.].

6. The playback function graphically displays joint movements. Enter the [Graph Name] and click the [Add] button.

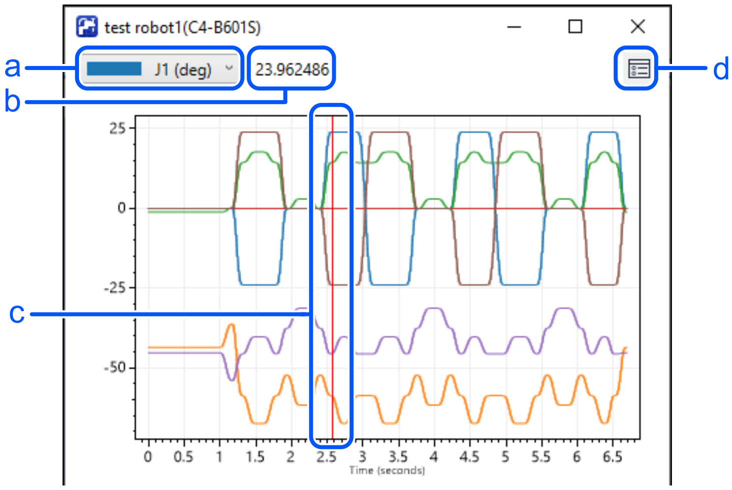


7. Select a joint to graph, and then click the [Run] button. A graph window appears.



8. Click the toolbar  [Playback] button to replay the logged operation.

Graph window functions



Symbol	Description
a	Select the joint whose value will be checked.
b	The selected joint's value is displayed.
c	Drag the vertical indicator with the mouse to adjust the playback position.
d	You can reset the window to its initial display or export data to a CSV file.

9.3.1.7 Create Movie

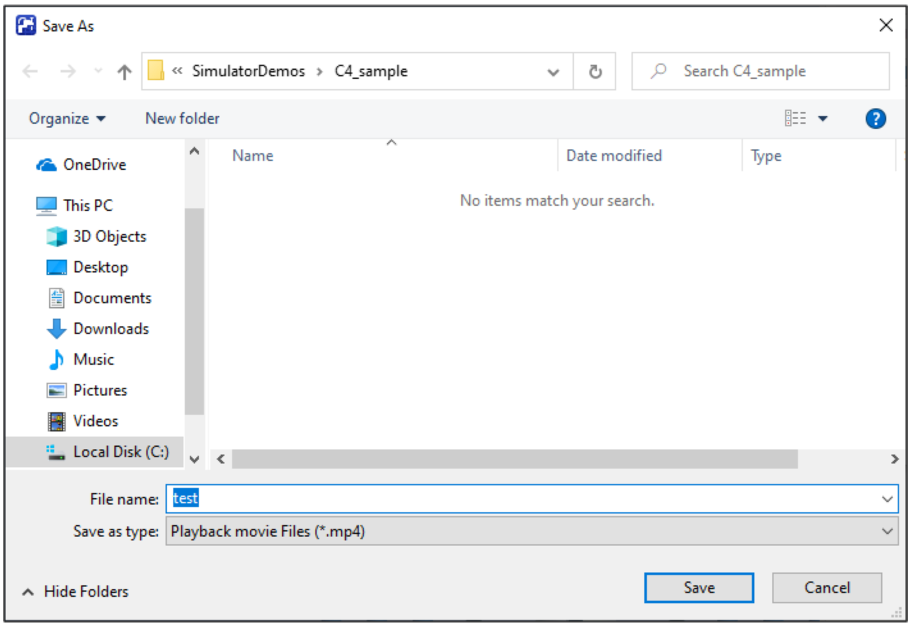
You can store the simulation results in movie files.

In simulation mode, you can execute projects and save movie files in MP4 format.

In playback mode, you can replay simulation results (log files) and save movie files in MP4 format.

Saving movie files

1. Click  [Create Movie] in the simulator toolbar.
2. The [Save As] dialog box appears. Specify the folder and file name where the movie will be saved.



3. Click the [Save] button.

4. In simulation mode, run the project. When project has finished executing, a movie file is created.

   In playback mode, the logged operation will be replayed. When playback ends or is stopped midway, a movie file is created.

KEY POINTS

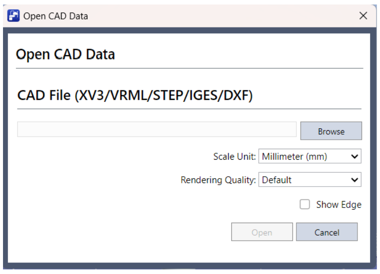
A movie file is created each time a project is executed or a logged action is played back. To exit movie creation, click [Create Movie] again.

9.3.1.8 Loading the CAD file

The CAD file can be loaded to lay out the hand or CAD object data in the 3D display. For details of available CAD data, see below.

Specifications and precautions for the 3D display - "Available CAD data for 3D display"

Pressing the  [CAD] button on the toolbar opens the [Open CAD Data] dialog box.



Item	Description
[Browse] button	Displays the dialog box for selecting the file. Select a CAD file to load.
Scale Unit	Select a unit of length used in the CAD data in order to match the unit with the simulator. This can be changed in the property grid after loading the data.
Rendering Quality	Specify the rendering quality. ▪ If "Fine" is selected, the data is displayed in details but it takes time. ▪ If "Fast" is selected, the details are not displayed (e.g. the screw holes are shown as a square), but the data can be displayed faster.
Show Edge	Selects whether to show edges (edge lines) in the display after opening (importing) CAD data. When checked, the ShowEdge property is True and the Edge is shown in the CAD data display.
[Open] button	Starts loading the data.

9.3.1.9 Saving the CAD file

The loaded CAD file can be converted to the XVL format to save. Converting the file to the XVL format can reduce the file size, resulting in a shorter loading time.

Saving CAD files

1. Select the CAD object to save in the layout object.
2. Click the [Click to Save] setting for the [Save as XVL...] property in the property grid.
3. The [Save As] dialog box appears. Click [Save].
4. If the file is saved successfully, the confirmation message appears. Click [Yes].

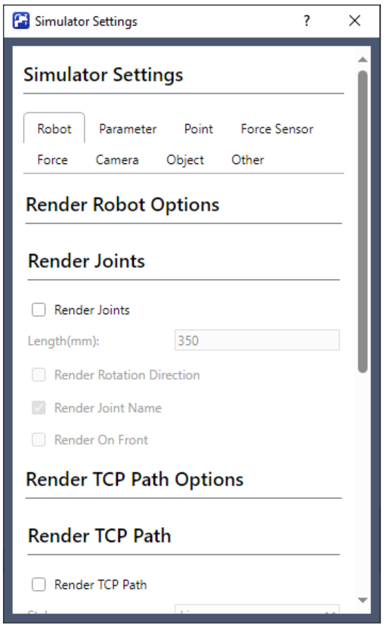
9.3.2 Simulator Settings

Pressing the [Simulator Settings] button displays the [Simulator Settings] dialog box.

The settings will be kept after restarting the Epson RC+ 8.0.

9.3.2.1 [Robot]

Settings related to 3D display of the robot.



Render Robot Options

Item	Description
Render Joints	Displays an arrow that shows a supporting point and a rotation axis of the robot.
- Length	Set a length for rendering the joints.
- Render Rotation Direction	Display a rotation direction when rendering the joints.
- Render Joint Name	Display the joint names (J1, J2, J3, J4, J5, J6) when rendering the joints.
- Render On Front	Display the arrow of "Render Joints" in front of the robot.

Render TCP Path Options

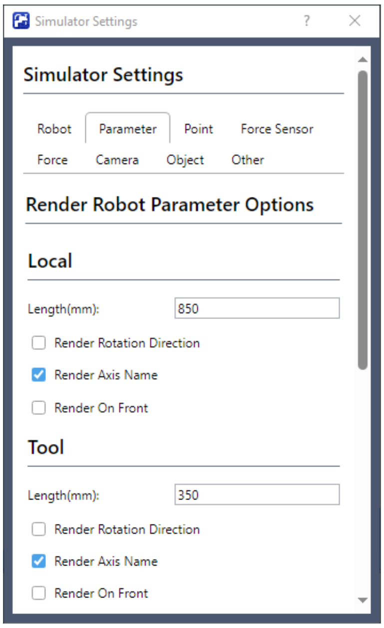
Item	Description
Render TCP Path	Displays the path of the origin point on active Tool coordinate system for a fixed time.

Item	Description
Render Singularity Avoiding path	Only displays the path of operation to avoid render singularity for the path of the origin point on active Tool coordinate system.
Style	Select line or dot to indicate the paths.
Width	Specify the line width of paths.
Radius	Specify the diameter of dots indicating paths.
Color	Specify the color of paths.

9.3.2.2 [Parameter]

Settings related to 3D display of robot parameters.

You can set the local coordinate system, tool coordinate system, ECP coordinate system, and plane origin coordinate system.

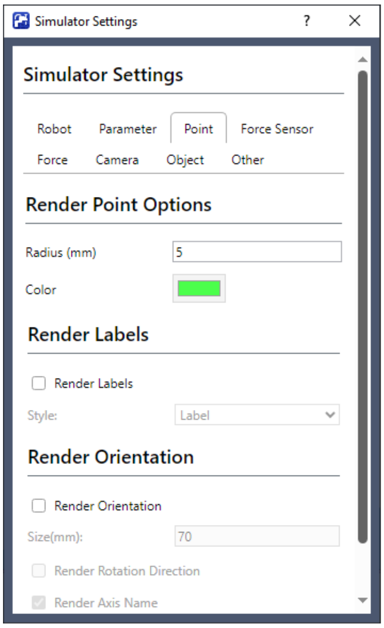


Render Robot Parameter Options

Item	Description
Length	Set a length of the coordinate axis.
Render Rotation Direction	Display the rotational direction of the coordinate axis.
Render Axis Name	Displays the coordinate axis names (X, Y, Z, U, V, W).
Render On Front	Display the arrow of "Render Robot Parameter" in front of the robot or the object.

9.3.2.3 [Point]

Settings related to 3D display of the point data of the robot.



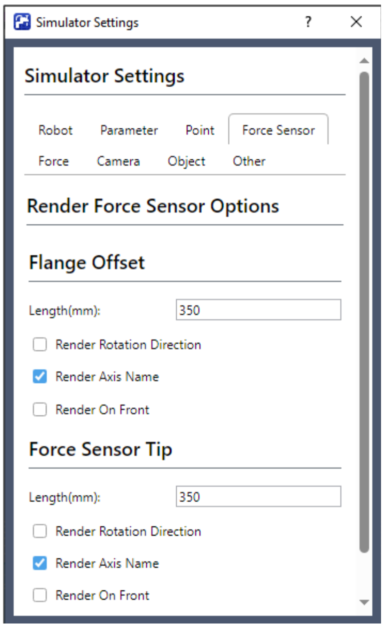
Render Point Options

Item	Description
Radius	Specify the diameter of dots indicating points.
Color	Specify the color of dots indicating points.
Render Labels	Display a point label.
Style	Set a display style when rendering the point label. ▪ Label: Label registered in point file. ▪ Number: Point number ▪ NumberAndLabel: Point number and label
Render Orientation	Display an orientation at the point as a coordinate axis.
Size	Set a size of coordinate axis that shows the orientation at the point.
Render Rotation Direction	Display the rotational direction on the coordinate axis that shows the orientation at the point.
Render Axis Name	Displays the coordinate axis name (X, Y, Z, U, V, W) of the coordinate axis that shows the orientation at the point.

9.3.2.4 [Force Sensor]

Settings related to 3D display of the force sensor.

You can set the coordinate system that shows a flange offset and tip of the force sensor.



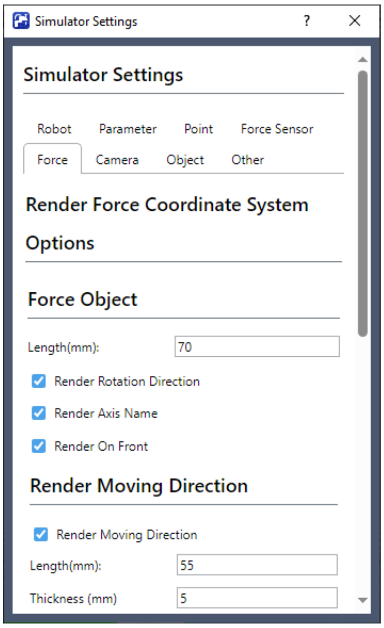
Render Force Sensor Options

Item	Description
Length	Set a length of the coordinate axis.
Render Rotation Direction	Display the rotational direction of the coordinate axis.
Render Axis Name	Displays the coordinate axis names (X, Y, Z, U, V, W).
Render On Front	Display the arrow of "Render Force Sensor" in front of the robot or the object.

9.3.2.5 [Force]

Settings related to 3D display of the force control data.

You can set the force coordinate system and moving direction for force object or force guide object.



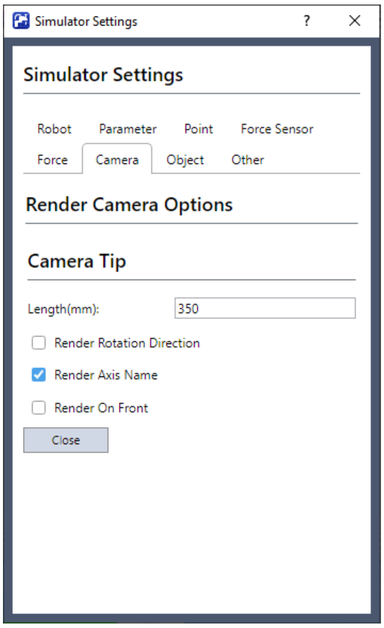
Render Force Coordinate System Options

Item	Description
Length	Set a length of the coordinate axis of the force coordinate system.
Render Rotation Direction	Display the rotational direction of the coordinate axis.
Render Axis Name	Displays the coordinate axis names (Fx, Fy, Fz, Tx, Ty, Tz).
Render On Front	Display the arrow of "Render Force Coordinate System" in front of the robot or the force sensor.
Render Moving Direction	Show the moving direction of the robot by the force control.
Length	Set a length of an arrow that shows the moving direction of the robot.
Thickness	Set a thickness of an arrow that shows the moving direction of the robot.
Color	Set a color of an arrow that shows the moving direction of the robot.
Render Labels	Show a label of the force object and force guide object name.

9.3.2.6 [Camera]

Settings related to 3D display of the camera object.

You can set the display of the coordinate system that shows the camera tip.



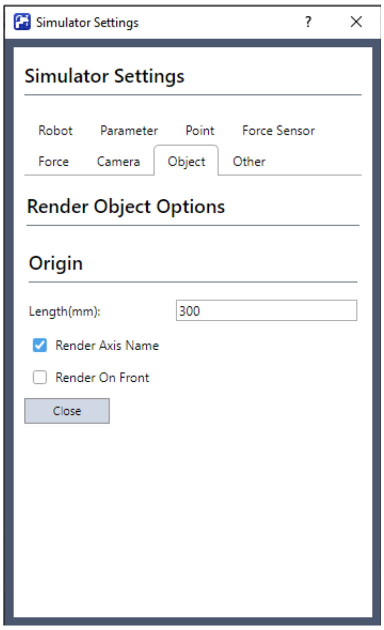
Render Camera Options

Item	Description
Length	Set a length of the coordinate axis.
Render Rotation Direction	Display the rotational direction of the coordinate axis.
Render Axis Name	Displays the coordinate axis names (X, Y, Z, U, V, W).
Render On Front	Display the arrow of "Render Camera" in front of the object.

9.3.2.7 [Object]

Settings related to 3D display of the general objects including robot, CAD object, and simple object.

You can set up the display of the coordinate system to show the origin of objects.

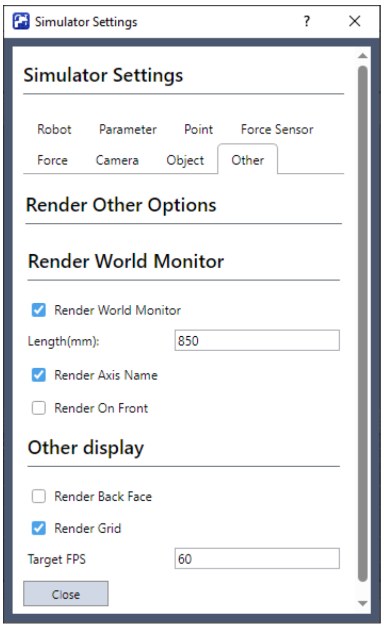


Render Object Options

Item	Description
Length	Set a length of the coordinate axis which shows an origin of the object.
Render Axis Name	Displays the coordinate axis names (X, Y, Z).
Render On Front	Display the arrow of "Render Object" in front of the object.

9.3.2.8 [Other]

Other settings related to 3D display.



Render Other Options

Item	Description
Render World Monitor	Displays the World coordinates.
Length	Set a length of the coordinate axis which shows a world coordinate.
Render Axis Name	Displays the coordinate axis names (X, Y, Z).
Render On Front	Display the arrow of "Render Other" in front of the object.
Render Back Face	Displays the surface of polygons.
Render Grid	Displays the grids on the simulator.
Target FPS	Sets 3D drawing performance. This is a target time of how long you want to take time to execute one 3D drawing (one frame). Depending on the model of computer you are using, the CPU may be overloaded and 3D drawing and RC+ operation may become slow. In these cases, setting a lower target FPS may improve the problem.

9.3.3 Part/Mounted Device Settings

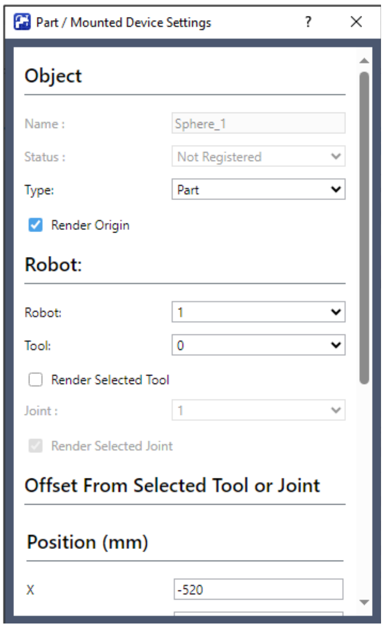
You can move the layout objects along with the robot like part such as workpieces grasped by the robot or devices mounted on the robot arm.

Set the layout objects for part/mounted device settings in the [Part/Mounted Device Settings] dialog. In the dialog, there are two methods to display

- Right-click the target object in the [Layout] panel. Select [Part/Mounted Device Settings] from the displayed context menu.
- If you click the [Down arrow] button shown in [Type] on the Property Grid and change it to Part or Mounted Device, the [Part/Mounted Device Settings] dialog will appear.

There are two methods to reset the Part/Mounted Device Settings and return the [Type] to "Layout".

- Click the [UnRegister] button on the [Part/Mounted Device Settings] dialog.
- Select the "Layout" on the [Type] property of property grid.



Item	Description
Type	Select from the following. ▪ Layout: Layout object (default) ▪ Part: Part object ▪ Mounted Device: Mounted device
Render Object Origin	Display the object origin.
Robot	Set a robot that relates to the selected object.
Tool	When the [Type] is "Part", set the tool coordinate system to place an object.
Render Selected Tool	Display the tool coordinate system.
Joint	When the [Type] is "Mounted Device", set joint number to place an object.
Render Selected Joint	Display joints.
Offset From Selected Tool or Joint	Set a relative position from the selected tool or joint.
Zero Clear	Set the offset value to "0.000".
Register	Register an object to the selected type.
UnRegister	Return the registered [Type] of the object to "Layout".
Cancel	Cancel the settings.

The Part/Mounted Device Settings can also be configured from the property grid.

From the property grid, you can configure the following settings.

Property	Value
Type	Sets the object type being set. Click ▾ and select the type from the drop-down list. ▪ Layout: Layout object (default) ▪ Part: Part object ▪ Mounted Device: Mounted device

Property	Value
Robot	Display the robot number to which the selected object is associated. If multiple robots are set, you can also change robots.
Tool	If the Type is "Part", the number of the tool coordinate system in which the object is placed is displayed. If multiple tool coordinate systems are set, the tool coordinate system can also be changed.
Joint	If the Type is "Mounted Device", the number of the joint where the object is placed is displayed. This can also be changed.
Offset Position	Display the position relative to the tool or joint being set. This can also be changed. It is linked to the setting value of the selected object's Position property, Tool property, or Joint property. Therefore, changes to these properties may cause the values to be updated automatically. Even in such cases, the values are correct as settings and do not affect the position of the object in 3D.
Offset Rotation	Display the orientation relative to the tool or joint being set. This can also be changed. It is linked to the setting value of the selected object's Rotation property, Tool property, or Joint property. Therefore, changes to these properties may cause the setting values to be updated automatically as follows: When Y is set at +90 degrees or -90 degrees, the X and Z settings are swapped. Even in such cases, the values are correct as settings and do not affect the orientation of the object in 3D.

9.3.4 Collision detection

In the simulation, collisions can be detected between the robots including its hand and the layout objects.

Here we describe the settings and details of collision detection.

9.3.4.1 Basic settings for collision detection

In the [Property Grid] of the robot, the following can be configured.

Collision Property	Value
Check	Enables / disables the collision detection for layout objects. ▪ Enable: True (default) ▪ Disable: False Even if this is enabled, it does not detect collision between the robot base and layout objects.
Check Self	Enables / disables the collision detection for a robot itself. ▪ Enable: True (default) ▪ Disable: False
Color	Specify color to be used when collision of the arms is detected. Default: 168,0,0

9.3.4.2 Target of collision detection

In the Property Grid of the layout objects, the following can be configured.

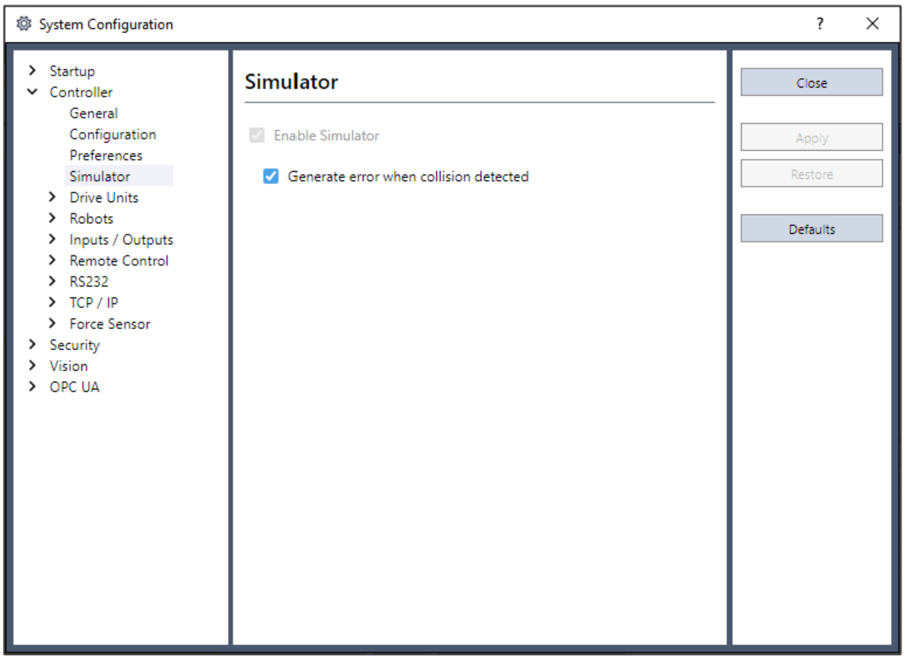
Collision

Property	Object	Value
Check	All	Enables / disables the collision detection. ▪ Enable: True (default) ▪ Disable: False Collision with the robot flange is not detected even when this property is set "True".
Show result	All	Specify how to display the color configured in Color property when the a collision is detected. ▪ Entirely: Whole (default) ▪ Collision point: Point ▪ Entire object and collision point: WholeAndPoint
Color	All	Specify the color to be used when collision of the arms is detected. Default: 168,0,0

9.3.4.3 Generate error when collision is detected

When you open the [Setup]-[System Configuration]-[Controller]-[Simulator] and check the [Generate error when collision detected] checkbox, if a collision is detected during SPEL+ program execution, an error occurs in the controller and the program stops.

After changing the checkbox and clicking the [Apply] button, click the [Close] button.



KEY POINTS

- The purpose of this function is to find where the program has a problem. It is not to prevent the collision of robots.
- It cannot guarantee that it has enough time for robots to stop when the simulator detects the collision.

9.3.4.4 Caution about the collision detection of Floor / Wall

A collision is detected when a floor or wall is in contact with the robot.

If the robot or plane positions are changed so that the robot passes completely through the plane, then no collision is detected.

9.3.4.5 Accuracy of collision detection

The collision detection in the simulator cannot guarantee accuracy. Make sure to have a margin when you apply the simulation result for a real robot system.

9.3.4.6 Caution about the CAD data

Collision cannot be detected when the CAD data has only the wire frame models. To use the collision detection function, add surface to the CAD data.

For restrictions on the simulator, see below.

Simulator Specifications and Restrictions

9.3.5 CAD To Point (6-axis Robot)

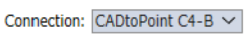
CAD To Point outputs the edge (edge line) information included in the CAD data as a point data. This function enables the user to generate a point data according to the path by sequentially selecting the edges of the CAD data shown in the 3D display. Since this function automatically registers points of manipulator motion based on the CAD data of the workpiece, it can save time to develop a program.

Follow the simple CAD data sample below to use CAD To Point. In this example, a motion in which the tip of a syringe traces a periphery of the CAD object (tray) will be created.

Follow the steps below:

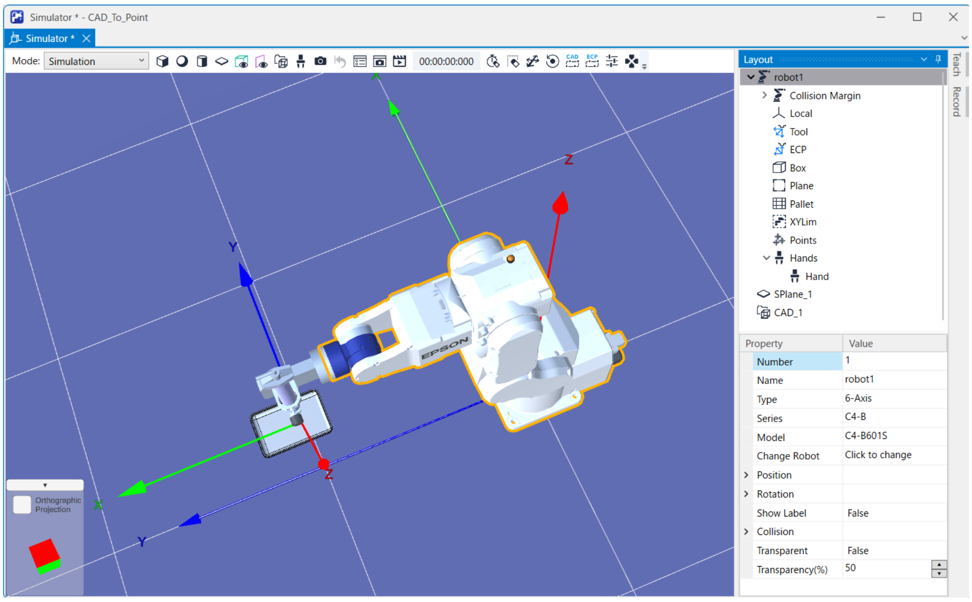
- Connect to the virtual controller (CADtoPoint C4-B)
- Open a project
- Select edges of the CAD object to generate a motion path
- Export the edges as point data
- Create a program
- Operate the robot by executing the program

1. Connect to the virtual controller (CADtoPoint C4-B)

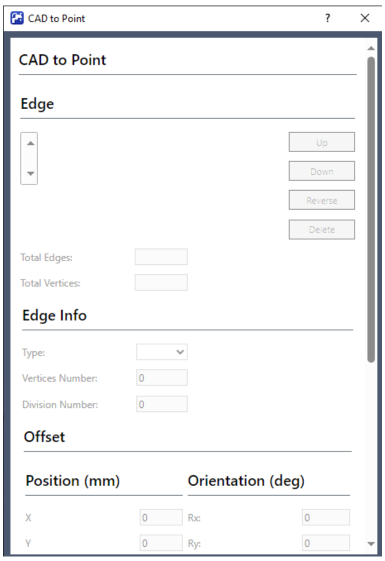
   Select "CADtoPoint C4-B" from the Epson RC+ 8.0 toolbar [Current controller connection] list box. When the connection is completed, "CADtoPoint C4-B" will be displayed in [Current controller connection] list box.

   Click the [Simulator] button on the toolbar to display the Simulator window. CADtoPoint C4-B contains CAD objects: "Work" and the Hand are placed.

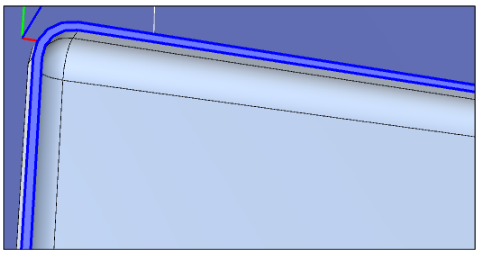


2. Open a project
   i. Click [Open...] from [Project] on Epson RC+ 8.0 menu.
   ii. Select [Projects]-[SimulatorDemos]-[CAD_To_Point_C4_B].
   iii. Click the [Open] button.

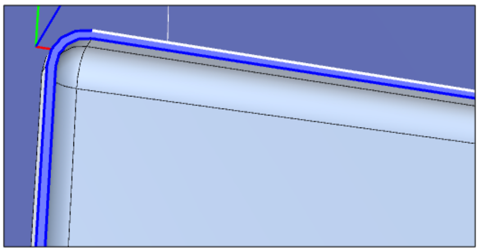
3. Select an edge of CAD object and create a motion path of the robot
   i. a. Click the CAD [CAD to Point] button on the toolbar to display the [CAD to Point] dialog box.



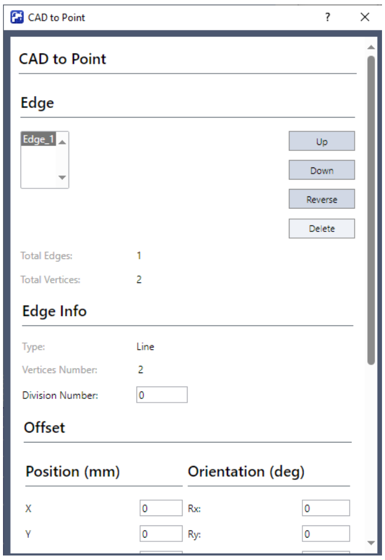
   ii. Hover the mouse over the CAD object and select a part having the edges. The selected part turns to be light blue and the edges are shown in blue.



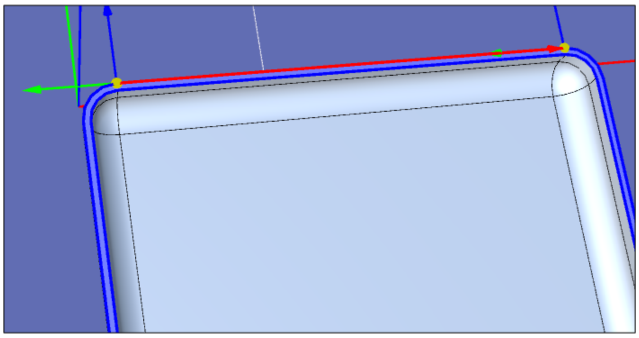
iii. Hover the mouse over a desired blue edge. The selected edge turns to be white. Select the line first. This sample program will not operate properly when selecting the curve first since this is designed to select the line first.



iv. Click the white edge. The selected edge will be shown in the [Edge] box of the [CAD to Point] dialog box.

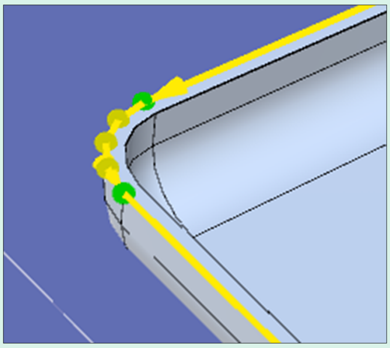
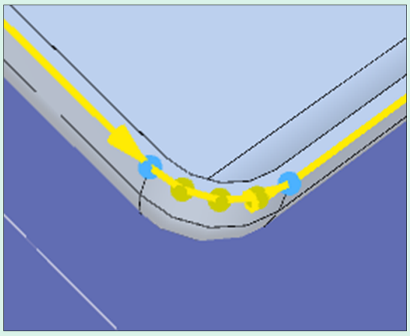


v. In the 3D View, the selected edge is indicated with a red arrow.

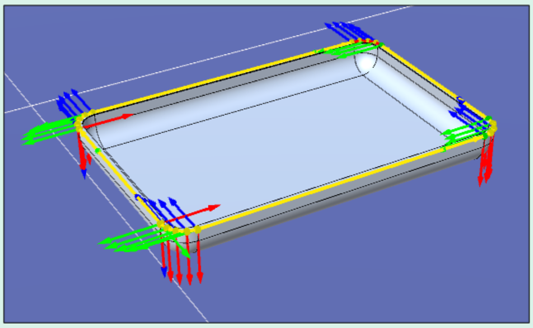


TIP

- The arrow indicates the direction from the starting point to the end point. The direction of the arrow can be reversed by clicking the [Reverse] button.
- If the start and end points of a consecutive edge with the same moving direction are piled up, the vertex color changes. When the both position (X,Y,Z) and orientation (U,V,W) match, the vertex is displayed in green. When only the position (X,Y,Z) matches, the vertex is displayed in light blue

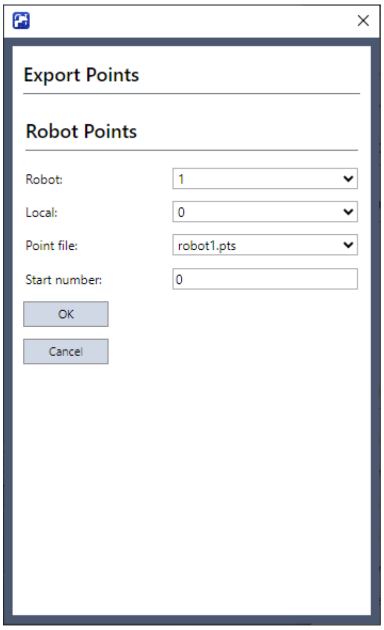



- The image will be shown as below after you select the edges one by one in counterclockwise to trace the periphery.

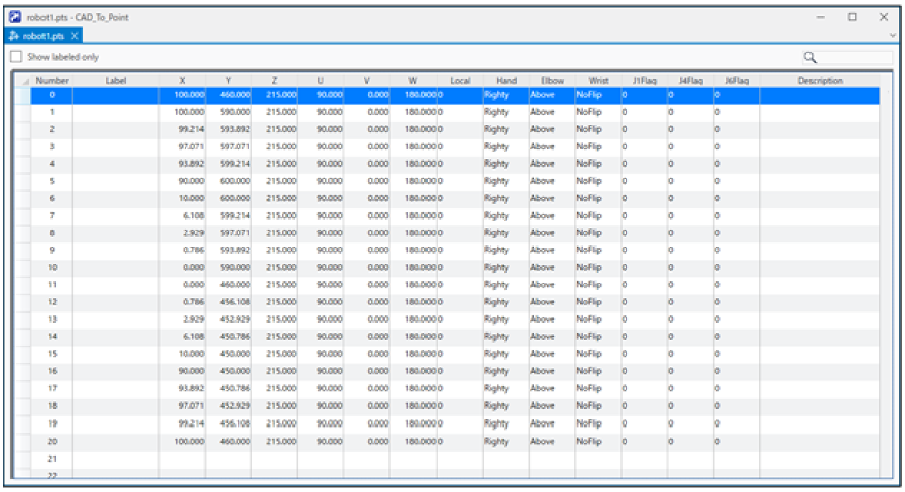


4. Export the edges as point data

Click the [Export] button on the [CAD to Point] dialog to display the [Export Points] dialog.



Click the [OK] button to output the point data to rows No. 0-20 in point file "robot1.pts".



robot1.pts - CAD_To_Point

robot1.pts

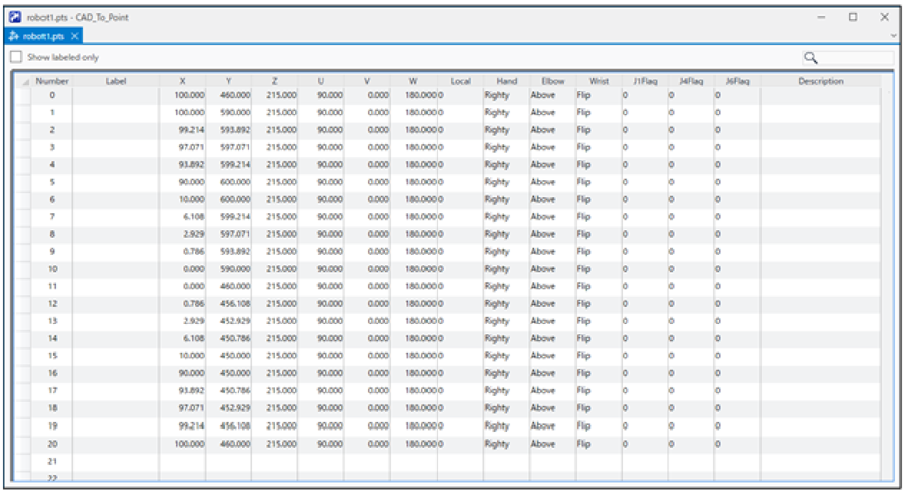
Show labeled only

Number	Label	X	Y	Z	U	V	W	Local	Hand	Elbow	Wrist	J1Flag	J4Flag	J6Flag	Description
0		100.000	460.000	215.000	90.000	0.000	180.000	0	Righty	Above	NoFlip	0	0	0	
1		100.000	590.000	215.000	90.000	0.000	180.000	0	Righty	Above	NoFlip	0	0	0	
2		99.214	593.892	215.000	90.000	0.000	180.000	0	Righty	Above	NoFlip	0	0	0	
3		97.071	597.071	215.000	90.000	0.000	180.000	0	Righty	Above	NoFlip	0	0	0	
4		93.892	599.214	215.000	90.000	0.000	180.000	0	Righty	Above	NoFlip	0	0	0	
5		90.000	600.000	215.000	90.000	0.000	180.000	0	Righty	Above	NoFlip	0	0	0	
6		10.000	600.000	215.000	90.000	0.000	180.000	0	Righty	Above	NoFlip	0	0	0	
7		6.108	599.214	215.000	90.000	0.000	180.000	0	Righty	Above	NoFlip	0	0	0	
8		2.929	597.071	215.000	90.000	0.000	180.000	0	Righty	Above	NoFlip	0	0	0	
9		0.786	593.892	215.000	90.000	0.000	180.000	0	Righty	Above	NoFlip	0	0	0	
10		0.000	590.000	215.000	90.000	0.000	180.000	0	Righty	Above	NoFlip	0	0	0	
11		0.000	460.000	215.000	90.000	0.000	180.000	0	Righty	Above	NoFlip	0	0	0	
12		0.786	456.108	215.000	90.000	0.000	180.000	0	Righty	Above	NoFlip	0	0	0	
13		2.929	452.929	215.000	90.000	0.000	180.000	0	Righty	Above	NoFlip	0	0	0	
14		6.108	450.786	215.000	90.000	0.000	180.000	0	Righty	Above	NoFlip	0	0	0	
15		10.000	450.000	215.000	90.000	0.000	180.000	0	Righty	Above	NoFlip	0	0	0	
16		90.000	450.000	215.000	90.000	0.000	180.000	0	Righty	Above	NoFlip	0	0	0	
17		93.892	450.786	215.000	90.000	0.000	180.000	0	Righty	Above	NoFlip	0	0	0	
18		97.071	452.929	215.000	90.000	0.000	180.000	0	Righty	Above	NoFlip	0	0	0	
19		99.214	456.108	215.000	90.000	0.000	180.000	0	Righty	Above	NoFlip	0	0	0	
20		100.000	460.000	215.000	90.000	0.000	180.000	0	Righty	Above	NoFlip	0	0	0	
21															

5. Create a program

   i. Set the proper robot orientation for the point data.

      Open the point file "robot1.pts" from the layout object, and change the wrist orientation (Wrist) of the exported No. 0-20 from "NoFlip" to "Flip".



robot1.pts - CAD_To_Point

robot1.pts

Show labeled only

Number	Label	X	Y	Z	U	V	W	Local	Hand	Elbow	Wrist	J1Flag	J4Flag	J6Flag	Description
0		100.000	460.000	215.000	90.000	0.000	180.000	0	Righty	Above	Flip	0	0	0	
1		100.000	590.000	215.000	90.000	0.000	180.000	0	Righty	Above	Flip	0	0	0	
2		99.214	593.892	215.000	90.000	0.000	180.000	0	Righty	Above	Flip	0	0	0	
3		97.071	597.071	215.000	90.000	0.000	180.000	0	Righty	Above	Flip	0	0	0	
4		93.892	599.214	215.000	90.000	0.000	180.000	0	Righty	Above	Flip	0	0	0	
5		90.000	600.000	215.000	90.000	0.000	180.000	0	Righty	Above	Flip	0	0	0	
6		10.000	600.000	215.000	90.000	0.000	180.000	0	Righty	Above	Flip	0	0	0	
7		6.108	599.214	215.000	90.000	0.000	180.000	0	Righty	Above	Flip	0	0	0	
8		2.929	597.071	215.000	90.000	0.000	180.000	0	Righty	Above	Flip	0	0	0	
9		0.786	593.892	215.000	90.000	0.000	180.000	0	Righty	Above	Flip	0	0	0	
10		0.000	590.000	215.000	90.000	0.000	180.000	0	Righty	Above	Flip	0	0	0	
11		0.000	460.000	215.000	90.000	0.000	180.000	0	Righty	Above	Flip	0	0	0	
12		0.786	456.108	215.000	90.000	0.000	180.000	0	Righty	Above	Flip	0	0	0	
13		2.929	452.929	215.000	90.000	0.000	180.000	0	Righty	Above	Flip	0	0	0	
14		6.108	450.786	215.000	90.000	0.000	180.000	0	Righty	Above	Flip	0	0	0	
15		10.000	450.000	215.000	90.000	0.000	180.000	0	Righty	Above	Flip	0	0	0	
16		90.000	450.000	215.000	90.000	0.000	180.000	0	Righty	Above	Flip	0	0	0	
17		93.892	450.786	215.000	90.000	0.000	180.000	0	Righty	Above	Flip	0	0	0	
18		97.071	452.929	215.000	90.000	0.000	180.000	0	Righty	Above	Flip	0	0	0	
19		99.214	456.108	215.000	90.000	0.000	180.000	0	Righty	Above	Flip	0	0	0	
20		100.000	460.000	215.000	90.000	0.000	180.000	0	Righty	Above	Flip	0	0	0	
21															

   ii. Create the following program in Main.prg program.

```
Function main

  Motor On
  TLSet 1, XY(-112, -41, 80, 0, -90, 0)
  Tool 1

  Go P0
  Move P1 CP
  Arc P3, P5 CP
  Move P6 CP
  Arc P8, P10 CP
  Move P11 CP
  Arc P13, P15 CP
  Move P16 CP
  Arc P18, P20 CP
```

```
  Pulse 0, 0, 0, 0, 0, 0
  Motor Off
Fend
```

> **TIP**
>
> By using Tool coordinate system 1, the tip of the syringe can trace outlines of workpiece.

iii. Click the [Build] button on the toolbar. The program will be built.

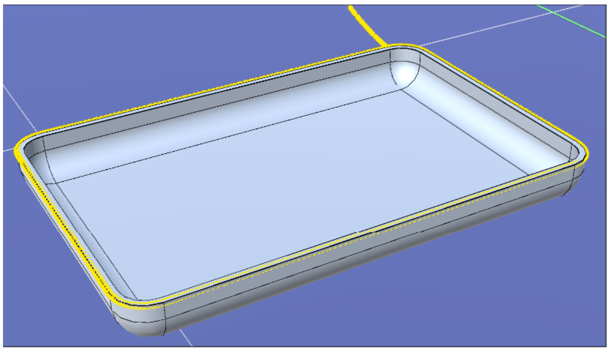
When the build is complete, the message "Build complete, no errors" appears in the status window.

6. Operate the robot by executing the program

i. Click the [Run Window] button on the toolbar to display the Run window.

ii. Click the [Start] button. When the message "Are you ready to start?" appears, click [Yes].

iii. The program will be executed. Check that the manipulator moves from P0 to P20 sequentially and the tip of the syringe traces the edge of Work in counterclockwise direction.



9.3.6 CAD To Point (SCARA Robot)

This section describes CAD to Point for a SCARA robot.

In this example, a motion in which the tip of a syringe traces a periphery of the CAD object (tray) will be created in the same way as a 6-axis robot.

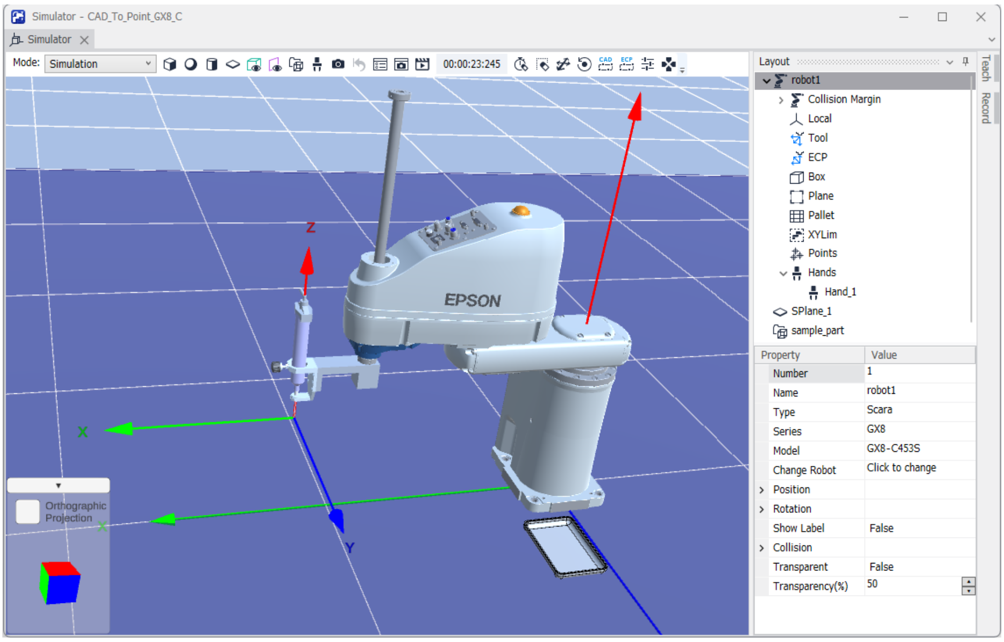
Operate the following procedures:

- Connect to the virtual controller (CADtoPoint GX8-C)
- Open a project
- Select edges of the CAD object to generate a motion path
- Export the edges as point data
- Create a program
- Operate the robot by executing the program

1. Connect to the virtual controller (CADtoPoint GX8-C)

Select "CADtoPoint GX8-C" from the Epson RC+ 8.0 toolbar [Current controller connection] list box. When the connection is completed, "CADtoPoint GX8-C" will be displayed in [Current controller connection] list box.

Click the [Simulator] button on the toolbar to display the Simulator window. "CADtoPoint GX8-C" contains CAD objects: "sample_part" and and Hand objects are placed.

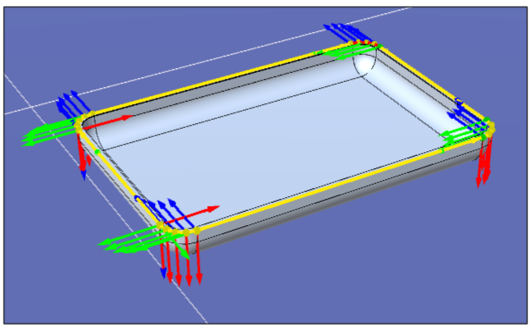


2. Open a project
   i. Click [Open...] from [Project] on Epson RC+ 8.0 menu.
   ii. Select [Projects]-[SimulatorDemos]-[CAD_To_Point_GX8_C].
   iii. Click the [Open] button.
3. Select an edge of CAD object and create a motion path of the robot

   For selection of surface including edges and edges, see below.

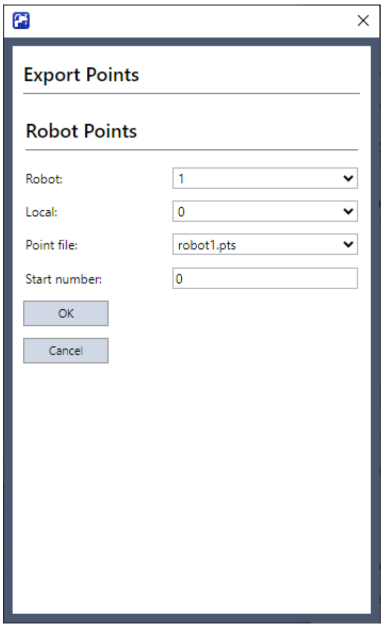
   CAD To Point (6-axis Robot)

   The image will be shown as below after you select the edges one by one in counterclockwise to trace the periphery.

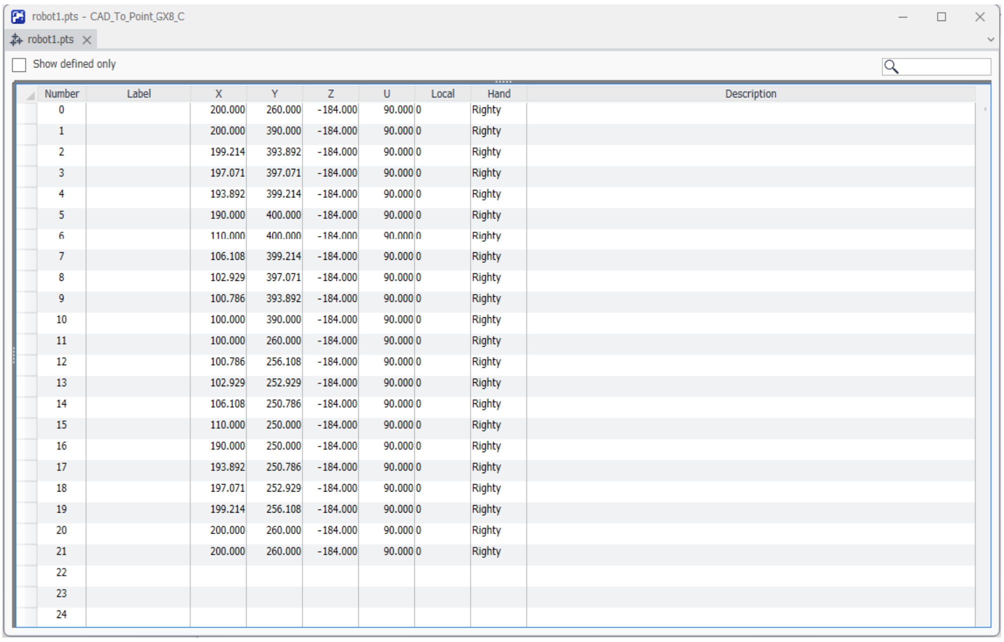


4. Export the edges as point data

   Click the [Export] button on the [CAD to Point] dialog to display the [Export Points] dialog.



Click the [OK] button to output the point data to rows No. 0-21 in point file "robot1.pts".



Number	Label	X	Y	Z	U	Local	Hand	Description
0		200.000	260.000	-184.000	90.000	0	Righty	
1		200.000	390.000	-184.000	90.000	0	Righty	
2		199.214	393.892	-184.000	90.000	0	Righty	
3		197.071	397.071	-184.000	90.000	0	Righty	
4		193.892	399.214	-184.000	90.000	0	Righty	
5		190.000	400.000	-184.000	90.000	0	Righty	
6		110.000	400.000	-184.000	90.000	0	Righty	
7		106.108	399.214	-184.000	90.000	0	Righty	
8		102.929	397.071	-184.000	90.000	0	Righty	
9		100.786	393.892	-184.000	90.000	0	Righty	
10		100.000	390.000	-184.000	90.000	0	Righty	
11		100.000	260.000	-184.000	90.000	0	Righty	
12		100.786	256.108	-184.000	90.000	0	Righty	
13		102.929	252.929	-184.000	90.000	0	Righty	
14		106.108	250.786	-184.000	90.000	0	Righty	
15		110.000	250.000	-184.000	90.000	0	Righty	
16		190.000	250.000	-184.000	90.000	0	Righty	
17		193.892	250.786	-184.000	90.000	0	Righty	
18		197.071	252.929	-184.000	90.000	0	Righty	
19		199.214	256.108	-184.000	90.000	0	Righty	
20		200.000	260.000	-184.000	90.000	0	Righty	
21		200.000	260.000	-184.000	90.000	0	Righty	
22								
23								
24								

5. Create a program

   i. Create the following program in Main.prg program.

```
Function main

  Speed 50, 50, 50
  Accel 100,100,100,100,100,100
  SpeedS 100
  AccelS 400

  TLSet 1, XY(-131, 0, -87, 0, 0, 0)
  Tool 1
  Go P0 :Z(-100)
  SpeedS 50
```

```
    Go P0
    Move P1 CP
    Arc P3, P5 CP
    Move P6 CP
    Arc P8, P10 CP
    Move P11 CP
    Arc P13, P15 CP
    Move P16 CP
    Arc P18, P20 CP

    Go P0 :Z(-100)

    Pulse 0, 0, 0, 0, 0, 0
    Motor Off

Fend
```

> **TIP**
>
> By using Tool coordinate system 1, the tip of the syringe can trace outlines of workpiece.

ii. Click the [Build] button on the toolbar. The program will be built.

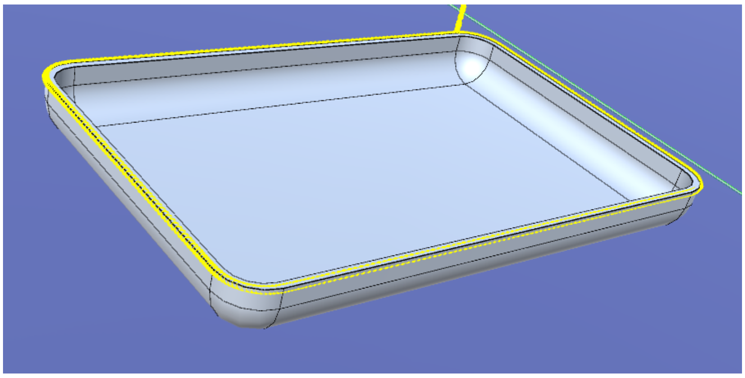
When the build is complete, the message "Build complete, no errors" appears in the status window.

6. Operate the robot by executing the program

i. Click the [Run Window] button on the toolbar to display the Run window.

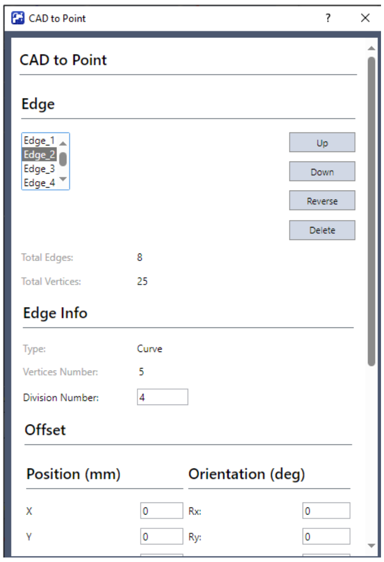
ii. Click the [Start] button. When the message "Are you ready to start?" appears, click [Yes].

iii. The program will be executed. Check that the manipulator moves from P0 to P21 sequentially and the tip of the syringe traces the edge of sample_work in counterclockwise direction.



9.3.7 Function of CAD to Point

Press the [CAD to Point] button on the toolbar to display the [CAD to Point] dialog box.



Edge

Item	Description
[Up] button	Move the order of the selected edge to up.
[Down] button	Move the order of the selected edge to down.
[Reverse] button	Switch the start point and end point of the selected edge. Red arrow on the edge indicates the direction from start point to end point.
[Delete] button	Delete the selected edge.

Edge Information

Item	Description
Type	Display the type of the selected edge. Types are Line, Curve, and Composite Curve.
Vertex Number	Display the number of vertex of selected edge. Increase or decrease division number to increase or decrease the number of vertex.
Division Number	Set the division number of the selected edge.

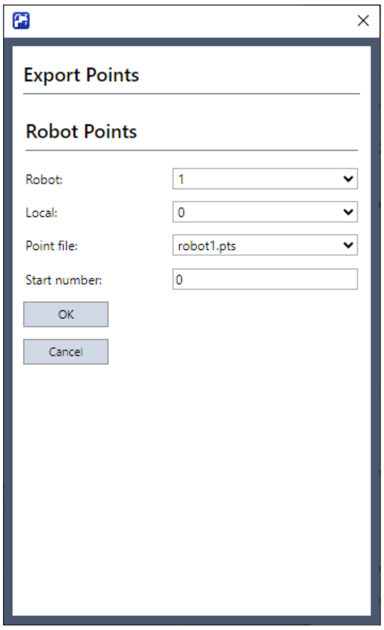
Offset

Item	Description
Pos.	Enable to move the vertex position to X, Y, and Z direction. Refer to the coordinate system of CAD object for the directions. Also, enable to expand or decrease from the center of the arc to radius direction if the edge is curve or composite curve which can approximate to the arc.
Orientation	Enable to rotate tool orientation to Rx, Ry, and Rz direction. Refer to the coordinate system of CAD object for the directions.

Render Option

Item	Description
Render Approach	Display the Z-axis (red arrow) of the render orientation to the vertex. It is useful when Z-axis of the render orientation cannot be seen due to workpieces.
Render Orientation	Display the orientation on the vertex. Only displays when the vertex is displayed.
Render CAD Origin	Display the origin of CAD object in coordinate system.
Render Vertex	Display the vertex of the selected edge.
Vertex Size	Set the vertex size.
Arrow Size	Change an arrow size that is directed to start point to end point of the selected edge.

Click the [Export Points] button on the [CAD to Point] dialog to display the [Export Points] dialog.



Point output

Item	Description
Robot	Set the robot to output the points.
Local	Set the local coordinate to output the points.
Point File	Set the point file to output the points.
Start Number	Set the start number of outputting points.

9.3.8 CAD to Point for ECP (6-axis Robot)

CAD To Point for ECP is a function to output edge line information which is included in CAD data as point data to operate external control point (ECP) motion. You can use the ECP function with the virtual controller. If you want to use it with the actual controller, you will need to purchase ECP license. By the robot grasps workpiece and selecting edges of CAD objects on the 3D view sequentially, you can generate point data along with the motion path. The robot motion points can be registered automatically based on CAD data such as part. Therefore, development time of programs can be shortened.

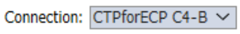
Use a sample using easy CAD data to execute CAD To Point for ECP.

In the example, create a motion to follow an outer circumference of grasped CAD object (tray) by the robot on the edge of fixed syringe.

Operate the following procedures:

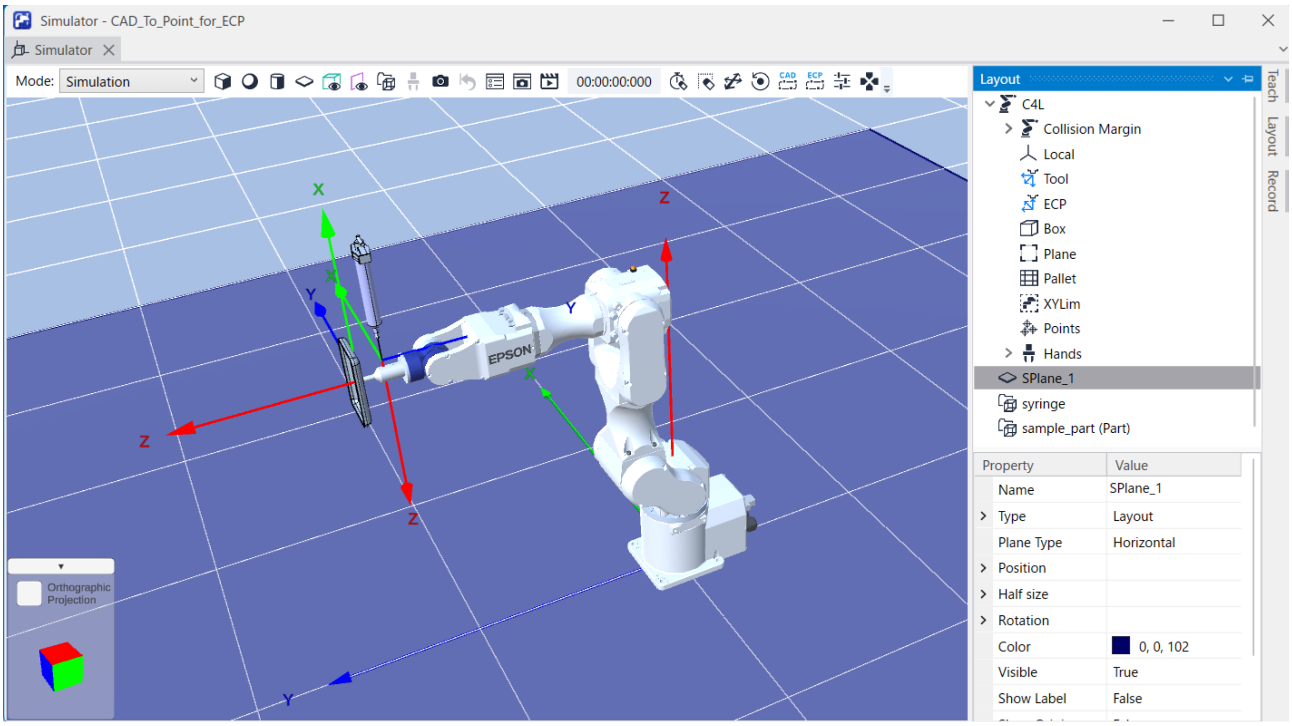
- Connect to virtual controller "CTPforECP C4-B"
- Open a project
- Select CAD Object and ECP
- Select an edge of CAD object and create a motion path of the robot
- Output as point data
- Create a program
- Operate the robot by executing the program

1. Connect to virtual controller "CTPforECP C4-B"



Select "CTPforECP C4-B" of [Connection:] on Epson RC+ 8.0 toolbar. When the connection is complete, "CTPforECP C4-B" is displayed in the [Connection:] box.

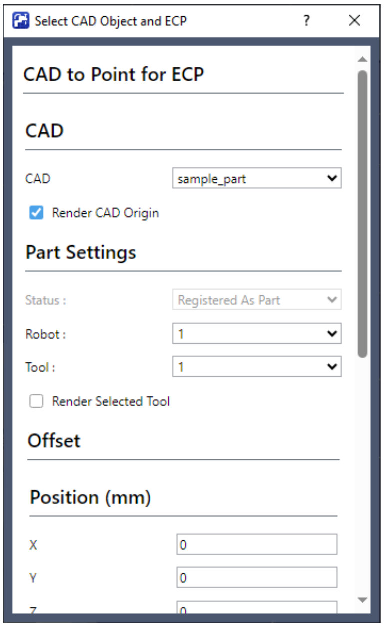
Click the [Simulator] button on the toolbar to display the Simulator window. CAD objects: "sample_part" and "syringe", and Hand objects are placed.



2. Open a project
   i. Click [Open...] from [Project] on Epson RC+ 8.0 menu.
   ii. Select [Projects]-[SimulatorDemos]-[CTP_for_ECP_C4_B].
   iii. Click the [Open] button.

3. Select CAD Object and ECP

i. Click the  [CAD to Point for ECP button on the toolbar to display the [Select CAD Object and ECP] dialog.

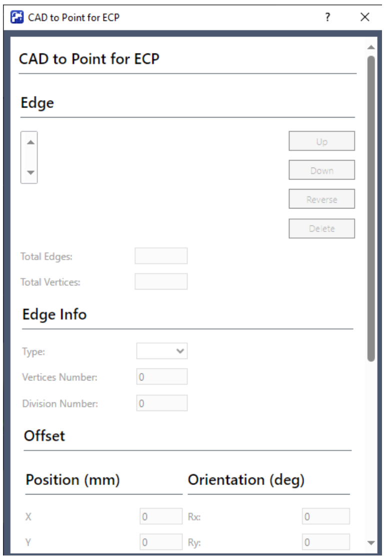


ii. Set as follows.

- CAD: sample_part
- Robot: 1
- Tool: 1
- Offset settings (X, Y, Z, Rx, Ry, Rz): 0.000
- ECP: 1

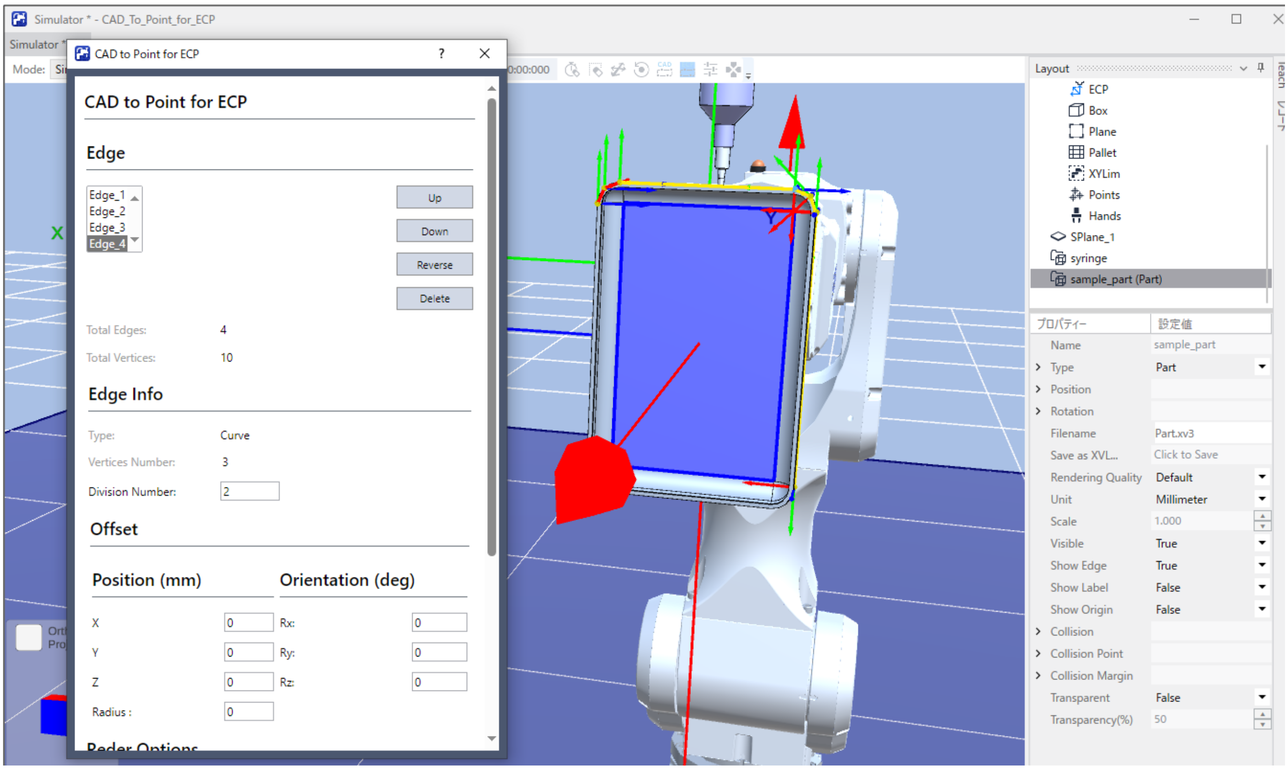
4. Select an edge of CAD object and create a motion path of the robot

i. Click the [Select Edge] button to display the [CAD to Point (ECP Support)] dialog.

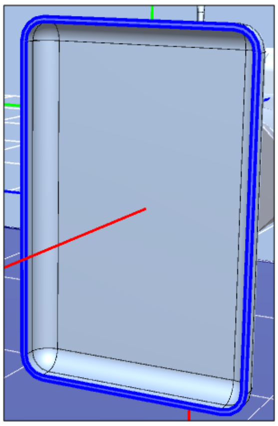


ii. To operate the sample program properly, select edges in counter clockwise rotation sequentially from edge of straight part of right tray. For selection of surface including edges and edges, see below.

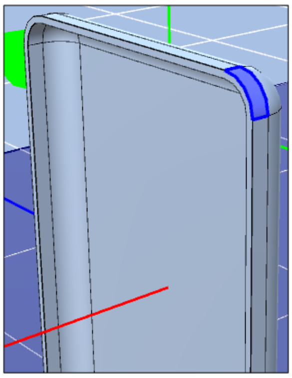
CAD To Point (6-axis Robot)



The straight part indicates edges on flat surface on the outer circumference.



The curve part indicates edges on side of the tray.



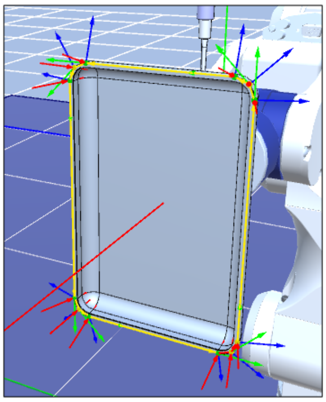
For division number and offset for each edge, refer to the following values.

Edge number			1	2	3	4	5	6	7	8
Type			Straight	Curve	Straight	Curve	Straight	Curve	Straight	Curve
Division number			0	2	0	2	0	2	0	2
Offset	Position (mm)	X	0	0	0	0	0	0	0	0
		Y	0	0	0	0	0	0	0	0
		Z	0	0	0	0	0	0	0	0
	Orientation (deg)	Rx	20	-70	20	-70	20	-70	20	-70
		Ry	0	0	0	0	0	0	0	0
		Rz	0	0	-90	0	180	90	90	180

TIP

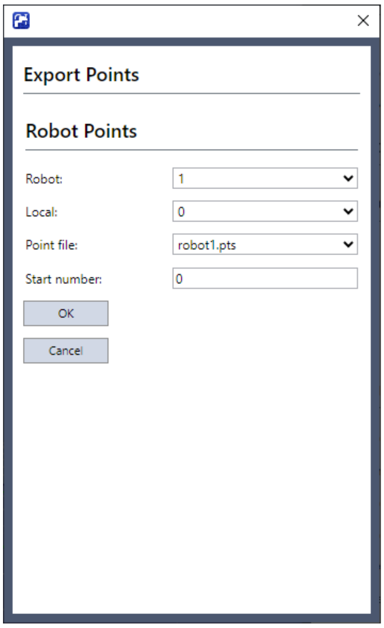
The arrow direction of edges indicates direction of start and end point of generating point. By clicking the [Reverse] button, you can reverse the arrow direction. Make sure to set the arrow direction will be counterclockwise rotation.

When all edges are set properly, it will be as follows.



5. Export the edges as point data

Click the [Export Points] button of the [CAD to Point (ECP Support)] dialog to display the [Export Points] dialog.



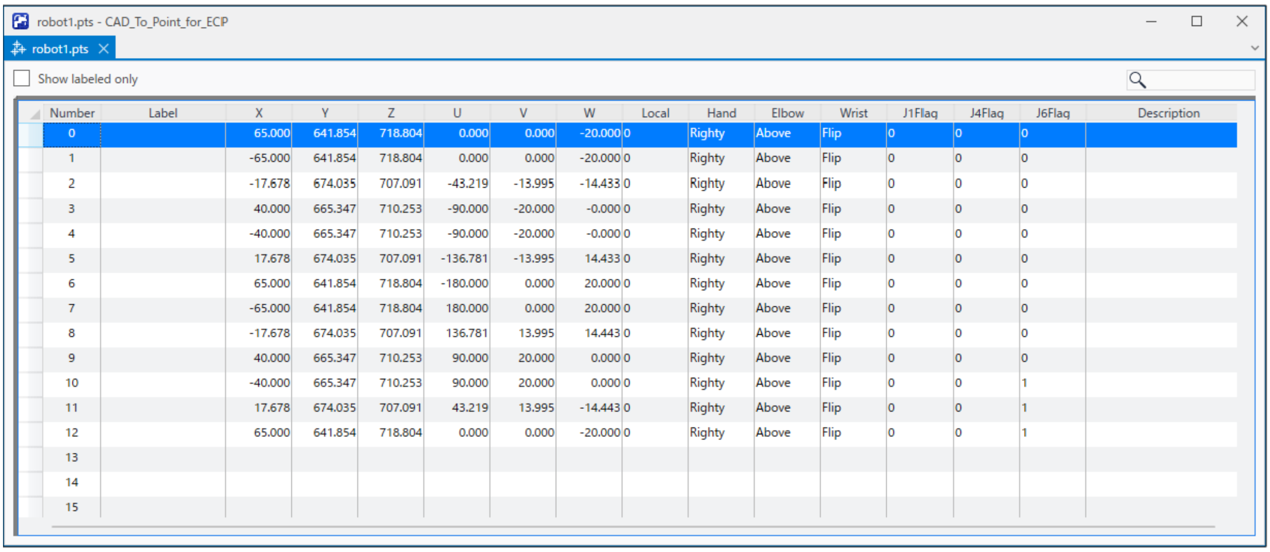
Click the [OK] button to output the point data to rows No. 0-12 in point file "robot1.pts".

6. Create a program

i. Set the proper robot orientation for the point data.

Open the point file "robot1.pts" from the layout objects and perform the following.

- Wrist orientation (Wrist) of output point No.0-12: NoFlip → Flip
- J6Flag of point No. 10-12: 0 → 1



Number	Label	X	Y	Z	U	V	W	Local	Hand	Elbow	Wrist	J1Flag	J4Flag	J6Flag	Description
0		65.000	641.854	718.804	0.000	0.000	-20.000	0	Righty	Above	Flip	0	0	0	
1		-65.000	641.854	718.804	0.000	0.000	-20.000	0	Righty	Above	Flip	0	0	0	
2		-17.678	674.035	707.091	-43.219	-13.995	-14.433	0	Righty	Above	Flip	0	0	0	
3		40.000	665.347	710.253	-90.000	-20.000	-0.000	0	Righty	Above	Flip	0	0	0	
4		-40.000	665.347	710.253	-90.000	-20.000	-0.000	0	Righty	Above	Flip	0	0	0	
5		17.678	674.035	707.091	-136.781	-13.995	14.433	0	Righty	Above	Flip	0	0	0	
6		65.000	641.854	718.804	-180.000	0.000	20.000	0	Righty	Above	Flip	0	0	0	
7		-65.000	641.854	718.804	180.000	0.000	20.000	0	Righty	Above	Flip	0	0	0	
8		-17.678	674.035	707.091	136.781	13.995	14.443	0	Righty	Above	Flip	0	0	0	
9		40.000	665.347	710.253	90.000	20.000	0.000	0	Righty	Above	Flip	0	0	0	
10		-40.000	665.347	710.253	90.000	20.000	0.000	0	Righty	Above	Flip	0	0	1	
11		17.678	674.035	707.091	43.219	13.995	-14.443	0	Righty	Above	Flip	0	0	1	
12		65.000	641.854	718.804	0.000	0.000	-20.000	0	Righty	Above	Flip	0	0	1	
13															
14															
15															

ii. Create the following program in Main.prg program.

```
Function main

  Motor On
  Power High

  Tool 1
  ECP 1
```

```
    Go P0

    Move P1 ECP CP
    Arc3 P2, P3 ECP CP

    Move P4 CP
    Arc3 P5, P6 ECP CP

    Move P7 CP
    Arc3 P8, P9 ECP CP

    Move P10 CP
    Arc3 P11, P12 ECP CP

    Pulse 0, 0, 0, 0, 0, 0
    Motor Off

Fend
```

iii. Click the [Build] button on the toolbar. The program will be built.

When the build is complete, the message "Build complete, no errors" appears in the status window.

7. Operate the robot by executing the program

   i. Click the [Run Window] button on the toolbar to display the Run window.
   ii. Click the [Start] button. When the message "Are you ready to start?" appears, click [Yes].
   iii. Confirm that the program is executed and follow an outer circumference of grasped CAD object (tray) by the robot on the edge of fixed syringe to operate ECP motion.

Functions of CAD to Point for ECP

Click the ECP [CAD to Point for ECP button on the toolbar to display the [CAD to Point (ECP Support)] dialog. For functions, see below.

CAD To Point (6-axis Robot) - "Function of CAD to Point"

9.3.9 CAD to Point for ECP (SCARA Robot)

This section describes CAD to Point for ECP for a SCARA robot.

In the example, create a motion to follow an outer circumference of grasped CAD object (tray) by the robot on the edge of fixed syringe.

Operate the following procedures:

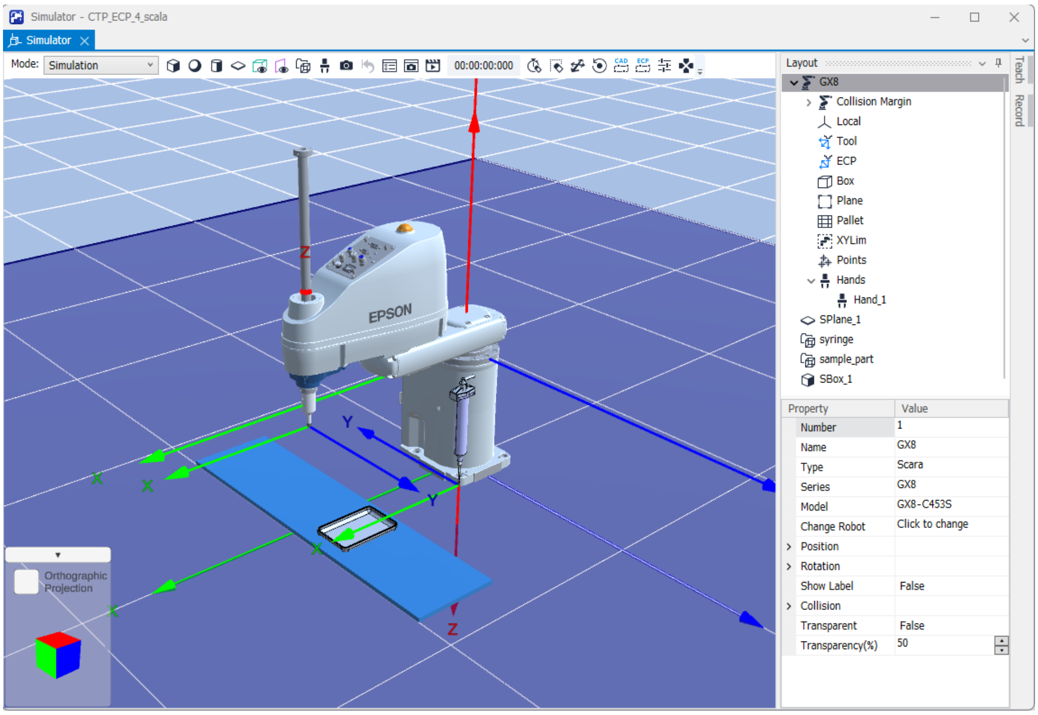
- Connect to virtual controller "CTPforECP GX8-C"
- Open a project
- Select CAD Object and ECP
- Select an edge of CAD object and create a motion path of the robot
- Output as point data
- Create a program

- Operate the robot by executing the program

1. Connect to virtual controller "CTPforECP GX8-C"

Select "CTP for ECP" of [Connection:] on Epson RC+ 8.0 toolbar. When the connection is complete, "CTP for ECP" is displayed in the [Connection:] box.

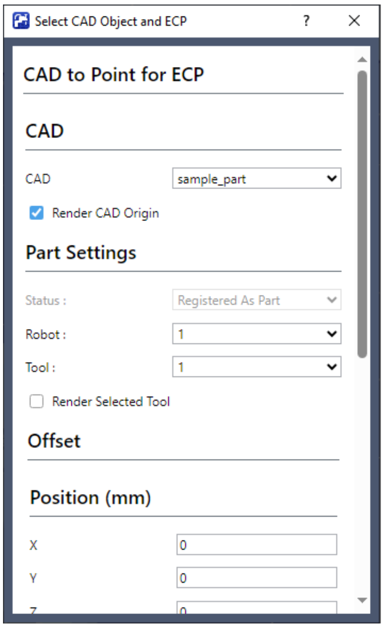
Click the  [Simulator] button on the toolbar to display the Simulator window. CAD objects: "sample_part" and "syringe", and Hand objects are placed.



2. Open a project
   i. Click [Open...] from [Project] on Epson RC+ 8.0 menu.
   ii. Select [Projects]-[SimulatorDemos]-[CTP_for_ECP_GX8_C].
   iii. Click the [Open] button.

3. Select CAD Object and ECP
   i. Click the 

[CAD to Point for ECP button on the toolbar to display the [Select CAD Object and ECP] dialog.

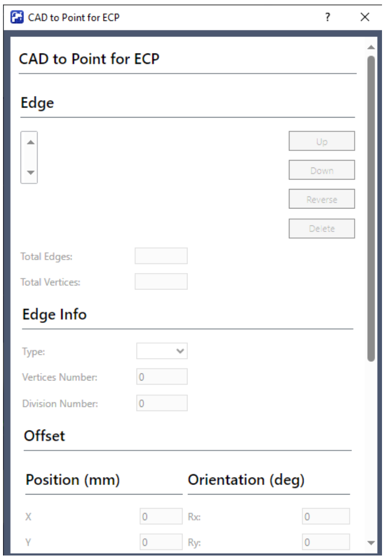


ii. Set as follows.

- CAD: sample_part
- Robot: 1
- Tool: 1
- Offset settings (X, Y, Z, Rx, Ry, Rz): 0, 0, -2, 0, 0, 0
- ECP: 1

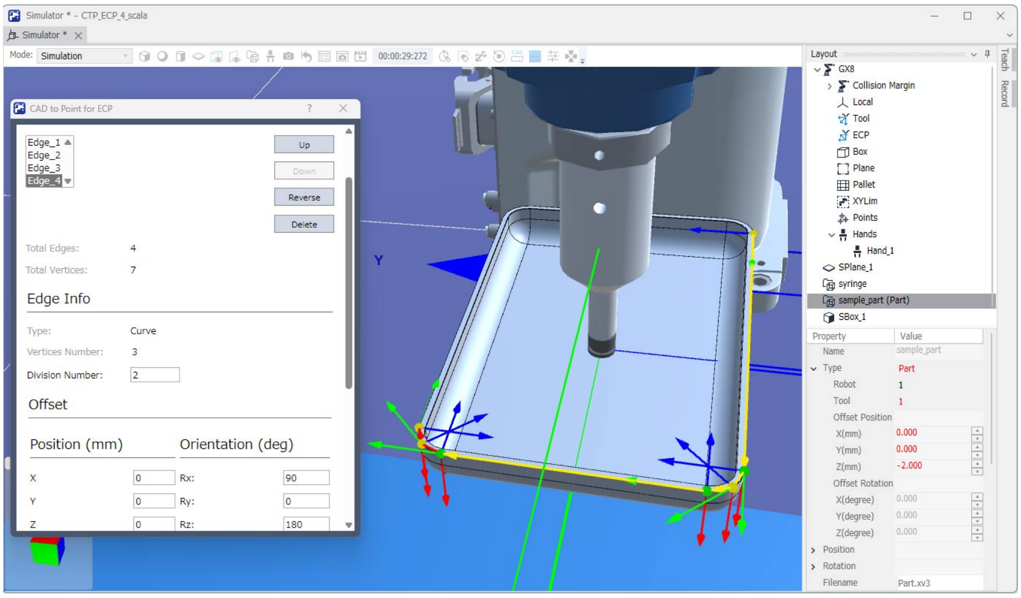
4. Select an edge of CAD object and create a motion path of the robot

i. Click the [Select Edge] button to display the [CAD to Point (ECP Support)] dialog.

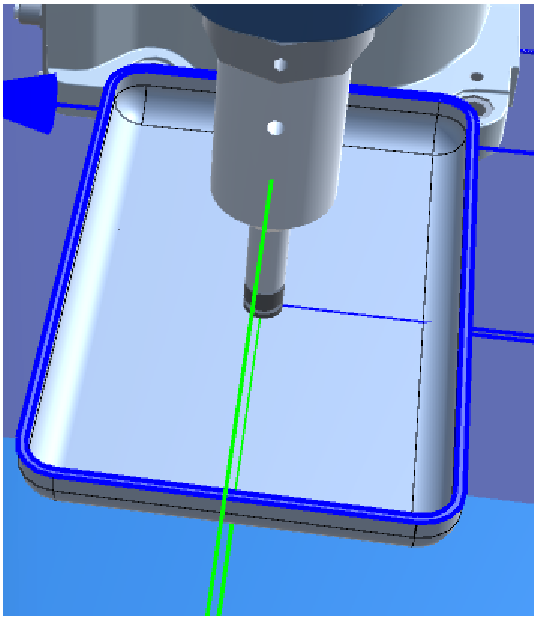


ii. To operate the sample program properly, select edges in clockwise rotation sequentially from edge of straight part of right tray. For selection of surface including edges and edges, see below.

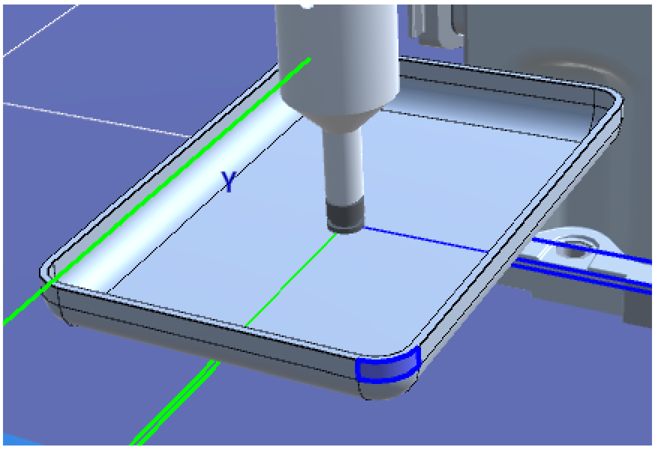
CAD To Point (6-axis Robot)



The straight part indicates edges on flat surface on the outer circumference.



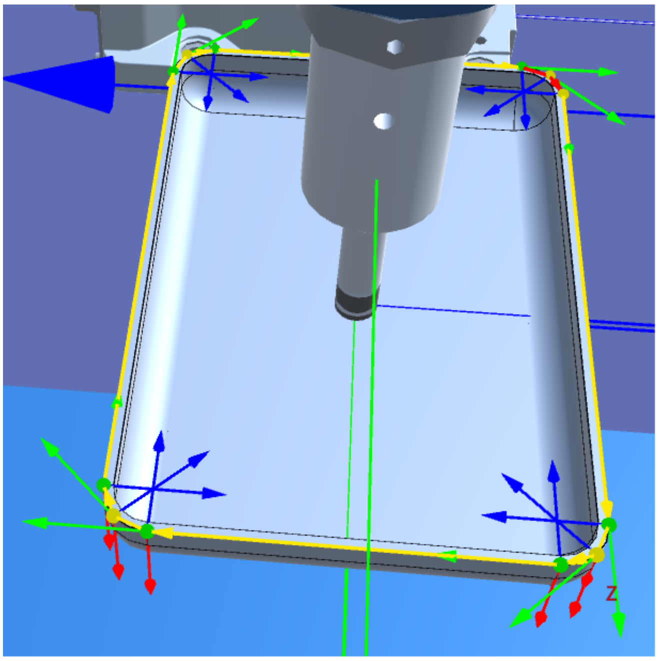
The curve part indicates edges on side of the tray.



For division number and offset for each edge, refer to the following values.

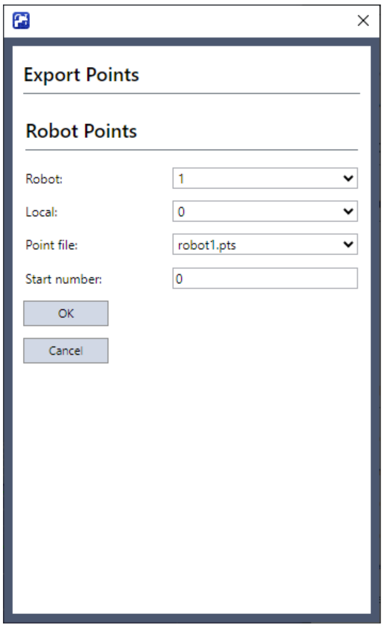
Edge number			1	2	3	4	5	6	7	8
Type			Straight	Curve	Straight	Curve	Straight	Curve	Straight	Curve
Division number			0	2	0	2	0	2	0	2
Offset	Position (mm)	X	0	0	0	0	0	0	0	0
		Y	0	0	0	0	0	0	0	0
		Z	0	0	0	0	0	0	0	0
	Orientation (deg)	Rx	0	90	0	90	0	90	0	90
		Ry	0	0	0	0	0	0	0	0
		Rz	0	180	90	180	180	0	-90	-90

When all edges are set properly, it will be as follows.



5. Export the edges as point data

Click the [Export Points] button of the [CAD to Point (ECP Support)] dialog to display the [Export Points] dialog.



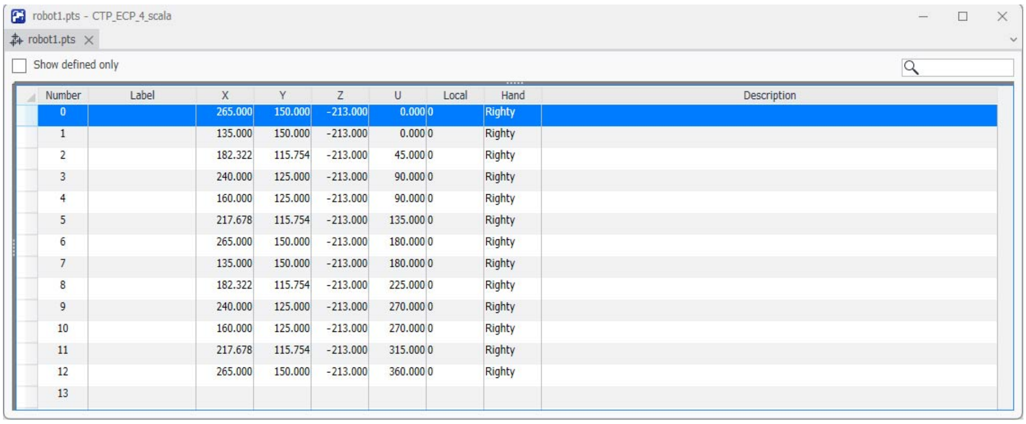
Click the [OK] button to output the point data to rows No. 0-12 in point file "robot1.pts".

6. Create a program

   i. Set the proper robot orientation for the point data.

   Open the point file "robot1.pts" from the layout objects and perform the following.

   - Output U value of point No. 6: -180.000 → 180.000
   - Output U value of point No. 8: -135.000 → 225.000
   - Output U value of point No. 9-10: -90.000 → 270.000
   - Output U value of point No. 11: -45.000 → 315.000
   - Output U value of point No. 12: 0.000 → 360.000



Number	Label	X	Y	Z	U	Local	Hand	Description
0		265.000	150.000	-213.000	0.000	0	Righty	
1		135.000	150.000	-213.000	0.000	0	Righty	
2		182.322	115.754	-213.000	45.000	0	Righty	
3		240.000	125.000	-213.000	90.000	0	Righty	
4		160.000	125.000	-213.000	90.000	0	Righty	
5		217.678	115.754	-213.000	135.000	0	Righty	
6		265.000	150.000	-213.000	180.000	0	Righty	
7		135.000	150.000	-213.000	180.000	0	Righty	
8		182.322	115.754	-213.000	225.000	0	Righty	
9		240.000	125.000	-213.000	270.000	0	Righty	
10		160.000	125.000	-213.000	270.000	0	Righty	
11		217.678	115.754	-213.000	315.000	0	Righty	
12		265.000	150.000	-213.000	360.000	0	Righty	
13								

   ii. Create the following program in Main.prg program.

```
Function main2

  Motor On
  Power High

  Tool 1
  ECP 1
```

```
    Go P0 +Z(-10)

    Go P0

    Move P1 ECP CP
    Arc3 P2, P3 ECP CP

    Move P4 CP
    Arc3 P5, P6 ECP CP

    Move P7 CP
    Arc3 P8, P9 ECP CP

    Move P10 CP
    Arc3 P11, P12 ECP CP

    Move P12 +Z(-50)

    Pulse 0, 0, 0, 0
    Motor Off

Fend
```

iii. Click the [Build] button on the toolbar. The program will be built.

When the build is complete, the message "Build complete, no errors" appears in the status window.

7. Operate the robot by executing the program

   i. Click the [Run Window] button on the toolbar to display the Run window.
   ii. Select a function "main2" created in step 6, and then click the [Start] button." When the message "Are you ready to start?" appears, click [Yes].
   iii. Confirm that the program is executed and follow an outer circumference of grasped CAD object (tray) by the robot on the edge of fixed syringe to operate ECP motion.

Functions of CAD to Point for ECP

Click the ECP [CAD to Point for ECP button on the toolbar to display the [CAD to Point (ECP Support)] dialog. For functions, see below.

CAD To Point (6-axis Robot) - "Function of CAD to Point"

Combination with Pick & Place operation

This sample project contains a program that combines the Pick & Place operation of "sample_part" and the ECP operation created up to this point as a "main" function. Confirm the operation by following the steps.

1. Click the [Run Window] button on the toolbar to display the Run window.

2. Select the function "SetDefaultSetting" and then click the [Start] button.

   When the message "Are you ready to start?" appears, click [Yes]. The robot and "sample_part" will be reset to their default positions and postures.

3. Select the function "main", and then click the [Start] button.

As in (2), When the message "Are you ready to start?" appears, click [Yes]. You will see the robot grasp the "sample_part" on the moving conveyor, trace its perimeter with ECP motion, and then reposition the "sample_part" on the conveyor.

The Pick & Place operation can be performed with the SimSet command. For details, refer to the following manual.

"SPEL+ Language Reference"

9.3.10 Virtual controller

To execute programs in the simulator, you need to create a virtual controller with defined robot and layout.

Robot settings and layout settings for the 3D display are saved for each virtual controller. If you want to transfer the robot or layout data, you can copy and transfer the data.

The virtual controller connection created by the Epson RC+ 8.0 cannot be used in lower versions of Epson RC+.

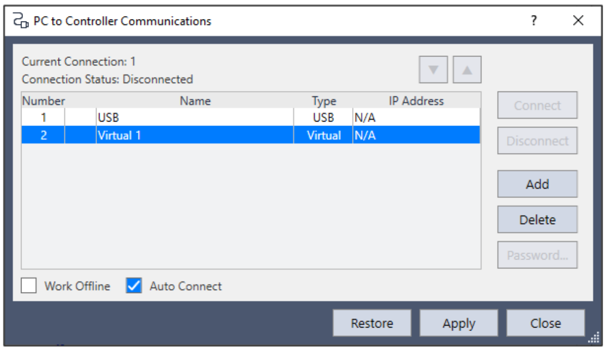
Create a new virtual controller

See details below.

Working with the user created system

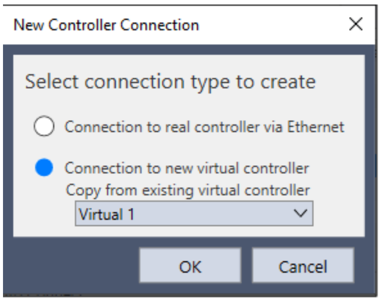
Copy the sample or configured virtual controller

1. Click the  [Connection] button on the Epson RC+ 8.0 toolbar. The [PC to Controller Communications] dialog box is displayed.

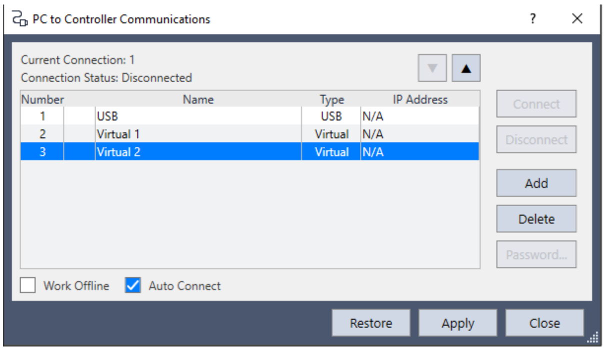


2. Click the [Add] button. The [New Controller Connection] dialog appears.

3. Select [Connection to new virtual controller] and specify a virtual controller from the list box. Click the [OK] button.



4. A new virtual controller named "Virtual 2" is created. Click the [Apply] button.



5. Click the [Close] button to return to the Epson RC+ 8.0 main window.

6. Connect to “Virtual 2” and display the [Robot Simulator] window. The robot setting and layout setting of 3D display has been taken over from “Virtual 1”.

7. When you want to change the robot type, use the [Change Robot] in the robot object property. See details below.

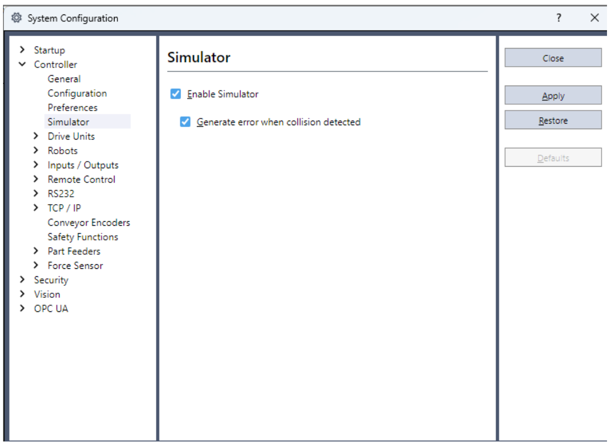
Property Grid Pane

9.3.11 Connection with controller

Enable the Simulator in controller

From the Epson RC+ 8.0 menu-[Setup]-[System Configuration]-[Controller]-[Simulator], check the [Enable Simulator] checkbox to enable the simulator function.

After checking the checkbox, click the [Apply] button and then click the [Close] button.



If collision with the simulator object is detected during a Jog motion or a robot motion command execution when the Simulator is enabled, the Manipulator stops operation and a Warning occurs.

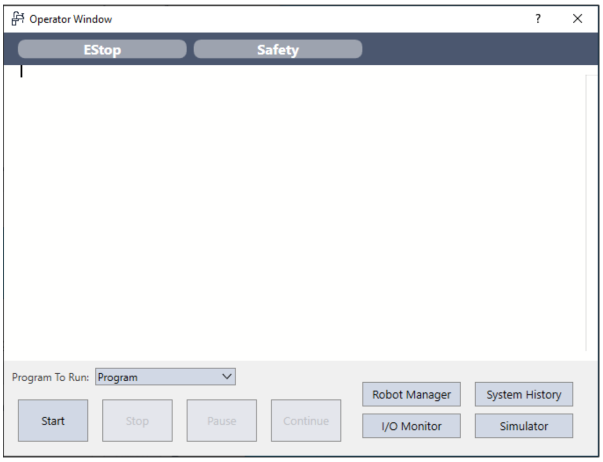
To avoid collision with peripherals by using the Simulator, set 15 mm or greater margins to the simulator object.

Function restrictions when connected with controller

- You cannot change the manipulator from the [Robot Simulator] window.
- You cannot select and move the manipulator arms in the [Robot Simulator] window, except during the controller Dry run.
- When a robot connected to the controller is not supported by the simulator, the robot will not appear in the layout tree and 3D window.
- The [Record/Playback] functions are not available.

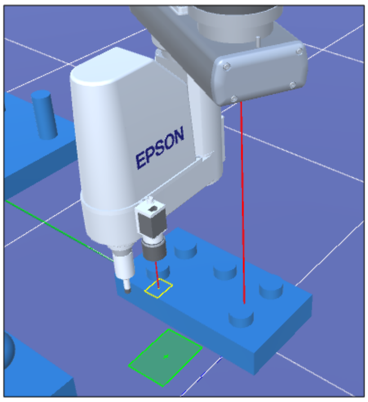
Operator Window

When you enable the simulator, the [Simulator] button is added to the Operator Window. When you click on the [Simulator] button, the 3D display window appears.



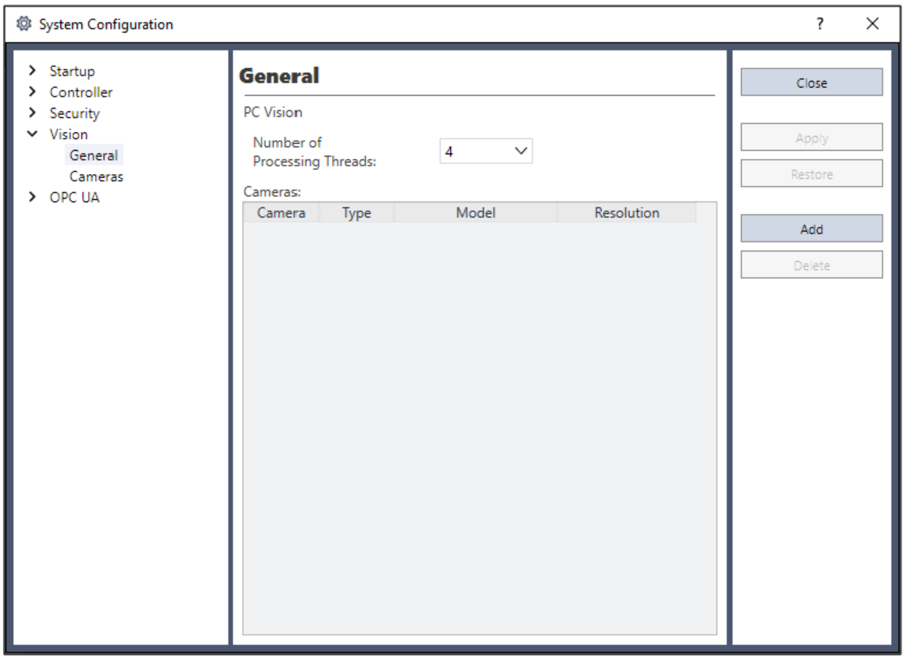
9.3.12 Virtual Camera Settings

Virtual camera settings are function to select camera or lens, and install as the fixed camera or mount as the mobile camera to the robot.

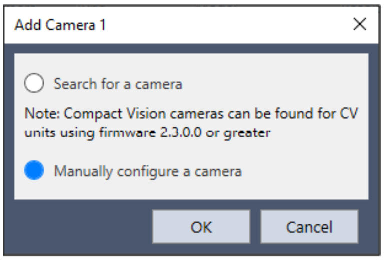


Add virtual cameras

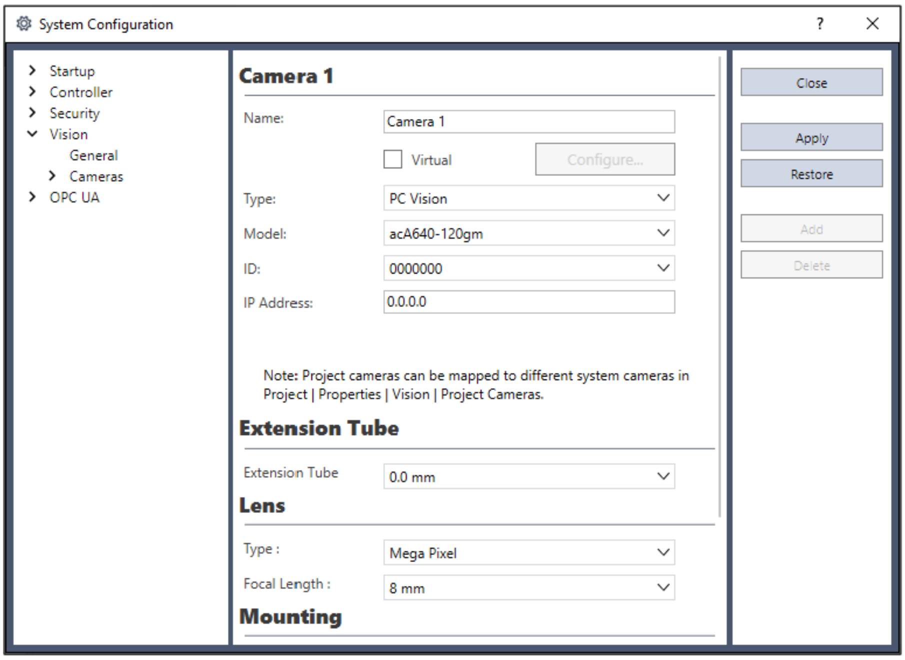
1. When you click the toolbar  [Camera] button, the camera settings window of the [System Configuration] dialog appears. Click the [Add] button.



2. Select [Manually Configure a Camera] and click the [OK] button.



3. The Configure Camera window is displayed. Set the camera type, mount type, etc.



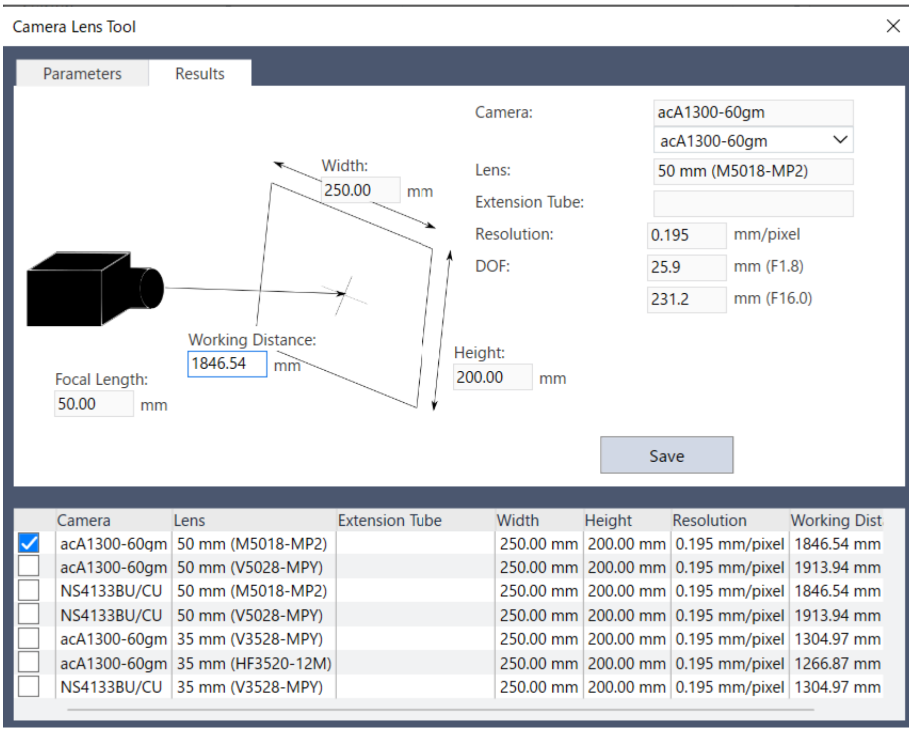
4. Check the [Virtual] or [Show simulator camera view in vision guide] checkbox and click the [Apply] button. The camera is added to the 3D view.

In addition to manually selecting extension tubes, lenses, and such as described above, another method for adding a virtual camera is to use the [Camera Lens Tool].

When parameters such as the width and height of the field of view are entered, the optimum combination of camera, lens, and extension tube can be automatically calculated and selected.

For details on Camera Lens Tool, refer to the following manual.

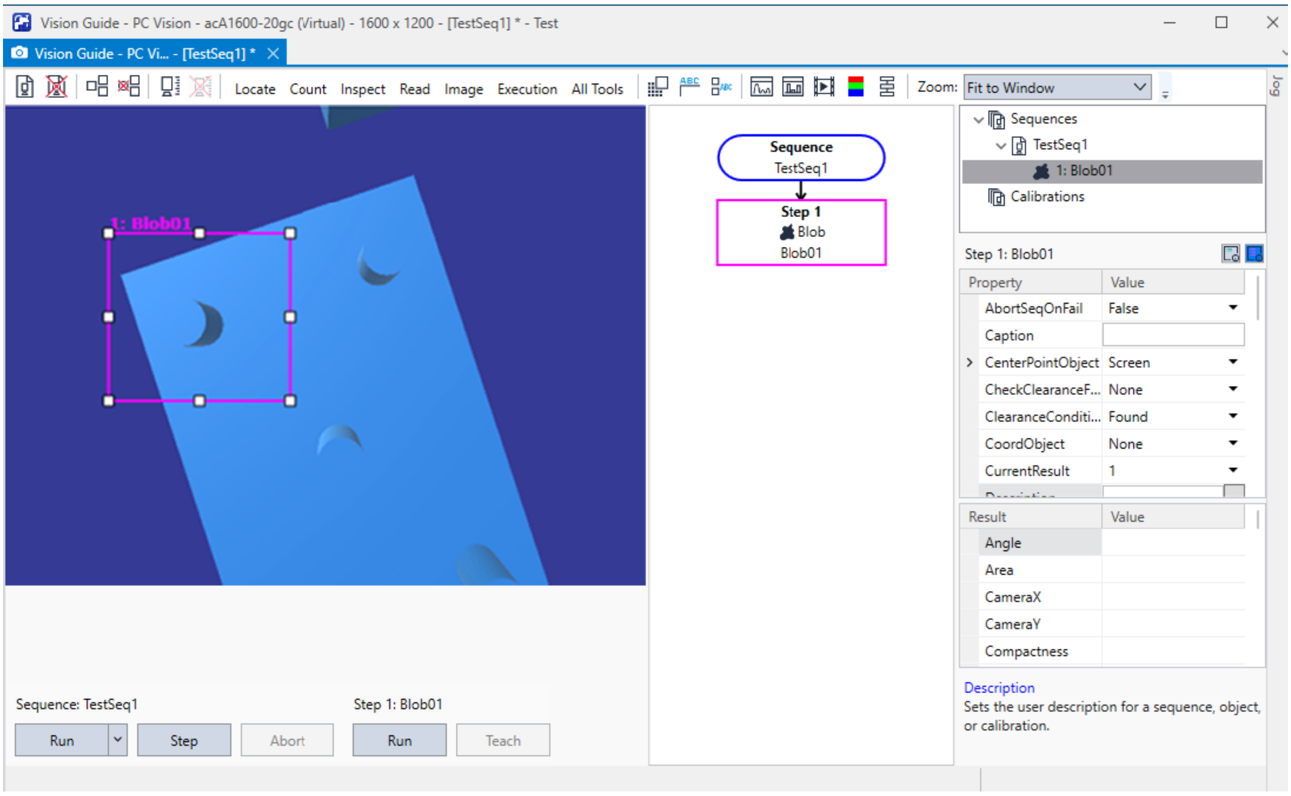
“Vision Guide 8.0 Ver.8.0 Hardware & Setup - Setup - Software Configuration - Camera Lens Tool”



Vision Guide synchronization

A virtual camera can be configured as a camera in Vision Guide.

The image on the 3D display, as seen from the virtual camera, allows for the use of Vision Guide's image processing capabilities. However, it is only available when PC Vision is selected as the camera type.



KEY POINTS

Depending on the combinations of the actual cameras and lens, mechanical vignetting occurs and the area around the image is captured black. The combinations that cause mechanical vignetting are shown below. Please be noted when using an actual camera.

Camera model	Lens type	Focal length
acA1300-60gm	Standard	12 mm
acA2500-20gm/gc	Standard	8mm, 12mm, 16mm, 50mm
	mega-pixel	8mm, 12mm, 16mm, 25mm, 50mm
	mega-pixel (HF)	8mm, 12mm, 16mm, 25mm
acA5472-5gm/gc	Standard	8mm, 12mm, 16mm, 50mm
	mega-pixel	8mm, 12mm, 16mm, 25mm, 50mm
	mega-pixel (HF)	8mm, 12mm, 16mm, 25mm

9.3.13 Motion restriction by BOX

By using the BOX command together with the GetRobotInsideBox function or the OnErr command, robot power and motion can be restricted when the tool center point (TCP) enters the approach check area (BOX).

Sample project using BOX

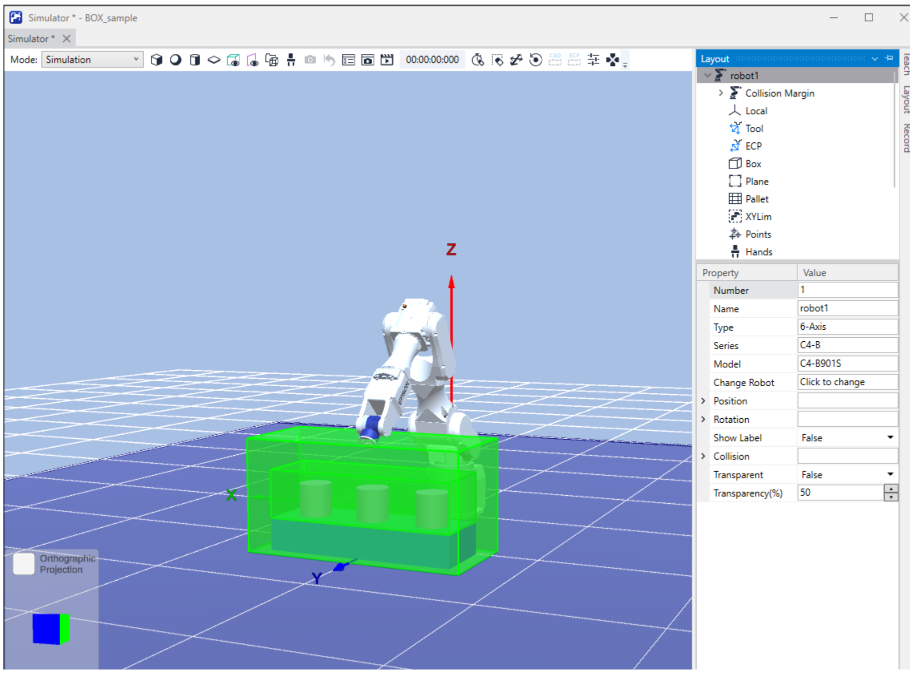
In the sample project, BOX2 is set outside BOX1. When the tool center point calculated based on the currently selected tool enters BOX2, the robot will stop temporarily. If the program execution is continued, the robot will resume operation in the restricted status (low speed, low power). Then, when the robot enters BOX1 inside BOX2, the robot will abort operation.

Use the sample project to execute the motion restriction using BOX. The projects is located in the following folders:

- For 6-axis robot: `\EpsonRC80\projects\SimulatorDemos\BOX_sample_C4_B`
- For SCARA robot: `\EpsonRC80\projects\SimulatorDemos\BOX_sample_CX8_C`

For details of the usage of the project, see below.

Working with the samples

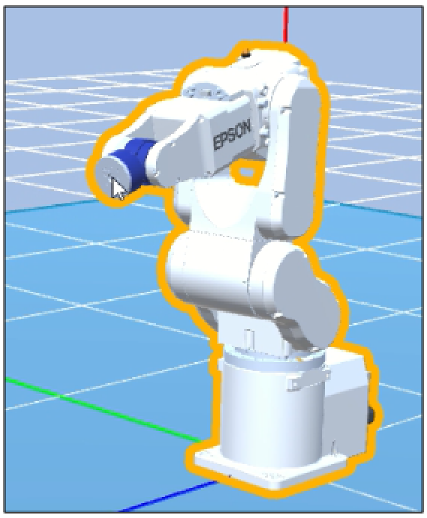


9.3.14 Virtual Direct Teaching

Virtual direct teaching is a function to perform a jog operation by dragging the robot. The operation simulates direct teaching.

How to use virtual direct teaching

1. Click the  [Direct Teach] button on the Epson RC+ 8.0 toolbar. When you mouse over the robot, the selected joint will appear in yellow.
2. Click the robot. The mouse cursor matches TCP.



3. Operate the robot by dragging.

   The robot operates so that the mouse cursor matches TCP. The robot operates with the arm orientation reflects the robot operation panel.

Virtual direct teaching restriction

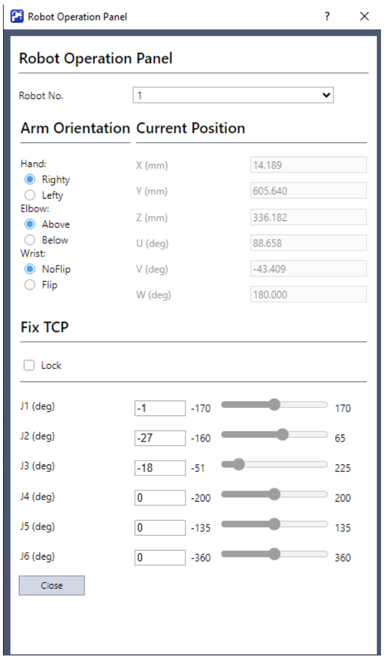
- When the mouse cursor is outside the robot operating range: Mouse cursor does not match TCP. The robot will be elbow singularity.

> **TIP**
>
> You can also rotate objects when the [Direct Teach] button is clicked. See details below.
>
> **3D Display** - "Rotate the robot / layout object"

9.3.15 Jog Operation on the Robot Operation Panel

Robot operation panel supports jog operation.

Click Epson RC+ 8.0 toolbar [Robot Operation Panel] button to display the [Robot Operation Panel] dialog.

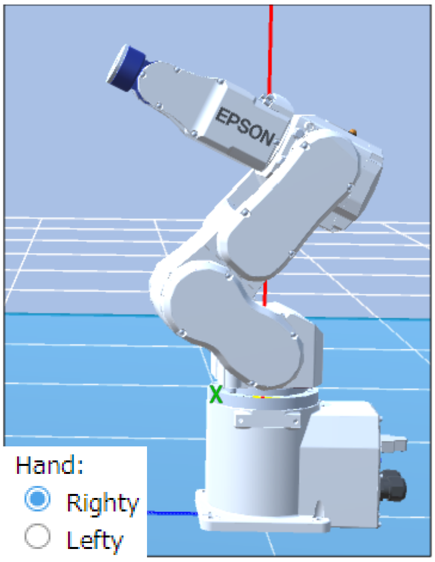


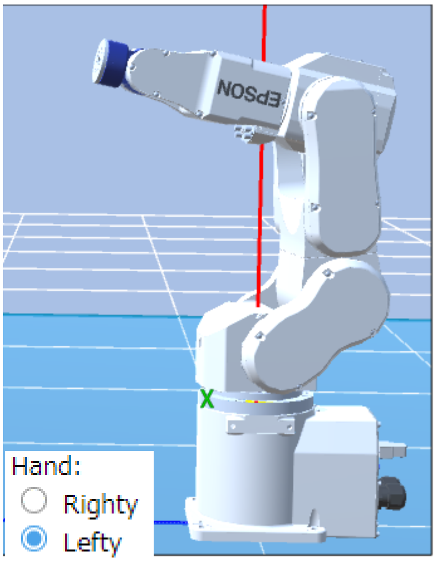
Item	Description
Robot No.	Displays the number of the robot to be operated. You can select the robot to be operated from the pull-down menu.
Arm Orientation	Displays the orientation flag of the robot to be operated. Select the button to change the orientation flag.
Current Position	Displays the coordinates (XYZ) and orientation (UVW) of the tool coordinate system selected in Robot Manager.
Fix TCP	Specify TCP to fix. You can use the TCP with the fixed orientation change.

Item	Description
Track bar for each axis	Displays the current value, maximum value, and minimum value of each axis. When you operate the track bar, the corresponding joint move.

Changing point flag with radio button

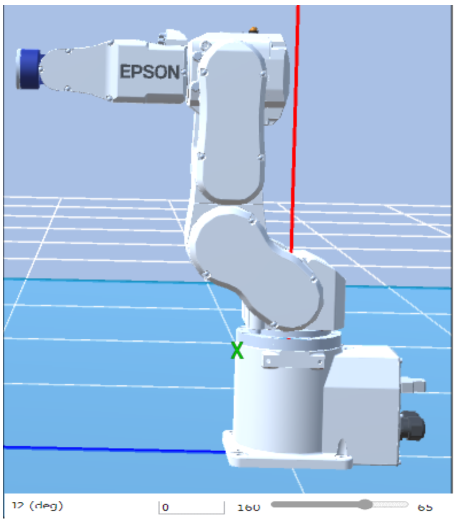
By selecting the radio button, you can change the point flag of the robot to be operated.

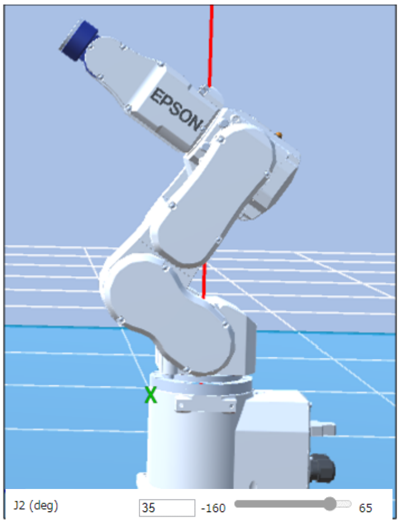




Joint operation by track bar

Drag the movable part of the track bar to move the corresponding joint.





You can also operate the track bar in the following two ways:

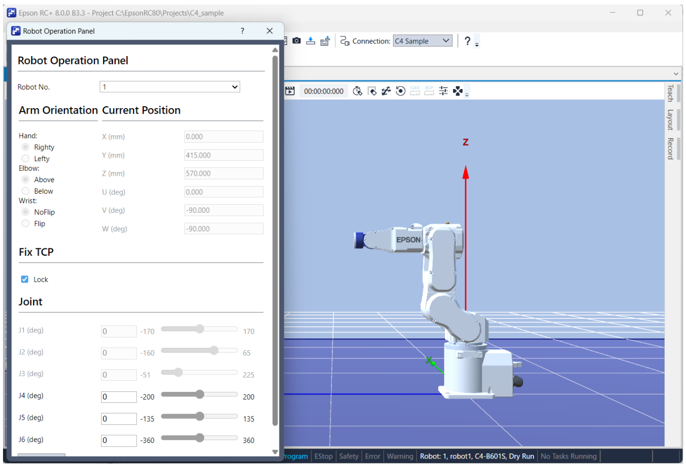
- Click the track bar.
  - Click the right side of the movable part of the track bar to change the value by +10.
  - Click the left side of the movable part of the track bar to change the value by -10.
- Press the arrow keys.
  - After selecting the track bar, press the [→] key to change the value by +1.
  - After selecting the track bar, press the [←] key to change the value by -1.

Changing orientation with fixed TCP

You can change the orientation of the robot with TCP fixed. However, this function fixes the origin position of the tool coordinate system 0. Follow the steps below.

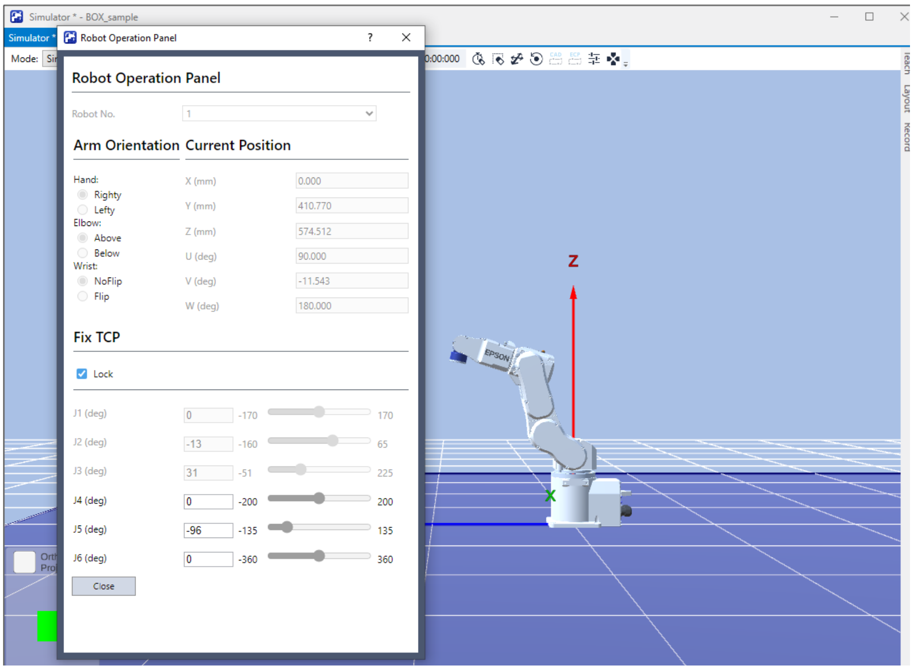
1. Check the [Robot Operation Panel] – [Lock] check box.

   The TCP to be fixed is specified. The track bars and point flags from J1 to J3 are disabled.



2. Move the enabled track bar on [Robot Operation Panel].

   The axis operated by the track bar moves with TCP and point flags held.



3. Uncheck the [Robot Operation Panel] – [Lock] check box.

   The [Lock] display becomes default and the function to fix TCP is completed. The track bars for J1 to J3 are enabled.

In the following three cases, TCP cannot be used for changing the fixed orientation. In these cases, the track bars for J1 to J3 are disabled.

- The robot is singularity.
- TCP is near the Z axis.
- The current value of whichever J1 to J3 is the maximum or minimum value.

9.4 Simulator Specifications and Restrictions

This section describes the simulator specification, its restrictions, and precautionary statements.

9.4.1 Epson RC+ 8.0 package

Epson RC+ 8.0 has two packages:

- Epson RC+ 8.0: Standard package for developing the robot system
- Epson RC+ 8.0 Trial: Trial package for limited use (program execution on PC) * It cannot connect with a robot controller.

	Simulation, Program execution on PC	Connection with controller	Connection with controller + 3D display
Epson RC+ 8.0	OK *2	OK	OK *1
Epson RC+ 8.0 Trial	OK *2	×	×

*1 Requires the configuration to enable the simulator functions in Epson RC+ 8.0. See below.

Connection with controller

*2 Total execution time of program is limited.

9.4.2 Specifications and precautions for the 3D display

Available robots for 3D display

In the future, we will add more robots for 3D display. Please contact the supplier of your region for the latest information.

Some robots cannot use this function. For details on unsupported models, see below.

Appendix C: Simulator Functions List of Unsupported Manipulator Models

KEY POINTS

- The flexible duct is displayed roughly.
  - Check the manipulator manual for the dimensions.
  - The duct actually vibrates while the manipulator is moving, the simulator doesn't display the vibration. Check how the duct vibrates with your real manipulator.
- The bellow for Cleanroom or Protected-model is displayed roughly
  - Check the manipulator manual for the dimensions.

Available CAD data for 3D** display**

The following format is available for 3D display to show the robot hand and CAD object.

- VRML 2.0 *1
- STEP (AP203/AP214) *2
- IGES
- DXF
- AutoCAD R software DXF Format (DXF R13, DXF R14, DXF 2000/2000i, DXF 2002)

*1 Limits of reading: VRML2.0 prototype is not supported.

*2 Limits of reading: The file whose character code is only ASCII can be read. In addition, if Color is configured in Face, the specified Color is displayed.

> **KEY POINTS**
>
> The data file must be saved in the specified folder on the PC. It is not saved in the Epson RC+.

The character code of file path for CAD data

For CAD data files in VRML 2.0 format or IGES format, the data cannot be loaded if character code is included that differs from the language in the operating environment used in the file path (file name, folder name). Change the file and folder name to be the same character code as the language in operating environment.

CAD data loading restrictions

The total number of polygons and polylines is limited to a million each. When the error message appears, reduce the number of polygons and polylines.

CAD data setup orientation

Some CAD data coordinates may be different from those of the simulator.

Adjust the coordinates to the correct position by changing [Property]-[Rotation] after loading the CAD data.

When loading CAD data as a hand, set the origin of the CAD data in the Tool0 position of the manipulator. Set the coordinates to the correct position by changing [Property]-[Position] after loading the CAD data.

Number of available layout objects

You can create as many layout objects as you want.

However, when there are many objects to display, the display update interval becomes longer and the judgment of collision detection becomes rough.

Especially for CAD data, displaying data that is too complicated is not recommended.

Shape of CAD object

The shape of objects may be displayed incorrectly (such as a clearance arises between the faces) depending on the CAD data. In such a case, the shape may be improved by converting the data to a different format.

Front-back relation of objects in semi-transparent display

The front-back relation of the objects may be incorrect when displaying the CAD and Hand objects in semi-transparent display.

Rendering speed

It may take a few seconds to render the objects depending on the display adapter, and operability such as in selecting the object may be decreased. It is recommended to update the driver to the latest version.

9.4.3 Specifications and precautions for Simulation (program execution on PC)

Overview

The Simulator produces the robot motions virtually on your PC.

It is designed to make the performance gap between the real system and the virtual system as small as possible. However, a few differences in the virtual system are inevitable. The operation time prediction and collision detection do not guarantee the precision.

Fully understand the contents in this chapter and check if the real system operates without any problems before you go to full-scale operation.

Operation time prediction

Operation time displayed in the [Robot Simulator] window is approximate time required for executing the program.

Time for the motion commands such as Go, Jump reflects the Speed and Accel values in the program. The operation time may vary when you operate the real robots from the displayed operation time according to conditions such as the Fine setting and servo delay. In particular, when small ranges are used with the Fine instruction, the real robots need a longer operation time for accurate positioning.

The simulation cannot guarantee the precision but the margin of error in the operation time is within 10% when you execute motions with the standard cycle time (with the default Fine settings).

- Considered in the operation time prediction
    - Robot model
    - Speed settings (Speed, Speeds, etc.)
    - Acceleration settings (Accel, Accels, etc.)
    - Load (Weight, Inertia)
    - Others (ARCH, CP)
- Not considered in the operation time prediction
    - Fine setting
        - Error within 10% from the default (Motions of standard cycle time)
        - With larger setting than the default, the operation time will be shorter.
        - With smaller setting than the default, the operation time will be longer.
    - Servo delay
        - With the real robots, the operation will be longer.

Time for the other commands than the motion commands is a virtually executed time on PC; therefore the actual time varies widely depending on the PC performance.

When measuring the motion time between two points, as simple program as possible is recommended. (See: "Measure the robot operation time" - **Working with the user created system** -)

Collision Detection Precision

The Simulator Collision Detection provides an indication whether robots collide with the peripheral equipment or not when the program is executed. It does not consider the error in path due to servo delay. Make sure to have a margin when you apply the simulation result for a real robot system.

The Simulator judges collisions more accurately when the robot motion speed is slow.

The judgment of collision detection during program execution is accomplished with the 3D display update. When your PC has high graphics performance, the collision judgment becomes more accurate.

In Playback mode, judgments on collision detections are made in all steps. This allows for more accurate detection.

The Simulator cannot guarantee the precision but the margin of error in the collision detection is within 10 mm when you execute motions with Speed 100% on a PC of the recommended specifications.

Motion duty and Overload error

In the Simulator, you cannot detect overload errors. Even if the motion duty is excessively high and the robot should stop due overload error, it keeps operating.

Duty 50% - As a measure of possible duty, the robot can really keep moving at duty 50% with the maximum acceleration/deceleration speed and without the overload error. However, it depends on the robot model type, load, points to go to, and acceleration/deceleration speed setting, etc.

Execution on PC that does not meet recommended system requirements

You can install the Epson RC+ and use the Simulator functions on a PC that does not meet the recommended system requirements. However, it doesn't guarantee the correct motions because the following may happen:

- Operation time prediction is not accurate
- Collision Detection has a large margin of error
- 3D display skips updates

9.4.4 Specifications and Precautions of Controller Setting on Epson RC+

Restriction on the controller settings

When a virtual controller is connected, the following settings cannot be made in [Setup]-[System Settings]-[Controller]. The items that cannot be set are displayed in gray.

- IP Address settings, etc. on [Configuration]
- Dry Run settings, etc. on [Preferences]

Backup and restore of the controller setting

The setting data that you backup in the virtual controller can be restored in a controller. Also, the setting data that you backup in a controller can be restored in a virtual controller.

However, there are restrictions. See details below.

[Controller] Command (Tools Menu) - "Backup Controller", "Restore Controller"

9.4.5 Restriction on SPEL+ command execution

I/O operation and commands (On, Off, SW, Ctr, etc.)

All I/Os, including optional boards, are available. Operating I/O data is stored in the PC memory (virtual I/O mode). The I/O input status can be changed from the Epson RC+ I/O Monitor window. Also, the I/O input status can be changed using the SetSw or SetIn statements in a SPEL+ program.

> KEY POINTS
>
> Even if you specify an asynchronous On/Off command, the I/O status cannot be changed after the specified time.
> The Ctr function always returns 0.

Ethernet/RS-232C communication command (Print #, Input # , OpenCom, OpenNet, etc.)

All 16 Ethernet ports are available. However, an Ethernet port requires configuration of the IP address and TCP/IP port.

For RS-232C on the Controller, all 8 ports, including the optional RS-232C board, are available.

Note:

For RC800, RC700 and RC90 series Controller, up to 5 ports, including the standard port and the optional RC-232C port, are available. T series and VT series Manipulator do not have RS-232C ports on the Controller. Be careful of the number of the ports when using the project created in the virtual controller to the Controller.

As a default, Ethernet / RS-232C communication commands do not perform actual communication.

To use the actual Ethernet/RC-232C ports, be sure to configure as described in (3).

Output data from Print # , etc. is saved in the communication output file. In the input by Input #, etc, the return value is 0 (numeric data) or blank (string). However, if you create a communication response file, the return value depends on the file content.

Communication output file

When an OpenCom or OpenNet command is called, a communication output file is created. For the output destination folder of the communication output file, see below.

- **RS-232C Communications**
- **TCP/IP Communications**

DummySend***.dat: Communication output file (*** is the port number)

When a communication output file already exists, the previous output data is deleted. The file is deleted when you switch the project; save the file in a proper folder if you need.

When executing the following program,

```
OpenCom #1
Print #1, 123
Print #1, "TEST DATA"
CloseCom #1
```

The DummySend001.dat file will contain the following:

```
123
TEST DATA
```

Communication response file

Copy the communication response file to the designated folder in advance. For the communication response file folder, see below.

- **RS-232C Communications**
- **TCP/IP Communications**

The file is deleted when you change the project; save the file in another folder if you need to keep it.

When calling OpenCom or OpenNet command, the communication response file is loaded.

DummyRead***.dat: Communication response file (*** is a port number)

When the following DummyRead001.dat file is used,

```
321
Test Data
```

When executing the following program,

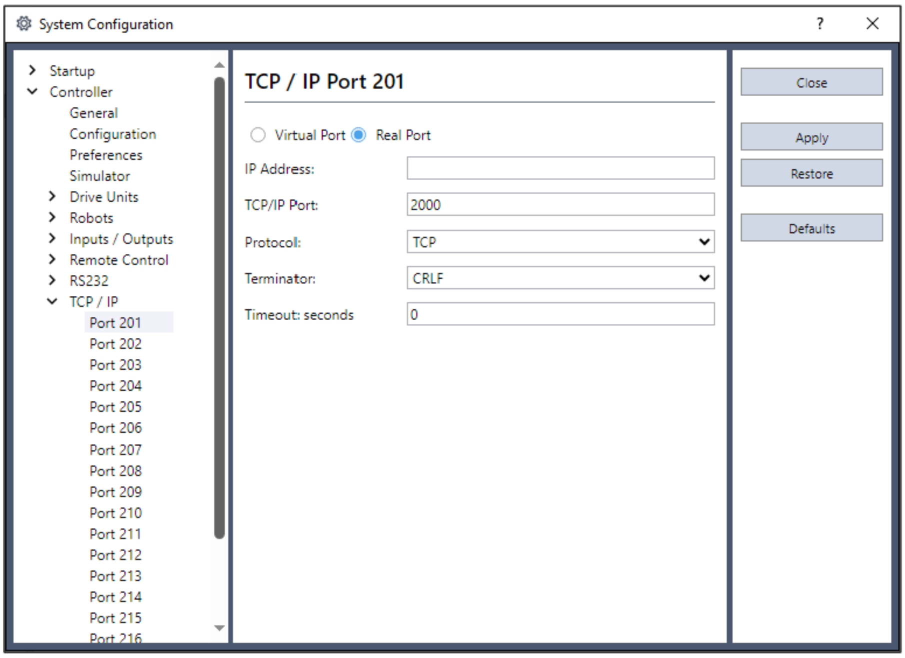
```
Integer i
String s$
OpenCom #1
Input # 1, i
Input # 1, s$
CloseCom #1
Print i
Print s$
```

the return values are i = 321 (numeric data), and s$ = "Test Data" (string).

How to enable the actual ports of Ethernet/RS-232C in the virtual controller

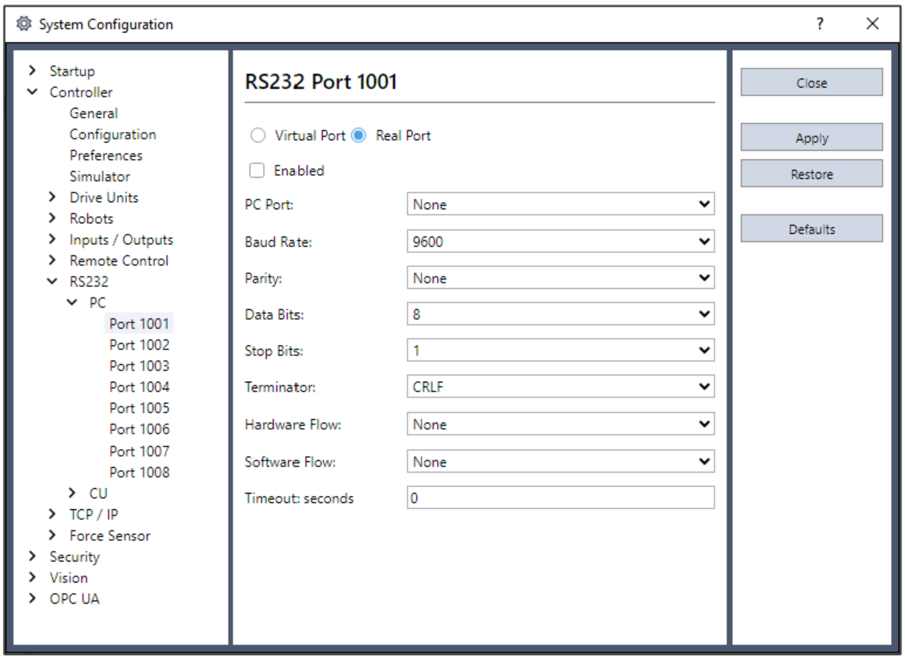
Actual ports become available when [Real] is selected in [Setup]-[System Configuration]-[Controller]-[TCP/IP] from the Epson RC+ 8.0 menu.

Change the port settings, then click the [Apply] button and the [Close]button.



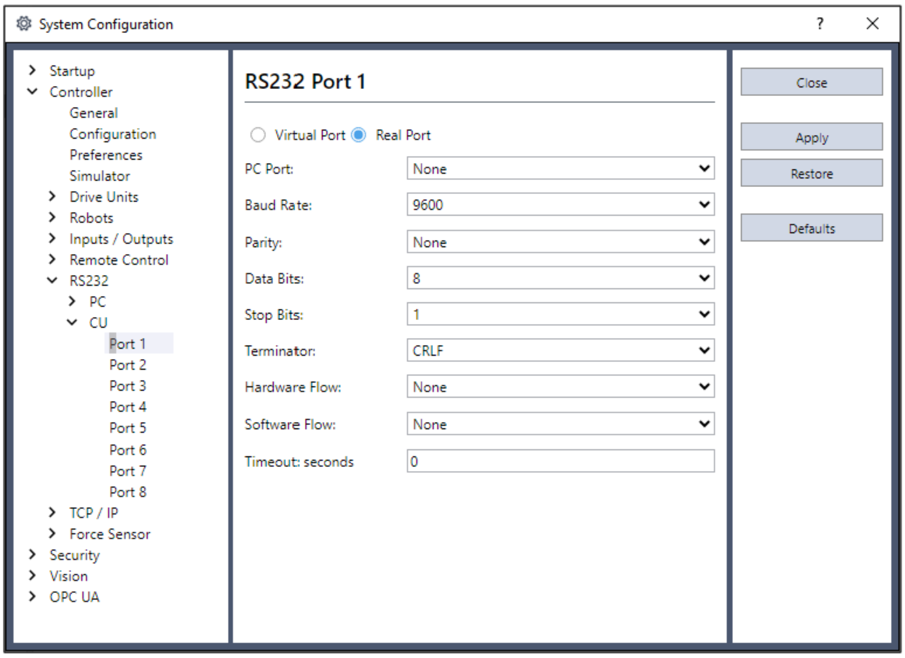
Actual ports become available when [Real] is selected in [Setup]-[System Configuration]-[Controller]-[RS-232]-[PC] from the Epson RC+ 8.0 menu.

Select the PC port, then click the [Apply] button and the [Close] button.



Actual ports become available when [Real] is selected in [Setup]-[System Configuration]-[Controller]-[RS-232]-[Controller].

Select the PC port, then click the [Apply] button and the [Close] button.



To use the actual Ethernet/RC-232C ports, select the [Actual] button in the configuration dialog.

Vision command (VRun, VGet, etc.)

Vision sequence can be executed with an image file set in ImageFile property as an input image. Also the result can be acquired by VGet. When the PC vision is set and the GigE camera is connected, vision commands such as VRun and VGet the can be executed using actual camera image. In this case, commands can be executed from the virtual camera function like the Compact Vision, when the GigE camera is not connected.

For details on Vision Guide, refer to the following manual.

"Epson RC+ Option Vision Guide 8.0"

Other restrictions

- For the Wait command, the following syntax is not supported:
  Wait InsideBox()
  Wait InsidePlane()
- For the Time and Date commands, the time can be displayed, but the time setting is not available.
- For the SimSet commands, the motion of the parts that specify Pick or Place and movement or rotation of the objects that specify PositonX, PositionY, PositionZ, RotaitonX, RotationY, or RotationZ cannot be recorded and produced by the [Record/Playback] functions.

Program total execution time

In the virtual controller, the total execution time of programs is limited to one hour.

If total execution is over one hour, a warning message appears.

You can execute the program again after the warning is displayed. The total execution timer will be reset.

9.4.6 Specifications and precautions of Epson RC+ 8.0 Trial

Version upgrade from Epson RC+ 8.0 Trial to Epson RC+ 8.0

Follow the procedures below to install Epson RC+ 8.0 to the PC. There is no need to uninstall Epson RC+ 8.0 Trial from your PC.

Appendix B: Epson RC+ 8.0 Software

> **KEY POINTS**
>
> You can continue to use the projects and virtual controllers (layout) that you used in the Epson RC+ 8.0 Trial in the Epson RC+ 8.0 Standard version.

10. Motion System

Epson RC+ supports the motion systems listed below.

- Standard Motion System
- PG Motion System

10.1 Standard Motion System

The standard motion system consists of the Control Units and the Drive Units (optional, up to three units).

You can connect one robot to the Control Unit directly. For details on the Robot Controller and maintenance, refer to the following manual:

"Robot Controller Manual"

The Drive Units are automatically recognized at the startup of the Control Unit if they are connected to the system.

When addition and removal of the Drive Units are automatically recognized, the startup time becomes longer in order to reboot the Control Unit.

10.2 Drive Module Software Configuration

The drive module is configured at the factory before shipment. It is automatically recognized by the controller. You do not have to configure the settings. Also, you do not have to configure the settings for the drive module in the Drive Unit which is automatically recognized.

10.3 PG Motion System

The PG (Pulse Generator) Motion System is an option.

When a PG board is installed in the controller, it is automatically recognized. You can select it in the robot configuration dialog.

For details on how to use, refer to the following manual:

"Robot Controller Option PG Motion System"

11. Robot Configuration

This chapter contains information for adding robots and configuring additional axes.

- Robot configuration: Adding a standard robot
- Additional axes configuration: Adding a robot with additional axes

11.1 Robot Configuration

⚠ CAUTION

Each robot is configured before shipment. Therefore, it is normally unnecessary to change the settings. If you change the settings, it may cause the robot to malfunction or perform unusual motion. This is extremely hazardous and you should be careful.

11.1.1 Adding a standard robot

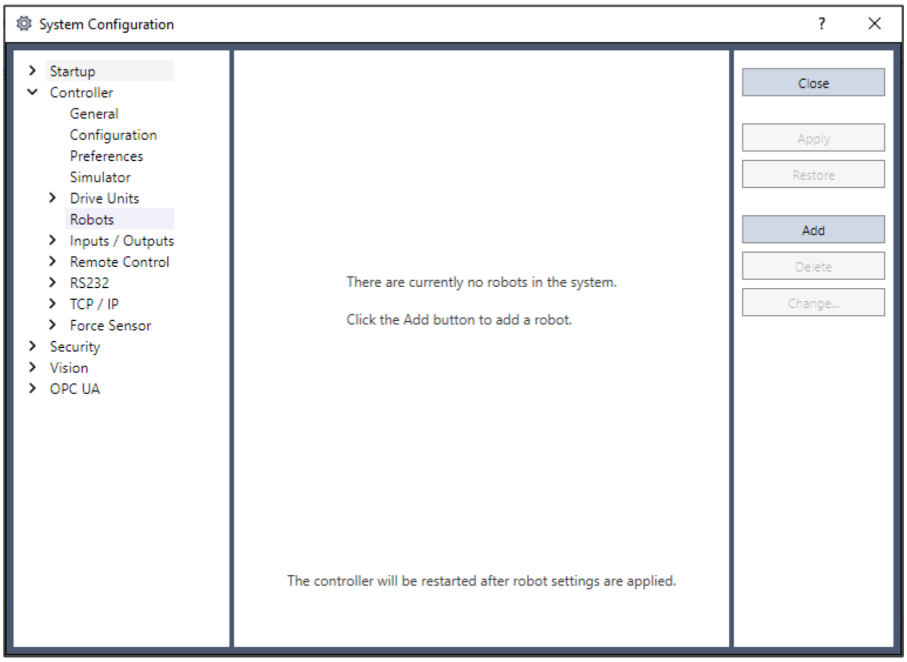
KEY POINTS

If you have purchased the PG motion system (option), you can add up to three user-defined robots. For details, refer to the following manual.

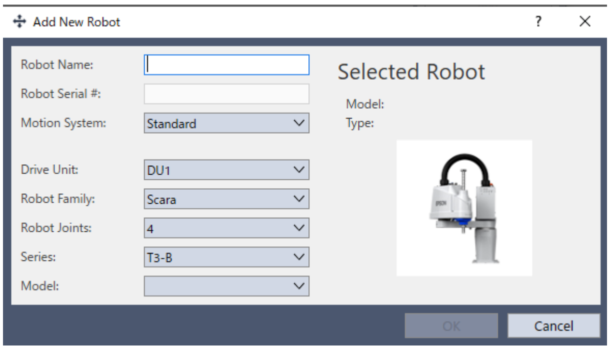
"Robot Controller Option PG Motion System"

1. In the Epson RC+ 8.0 menu, select [Setup]-[System Configuration].

2. Select [Controller]-[Robots].



3. Click the [Add] button. The dialog shown below appears.



4. Type in a name for the new manipulator and enter the serial number on the manipulator's nameplate. Any serial number can be used, but it is recommended that you use the number that is stamped on the manipulator.

5. Select a motion system to use from the [Motion System] dropdown list. If there are no other motion systems installed, then "Standard" will already be selected.

6. Select a Drive Unit for your manipulator from the [Drive Unit] dropdown list.

7. Select the robot type in the [Robot Family] box.

8. Select the series name of the manipulator in the [Series] box.

9. Select the robot model in the [Model] box. Available robots will be displayed according to the format of the currently installed motor driver. If you use [Dry run], all robots selected in step 8 will be shown.

10. Click the [OK] button. The controller will be restarted.

KEY POINTS

For the Controller with Safety Board, assign a robot using the safety function to Robot 1. A Safety Board password is necessary when using Controller with Safety Board.

11.1.2 Calibrating a standard robot

The calibration method differs according to the manipulator model.

For details, refer to the following manual.

"Manipulator Manual - Calibration"

11.1.3 Changing robot system parameters

The following robot parameters can be changed from the [System Configuration] dialog.

- Enable/Disable Joints

  You can disable one or more joints from [Setup]-[System Configuration]-[Robots]-[Robot**]-[Configuration]. On robots with a ball screw Z axis, you must disable both joints 3 and 4 together.

- Hofs

  Hofs are the joint home offsets. You can view and edit the values from [System Configuration]-[Robots]-[Robot**]-[Calibration]. However, it is recommended that you use the Robot Calibration wizard to set these values. These values are unique for each robot and are supplied from the factory. Hofs are especially important for SCARA robots because the values determine that both lefty and righty hand orientation will position the robot at the same point.

- CalPls

  CalPls values are joint calibration offsets. You can view and edit the values from [System Configuration]-[Robots]-[Robot**]-[Calibration]. However, it is recommended that you use the Robot Calibration wizard to set these values. These values are unique for each robot and are supplied from the factory. CalPls values are used to calibrate joint position after replacing a motor or encoder.

These are one-time settings for each robot. Additional robot parameters can be set from the Robot Manager.

To change robot parameters, follow these steps:

1. In the Epson RC+ 8.0 menu, select [Setup]-[System Configuration].
2. Select [Controller]-[Robots]-[Robot**]-[Calibration].
3. Execute the calibration wizard or change values for Hofs or CalPls.
4. Click [Apply] to make the changes permanent.

Saving robot calibration data

You can save and load individual robot calibration files. This is useful for moving a robot from one controller to another. When you save calibration data, a file is created with an MPD file extension. This file contains Hofs and CalPls values.

To save robot calibration data

1. In the Epson RC+ 8.0 menu, select [Setup]-[System Configuration].
2. Select [Controller]-[Robots]-[Robot**]-[Calibration].
3. Ensure that the robot serial number is correct. The serial number will be used to create the default file name. It is recommended that the serial number be used.
4. Click the [Save] button. Specify the destination directory and click the [Save] button.

Loading robot calibration data

To load robot calibration data

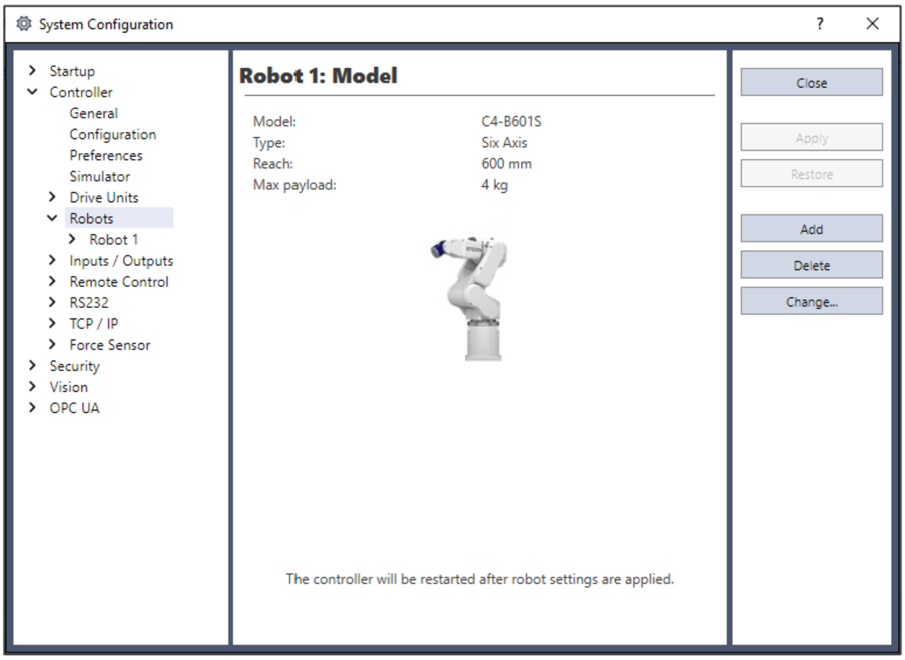
1. In the Epson RC+ 8.0 menu, select [Setup]-[System Configuration].
2. Select [Controller]-[Robots]-[Robot**]-[Calibration].
3. Click the [Load] button.
4. Select the desired MPD file and click the [Open] button.

11.1.4 Deleting a standard robot

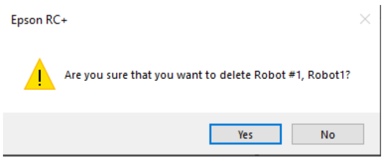
1. In the Epson RC+ 8.0 menu, select [Setup]-[System Configuration].

2. Select [Controller]-[Robots]-[Robot**].

   > **KEY POINTS**
   >
   > You can only delete the last robot.



3. Click the [Delete] button. The message shown below will appear.



4. Click the [Yes] button. The controller will be restarted.

KEY POINTS

If you delete only an additional axis from its installed robot, see , below.

Deleting the additional axes

11.1.5 Changing the robot model

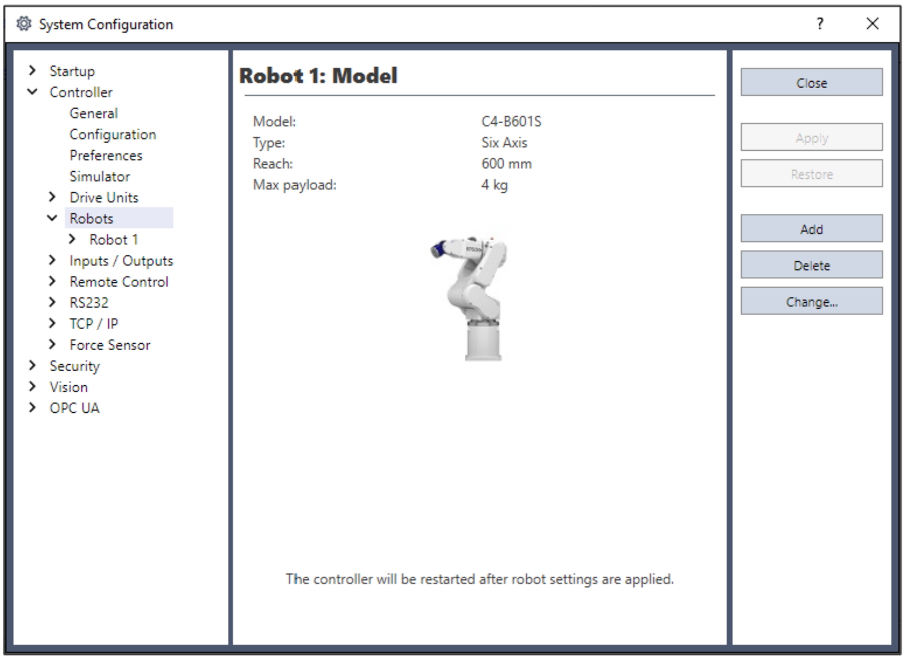
CAUTION

Changing the manipulator should be done with great caution. It initializes the robot calibration parameters (Hofs, CalPls), additional axis information, and PG parameter data. A Safety Board password is necessary when using Controller with Safety Board. Before changing the robot, make sure to save the calibration data by following the procedure below.

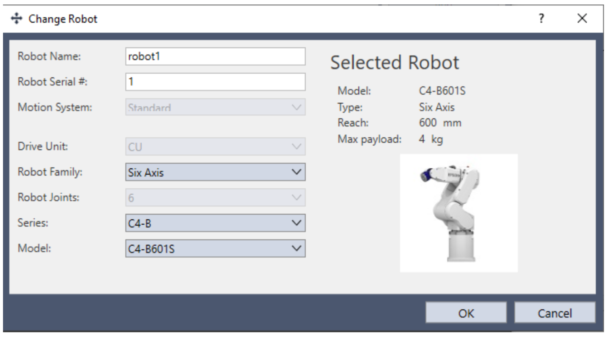
1. In the Epson RC+ 8.0 menu, select [Setup]-[System Configuration].
2. Select [Controller]-[Robots]-[Robot**]-[Calibration]. Then click the [Save] button.

1. In the Epson RC+ 8.0 menu, select [Setup]-[System Configuration].

2. Select [Controller]-[Robots]-[Robot**].



3. Click the [Change] button. The dialog shown below appears.



4. Input the robot name and serial number printed on the name plate of the manipulator. Any serial number can be used, but it is recommended that you use the number that is stamped on the manipulator.

5. Select the robot type in the [Robot Family] box.

6. Select the series name of the manipulator in the [Series] box.

7. Select the robot model in the [Model] box. Available robots will be displayed according to the format of the currently installed motor driver. If you use [Dry run], all robots selected in step 6 will be shown.

8. Click the [OK] button. The controller will be restarted.

KEY POINTS

For the Controller with Safety Board, assign a robot using the safety function to Robot 1.

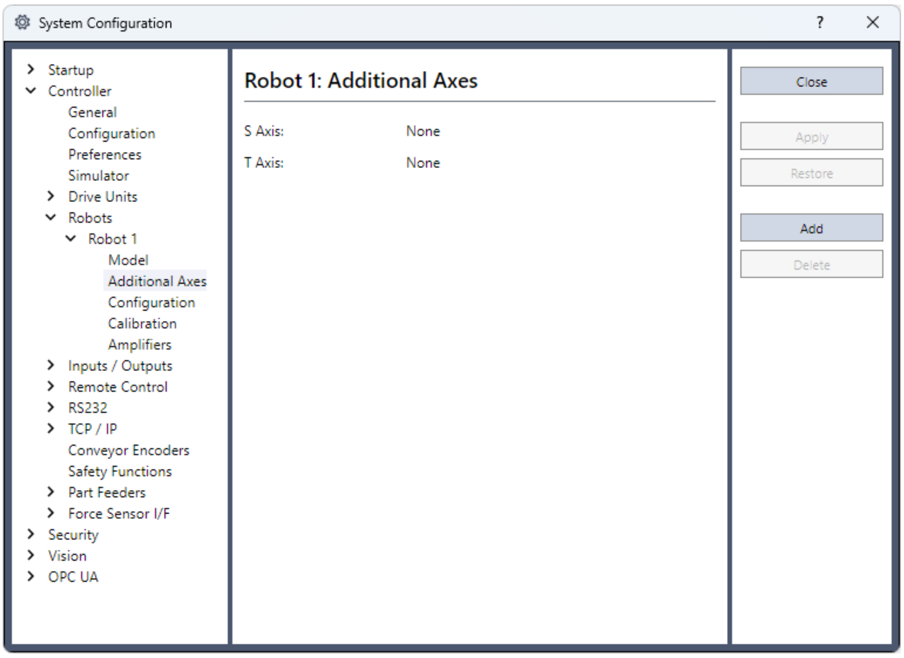
11.2 Configuration of Additional Axes

Using the additional axes feature, you can configure the axes that move with the manipulator.

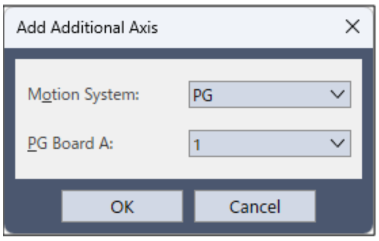
You can configure up to two additional axes (S and T).

11.2.1 Adding the additional S axis

1. In the Epson RC+ 8.0 menu, select [Setup]-[System Configuration]-[Robots]-[Robot**]-[Additional Axes].



2. Click the [Add] button. The dialog shown below appears.



3. Select “PG” for a motion system.

4. Select a [PG Board A].

5. Click the [OK] button. The controller will be restarted.

11.2.2 Adding the additional T axis

KEY POINTS

After the additional S axis has been added to the robot, you can add the additional T axis.

The procedure is the same as for the S axis. See below.

Adding the additional S axis

11.2.3 Changing the parameters of robot with additional axes installed

For details, refer to the following manual.

"Robot Controller Option PG Motion System"

11.2.4 Differences of the standard robot and robot with additional axes

The robot with additional axes installed has some parts which are different from the standard robot when using GUI and SPEL+ commands.

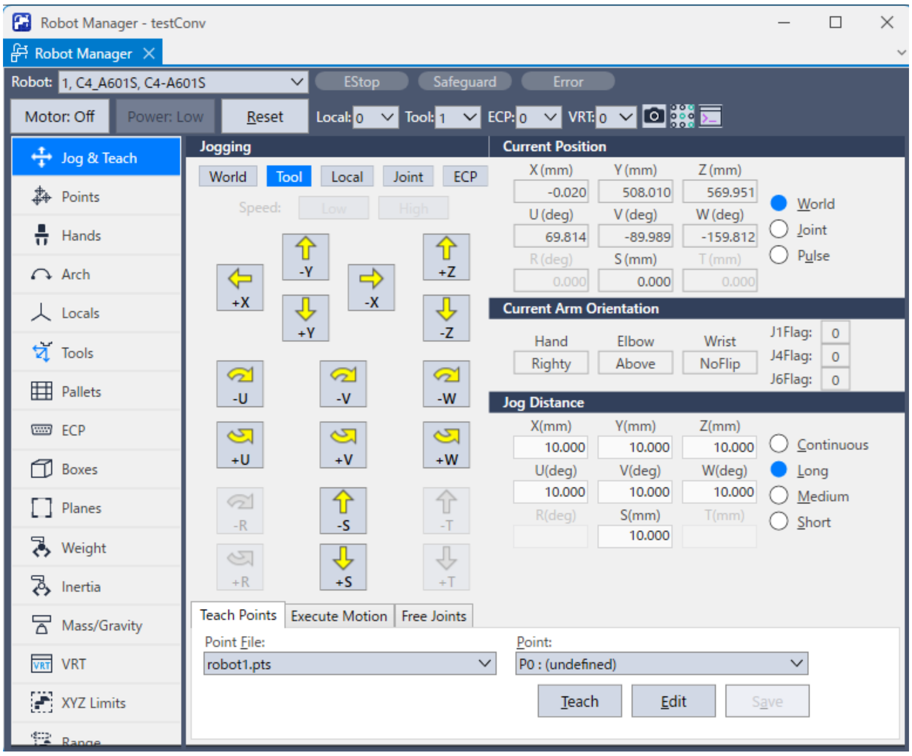
For instructions on SPEL+ command, refer to the following manual.

"SPEL+ Language Reference"

The main differences in the Epson RC+ 8.0 GUI are as below.

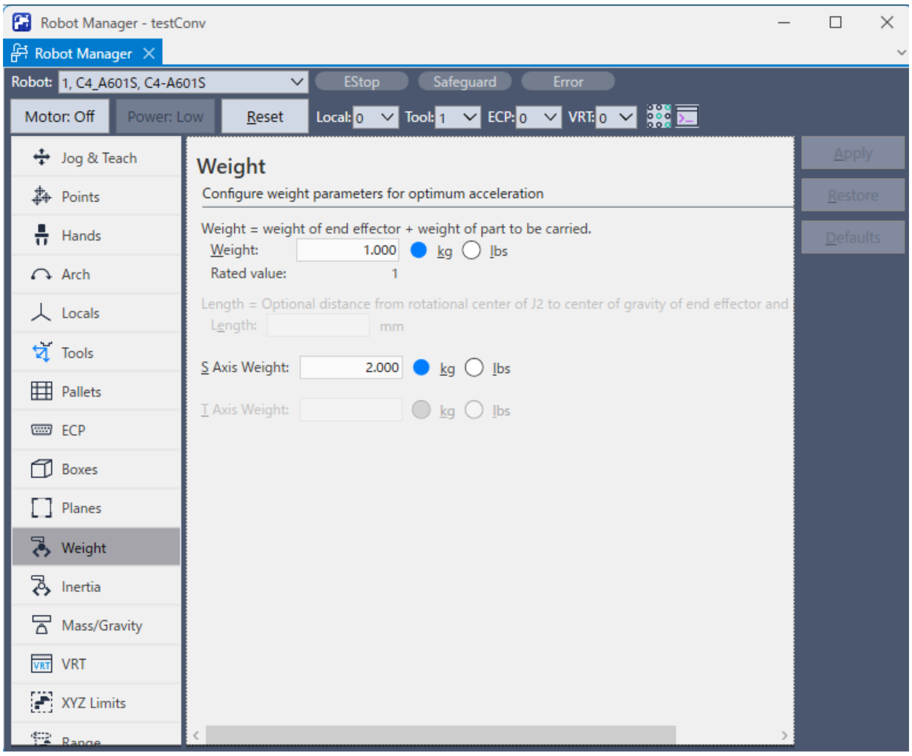
[Tools]-[Robot Manager]-[Jog & Teach] Panel

You can jog the additional S and T axes. When the additional T axis is not installed, the jog buttons will be dimmed.



[Tools]-[Robot Manager]-[Weight] Panel

You can change the Weight parameters for S and T axes. When the additional T axis is not installed, the corresponding weight setting will be dimmed.



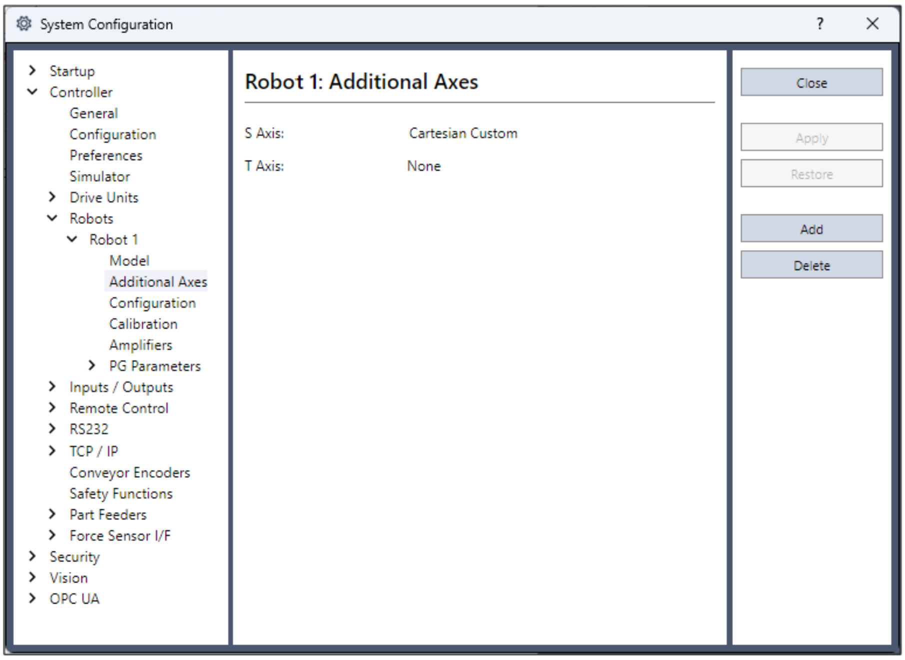
11.2.5 Deleting the additional axes

KEY POINTS

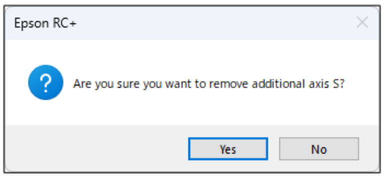
- When the additional T axis is installed, delete it first.
- When only the additional S axis is installed, delete it.

1. In the Epson RC+ 8.0 menu, select [Setup]-[System Configuration].

2. Select [Controller]-[Robots]-[Robot**]-[Additional Axes].



3. Click the [Delete] button. The message shown below will appear.



4. Click the [Yes] button. The controller will be restarted.

12. Inputs and Outputs

12.1 Overview

The controller I/O has the following types of I/O:

Standard I/O
: This digital I/O comes standard with the controller.

Expansion I/O
: This is optional digital I/O that can be added to the controller to expand standard I/O. The boards with 24 inputs and 16 outputs can be added. (Boards cannot be added to T, VT series Manipulator.)

Fieldbus master I/O
: An optional board for the controller to expand the standard I/O. You can add one of the following boards which support the fieldbus master board. (PC)
DeviceNet, EtherNet/IP, PROFIBUS-DP

Fieldbus slave I/O
: The Fieldbus slave includes the standard functions (Modbus RTU and Modbus TCP), and the options. As an option to add to the controller, you can add one board (controller: RC700 series, RC90 series) or module (manipulator: T, VT, Controller: RC800 series) corresponding to the following fieldbus slaves:
DeviceNet, EtherNet/IP, PROFIBUS-DP, CC-Link, PROFINET, EtherCAT

Hand I/O
: Standard digital I/O for T series.

Memory I/O
: This is built-in memory bits that can be used for inter-task communications.

Analog I/O
: This is option to add analog input/output function to the controller. (Boards cannot be added to T, VT series Manipulator.)

For Standard, Expansion, Fieldbus master, and Fieldbus slave I/O, there are input bits numbered starting with 0, and output bits numbered starting with 0.

For memory I/O, each memory bit is both an input and an output.

For details on I/O wiring, refer to the manuals below.

- "Robot Controller Manual": RC800 Series, RC700 Series, RC700-D, RC700-E, RC90 Series
- "Manipulator Manual" T, VT Series

12.2 I/O Commands

The SPEL+ language has several commands for inputs and outputs listed below. For details on commands, refer to the following manual:

"SPEL+ Language Reference"

Input Commands

In
: Reads one byte of input bits.

InBCD
: Reads one byte of input bits in Binary Coded Decimal format.

InW
: Reads one word of input bits.

Oport
: Reads one output bit.

Sw
: Reads one input bit.

Output Commands

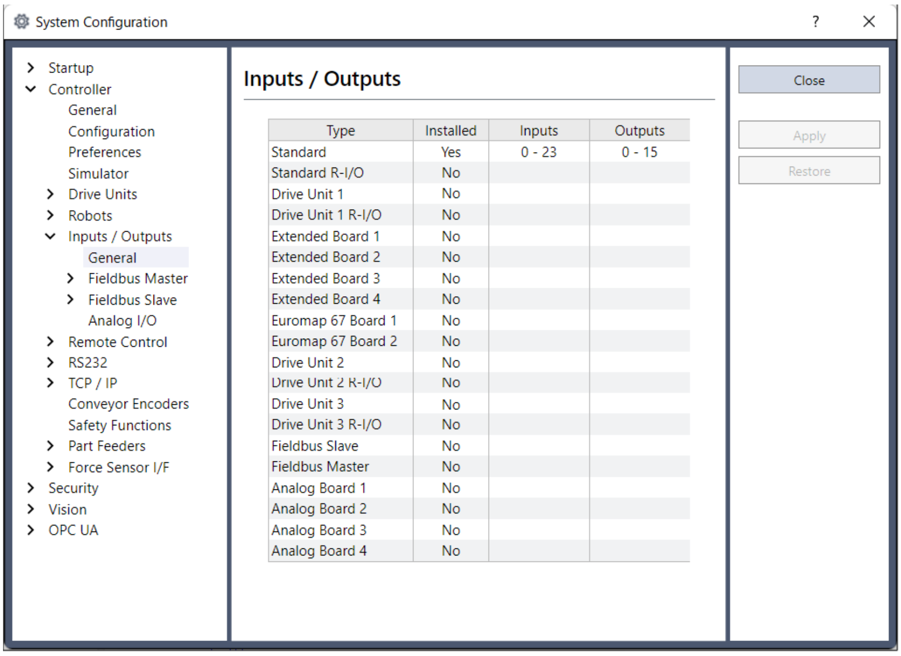
Off
    Turns off one output bit with optional time.
On
    Turns on one output bit with optional time.
OpBCD
    Sets one byte of output bits in Binary Coded Decimal format.
Out
    Sets / reads one byte of output bits.
OutW
    Sets / reads one word of output bits.

Memory I/O Commands

MemOff
    Turns off one memory bit.
MemOn
    Turns on one memory bit.
MemOut
    Sets / reads one byte of memory bits.
MemSw
    Reads one bit of memory.

12.3 I/O Configuration

To view the current I/O configuration, click [Setup]-[System Configuration]-[Controller]-[Inputs/Outputs]. This will show you what I/O is installed on the controller.



Standard I/O

The expansion I/O board is automatically configured by the controller. To add expansion I/O boards, refer to Robot Controller manual

"Robot Controller Manual"

The standard I/O in the Drive Unit automatically increases depending on the number of Drive Unit.

Fieldbus master I/O and fieldbus slave I/O

For details on how to configure, add, and check boards, refer to the following manual:

"Robot Controller Option: Fieldbus I/O Manual"

Analog I/O

The board is automatically configured by the controller. To configure, add, or confirm the analog I/O boards, refer to the following manual.

"Robot Controller Manual"

12.4 Monitoring I/O

To monitor I/O, use the I/O Monitor tool by selecting [Tools]-[I/O Monitor]. From the I/O monitor, you can view inputs and outputs or memory I/O in bit, byte, and word formats.

See details below.

[I/O Monitor] Command (Tools Menu)

12.5 Virtual I/O

The Controller supports virtual I/O. When enabled, virtual I/O allows you to simulate your hardwired I/O. You can turn on / off any input bit or output bit. Normally this is used when the controller is in Dry Run mode with no robot or I/O connected.

Virtual I/O Commands

- SetIn Set the value of an : 8 bit input port.
- SetInW Set the value of a : 16 bit input port.
- SetSw Set the value of one input bit.

12.6 Fieldbus Master I/O

The Fieldbus master I/O is an option.

For details on how to use, refer to the following manual:

"Robot Controller Option: Fieldbus I/O Manual"

12.7 Fieldbus Slave I/O

The Fieldbus slave I/O includes the standard functions (Modbus RTU and Modbus TCP), and the options.

For types and usage of optional Fieldbus slaves, refer to the following manual:

"Robot Controller Option: Fieldbus I/O"

12.7.1 Modbus Slave

Modbus TCP and Modbus RTU can be used as the Fieldbus slave I/O as standard.

KEY POINTS

Modbus cannot be used when other Fieldbus slave boards are installed.

Modbus has several derived protocols. Although connection with the standard Modbus protocol has been confirmed, use the Modbus slave in the system after checking connectivity with equipment to be connected.

12.7.2 Supported Functions

The Controller supports following Modbus functions.

Function code	Function name	Description
1	Read Coil Status	Use this function to read the status of the input bit port. No broadcast.
2	Read Input Status	Use this function to read the status of the output bit port. No broadcast.
3	Read Holding Registers	Use this function to read the status of the input word port. No broadcast.
4	Read Input Registers	Use this function to read the status of the output word port. No broadcast.
5	Force Single Coil	Use this function to configure an input bit port.
6	Preset Single Register	Use this function to configure an input word port.
15	Force Multiple Coils	Use this function to configure several input bit ports.
16	Preset Multiple Registers	Use this function to configure several input word port.

12.7.3 Address map

Input I/O		Output I/O

Fieldbus I/O Address		Modbus Address	
Word	Bit	Holding register	Coil
32	512	40032	512
	513		513
	514		514
	515		515
	516		516
	517		517
	518		518
	519		519
	520		520
	521		521
	522		522
	523		523
	524		524
	525		525
	526		526
	527		527
33	528	40033	528
	529		529
	530		530
	531		531
	532		532
	533		533
	534		534
	535		535
	536		536
	537		537
	538		538
	539		539
	540		540
	541		541
	542		542
	543		543

Fieldbus I/O Address		Modbus Address	
Word	Bit	Holding register	Coil
32	512	30032	10512
	513		10513
	514		10514
	515		10515
	516		10516
	517		10517
	518		10518
	519		10519
	520		10520
	521		10521
	522		10522
	523		10523
	524		10524
	525		10525
	526		10526
	527		10527
33	528	30033	10528
	529		10529
	530		10530
	531		10531
	532		10532
	533		10533
	534		10534
	535		10535
	536		10536
	537		10537
	538		10538
	539		10539
	540		10540
	541		10541
	542		10542
	543		10543

159	2544	40159	2544
	2545		2545
	2546		2546
	2547		2547
	2548		2548
	2549		2549
	2550		2550
	2551		2551
	2552		2552
	2553		2553
	2554		2554
	2555		2555
	2556		2556
	2557		2557
	2558		2558
	2559		2559

159	2544	30159	12544
	2545		12545
	2546		12546
	2547		12547
	2548		12548
	2549		12549
	2550		12550
	2551		12551
	2552		12552
	2553		12553
	2554		12554
	2555		12555
	2556		12556
	2557		12557
	2558		12558
	2559		12559

KEY POINTS

Note that the addresses are specified with the numbers fewer than the originals by 1. The address to access to the input bit port 512 is 511.

12.7.4 Modbus RTU

Modbus RTU is the Fieldbus which uses the serial communication. It can be used with the RS-232C port which is installed to the Controller as standard, and the optional extended RS-232C port.

12.7.5 Modbus TCP

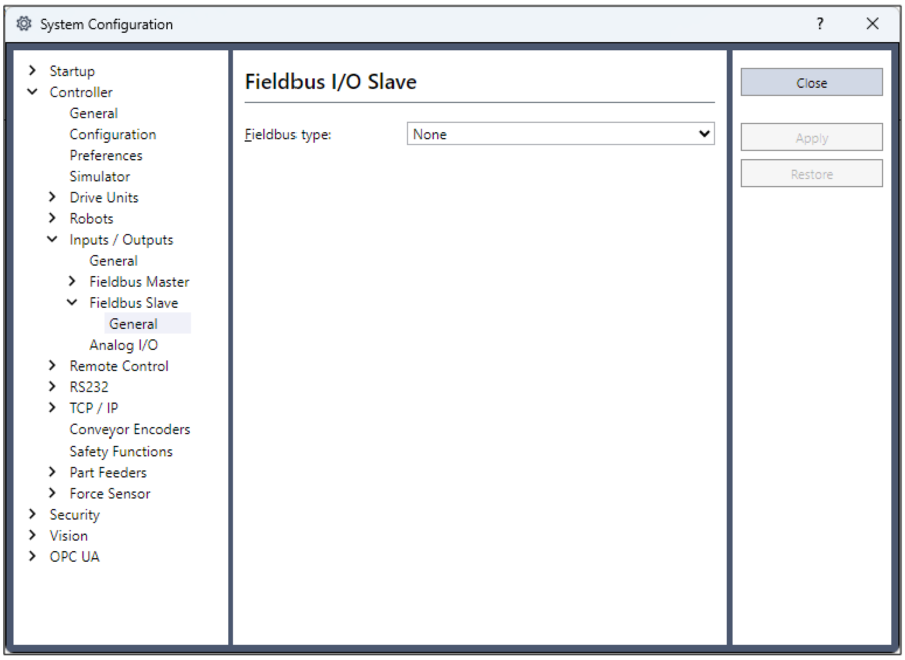
Modbus TCP is the Fieldbus which uses the Ethernet communication (socket communication). It can be used with the Ethernet installed to the Controller as standard.

12.7.6 How to Configure Modbus

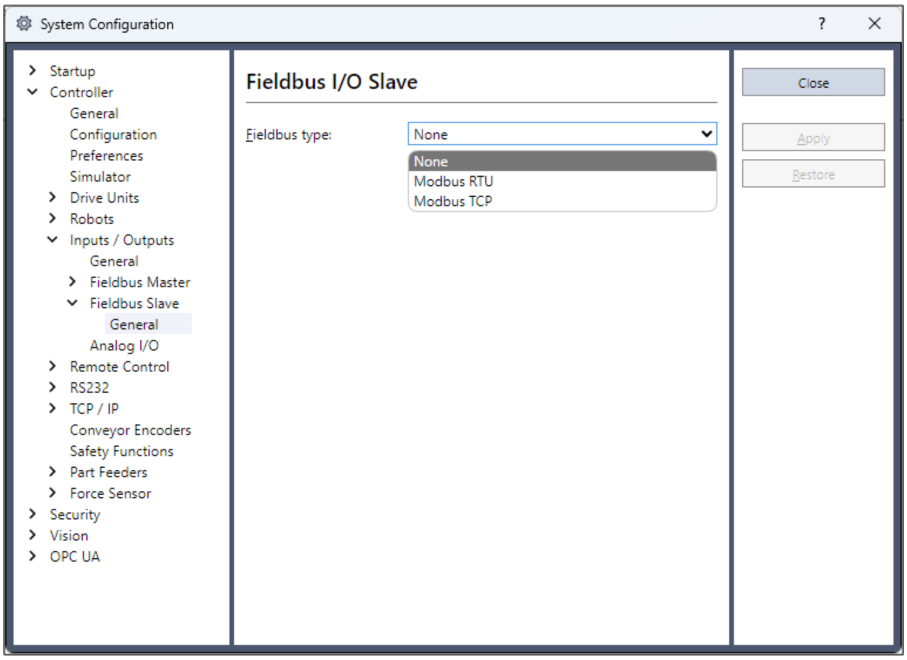
Use of Modbus

Modbus can be enabled in the following dialog box. This dialog box appears when the optional Fieldbus slave board is not installed.

[System Configuration]-[Controller]-[Inputs/Outputs]-[Fieldbus Slave]-[General]



Select any of "Disable", "Modbus RTU", or "Modbus TCP" from the pull-down menu.

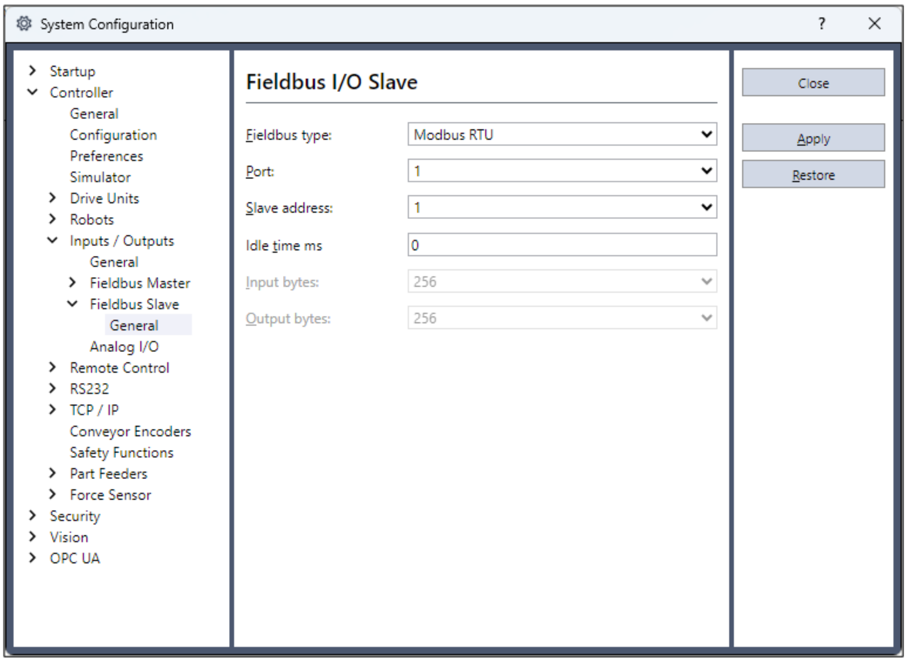


KEY POINTS

Modbus does not function if the Fieldbus slave board is installed while either Modbus RTU or Modbus TCP is selected. However, setting will be held.

Modbus RTU detailed settings

If "Modbus RTU" is selected in [Fieldbus Type:], the detailed setting window for using Modbus RTU is displayed. Set each item.



[Port]

Select the serial port number to be used. Other settings such as the baud rate are done in the RS-232C configuration dialog box (other menu).

KEY POINTS

- If unused port number is selected, a controller error occurs after rebooting the Controller.
- To change the settings of the selected port such as a baud rate, disable Modbus in advance. Settings cannot be changed if the port is set to Modbus.

[Slave Address]

For the Modbus RTU slaves, the slave address set to the transmission frame is checked and the request for that address is only processed.

KEY POINTS

Set the desired address. Be careful not to conflict with other equipment.

[Idle time]

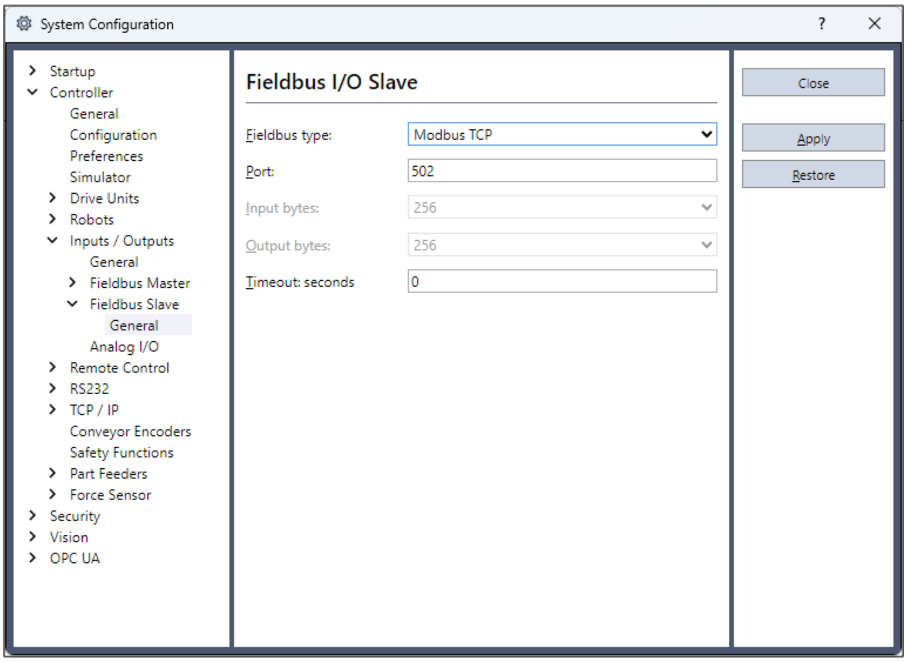
Set the idle time to be added to the sending frame specified by the Modbus RTU protocol. By the protocol specification, the time for 3.5 characters is defined before and after the sending frame.

The idle time can be set in units of 1ms. If "0" is specified for the setting value, the time for 3.5 characters will be set.

Set this item if the connected equipment cannot receive a response with the time for 3.5 characters.

Modbus TCP detailed settings

If "Modbus TCP" is selected for the [Fieldbus Type], the dialog box to enable Modbus TCP will be displayed. Set each item.



[Port]

Select the port number to be used. Default is "502".

KEY POINTS

Set the port number not to conflict with other system.

[Timeout]

In [Timeout], set the time to automatically disconnect if there is no transmission or reception after the port is connected. If the connection is lost, connect again.

Set the time from 0 seconds to 60 seconds in increments of 1-second.

CAUTION

If you set "0" in the [Timeout] box, time out duration is infinite. In this case, the task continues to execute even without the communication from client. This means the robot may keep moving and cause unexpected damage. Ensure the ways other than the communication to stop the task.

KEY POINTS

While connecting with ModBus TCP, the error 7103 "Fieldbus slave. Timeout error occurred during I/O data transform." could occur on the controller. The error may not be cleared when you reset it immediately after it occurs. After occurring the error, reset it after 10ms has passed.

13. Remote Control

By using Input/Output, Ethernet (TCP/IP), and RS-232C, the controller can control manipulators from an external device.

The external device can execute several commands, including Motor On/Off, Start, Pause, Continue, and Stop.

For the remote I/O You must function, refer to the following manual:

"Epson RC+ 8.0"Remote Control Reference"

13.1 Remote I/O

There are three basic steps required for remote control configuration:

1. Configure Remote Control inputs and outputs using the [Remote Control] tab on the [Setup]-[System Configuration]-[Remote Control] page.

   In default, the remote function is assigned to Input #0 to 7 and Output #0 to 8 in advance. If you need to change it, please set it.

2. Set the control device to remote on the [Setup]-[System Configuration]-[Configuration] page.

   To accept external remote inputs, assign the remote function and set remote to the control device. When control device setting is remote, the controller is only controllable from the remote device.

Remote control function can be used in the following systems.

Example: Control the robot from a PLC

Use remote control to control the robot (controller) from a PLC.

When using a PLC, you will need to be familiar with the handshake required to use remote inputs. See details below.

Remote Input Handshake Timing

Example: Control the robot using a push button box with buttons and lights

The lights are connected to remote control outputs on the controller to indicate status, such as AutoMode, MotorOn, Error, etc.

The buttons are connected to remote inputs to control motor power and start programs.

For details of each I/O connection, refer to the following manuals:

- "Robot Controller Functions"
  - "I/O Connector"
  - "I/O Remote Settings"
  - "Expansion I/O board"
- "Robot Controller Option Fieldbus I/O"

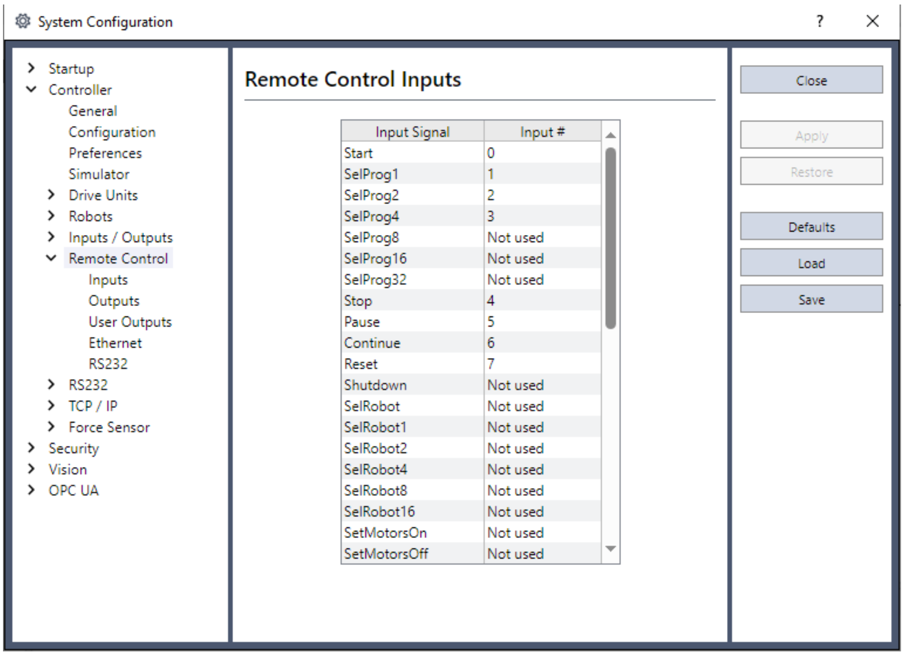
13.1.1 Remote Control Input Output Configuration

The following is the procedure to assign the remote control functions to the I/O system.

1. The [System Configuration] command opens a dialog that contains several pages that are used to configure the system for the Epson RC+ 8.0 environment. Select [Controller]-[Remote]-[Inputs] or [Outputs].

2. For each input or output you want to use for remote control, click on the Input # or Output # cell for the desired signal, then click the dropdown arrow and select a bit number in the list.

3. Click the [Apply] button to save the new setting, then click the [Close] button.



For details on the dialog setting, see below.

[Setup]-[System Configuration]-[Controller]-[Remote Control]

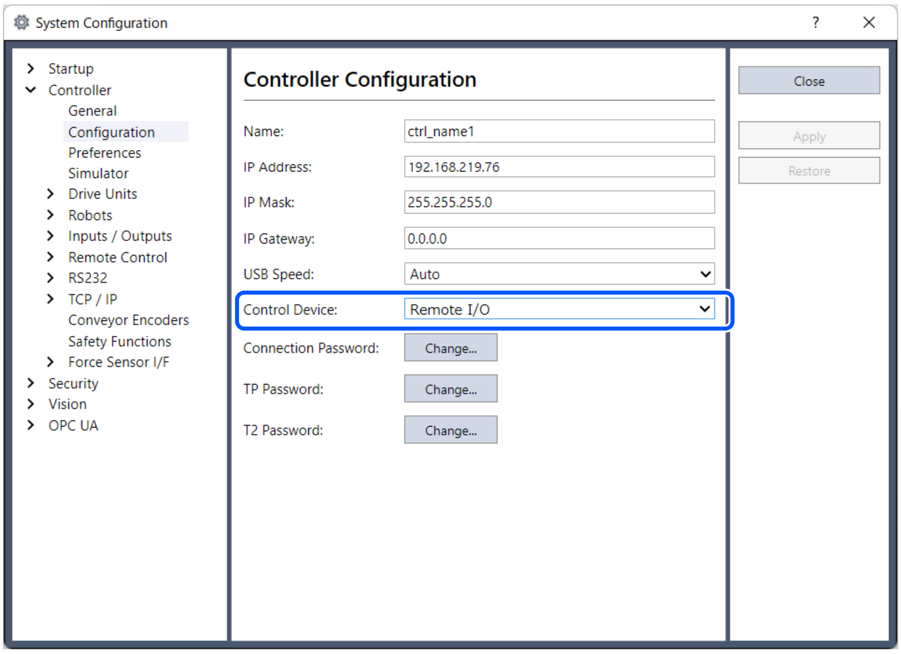
13.1.2 Control Device Configuration

The following is the procedure to set the control device to "Remote I/O".

1. The [System Configuration] command opens a dialog that contains several pages that are used to configure the system for the Epson RC+ 8.0 environment. Select [Controller]-[Configuration].

   Select "Remote I/O" in the [Control Device] box.

2. Click the [Apply] button to save the new setting, then click the [Close] button.



For details on the dialog setting, see below.

[Setup]-[System Configuration]-[Controller]-[Configuration] Page

13.1.3 Auto Mode with Remote Control

To run in auto cycle with remote control

1. The host device (e.g. PLC) should wait for the AutoMode and Ready remote output to turn on before issuing remote commands.
2. Now the remote input commands will be accepted.

To monitor remote operation from the Epson RC+ 8.0 Operator Window

1. Set the Epson RC+ 8.0 Start Up Mode to “Auto”. See details below.

   Start Mode

2. The PC should also be configured to automatically log into Windows and start Epson RC+ 8.0 at Windows start. See details below.

   Auto Start

13.1.4 Teach Mode with Remote Control

When using a remote control with Teach Mode ON, no remote input commands can be used. Remote status outputs will still operate.

> ⚠ WARNING
>
> Remote status outputs (such as MotorOn, Home, etc.) will operate when Teach Mode is ON, even when the enable switch (dead man's switch) is disengaged.
>
> Remote status outputs (MotorOn, Home, etc.) will operate if the operation mode is TEACH even when the enable switch (deadman switch) is disabled.

You can monitor teach mode status using the TeachMode remote output.

13.1.5 Debugging Remote Control

You can debug programs using Remote Control from the Epson RC+ 8.0 development environment. To run programs by remote control for debugging:

1. Create a program in the same manner as usual.
2. Open the Run window and click the [Enable Remote I/O] button.
3. Now the remote commands will be accepted.

Breakpoint setting and output to the Run window is available.

KEY POINTS

If you cannot wire the I/O, use virtual I/O mode for debugging. Remote function is also available when virtual I/O is enabled.

13.1.6 Remote Input

Remote inputs are used to control the Manipulators and start programs. Certain conditions must be met before inputs are enabled, as shown in the table below.

To accept external remote inputs, assign the remote function and set remote to the control device. When external remote input is available, "AutoMode output" turns ON.

Except "SelProg", the signals execute each function when the signal starts in input acceptance condition. The function executes automatically. Therefore, no special programming is needed.

KEY POINTS

- If error occurs, execute the Reset command to clear the error condition before executing the operation command. Use the "Error output" and "Reset input" to monitor the error status and clear error conditions from the remote device.
- If the remote input command does not meet the input acceptance conditions, a CmdError signal is output. The CmdError signal is not set as the default for remote I/O output signals. When using the remote function, set the CmdError signal for the remote I/O output signal.

13.1.6.1 For RC90, RC700 :T, VT series:

Name	Default	Description	Input acceptance condition (*1)
Start	0	Executes function selected at SelProg. (*2) (*13)	Ready output ON Error output OFF EStopOn output OFF SafeguardOn output OFF EStopOff output ON Pause input OFF Stop input OFF
SelProg1	1	Specifies the executing Main function number. (*2)	
SelProg2	2		
SelProg4	3		
SelProg8	Not set		
SelProg16	Not set		
SelProg32	Not set		

Name	Default	Description	Input acceptance condition (*1)
Stop	4	All tasks and commands are stopped.	
Pause	5	All tasks are paused. (*3)	Running output ON
Continue	6	Continues the paused task.	Paused output ON Pause input OFF Stop input OFF
Reset	7	Resets emergency stop and error. (*4)	Ready output ON
Shutdown	Not set	Terminates the system	
ForcePowerLow	Not set	Operates as the forced low power function. The robot is operated in the low power mode. Power High control from the command is not accepted. Executes the following according to the controller preferences. Stops or temporarily stops all the tasks and commands. (*12)	Any time This input is acceptable even AutoMode output is OFF.
SelRobot	Not set	Changes the output condition of MotorsOn, AtHome, PowerHigh, and MCalReqd. (*9)	
SelRobot1 SelRobot2 SelRobot4 SelRobot8 SelRobot16	Not set	Specify the number of robot which executes a command. (*5)	
SetMotorsOn	Not set	Turn ON robot motors. (*5) (*6)	Ready output ON EStopOn output OFF SafeguardOn output OFF EStopOff output ON SetMotorsOff input OFF
SetMotorsOff	Not set	Turn OFF robot motors. (*5)	Ready output ON
SetPowerHigh	Not set	Set the robot power mode to High (*5)	Ready output ON EStopOn output OFF SafeguardOn output OFF EStopOff output ON SetPowerLow input OFF
SetPowerLow	Not set	Set the robot power mode to Low. (*5)	Ready output ON

Name	Default	Description	Input acceptance condition (*1)
Home	Not set	Move the Robot Arm to the home position defined by the user.	Ready output ON Error output OFF EStopOn output OFF SafeguardOn output OFF EStopOff output ON MotorsOn output ON Pause input OFF Stop input OFF
MCal	Not set	Execute MCal (*5) (*7)	Ready output ON Error output OFF EStopOn output OFF SafeguardOn output OFF EStopOff output ON MotorsOn output ON Pause input OFF Stop input OFF
Recover	Not set	After the safeguard is closed, recover to the position where the safeguard was open.(*8)	Paused output ON Error output OFF EStopOn output OFF SafeguardOn output OFF EStopOff output ON RecoverReqd output ON Pause input OFF Stop input OFF
ExtCmdSet	Not set	Commands for an extended remote IO. For details, refer to the following manual. "Remote Control Reference-. Remote I/O to Be Used"	
ExtRespGet	Not set		
ExtCmdReset	Not set		
ResetAlarm	Not set	Cancel the alarm (*11)(*14)	
SelAlarm1 SelAlarm2 SelAlarm4 SelAlarm8	Not set	Specify the alarm number to cancel (*10)(*14)	

ALIVE	Not set	Input signal for alive monitoring of the controller. Same signal as the input will be output to ALIVE output. The master equipment can perform alive monitoring of the controller by switching the input periodically and checking the output signal.	
ExtCmd_0-15	Not set	Commands for an extended remote IO. For details, refer to the following manual. "Remote Control Reference-. Remote I/O to Be Used"	
ExtCmd_16-31	Not set		
ExtCmd_32-47	Not set		
ExtCmd_48-63	Not set		
ExtCmd_64-79	Not set		
ExtCmd_80-95	Not set		
ExtCmd_96-111	Not set		
ExtCmd_112-127	Not set		

*1: "AutoMode output" ON is omitted from the table. This is an input acceptance condition for all functions.

*2: "Start input" executes Function specified by the following six bits: SelProg 1, 2, 4, 8, 16, and 32.

Function Name	SelProg1	SelProg2	SelProg4	SelProg8	SelProg16	SelProg32
Main	0	0	0	0	0	0
Main1	1	0	0	0	0	0
Main2	0	1	0	0	0	0
Main3	1	1	0	0	0	0
⋮						
Main60	0	0	1	1	1	1
Main61	1	0	1	1	1	1
Main62	0	1	1	1	1	1
Main63	1	1	1	1	1	1

0=OFF, 1=ON

*3: "NoPause task" and "NoEmgAbort task" do not pause. For details, refer to Help or the following manual:

"SPEL+ Language Reference- Pause""

*4: Turns OFF the I/O output and initializes the robot parameter. For details, refer to Help or the following manual:

"SPEL+ Language Reference- Reset""

*5: The values specified by "SelRobot1, 2, 4, 8, and 16" correspond to the robot numbers, when multiple robots are connected to the controller.

Robot number	SelRobot1	SelRobot2	SelRobot4	SelRobot8	SelRobot16
0(All)	0	0	0	0	0

Robot number	SelRobot1	SelRobot2	SelRobot4	SelRobot8	SelRobot16
1	1	0	0	0	0
2	0	1	0	0	0
3	1	1	0	0	0
⋮					
13	1	0	1	1	0
14	0	1	1	1	0
15	1	1	1	1	0
16	0	0	0	0	1

0=OFF, 1=ON

*6: The robot parameter will be initialized.

For details, refer to Help or the following manual:

"SPEL+ Language Reference- Motor"

*7: For details, refer to Help or the following manual:

"SPEL+ Language Reference- MCal"

*8: This is for experienced users only. Make sure that you fully understand the input specification before using.

CmdRunning output and CmdError output will not change for this input.

"NoEmgAbort task" will not stop by this input.

When the input changes from ON to OFF, all tasks and commands will stop.

*9: This function changes the output condition of MotorsOn, AtHome, PowerHigh, and MCalReqd.

By setting this signal with the condition selected using SelRobot1 - SelRobot16, you can switch the output condition.

Once you select the condition, it will be kept until you change it or turn off / restart the Controller. All manipulators are selected as default.

*10: The values specified by "SelAlarm1, 2, 4, and 8" correspond to the alarm numbers.

Alarm number	Target	SelAlarm1	SelAlarm2	SelAlarm4	SelAlarm8
1	Controller battery	1	0	0	0
2	Battery of the robot connected to CU	0	1	0	0
3	Grease of the robot connected to CU	1	1	0	0
4	Battery of the robot connected to DU1	0	0	1	0
5	Grease of the robot connected to DU1	1	0	1	0
6	Battery of the robot connected to DU2	0	1	1	0
7	Grease of the robot connected to DU2	1	1	1	0

Alarm number	Target	SelAlarm1	SelAlarm2	SelAlarm4	SelAlarm8
8	Battery of the robot connected to DU3	0	0	0	1
9	Grease of the robot connected to DU3	1	0	0	1

0=OFF, 1=ON

The following parts are subject to grease up.

- 6-axis robot: Bevel gear on the Joint #6
- SCARA, RS series: Ball screw spline unit on the Joint # 3

*11: The specified alarm can be canceled by selecting the conditions using SelAlarm1-SelAlarm8 and setting this signal.

*12: Operation of all tasks and commands, power mode of the robot, and PowerHigh command are executed by the setting value of the controller preferences.

- Preferences (1): "Motor power low when ForcePowerLow signal OFF"
- Preferences (2): "ForcePowerLow signal change pauses all tasks"

For details on controller preferences, see below.

[Setup]-[System Configuration]-[Controller]-[Preferences] Page

Preferences (1)	Preferences (2)	ForcePowerLow	All tasks and commands	Power mode	PowerHigh
0	0	1→0	Abort	Low only	Accept
0	0	0→1	Abort	Low only	Not accept
0	1	1→0	Continue	High/Low	Accept
0	1	0→1	Pause	Low only	Not accept
1	0	1→0	Abort	Low only	Not accept
1	0	0→1	Abort	Low only	Accept
1	1	1→0	Pause	Low only	Not accept
1	1	0→1	Continue	High/Low	Accept

*13: Do not execute the Restart command of the SPEL+ program and the Start signal of the remote input at the same time. Error 2503 may occur when the programs are executed at the same time.

*14: For firmware versions 7.5.3.X or later: the warning for the maintenance data cannot be canceled with ResetAlarm.

13.1.6.2 For RC800 series:

Name	Default	Description	Input acceptance condition (*1)
Start	0	Executes function selected at SelProg. (*2) (*3)	Ready output ON Error output OFF EStopOn output OFF SafeguardOn output OFF EStopOff output ON Pause input OFF Stop input OFF
SelProg1	1	Specifies the executing Main function number. (*2)	
SelProg2	2		
SelProg4	3		
SelProg8	Not set		
SelProg16	Not set		
SelProg32	Not set		
Stop	4	All tasks and commands are stopped.	
Pause	5	All tasks are paused. (*4)	Running output ON
Continue	6	Continues the paused task.(*5)	Paused output ON Pause input OFF Stop input OFF
ContinueManualRecover	Not set	Resumes a task that has been halted (manual recovery) (*6).	
Reset	7	Resets emergency stop and error. (*7)	Ready output ON
ForcePowerLow	Not set	Operates as the forced low power function. The robot is operated in the low power mode. Power High control from the command is not accepted. Executes the following according to the controller preferences. Stops or temporarily stops all the tasks and commands. (*8)	Any time This input is acceptable even AutoMode output is OFF.
SelRobot	Not set	Changes the output condition of MotorsOn, AtHome, PowerHigh, and MCalReqd. (*9)	
SelRobot1 SelRobot2 SelRobot4 SelRobot8 SelRobot16	Not set	Specify the number of robot which executes a command. (*10)	

Name	Default	Description	Input acceptance condition (*1)
SetMotorsOn	Not set	Turn ON robot motors. (*10) (*11)	Ready output ON EStopOn output OFF SafeguardOn output OFF EStopOff output ON SetMotorsOff input OFF
SetMotorsOff	Not set	Turn OFF robot motors. (*10)	Ready output ON
SetPowerHigh	Not set	Set the robot power mode to High (*10)	Ready output ON EStopOn output OFF SafeguardOn output OFF EStopOff output ON SetPowerLow input OFF
SetPowerLow	Not set	Set the robot power mode to Low. (*10)	Ready output ON
Home	Not set	Move the Robot Arm to the home position defined by the user.	Ready output ON Error output OFF EStopOn output OFF SafeguardOn output OFF EStopOff output ON MotorsOn output ON Pause input OFF Stop input OFF

Name	Default	Description	Input acceptance condition (*1)
MCal	Not set	Execute MCal (*10) (*12)	Ready output ON Error output OFF EStopOn output OFF SafeguardOn output OFF EStopOff output ON MotorsOn output ON Pause input OFF Stop input OFF
Recover	Not set	After the safeguard is closed, recover to the position where the safeguard was open.(*13)(*14)	Paused output ON Error output OFF EStopOn output OFF SafeguardOn output OFF EStopOff output ON RecoverReqd output ON Pause input OFF Stop input OFF
ExtCmdSet	Not set	Commands for an extended remote IO. For details, refer to the following manual. "Remote Control Reference-. Remote I/O to Be Used"	
ExtRespGet	Not set		
ExtCmdReset	Not set		
ALIVE	Not set	Input signal for alive monitoring of the controller. Same signal as the input will be output to ALIVE output. The master equipment can perform alive monitoring of the controller by switching the input periodically and checking the output signal.	
ExtCmd_0-15	Not set	Commands for an extended remote IO. For details, refer to the following manual. "Remote Control Reference-. Remote I/O to Be Used"	
ExtCmd_16-31	Not set		
ExtCmd_32-47	Not set		
ExtCmd_48-63	Not set		
ExtCmd_64-79	Not set		
ExtCmd_80-95	Not set		
ExtCmd_96-111	Not set		

Name	Default	Description	Input acceptance condition (*1)
ExtCmd_112-127	Not set		
SelAxis	Not set	Target Robot	
SelAxis1-4	Not set	Target axis (*15)	
ResetCtrlParts	Not set	Clear Maintenance data (for Controllers) (*16)	
SelCtrlParts1-8	Not set	Select Maintenance data (for Controllers) (*16)	
ResetRbParts	Not set	Clear Maintenance data (for robots) (*17)	
SelRbParts1-8	Not set	Select Maintenance data (for robots) (*17)	

*1: "AutoMode output" ON is omitted from the table. This is an input acceptance condition for all functions.

*2: "Start input" executes Function specified by the following six bits: SelProg 1, 2, 4, 8, 16, and 32.

Function Name	SelProg1	SelProg2	SelProg4	SelProg8	SelProg16	SelProg32
Main	0	0	0	0	0	0
Main1	1	0	0	0	0	0
Main2	0	1	0	0	0	0
Main3	1	1	0	0	0	0
⋮						
Main60	0	0	1	1	1	1
Main61	1	0	1	1	1	1
Main62	0	1	1	1	1	1
Main63	1	1	1	1	1	1

0=OFF, 1=ON

*3: Do not execute the Restart command of the SPEL+ program and the Start signal of the remote input at the same time. Error 2503 may occur when the programs are executed at the same time.

*4: Pause command is not available for "NoPause task" and "NoEmgAbort task". For details, refer to Help or the following manual:

"SPEL+ Language Reference- Pause"

*5: Continues paused tasks.

The "Auto safeguard position recovery" setting in the Controller preferences is for control from Epson RC+8.0. This setting does not enable or disable position recovery with the command.

If you want to control the enabling/disabling of auto recovery by remote command, use the Continue command and ContinueManualRecover command separately.

*6: Continues paused tasks.

With the Continue command, processing equivalent to the Recover command is also performed. Since no processing equivalent to the Recover command is performed with this command, the "Auto safeguard position recovery" is not performed.

The Recover command must be executed before this command is executed.

[Use case]

This command is used to return to the original position (safeguard open position) with the Recover command and resume the program while confirming that no collision will occur during the return (Recover Position) operation.

[How to recover/return]

- Safeguard closed → Recover → ContinueManualRecover
- Safeguard closed →Continue
- Safeguard closed →ContinueManualRecover → Continue

[Supplementary description]

Before executing this command, perform a Recover Position operation with the Recover command. Use the Continue command to perform the recovery operation and resume task execution all at once. If you execute the ContinueManualRecover command without executing the Recover command, an error will result.

*7: I/O output will be turned off and the robot parameter will be initialized. For details, refer to Help or the following manual:

"SPEL+ Language Reference- Reset"

*8: Operation of all tasks and commands, power mode of the robot, and PowerHigh command are executed by the setting value of the controller preferences.

- Preferences (1): "Motor power low when ForcePowerLow signal OFF"
- Preferences (2): "ForcePowerLow signal change pauses all tasks"

For details on controller preferences, see below.

[Setup]-[System Configuration]-[Controller]-[Preferences] Page

Preferences (1)	Preferences (2)	ForcePowerLow	All tasks and commands	Power mode	PowerHigh
0	0	1→0	Abort	Low only	Accept
0	0	0→1	Abort	Low only	Not accept
0	1	1→0	Continue	High/Low	Accept
0	1	0→1	Pause	Low only	Not accept
1	0	1→0	Abort	Low only	Not accept
1	0	0→1	Abort	Low only	Accept
1	1	1→0	Pause	Low only	Not accept
1	1	0→1	Continue	High/Low	Accept

*9: This function changes the output condition of MotorsOn, AtHome, PowerHigh, and MCalReqd.

By setting this signal with the condition selected using SelRobot1 - SelRobot16, you can switch the output condition.

Once you select the condition, it will be kept until you change it or turn off / restart the Controller. All manipulators are selected as default.

*10: The values specified by "SelRobot1, 2, 4, 8, and 16" correspond to the robot numbers, when multiple robots are connected to the controller.

Robot number	SelRobot1	SelRobot2	SelRobot4	SelRobot8	SelRobot16
0(All)	0	0	0	0	0
1	1	0	0	0	0
2	0	1	0	0	0
3	1	1	0	0	0

0=OFF, 1=ON

With the RC800 series, 0 to 4 can be specified. If 5 or higher is specified, an error will occur because the robot cannot be registered.

*11: The robot parameter will be initialized. For details, refer to Help or the following manual:

"SPEL+ Language Reference- Motor"

*12: For details, refer to Help or the following manual:

"SPEL+ Language Reference- MCal"

*13: This is for experienced users only. Make sure that you fully understand the input specification before using.

CmdRunning output and CmdError output will not change for this input.

"NoEmgAbort task" will not stop by this input.

When the input changes from ON to OFF, all tasks and commands will stop.

*14: After the safeguard is closed, recover to the position where the safeguard was open..

The "Auto safeguard position recovery" setting in the Controller preferences is for control from Epson RC+8.0. This setting does not enable or disable position recovery with the command.

If you want to control the enabling/disabling of auto recovery by remote command, use the Continue command and ContinueManualRecover command separately.

*15: Select conditions with SelAxis1-SelAxis4 and perform axis switching with SelAxis.

Name	Default	Description	Input acceptance condition
SelAxis	Not set	Change the Maintenance command's reflection conditions (*a)	AutoMode output ON
SelAxis1 SelAxis2 SelAxis4	Not set	Specify the number of axis which executes a command. (*b)	AutoMode output ON

*a: Switches conditions for reflecting Maintenance.

By setting this signal with the condition selected using SelAxis1 - SelAxis4, you can switch the target condition.

Once you select the condition, it will be kept until you change it or turn off / restart the Controller. All Axis are selected as default.

The selectable axis numbers vary depending on SelAxis.

If a non-targeted axis is selected, the command execution is ignored.

*b: The values specified by "SelAxis1, 2, 4" correspond to the robot axis number.

You can specify from axis 1 to 6.

Axis number	SelAxis1	SelAxis2	SelAxis4
0 (Reserve)	0	0	0
1	1	0	0
2	0	1	0
⋮			
6	0	1	1
7 (Reserve)	1	1	1

0=OFF, 1=ON

*16: Select conditions with SelCtrlParts1 - SelCtrlParts8, and clear Controller Maintenance data with ResetCtrlParts.

Name	Default	Description	Input acceptance condition
ResetCtrlParts	Not set	Clear Controller Maintenance data (*a)	AutoMode output ON
SelCtrlParts1 SelCtrlParts2 SelCtrlParts4 SelCtrlParts8	Not set	Specify the Maintenance number to be cleared (*b)	AutoMode output ON

*a: Select conditions with SelCtrlParts1 - SelCtrlParts8 to clear the specified Maintenance data. Data is cleared regardless of whether errors and warnings occur.

*b: The target of data clearing is specified by SelCtrlParts1-8 and executed by ResetCtrlParts.

The value specified by the four bits of "SelCtrlParts1, 2, 4, 8" is a part-specifying number used to reset the data.

SelCtrlParts1	SelCtrlParts2	SelCtrlParts4	SelCtrlParts8	Parts (Controller)
0	0	0	0	(Reserve)
1	0	0	0	Battery
0	1	0	0	(Reserve)
⋮				
0	1	1	1	(Reserve)
1	1	1	1	(Reserve)

0=OFF, 1=ON

*17: Select conditions with SelRbParts1 - SelRbParts8, and clear robot Maintenance data with ResetRbParts.

Name	Default	Description	Input acceptance condition
ResetRbParts	Not set	Clear Controller Maintenance data (*a)	AutoMode output ON

Name	Default	Description	Input acceptance condition
SelRbParts1 SelRbParts2 SelRbParts4 SelRbParts8	Not set	Specify the Maintenance number to be cleared (*b)	AutoMode output ON

*a: Select conditions with SelRbParts1 - SelRbParts8 to clear the specified Maintenance data. Data is cleared regardless of whether errors and warnings occur.

*b: The target of data clearing is specified by SelRbParts1-8 and executed by ResetRbParts.

The value specified by the four bits of "SelRbParts1, 2, 4, 8" is a part-specifying number used to reset the data.

SelRbParts1	SelRbParts2	SelRbParts4	SelRbParts8	Parts (Robot)
0	0	0	0	(Reserve)
1	0	0	0	Battery
0	1	0	0	Timing belt
1	1	0	0	Grease
0	0	1	0	Motor
1	0	1	0	Reduction gear unit
0	1	1	0	Ball screw spline
1	1	1	0	(Reserve)
⋮				
0	1	1	1	Reserve
1	1	1	1	Reserve

0=OFF, 1=ON

13.1.7 Remote Outputs

Remote outputs provide status for the Manipulator(s) and Controller.

Remote outputs provide the assigned function used with any control device. The outputs execute automatically. Therefore, no special programming is needed.

13.1.7.1 For RC90, RC700 :T, VT series:

Name	Default	Description
Ready	0	Turns ON when the controller startup completes and no task is running.
Running	1	Turns ON when task is running. Turn OFF when "Paused output" is ON.
Paused	2	Turns ON when pause task exists.
Error	3	Turn ON in the error condition. Use "Reset input" to recover from the error condition. (*14)

Name	Default	Description
EStopOn	Not set	Turns OFF except in Emergency Stop. Turns ON in Emergency Stop. Turns OFF when the controller power is OFF. (*11) (*12)
SafeguardOn	5	Turn ON with safety door open
SError	6	Turns ON when critical error occurs. When a critical error occurs, "Reset input" does not function. Reboot the controller to recover. (*14)
Warning	7	Turns ON when warning occurs. Task can be executed as usual even a warning condition. However, take action for the warning as soon as possible. (*14)
EStopOff	8	Turns ON except in Emergency Stop. Turns OFF in Emergency Stop. Turns OFF when the controller power is OFF. (*12)
MotorsOn	Not set	Turns ON when the robot motor is ON. (*5)
AtHome	Not set	Turns ON when the robot is in the home position. (*5)
PowerHigh	Not set	Turns ON when the robot's power mode is High. (*5)
MCalReqd	Not set	Turns ON when the robot hasn't executed MCal. (*5)
RecoverReqd	Not set	Turns ON when at least one robot is waiting for Recover after the safeguard is closed.
RecoverInCycle	Not set	Turns ON when at least one robot is executing Recover.
WaitingRC	Not set	Turns ON when the Controller is waiting to connect with RC+.
CmdRunning	Not set	Turns ON when an input command is executing.
CmdError	Not set	Turns ON when an input command cannot be accepted.
CurrProg1 CurrProg2 CurrProg4 CurrProg8 CurrProg16 CurrProg32	Not set	Indicates the running or the last main function number (*1)
AutoMode	Not set	Turns ON in remote input acceptable status. (*2)
TeachMode	Not set	Turns ON in TEACH mode. (*12)
TestMode	Not set	Turns ON in TEST mode.
EnableOn	Not set	Turns ON when the enable switch is ON. (*12)
ErrorCode1 ⋮ ErrorCode8192	Not set	Indicates the error number.
InsideBox1 ⋮ InsideBox15	Not set	Turns ON when the robot is in the approach check area. (*3)

Name	Default	Description
InsidePlane1 : InsidePlane15	Not set	Turns ON when a robot is on the approach plane area. (*4)
Alarm	Not set	Turns ON when any of the alarms is occurring. (*9) (*15)
Alarm1	Not set	Turns ON when a battery alarm of the controller is occurring. (*13) (*15)
Alarm2	Not set	Turns ON when a battery alarm of the robot connected to CU is occurring. (*13) (*15)
Alarm3	Not set	Turns ON when a grease alarm of the robot connected to CU is occurring. (*10) (*13) (*15)
Alarm4	Not set	Turns ON when a battery alarm of the robot connected to DU1 is occurring. (*13) (*15)
Alarm5	Not set	Turns ON when a grease alarm of the robot connected to DU1 is occurring. (*10) (*13) (*15)
Alarm6	Not set	Turns ON when a battery alarm of the robot connected to DU2 is occurring. (*13) (*15)
Alarm7	Not set	Turns ON when a grease alarm of the robot connected to DU2 is occurring. (*10) (*13) (*15)
Alarm8	Not set	Turns ON when a battery alarm of the robot connected to DU3 is occurring. (*13) (*15)
Alarm9	Not set	Turns ON when a grease alarm of the robot connected to DU3 is occurring. (*10) (*13) (*15)
PositionX	Not set	Outputs current X coordinate in the World coordinate system. (*6) (*7)
PositionY	Not set	Outputs current Y coordinate in the World coordinate system. (*6) (*7)
PositionZ	Not set	Outputs current Z coordinate in the World coordinate system. (*6) (*7)
PositionU	Not set	Outputs current U coordinate in the World coordinate system. (*6) (*7)
PositionV	Not set	Outputs current V coordinate in the World coordinate system. (*6) (*7)
PositionW	Not set	Outputs current W coordinate in the World coordinate system. (*6) (*7)
Torque1	Not set	Outputs the current torque value of Joint #1. (*6) (*7)
Torque2	Not set	Outputs the current torque value of Joint #2. (*6) (*7)
Torque3	Not set	Outputs the current torque value of Joint #3. (*6) (*7)
Torque4	Not set	Outputs the current torque value of Joint #4. (*6) (*7)
Torque5	Not set	Outputs the current torque value of Joint #5. (*6) (*7)
Torque6	Not set	Outputs the current torque value of Joint #6. (*6) (*7)
CPU	Not set	Outputs the CPU load factor of the user program. (*8)
ESTOP	Not set	Outputs how many times emergency stops have been executed.
ALIVE	Not set	Output signal for alive monitoring of the controller. The signal input by ALIVE input will be output. The master equipment can perform alive monitoring of the controller by switching the input periodically and checking the output signal.
ForceControlOn	Not set	Turns ON when the robot is executing the force control function. (*5)

Name	Default	Description
ExtCmdGet	Not set	Commands for an extended remote IO. For details, refer to the following manual. "Remote Control Reference - 4. Remote I/O to Be Used"
ExtRespSet	Not set	
ExtCmdResult	Not set	
ExtError	Not set	
ExtResp_0-15	Not set	
ExtResp_16-31	Not set	
ExtResp_32-47	Not set	
ExtResp_48-63	Not set	
ExtResp_64-79	Not set	
ExtResp_80-95	Not set	
ExtResp_96-111	Not set	
ExtResp_112-127	Not set	

*1: Outputs the current or the last function number of "CurrProg1, 2, 4, 8, 16, or 32".

Function Name	CurrProg1	CurrProg2	CurrProg4	CurrProg8	CurrProg16	CurrProg32
Main	0	0	0	0	0	0
Main1	1	0	0	0	0	0
Main2	0	1	0	0	0	0
Main3	1	1	0	0	0	0
⋮						
Main60	0	0	1	1	1	1
Main61	1	0	1	1	1	1
Main62	0	1	1	1	1	1
Main63	1	1	1	1	1	1

0=OFF, 1=ON

*2: Remote function is available in the following conditions.

- The setting is Auto mode and the control device is remote.
- The setting is Program mode and Remote I/O is enabled.

*3: For details, refer to Help or the following manual:

"SPEL+ Language Reference - - Box"

*4: For details, refer to Help or the following manual:

- "SPEL+ Language Reference - - Plane"

*5: Manipulator status is output as follows, according to the condition selected in SelRobot. Wait at least 40 ms before inputting the signal after changing the condition in SelRobot.

Name	(SelRobot1 - SelRobot16) condition when inputting SelRobot	
	0: All robots are selected	1 - 16: Particular robot number is selected
MotorsOn	Turns ON when at least one motor is ON.	Turns ON when the motor of the selected robot is ON.
AtHome	Turns ON when all robots are in the home position.	Turns ON when the selected robot is in the home position.
PowerHigh	Turns ON when at least one robot's power mode is High.	Turns ON when the selected robot's power mode is High.
MCalReqd	Turns ON when at least one robot hasn't executed MCal.	Turns ON when the selected robot hasn't executed MCal.
ForceControlOn	Turns ON when at least one robot is executing the force control function.	Turns ON when the selected robot is executing the force control function.

*6: Outputs information of the selected robot when SelRobot1, SelRobot2, SelRobot4, SelRobot8, and SelRobot16 are set. If not, information of Robot 1 will be output.

*7: Outputs information in Real format.

*8: Outputs the total load factor of the user created tasks. For details on the CPU load factor, refer to the task manager.

*9: The signal turns on when the alarm occurs either in the controller alarm information or the robot alarm information.

*10: For details about the target of grease up, refer to each manipulator manuals.

*11: EStopOn is not recommended because the output in the emergency stop state and the controller power off state do not match. Assign EStopOff to output the emergency stop status.

The default setting has also been changed to the specification that EStopOff is assigned.

When using EPSON RC+7.0 Ver. 7.5.0 or earlier in connection with a newly purchased Controller, EstopOff output can be used according to the following procedure:

1. Set the remote I/O output settings to their default status.
2. Reset each remote I/O output.

*12: Do not use the following signals for safety-related functions. The signals do not meet Cat 3&PLd.

- EStopOn
- EStopOff
- TeachMode
- EnableOn

*13: For firmware versions 7.5.2.x or earlier: The occurrence of battery alarm and grease alarm are monitored every 5 minutes. The alarm occurrence and output timing on the controller are different. It may be output up to 5 minutes after the alarm is occurred on the controller.

For firmware versions 7.5.2.x or earlier: “Alarm” will on when battery alarm or grease alarm of Controller or Manipulator occurs with “Robot maintenance” enabled. For maintenance, refer to the following manuals:

- For FRC700-D, RC700-E: "Robot Controller Manual - Alarm"
- For RC700, RC90 series: "Robot Controller Maintenance Manual - Alarm"

- For T, VT series: "Robot Maintenance Manual - Alarm"

*14: Following shows status number/error number of output (Error, SError, Warning).

- Error: 1000 to 8000s
- SError: 9000s
- Warning: 410 to 900s

For more information of status number/error number, refer to following manual.

"Status Code / Error Code List" manual

*15: For firmware versions 7.5.3.x or later: When a battery or grease is set in the robot maintenance, Alarm or Alarm 1 to 9 will not be activated.

13.1.7.2 For RC800 series:

Name	Default	Description
Ready	0	Turns ON when the controller startup completes and no task is running.
Running	1	Turns ON when task is running. Turn OFF when "Paused output" is ON.
Paused	2	Turns ON when pause task exists.
Error	3	Turn ON in the error condition. Use "Reset input" to recover from the error condition. (*1)
EStopOn	Not set	Turns OFF except in Emergency Stop. Turns ON in Emergency Stop. Turns OFF when the controller power is OFF. (*2) (*3)
SafeguardOn	5	Turn ON with safety door open
SError	6	Turns ON when critical error occurs. When a critical error occurs, "Reset input" does not function. Reboot the controller to recover. (*1)
Warning	7	Turns ON when warning occurs. Task can be executed as usual even a warning condition. However, take action for the warning as soon as possible. (*1)
EStopOff	8	Turns ON except in Emergency Stop. Turns OFF in Emergency Stop. Turns OFF when the controller power is OFF. (*3)
MotorsOn	Not set	Turns ON when the robot motor is ON. (*4)
AtHome	Not set	Turns ON when the robot is in the home position. (*4)
PowerHigh	Not set	Turns ON when the robot's power mode is High. (*4)
MCalReqd	Not set	Turns ON when the robot hasn't executed MCal. (*4)
RecoverReqd	Not set	Turns ON when at least one robot is waiting for Recover after the safeguard is closed.
RecoverInCycle	Not set	Turns ON when at least one robot is executing Recover.
WaitingRC	Not set	Turns ON when the Controller is waiting to connect with RC+.
CmdRunning	Not set	Turns ON when an input command is executing.

Name	Default	Description
CmdError	Not set	Turns ON when an input command cannot be accepted.
CurrProg1 CurrProg2 CurrProg4 CurrProg8 CurrProg16 CurrProg32	Not set	Indicates the running or the last main function number (*5)
AutoMode	Not set	Turns ON in remote input acceptable status. (*6)
TeachMode	Not set	Turns ON in TEACH mode. (*3)
TestMode	Not set	Turns ON in TEST mode.
EnableOn	Not set	Turns ON when the enable switch is ON. (*3)
ErrorCode1 ⋮ ErrorCode8192	Not set	Indicates the error number.
InsideBox1 ⋮ InsideBox15	Not set	Turns ON when the robot is in the approach check area. (*7)
InsidePlane1 ⋮ InsidePlane15	Not set	Turns ON when a robot is on the approach plane area. (*8)
PositionX	Not set	Outputs current X coordinate in the World coordinate system. (*9) (*10)
PositionY	Not set	Outputs current Y coordinate in the World coordinate system. (*9) (*10)
PositionZ	Not set	Outputs current Z coordinate in the World coordinate system. (*9) (*10)
PositionU	Not set	Outputs current U coordinate in the World coordinate system. (*9) (*10)
PositionV	Not set	Outputs current V coordinate in the World coordinate system. (*9) (*10)
PositionW	Not set	Outputs current W coordinate in the World coordinate system. (*9) (*10)
Torque1	Not set	Outputs the current torque value of Joint #1. (*9) (*10)
Torque2	Not set	Outputs the current torque value of Joint #2. (*9) (*10)
Torque3	Not set	Outputs the current torque value of Joint #3. (*9) (*10)
Torque4	Not set	Outputs the current torque value of Joint #4. (*9) (*10)
Torque5	Not set	Outputs the current torque value of Joint #5. (*9) (*10)
Torque6	Not set	Outputs the current torque value of Joint #6. (*9) (*10)
CPU	Not set	Outputs the CPU load factor of the user program. (*11)
ESTOP	Not set	Outputs how many times emergency stops have been executed.

Name	Default	Description
ALIVE	Not set	Output signal for alive monitoring of the controller. The signal input by ALIVE input will be output. The master equipment can perform alive monitoring of the controller by switching the input periodically and checking the output signal.
ForceControlOn	Not set	Turns ON when the robot is executing the force control function. (*4)
ExtCmdGet	Not set	Commands for an extended remote IO. For details, refer to the following manual. "Remote Control Reference - 4. Remote I/O to Be Used"
ExtRespSet	Not set	
ExtCmdResult	Not set	
ExtError	Not set	
ExtResp_0-15	Not set	
ExtResp_16-31	Not set	
ExtResp_32-47	Not set	
ExtResp_48-63	Not set	
ExtResp_64-79	Not set	
ExtResp_80-95	Not set	
ExtResp_96-111	Not set	
ExtResp_112-127	Not set	
GetPartsStsCtrl0-15	Not set	Maintenance status (Controllers) (*12)
GetPartsStsRb0-15	Not set	Maintenance status (robots) (*13)

*1: Following shows status number/error number of output (Error, SError, Warning).

- Error: 1000 to 8000s
- SError: 9000s
- Warning: 410 to 900s

For more information of status number/error number, refer to following manual.

"Status Code / Error Code List" manual

*2: EStopOn is not recommended because the output in the emergency stop state and the controller power off state do not match. Assign EStopOff to output the emergency stop status.

The default setting has also been changed to the specification that EStopOff is assigned.

When using EPSON RC+7.0 Ver. 7.5.0 or earlier in connection with a newly purchased Controller, EstopOff output can be used according to the following procedure:

1. Set the remote I/O output settings to their default status.
2. Reset each remote I/O output.

*3: Do not use the following signals for safety-related functions. The signals do not meet Cat 3&PLd.

- EStopOn
- EStopOff
- TeachMode
- EnableOn

*4: Manipulator status is output as follows, according to the condition selected in SelRobot. Wait at least 40 ms before inputting the signal after changing the condition in SelRobot.

Name	(SelRobot1 - SelRobot16) condition when inputting SelRobot	
	0: All robots are selected	1 - 16: Particular robot number is selected
MotorsOn	Turns ON when at least one motor is ON.	Turns ON when the motor of the selected robot is ON.
AtHome	Turns ON when all robots are in the home position.	Turns ON when the selected robot is in the home position.
PowerHigh	Turns ON when at least one robot's power mode is High.	Turns ON when the selected robot's power mode is High.
MCalReqd	Turns ON when at least one robot hasn't executed MCal.	Turns ON when the selected robot hasn't executed MCal.
ForceControlOn	Turns ON when at least one robot is executing the force control function.	Turns ON when the selected robot is executing the force control function.

*5: Outputs the current or the last function number of CurrProg1, 2, 4, 8, 16, or 32.

Function Name	CurrProg1	CurrProg2	CurrProg4	CurrProg8	CurrProg16	CurrProg32
Main	0	0	0	0	0	0
Main1	1	0	0	0	0	0
Main2	0	1	0	0	0	0
Main3	1	1	0	0	0	0
⋮						
Main60	0	0	1	1	1	1
Main61	1	0	1	1	1	1
Main62	0	1	1	1	1	1
Main63	1	1	1	1	1	1

0=OFF, 1=ON

*6: Remote function is available in the following conditions.

- The setting is Auto mode and the control device is remote.
- The setting is Program mode and Remote I/O is enabled.

*7: For details, refer to Help or the following manual:

"SPEL+ Language Reference - - Box"

*8: For details, refer to Help or the following manual:

- "SPEL+ Language Reference - - Plane"

*9: Outputs information of the selected robot when SelRobot1, SelRobot2, SelRobot4, SelRobot8, and SelRobot16 are set. If not, information of Robot 1 will be output.

*10: Outputs information in Real format.

*11: Outputs the total load factor of the user created tasks. For details on the CPU load factor, refer to the task manager.

*12: Warning status of each part is indicated by bit.

Bit	Parts (Controller)
0	(Reserve)
1	Battery
2	(Reserve)
⋮	
14	(Reserve)
15	(Reserve)

*13: Warning status of each part is indicated by bit.

Bit	Parts (Robot)
0	(Reserve)
1	Battery
2	Timing belt
3	Grease
4	Motor
5	Reduction gear unit
6	Ball screw spline
7	(Reserve)
⋮	
14	(Reserve)
15	(Reserve)

13.1.8 Remote Input Handshake Timing

The following charts indicate the timing sequences for the primary operations of the Controller.

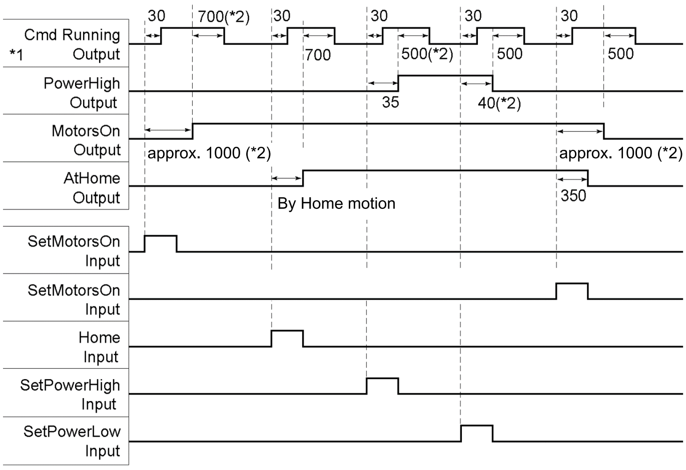
The indicated time lapses (time durations) should be referred to only as reference values. The actual timing values vary depending on some factors such as the numbers of manipulators and running tasks.

Check carefully and refer to the following charts for the timing interrelation when you enter an input signal. During system design, make sure that you actuate only one remote input operation at a time, otherwise an error will occur.

The pulse width of an input signal must be 25 or more milliseconds to be detected.

[Unit: msec]

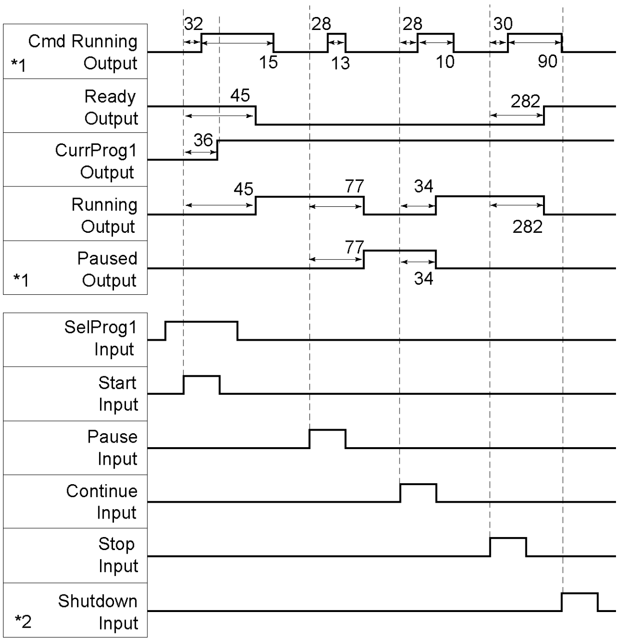
Timing Diagram for Operation Execution Sequence



*1 The motion of the CmdRunnig can be different from this figure according to the condition.

*2 Refer to only as reference value for a robot. It can be different according to the number of robots.

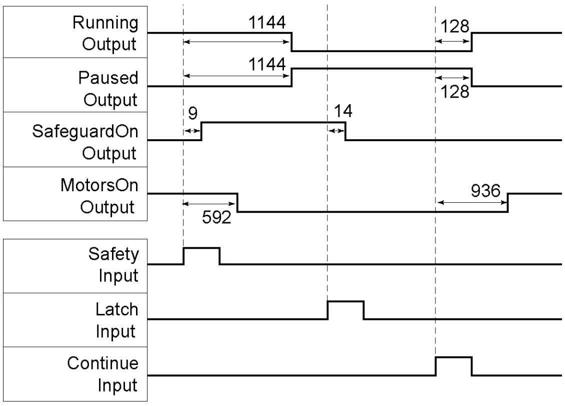
Timing Diagram for Program Execution Sequence



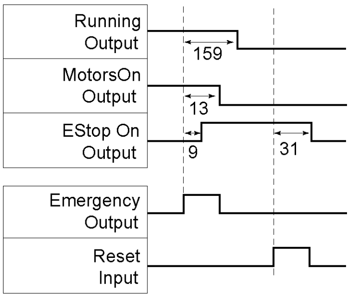
*1 Differs according to the setting condition of Quick Pause and the program running condition when the PAUSE is input.

*2 Shutdown input can be accepted when the Ready output is ON.

Timing Diagram for Safety Door Input Sequence

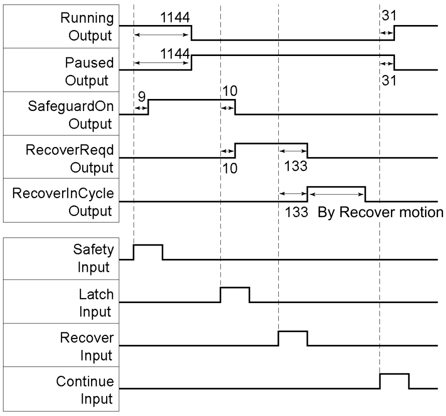


Timing Diagram for Emergency Stop Sequence



If an error occurs, the Error remote output is turned on. To clear the error, turn on the Reset input. In an error condition, no other input will be accepted.

Timing Diagram for Recover Sequence



13.2 Remote Ethernet

Remote Ethernet makes it possible to control the robot and controller from external equipment by sending the remote commands through Ethernet (TCP/IP).

> **KEY POINTS**
>
> For enhanced security, passwords must be set when connecting Controllers and PCs.

See details below.

- Security for Controller Ethernet Connection
- Security for Compact Vision CV2 Ethernet Connection

- Security for Part Feeding Ethernet Connection

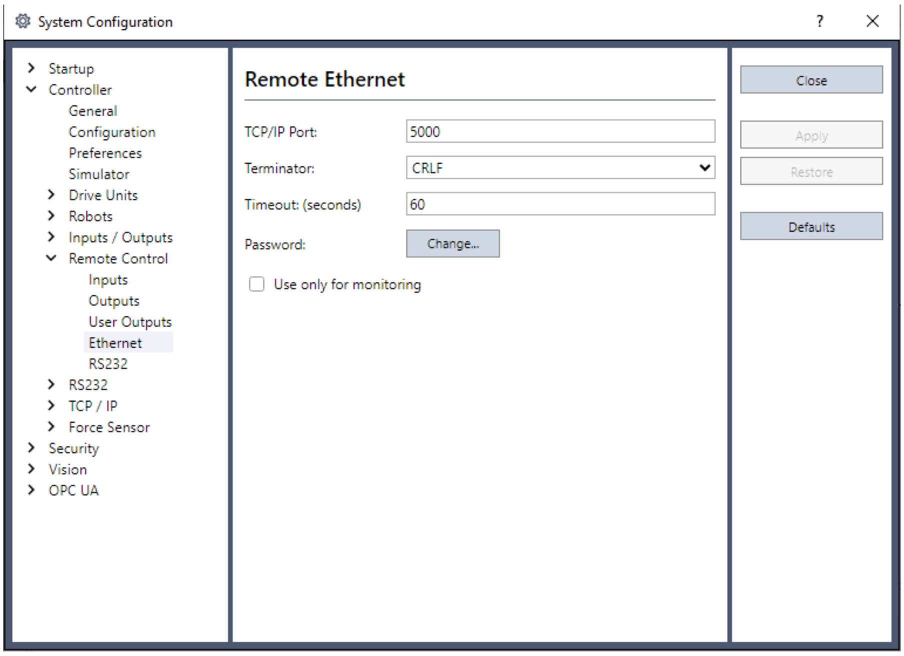
13.2.1 Remote Ethernet Configuration

To set the remote Ethernet functions valid, follow the procedures below to configure the parameter.

1. The [System Configuration] command opens a dialog that contains several pages that are used to configure the system for the Epson RC+ 8.0 environment. Select [Controller]-[Remote]-[Ethernet].

2. Configure the necessary items for the remote Ethernet control.

3. Click the [Apply] button to save the new setting, then click the [Close] button.



For details on the dialog setting, see below.

[Setup]-[System Configuration]-[Controller]-[Remote Control]

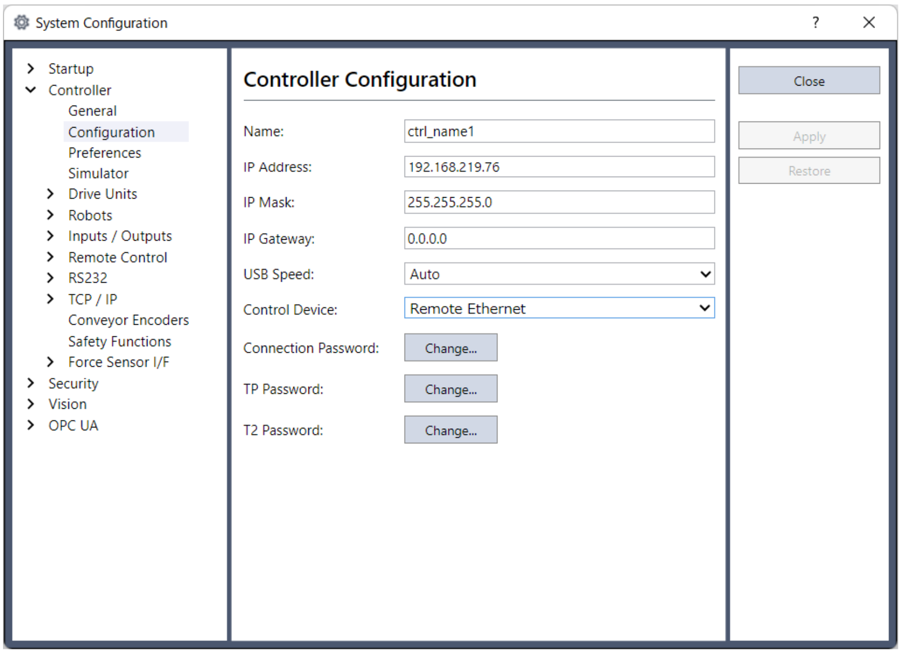
If you select "Use only for monitoring" and only acquire values using Remote Ethernet control, the "Control Device Configuration" in the next section will not be necessary .

13.2.2 Control Device Configuration

Set the control device to "Remote Ethernet" by the following procedure.

1. The [System Configuration] command opens a dialog that contains several pages that are used to configure the system for the Epson RC+ 8.0 environment. Select [Controller]-[Configuration]. Select "Remote Ethernet" in the [Controller device] box.

2. Click the [Apply] button to save the new setting, then click the [Close] button.



For details on the dialog setting, see below.

[Setup]-[System Configuration]-[Controller]-[Configuration] Page

13.2.3 Remote Ethernet Control Execution

Set the remote control available by the following procedure.

1. Connect from client equipment to the specified port in the Remote Ethernet of the Controller.
2. Specify the password set in the Remote Ethernet to the parameter and send the Login command.
3. Client equipment has to wait until Auto (GetStatus command response) is ON, before execution of remote command.
4. Now remote command will be accepted. Each command executes the function the input acceptance condition.

13.2.4 Debugging Remote Ethernet Control

Program debug from Epson RC+ 8.0 development environment is capable as follows.

- Create a program in the same manner as usual.
- Open the Run window and click the [Enable Remote ETH] button.

  When you only acquire values using the Remote Ethernet control, the [Enable Remote ETH] button is not displayed. Click the button of the device specified as the control device.
- Now the remote commands will be accepted.

  Breakpoint setting and output to the Run window is available.

KEY POINTS

- If not Login within 5 minutes from external equipment, the connection will be cut down automatically. After Login, if no command is sent within the timeout duration of the remote Ethernet, connection will be cut down. In this case, establish the connection again.
- If error occurs, execute the Reset command to clear the error condition before executing the operation command. To clear the error condition from external equipment by monitoring, use the "GetStatus" and "Reset" command.

CAUTION

If you set "0" in the [Timeout] box, time out duration is infinite. In this case, the task continues to execute even without the communication from client. This means the robot may keep moving and cause unexpected damage. Ensure the ways other than the communication to stop the task.

13.2.5 Remote Ethernet Command

Format: $ remote command {, parameter....} terminator

Note:

Remote command with parameters uses (,) (comma) to separate $ remote command and parameters. Do not put spaces before and after comma-separated characters and parameters.

13.2.5.1 For RC90, RC700 :T(VT series:

Remote command	Parameter	Contents	Input acceptance condition (*1)
Login	Password	Start the Controller Remote Ethernet function Authentication by password Execute Login correctly, commands execution is enabled until Logout	Available any time (*2)
Logout		Exit Controller Remote Ethernet function After Logout, execute the Login command to start remote Ethernet function. Logout during task execution causes an error.	Available any time (*2)
Start	Function No.	Execute the function of specified number (*3)(*11)	Auto ON Ready ON Error OFF EStop OFF Safeguard OFF
Stop		Stop all tasks and commands	Auto ON

Remote command	Parameter	Contents	Input acceptance condition (*1)
Pause		All tasks are paused. (*4)	Auto ON Running ON
Continue		Continue paused tasks	Auto ON Paused ON
Reset		Resets emergency stop and error. (*5)	Auto ON Ready ON
SetMotorsOn	Robot number	Turn ON the robot motor (*6)(*7)	Auto ON Ready ON EStop OFF Safeguard OFF
SetMotorsOff	Robot number	Turn OFF the robot motor (*7)	Auto ON Ready ON
SetCurRobot	Robot number	Select the manipulator	Auto ON Ready ON
GetCurRobot		Acquires the current manipulator number	Available any time (*2)
Home	Robot number	Move the arm to home position defined by user	Auto ON Ready ON Error OFF EStop OFF Safeguard OFF
GetIO	I/O bit No.	Acquire the specified I/O bit	Available any time (*2)
SetIO	I/O bit No. & value	Set the I/O specified bit 1: Turn ON the bit 0: Turn OFF the bit	Ready ON
GetIOByte	I/O port No.	Acquire the specified I/O port (8 bit)	Available any time (*1)
SetIOByte	I/O port No. & value	Set the I/O specified port (8 bit)	Ready ON
GetIOWord	I/O word port No.	Acquire the specified I/O word port (16 bit)	Available any time (*2)
SetIOWord	I/O word port No. & value	Set the I/O specified word port (8 bit)	Auto ON Ready ON
GetMemIO	Memory I/O bit No.	Acquire the specified memory I/O bit	Available any time (*2)
SetMemIO	Memory I/O bit No., value	Set the specified memory I/O bit 1: Turn ON the bit 0: Turn OFF the bit	Auto ON Ready ON
GetMemIOByte	Memory I/O Port number	Acquire the specified memory I/O port	Available any time (*2)

Remote command	Parameter	Contents	Input acceptance condition (*1)
SetMemIOByte	Memory I/O port No., value	Set the specified memory I/O port (8 bit)	Auto ON Ready ON
GetMemIOWord	Memory I/O word port No.	Acquire the specified memory I/O word port (16 bit)	Available any time (*2)
SetMemIOWord	Memory I/O word port No., value	Set the specified memory I/O word port (16 bit)	Auto ON Ready ON
GetVariable	Parameter name{,type}	Acquire the value of backup (Global Preserve) parameter (*8)	Available any time (*2)
	[Parameter name] (Array element), [Parameter name type], [Number to acquire]	Acquire the value of backup (Global Preserve) array parameter (*9)	
SetVariable	Parameter name,value{, type}	Set the value in the backup (Global Preserve) parameter (*8)	Auto ON Ready ON
GetStatus		Acquire the Controller state	Available any time (*1)
Execute	Command Working with strings	Execute the command (*10)(*11) (*12)	Auto ON Ready ON Error OFF EStop OFF Safeguard OFF
Abort		Abort the command execution	Auto ON
GetAlm		Acquire the alarm state (*13)	Available any time (*2)
ResetAlm	Alarm number	Reset the alarm of the specified alarm number (*13)	Auto ON Ready ON

*1: The Controller state bit from GetStatus.

*2: "Available any time" is applicable if the following conditions are met.

- When "Remote Ethernet" is set as the control device, or,
- "Remote Ethernet" is not set as the control device, but set to be used for monitoring.

*3: Execute the function specified in the Main[Function No.].

Function Name	Function No.
Main	0
Main1	1
Main2	2
Main3	3
Main4	4

Function Name	Function No.
Main5	5
Main6	6
Main7	7
⋮	⋮
Main63	63

*4: Pause command is not available for "NoPause task" and "NoEmgAbort task".

For details, refer to Help or the following manual:

"SPEL+ Language Reference- Pause""

*5: I/O output will be turned off and the robot parameter will be initialized.

For details, refer to Help or the following manual:

"SPEL+ Language Reference- Reset""

*6: The robot parameter will be initialized.

For details, refer to Help or the following manual:

""SPEL+ Language Reference- Motor""

*7: When 0 is specified for the manipulator number, all the manipulator will be operated

If you wish to operate particular manipulator, specify the manipulator number (1 to 16) of the target manipulators.

*8: Parameter type means {Boolean | Byte | Double | Integer | Long | Real | String | Short | UByte | UShort | Int32 | UInt32 | Int64 | Specify UInt64}

Type specified: for the backup parameters when the parameter name and type are same.

Type not specified: for the backup parameters when the parameter names are same.

*9: For the array element, specify an element you acquire as the following:

You need to specify an element when acquiring from the head of the array.

1D array	Parameter name (0)	Acquire from the head.
	Parameter name (Element number)	Acquire from the specified element number.
2D array	Parameter name (0.0)	Acquire from the head.
	Parameter name (Element number 1, 2)	Acquire from the specified element number.
3D array	Parameter name (0,0,0)	Acquire from the head.
	Parameter name (Element number 1, 2, 3)	Acquire from the specified element number.

- You cannot omit the parameter type and number to acquire.
- You cannot specify a string for the parameter type.
- Available number to acquire is up to 100.
- If you specify a number over the number of array elements, you have an error.

For example:

```
$GetVariable,gby2(3,0),Byte,3
```

It acquires values of gby2(3,0), gby2(3,1), gby2(3,2) of Byte type 2D array parameter gby2.

*10: Specify the command and parameters in double quotation marks (" ").

Command strings to be executed are restricted to 256 bytes, and execution result strings are restricted to 4060 bytes.

Robot motion command will be executed to the selected manipulator. Check which robot is selected by using GetCurRobot before command execution.

Following commands are available while Execute is running.

Commands available while Execute is running

Remote command

- Abort
- GetStatus
- SetIO
- SetIOByte
- SetIOWord
- SetMemIO
- SetMemIOByte
- SetMemIOWord

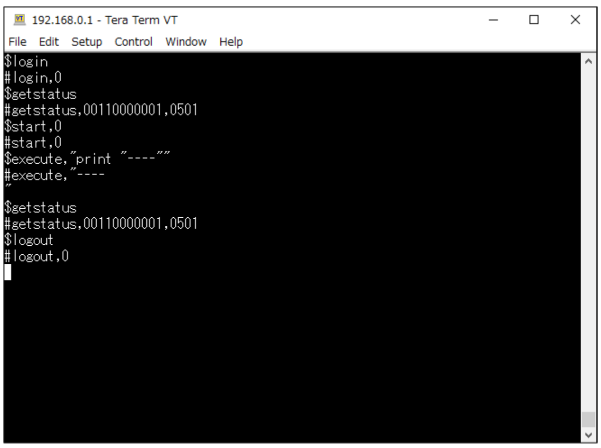
When the commands specified in (SetIO, SetIOByte, SetIOWord, SetMemIO, SetMemIOByte, SetMemIOWord) are the same and executed at the same time, the command executed later will result in error. Make sure to check the execution result by using GetStatus after the execution of Execute command and output command which the Execute command is being executed.

*11: To execute commands including PC function (PC file, PC RS-232C, Database access, DLL calling), be sure to execute while the Epson RC+ 8.0 is connected. If the Epson RC+ 8.0 is not connected, command execution will result in error.

*12: "Command string" parameter of Execute command encloses in (" ") (double quotation marks). If the parameter includes (" "), refer to the following execution example. In SPEL+ language, Chr$(34) is used for (" ") (double quotation marks). Refer to #Print in SPEL+ Language Reference.

Example: TeraTerm

((New-line Receive: AUTO, Transmit: CR+LF, Local echo: ON)



*13: For details on alarms, refer to the following manuals:

- For RC700-D, RC700-E:"Robot Controller Manual - Alarm"
- For RC700, RC90 series:"Robot Controller Maintenance Manual - Alarm"
- For T, VT series:"Robot Maintenance Manual - Alarm"

13.2.5.2 For RC800 series:

Remote command	Parameter	Contents	Input acceptance condition (*1)
Login	Password	Start the Controller Remote Ethernet function Authentication by password Execute Login correctly, commands execution is enabled until Logout	Available any time (*2)
Logout		Exit Controller Remote Ethernet function After Logout, execute the Login command to start remote Ethernet function. Logout during task execution causes an error.	Available any time (*2)
Start	Function No.	Execute the function of specified number (*3)(*11)	Auto ON Ready ON Error OFF EStop OFF Safeguard OFF
Stop		Stop all tasks and commands	Auto ON
Pause		All tasks are paused. (*4)	Auto ON Running ON
Continue		Continue paused tasks	Auto ON Paused ON
ContinueManualRestart		Continues the paused task. (Automatically recovers)	Auto ON Paused ON
Reset		Resets emergency stop and error. (*5)	Auto ON Ready ON
SetMotorsOn	Robot number	Turn ON the robot motor (*6)(*7)	Auto ON Ready ON EStop OFF Safeguard OFF
SetMotorsOff	Robot number	Turn OFF the robot motor (*7)	Auto ON Ready ON
SetCurRobot	Robot number	Select the manipulator	Auto ON Ready ON

Remote command	Parameter	Contents	Input acceptance condition (*1)
GetCurRobot		Acquires the current manipulator number	Available any time (*2)
Home	Robot number	Move the arm to home position defined by user	Auto ON Ready ON Error OFF EStop OFF Safeguard OFF
GetIO	I/O bit No.	Acquire the specified I/O bit	Available any time (*2)
SetIO	I/O bit No. & value	Set the I/O specified bit 1: Turn ON the bit 0: Turn OFF the bit	Ready ON
GetIOByte	I/O port No.	Acquire the specified I/O port (8 bit)	Available any time (*1)
SetIOByte	I/O port No. & value	Set the I/O specified port (8 bit)	Ready ON
GetIOWord	I/O word port No.	Acquire the specified I/O word port (16 bit)	Available any time (*2)
SetIOWord	I/O word port No. & value	Set the I/O specified word port (8 bit)	Auto ON Ready ON
GetMemIO	Memory I/O bit No.	Acquire the specified memory I/O bit	Available any time (*2)
SetMemIO	Memory I/O bit No., value	Set the specified memory I/O bit 1: Turn ON the bit 0: Turn OFF the bit	Auto ON Ready ON
GetMemIOByte	Memory I/O Port number	Acquire the specified memory I/O port	Available any time (*2)
SetMemIOByte	Memory I/O port No., value	Set the specified memory I/O port (8 bit)	Auto ON Ready ON
GetMemIOWord	Memory I/O word port No.	Acquire the specified memory I/O word port (16 bit)	Available any time (*2)
SetMemIOWord	Memory I/O word port No., value	Set the specified memory I/O word port (16 bit)	Auto ON Ready ON
GetVariable	Parameter name{,type}	Acquire the value of backup (Global Preserve) parameter (*8)	Available any time (*2)
	[Parameter name] (Array element), [Parameter name type], [Number to acquire]	Acquire the value of backup (Global Preserve) array parameter (*9)	
SetVariable	Parameter name,value{, type}	Set the value in the backup (Global Preserve) parameter (*8)	Auto ON Ready ON

Remote command	Parameter	Contents	Input acceptance condition (*1)
GetStatus		Acquire the Controller state	Available any time (*1)
Execute	Command Working with strings	Execute the command (*10)(*11) (*12)	Auto ON Ready ON Error OFF EStop OFF Safeguard OFF
Abort		Abort the command execution	Auto ON
ResetCtrlParts	Parts number	Maintenance (Controller parts reset)	Available any time (*1)
ResetRbParts	Robot number,parts number	Maintenance (robot parts reset)	Available any time (*1)

*1: The Controller state bit from GetStatus.

*2: "Available any time" is applicable if the following conditions are met.

- When "Remote Ethernet" is set as the control device, or,
- "Remote Ethernet" is not set as the control device, but set to be used for monitoring.

*3: Execute the function specified in the Main[Function No.].

Function Name	Function No.
Main	0
Main1	1
Main2	2
Main3	3
Main4	4
Main5	5
Main6	6
Main7	7
⋮	⋮
Main63	63

*4: Pause command is not available for "NoPause task" and "NoEmgAbort task".

For details, refer to Help or the following manual:

"SPEL+ Language Reference- Pause""

*5: I/O output will be turned off and the robot parameter will be initialized.

For details, refer to Help or the following manual:

"SPEL+ Language Reference- Reset""

*6: The robot parameter will be initialized.

For details, refer to Help or the following manual:

""SPEL+ Language Reference- Motor""

*7: When 0 is specified for the manipulator number, all the manipulator will be operated

If you wish to operate particular manipulator, specify the manipulator number (1 to 16) of the target manipulators.

*8: Parameter type means {Boolean | Byte | Double | Integer | Long | Real | String | Short | UByte | UShort | Int32 | UInt32 | Int64 | Specify UInt64}

Type specified: for the backup parameters when the parameter name and type are same.

Type not specified: for the backup parameters when the parameter names are same.

*9: For the array element, specify an element you acquire as the following:

You need to specify an element when acquiring from the head of the array.

1D array	Parameter name (0)	Acquire from the head.
	Parameter name (Element number)	Acquire from the specified element number.
2D array	Parameter name (0.0)	Acquire from the head.
	Parameter name (Element number 1, 2)	Acquire from the specified element number.
3D array	Parameter name (0,0,0)	Acquire from the head.
	Parameter name (Element number 1, 2, 3)	Acquire from the specified element number.

- You cannot omit the parameter type and number to acquire.
- You cannot specify a string for the parameter type.
- Available number to acquire is up to 100.
- If you specify a number over the number of array elements, you have an error.

For example:

```
$GetVariable,gby2(3,0),Byte,3
```

It acquires values of gby2(3,0), gby2(3,1), gby2(3,2) of Byte type 2D array parameter gby2.

*10: Specify the command and parameters in double quotation marks (" ").

Command strings to be executed are restricted to 256 bytes, and execution result strings are restricted to 4060 bytes.

Robot motion command will be executed to the selected manipulator. Check which robot is selected by using GetCurRobot before command execution.

Following commands are available while Execute is running.

Commands available while Execute is running

Remote command

- Abort
- GetStatus

- SetIO
- SetIOByte
- SetIOWord
- SetMemIO
- SetMemIOByte
- SetMemIOWord

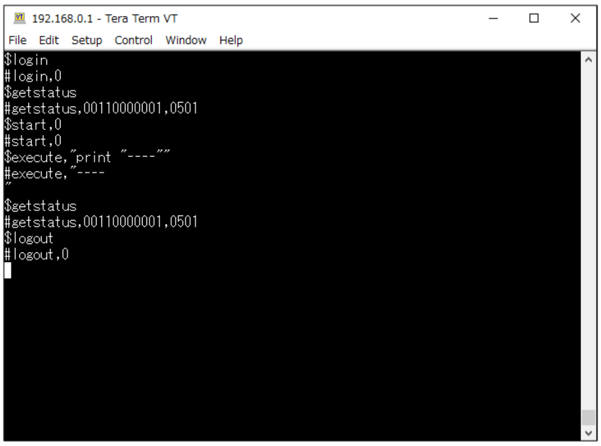
When the commands specified in (SetIO, SetIOByte, SetIOWord, SetMemIO, SetMemIOByte, SetMemIOWord) are the same and executed at the same time, the command executed later will result in error. Make sure to check the execution result by using GetStatus after the execution of Execute command and output command which the Execute command is being executed.

*11: To execute commands including PC function (PC file, PC RS-232C, Database access, DLL calling), be sure to execute while the Epson RC+ 8.0 is connected. If the Epson RC+ 8.0 is not connected, command execution will result in error.

*12: "Command string" parameter of Execute command encloses in (" ") (double quotation marks). If the parameter includes (" "), refer to the following execution example. In SPEL+ language, Chr$(34) is used for (" ") (double quotation marks). Refer to #Print in SPEL+ Language Reference.

Example: TeraTerm

(New-line Receive: AUTO, Transmit: CR+LF, Local echo: ON)



13.2.6 Monitoring command

When the Remote Ethernet control is not set as the control device but set to be used for monitoring, following commands are only available to be executed.

Remote command

- Login
- Logout
- GetIO
- GetIOByte
- GetIOWord
- GetMemIO
- GetMemIOByte
- GetMemIOWord
- GetVariable

- GetStatus
- GetCurRobot
- GetAlm

13.2.7 Response

When the Controller receives the command correctly, the response in the following format is shown in the executing command.

Command	Format
Remote command that acquire the value except the following commands	#[Remote command],[0] terminator
GetCurRobot	#GetCurRobot,[Robot number] Terminator
GetIO	#GetIO,[0 | 1] terminator *1
GetMemIO	#GetMemIO,[0 | 1] terminator *1
GetIOByte	#GetIOByte,[Hex string (00 to FF) of Byte (8Bit)] terminator
GetMemIOByte	#GetMemIOByte,[Hex string (00 to FF) of Byte (8Bit)] terminator
GetIOWord	#GetIOWord,[Hex string (0000 to FFFF) of Word (16Bit)] terminator
GetIOMemWord	#GetMemIOWord,[Hex string (0000 to FFFF) of Word (16Bit)] terminator
GetVariable	#GetVariable,[Parameter value] terminator
GetVariable (in case of array)	#GetVariable,[Parameter value 1],[Parameter value 2],...,terminator *4
GetStatus	#GetStatus,[Status],[Error, Warning code] terminator Example) # GetStatus,aaaaaaaaaaa,bbbb *2 *3
Execute	If the value is returned as a result of command execution # Execute,"[Execution result]" terminator
GetAlm	#GetAlm,[number of alarms],[alarm number]..terminator e.g) When no alarm is occurring # GetAlm,0 terminator e.g.) When Alarm 1 and 9 are occurring # GetAlm,2,1,9 terminator

*1 [0 | 1] I/O bit ON: 1/ OFF: 0

*2 aaaaaaaaaaa: Status

In the example above, 11 digits "aaaaaaaaaaa" is for the following 11 flags.

Test, Teach, Auto, Warning, SError, Safeguard, Estop, Error, Paused, Running, Ready

ON: 1/ OFF: 0

If Ready and Auto are ON, it is "00100000001".

*3 bbbb part: Error / Warning code

It is indicated in 4 digits. If there is no error and warning, it is 0000.

e.g.)1: # GetStatus,0100000001,0000

The bits for Auto and Ready are ON (1).

This means that AutoMode is ON and in Ready state. You can execute the command.

e.g.)2#: GetStatus,00110000010,0517

This means the warning occurs during the operation. Take appropriate action for the warning code. (In this case, warning code is 0517)

Flag	Contents
Test	Turn ON in the TEST mode
Teach	Turn ON in the TEACH mode
Auto	Turn ON in the remote input acceptance condition
Warning	Turn ON in the warning condition Task can be executed as usual even a warning condition. However, take action for the warning as soon as possible.
SError	Turn ON in serious error condition. When a serious error occurs, "Reset input" is not available for recovery. Reboot the controller to recover.
Safeguard	Turn ON with safety door open
EStop	Turn ON in the emergency condition
Error	Turn ON in the error condition. Use "Reset input" to recover from the error condition.
Paused	Turn ON with paused task
Running	Turn ON with task executing Turn OFF when "Paused output" is ON
Ready	Turn ON with the Controller completed the startup and no task executing

*4 It returns values of specified number in the Number to acquire.

Error response

When the Controller cannot receive the remote command correctly, the error response is shown in the following format.

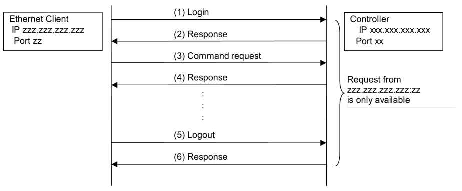
Format: ![Remote command],[Error code] terminator

Error code	Contents
10	Remote command does not begin with $
11	Remote command is wrong Login is not executed
12	Remote command format is wrong
13	Login command password is wrong
14	Specified number to acquire is out of range (1 or more and 100 or less) Number to acquire is omitted Specified a string parameter

Error code	Contents
15	Parameter is not existed Dimension of parameter is wrong Element out of range is called
19	Request time out
20	Controller is not ready
21	Cannot execute since the Execute is running
98	Password is required for Login when using the global IP address.
99	System error Communication error

13.2.8 Response timing of Remote Ethernet control

Communication sequence



13.3 Remote RS232

Remote RS232 makes it possible to control the robot and controller from external equipment by sending the remote commands through RS-232C.

KEY POINTS

RC800-A does not support remote RS232.

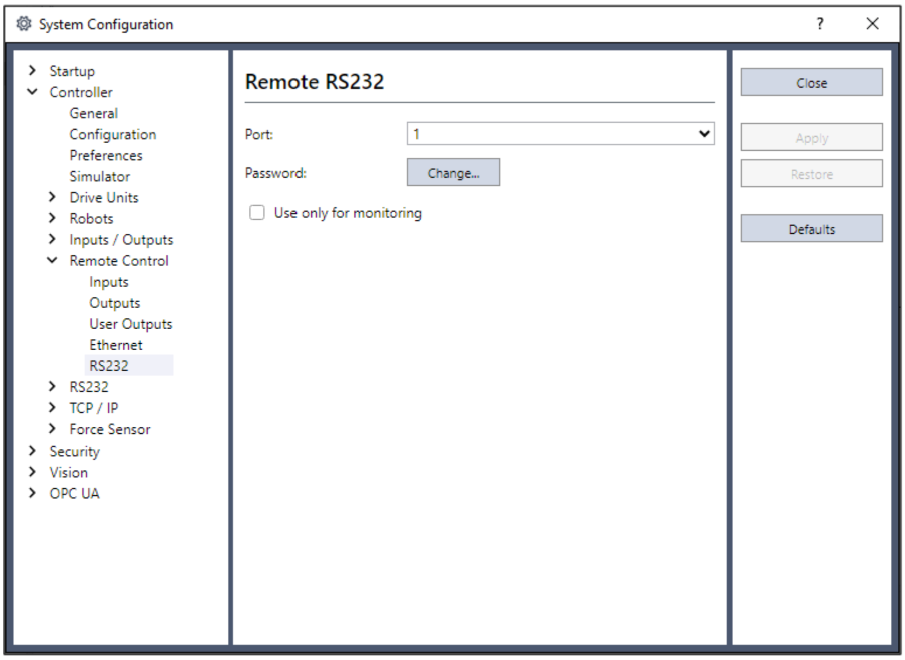
13.3.1 Remote RS232 setting

To set the remote RS232 functions valid, follow the procedures below to configure the remote RS232.

1. The [System Configuration] command opens a dialog that contains several pages that are used to configure the system for the Epson RC+ 8.0 environment. Select [Controller]-[Remote]-[RS232].

2. Configure the necessary items for the remote RS232 control.

3. Click the [Apply] button to save the new setting, then click the [Close] button.



For details on the dialog setting, see below.

[Setup]-[System Configuration]-[Controller]-[Remote Control]

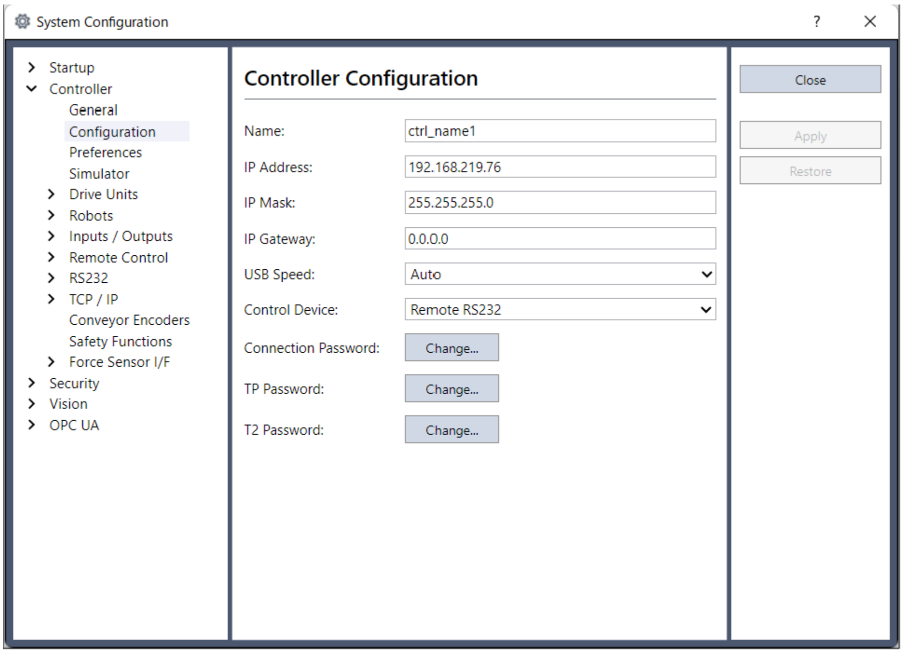
If you select the "Use only for monitoring" checkbox and only acquire values using Remote RS232 control, the control device setting in the next section will not be necessary.

13.3.2 Control device setting

Set the control device to "Remote RS232" by the following procedure.

1. The [System Configuration] command opens a dialog that contains several pages that are used to configure the system for the Epson RC+ 8.0 environment. Select [Controller]-[Configuration]. Select "Remote RS232" in the [Control Device] box.

2. Click the [Apply] button to save the new setting, then click the [Close] button.



For details on the dialog setting, see below.

[Setup]-[System Configuration]-[Controller]-[Configuration] Page

13.3.3 Execution of remote RS232 control

Set the remote RS232 control available by the following procedure.

1. Open RS-232C port that is connected from client equipment to the specified port in the Remote RS232 of the Controller, using the communication parameter specified in the RS-232C port setting.
2. Send the remote start command (EOT).
3. Specify the password set in the Remote RS232 to the parameter and send the Login command.
4. Client equipment has to wait until Auto (GetStatus command response) is ON, before execution of remote command.
5. Now remote command will be accepted. Each command executes the function when the input acceptance condition is satisfied.

13.3.4 Debugging remote RS232 control

Program debug from Epson RC+ 8.0 development environment is capable as follows.

- Create a program in the same manner as usual.

Open the Run window and click the [Enable RS232] button.
When you only acquire values using the Remote RS232 control, the [Enable RS232] button is not displayed. Click the button of the device specified as the control device.

Now the remote commands will be accepted.

Breakpoint setting and output to the Run window is available.

KEY POINTS

After Login, if no command is sent within the RS-232C's timeout duration, a timeout error will occur. In this case, re-execute from sending remote start command.

If error occurs, execute the Reset command to clear the error condition before executing the operation command. To clear the error condition from external equipment by monitoring, use the "GetStatus" and "Reset" command.

CAUTION

If you set "0" in the [Timeout] box, time out duration is infinite. In this case, the task continues to execute even without the communication from client. This means the robot may keep moving and cause unexpected damage. Ensure the ways other than the communication to stop the task.

13.3.5 Remote RS232 Command

Remote start

Start the Remote RS232 function of the Controller.

EOT 1 Byte

EOT : &H04(&H is hexadecimal)

Request format

STX 1 Byte	Command 1 Byte	Data Variable	ETX 1 Byte	BCC 1 Byte

- STX: &H02
- ETX: &H03
- BCC : Checksum of sent and received data

  XOR value from the command to ETX per 1Byte

Remote command	Send command	Data	Description	Input acceptance condition (*1)
Login	'L' &H4C	Password	Authentication by password Execute Login correctly, commands execution is enabled until Logout	Available any time (*2)
Logout	'l' &H6C		After Logout, execute the Login command to start remote RS232 function. Logout during task execution causes an error.	Available any time (*2)
Start	'G' &H47	Function No. (1 Byte)	Execute the function of specified number (*3)(*11) Example) Execute 'main' &H02&H47&H00&H03&H44	Auto ON Ready ON Error OFF EStop OFF Safeguard OFF
Stop	'Q' &H51		Stop all tasks and commands	Auto ON
Pause	'P' &H50		All tasks are paused. (*4)	Auto ON Running ON
Continue	'C' &H43		Continue paused tasks	Auto ON Paused ON
Reset	'R' &H52		Resets emergency stop and error. (*5)	Auto ON Ready ON
SetMotorsOn	'M' &H4D	Robot number (1 Byte)	Turn ON the robot motor (*6)(*7)	Auto ON Ready ON EStop OFF Safeguard OFF

Remote command	Send command	Data	Description	Input acceptance condition (*1)
SetMotorsOff	'N' &H4E	Robot number (1 Byte)	Turn OFF the robot motor (*7)	Auto ON Ready ON
SetCurRobot	'Y' &H59	Robot number (1 Byte)	Select the robot	Auto ON Ready ON
GetCurRobot	'y' &H79		Acquire the current robot number	Available any time (*2)
Home	'H' &H48	Robot number (1 Byte)	Move the arm to home position defined by user	Auto ON Ready ON Error OFF EStop OFF Safeguard OFF
GetIO	'i' &H69	I/O bit No (2 Byte)	Acquire the specified I/O bit Example) Acquire the I/O bit 1 &H02&H69&H0001&H03&H6B	Available any time (*2)
SetIO	'I' &H49	[I/O bit No.] (2Byte) [value] (1Byte)	Set the I/O specified bit &H01: Turn ON the bit &H00: Turn OFF the bit Example) Turn ON the I/O bit 1 &H02&H49&H0001&H01&H03&H4A	Auto ON Ready ON
GetIOByte	'b' &H62	I/O port No. (1 Byte)	Acquire the specified I/O port (8 bit) (*8) Example) Acquire the I/O port 1 &H02&H62&H01&H03&H60	Available any time (*2)
SetIOByte	'B' &H42	[I/O port No.] (1Byte) [value] (1Byte)	Set the I/O specified port (8 bit) (*8) Example) Set &H0F to the I/O port 1 &H02&H42&H01&H0F&H03&H4F	Auto ON Ready ON
GetIOWord	'w' &H77	I/O Word Port number (1 Byte)	Acquire the specified I/O word port (16 bit) (*8) Example) Acquire the I/O word port 1 &H02&H77&H01&H03&H75	Available any time (*2)
SetIOWord	'W' &H57	[I/O Word Port number] (1Byte) [value] (2 Byte)	Set the I/O specified word port (16 bit) (*8) Example) Set &H010F to the I/O word port 1 &H02&H57&H01&H010F&H03&H5B	Auto ON Ready ON

Remote command	Send command	Data	Description	Input acceptance condition (*1)
GetMemIO	'o' &H6F	Memory I/O bit No. (2 Byte)	Acquire the specified memory I/O bit (*8) Example) Acquire memory I/O bit 1 &H02&H6F&H0001&H03&H6D	Available any time (*2)
SetMemIO	'O' &H4F	[Memory I/O bit No.] (2Byte) [value] (1 Byte)	Set the I/O specified bit (*8) &H01: Turn ON the bit &H00: Turn OFF the bit Example) Turn ON the memory I/O bit 1 &H02&H4F&H0001&H01&H03&H4C	Auto ON Ready ON
GetMemIOByte	't' &H74	Memory I/O Port number (1 Byte)	Acquire the specified memory I/O port (8 bit) (*8) Example) Acquire the memory I/O port 1 &H02&H74&H01&H03&H76	Available any time (*2)
SetMemIOByte	'T' &H54	[Memory I/O port number] (1Byte) [value] (1 Byte)	Set the I/O specified port (8 bit) (*8) Example) Set &H0F to the memory I/O port 1 &H02&H54&H01&H0F&H03&H59	Auto ON Ready ON
GetMemIOWord	'u' &H75	Memory I/O word port (1Byte)	Acquire the specified memory I/O word port (16 bit) (*8) Example) Acquire the memory I/O word port 1 &H02&H75&H01&H03&H77	Available any time (*2)
SetMemIOWord	'U' &H55	[Memory I/O word port No. (1Byte) [value] (1Byte)	Set the I/O specified word port (16 bit) (*8) Example) Set &H010F to the memory I/O word port 1 &H02&H55&H01&H010F&H03&H59	Auto ON Ready ON
GetVariable	'v' &H76	[Parameter name], (&H2C) [type] (1 Byte)	Acquire the value of backup (Global Preserve) parameter (*8) Example) Acquire the Global Integer g_Status &H02&H76&H67&H5F&H53&H74&H61&H74&H75 &H73&H2C&H03&H03&H56	Available any time (*2)

Remote command	Send command	Data	Description	Input acceptance condition (*1)
		[Parameter name], (&H2C) (Array element) (&H2C) [Parameter type] (1 Byte), (&H2C) [Number to acquire] (2 Byte)	Acquire the value of backup (Global Preserve) array parameter (*9) Example) Acquire all of Global Integer g_intArray(10) &H02&H76&H67&H5F&H69&H6E&H74&H41&H72&H72&H61&H79&H2C&H0000&H2C&H03&H2C&H000A&H03&H42 Example) Acquire 10 elements from elements (3,5,0) of Global Integer g_int3Array(10,10,10) &H02&H76&H67&H5F&H69&H6E&H74&H33&H41&H72&H72&H61&H79&H2C&H0003&H2C&H0005&H2C&H0000&H2C&H03&H2C&H000A&H03&H77	
SetVariable	'V' &H56	[Parameter name], (&H2C) [value] (type size) (&H2C), [type] (1 Byte)	Set the value in the backup (Global Preserve) parameter (*8) (Example) Set 0 (&H0000) to Global Integer g_Status &H02&H56&H67&H5F&H53&H74&H61&H74&H75&H73&H2C&H0000&H2C&H03&H03&H5A	Auto ON Ready ON
GetStatus	'S' &H53		Acquire the Controller state	Available any time (*2)
Execute	'X' &H58	Command Working with strings	Execute the command (*10) (*11) Example) Execute 'print here' &H02&H58&H22&H70&H72&H69&H6E&H74&H20&H68&H65&H72&H65&H22&H03&H10	Auto ON Ready ON Error OFF EStop OFF Safeguard OFF
Abort	'A' &H41		Abort the command execution (*10)	Auto ON
GetAlm	'z' &H7A		Acquire the alarm state	Available any time (*2)
ResetAlm	'Z' &H5A	Alarm number (1 Byte)	Reset the alarm of the specified alarm number e.g.) When resetting the Alarm 5 &H02&H5A&H05&H03&H5C	Auto ON Ready ON

*1: The Controller state bit from GetStatus.

*2: "Available any time" is applicable if the following conditions are met.

- When "Remote Ethernet" is set as the control device, or,
- "Remote Ethernet" is not set as the control device, but set to be used for monitoring.

*3: Execute the function specified in the Main[Function No.].

Function Name	Function No.
Main	0
Main1	1
Main2	2
Main3	3
Main4	4
Main5	5
Main6	6
Main7	7
⋮	⋮
Main63	63

*4: Pause command is not available for "NoPause task" and "NoEmgAbort task".

For details, refer to Help or the following manual:

"SPEL+ Language Reference- Pause""

*5: I/O output will be turned off and the robot parameter will be initialized.

For details, refer to Help or the following manual:

"SPEL+ Language Reference- Reset""

*6: The robot parameter will be initialized.

For details, refer to Help or the following manual:

""SPEL+ Language Reference- Motor""

*7: When "0" is specified for the manipulator number, all the manipulator will be operated

If you wish to operate particular manipulator, specify the manipulator number (1 to 16) of the target manipulators.

*8: Parameter type

Parameter type	Type value(1Byte)
Boolean	&H00
Byte	&H01
Double	&H02
Integer	&H03
Long	&H04
Real	&H05
String	&H06

Parameter type	Type value(1Byte)
UByte	&H07
Short	&H08
UShort	&H09
Int32	&H0A
UInt32	&H0B
Int64	&H0C
UInt64	&H0D

For the backup parameters when the parameter name and type are same.

*9: For the array element, specify an element you acquire as the following:

You need to specify an element when acquiring from the head of the array.

Specify the array element in 2Byte value.

1D array	Parameter name&H2C&H0000	Acquire from the head.
	Parameter name, element number.	Acquire from the specified element number.
2D array	Parameter name &H2C&H0000&H2C&H0000	Acquire from the head.
	Parameter name, element number 1, element number 2	Acquire from the specified element number.
3D array	Parameter name &H2C&H0000&H2C&H0000&H2C&H0000	Acquire from the head.
	Parameter name, element number 1, element number 2, element number 3	Acquire from the specified element number.

You cannot specify a string for the parameter type.

Available number to acquire is up to 100.

If you specify a number over the number of array elements, you have an error.

*10: Specify the command and parameters in double quotation marks (" ").

Command strings to be executed are restricted to 256 bytes, and execution result strings are restricted to 4060 bytes.

Robot motion command will be executed to the selected manipulator. Check which robot is selected by using GetCurRobot before command execution.

Following commands are available while Execute is running.

Commands available while Execute is running

Remote command

- Abort
- GetStatus
- SetIO
- SetIOByte

- SetIOWord
- SetMemIO
- SetMemIOByte
- SetMemIOWord

When the commands specified in (SetIO, SetIOByte, SetIOWord, SetMemIO, SetMemIOByte, SetMemIOWord) are the same and executed at the same time, the command executed later will result in error. Make sure to check the execution result by using GetStatus after the execution of Execute command and output command which ** the Execute command is being executed.

*11: To execute commands including PC function (PC file, PC RS-232C, Database access, DLL calling), be sure to execute while the Epson RC+ 8.0 is connected. If the Epson RC+ 8.0 is not connected, command execution will result in error.

13.3.6 Monitoring command

When the remote RS232 control is not set as the control device but set to be used for monitoring, following commands are only available to be executed.

Remote command

- Login
- Logout
- GetIO
- GetIOByte
- GetIOWord
- GetMemIO
- GetMemIOByte
- GetMemIOWord
- GetVariable
- GetStatus
- GetCurRobot
- GetAlm

13.3.7 Response

When the Controller receives the command correctly, the response in the following format is shown in the executing command.

Response Format

ACK 1 Byte	Command 1Byte	Data Variable	ETX 1 Byte	BCC 1 Byte

- ACK : &H06
- ETX: &H03
- BCC : Checksum of sent and received data

  XOR value from the command to ETX per 1Byte

Command	Format
Remote command that acquire the value except the following commands	[ACK][Command] (1Byte)[ETX][BCC]
GetCurRobot	[ACK]'y'[Robot number] [ETX][BCC]
GetIO	[ACK] 'i'[&H00 | &H01] [ETX][BCC] *1
GetMemIO	[ACK] 'o'[&H00 | &H01] [ETX][BCC] *1
GetIOByte	[ACK]'b'[Byte value (8Bit) (&H00 to &HFF)] [ETX][BCC]
GetMemIOByte	[ACK]'t'[Byte value (8Bit) (&H00 to &HFF)] [ETX][BCC]
GetIOWord	[ACK]'w'[Word value (16Bit) (&H0000 to &HFFFF)] [ETX][BCC]
GetIOMemWord	[ACK]'u'[Word value (16Bit) (&H0000 to &HFFFF)] [ETX][BCC]
GetVariable	[ACK]'v'[Parameter value]*5 [ETX][BCC]
GetVariable (in case of array)	[ACK]'v'[Parameter value 1]*5[Parameter value 2]*5 ...[ETX][BCC] *4
GetStatus	[ACK]'S'[Status][Error, Warning code] [ETX][BCC] e.g: [ACK] 'S'[aaaaaaaaaaa][bbbb][ETX][BCC] *2 *3
Execute	If the value is returned as a result of command execution [ACK]'X'"[Execution result]" [ETX][BCC]
GetAlm	[ACK]'z' [number of alarms][alarm number]... [ETX][BCC] e.g) When no alarm is occurring &H06&H7A&H00&H03&H79 e.g.) When Alarm 1 and 9 are occurring &H06&H7A&H02&H01&H09&H03&H73

*1 [&H00 | &H01] I/O bit ON: &H01/ OFF: &H00

*2 aaaaaaaaaaa: Status

In the example above, 11 digits [aaaaaaaaaaa] is for the following 11 flags.

Test, Teach, Auto, Warning, SError, Safeguard, Estop, Error, Paused, Running, Ready

&H01 is ON /&H00 is OFF

If Ready and Auto are ON, it is

[&H00&H00&H01&H00&H00&H00&H00&H00&H00&H00&H01].

*3 bbbb part: Error / Warning code

It is indicated in 4 digits. If there is no error and warning, it is "0000"(&H30&H30&H30&h30).

Example1: [ACK]

'S[&H00&H00&H01&H00&H00&H00&H00&H00&H00&H00&H01][&H30&H30&H30&h30]

The bits for Auto and Ready are &H01.

This means that AutoMode is ON and in Ready state. You can execute the command.

Example2: [ACK]

'S[&H00&H00&H01&H01&H00&H00&H00&H00&H00&H01&H00][&H30&H35&H31&h37]

This means the warning occurs during the operation. Take appropriate action according to the warning code. (In this case, warning code is 0517)

Flag	Contents
Test	Turn ON in the TEST mode
Teach	Turn ON in the TEACH mode
Auto	Turn ON in the remote input acceptance condition
Warning	Turn ON in the warning condition Task can be executed as usual even a warning condition. However, take action for the warning as soon as possible.
SError	Turns ON when critical error occurs. When a critical error occurs, "Reset input" does not function. Reboot the controller to recover.
Safeguard	Turn ON with safety door open
EStop	Turn ON in the emergency condition
Error	Turns ON when an error occurs. Use "Reset input" to recover from the error.
Paused	Turn ON with paused task
Running	Turn ON with task executing Turn OFF when "Paused output" is ON
Ready	Turn ON with the Controller completed the startup and no task executing

*4 It returns values of specified number in the Number to acquire.

*5 Binary data. If you convert it to the specified data type after acquisition, conversion process is required.

Error response

When the Controller cannot receive the remote command correctly, the error response is shown in the following format.

NAK 1 Byte	Command 1 Byte	Error code 2 Byte	ETX 1 Byte	BCC 1 Byte

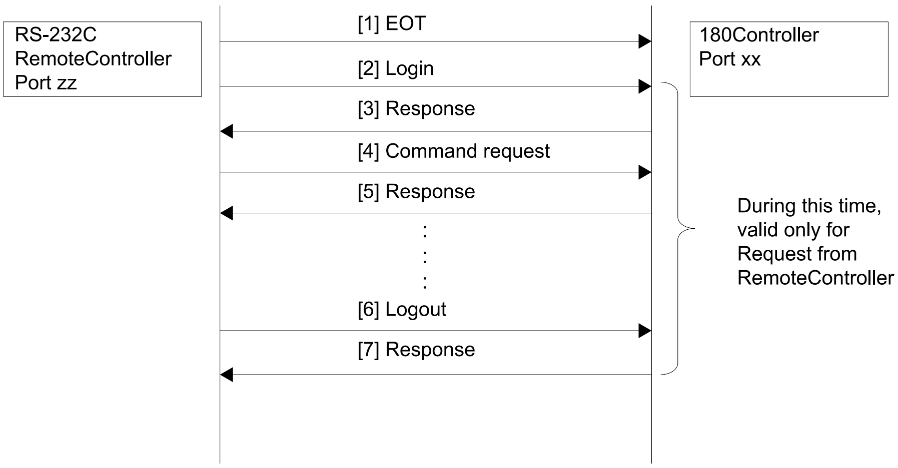
- NAK : &H15
- ETX : &H03
- BCC : Checksum of sent and received data

  XOR value from the command to ETX per 1Byte

Error code	Contents
10	Remote command does not begin with $

Error code	Contents
11	Remote command is wrong Login is not executed
12	Remote command format is wrong
13	Login command password is wrong
14	Specified number to acquire is out of range (1 or more and 100 or less) Number to acquire is omitted Specified a string parameter
15	Parameter is not existed Dimension of parameter is wrong Element out of range is called
16	BCC is wrong
19	Request time out
20	Controller is not ready
21	Cannot execute since the Execute is running
99	System error , Communication error

13.3.8 Response timing of Remote Ethernet control



13.4 User-defined Remote Output I/O

13.4.1 What is user-defined remote output I/O?

User-defined remote output I/O is the remote output I/O that the user arbitrarily sets the output conditions.

Output to the I/O without creating dedicated tasks to the user program is possible.

- 8 user-defined remote output I/O are available.
- Output condition is specified by the condition expression of the SPEL language.
- Evaluation of output conditions are performed in 30 ms periods.
- Output method when condition is met can be selected from Level, Pulse, and Latch.

- Output polarity (active low/ active high) when condition is met can be selected.

13.4.2 Output conditions

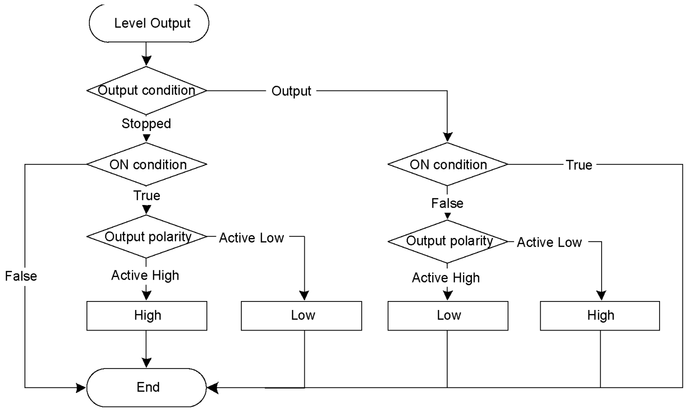
Output conditions consist of the ON and OFF conditions. The OFF condition is set only when the output method is "Latch".

- [ON condition] Set the condition expression to start output.
- [OFF condition] Set the condition expression to terminate the latch output.

13.4.3 Output

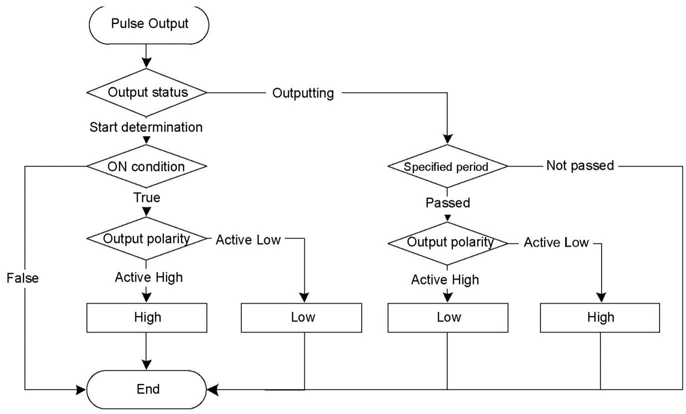
Level output

It outputs with the selected polarity while the ON condition is met. Output terminates when the ON condition is not met.



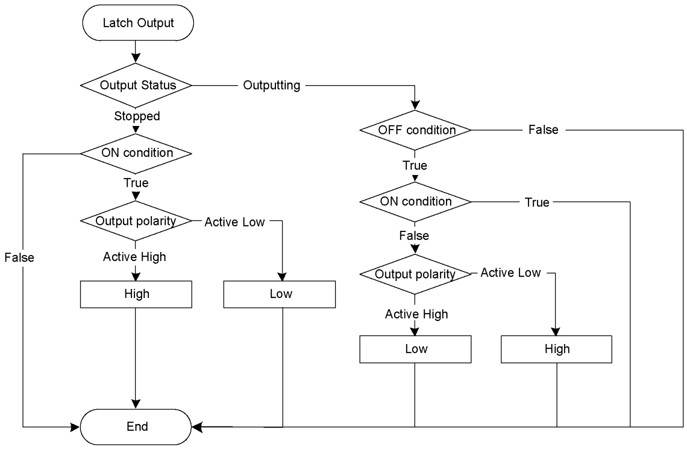
Pulse output

It outputs with the selected polarity for the specified time (10 ms unit) after the ON condition is met. Output terminates when the specified time elapses.



Latch Output

Output starts with the selected polarity when ON condition is met. Output terminates when OFF condition is met and ON condition is not met at the same time.



13.4.4 Restrictions

Condition expressions of the SPEL language are used to specify conditions. However, they have following restrictions.

- Variables cannot be used.
- Labels cannot be used.
- Available functions are limited.

Available Functions

- A
  - Abs, Acos, Agl, And, Arm, ArmDef, Asc, Asin, Atan, Atan2
- B
  - BClr, BClr64, BoxDef, BSet, BSet64, BTst, BTst64
- C
  - Cos, CR, CS, CT, CtrlDev, CtrlInfo, CU, CurPos, CV, CW, CX, CY, CZ
- D
  - DegToRad, DispDev, Dist
- E
  - ECP, ECPDef, ECPSet, ElapsedTime, Era, Errb, ErrorOn, Ert, EStopOn
- F
  - Fine, Fix
- G
  - GetRobotInsideBox, GetRobotInsidePlane
- H
  - Hand, Hofs, HomeDef, Hour
- I
  - In, InBCD, Inertia, InPos, InReal, InsideBox, InsidePlane, InW
- J
  - J1Angle, J1Flag, J4Flag, J6Flag, JRange
- L
  - LatchState, LimitTorque, LimZMargin, Local, LocalDef, LShift, LShift64
- M
  - MCalComplete, MemIn, MemInW, MemSw, Motor
- O
  - OLRate, Oport

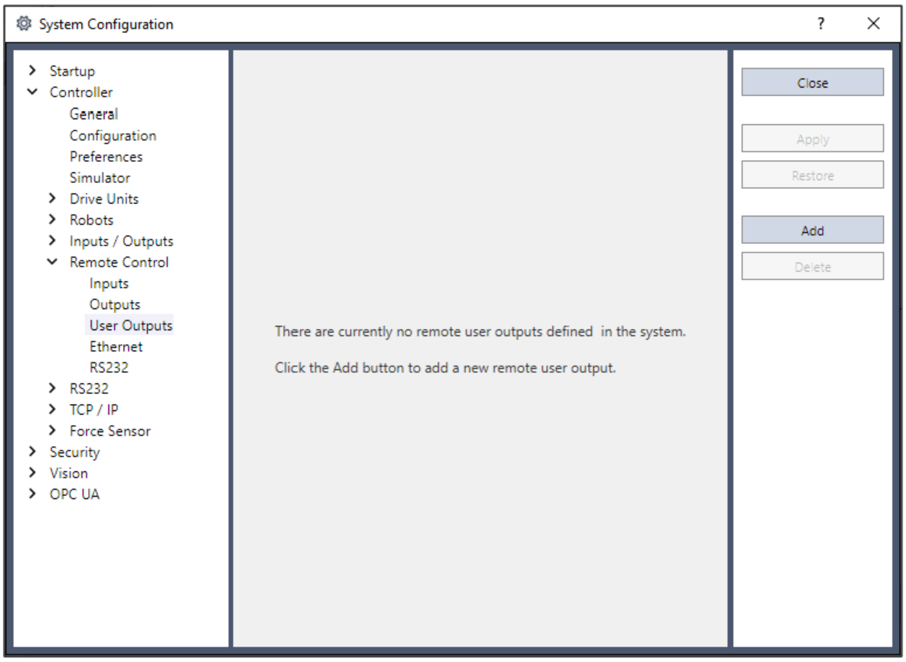
- P
  - PAgl, PauseOn, PDef, PG_Lspeed, Plane, PlaneDef, PLocal, Pls, PPls, Power, PTPBoost
- Q
  - QPDecelR, QPDecelS
- R
  - RadToDeg, RealAccel, RealPls, RealPos, RecoverPos, Rnd, RobotInfo, RobotType, RShift, RShift64
- S
  - SafetyOn, Sgn, SF_GetParam, SF_GetParam$, SF_GetStatus, SF_LimitSPeedS, SF_LimitSpeedSEnable, SF_RealSpeedS, SF_PeakSpeedS, SF_PealSpeedSClear, Sin, Speed, SpeedFactor, SpeedR, SpeedS, Sqr, Stat, Sw, SyncRobots, SysErr
- T
  - Tan, TaskDone, TaskInfo, TaskState, TCLim, TcSpeed, TeachOn, Time, TLDef, TLSet, Tool
- V
  - Val
- W
  - Weight
- X
  - XY, XYLimDef

13.4.5 How to set the user-defined output remote I/O

Add the use-defined output remote I/O

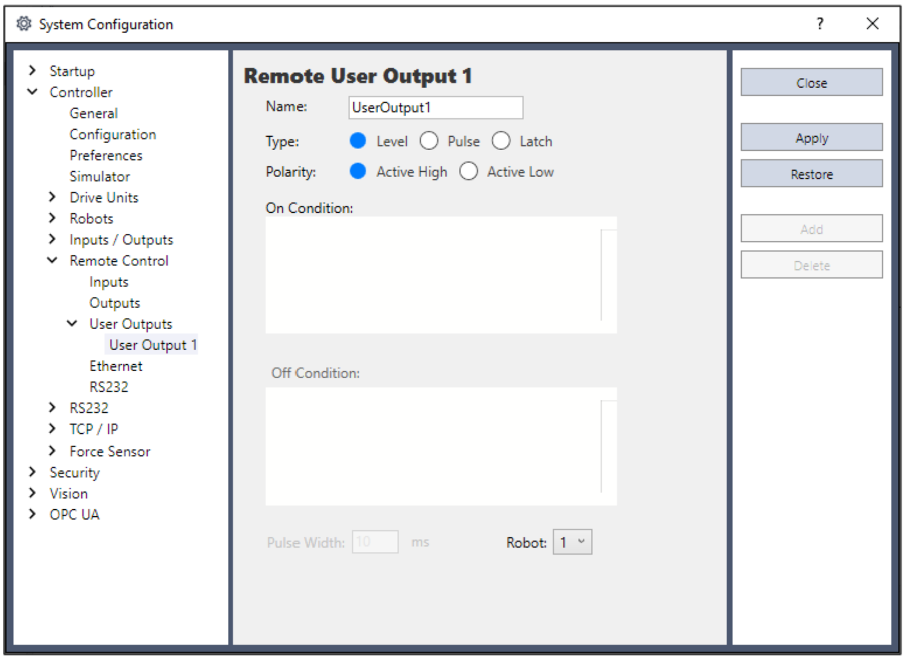
The user-defined output remote I/O are not defined as default. To use them, add the output remote I/O in the configuration dialog box and configure the output condition. The added I/O will be available in the remote output setting.

[System Configuration] - [Controller] - [Remote Control] - [User Outputs]



Click the [Add] button. The window shown below appears.

Select the items and set the condition expressions. Then click the [Apply] button.



- [Name]

  Set the name of the signal. Default is "UseroutputX".

  X = I/O number

  The name specified here is displayed in the remote output setting and the I/O monitor.

- [Type]

  Select the output type.

- [Polarity]

  Select the polarity to output when the condition is met.

  Active high: High Active low: Low

- [On Condition]

  Set the condition to start output. Setting is required for all output types.

- [Off Condition]

  Setting is required if the Latch output is selected.

- [Robot]

  Setting is required if the expression related to the manipulator is used for On and Off conditions.

  The conditions for only one robot can be set.

  This setting is not necessary if the manipulator-related condition is not used.

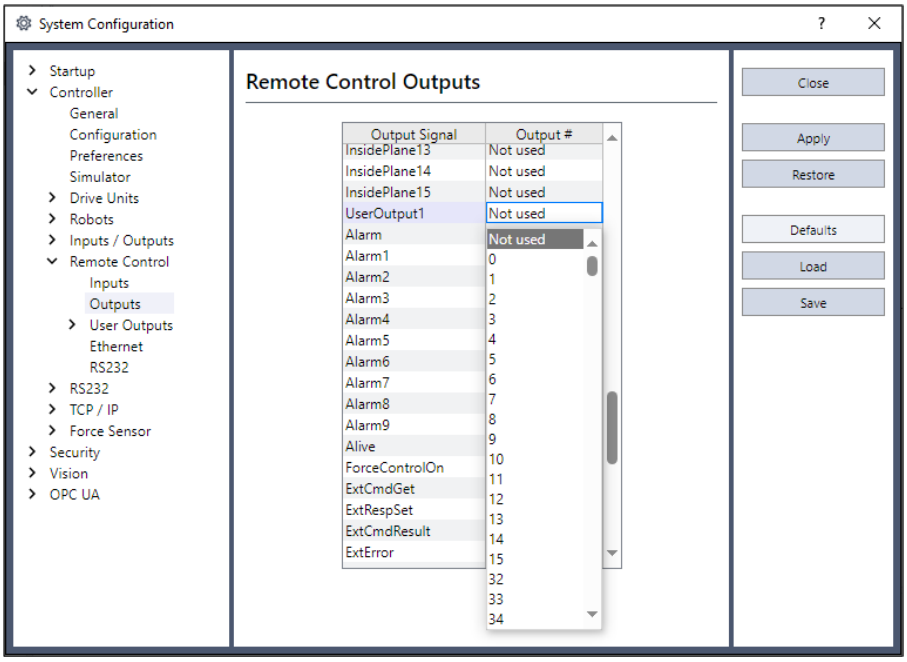
> **KEY POINTS**
>
> If unregistered robot number is specified, an initialization error occurs at the Controller restart.

Setting of remote outputs

To enable the added I/O outputs, assign the registered user definition to the target I/O.

Assignment is done by the remote output.

[System Configuration] - [Controller] - [Remote Controller] - [Outputs]

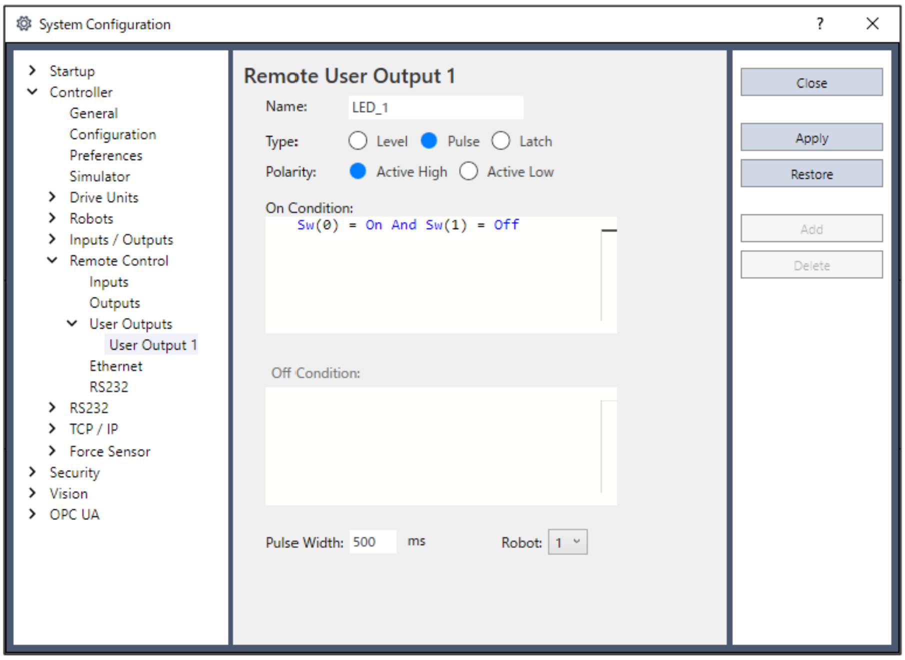


The names of the added signals appear in [Output Signal]. Select the bit to output.

13.4.6 Usage example

If you want to turn ON the bit port 8 of the standard I/O for 500 ms when the bit port 0 of the standard I/O input is ON and the bit port 1 is OFF:

User definition



1. Set "LED_1" for the [Name] in this example. Change the setting if necessary.

2. Select [Pulse] for [Type].

3. Select [Active High] for [Polarity] in order to output ON.

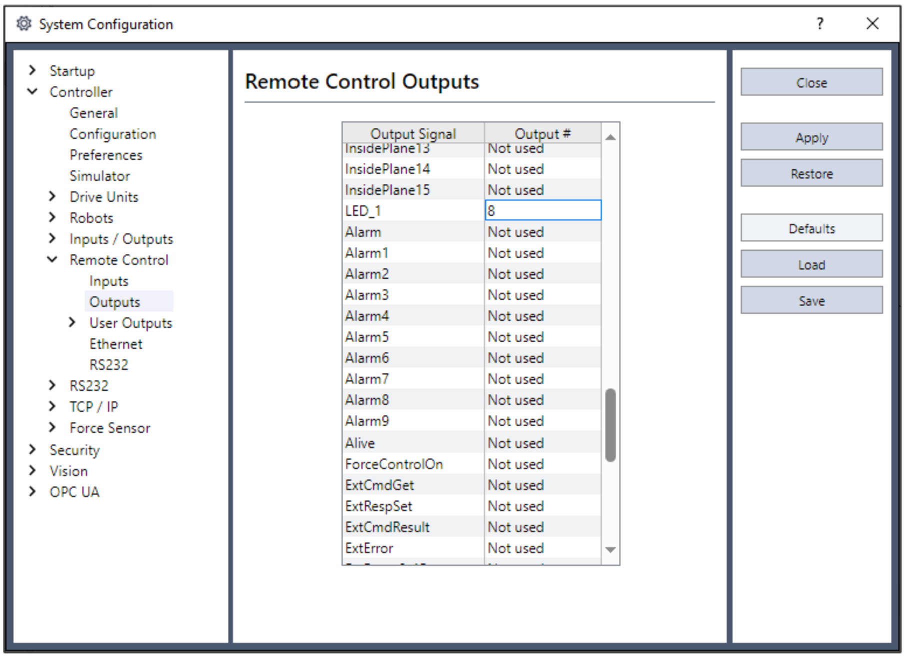
4. Set [On condition]. In this example, set the following condition expression.

```
Sw(0)  =  On  And Sw(1)  =  Off
```

5. Set “500” for [Pulse Width].

6. Click the [Apply] button.

Setting in the [Remote Control Outputs]



1. Select the output bit “8” for the configured name (LED_1).
2. Click the [Apply] button.

Now, output to the I/O will be executed according to the condition expression after rebooting.

14. RS-232C Communications

The Robot Controller supports:

- Windows part: Standard RS-232C port x 2 (Standard: Port 1001 only, High-speed: Port 1001, 1002)
- Standard RS-232C: 1 port as standard

  However, it cannot be used with the RC800 series.
- Expanded RS-232C: Option RS-232C port x 4 at maximum (2 ports per board)

  However, RS-232C port x 2 at maximum when using Force Sensor I/F board with RC700 series. (For a board, one is the maximum.)

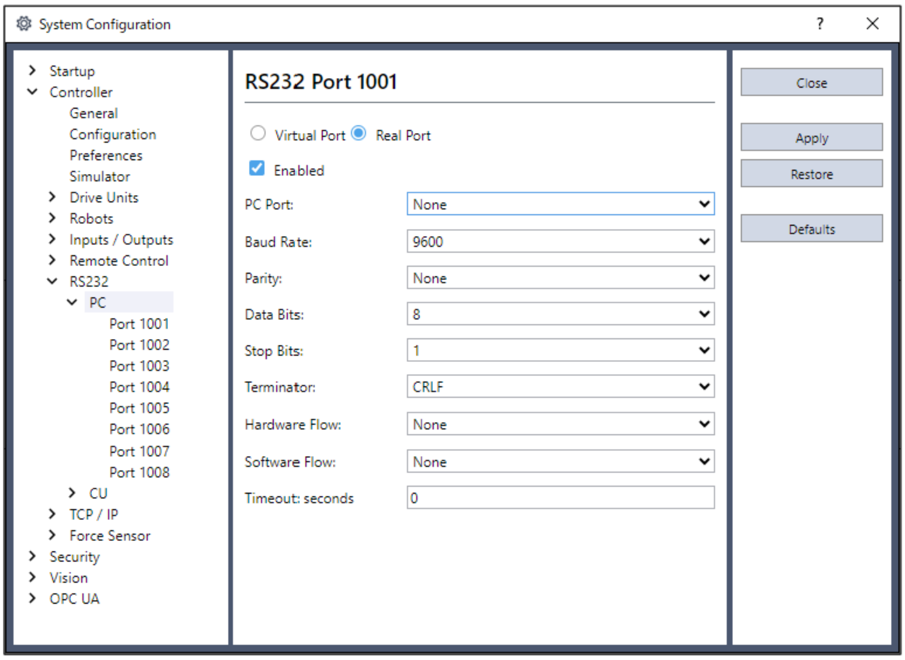
For details on RS-232C board installation, refer to the following manual:

"Robot Controller Manual"

14.1 RS-232C Software Configuration

To configure a Windows Part RS-232C port

1. The [System Configuration] command opens a dialog that contains several pages that are used to configure the system for the Epson RC+ 8.0 environment.



2. Select [Controller]-[RS-232]-[PC].

3. Set the [Enabled] check box.

4. Change the settings as desired.

5. Click the [Apply] button. The new settings are saved.

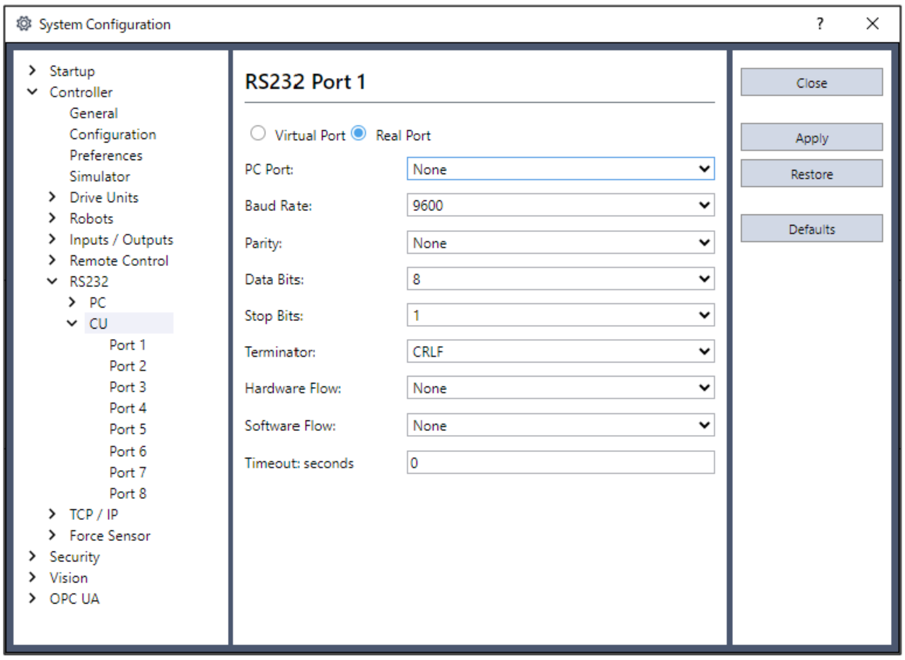
6. Click the [Close] button.

KEY POINTS

If several ports are used for communication at one time with more than a 19200 baud rate, error 2929 or 2922 may occur. In this case, select a lower baud rate or avoid using several ports at one time.

To configure a standard / option RS-232C port

1. The [System Configuration] command opens a dialog that contains several pages that are used to configure the system for the Epson RC+ 8.0 environment.



2. Select [Controller]-[RS-232]-[CU].

3. Select a port to configure.

4. Change the settings as desired.

5. Click the [Apply] button. The new settings are saved.

6. Click the [Close] button.

14.2 RS-232C Commands

The following is a list of all of the commands associated with RS-232C communications.

For details, refer to Help or the following manual:

"SPEL+ Language Reference"

Command	Description
OpenCom	Opens a communications port.
ChkCom	Returns port status: the number of bytes waiting to be read or error condition.
CloseCom	Closes a communications port.
SetCom	Sets communications port parameters at runtime or from the Command window.
Print #	Sends characters out of the port.
Input #	Receives characters from the port into one or more variables.
Line Input #	Receives one line characters from the port into one string variable.
Read #	Receives one or more characters from the port into one string variable.
ReadBin#	Receives one or more bytes from the port.
Write #	Sends characters out of the port.

Command	Description
WriteBin#	Sends one or more bytes out of the port.

15. TCP/IP Communications

Epson RC+ 8.0 supports 16 TCP/IP ports that allow peer to peer communications.

This chapter contains instructions on using TCP/IP, including IP addresses of LAN port and Windows TCP/IP configuration.

15.1 TCP/IP Setup

Before you can use TCP/IP communications between PCs and controllers, you must configure your network.

The following sections describe basic network configuration.

15.1.1 Ethernet Hardware

The Controller includes a built in Ethernet interface with an RJ45 connector accessible from the controller front panel. It supports 10BaseT (10 Mbps) and 10BaseTX (100 Mbps).

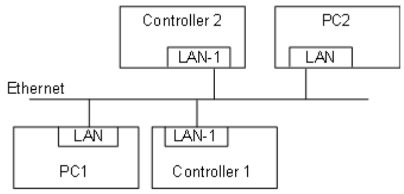
Your PC will need a 10BaseT 10/100 adapter to communicate with the Controller via Ethernet.

15.1.2 IP Address

The controller has a fixed IP address that you can configure from Epson RC+ 8.0. To configure the IP address, mask, and gateway for the controller, see below.

[System Configuration] Command (Setup Menu)

The following table shows a typical IP address configuration.



Host Name	IP Address	Subnet	Subnet Mask
PC1	192.168.0.1	192.168.0	255.255.255.0
PC2	192.168.0.2	192.168.0	255.255.255.0
Controller1	192.168.0.3	192.168.0	255.255.255.0
Controller2	192.168.0.4	192.168.0	255.255.255.0

In this example, the network address (subnet) is 192.168.0. With a subnet mask of 255.255.255.0, there can be 254 hosts on this subnet. (0 and 255 cannot be used.)

Refer to the Microsoft Windows operating system manual for instructions on setting the PC IP address.

15.1.3 IP Gateway

If you are connecting PCs and controllers on different networks, you will need to route traffic between the networks using one or more routers. Each device communicating via Ethernet will need to have their default gateway address set to the address of the router for its subnet.

To configure the gateway address for the controller, see below.

[System Configuration] Command (Setup Menu)

15.1.4 Testing Windows TCP/IP setup

Use the ping command from a Command Window to test communications.

First, do a loopback test to check if you can ping your own address by using the local IP address:

```
C:\>ping 127.0.0.1
Pinging 127.0.0.1 with 32 bytes of data:
Reply from 127.0.0.1: bytes=32 time<10ms TTL=128
Reply from 127.0.0.1: bytes=32 time<10ms TTL=128
Reply from 127.0.0.1: bytes=32 time<10ms TTL=128
Reply from 127.0.0.1: bytes=32 time<10ms TTL=128
C:\>
```

Ping your PC's IP address:

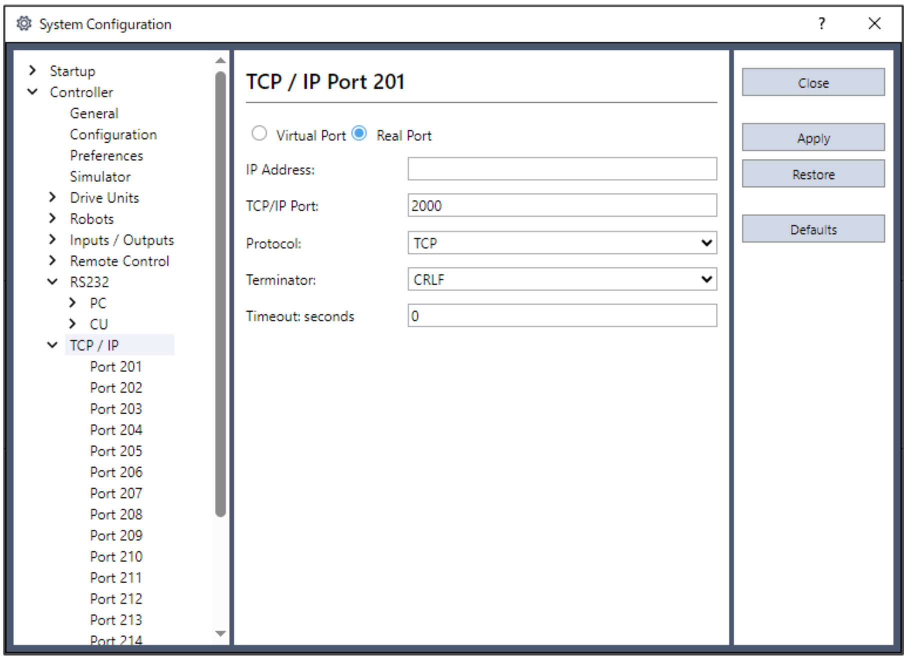
```
C:\>ping 192.168.0.1
Pinging 192.168.0.1 with 32 bytes of data:
Reply from 192.168.0.1: bytes=32 time<10ms TTL=128
Reply from 192.168.0.1: bytes=32 time<10ms TTL=128
Reply from 192.168.0.1: bytes=32 time<10ms TTL=128
Reply from 192.168.0.1: bytes=32 time<10ms TTL=128
C:\>
```

Now ping controller on the network. :

```
C:\>ping 192.168.0.3
Pinging pc2 [192.168.0.3] with 32 bytes of data:
Reply from 192.168.0.3: bytes=32 time<10ms TTL=128
Reply from 192.168.0.3: bytes=32 time<10ms TTL=128
Reply from 192.168.0.3: bytes=32 time<10ms TTL=128
Reply from 192.168.0.3: bytes=32 time<10ms TTL=128
C:\>
```

15.2 TCP/IP Software Configuration

You can configure TCP/IP settings for the controller in a SPEL+ program using the SetNet command. You can also configure settings from the [TCP/IP] tab on the [Setup]-[System Configuration] dialog.



To configure a TCP/IP port

1. Select the TCP/IP port you want to configure from [Setup]-[System Configuration]-[Controller]-[TCP/IP].
2. Enter the IP address for the controller or PC that you want this controller to communicate with.

   The controller does not support DNS. You must specify an IP address for the host you are communicating with. You cannot specify a name for the host.
3. Enter the TCP/IP port number. This must be the same port number that is used on the host device. It must be different from any of the other TCP/IP port numbers used for the other TCP/IP ports.
4. Change the other settings as desired.
5. Click the [Apply] button. The new settings are saved.
6. Click the [Close] button.

15.3 TCP/IP Commands

Here is a list of all of the commands associated with TCP/IP communications.

For details, refer to Help or the following manual:

"SPEL+ Language Reference"

Command	Description
OpenNet	Opens a TCP/IP port.
ChkNet	Returns port status: the number of bytes waiting to be read or error condition.
CloseNet	Closes a TCP/IP port.
SetNet	Sets communications port parameters at runtime or from the Command window.
Print #	Sends characters out of the port.
Input #	Receives characters from the port into one or more variables.
Line Input #	Receives one line characters from the port into one string variable.
Read #	Receives one or more characters from the port into one string variable.
ReadBin#	Receives one or more bytes from the port.
Write #	Sends characters out of the port.
WriteBin#	Sends one or more bytes out of the port.

16. Security

16.1 Overview

The Security function allows you to manage Epson RC+ 8.0 users and also monitor usage.

The administrator and user to use the security function can be set by the customer.

When the Security function is activated, administrators can add groups and users. Each group can have one or more rights associated with it. For example, you can create a group called Maintenance that has rights to edit robot points, use Jog & Teach, and enable you to use the Command Window. When a user attempts to do something that he/she does not have a right for, a message "Permission denied" will be displayed.

Each login session is recorded in a Microsoft Access compatible data base.

Security Log Viewer is included that allows you to view each session's activity.

User can login to Epson RC+ with a name and password. Depending on your settings, Epson RC+ can use the Windows username to log in automatically.

16.2 Security Configuration

Epson RC+ 8.0 requires a path for security files. If you have more than one system on a network, it is recommended that you setup the security files path for all systems to store the security logs in a server on the network.

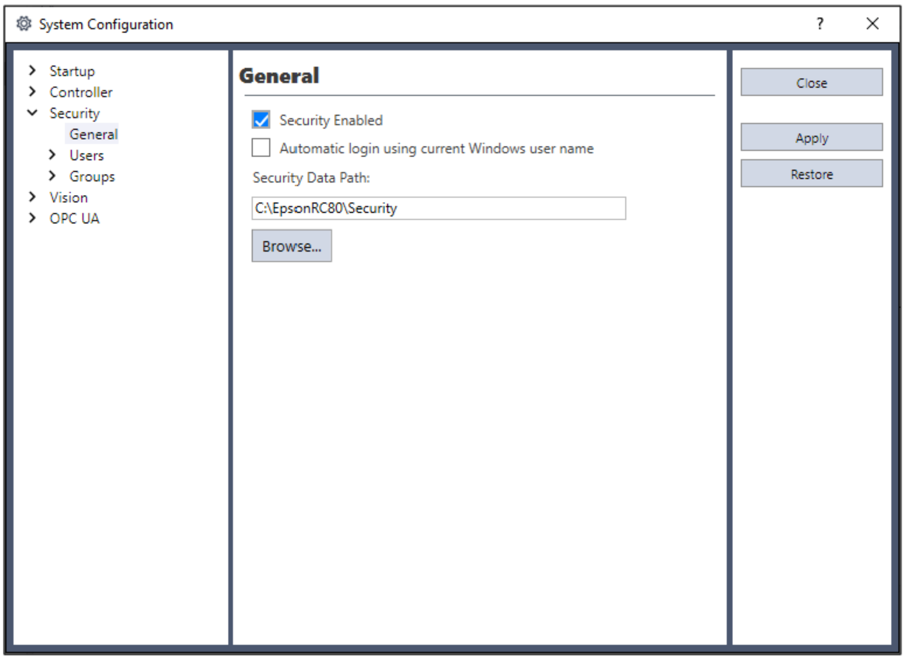
To administer Epson RC+ 8.0 security:

1. Start Epson RC+ 8.0.
2. Select [Setup]-[System Configuration].
3. Select [Security].
4. On the [General] tree, type in the path for your security files or click the [Browse] button.
5. Select [Users].
6. To add a user, click the [New] button.

Each new user belongs to the Guest group by default. Click in the group field, then click the dropdown button to select the desired group.

16.2.1 [Other]

This tab allows you to configure the general security settings.



Automatic log in using current Windows user name

Check this box if you want Epson RC+ 8.0 to use the current Windows login ID.

When the Security function is active, check users in the security system when the Epson RC+ 8.0 starts up. If the login is successful, the login dialog is not displayed.

Security data path

This is the path where security files will be stored.

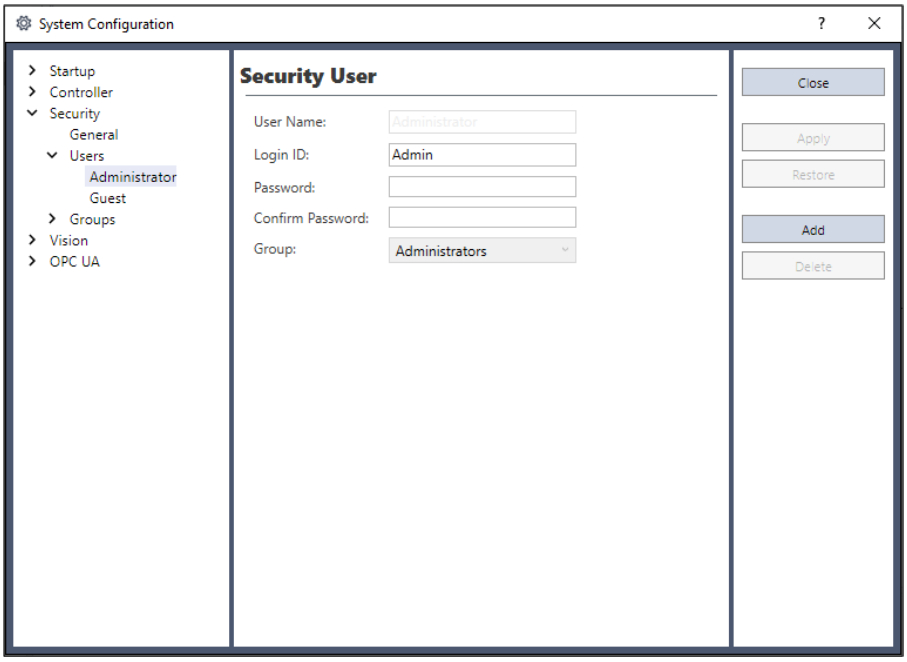
This path should be protected with Windows security rights so that only Administrators can delete the files in this path. All other Epson RC+ users should have only read rights to the files in this path.

16.2.2 Security User: Administrator

This page allows you to add and remove Epson RC+ 8.0 users.

Two users are permanent: Administrator and Guest. Only the passwords can be changed for these users.

You should always use a password for the Administrator. No password is set at shipment time.



To add a user

1. Click the [Add] button.
2. A new user will be added to the tree.
3. Click the new user [Group].
4. Click the dropdown button and select the group for the user.

To delete a user

1. Click the [User] you want to delete.
2. Click the [Delete] button.
3. A confirmation message to delete the user will appear.

To change a user's group

1. Click the [Group] for the user you want to change.
2. Click a dropdown button in the field and select a new group.

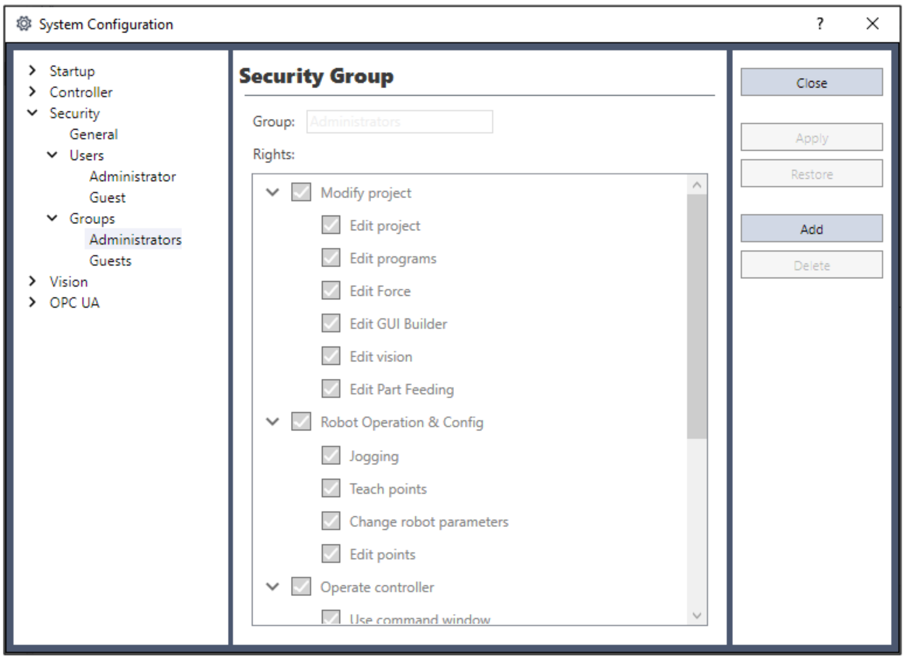
Editing Name, Login ID, and Password

1. Click the [User] you want to change.
2. Edit the field. All fields are not case sensitive.

16.2.3 Security Group: Administrator

This page allows you to configure user groups.

Every Epson RC+ 8.0 user must belong to a group. Two groups cannot be deleted or modified: Administrators and Guests. Administrators have full rights. Guests have no rights.



To add a group

1. Click the [Add] button.
2. Type in a name for the group.
3. Click the [Apply] button.

To delete a group

1. Select the group you want to delete.
2. Click the [Delete] button.
3. A confirmation message to delete the group will appear.

To change rights for groups

1. Select the group you want to change rights for.

   Note that you cannot change rights for Administrators and Guests.

2. Add a right.

   Set the checkbox(es) for the desired rights in the [Rights] checkbox list.

3. Remove a right.

   Clear the checkbox(es) for the rights you want to remove in the [Right] checkbox list.

Group Rights

The list below shows the rights that are available for user groups. Administrators have full rights. Guests have no rights.

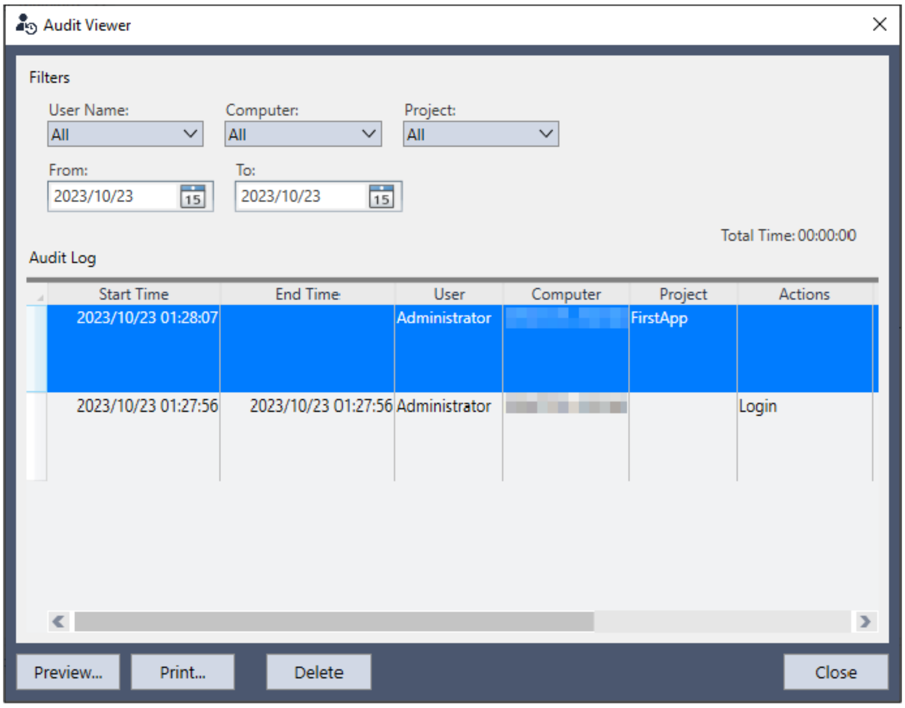
Right	Description
Editing a project	Users can edit projects.
Editing Programs	Users can edit program.
Edit force	Users can edit force parameters.
Edit GUI Builder	Users can edit GUI Builder parameters.

Right	Description
Edit vision	Users can edit vision parameters.
Edit Part Feeding	Users can edit Part Feeding parameters.
Jog	Users can open the [Jog & Teach] dialog and jog a robot.
Teaching	The points can be taught and deleted from the [Jog & Teach] dialog.
Change robot parameter	Users can open the [Robot Manager] dialog and change the settings.
Edit points	Users can change points.
Use command window	Users can open the command window and execute commands.
Change memory I/O	Users can turn ON/OFF memory I/O bits.
Output port ON	Users can turn ON/OFF outputs.
License configuration	Users can change license configurations in [Setup]-[License Configuration].
Configure system	Users can configure the entire Epson RC+ system.
Configure security	Users can change security settings.
Check security log	Users can see security logs.
Delete security log	Users can delete security logs in [Tools]-[Audit Viewer].

16.3 Security Audit Viewer

When the Security function is enabled, Epson RC+ 8.0 will keep track of who logs into the system and actions performed.

Activity is stored to the security data path in the Microsoft Access compatible data base format. To view the security logs, select [Audit Viewer] from the [Tools] menu.



16.4 SPEL+ Security Command

Here are the SPEL+ commands that are enabled with the Security function.

For details, refer to Help or the following manual:

"SPEL+ Language Reference"

Command	Description
LogIn Function	Logs in the application as another user at runtime.
GetCurrentUser$ Function	Returns the login ID of the current user.

17. Conveyor Tracking

17.1 Overview

Conveyor Tracking is a process in which a robot picks up parts from a stationary or moving conveyor that are found by a vision system or sensor.

The Epson RC+ 8.0 Conveyor Tracking option supports both tracking and indexed conveyor systems.

Tracking conveyor system

Conveyor moves constantly. Vision system or sensor system finds the parts on it and robot picks them up as they move. During tracking, the robot can move along with the part as it picks up parts.

Indexed conveyor system

Conveyor moves a specified distance and then stops. The vision system finds the parts and robot picks up each part. After finding and picking up all parts, the conveyor moves again.

Up to 16 logical conveyors can be defined in each project. To define a logical conveyor, set a conveyor number, a robot number, encoder and select vision or sensor.

Multiple conveyors and multiple-robot conveyors are supported.

Applicable conveyor tracking models and connection methods are shown below.

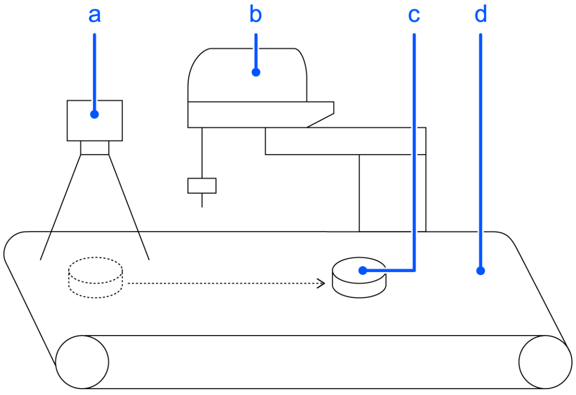
Controller	Supported/Not Supported	Connection Method	
		Conveyor Tracking Option Kit B	Pulse Generator (PG) Board
RC90 Series	Supported	Not supported	Max 2 boards supported
RC700-A	Supported	Not supported	Max 4 boards supported
RC700-D, RC700-E	Supported	Not supported	Max 3 boards supported
T series Manipulator	Not supported	-	-
VT series Manipulator	Not supported	-	-
RC800 series	Supported	Supported	Max 3 boards supported

For details on multi robots, refer to the following manual.

"Robot Controller Manual"

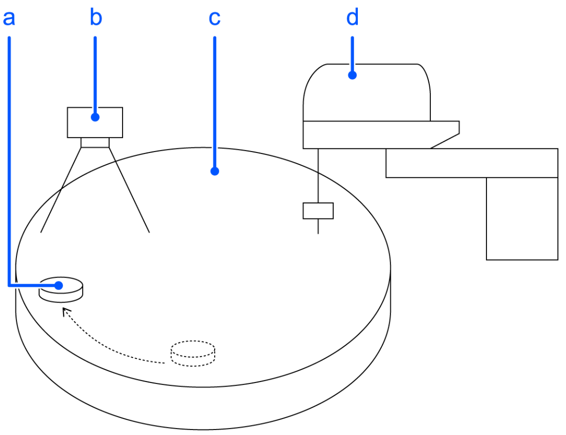
There are two types of conveyors that can be used with the Conveyor Tracking option: straight conveyors and circular conveyors, as shown in the figure below. These conveyors have different calibration and programming methods.

- Straight conveyor tracking system



Symbol	Description
a	Camera
b	Robot
c	Workpiece
d	Straight conveyor

- Circular conveyor tracking system

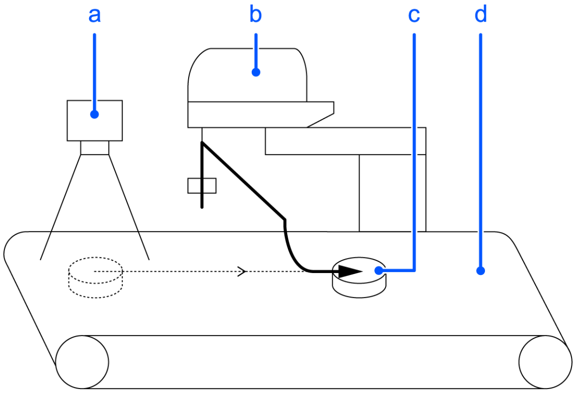


Symbol	Description
a	Workpiece
b	Camera
c	Circular conveyor
d	Robot

17.2 Conveyor Tracking Processes

Tracking conveyor system

1. Vision system or sensor system finds the parts on a continuously moving conveyor.
2. Robot picks up the parts on the conveyor as they move.

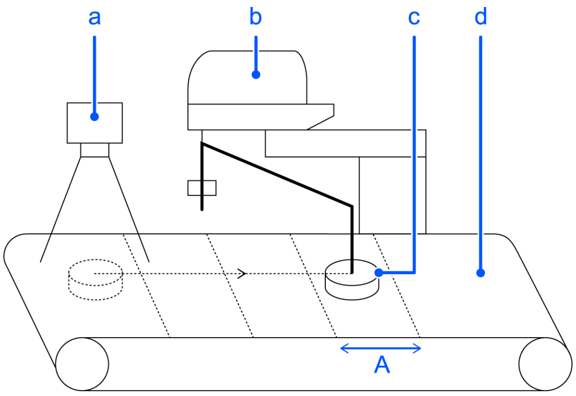


Configuration example of Tracking Conveyor System (Vision System)

Symbol	Description
a	Camera
b	Robot
c	Workpiece
d	Straight conveyor

Indexed conveyor system

1. Conveyor moves a specified distance.
2. Vision system or sensor system finds the parts on the conveyor when it stops.
3. Robot picks up the parts found by vision system or sensor system.
4. After finding and picking up all parts, conveyor moves by the specified distance again.



Configuration example of Indexed Conveyor System (Vision System)

Symbol	Description
a	Camera
b	Robot
c	Workpiece
d	Straight conveyor
A	Index interval

17.3 System Structure

Structure of Vision Conveyor Tracking System

The structure of the vision conveyor tracking system is shown in the figures below.

Not to reduce the picking accuracy, you need to set the same timing for the camera to search for parts on the conveyor and for the encoder on the conveyor to latch position. It is also recommended to use a photoelectric sensor to match the timing of imaging a moving workpiece.

There are hardware trigger and software trigger as trigger methods for latching the camera image and encoder count.

Hardware trigger:

Structure that inputs from the photoelectric sensor or controller I/O to the trigger terminal of the camera and the encoder latch terminal of the PG board
Asynchronous reset mode

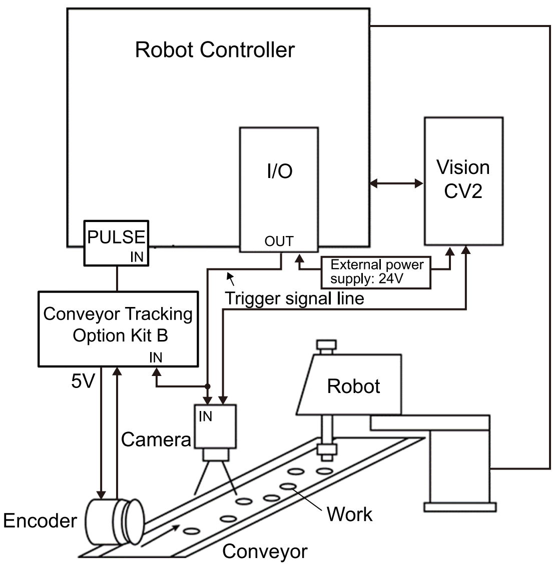
Software trigger:

Structure without trigger signal The work is imaged by executing the vision sequence, and the SPEL command that latches the encoder count is executed.

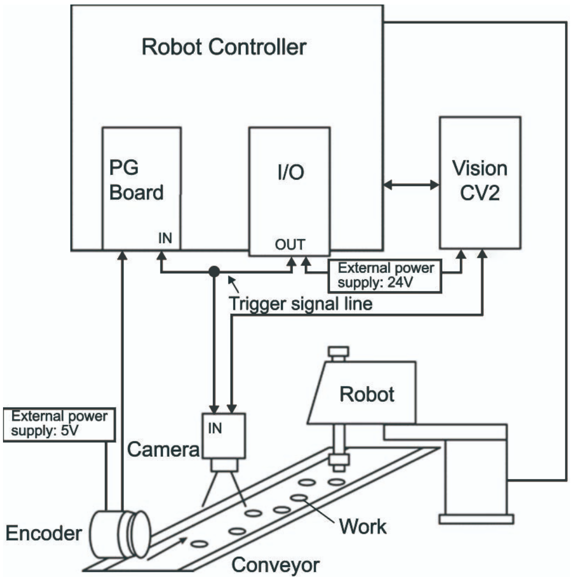
Vision Conveyor Tracking Structure Example 1 (using a hardware trigger)

Trigger signal that latches the camera image and encoder count is output from the controller I/O

- When using conveyor tracking option kit B



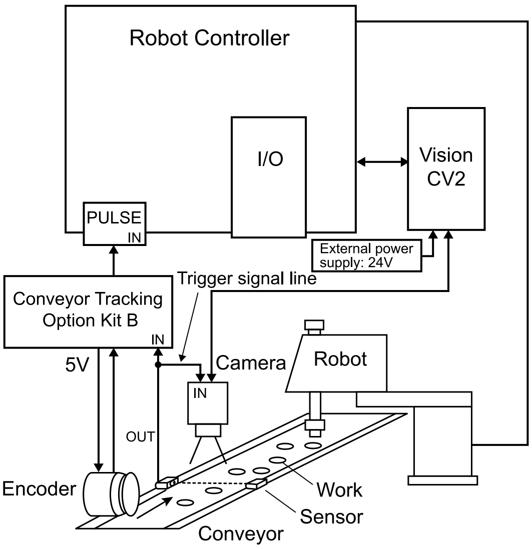
- When using PG board



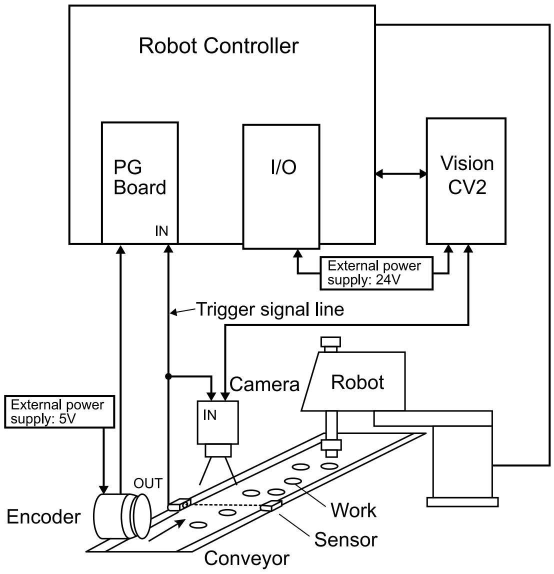
Vision Conveyor Tracking Structure Example 2 (using a hardware trigger)

Trigger signal that latches the camera image and encoder count is output from the photoelectric sensor

- When using conveyor tracking option kit B



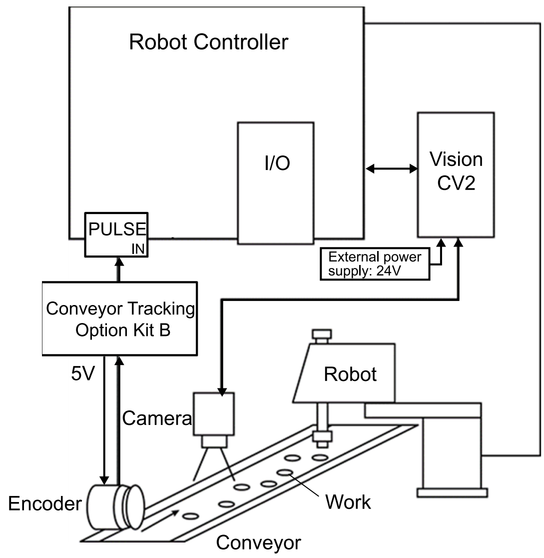
- When using PG board



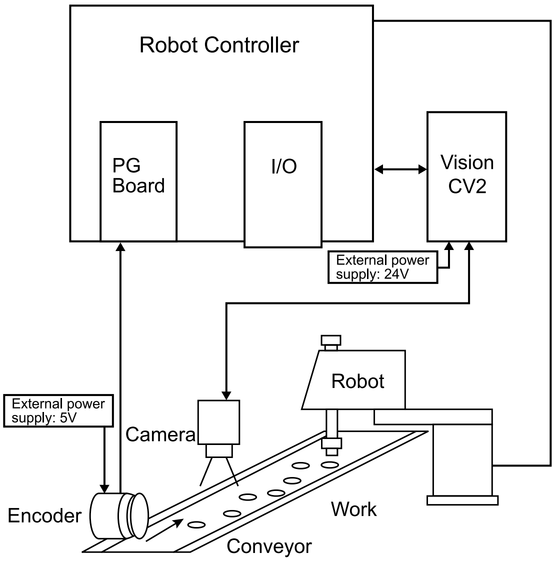
Vision Conveyor Tracking Structure Example 3 (using a software trigger)

Trigger signal is not used

- When using conveyor tracking option kit B



- When using PG board

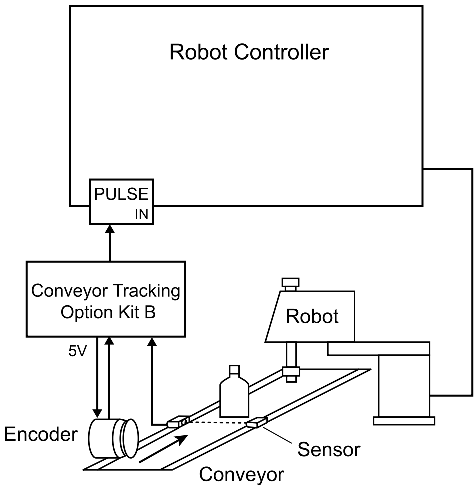


Structure of Sensor Conveyor Tracking System

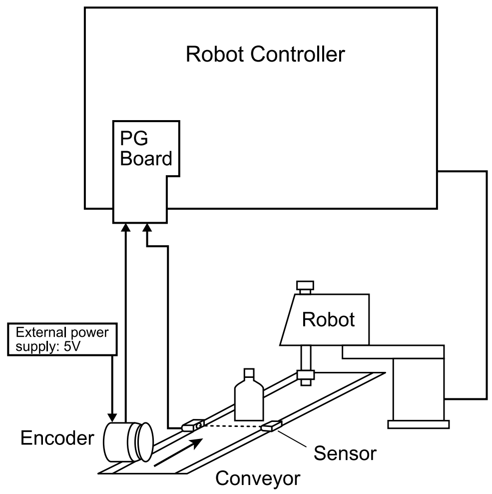
The structure of the Sensor Conveyor Tracking System is shown in the figure below. This system uses a hardware trigger.

Sensor Conveyor Tracking Overview

- When using conveyor tracking option kit B



- When using PG board



17.4 Hardware Installation

To use conveyor tracking, you must install encoders for each physical conveyor on the system. Each encoder is wired to a box connector (CH1, CH2) on conveyor tracking option kit B or to a single channel on the pulse generator (PG) board. Conveyor tracking option kit B allows up to two encoders to be installed. The pulse generator (PG) board allows up to four encoders to be installed. A trigger input is also provided for each encoder to latch position, such as when used with a vision camera with a strobe.

17.4.1 Conveyor Tracking Option Kit B

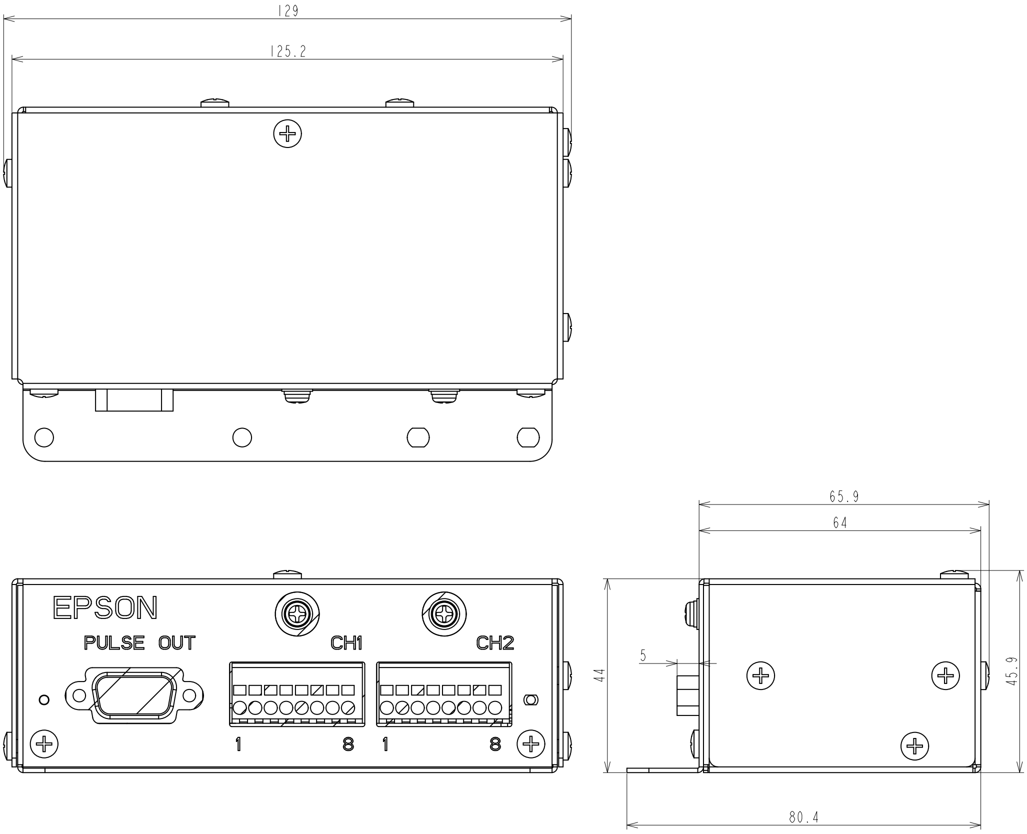
17.4.1.1 Conveyor Tracking Option Kit B Specifications

The following table shows the main specifications of conveyor tracking option kit B.

Name	Conveyor tracking option kit B
Compatible Controller	RC800 series
Encoder channels	2 channels
Encoder Type	ABZ phase differential input (RS-422 line receiver)
Input Pulse Rate	Max. 5 MPPS
Input Signal	Conveyor pulse latch input
Connector	D-SUB 15 pins 3 rows (Controller side) Push-in (encoder side)
Cable Set	3m, 10m

17.4.1.2 External Dimensions

The figures below show the external dimensions of the pulse box for the conveyor tracking option kit B.



17.4.1.3 Signal Connections

Signal connections of Controller PULSE IN connector

The signal connections of the PULSE IN connector, which is the Controller's conveyor tracking I/F port, are shown in the table below.

Pin	Dir	Signal	Description
1	In	+A2	Phase +A signal for Counter 2 for CH2
2	In	+B2	Phase +B signal for Counter 2 for CH2
3	In	+A1	Phase +A signal for Counter 1 for CH1
4	In	+B1	Phase +B signal for Counter 1 for CH1
5	In	TRG1_IN	Trigger input for Counter 1
6	In	-A2	Phase -A signal for Counter 2 for CH2
7	In	-B2	Phase -B signal for Counter 2 for CH2
8	In	-A1	Phase -A signal for Counter 1 for CH1
9	In	-B1	Phase -B signal for Counter 1 for CH1
10	-	GND	Power supply ground
11	In	TRG2_IN	Trigger input for Counter 2
12	Com	TRG2_COMMON	Trigger common for Counter 2 for CH2
13	Com	TRG1_COMMON	Trigger common for Counter 1 for CH1

14	-	GND	Power supply ground
15	Out	+24V	24V power supply for encoder power generation

Carefully note the following when creating a cable for conveyor tracking applications.

⚠ CAUTION

24 VDC for operation of conveyor tracking option kit B is supplied through the PULSE IN connector. Use carefully as explained below.

- Do not connect anything other than a conveyor tracking cable to the PULSE IN connector. Otherwise, connected equipment and Controllers may get damaged.
- Do not use 24 VDC even if the customer prepares conveyor tracking cable. Our guarantee does not cover damage to connected equipment or Controllers used in this way.
- Commercially available D-SUB 15-pin cables cannot be used due to different pin assignments and wiring. When preparing cables yourself, follow the table, ensuring that the pin assignments and wiring are correct.

Signal connections of PULSE OUT connector with conveyor tracking option kit B

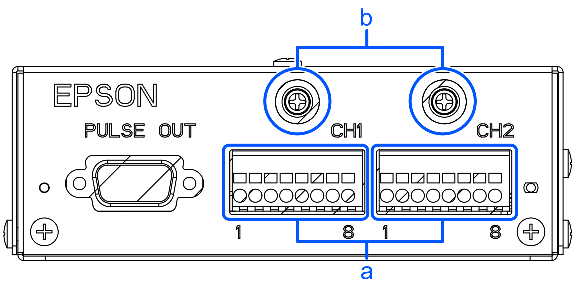
The signal connections of PULSE OUT connectors with conveyor tracking option kit B is shown in the table below.

Pin	Dir	Signal	Description
1	Out	+A2	Phase +A signal for Counter 2 for CH2
2	Out	+B2	Phase +B signal for Counter 2 for CH2
3	Out	+A1	Phase +A signal for Counter 1 for CH1
4	Out	+B1	Phase +B signal for Counter 1 for CH1
5	Out	TRG1_IN	Trigger input for Counter 1
6	Out	-A2	Phase -A signal for Counter 2 for CH2
7	Out	-B2	Phase -B signal for Counter 2 for CH2
8	Out	-A1	Phase -A signal for Counter 1 for CH1
9	Out	-B1	Phase -B signal for Counter 1 for CH1
10	-	GND	Power supply ground
11	Out	TRG2_IN	Trigger input for Counter 2
12	Com	TRG2_COMMON	Trigger common for Counter 2 for CH2
13	Com	TRG1_COMMON	Trigger common for Counter 1 for CH1
14	-	GND	Power supply ground
15	In	+24V	24V power supply for encoder power generation

⚠ CAUTION

Do not connect cables other than those supplied with conveyor tracking option kit B. Such cables may damage connected equipment and conveyor tracking pulse boxes.

Signal wiring for encoder connectors (CH1, CH2) with conveyor tracking option kit B



The signal connections of CH1 and CH2, the encoder connection terminals (a) for users, are shown in the table below. The pins are arranged in the order of 1→8 as printed on the main unit, and CH1 and CH2 have the same signal connections.

CH1

Pin	Dir	Signal	Description
1	In	+B1	Phase +B signal for Counter 1
2	In	-B1	Phase -B signal for Counter 1
3	In	+A1	Phase+ A signal for Counter 1
4	In	-A1	Phase -A signal for Counter 1
5	In	TRG1_IN	Trigger input for Counter 1
6	Com	TRG1_COMMON	Trigger common for Counter 1
7	GND	GND	Power supply ground for +5 V power supply for encoder
8	Out	+5V	+5 V power supply for encoder

CH2

Pin	Dir	Signal	Description
1	In	+B2	Phase+ B signal for Counter 2
2	In	-B2	Phase- B signal for Counter 2
3	In	+A2	Phase+ A signal for Counter 2
4	In	-A2	Phase- A signal for Counter 2
5	In	TRG2_IN	Trigger input for Counter 2
6	Com	TRG1_COMMON	Trigger common for Counter 2
7	GND	GND	Power supply ground for +5 V power supply for encoder
8	Out	+5V	+5 V power supply for encoder

- Our conveyor system does not require Z-phase wiring. Insulate to prevent contact with panel, etc.
- Pins 1 to 4: Connect to encoder output +A, -A, +B, and -B.
- Pins 5 to 6: When latching conveyor pulse by external signal, connect to latch signal. Exactly when the signal is turned OFF to ON, the encoder pulse is latched.
- Pins 7 to 8: 5 V power supply and GND for encoders. Connect GND to 7 and power supply terminal to 8.
- Wiring shield screw: Secure the wiring shield with the shield fastening screw (b) on top of the connector.

17.4.1.4 Connection method

Conveyor tracking option kit B can be connected for 2 CHs. Push in to connect. Connect with the power off.

The wire to be connected should be processed into AWG24 to 16 or a ferrule terminal.

For details on connectors and wire connections, see the [SPT 1.5/ 8-H-3.5] specifications.

The counter trigger terminal is bipolar.

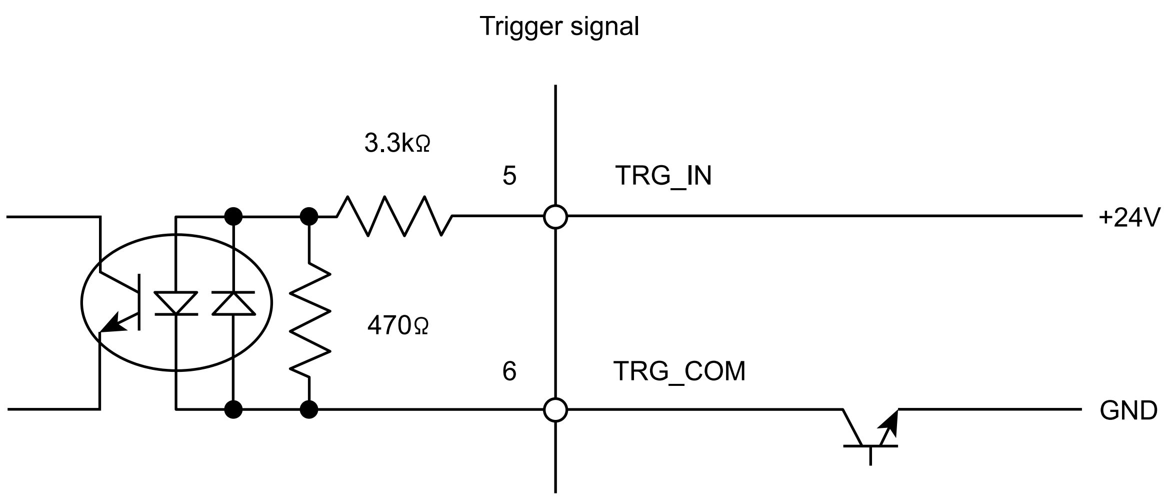
⚠ WARNING

Keep cables connected at all times and provide protection with guarding covers. Do not place heavy objects on cables, bend excessively, pull forcibly, or pinch. This may cause cable damage, wire breakage, or poor contact, resulting in improper system operation or electric shock.

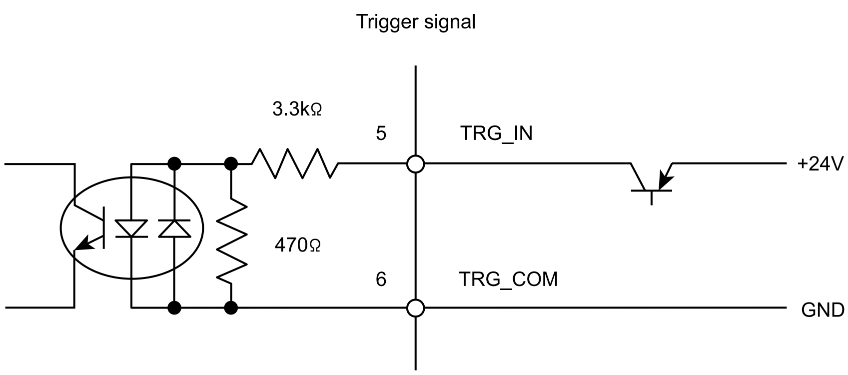
17.4.1.5 Connection Drawing

- Input voltage range: +24 V±10%
- Input current: 5 mA Typ, when +24 V input

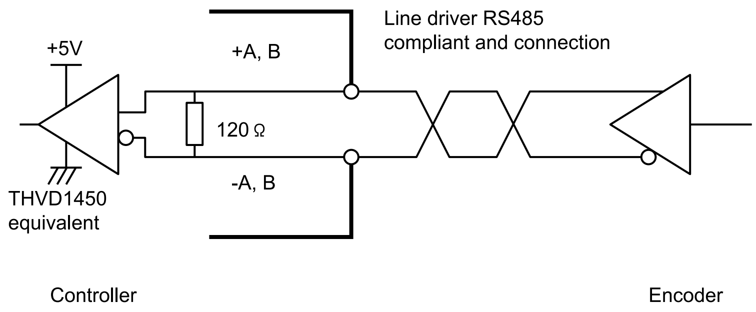
Input Circuit Drawing and Wiring Example 1



Input Circuit Drawing and Wiring Example 2

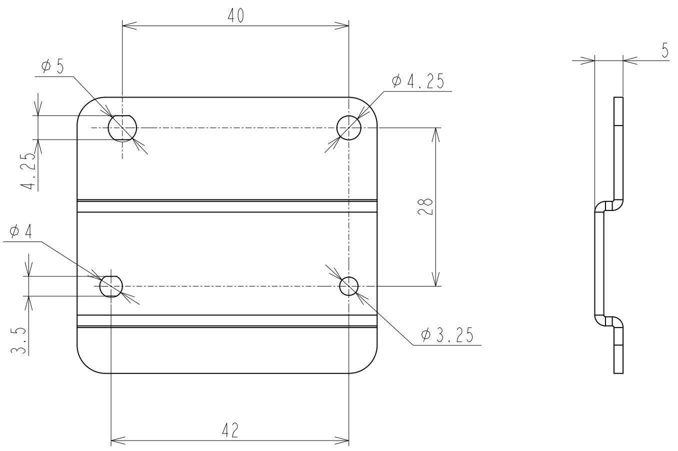


Input Circuit Drawing and Wiring Example 3



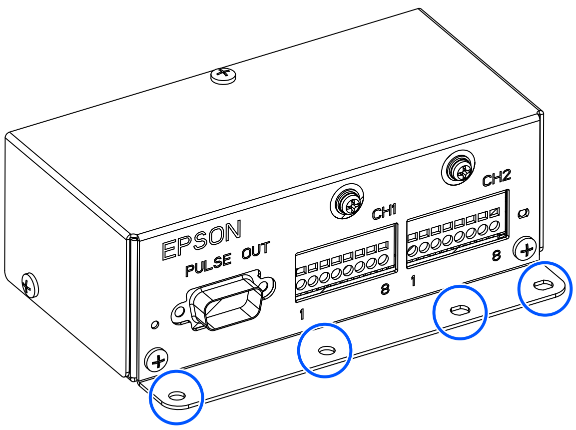
17.4.1.6 Methods for Installation

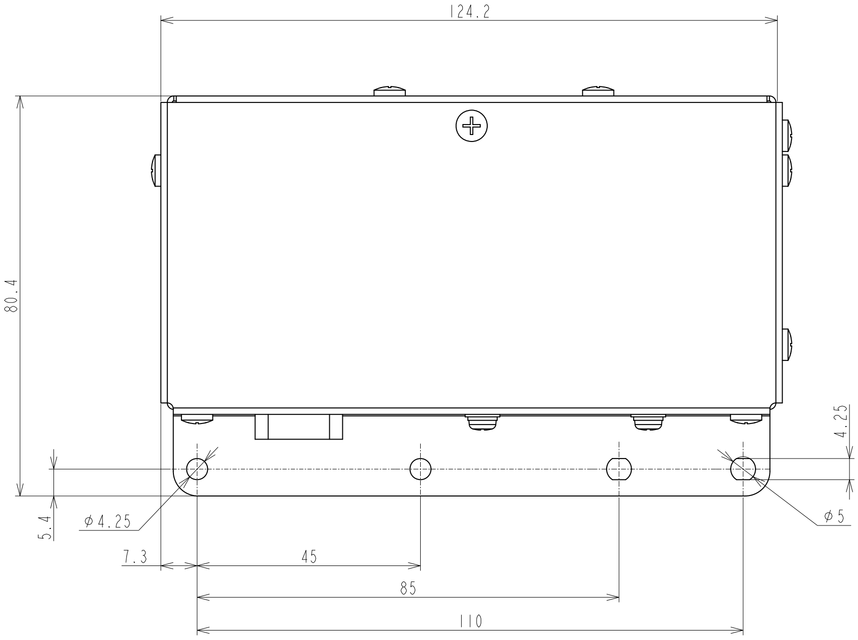
By using the included mounting panel or commercially-available DIN rail mountain brackets, the conveyor tracking option kit B can be installed in four different ways.



- Placed on the floor

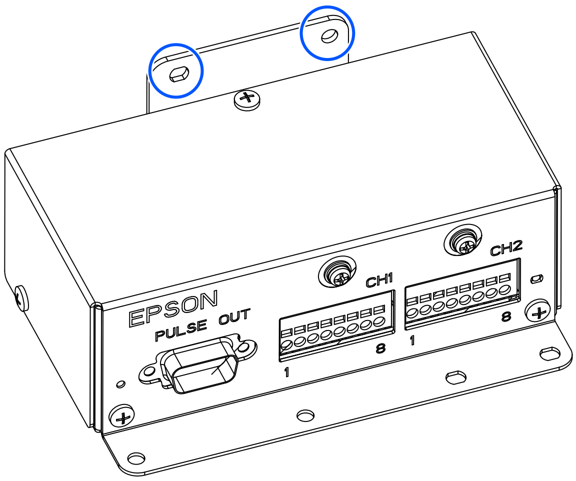
  Do not install the included panel, and use the securing holes on the bottom of the unit to secure the kit with screws.

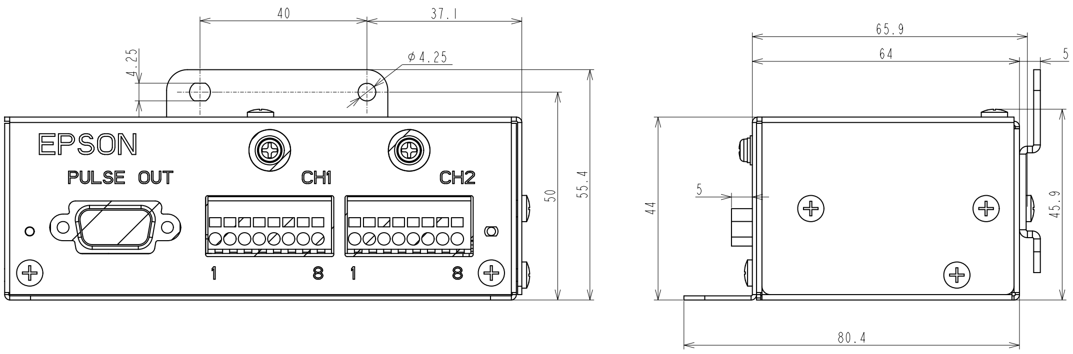




- Wall-mounted

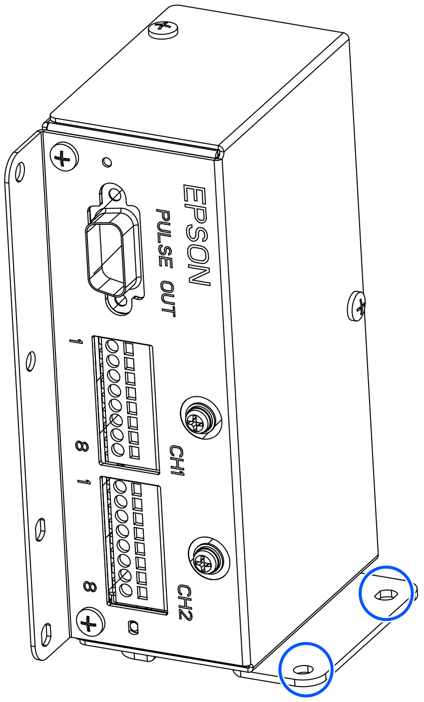
  Install the included panel on the rear of the unit and use the securing holes on the included panel to secure the kit with screws.

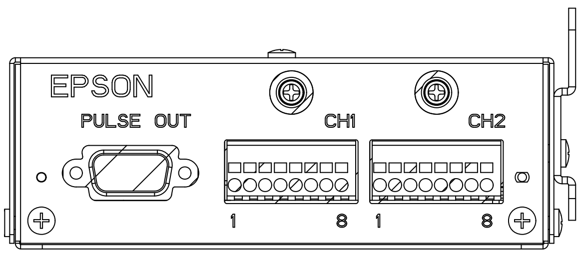


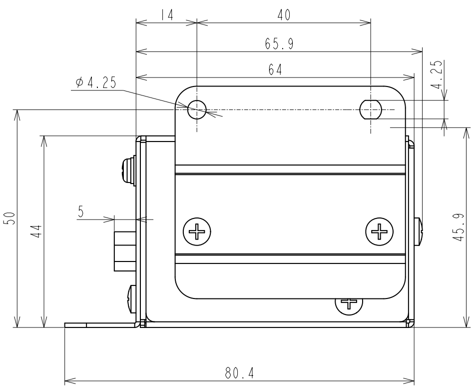


- Placed vertically

  Install the included panel on the side of the unit, turn the unit 90 degrees, and use the securing holes on the included panel to secure the kit with screws.

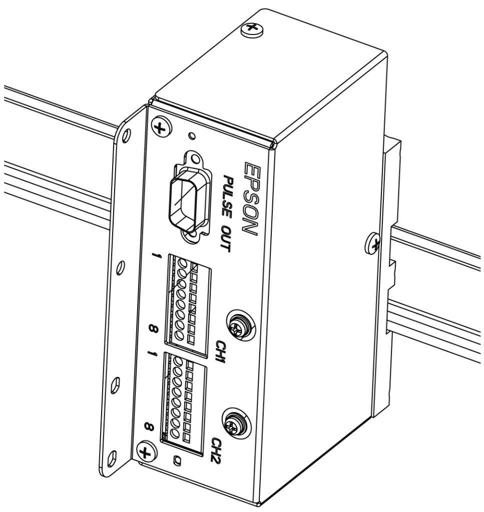


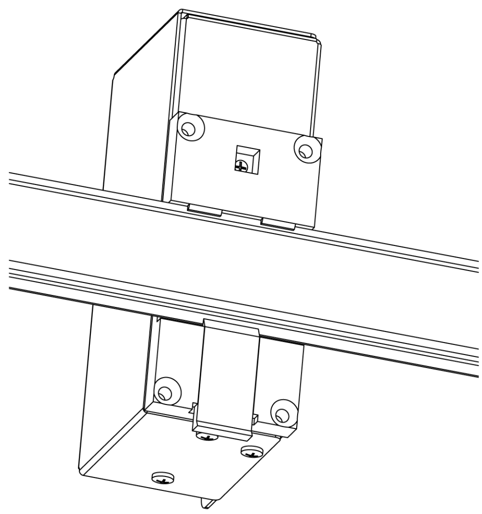


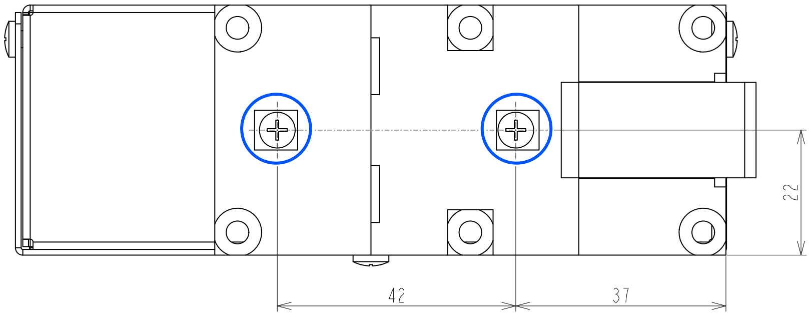


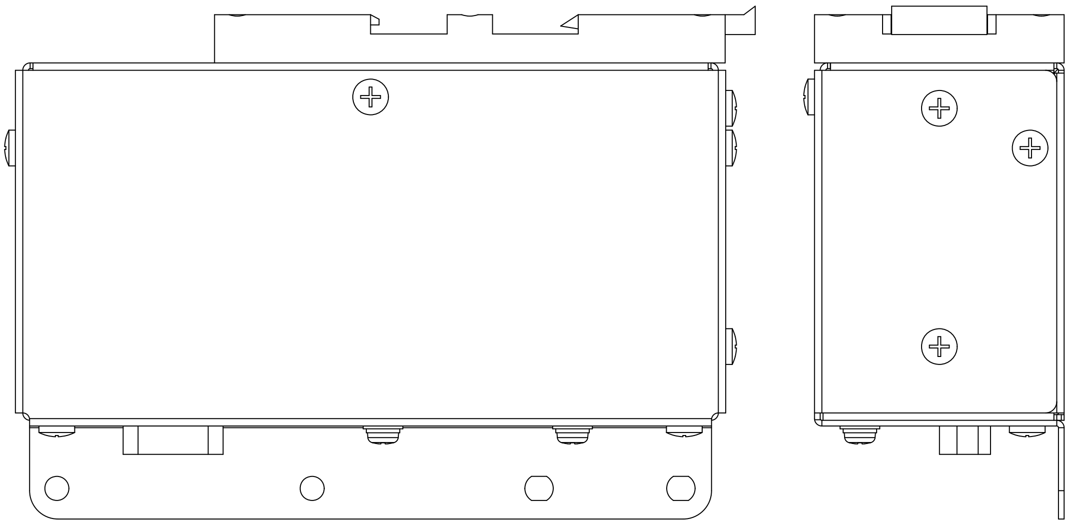
- DIN rail mounting

  Mount a DIN rail mounting bracket on the commercial DIN rail mounting bracket securing holes on the rear of the unit, and then secure the kit to the DIN rail. We do not sell DIN rails or DIN rail mounting brackets. Please use commercially available products.









DIN rail securing bracket that has been confirmed for installation: TAKACHI DRA-1

17.4.2 PG (Pulse Generator) Board

17.4.2.1 PG board specifications

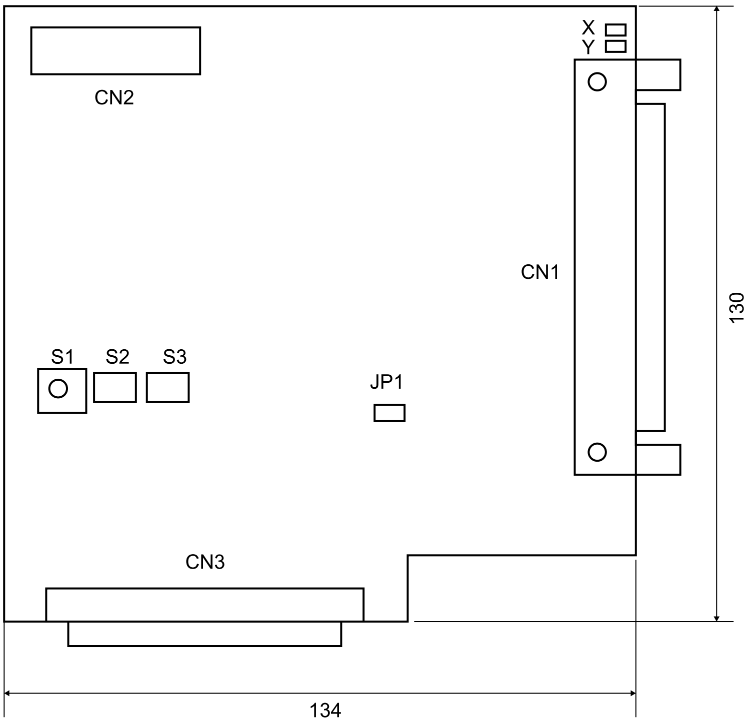
The table below shows the specification for the PG board.

Board Name	H756
Compatible Controller	RC800 Series/ RC700 Series RC90 Series
Encoder channels	4 channels / board
Encoder Type	ABZ phase differential input (RS-422 line receiver)
Input Pulse Rate	Max. 5 MPPS
Input Signal	Conveyor pulse latch input
Board Address	Set the DIP switch according the board number. (See DIP Switch Settings later in this chapter).
connector	DX10A - 100S (Hirose Electric Co. ,Ltd.)
External power supply	24V ±2V 200mA or under

The following rotary encoder models have been tested:

- OMRON E6B2-CWZ1X
- TAMAGAWA TS5312N512-2000C/T

The figure below shows the layout of the PG board.



17.4.2.2 DIP switch settings

The board address is set by DIP switch (S2, S3) on the PG Board according to the board number, as shown in the following table.

Board #	Address	S2				S3			
		1 (A15)	2 (A14)	3 (A13)	4 (A12)	1 (A11)	2 (A10)	3 (A9)	4 (A8)
1	1100h	OFF	OFF	OFF	ON	OFF	OFF	OFF	ON
2	1200h	OFF	OFF	OFF	ON	OFF	OFF	ON	OFF
3	1300h	OFF	OFF	OFF	ON	OFF	OFF	ON	ON
4	1400h	OFF	OFF	OFF	ON	OFF	ON	OFF	OFF

If you purchased the PG board separately, place the attached Board No. Label sticker on the board panel prior to installation of the board in the Control Unit and keep a written record of the address setting and the board number.

If you have purchased the PG Board with the Control Unit, the board address has been set properly before shipment and further settings should not be necessary.

17.4.2.3 Jumper settings

The jumpers are reserved and should not be changed.

17.4.2.4 Rotary switch settings

The rotary switch S1 is reserved and should not be changed.

S1 : Position of 1

17.4.2.5 Signal Connections

The table below lists the connectors on the PG board and the compatible connectors for wiring:

Receptacle on the Board		DXA10A-100S (manufacturer: Hirose Electric Co., Ltd.)
Wiring Plug Connectors	Individually pressed-in type	DX30-100P (for AWG#30), DX30A-100P (for AWG#28)
	Pressed-in-as-a-whole type	DX31-100P (for AWG#30), DX31A-100P (for AWG#28)
	Soldered type	DX40-100P
Connector for Wiring to the Cover		DX-100-CV1

17.4.2.6 Signal Assignments: PG board connector (DX10A-100S)

The signals on the PG board connector are assigned as shown in the table below.

Pin	Dir	Signal	Description	Pin	Dir	Signal	Description
1	-	-	Not used	51	-	-	Not used
2	-	-	Not used	52	-	-	Not used
3	-	-	Not used	53	-	-	Not used
4	-	-	Not used	54	-	-	Not used
5	-	-	Not used	55	-	-	Not used
6	-	-	Not used	56	-	-	Not used
7	-	-	Not used	57	-	-	Not used
8	-	-	Not used	58	-	-	Not used
9	-	-	Not used	59	-	-	Not used
10	In	TRG1	Trigger input for Counter 1	60	-	-	Not used
11	In	TRG2	Trigger input for Counter 2	61	-	-	Not used
12	In	TRG3	Trigger input for Counter 3	62	-	-	Not used
13	In	TRG4	Trigger input for Counter 4	63	-	-	Not used
14	In	EXTV	External power supply for Input circuit	64	In	EXTVGND	External power supply GND for Input circuit
15	In	EXTV	External power supply for Input circuit	65	In	EXTVGND	External power supply GND for Input circuit
16	-	-	Not used	66	-	-	Not used
17	-	-	Not used	67	-	-	Not used
18	-	-	Not used	68	-	-	Not used
19	-	-	Not used	69	-	-	Not used
20	-	-	Not used	70	-	-	Not used

Pin	Dir	Signal	Description	Pin	Dir	Signal	Description
21	-	-	Not used	71	-	-	Not used
22	-	-	Not used	72	-	-	Not used
23	-	-	Not used	73	-	-	Not used
24	-	-	Not used	74	-	-	Not used
25	In	+A1	Phase +A signal for Counter 1	75	In	+A3	Phase +A signal for Counter 3
26	In	-A1	Phase -A signal for Counter 1	76	In	-A3	Phase -A signal for Counter 3
27	In	+B1	Phase +B signal for Counter 1	77	In	+B3	Phase +B signal for Counter 3
28	In	-B1	Phase -B signal for Counter 1	78	In	-B3	Phase -B signal for Counter 3
29	In	+Z1	Phase +Z signal for Counter 1	79	In	+Z3	Phase +Z signal for Counter 3
30	In	-Z1	Phase -Z signal for Counter 1	80	In	-Z3	Phase -Z signal for Counter 3
31	-	-	Not used	81	-	-	Not used
32	-	-	Not used	82	-	-	Not used
33	-	-	Not used	83	-	-	Not used
34	-	-	Not used	84	-	-	Not used
35	-	-	Not used	85	-	-	Not used
36	-	-	Not used	86	-	-	Not used
37	-	-	Not used	87	-	-	Not used
38	-	-	Not used	88	-	-	Not used
39	-	-	Not used	89	-	-	Not used
40	-	-	Not used	90	-	-	Not used
41	In	+A2	Phase +A signal for Counter 2	91	In	+A4	Phase +A signal for Counter 4
42	In	-A2	Phase -A signal for Counter 2	92	In	-A4	Phase -A signal for Counter 4
43	In	+B2	Phase +B signal for Counter 2	93	In	+B4	Phase +B signal for Counter 4
44	In	-B2	Phase -B signal for Counter 2	94	In	-B4	Phase -B signal for Counter 4
45	In	+Z2	Phase +Z signal for Counter 2	95	In	+Z4	Phase +Z signal for Counter 4
46	In	-Z2	Phase -Z signal for Counter 2	96	In	-Z4	Phase -Z signal for Counter 4

Pin	Dir	Signal	Description	Pin	Dir	Signal	Description
47	-	-	Not used	97	-	-	Not used
48	-	-	Not used	98	-	-	Not used
49	-	-	Not used	99	-	-	Not used
50	-	GND	GND	100	-	GND	GND

Pin # 25 to 30, 41 to 46, 75 to 80, 91 to 96

Connect the pin numbers shown above with encoder output (+A, -A, +B, -B, +Z, -Z).

Pins # 10 to 13

When the conveyor pulse is latched by external signal, connect the pin numbers shown above with latch signal.

Exactly when the signal is turned OFF to ON, the encoder pulse is latched.

Pins # 14, 15, 64 and 65

When using the pin # 10 to 13, connect external power with the pin numbers shown above.

When not using the pin # 10 to 13, it is not necessary to connect external power with the pin numbers shown above.

17.4.2.7 Signal Assignments: PG board connector terminal block 1

The signals on PG board connector terminal block 1 are assigned as shown in the table below. The pin numbers in parentheses are the pins on the PG board connector.

Pin	Signal	Description	Pin	Signal	Description
1 (16)	-	Not used	26 (32)	-	Not used
2 (17)	-	Not used	27 (33)	-	Not used
3 (18)	-	Not used	28 (34)	-	Not used
4 (19)	-	Not used	29 (35)	-	Not used
5 (20)	-	Not used	30 (36)	-	Not used
6 (21)	-	Not used	31 (37)	-	Not used
7 (22)	-	Not used	32 (38)	-	Not used
8 (23)	-	Not used	33 (39)	-	Not used
9 (24)	-	Not used	34 (40)	-	Not used
10 (25)	+A1	Phase +A signal for Counter 1	35 (41)	+A2	Phase +A signal for Counter 2
11 (26)	-A1	Phase -A signal for Counter 1	36 (42)	-A2	Phase -A signal for Counter 2
12 (27)	+B1	Phase +B signal for Counter 1	37 (43)	+B2	Phase +B signal for Counter 2
13 (28)	-B1	Phase -B signal for Counter 1	38 (44)	-B2	Phase -B signal for Counter 2
14 (29)	+Z1	Phase +Z signal for Counter 1	39 (45)	+Z2	Phase +Z signal for Counter 2
15 (30)	-Z1	Phase -Z signal for Counter 1	40 (46)	-Z2	Phase -Z signal for Counter 2

Pin	Signal	Description	Pin	Signal	Description
16 (31)	-	Not used	41(47)	-	Not used
17 (48)	-	Not used	42 (49)	-	Not used
18 (9)	-	Not used	43 (50)	GND	GND
19 (60)	-	Not used	44 (61)	-	Not used
20 (10)	TRG1	Trigger input for Counter 1	45 (11)	TRG2	Trigger input for Counter 2
21 (1)	-	Not used	46 (5)	-	Not used
22 (2)	-	Not used	47 (6)	-	Not used
23 (3)	-	Not used	48 (7)	-	Not used
24 (4)	-	Not used	49 (8)	-	Not used
25 (14)	EXTV	External power supply	50 (64)	EXTVGND	External power supply ground

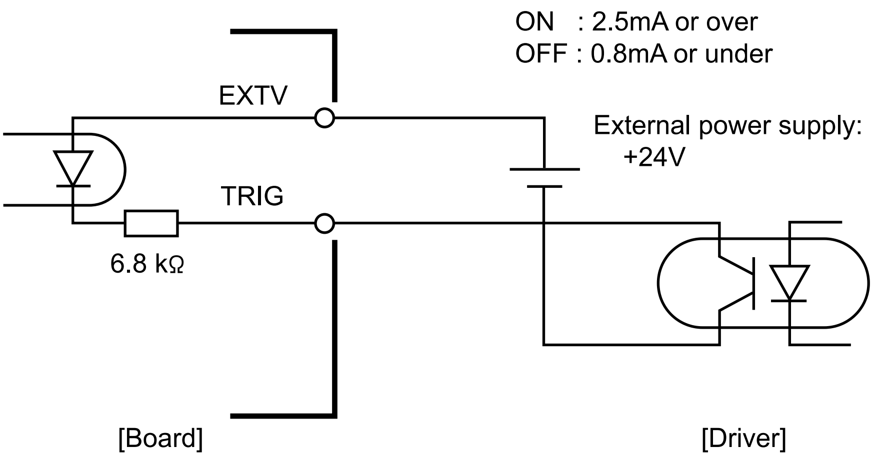
17.4.2.8 Signal Assignments: PG board connector terminal block 2

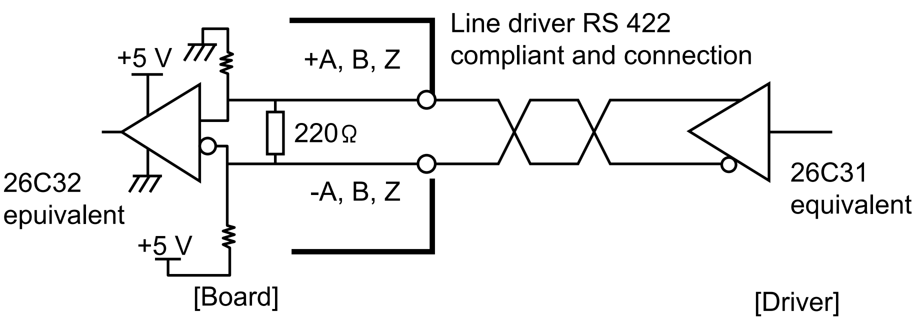
The signals on PG board connector terminal block 2 are assigned as shown in the table below. The pin numbers in parentheses are the pins on the PG board connector.

Pin	Signal	Description	Pin	Signal	Description
1 (66)	-	Not used	26 (82)	-	Not used
2 (67)	-	Not used	27 (83)	-	Not used
3 (68)	-	Not used	28 (84)	-	Not used
4 (69)	-	Not used	29 (85)	-	Not used
5 (70)	-	Not used	30 (86)	-	Not used
6 (71)	-	Not used	31 (87)	-	Not used
7 (72)	-	Not used	32 (88)	-	Not used
8 (73)	-	Not used	33 (89)	-	Not used
9 (74)	-	Not used	34 (90)	-	Not used
10 (75)	+A3	Phase +A signal for Counter 3	35 (91)	+A4	Phase +A signal for Counter 4
11 (76)	-A3	Phase -A signal for Counter 3	36 (92)	-A4	Phase -A signal for Counter 4
12 (77)	+B3	Phase +B signal for Counter 3	37 (93)	+B4	Phase +B signal for Counter 4
13 (78)	-B3	Phase -B signal for Counter 3	38 (94)	-B4	Phase -B signal for Counter 4
14 (79)	+Z3	Phase +Z signal for Counter 3	39 (95)	+Z4	Phase +Z signal for Counter 4
15 (80)	-Z3	Phase -Z signal for Counter 3	40 (96)	-Z4	Phase -Z signal for Counter 4
16 (81)	-	Not used	41 (97)	-	Not used
17 (98)	-	Not used	42 (99)	-	Not used
18 (59)	-	Not used	43 (100)	GND	GND

Pin	Signal	Description	Pin	Signal	Description
19 (62)	-	Not used	44 (63)	-	Not used
20 (12)	TRG3	Trigger input for Counter 3	45 (13)	TRG4	Trigger input for Counter 4
21 (51)	-	Not used	46 (55)	-	Not used
22 (52)	-	Not used	47 (56)	-	Not used
23 (53)	-	Not used	48 (57)	-	Not used
24 (54)	-	Not used	49 (58)	-	Not used
25 (15)	EXTV	External power supply	50 (65)	EXTVGND	External power supply ground

17.4.2.9 Encoder Input Circuit





17.5 Wiring Example of Vision Conveyor Tracking System

Example of Hardware Trigger (Asynchronous Reset Mode)

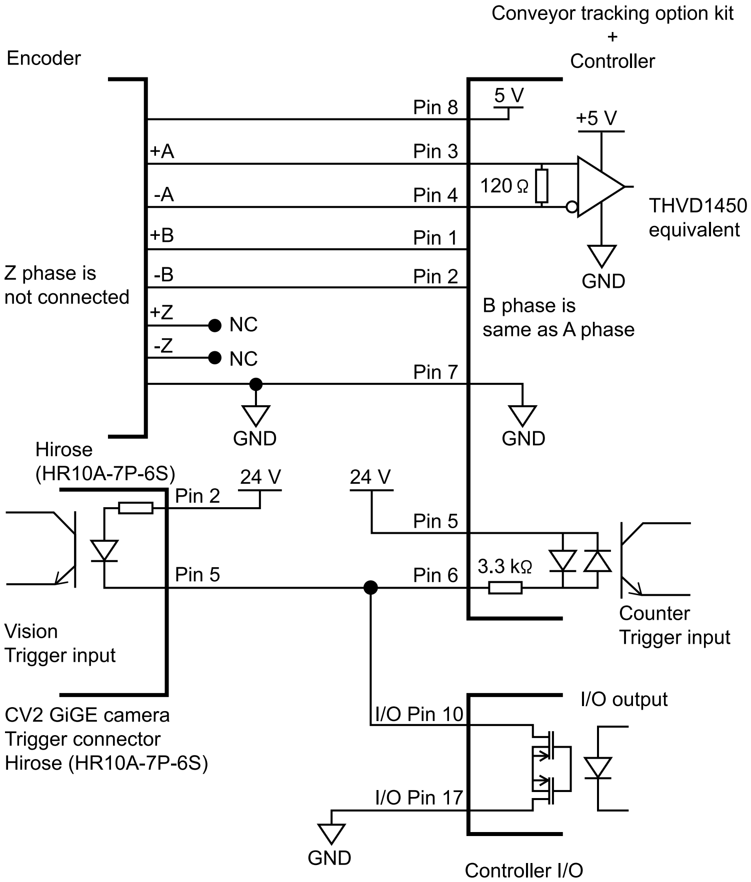
Wire the trigger terminal of the camera and the encoder latch terminal of the PG board to the I/O output of the controller. The PG board detects the trigger when the I/O output turns from OFF to ON. Set the camera so that it also detects the trigger when the I/O output turns from OFF to ON.

For software trigger, it is not necessary to connect the trigger terminal of the camera to the encoder latch terminal of the PG board.

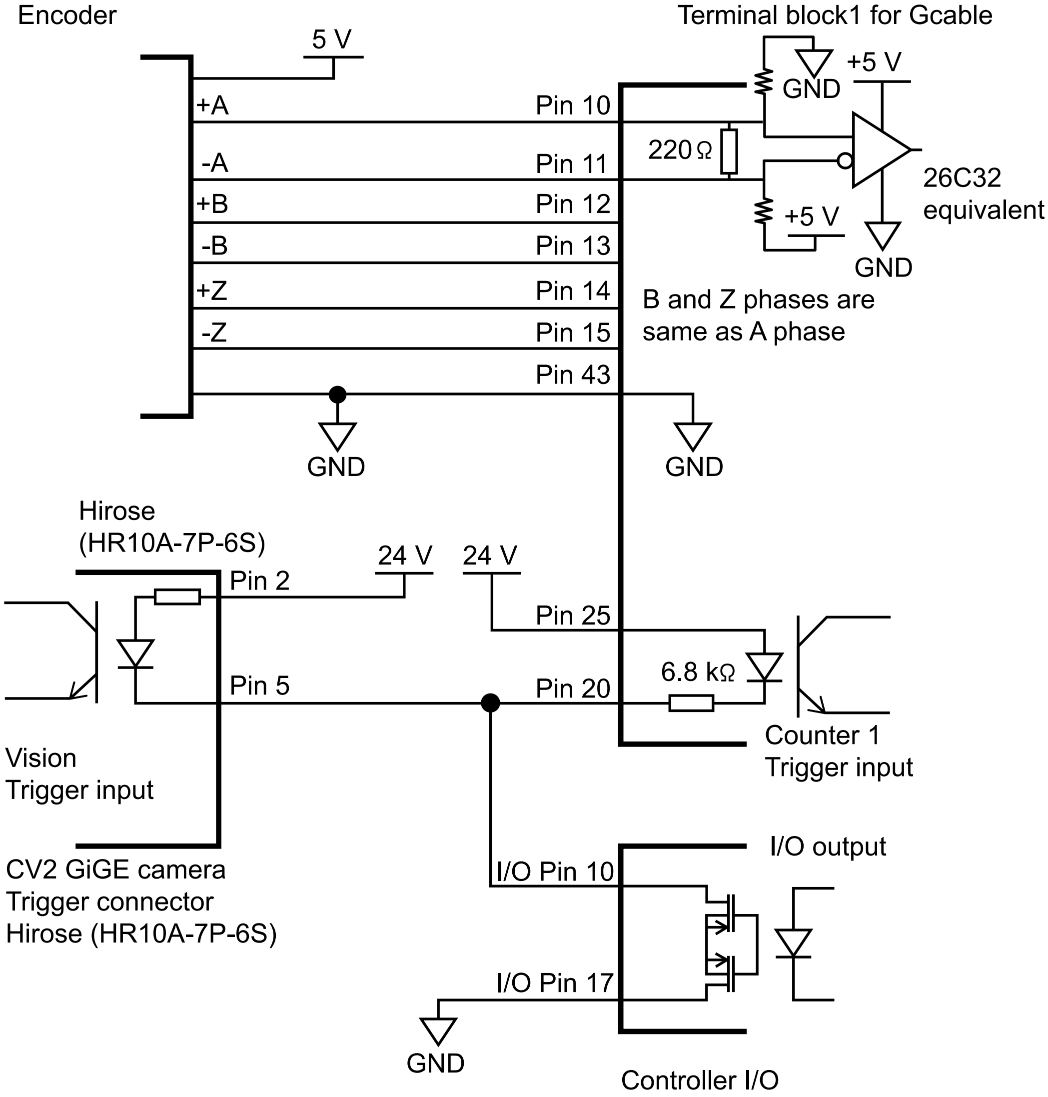
Wiring example of the hardware conveyor trigger

(CV2, Controller I/O output pin10 and Counter 1 are used)

- When using conveyor tracking option kit B



- When using PG board



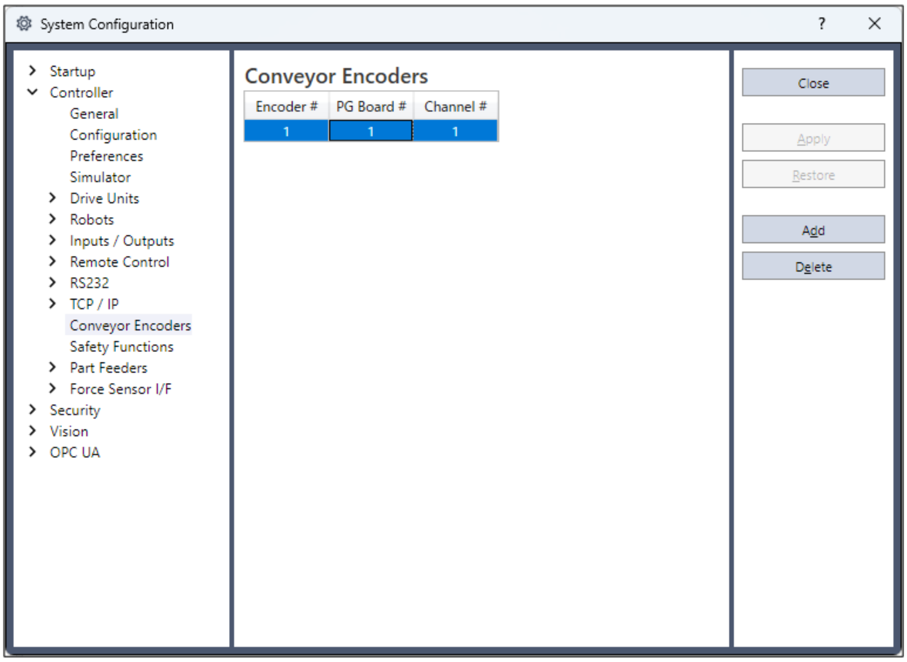
17.6 Conveyor Encoder Configuration

Before you can create any conveyors in a project, you must add conveyor encoders to the system. Each physical conveyor must have an encoder.

Before setting up, wire the encoder to the PULSE IN connector on the control unit. If using a Controller without a PULSE IN connector, install one PG board for every four encoders and wire the encoders to the board(s).

If the encoder is wired using a PG board, set up the conveyor encoder. If the encoder is wired to the PULSE IN connector, no setup is required.

To define system encoders in Epson RC+, select [Setup]-[System Configuration] and select the [Conveyor Encoders].



Click the [Add] button to add an encoder. Encoders are added in the order of Axis number.

You can delete the last encoder in the list. Select it, then click the [Delete] button.

17.7 Verifying Encoder Operation

After wiring the encoders, add them to Epson RC+ as described in the previous section. Next, confirm the operation by following the steps below.

1. Start Epson RC+.

2. Create a new project called “TestCnv”.

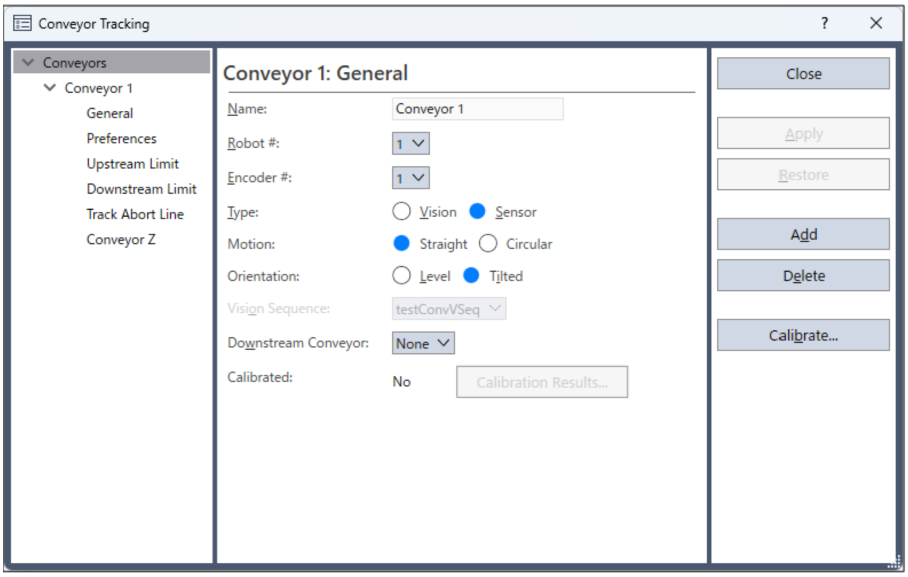
3. Set up the conveyor, using the information below for reference.

   Creating Conveyors in a Project

   - Encoder: Controller, Counter 1
   - Type: Sensor

   Select the encoder according to the actual wiring.

   Make sure to perform the calibration, otherwise the conveyor tracking system cannot work properly. When you only check the encoder operation, it is not necessary to calibrate the conveyors.



4. Now you can use the Cnv_Pulse function to read pulses from Encoder 1 from a program or from the monitor window.

For example, execute this print statement from the monitor window to read the pulses from encoder. Then move the conveyor and execute the command again.

```
>print cnv_pulse(1)
```

You can also use a simple program as shown below. Start the program and move the conveyor. When the conveyor starts moving, the value of Cnv_Pulse will be changed.

```
Function main
  Do
    Print Cnv_Pulse(1)
    Wait .5
  Loop
Fend
```

17.8 Verifying Hardware Conveyor Trigger / Vision Trigger

Verifying the Hardware Conveyor Trigger

1. Move the conveyor and stop it.

2. Check the encoder pulse. Enter the following to the command window.

```
> Print CnvPulse (Conveyor number)
```

3. Turn ON the I/O output number which the trigger is wired to. Latch the encoder pulse.

4. Check the latch pulse. Enter the following to the command window.

```
> Print Cnv_LPulse (Conveyor number)
```

- If the same value as obtained in the Step 2 is latched, the verification is completed.
- If not, check the hardware conveyor trigger wiring.

Verifying the Vision Trigger

1. Set the RuntimeAcquire property of the vision sequence to “Strobed” and the TriggerMode property to “Leading”.

2. Execute the vision sequence and put it into the trigger waiting state.
3. Turn ON the I/O output number which the trigger is wired to and release the shutter.
   - If the captured image is displayed on the VisionGuide window, the verification is completed.
   - If not, check the vision trigger wiring.

17.9 Key Terms

Here explains key terms used in this section.

Queue
: Waiting queue of the FIFO (First-In, First-Out) type for each conveyor.
With the queue, you can register the pose data of workpieces running on the conveyor and user data. When you add data, it will be registered to the end of the queue. When you delete data from the queue, the remaining data in the queue will be moved up automatically.

Queue depth
: The number of data entries registered in a queue.
Maximum number of queue data is each conveyor 1000. The maximum total of queue data for all conveyors is 1000.

Queue user data
: Optional real value that can be registered in a queue.
You can store additional information such as sorted data or part type determined by the image processing.

Downstream Conveyor
: Use this when using multiple conveyors and you run them continuously.
By making an association (upstream/downstream) between conveyors, you can move a queue using the Cnv_QueMove command. “Multiple conveyors” is not necessarily more than one conveyor. You can use one long physical conveyor and set upstream side and downstream side as different logical conveyors. This enables the robots cooperative work, for instance, robot at the downstream side can pick up the workpieces that the robot at upstream fails to pick in time.

Upstream Limit
: Dividing line in the upstream side of the Pickup Area.

Downstream Limit
: Dividing line in the downstream side of the Pickup Area.

Pickup Area
: The area between the upstream limit and downstream limit.
The robot picks parts which flow in the Pickup Area.
The robot starting pickup near the downstream limit continues its operation over the downstream limit. Make sure that the Pickup Area covers the whole robot motion range.
See details below.

Pickup Area

17.10 Conveyor Tracking Commands

All Conveyor Tracking commands begin with the same prefix: “Cnv_”.

Here is a list of all of the commands. For details, refer to Help or the following manual:

"SPEL+ Language Reference"

Command	Description / Usage
Cnv_AbortTrack	Aborts a motion command to a conveyor queue point.
Cnv_Accel function	Sets/ returns acceleration and deceleration of the conveyor.
Cnv_Accel	Sets acceleration and deceleration of the conveyor.

Command	Description / Usage
Cnv_AccelLim	Sets limit of acceleration and deceleration after the conveyor tracked.
Cnv_Accel function	Returns limit of acceleration and deceleration after the conveyor tracked.
Cnv_Adjust	Sets whether to retrieve the tracking delay offset of the conveyor.
Cnv_Accel function	Returns the result of the operation of retrieving the tracking delay offset of the conveyor and the offset.
Cnv_AdjustClear	Clears the tracking delay offset of the conveyor.
Cnv_AdjustSet	Sets the tracking delay offset of the conveyor.
Cnv_Downstream function	Returns the downstream limit for the specified conveyor.
Cnv_Downstream	Sets the downstream limit of the conveyor.
Cnv_Fine function	Returns the setting of the range to judge if the tracking motion is completed or not for the specified conveyor.
Cnv_Fine	Sets / returns the value of Cnv_Fine for one conveyor.
Cnv_Flag function	Returns the tracking state for the tracking abort line.
Cnv_Mode function	Returns the conveyor mode setting value.
Cnv_Mode	Sets the conveyor mode setting value.
Cnv_LPulse function	Returns the pulse latched by a conveyor trigger.
Cnv_Name$ function	Returns the name of the specified conveyor.
Cnv_Number function	Returns the number of a conveyor specified by name.
Cnv_OffsetAngle	Sets the angle offset. Usage: This command is available only for the circular conveyor.
Cnv_OffsetAngle function	Returns the offset angle.
Cnv_Point function	Returns a robot point in the specified conveyor's coordinate system derived from sensor coordinates. Usage: Use this function when registering a point in the queue.
Cnv_PosErr function	Returns deviation in current tracking position compared to tracking target.
Cnv_PosErrOffset	Sets an offset value to correct the deviation in current tracking position compared to tracking target.
Cnv_Pulse function	Returns the current position of a conveyor in pulses.
Cnv_QueAdd	Adds a robot point to a conveyor queue. Usage: Use this function when registering a point in the queue.
Cnv_QueGet function	Returns a point from the specified conveyor's queue. Usage: Use this command for robot tracking motion.
Cnv_QueLen function	Returns the number of items in the specified conveyor's queue. Usage: Use this command to keep the robot waiting until the part (queue) enters the tracking area.
Cnv_QueList	Displays a list of items in the specified conveyor's queue.

Command	Description / Usage
Cnv_QueMove	Moves data from upstream conveyor queue to downstream conveyor queue. Usage: Use this command for the multi conveyor system.
Cnv_QueReject	Sets / displays the minimum distance to prevent the double conveyors register. See details below. **Double Registration Prevention**
Cnv_QueReject function	Sets / returns and displays the queue reject distance for a conveyor.
Cnv_QueRemove	Removes items from a conveyor queue.
Cnv_QueUserData	Sets and displays user data associated with a queue entry.
Cnv_QueUserData function	Sets / returns and displays user data associated with a queue entry.
Cnv_RobotConveyor function	Returns the conveyor being tracked by a robot.
Cnv_Speed function	Returns the current speed of a conveyor.
Cnv_Trigger	Latches current conveyor position for the next Cnv_QueAdd statement. Usage: Use this command when using the software trigger.
Cnv_Upstream function	Returns the upstream limit for the specified conveyor.
Cnv_Upstream	Sets the upstream limit of the conveyor.

KEY POINTS

- To track a part as the conveyor moves, you must use Cnv_QueGet in a motion command statement.

  For example:

  ```
  Jump Cnv_QueGet(1)  'this tracks the part
  ```

  You cannot assign the result from Cnv_Que to a point and then track it by moving to the point.

  ```
  P1 = Cnv_QueGet(1)
  Jump P1             'this does not track the part!
  ```

  When you assign the result from Cnv_QueGet to a point, the coordinate values correspond to the position of the part when the point assignment was executed.

- After pausing the manipulator, the operation command of conveyor tracking cannot be continued. The error 4403 will occur.

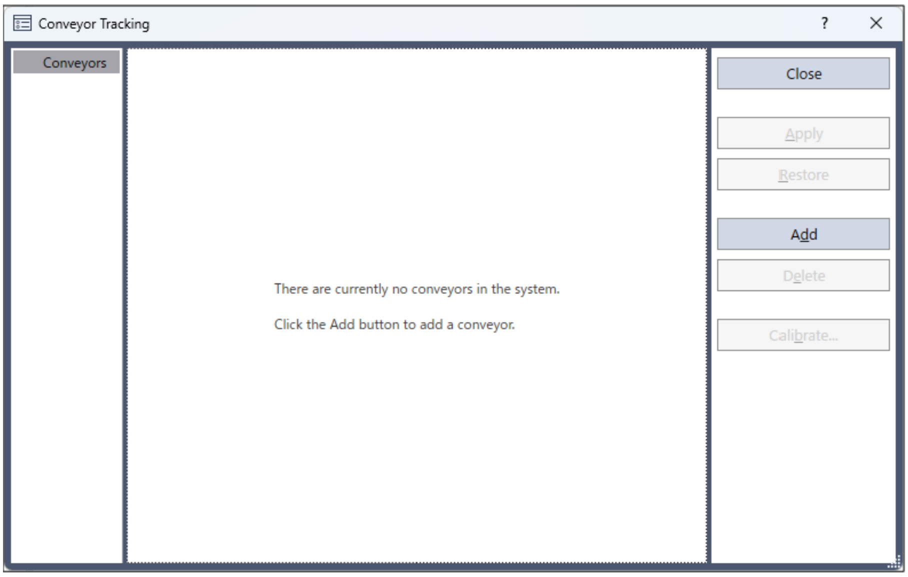
17.11 Creating Conveyors in a Project

Conveyors are configured for each Epson RC+ project. Up to 16 conveyors can be created per project. A conveyor is a logical entity that combines a robot with one or more conveyors.

There are two types of conveyors: vision and sensor. If you will be using a vision camera to find the parts on the conveyor, you must first create a vision sequence to find the parts. This vision sequence is required when you define the conveyor.

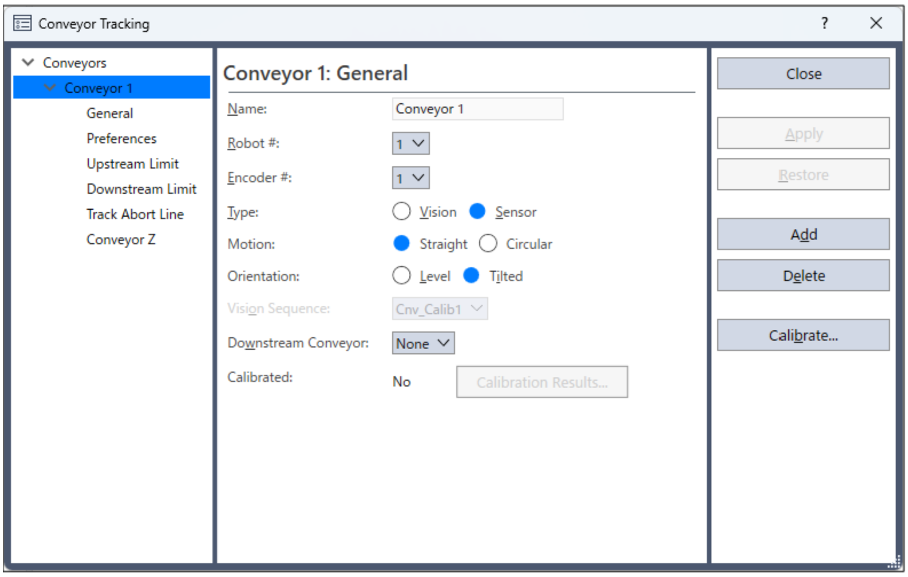
To add a conveyor to a project

1. Select [Tools]-[Conveyor Tracking] to open the [Conveyor Tracking] configuration dialog.



2. To add a conveyor, click the [Add] button.

   The dialog shown below appears.



3. Enter a name for the conveyor, then specify the Robot #, Encoder #, Type, Motion, and Orientation.

KEY POINTS

- A default conveyor name is created automatically when a new conveyor is added. You can change the name as desired.
- When you use a straight conveyor, select "Line" for [Motion].
- When you use a circular conveyor, select "Circular" for [Motion].

17.12 Configuring Conveyors

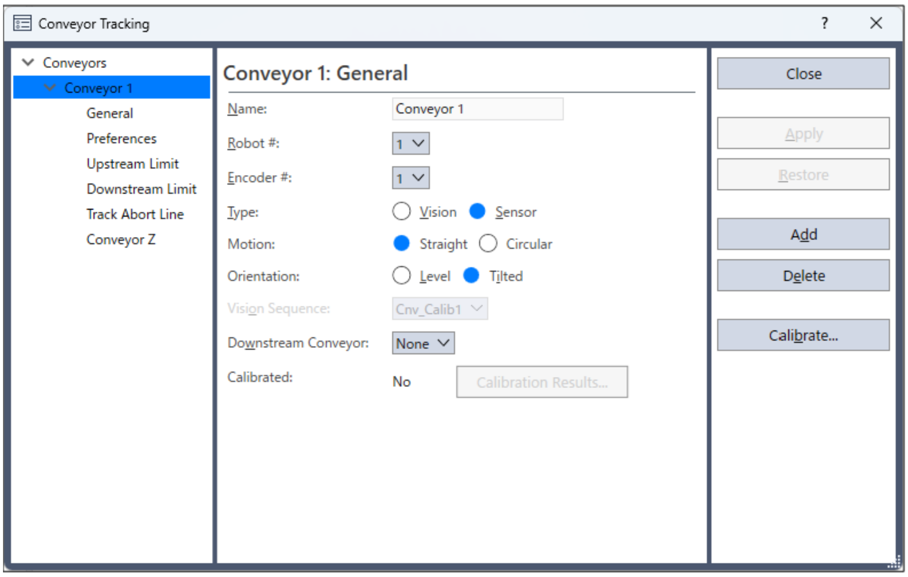
After a conveyor has been created, you can set its parameters.

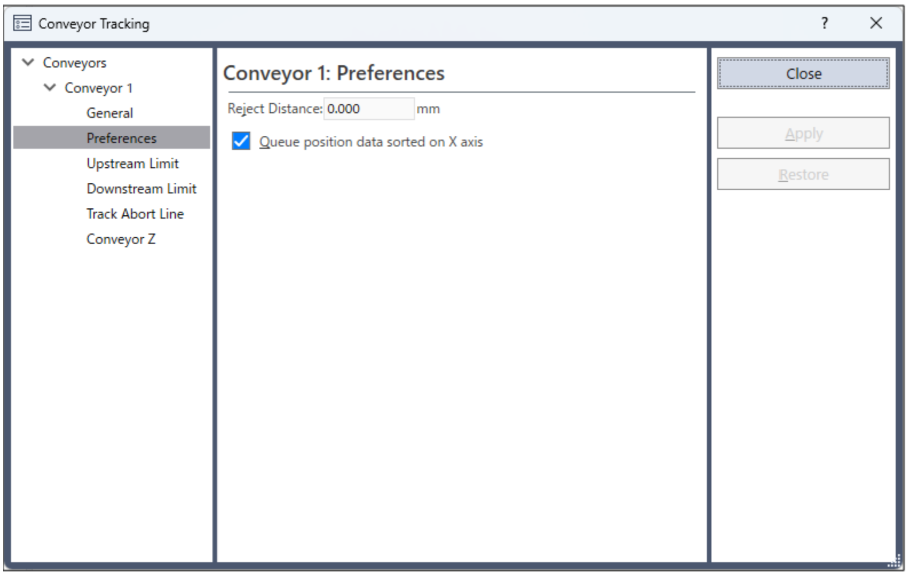
1. Select [Tools]-[Conveyor Tracking].

2. Select the conveyor you want to edit.

3. There are three setup pages shown in the tree under each conveyor: [General], [Preferences], [Upstream Limit], [Downstream Limit], and [Conveyor Z].
   - To change the [Upstream Limit] and [Downstream Limit], refer to 16.16 Pickup Area Changing the Upstream / Downstream limits positions.
   - To change the settings of Reject Distance and queue position data sort, click on [Preferences].
   - To change other parameters, click on [General].

4. Click on [General] or [Preferences]. The dialog shown below appears. Edit any of the configuration options.





5. Click [Apply] to save changes.

KEY POINTS

If you changed Robot #, Encoder #, Orientation, Type, or Vision Sequence, you need to calibrate the conveyor again.

The following table explains the parameters you can edit in the [General] and [Preferences] pages.

Name	You can name conveyors. There is a restriction on the number of characters to be input. Up to 16 bytes
Robot #	You can select a robot that is configured.
Encoder #	You can select an encoder that is configured. Select by encoder number.
Type	You can select vision type and sensor type. ▪ Vision: Detects workpieces using a camera. ▪ Sensor: Detects workpieces using a sensor.
Motion	You can select the conveyor motion; Straight conveyor or Circular conveyor.
Orientation	When you selected Straight conveyor, you can specify if the conveyor is level or tilted. Tilted is set by default. Normally you don't have to change it. ▪ Tilted: Conveyor slope is detected during the calibration. ▪ Level: Conveyor slope is not detected during the calibration. You need to observe the following: ▪ The conveyor must be level with the robot X and Y planes.
Vision Sequence	Select a vision sequence for the calibration. This is only necessary when using the vision type.
Downstream Conveyor	When two or more conveyors have been set, you can select a conveyor number for the downstream conveyor.

Calibrate…	Click this button to execute the calibration. The calibration procedure is different for each type and conveyor orientation.
Adjust Z	After the calibration is completed, you can calibrate the Z coordinate value of the conveyor again.
Reject Distance	You can set distance to prevent the duplicate registration of workpieces on conveyors. ▪ The distance also can be set from the SPEL program using the Cnv_QueReject command. ▪ If the distance is different from the one set by Cnv_QueReject command, the Cnv_QueReject command setting has precedence.
Queue position data sorted on X axis	You can select whether to sort the queue or not.

KEY POINTS

After the calibration, change the parameters for Robot #, Encoder #, Type, Motion, Orientation, Vision Sequence, Adjust Z, and Upstream/Downstream limit.

17.13 Vision Conveyors

A vision conveyor uses a camera to locate parts that will be retrieved by one or more robots. In this section, instructions are provided for vision conveyor calibration.

The straight conveyor and the circular conveyor have different calibration methods.

17.13.1 Vision conveyor camera and lighting

It is important to choose the correct camera and lighting for the vision conveyors used in your application.

For applications with a slow moving conveyor and non-critical pick up constraints, you may be able to use a Vision Guide camera and simple lighting with no strobe.

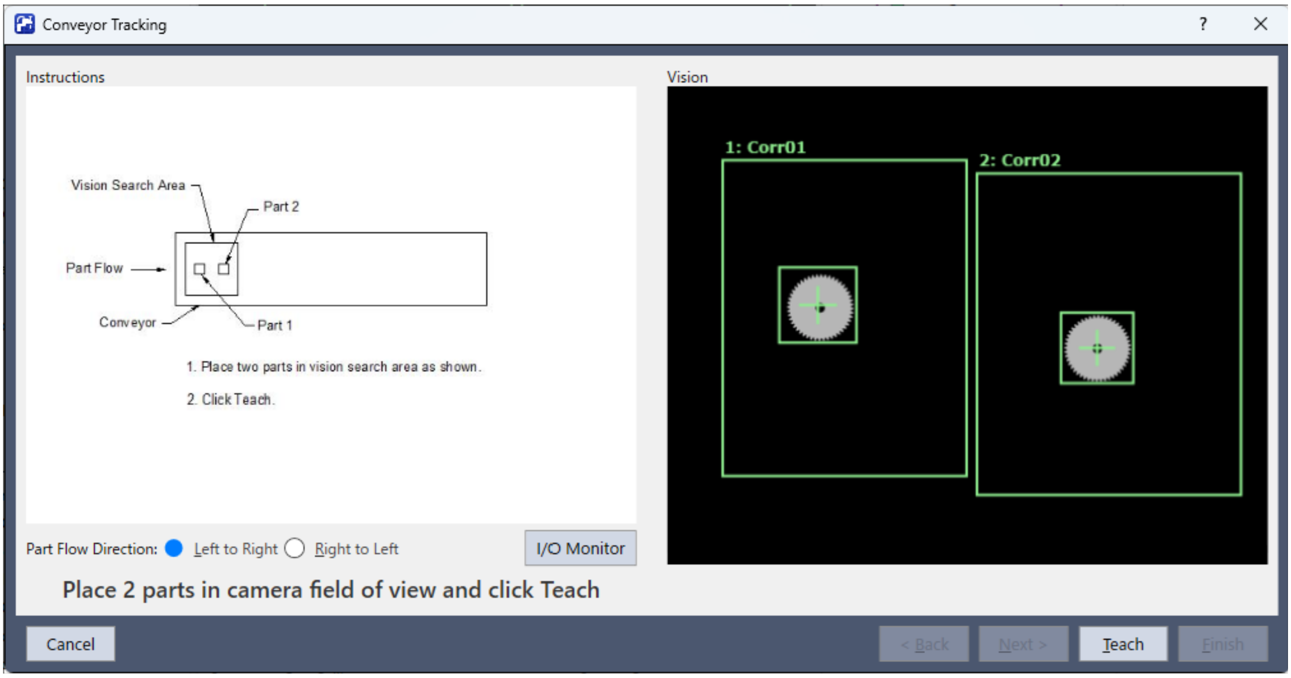
For applications with fast moving parts, you will need to use a camera that is capable of asynchronous reset along with a strobe lamp.

17.13.2 Vision calibration sequence

Before you can calibrate a vision conveyor, you must first create a calibration sequence. This sequence is used by the system during the calibration process and must be linked to a camera calibration. The conveyor system commands use camera coordinates in millimeters. Although you can use any type of Vision Guide camera calibration, you only need to use a Standalone calibration.

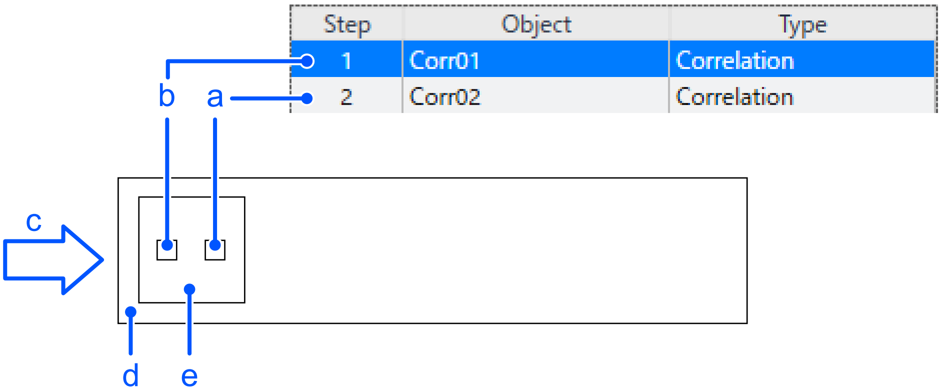
The calibration sequence needs a sequence that uses one object for each workpiece.

Place two workpieces on the conveyor as shown below.



It is recommended to place the two parts diagonally in the field of view. Also, the first object of a sequence must be taught with the robot as Part 1. The second object of a sequence must be taught with the robot as Part 2.

To make it as easy as possible for operators to calibrate the conveyor, the parts that will be found in the vision sequence should be located such that part 2 is after part 1 in the direction of part flow. In the figure below, object 1 in the vision sequence is Corr01, which locates Part 1. Object 2 is Corr02, which locates Part 2.



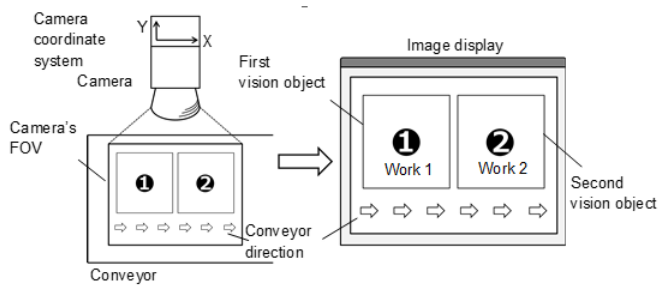
Symbol	Description
a	Work 2
b	Work 1
c	Direction of the conveyor motion
d	Conveyor
e	Vision search area.

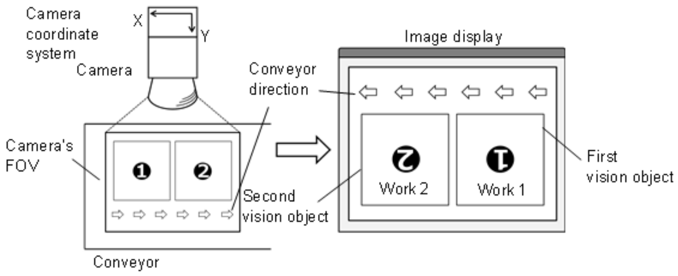
KEY POINTS

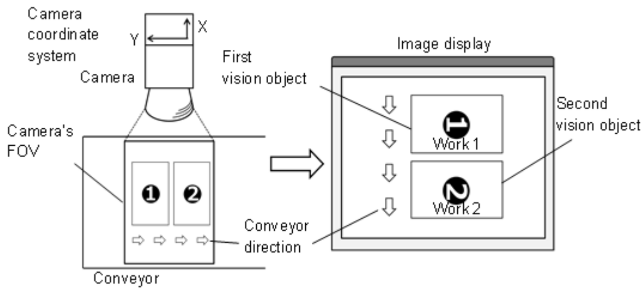
When calibrating the vision conveyor tracking, pay attention to the followings to calibrate properly.

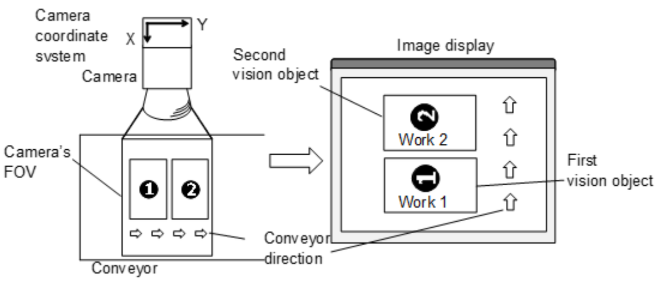
- Check the conveyor direction in the image display.
- In "teaching in the vision search area", place the work 1 on the upstream side and the work 2 on the downstream side.
- Set the objects for detecting the work 1 and 2 in numerical order in the calibration sequence.
- When you place something like letters or patterns around the workpiece to distinguish the workpiece 1 and 2, it will be easier to set the object order of the vision sequence.

The orientation of the camera's FOV displayed in the image display may differ from it actually looks. See the illustrations below. When the camera’s mounting direction is reversed, you need to pay attention to the positional relation of the workpieces and the vision objects.









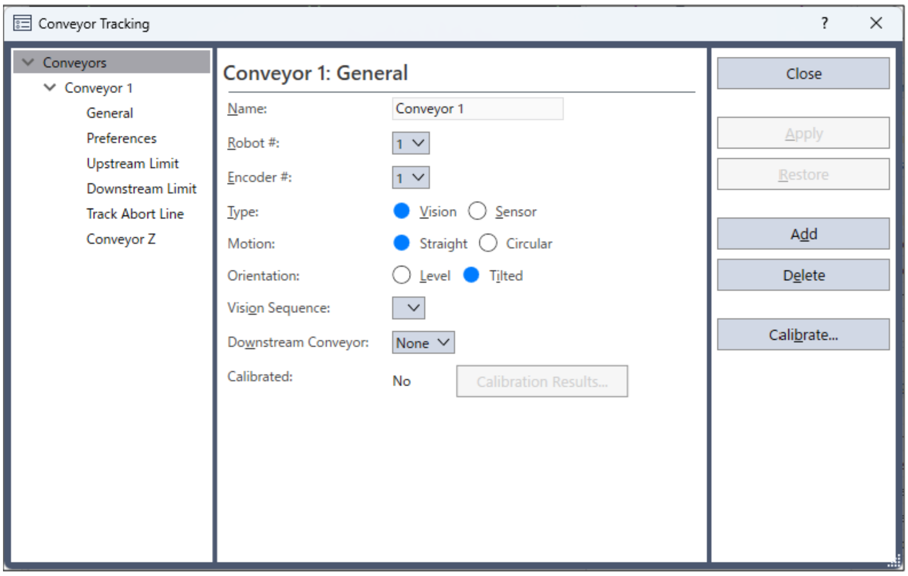
17.13.3 Vision conveyor calibration (Straight conveyor)

Follow these steps to calibrate a straight vision conveyor:

KEY POINTS

- When teaching part positions with the robot during calibration, it is important to position X, Y, and Z of each point accurately. The conveyor is calibrated in X, Y, Z, U, V, and W.
- To perform the fine calibration, in the step 15 and 17, set as wide distance as possible between the upstream limit and downstream limit. After the calibration, adjust the Pickup Area by resetting the upstream / downstream limits.
- For the level orientation, it determines the conveyor height with the position of robot end effector taught in the step 12. It cannot be used for the tilted conveyor for it does not detect the conveyor slope. The steps 19 to 20 are not displayed.
- For the tilted orientation, it calibrates the conveyor slope with the position of robot end effector taught in the steps 12, 14, 16, 18, and 20.

1. Select [Tools]-[Conveyor Tracking].
2. Select the conveyor you want to calibrate.
3. Select [Vision] for the [Type].

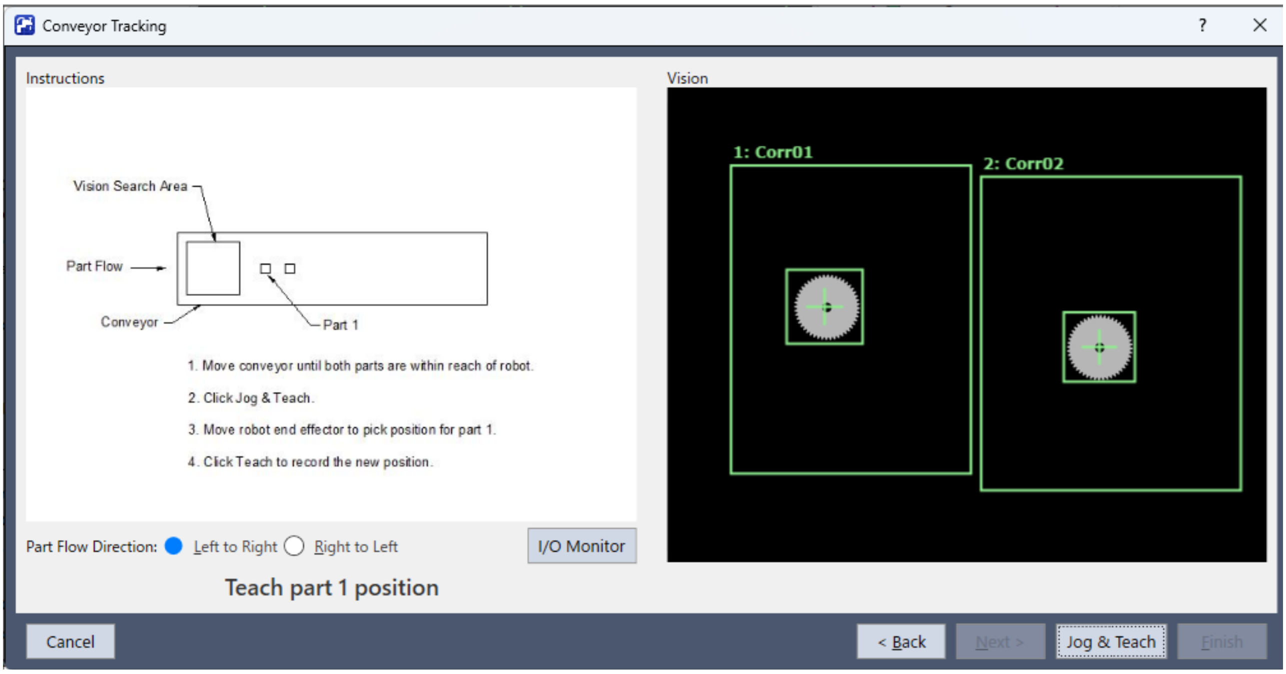


4. Set the [Vision Sequence].
5. Click the [Apply] button.
6. Click the [Calibrate] button. The [Conveyor Tracking Calibration] wizard is displayed. Follow the instructions for each step. Before you can proceed to the next step, you must click the [Teach] button. You can go back to previous steps using the [Back] button.
7. Select the [Part Flow Direction] to best match the conveyor you are calibrating. The instruction pictures will change according to the setting. [Part Flow Direction] is only used to aid in the instructions. It has no effect on the calibration.
8. Place two parts on the conveyor as shown in the figure in the wizard.
9. Check the live video in [Vision]. The camera orientation may not be the same as the picture.

10. Arrange the parts to be inside the range correctly and click the [Teach] button.

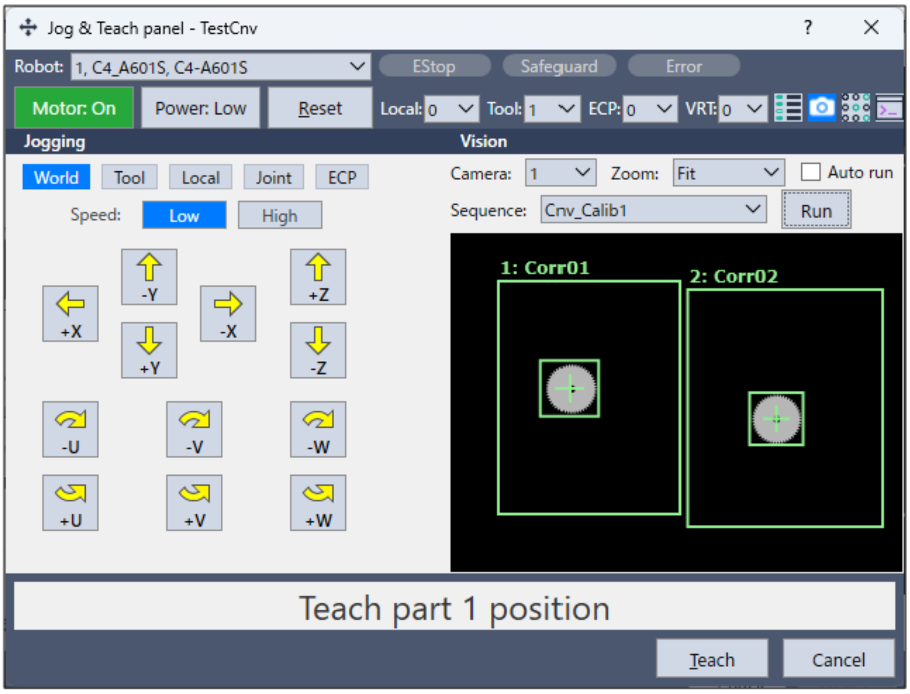
11. Move the conveyor until both parts are within reach of the robot. Do not move the parts, only the conveyor. Proper calibration cannot be performed unless the encoder pulse count value changes according to the workpiece position.

    Click the [Jog & Teach] button.

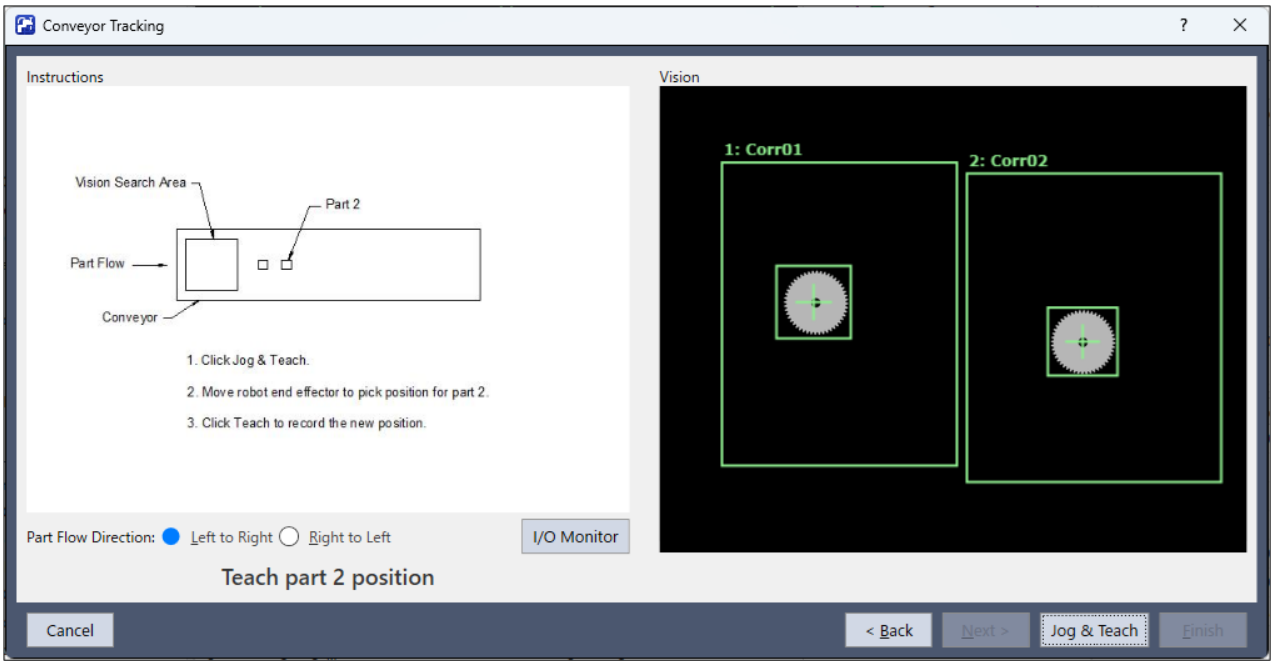


12. The [Jog & Teach panel] will appear. Click the jog buttons to move the robot end effector to the pick position for Part 1.

    Click the [Teach] button.

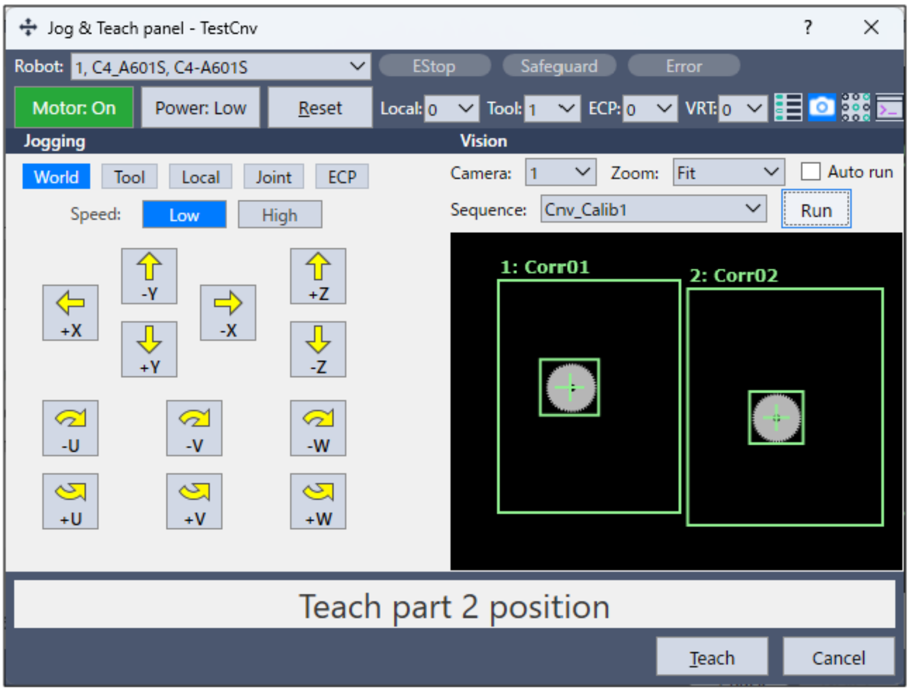


13. Click the [Jog & Teach] button.



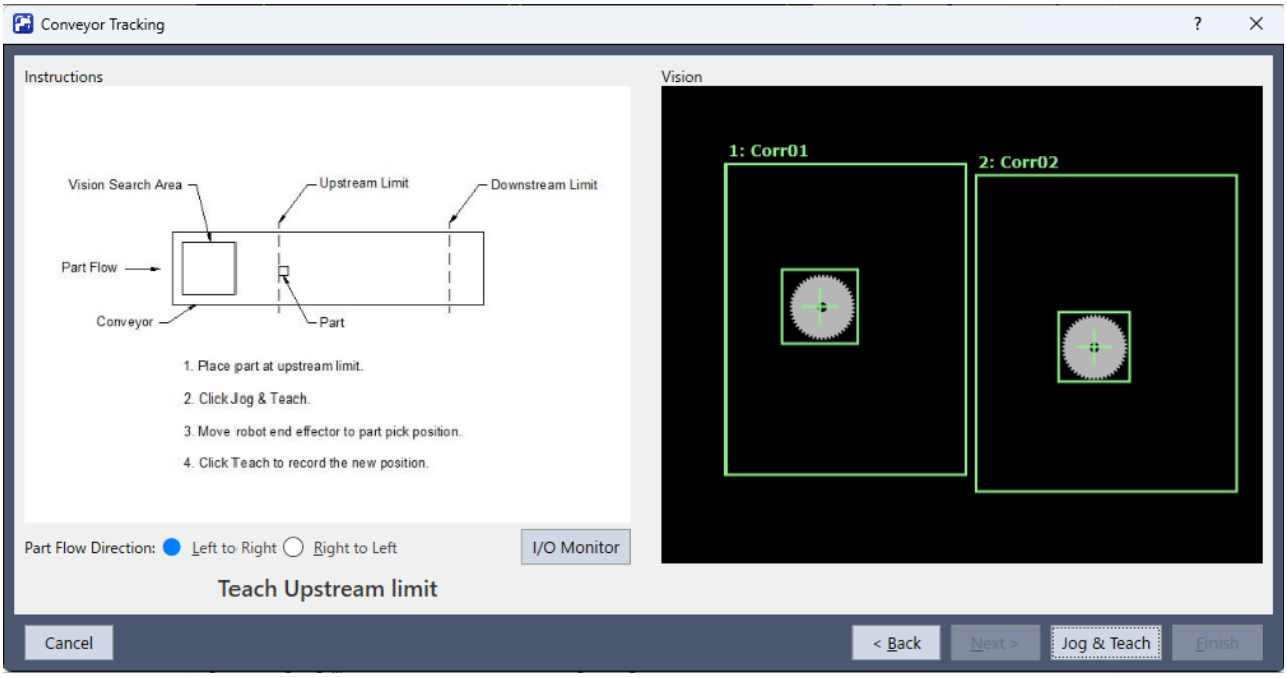
14. The [Jog & Teach panel] will appear. Click the jog buttons to move the robot end effector to the pick position for Part 2.

    Click the [Teach] button.



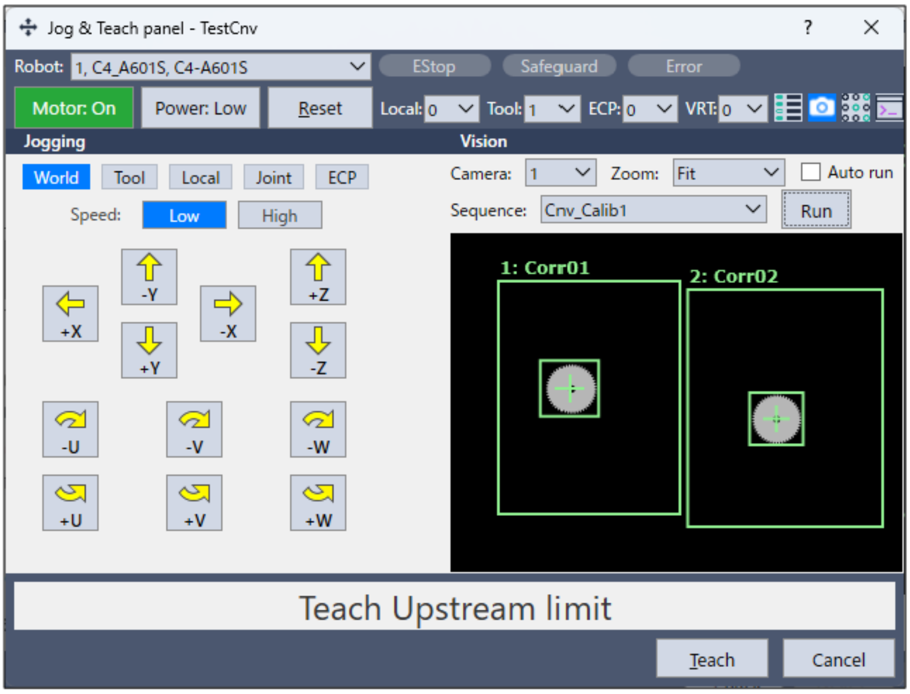
15. Now move or place the part at the upstream limit. The [Vision] image has not been updated from step 13. You can execute one work like [Instructions].

    Click the [Jog & Teach] button.



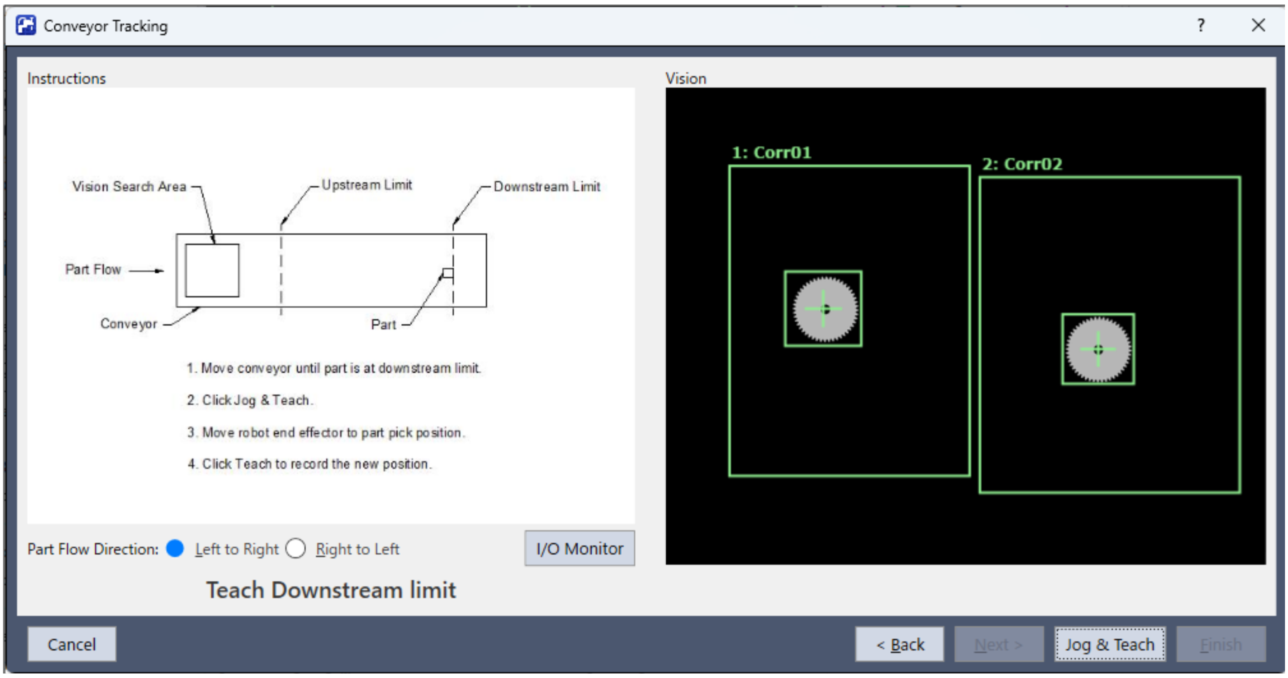
16. The [Jog & Teach panel] will appear. Click the jog buttons to move the robot end effector above the part.

 Click the [Teach] button.



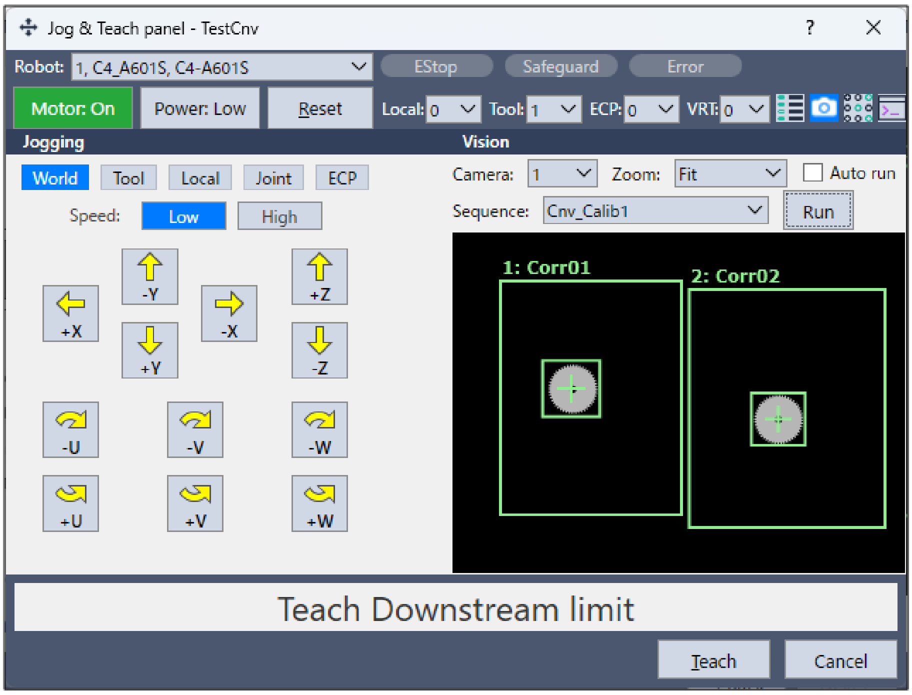
17. Move the conveyor so the part is at the downstream limit. Do not move the parts, only the conveyor. Proper calibration cannot be performed unless the encoder pulse count value changes according to the workpiece position.

 Click the [Jog & Teach] button.



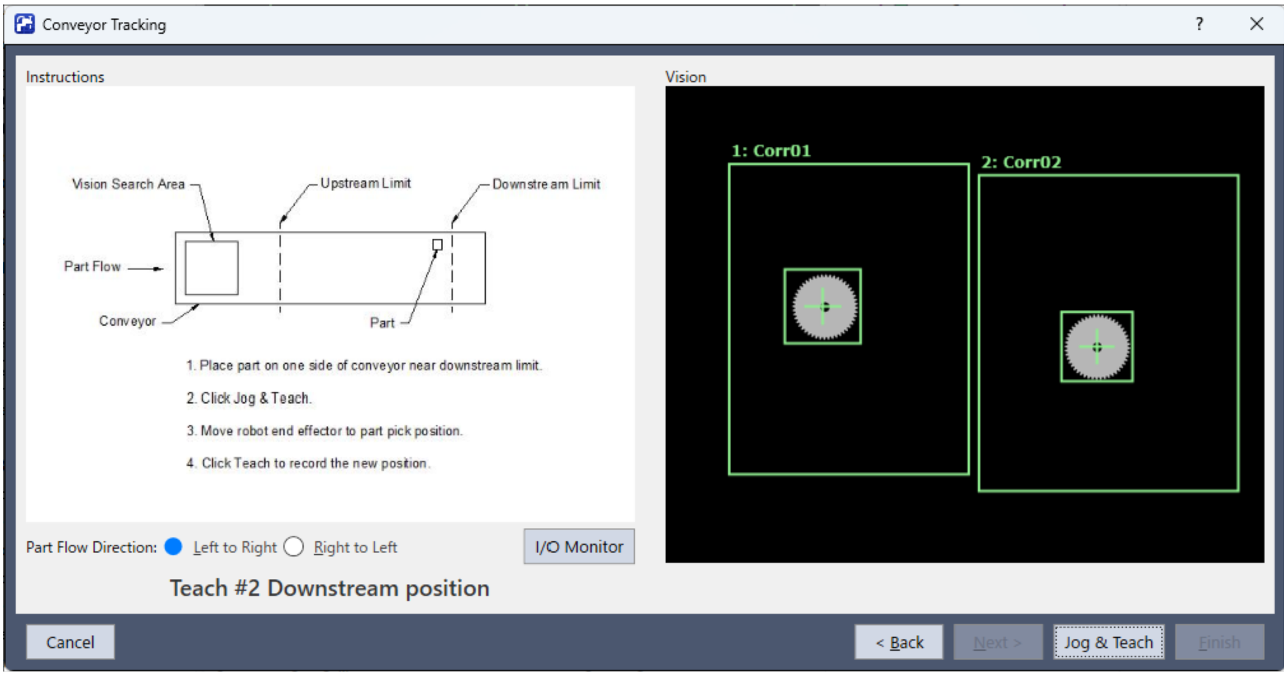
18. The [Jog & Teach panel] will appear. Click the jog buttons to move the robot end effector above the part.

    Click the [Teach] button.



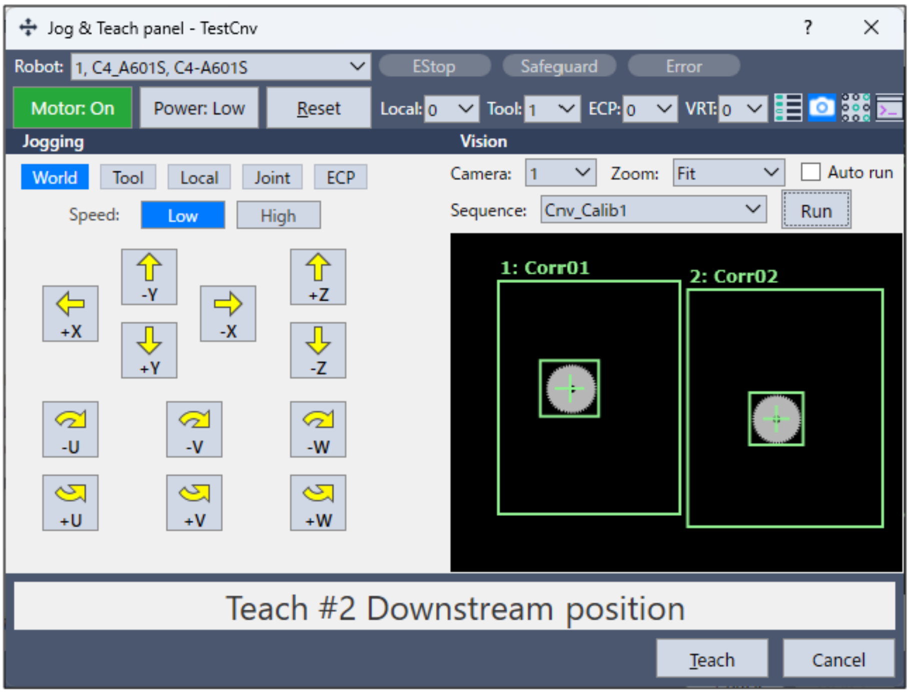
19. Place a part on one side of the conveyor near the downstream limit. This point is used to determine the tilt of the conveyor from side to side.

    Click the [Jog & Teach] button.



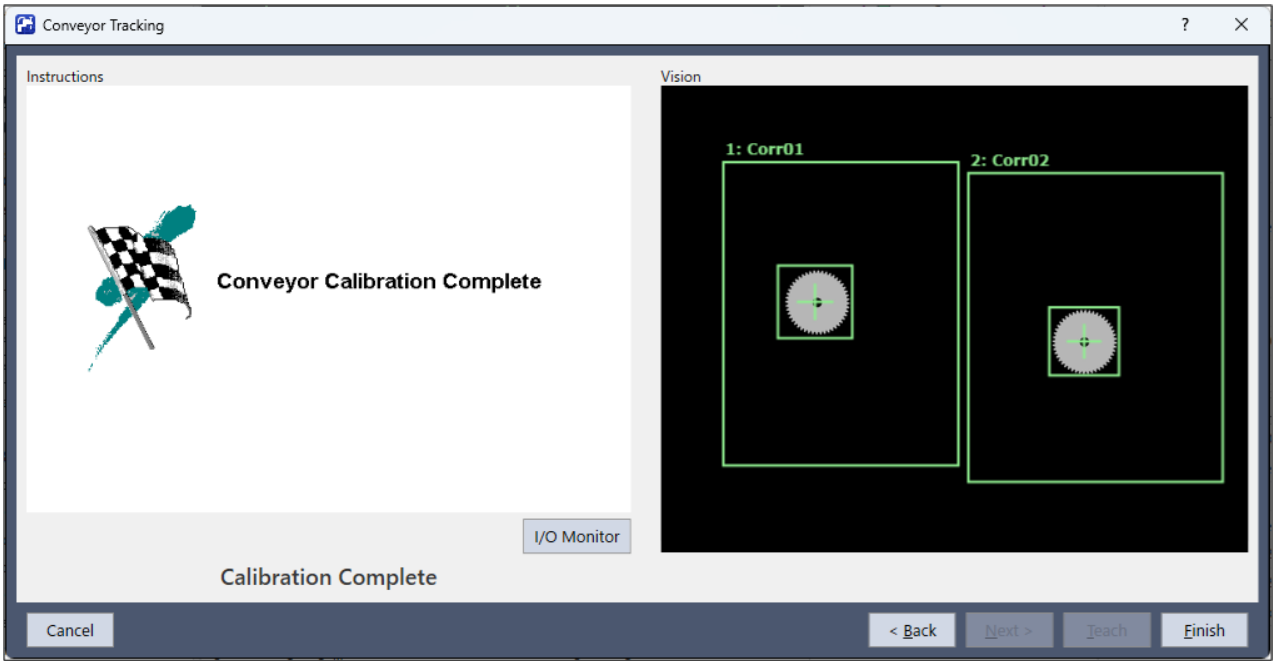
20. The [Jog & Teach panel] will appear. Click the jog buttons to move the robot end effector to the pick position.

    Click the [Teach] button.

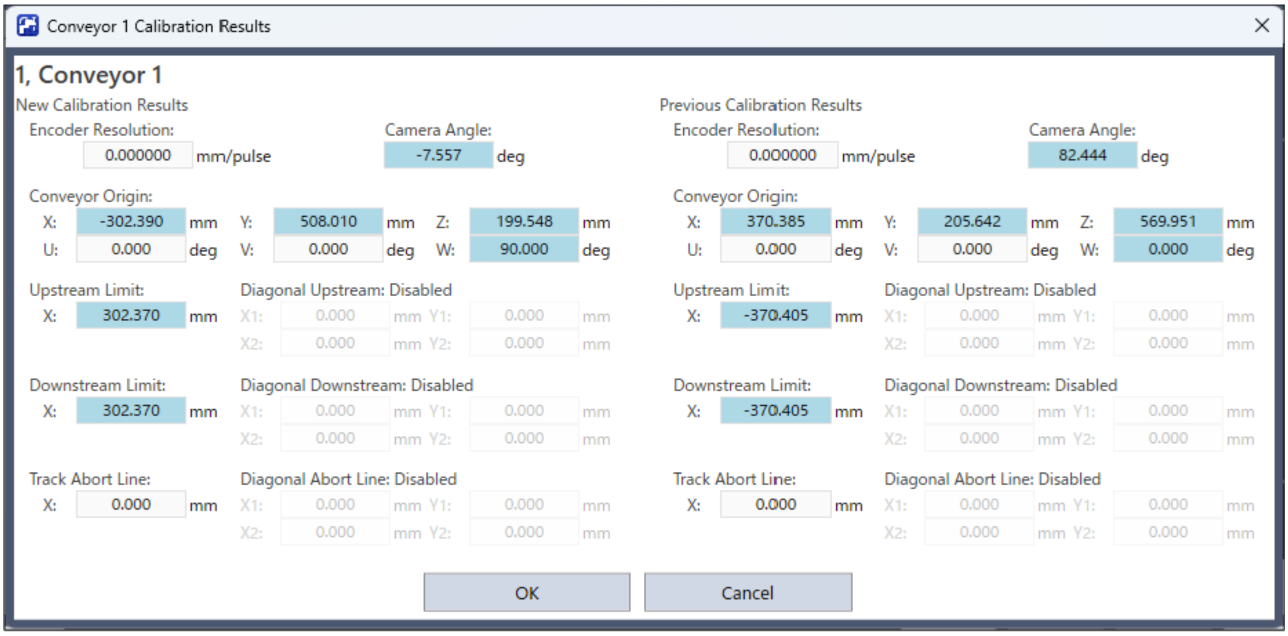


21. The calibration complete picture will be displayed.

    Click the [Finish] button.



22. The Calibration Results screen is displayed.
    - Click the [OK] button to complete the calibration.
    - Click the [Cancel] button to return to the Calibration Complete screen in step 21.



17.13.4 Vision conveyor calibration (Circular conveyor)

Follow these steps to calibrate a circular vision conveyor:

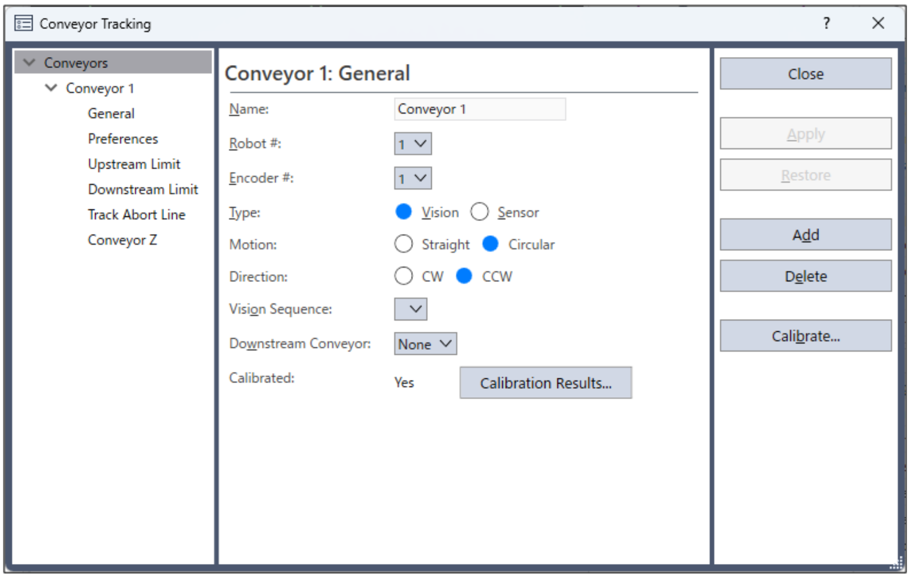
KEY POINTS

- When teaching part positions with the robot during calibration, it is important to position X, Y, and Z of each point accurately. The conveyor is calibrated in X, Y, Z, U, V, and W.
- To perform the fine calibration, in steps 13, 17, and 19, teach the position when the robot is directly above the parts 1 and set as wide a distance as possible between the points to be taught.

1. Select [Tools]-[Conveyor Tracking].
2. Select the conveyor you want to calibrate.
3. Select [Vision] for the [Type].
4. Select [Circular] for the [Motion].
5. Select the conveyor rotating direction for the [Direction].

KEY POINTS

Be careful not to calibrate with a wrong direction, otherwise, the robot will not track the parts.

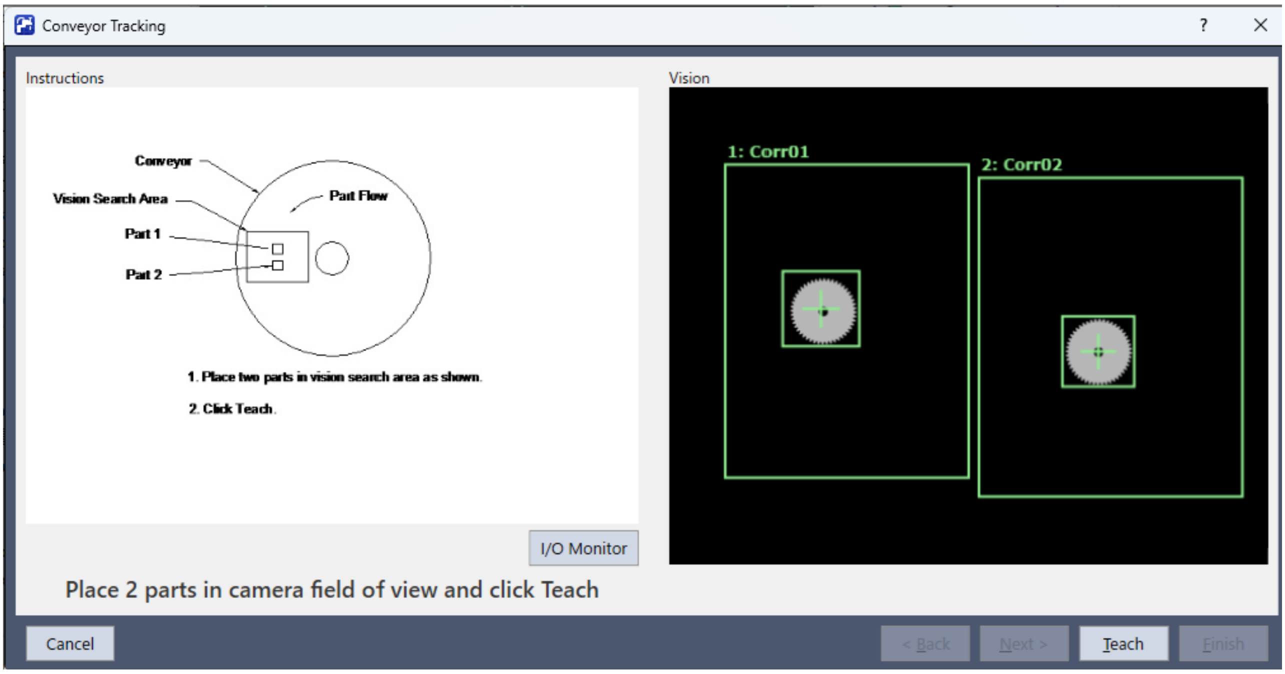


6. Set the [Vision Sequence].
7. Click the [Apply] button.
8. Click the [Calibrate] button.

   The [Conveyor Tracking Calibration] wizard is displayed.

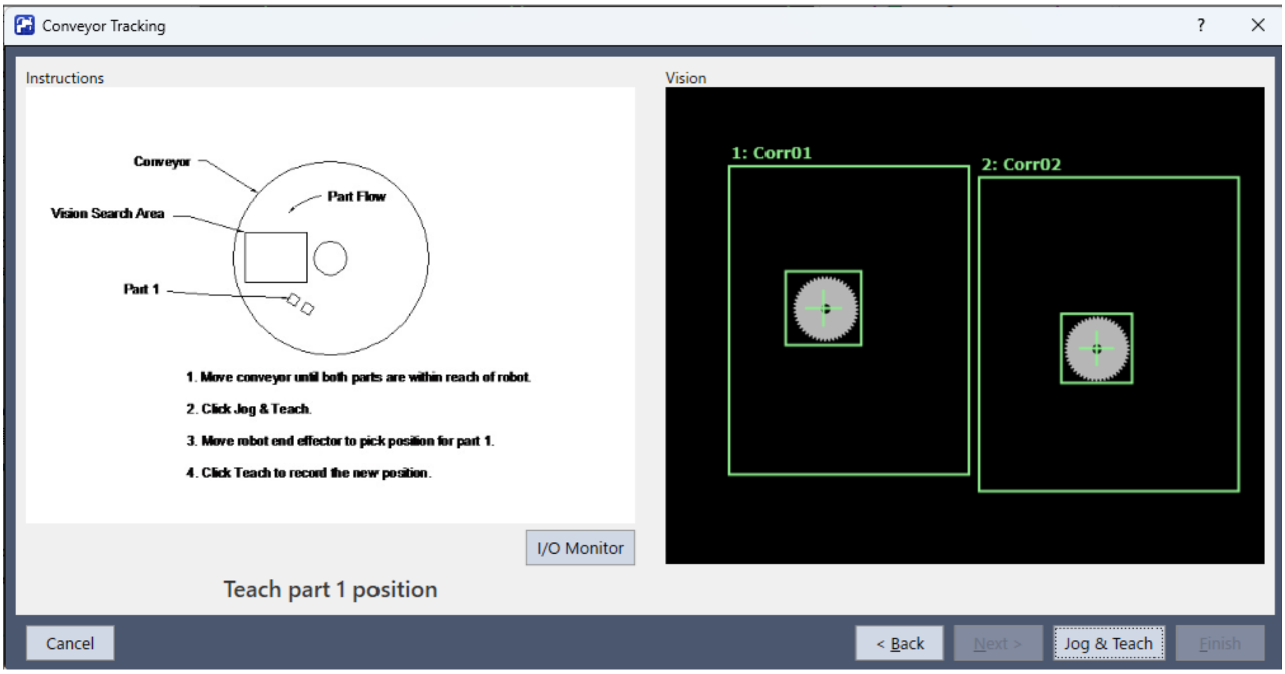
   Follow the instructions for each step. Before you can proceed to the next step, you must click the [Teach] button. You can go back to previous steps using the [Back] button.

9. Check if the conveyor direction shown in the wizard is the same as the conveyor you want to use.
10. Place two parts on the conveyor as shown in the figure in the wizard.
11. Select the [Vision] tab to see live video. The camera orientation may not be the same as the picture.
12. Arrange the parts to be inside the range correctly and click the [Teach] button.



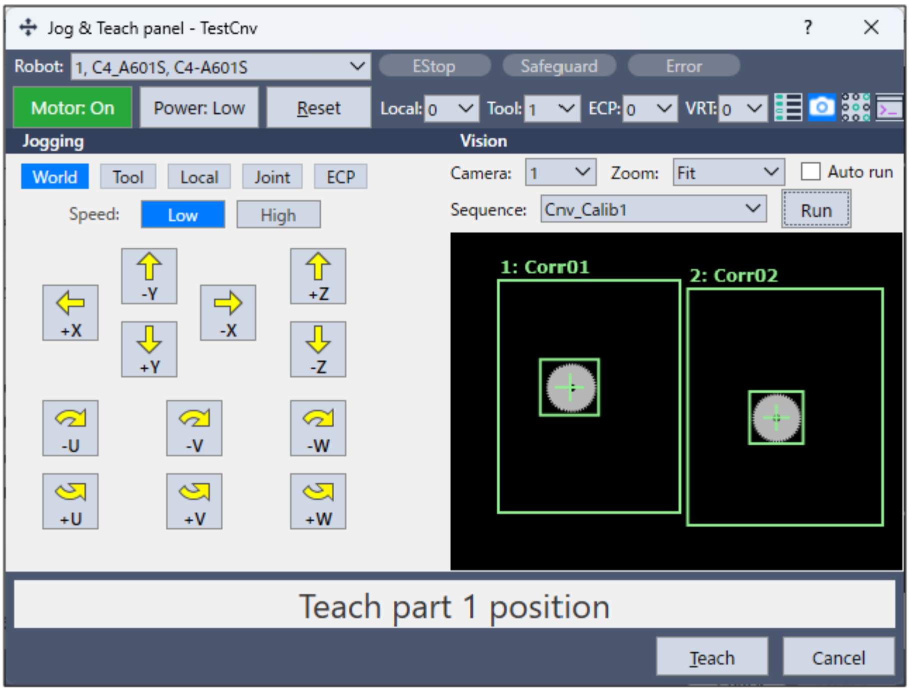
13. Move the conveyor until both parts are within reach of the robot. Do not move the parts, only the conveyor. Proper calibration cannot be performed unless the encoder pulse count value changes according to the workpiece position.

    Click the [Jog & Teach] button.

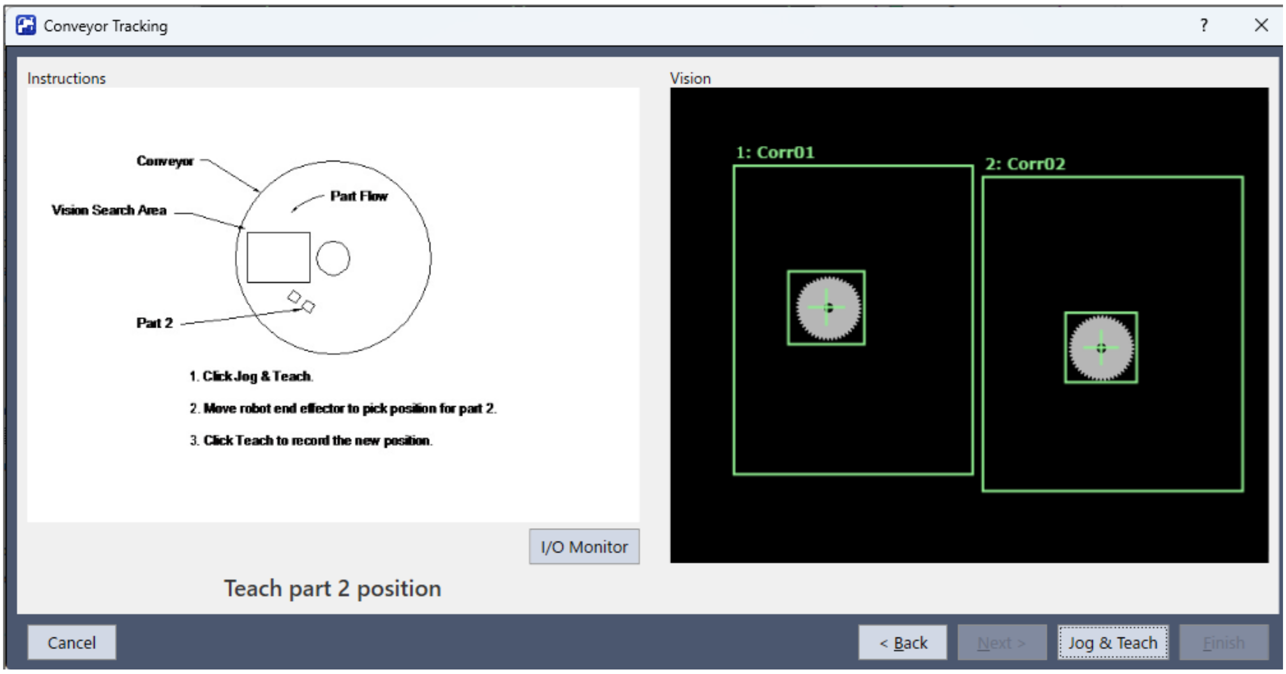


14. The [Jog & Teach panel] will appear. Click the jog buttons to move the robot end effector to the pick position for Part 1.

    Click the [Teach] button.

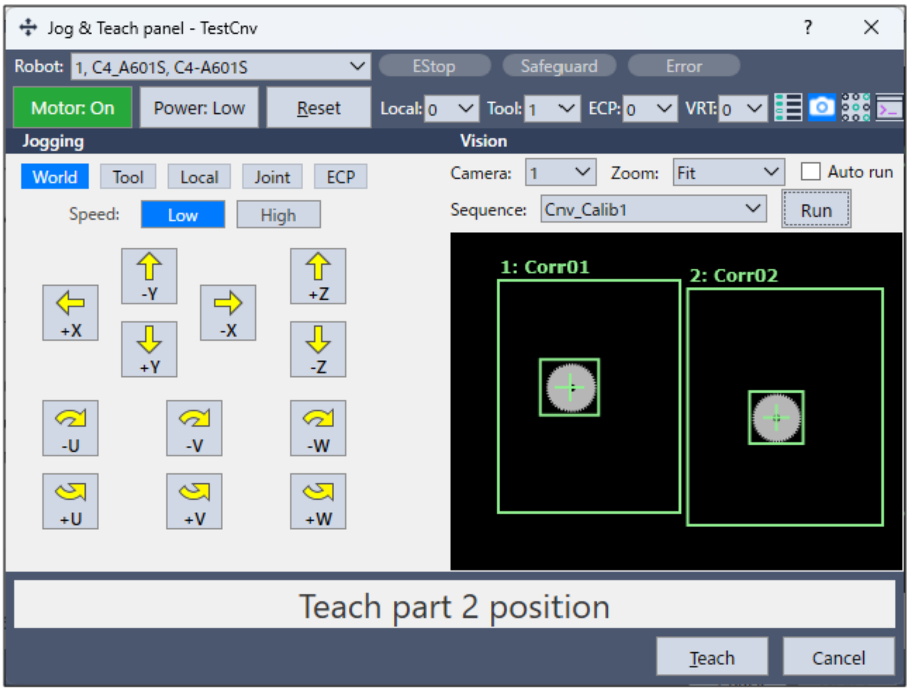


15. Click the [Jog & Teach] button.



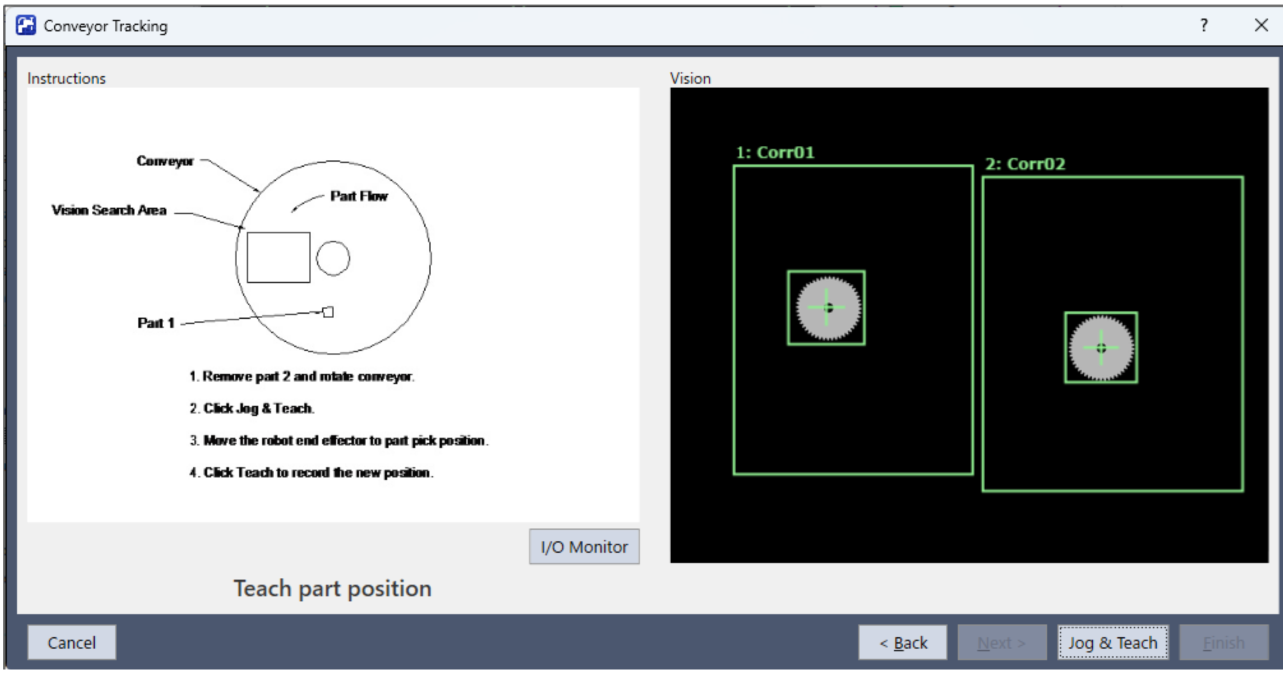
16. The [Jog & Teach panel] will appear. Click the jog buttons to move the robot end effector to the pick position for Part 2.

 Click the [Teach] button.



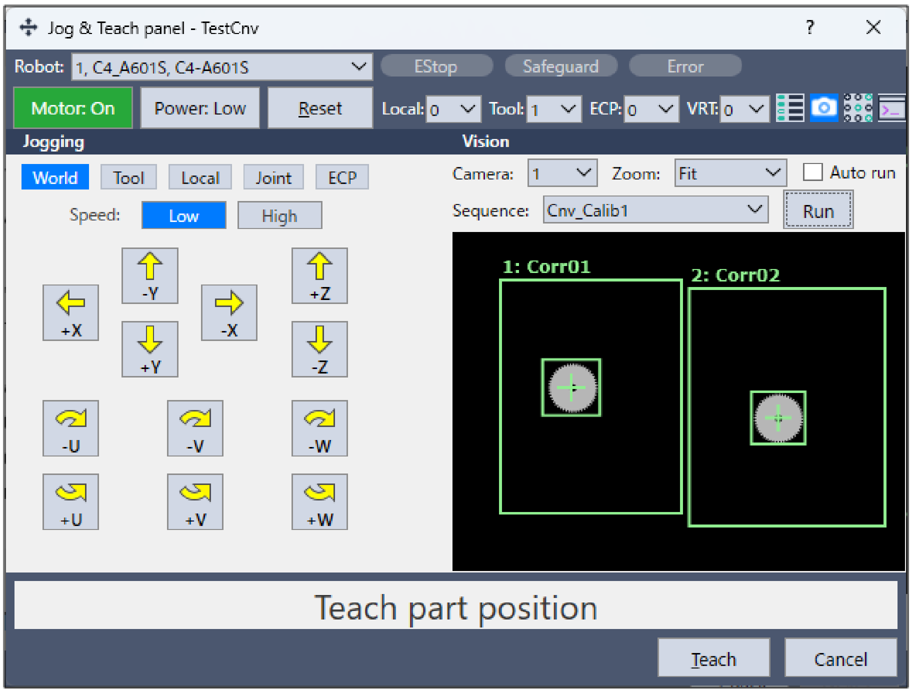
17. Remove Part 2. Move the conveyor to move Part 1.

Click the [Jog & Teach] button.



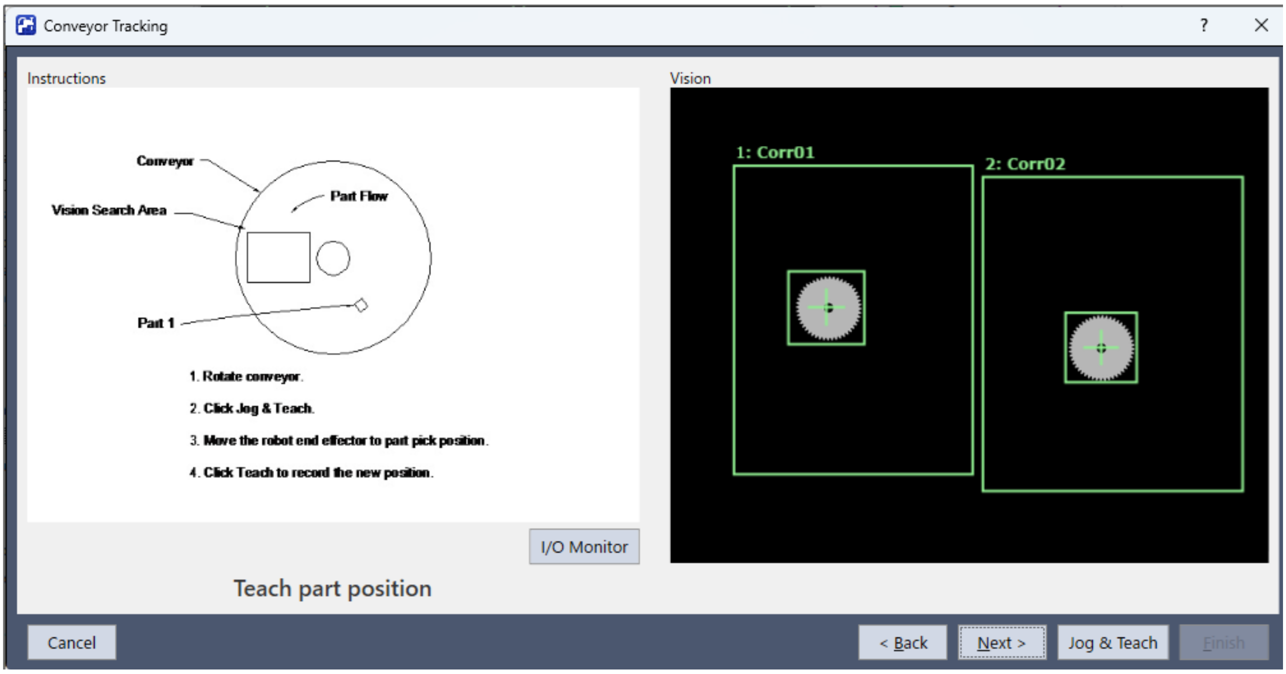
18. The [Jog & Teach panel] will appear. Click the jog buttons to move the robot end effector to the pick position.

Click the [Teach] button.



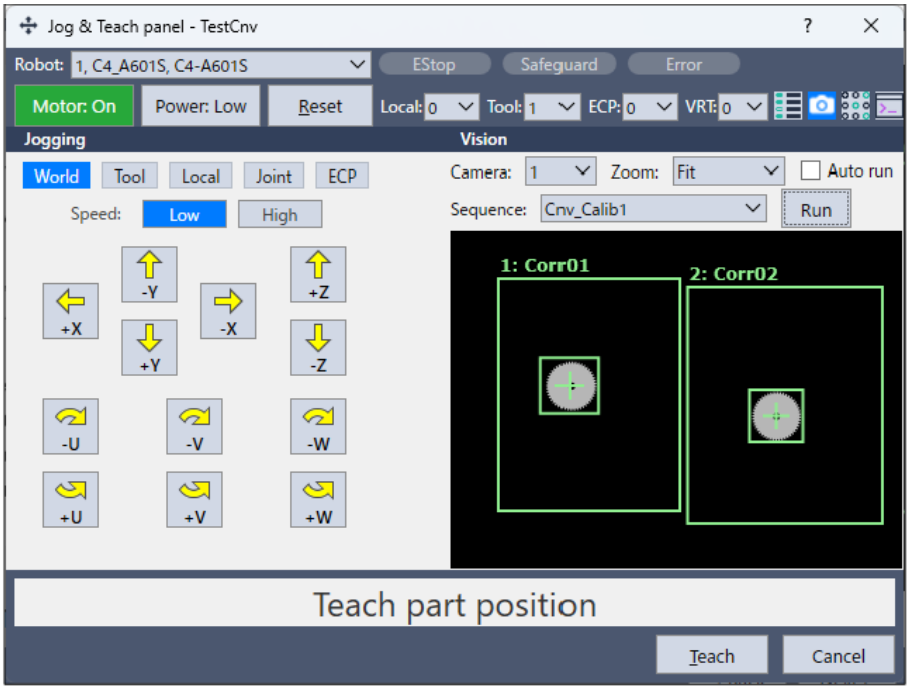
19. Move the conveyor by hand to move Part 1. Proper calibration cannot be performed unless the encoder pulse count value changes according to the workpiece position.

    Click the [Jog & Teach] button.



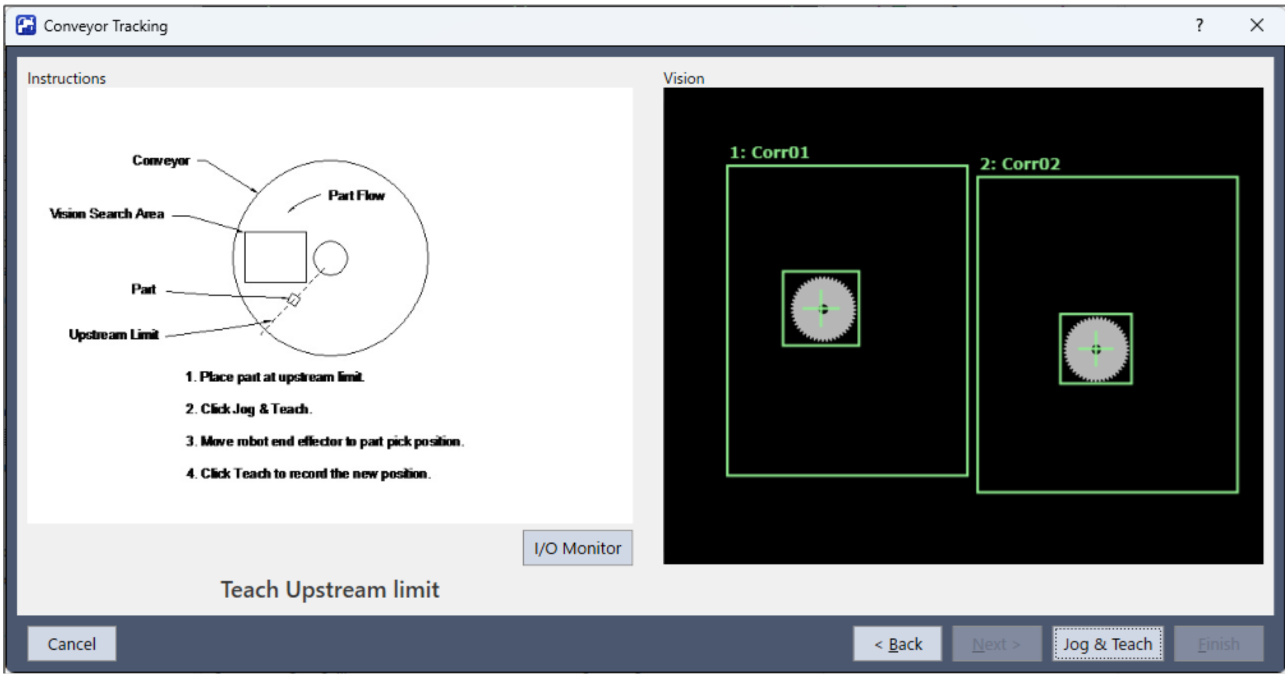
20. The [Jog & Teach panel] will appear. Click the jog buttons to move the robot end effector to the pick position.

    Click the [Teach] button.



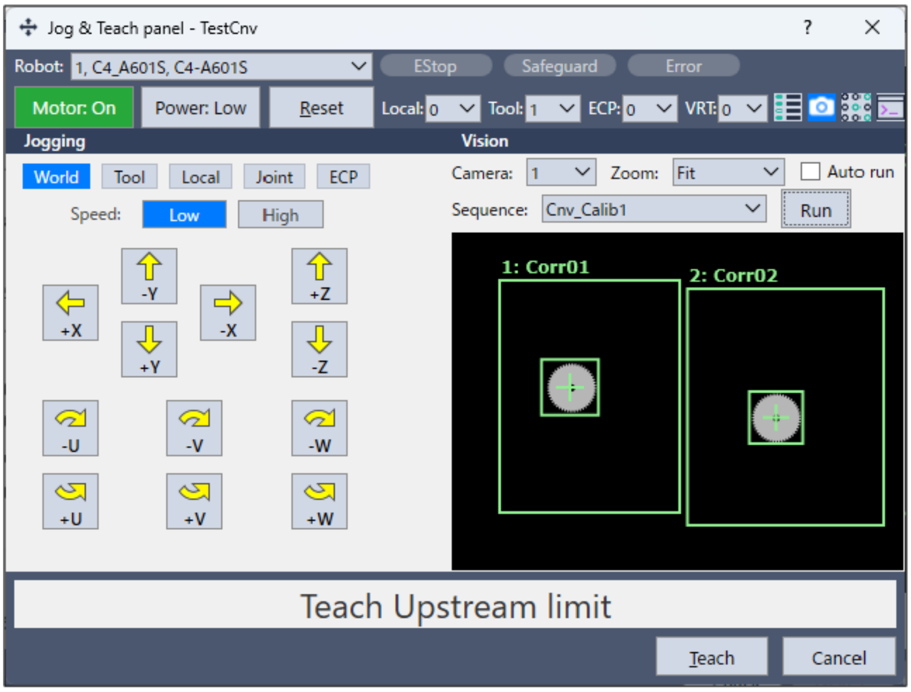
21. Place a part on the upstream limit.

 Click the [Jog & Teach] button.



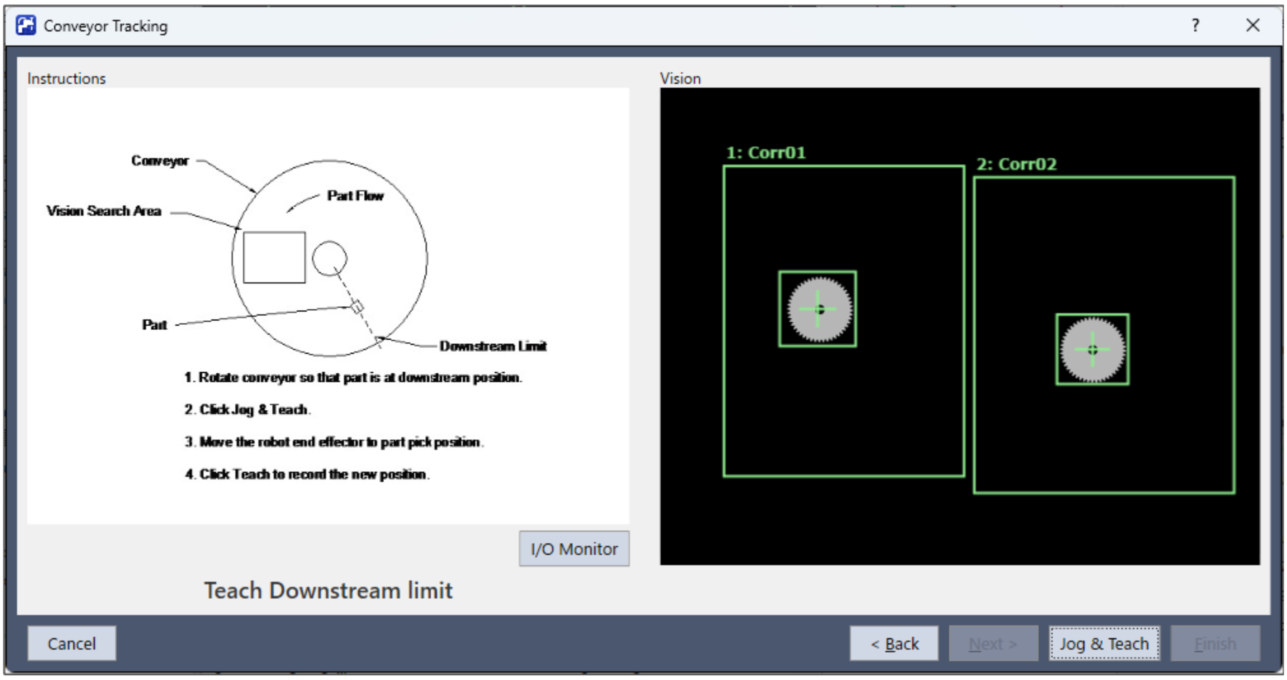
22. The [Jog & Teach panel] will appear. Click the jog buttons to move the robot end effector above the part.

 Click the [Teach] button.



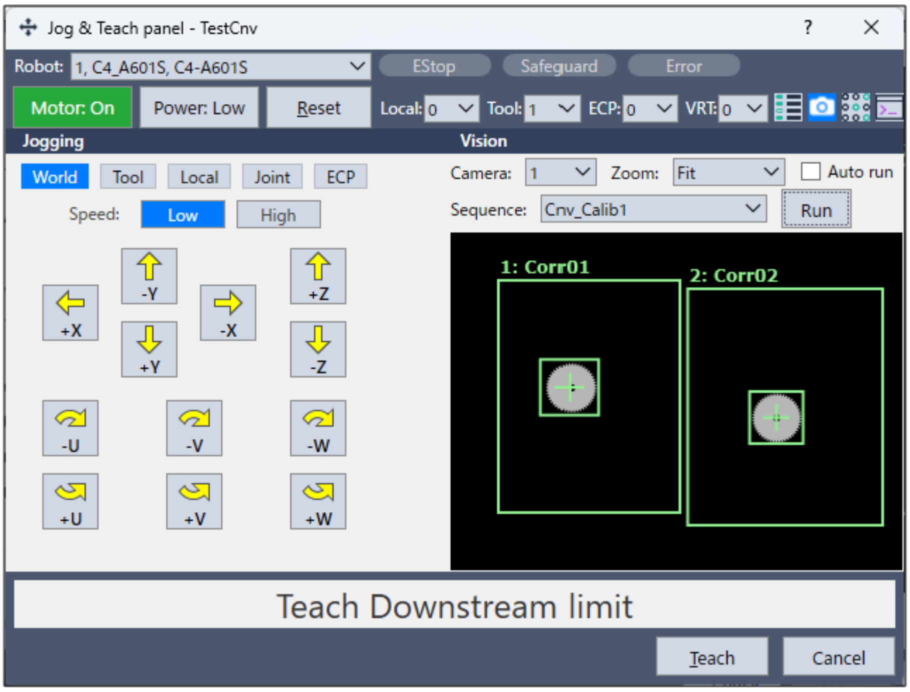
23. Move the conveyor so the part is at the downstream limit.

Click the [Jog & Teach] button.



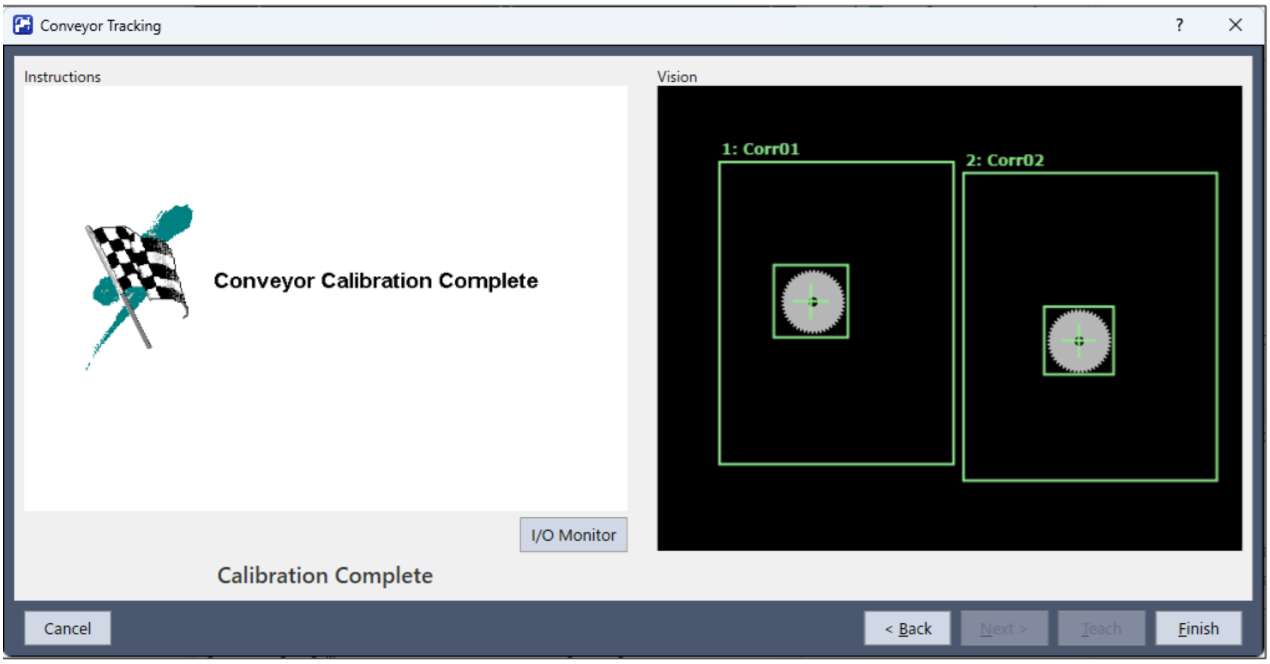
24. The [Jog & Teach panel] will appear. Click the jog buttons to move the robot end effector above the part.

Click the [Teach] button.

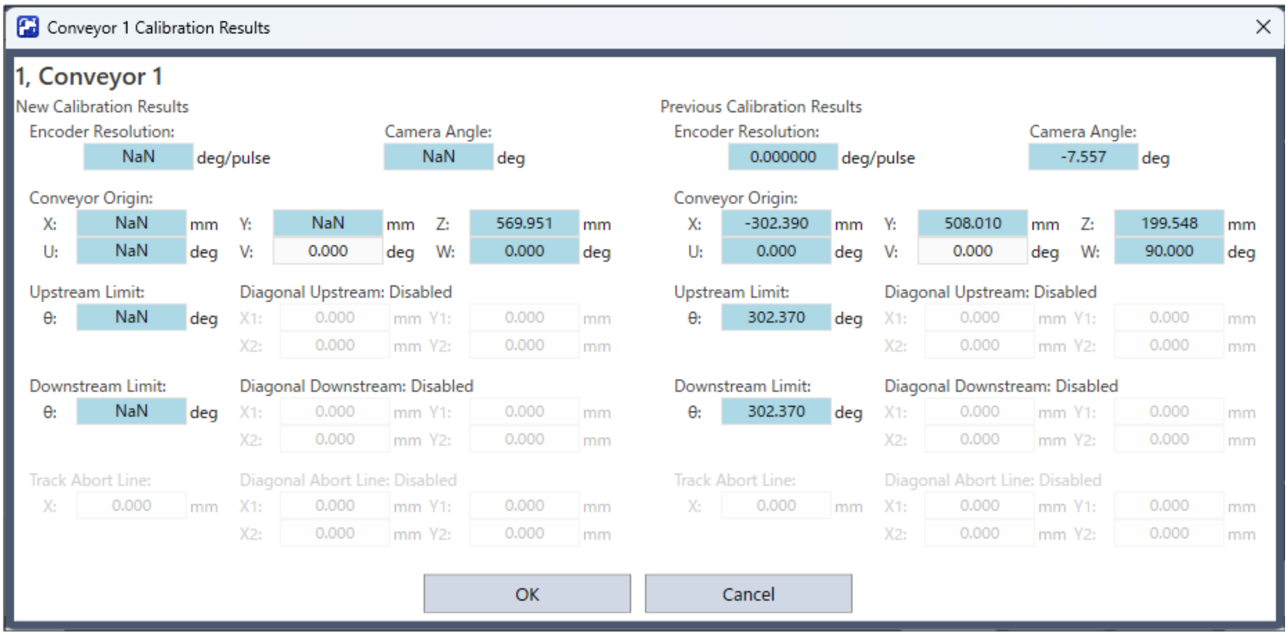


25. The calibration complete picture will be displayed.

Click the [Finish] button.



26. The Calibration Results screen is displayed.
   - Click the [OK] button to complete the calibration.
   - Click the [Cancel] button to return to the Calibration Complete screen in step 25.



17.13.5 Vision conveyor operation check

After the calibration, we recommend that you check if the vision conveyor works properly. Select a suitable method since the verification procedures vary depending on the system.

This section uses the program and the command window described in.

Sample Program

Method 1: When the conveyor can be stopped arbitrary and the conveyor speed can be 30 mm/sec or less

1. Clear the all queue data registered to each conveyor.

```
>Cnv_QueRemove 1,all
```

2. Place the part in the vision search area.

3. Execute the program “ScanConveyorStrobed” to register a queue.

4. Halt the program “ScanConveyorStrobed” and move the conveyor until the part enters the Pickup Area.

5. Pick up the part.

   When using the 6-axis robot, set U, V, and W values as shown below. When using the SCARA robot, setting of U, V, and W are not necessary.

```
>Go Cnv_Queget(1,0):U(90):V(0):W(180)
```

6. Check if the robot end effector is above the center of a part.

7. Move the conveyor at 50mm/sec or less and check if the robot follows the part. At this point, the end effector will be off the center of part but this is not a problem.

8. Stop the tracking motion of the robot.

```
>Cnv_AbortTrack
```

In case the following symptoms occur with the above method, the Vision Guide or conveyor calibration was not executed properly. Perform the calibration again.

- In step (6), the robot end effector is more than 1 mm away from the center of the part.
- The robot cannot follow the part when the conveyor is moved in step (7).

Method 2: When the conveyor can be stopped arbitrary and the conveyor speed can be 100 mm/sec or less

1. Clear the all queue data registered to each conveyor.

```
>Cnv_QueRemove 1,all
```

2. Place the part in the vision search area.

3. Execute the program “ScanConveyorStrobed” to register a queue.

4. Halt the program “ScanConveyorStrobed” and move the conveyor until the part enters the Pickup Area.

5. Pick up the part.

   When using the 6-axis robot, set U, V, and W values as shown below. When using the SCARA robot, setting of U, V, and W are not necessary.

```
>Go Cnv_Queget(1,0):U(90):V(0):W(180)
```

6. Check if the robot end effector is above the center of a part.

7. Change the mode to “High Power”.

```
>Power High
```

8. Move the conveyor and check if the robot follows the part. At this point, the end effector will be off the center of part but this is not a problem.

9. Stop the tracking motion of the robot.

```
>Cnv_AbortTrack
```

In case the following symptoms occur with the above method, the Vision Guide or conveyor calibration was not executed properly. Perform the calibration again.

- In step (6), the robot end effector is more than 2 mm away from the center of the part.
- The robot cannot follow the part when the conveyor is moved in step (8).

Method 3: When the conveyor can be stopped arbitrary

1. Clear the all queue data registered to each conveyor.

```
>Cnv_QueRemove 1,all
```

2. Place the part in the vision search area.

3. Execute the program “ScanConveyorStrobed” to register a queue.

4. Halt the program “ScanConveyorStrobed” and move the conveyor until the part enters the Pickup Area.

5. Pick up the part.

When using the 6-axis robot, set U, V, and W values as shown below. When using the SCARA robot, setting of U, V, and W are not necessary.

```
>Go Cnv_Queget(1,0):U(90):V(0):W(180)
```

6. Check if the robot end effector is above the center of a part.

7. Stop the tracking motion of the robot.

```
>Cnv_AbortTrack
```

8. Use the program “Main” to check if the robot follows the part.

At this point, change the wait time after tracking to 0.2 to 0.5 in the sample program.

In case the following symptoms occur with the above method, the Vision Guide or conveyor calibration was not executed properly. Perform the calibration again.

- In step (6), the robot end effector is more than 1 mm away from the center of the part.
- The robot moves to different position from the parts in step (8).

Method 4: When the conveyor cannot be stopped and the speed cannot be changed arbitrary

1. Move the conveyor.

2. Change the sample program as follows.
   - Change the wait time after tracking to 0.2 to 0.5
   - Set the tracking mode to "0"

3. Execute the sample program “Main”.

4. Place the part after the conveyor speed becomes constant.

5. Check if the robot follows the part.

6. Change the sample program as follows.
   - Set the tracking mode to "1"

7. Execute the sample program “Main”.

8. Place the part after the conveyor speed becomes constant.

9. Check if the robot follows the part.

In case the following symptoms occur with the above method, the Vision Guide or conveyor calibration was not executed properly. Perform the calibration again.

- When comparing step (5) and (9), a distance between the robot and the part is smaller in step (5).
- The robot moves to different position from the parts in step (5).

17.14 Sensor Conveyors

17.14.1 Sensor conveyor calibration (Straight conveyor)

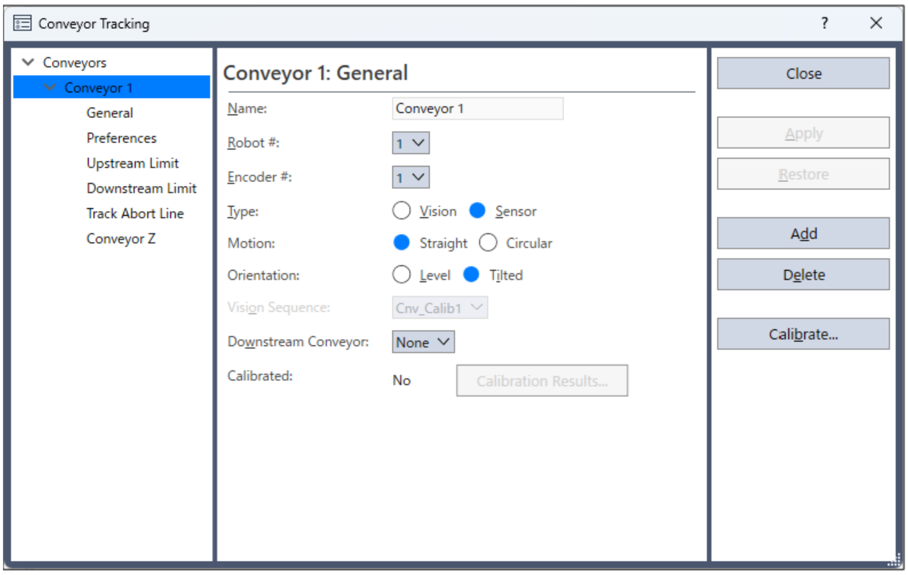
Follow these steps to calibrate a straight sensor conveyor:

KEY POINTS

- When teaching part positions with the robot during calibration, it is important to position X, Y, and Z of each point accurately. The conveyor is calibrated in X, Y, Z, U, V, and W.
- To perform the fine calibration, in the step 9 and 11, set as wide distance as possible between the upstream limit and downstream limit. After the calibration, adjust the Pickup Area by resetting the upstream / downstream limits.
- For the level orientation, the conveyor height is determined by the position of the robot end effector taught in step 8. It cannot be used for the tilted conveyor for it does not detect the conveyor slope. Steps 19 to 20 are not displayed.
- For the tilted orientation, it calibrates the conveyor slope with the position of robot end effector taught in the steps 8, 10, 12, and 14.

1. Select [Tools]-[Conveyor Tracking].

2. Select the conveyor you want to calibrate.



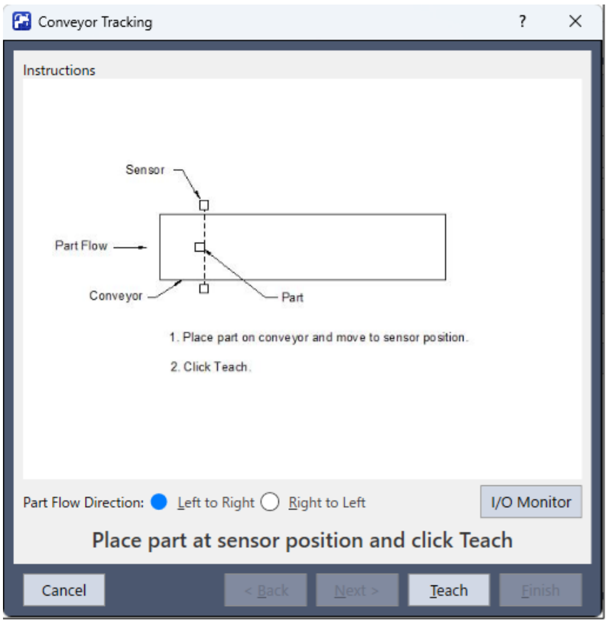
3. Click the [Calibrate] button. The [Conveyor Tracking Calibration] wizard is displayed.

4. Follow the instructions for each step. Before you can proceed to the next step, you must click the [Teach] button. You can go back to previous steps using the [Back] button.

5. Select the [Part Flow Direction] to best match the conveyor you are calibrating. The instruction pictures will change according to the setting. [Part Flow Direction] is only used to aid in the instructions. It has no effect on the calibration.

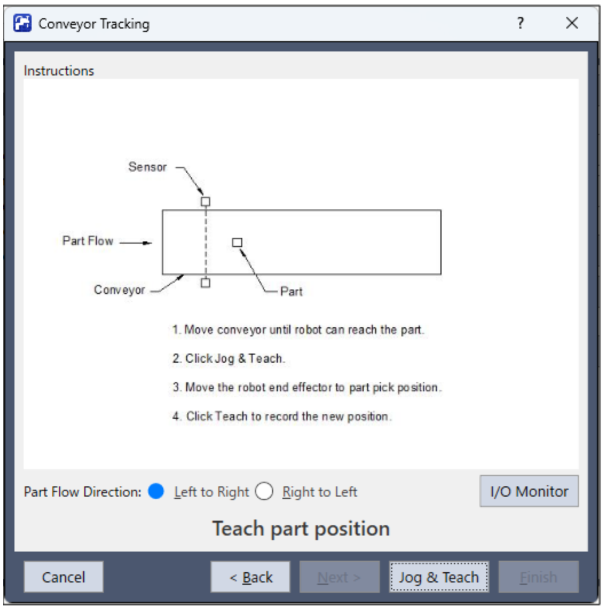
6. Place a part on the conveyor and move the conveyor toward the sensor until the sensor just turns on.

   Click the [Teach] button.



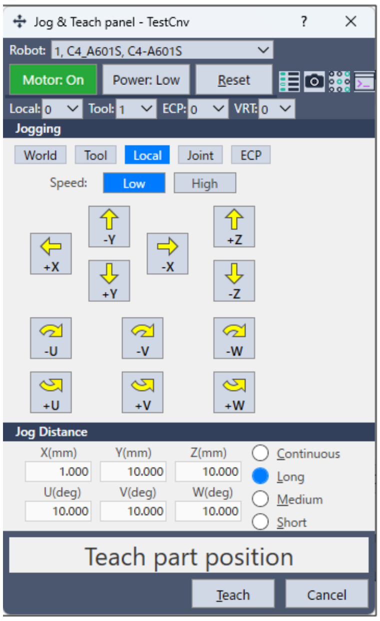
7. Move the conveyor until the part is within reach of the robot. Do not move the parts, only the conveyor.

   Click the [Jog & Teach] button.

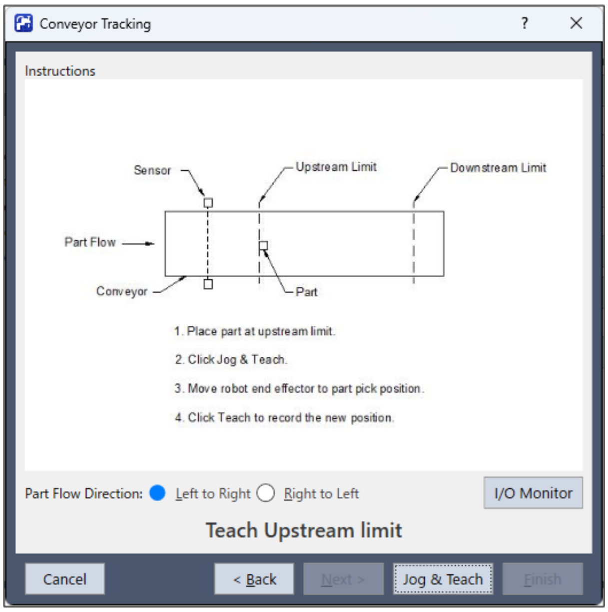


8. The [Jog & Teach panel] will appear. Click the jog buttons to move the robot end effector to the pick position.

   Click the [Teach] button.

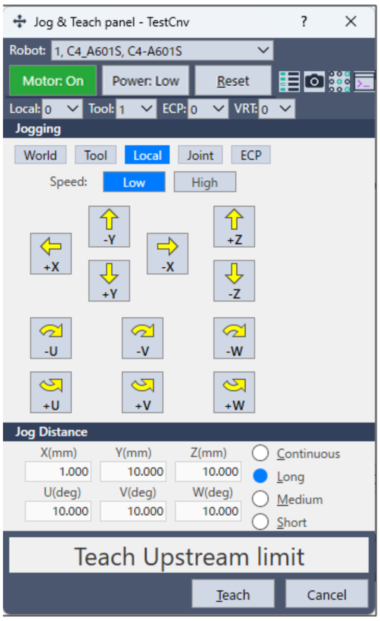


9. Now move or place the part at the upstream limit. Click the [Jog & Teach] button.



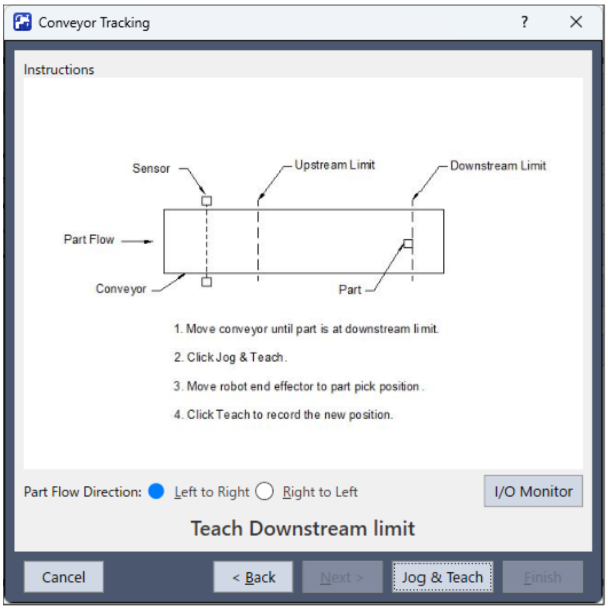
10. The [Jog & Teach panel] will appear. Click the jog buttons to move the robot end effector to the pick position.

Click the [Teach] button.



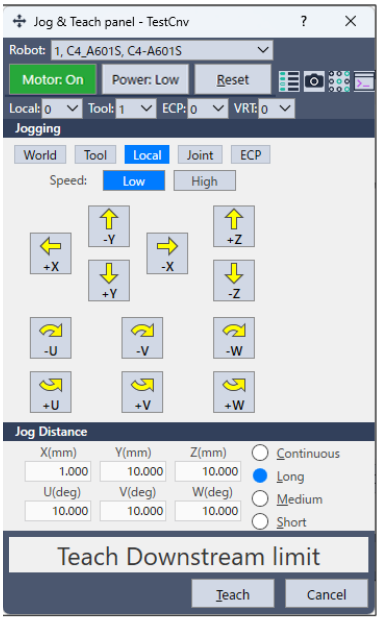
11. Move the conveyor so the part is at the downstream limit. Do not move the parts, only the conveyor.

    Click the [Jog & Teach] button.



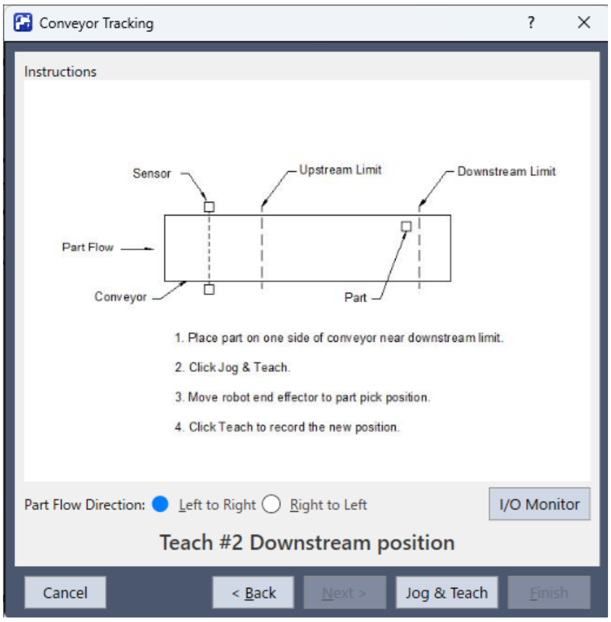
12. The [Jog & Teach panel] will appear. Click the jog buttons to move the robot end effector to the pick position.

    Click the [Teach] button.



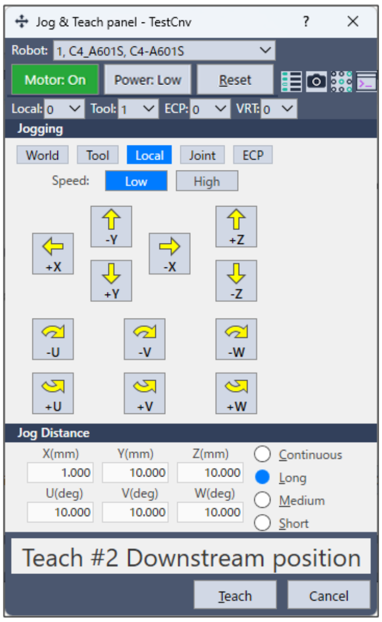
13. Place a part on one side of the conveyor near the downstream limit. This point is used to determine the tilt of the conveyor from side to side.

    Click the [Jog & Teach] button.

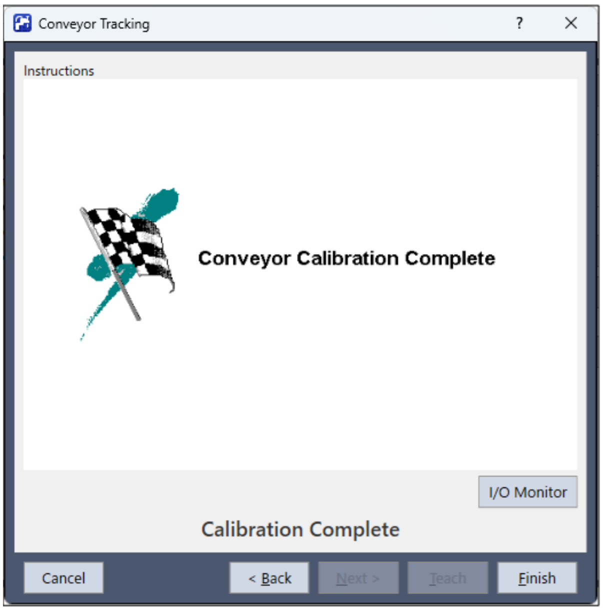


14. The [Jog & Teach panel] will appear. Click the jog buttons to move the robot end effector to the pick position.

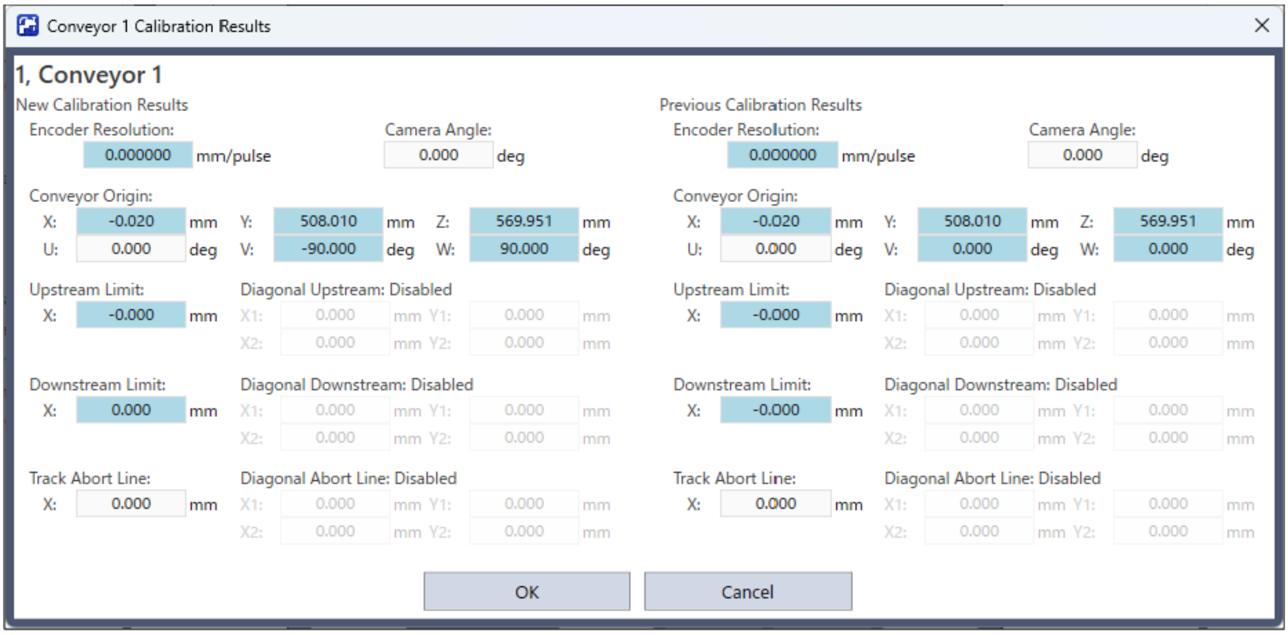
    Click the [Teach] button.



15. The calibration complete picture will be displayed. Click the [Finish] button.



16. The Calibration Results screen is displayed.
   - Click the [OK] button to complete the calibration.
   - Click the [Cancel] button to return to the Calibration Complete screen in step 15.



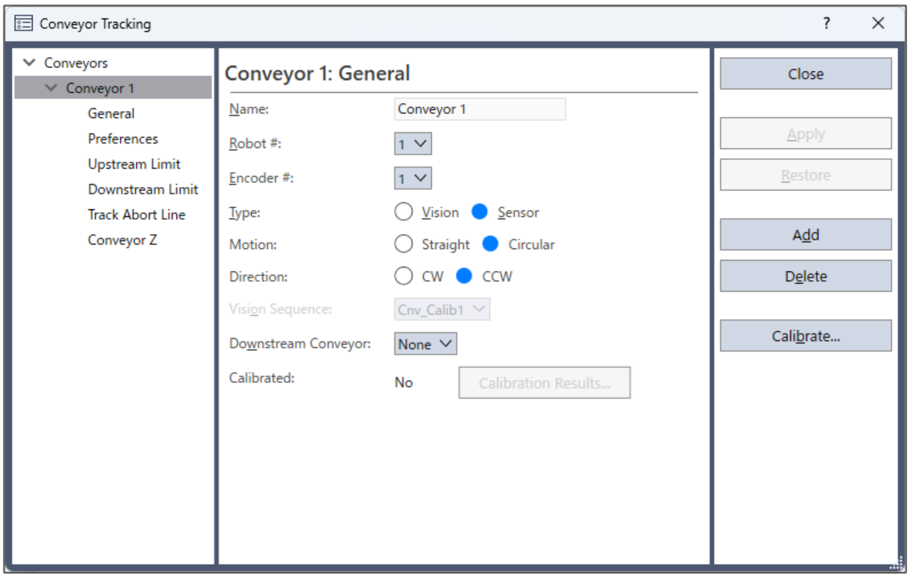
17.14.2 Sensor Conveyor Calibration (Circular conveyor)

Follow these steps to calibrate a circular sensor conveyor:

KEY POINTS

- When teaching part positions with the robot during calibration, it is important to position X, Y, and Z of each point accurately. The conveyor is calibrated in X, Y, Z, U, V, and W.
- To perform the fine calibration, in steps 10, 12, and 14, teach the position when the robot is directly above the parts and set as wide a distance as possible between the points to be taught.

1. Select [Tools]-[Conveyor Tracking].
2. Select the conveyor you want to calibrate.
3. Select [Sensor] for the [Type].
4. Select [Circular] for the [Motion].
5. Select the conveyor rotating direction for the [Direction]. Be careful not to calibrate with a wrong direction, otherwise, the robot will not track the parts.

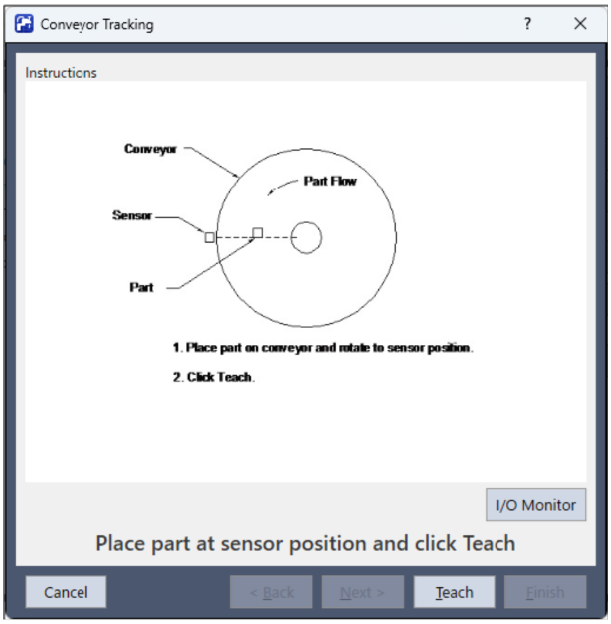


6. Click the [Apply] button.
7. Click the [Calibrate] button.

   The [Conveyor Tracking Calibration] wizard is displayed. Follow the instructions for each step. Before you can proceed to the next step, you must click the [Teach] button. You can go back to previous steps using the [Back] button.

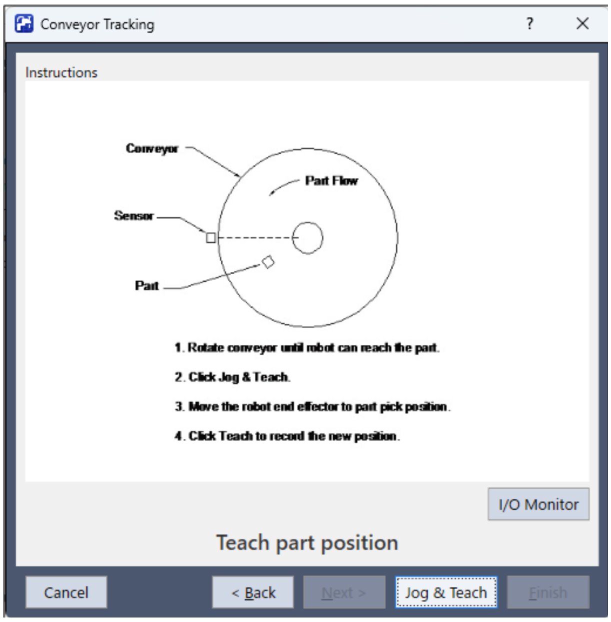
8. Check if the conveyor direction shown in the wizard is the same as the conveyor you want to use.
9. Place a part on the conveyor and move the conveyor toward the sensor until the sensor just turns on.

   Click the [Teach] button.



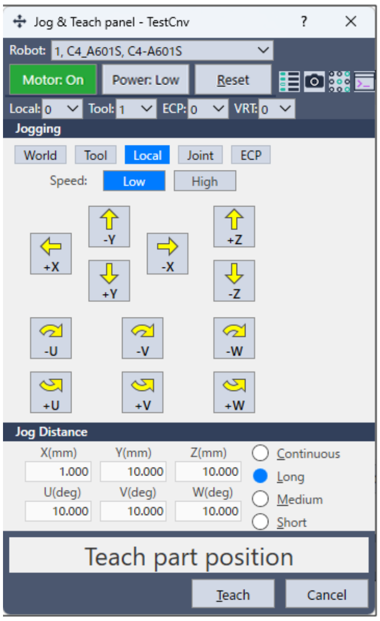
10. Move the conveyor by hand to move Part.

    Click the [Jog & Teach] button.



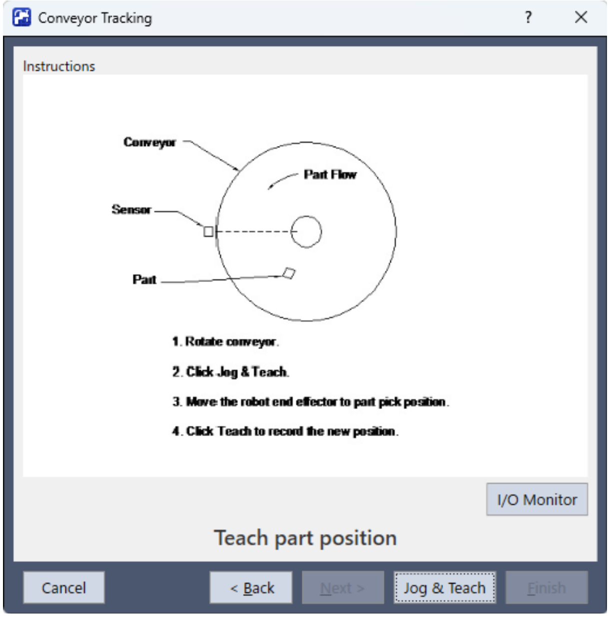
11. The [Jog & Teach panel] will appear. Click the jog buttons to move the robot end effector to the pick position.

    Click the [Teach] button.



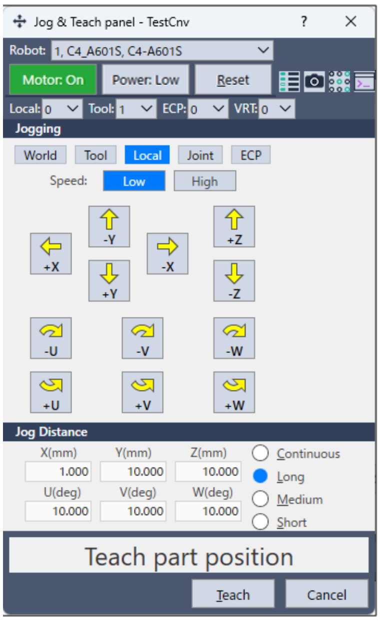
12. Move the conveyor to move Part.

    Click the [Jog & Teach] button.



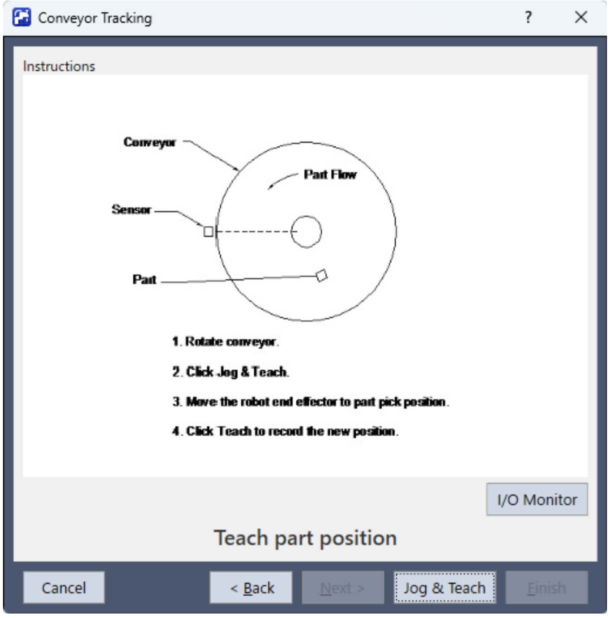
13. The [Jog & Teach panel] will appear. Click the jog buttons to move the robot end effector to the pick position.

    Click the [Teach] button.



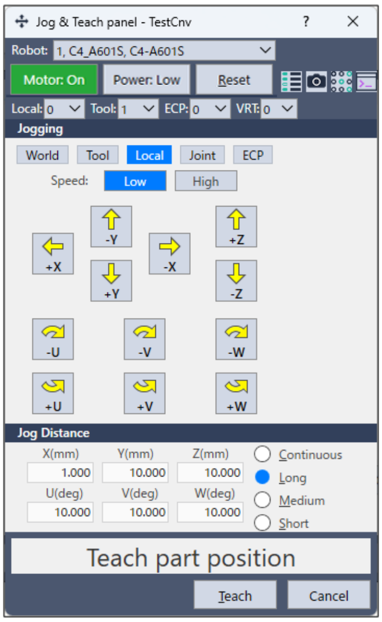
14. Move the conveyor to move Part.

    Click the [Jog & Teach] button.



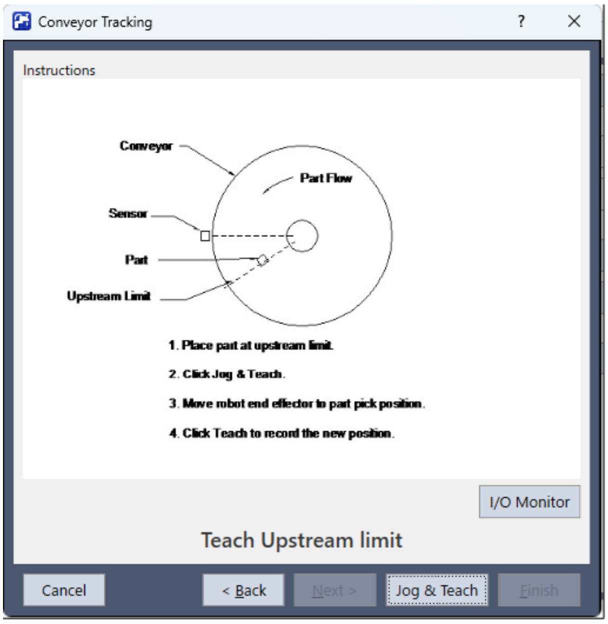
15. The [Jog & Teach panel] will appear. Click the jog buttons to move the robot end effector to the pick position.

    Click the [Teach] button.



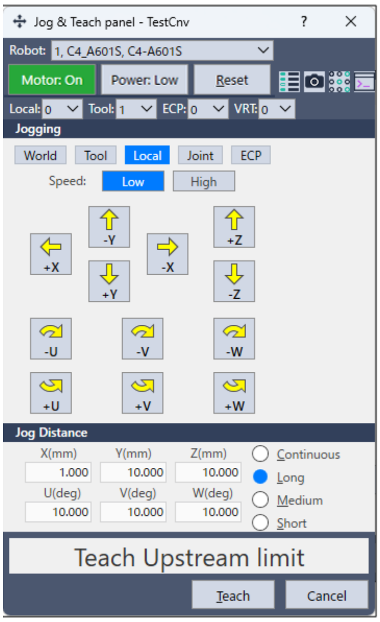
16. Place a part on the upstream limit.

    Click the [Jog & Teach] button.



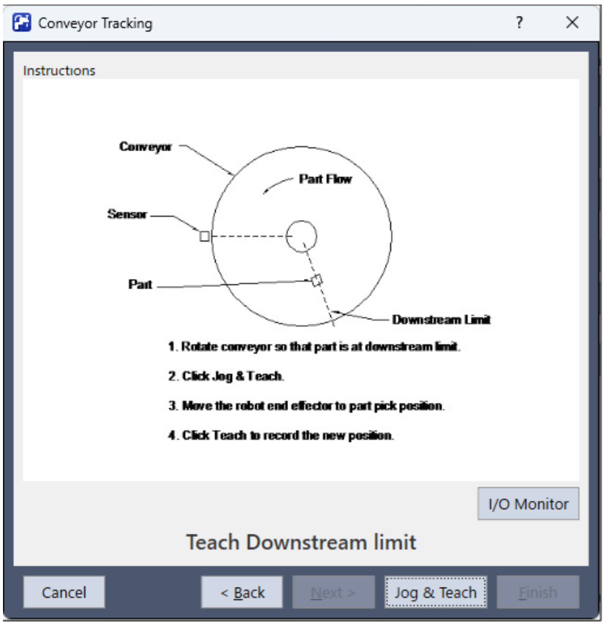
17. The [Jog & Teach panel] will appear. Click the jog buttons to move the robot end effector above the part.

    Click the [Teach] button.



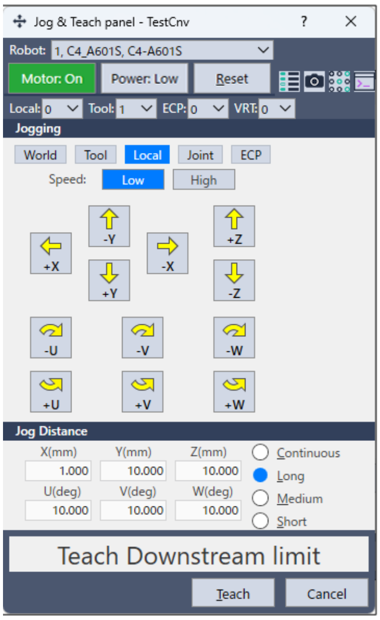
18. Move the conveyor so the part is at the downstream limit.

    Click the [Jog & Teach] button.



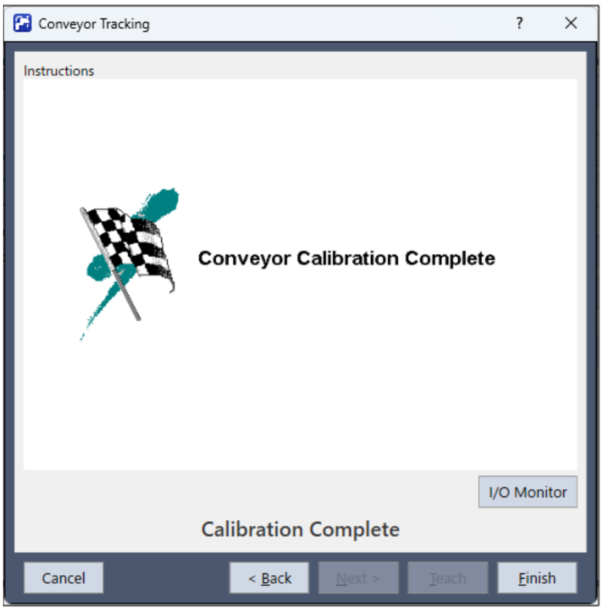
19. The [Jog & Teach panel] will appear. Click the jog buttons to move the robot end effector above the part.

    Click the [Teach] button.

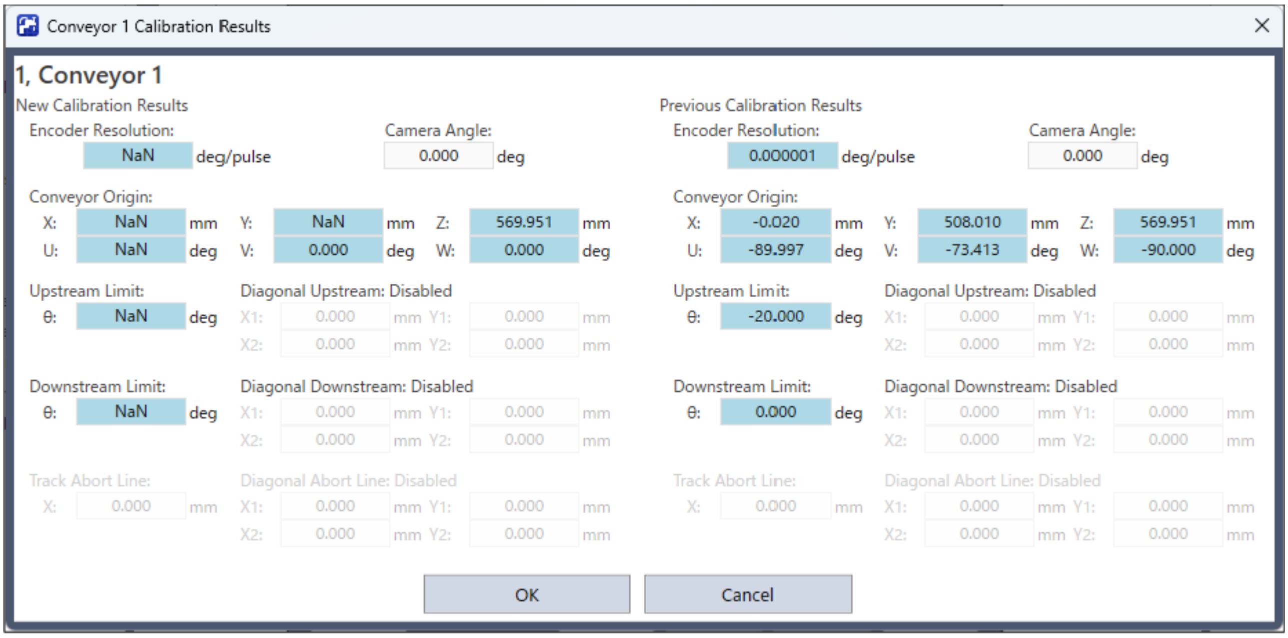


20. The calibration complete picture will be displayed.

    Click the [Finish] button.



21. The Calibration Results screen is displayed.
    - Click the [OK] button to complete the calibration.
    - Click the [Cancel] button to return to the Calibration Complete screen in step 20.



17.14.3 Sensor conveyor operation check

After the calibration, we recommend that you check if the sensor conveyor works properly. Select a suitable method since the verification procedures vary depending on the system.

This checking method uses the program and command window described below.

Sample Program

Method 1: When the conveyor can be stopped arbitrary and the conveyor speed can be 30 mm/sec or less

1. Clear the all queue data registered to each conveyor.

```
>Cnv_QueRemove 1,all
```

2. Detect the part using the sensor.

3. Execute the program "ScanConveyor" to register a queue.

4. Halt the program "ScanConveyor" and move the conveyor until the part enters the Pickup Area.

5. Pick up the part.

   When using the 6-axis robot, set U, V, and W values as shown below. When using the SCARA robot, setting of U, V, and W are not necessary.

   ```
   >Go Cnv_Queget(1,0):U(90):V(0):W(180)
   ```

6. Check if the robot end effector is above the center of a part.

7. Move the conveyor at 50mm/sec or less and check if the robot follows the part. At this point, the end effector will be off the center of part but this is not a problem.

8. Stop the tracking motion of the robot.

```
>Cnv_AbortTrack
```

In case the following symptoms occur with the above method, the conveyor calibration was not executed properly. Perform the calibration again.

- In step (6), the robot end effector is more than 1 mm away from the center of the part.
- The robot cannot follow the part when the conveyor is moved in step (7).

Method 2: When the conveyor can be stopped arbitrary and the conveyor speed can be 100 mm/sec or less

1. Clear the all queue data registered to each conveyor.

   ```
   >Cnv_QueRemove 1,all
   ```

2. Detect the part using the sensor.

3. Execute the program "ScanConveyor" to register a queue.

4. Halt the program "ScanConveyor" and move the conveyor until the part enters the Pickup Area.

5. Pick up the part.

When using the 6-axis robot, set U, V, and W values as shown below. When using the SCARA robot, setting of U, V, and W are not necessary. `>Go Cnv_Queget(1,0):U(90):V(0):W(180)`

6. Check if the robot end effector is above the center of a part.

7. Change the mode to "High Power".

   ```
   >Power High
   ```

8. Move the conveyor and check if the robot follows the part. At this point, the end effector will be off the center of part but this is not a problem.

9. Stop the tracking motion of the robot.

```
>Cnv_AbortTrack
```

In case the following symptoms occur with the above method, the conveyor calibration was not executed properly. Perform the calibration again.

- In step (6), the robot end effector is more than 2 mm away from the center of the part.
- The robot cannot follow the part when the conveyor is moved in step (8).

Method 3: When the conveyor can be stopped arbitrary

1. Clear the all queue data registered to each conveyor.

   ```
   >Cnv_QueRemove 1,all
   ```

2. Detect the part using the sensor.

3. Execute the program "ScanConveyor" to register a queue.

4. Halt the program "ScanConveyor" and move the conveyor until the part enters the Pickup Area.

5. Pick up the part.

   When using the 6-axis robot, set U, V, and W values as shown below. When using the SCARA robot, setting of U, V, and W are not necessary.

   ```
   >Go Cnv_Queget(1,0):U(90):V(0):W(180)
   ```

6. Check if the robot end effector is above the center of a part.

7. Stop the tracking motion of the robot.

   ```
   >Cnv_AbortTrack
   ```

8. Use the program "Main" to check if the robot follows the part.

   At this point, change the wait time after tracking to 0.2 to 0.5 in the sample program.

In case the following symptoms occur with the above method, the conveyor calibration was not executed properly. Perform the calibration again.

- In step (6), the robot end effector is more than 1 mm away from the center of the part.
- The robot moves to different position from the parts in step (8).

Method 4: When the conveyor cannot be stopped and the speed cannot be changed arbitrary

1. Move the conveyor.

2. Change the sample program as follows.
   - Change the wait time after tracking to 0.2 to 0.5
   - Set the tracking mode to "0"

3. Execute the sample program "Main".

4. Place the part after the conveyor speed becomes constant.

5. Check if the robot follows the part.

6. Change the sample program as follows.

- Set the tracking mode to "1"

7. Execute the sample program “Main”.

8. Place the part after the conveyor speed becomes constant.

9. Check if the robot follows the part.

In case the following symptoms occur with the above method, the conveyor calibration was not executed properly. Perform the calibration again.

- When comparing step (5) and (9), a distance between the robot and the part is smaller in step (5).
- The robot moves to different position from the parts in step (5).

17.15 Calibration Results

Calibration results will appear once conveyor calibration is complete, or when navigating from [Tools]-[Conveyor Tracking] and clicking the [Calibration Results...] button.

- Left: Latest calibration results
- Right: Previous results

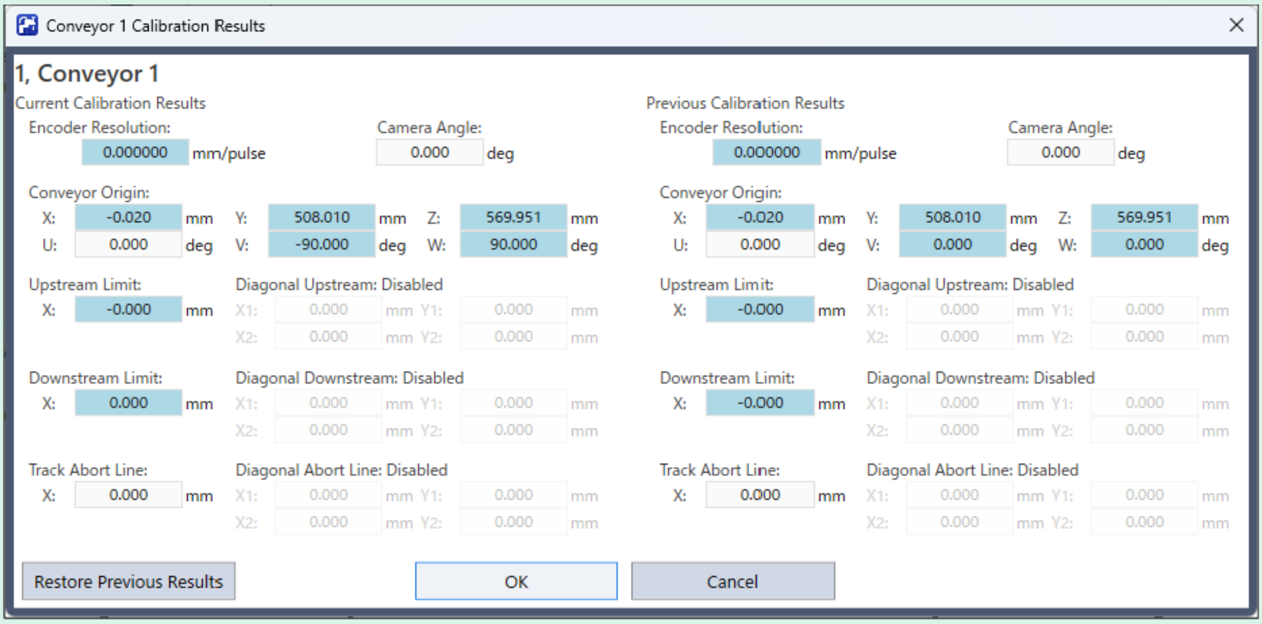
Items where values differ between the latest results and the previous results are highlighted in yellow.

The latest calibration results are used for conveyor tracking. Previous results have no effect on performance.

KEY POINTS

- The [Calibration Results...] button will be unavailable under the following circumstances:
    - When conveyor calibration has not been performed
    - When changing the Robot #, Encoder #, Orientation, Type, and Vision Sequence
- Diagonal upstream limit, diagonal downstream limit, tracking abort line, and diagonal tracking abort line values only appear when settings have been configured as described in the procedure below.

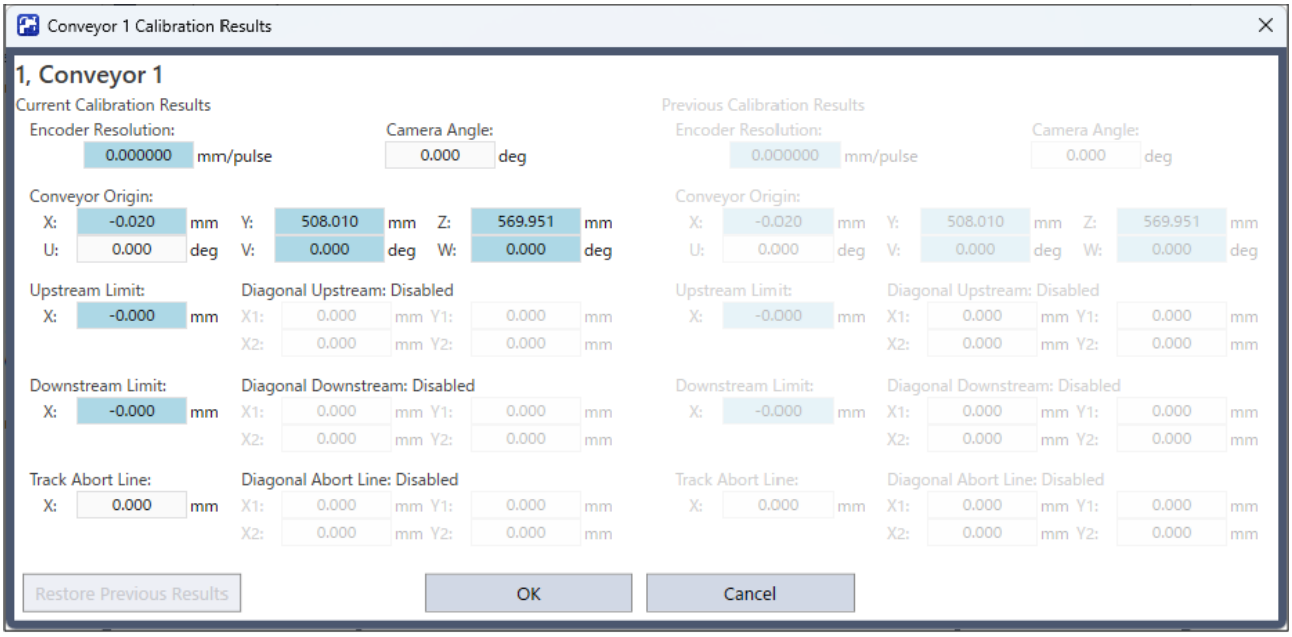
  Configuring Conveyors
- The [Restore Previous Results] button will not appear when calibration results are shown after conveyor calibration is complete.
- Values cannot be overwritten on the calibration results screen.



Performing conveyor tracking using previous results:

Click the [Restore Previous Results] button to overwrite the calibration results with the previous results as shown on the screen below. Only one set of previous results can be stored. Restoring previous results will wipe the previous results stored.

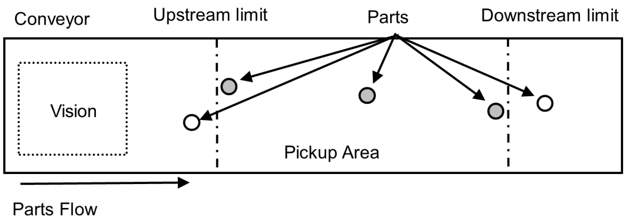
Click the [OK] button on this screen to restore results. Click the [Cancel] button to return the screen to the way it was before restoring. Note that you cannot undo restored calibration results once confirmed.



17.16 Pickup Area

The Pickup Area is the range where the robot can pick up parts.

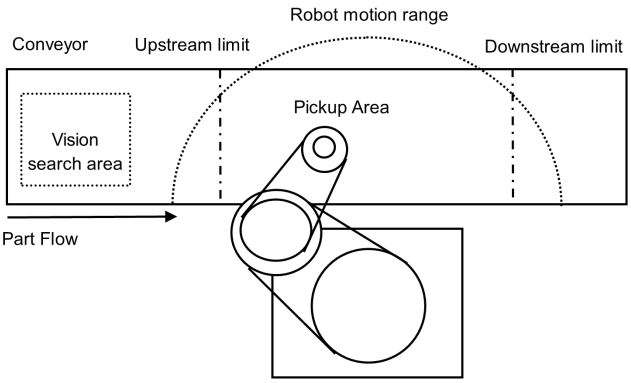
In the figure below, the robot can pick up the parts in gray.



If the Pickup Area is not appropriate, the robot cannot pick up parts. Follow the steps and cautions below to carefully set the Pickup Area.

To define the Pickup Area:

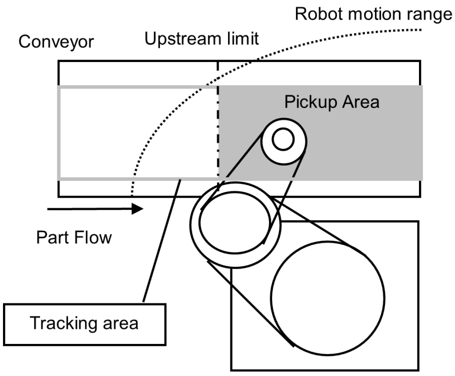
1. After calibration, the Pickup Area will be defined as shown in the following figure. Note that the positions of upstream limit and downstream limit depend on the positions you teach during the calibration.



2. Decide the upstream limit position.

The robot starts pickup from the line defined by the upstream limit. The Pickup Area from the upstream limit must be within the robot motion range. (See the figure below.)

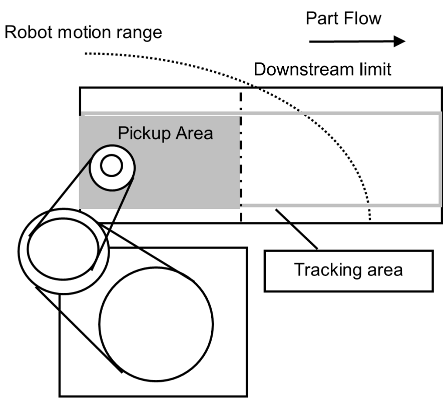
> **KEY POINTS**
>
> The robot does not start pickup until parts cross the upstream limit. If you set the upstream limit in uppermost position, you can reduce the robot standby time.



3. Decide the downstream limit position.

   Once the robot starts pickup, it continues its operation even over the downstream limit to complete the whole operation. Therefore, set the downstream limit in uppermost possible position so that the robot can operate within its motion range until it completes the operation. (See the figure below.)

> **KEY POINTS**
>
> The downstream limit position depends on the conveyor speed and robot position when it starts pickup. If the robot goes over the motion range during the operation, move the downstream limit to upper side.



17.16.1 Changing the Upstream / Downstream limits positions

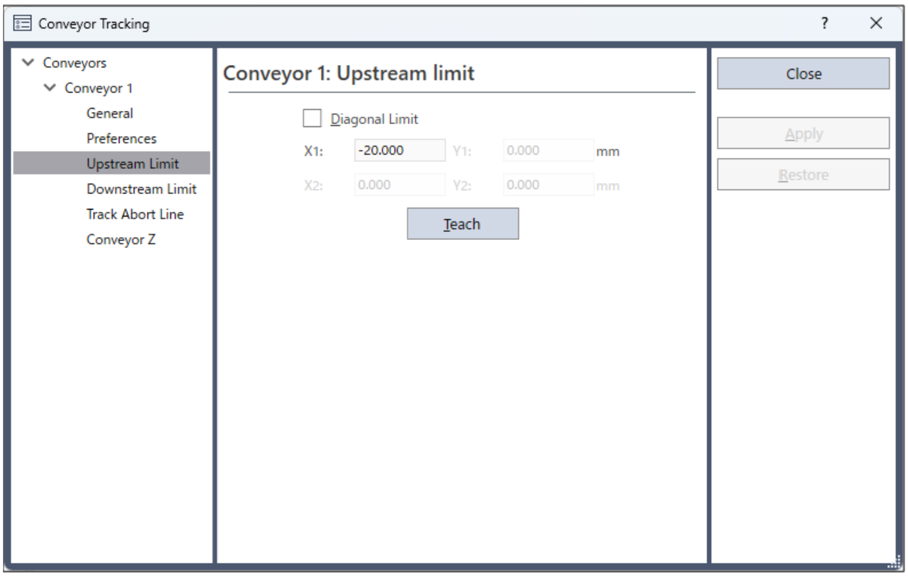
To change the upstream limit and downstream limit positions, follow the steps below.

To change the Upstream Limit:

- Select [Tools]-[Conveyor Tracking].
- Select the conveyor you want to edit.

- Click [Upstream Limit].
- The dialog shown below appears.

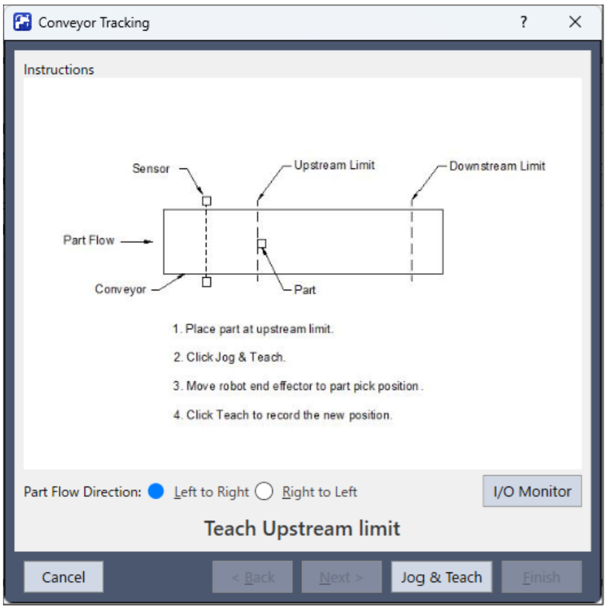
To define the X1 value, enter a value directly or use Jog & Teach. Entering values directly is for fine adjustment.



When you directly specify the value, enter the value in the box and click .

- When you use Jog & Teach, click the [Teach] button.
- The dialog shown below appears.

Follow the instructions given during calibration.



To change the downstream limit, click [Downstream Limit] and edit the value in the same way as the upstream limit.

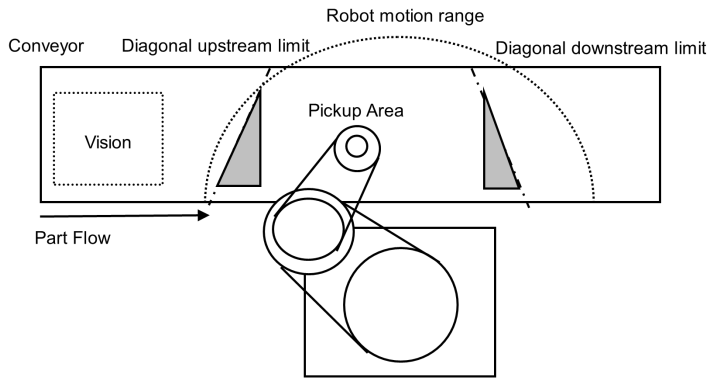
KEY POINTS

Upstream and downstream positions can be changed from the SPEL program by using Cnv_Upstream and Cnv_Downstream commands. (When diagonal upstream and downstream limits are used, they cannot be changed from the SPEL program.)

Diagonal Upstream / Downstream Limits

After the calibration, you can set the dividing lines for the Pickup Area (upstream limit / downstream limit) directed diagonally to the part flow.

When you change the dividing lines to diagonal positions, the Pickup Area also changes as shown below. The area indicated in gray is widened by changing the dividing lines to diagonal positions. In addition, diagonal dividing lines are called the diagonal upstream / downstream limits.



The following are the advantages you can get by widening the Pickup Area.

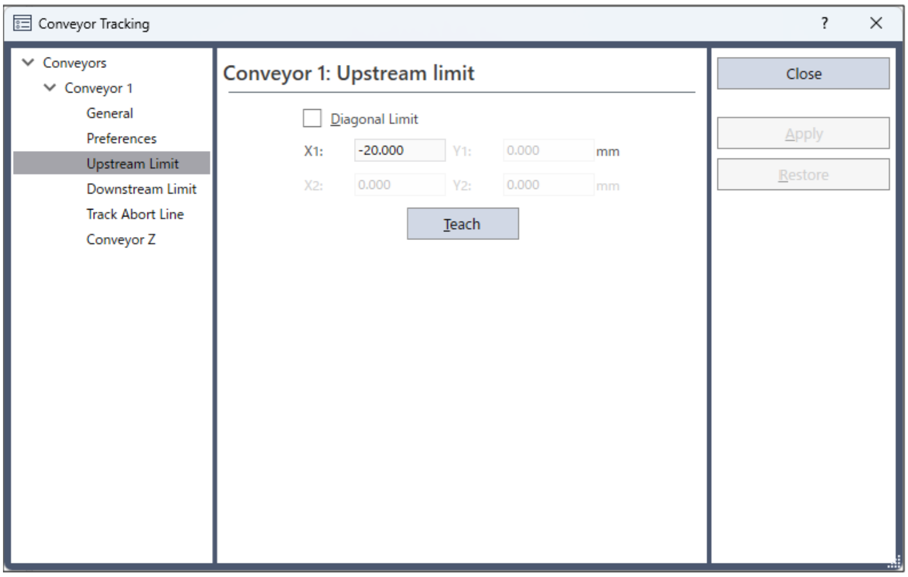
- Reduce robot standby time by widening the upper side Pickup Area.
- Less possibility of missing parts which flow longer after the downstream limit by widening the Pickup Area.

KEY POINTS

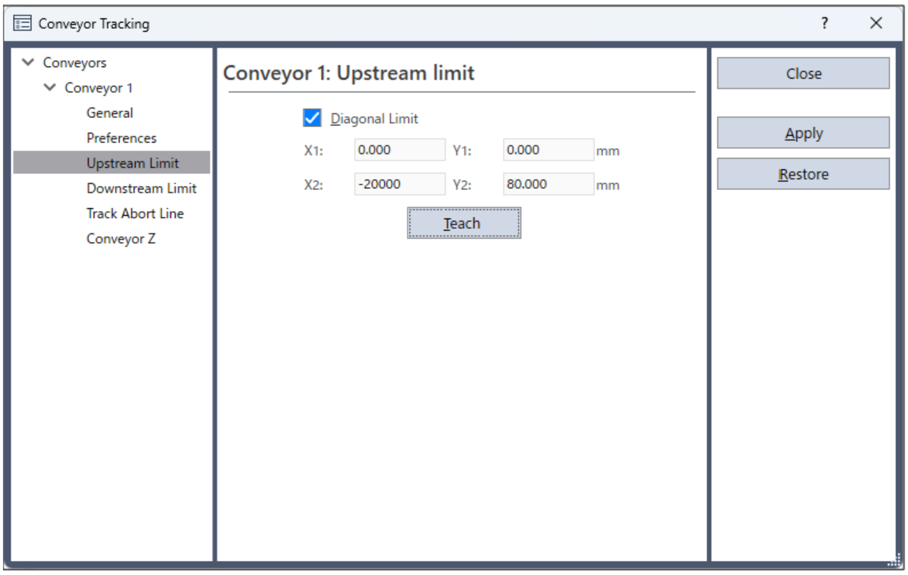
If there are too many parts on the conveyor for the robot to pick up, it only makes the robot move for longer distance and longer time and the number of parts the robot can pick up may decrease, even in a widened Pickup Area. The robot capacity refers to how fast or how many parts robot can pick up. The robot capacity depends on the pickup area width, robot standby position, and conveyor speed.

To set the diagonal upstream limit:

1. Select [Tools]-[Conveyor Tracking].
2. Select the conveyor you want to edit.
3. Click [Upstream Limit].
4. The dialog shown below appears.



Check the [Diagonal Limit] check box in [Upstream Limit] and click the [Apply] button. The dialog shown below appears.

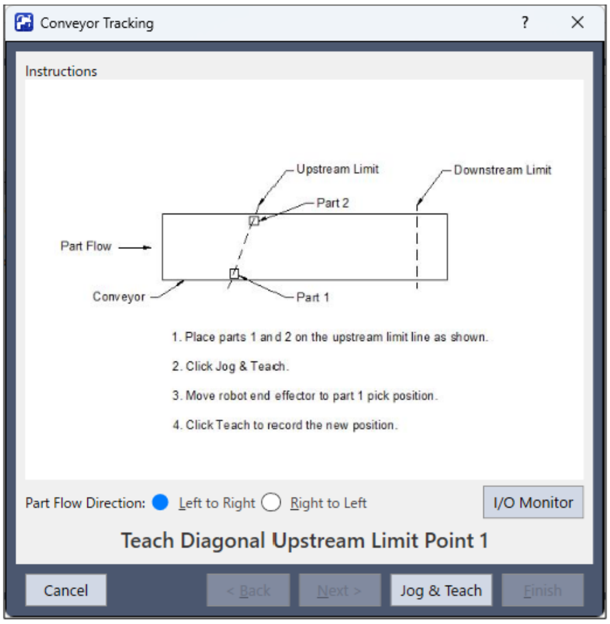


To define the values for X1, Y1, X2, Y2, enter the values directly or use Jog & Teach. Entering values directly is for fine adjustment.

5. When you directly specify the values, enter the values in the boxes and click the [Apply] button.

6. When you use Jog & Teach, click the [Teach] button.

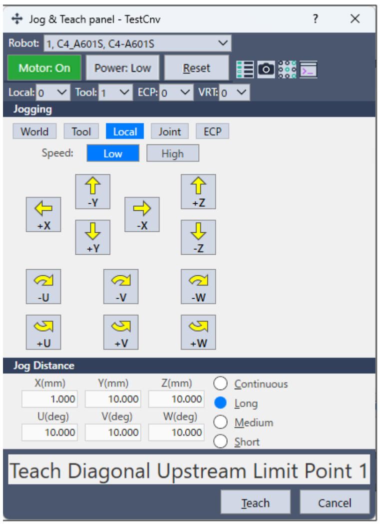
   The window shown below appears.



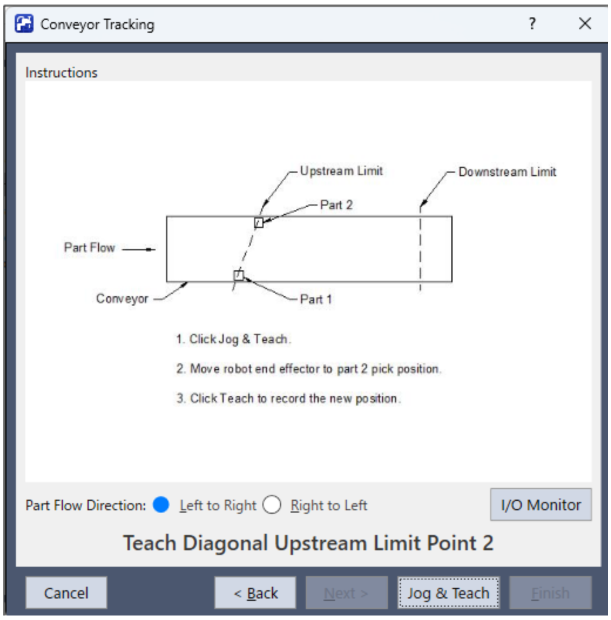
7. Place two parts on the conveyor.

   Click the [Jog & Teach] button.

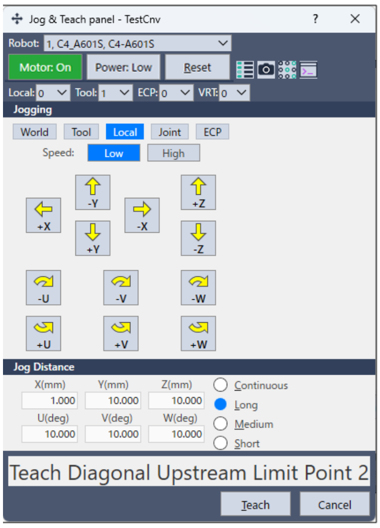
8. The [Jog & Teach panel] will appear. Click the jog buttons to move the robot end effector to the pick position. Click the [Teach] button.



9. The window shown below appears. Click [Jog & Teach].

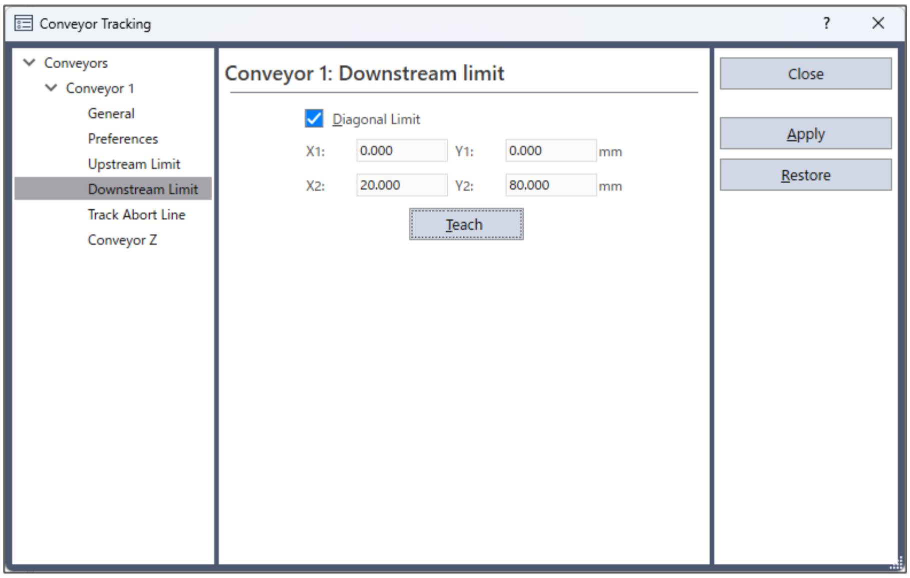


10. The [Jog & Teach panel] will appear. Click the jog buttons to move the robot end effector to the pick position. Click the [Teach] button.



To set the diagonal downstream limit, click [Downstream Limit] to display the downstream limit setting page and check the [Diagonal Limit] checkbox, then click the [Apply] button.

The dialog shown below appears. Click the [Teach] button and follow the directions in the wizard.



Note that the “Error 4415” occurs when the diagonal upstream / downstream limits are defined as in the following cases.

- They are perpendicular to the part flow direction.
- They are parallel to the part flow direction.
- The diagonal upstream limit and downstream limit cross on the conveyor.

17.17 Adjusting the Z value

You can adjust the conveyor Z value after the calibration is completed.

Adjusting the Z value is a function to change the work pickup height that has been determined during calibration.

In the following cases, adjust the Z value:

- To use a part that is different from the one defined during calibration.
- The tool has been changed on the robot after calibration.

To adjust the Z value:

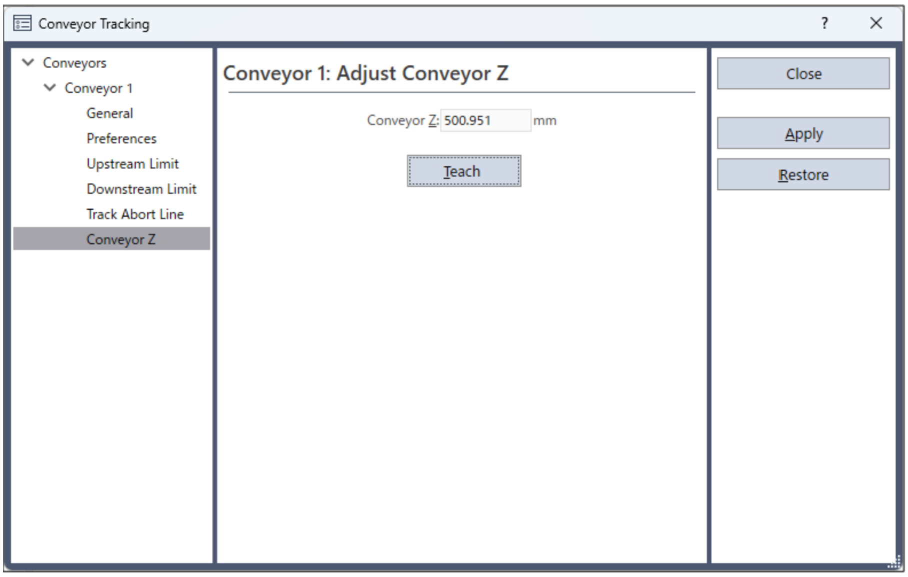
1. Select [Tools]-[Conveyor Tracking].

2. Select the conveyor you want to edit.

3. Click on [Conveyor Z].

4. The dialog shown below appears.

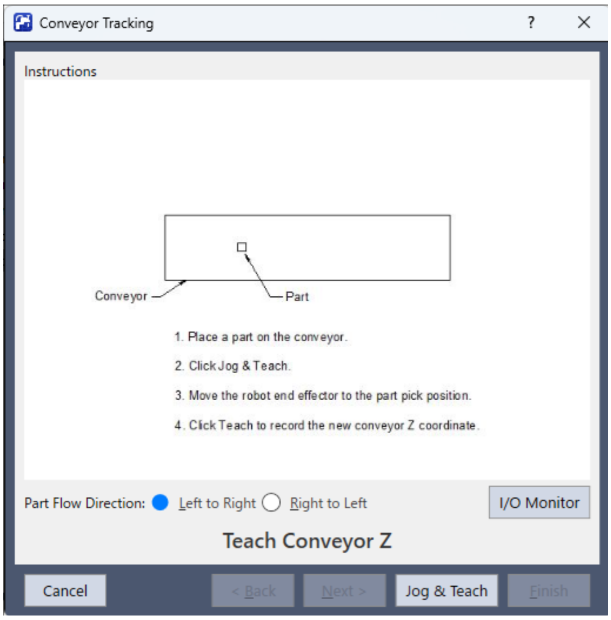
   Click on [Teach].



5. The window shown below appears.

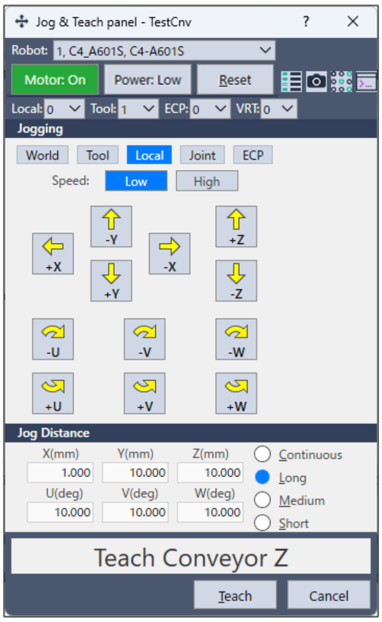
   Place a part in the robot motion range.

   Click [Jog & Teach].



6. The [Jog & Teach panel] will appear. Click the jog buttons to move the robot end effector to the pick position.

   Click the [Teach] button.



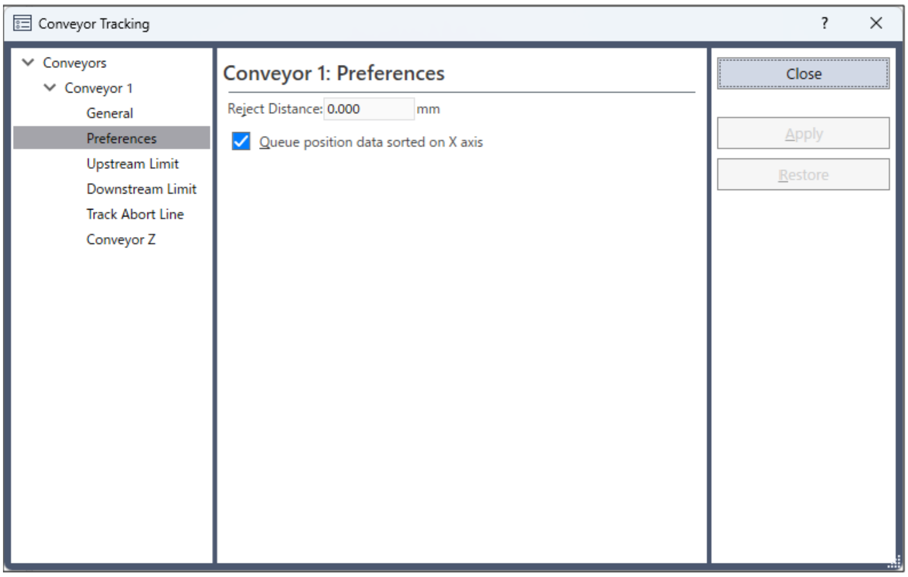
17.18 Queue Sorting

When you set the queue sorting, it registers the queue data in the order of position along the X axis in the conveyor local coordinate system. Set 0 or nothing for the index number of Cnv_QueGet command, the robot picks up parts from the downstream side.

To set the queue sorting

1. Select [Tools]-[Conveyor Tracking].

2. Click the conveyor you want to configure and select [Preferences].



3. Set the [Queue position data sorted on X axis] checkbox.

4. Click the [Apply] button.

KEY POINTS

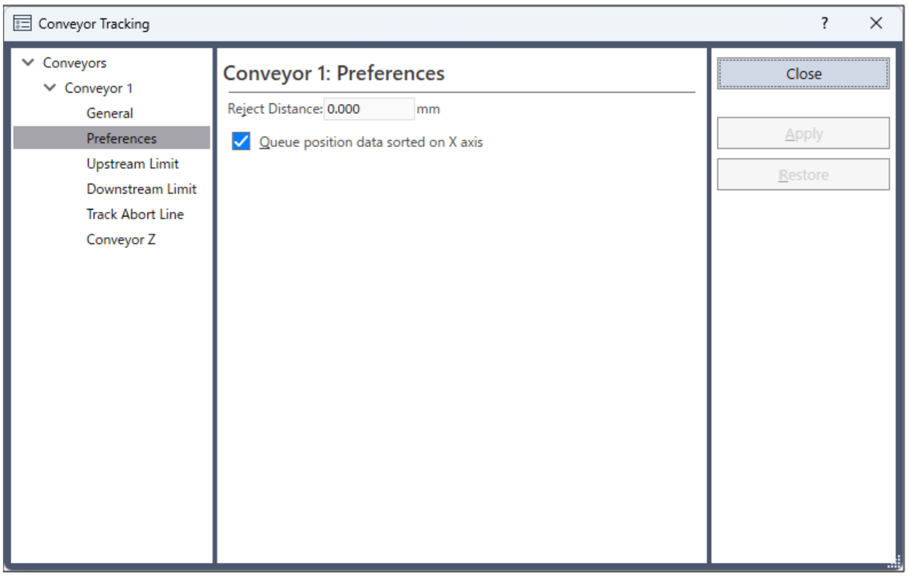
- When you set a diagonal upstream limit, register the queue data in the order of entering the Pickup Area.
- Also, when you set a diagonal upstream limit, note that the queue sorting cannot be canceled.
- The queue sorting function is applied to both upstream and downstream conveyors.

17.19 Double Registration Prevention

Cnv_QueReject avoids registering the same part doubly. If Cnv_QueReject value is not changed from the default (0 mm), the robot may perform the pickup motion at the position where the part is not placed, since the same part is registered to the queue several times.

Cnv_QueReject can be set by using the command or by following the steps below.

1. Select [Tools]-[Conveyor Tracking].
2. Click the conveyor you want to configure and select [Preferences].

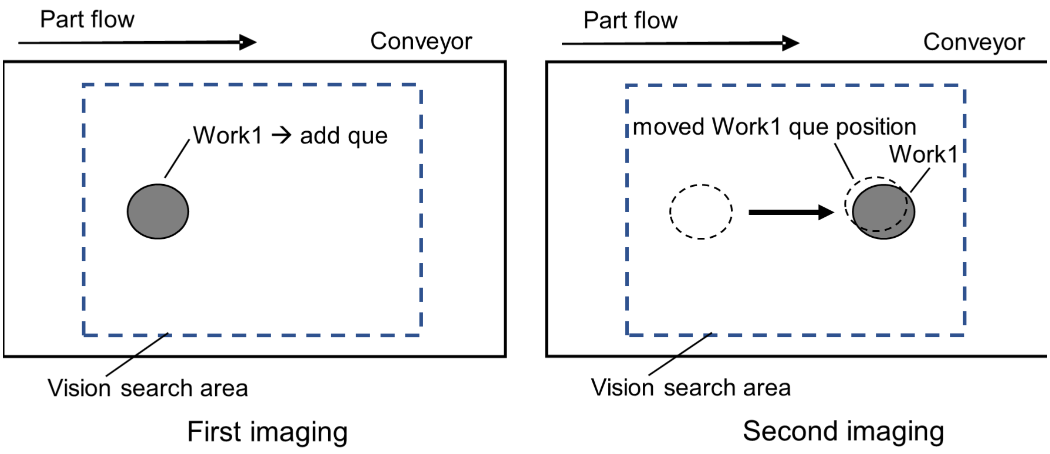


3. Set the value to [Reject Distance].
4. Click the [Apply] button.

KEY POINTS

If “Cnv_QueReject” is used in the program, the value set for “Cnv_QueReject” will be used instead of the value set in the above step.

When the same workpiece is imaged multiple times as shown in the figure below, the moved Work1 queue position which from the initially registered Work1 will not exactly match the coordinates of the newly registered Work1 queue due to the influence of the camera and conveyor etc. To prevent the same workpiece from being registered in the queue more than once, it is recommended to enter a value similar to the workpiece size as the double registration prevention distance.



17.20 Sample Program

Vision conveyor programming

Typically, two tasks are used to operate a vision conveyor.

One task finds parts with the vision system and adds them to the conveyor queue.

The other task checks for parts in the Pickup Area of the conveyor queue. When a part is in the Pickup Area, the robot is commanded to pick up the part and place it to the specified position.

The following example is a sample program that uses Xqt to execute two tasks from the "main" function.

- First task: "ScanConveyorStrobed" function
- Second task: "PickParts" function

A corresponding program is shown below.

Wiring Example of Vision Conveyor Tracking System

This sample program is a hardware trigger structure that uses controller I/O to trigger the camera and latch the encoder.

The following program is a sample with Conveyor #1.

This sample program automatically recovers when the robot tracks the workpiece that is out of the Pickup Area.

```
Function main
  Motor On
  Power High

  Speed 30
  Accel 30, 30

  Xqt ScanConveyorStrobed    'Task that registers queues
  Xqt PickParts              'Task that tracks parts (queue)
Fend

Function ScanConveyorStrobed
  Integer i, numFound, state, trigger
  Real x, y, u
  Boolean found
  trigger = 10               'Allocates pin10 of controller I/O
  Off trigger                'Turns OFF the camera trigger and encoder latch I/O
  Do
    VRun FindParts           'Searches for parts on the conveyor
```

```
    On trigger             'Turns ON the camera trigger and encoder latch I/O
    Do
      VGet FindParts.AcquireState, state
    Loop Until state = 3
    VGet FindParts.Parts.NumberFound, numFound
    'Registers the part that has been shot as a queue
    For i = 1 to numFound
      VGet FindParts.Parts.CameraXYU(i), found, x, y, u
      Cnv_QueAdd 1, Cnv_Point(1, x, y)
    Next i
    Off trigger            'Turns OFF the camera trigger and encoder latch I/O
    Wait 0.1
  Loop
Fend

Function PickParts
  OnErr GoTo ErrHandler
  Integer ErrNum
  Cnv_Mode 1,1             'Selects the tracking mode
  WaitParts:
  Do
    'Waits until a part (queue) enters the Pickup Area
    Wait Cnv_QueLen(1, CNV_QUELEN_PICKUPAREA) > 0
    'Starts tracking the part
    'When using the SCARA robot
    Jump Cnv_QueGet(1)
    Wait 0.1               'The robot moves at the same speed as the conveyor for
the Wait time specified to this part
    Jump P1                'Moves the picked part to a specified place
    Cnv_QueRemove 1, 0           'Clears the picked part (queue)
  Loop
  'Clears the parts (queue) in the downstream side from the Pickup Area
  'Automatically recovers from the error '"The specified queue data is outside
the set area"
  ErrHandler:
    ErrNum = Err
    If ErrNum = 4406 Then
      Cnv_QueRemove 1, 0
      EResume WaitParts
'Displays an error other than "The specified queue data is outside the set area"
    Else
      Print "Error!"
      Print "No.", Err, ":", ErrMsg$(Err, 1)
      Print "Line :", Erl(0)
      'User error occurred
      Error 8000
    EndIf
Fend
```

KEY POINTS

When you use software trigger, use the "ScanConveyorStrobed" function shown below.

```
Function ScanConveyorNonStrobed
  Integer i, numFound
  Real x, y, u
  Boolean found
  Do
    'Searches for parts on the conveyor
    VRun FindParts
    Cnv_Trigger 1           'Latches the encoder with software trigger
    VGet FindParts.Parts.NumberFound, numFound
    'Registers the part as a queue
    For i = 1 to numFound
      VGet FindParts.Parts.CameraXYU(i), found, x, y, u
      Cnv_QueAdd 1, Cnv_Point(1, x, y)
    Next i
    Wait 0.1
  Loop
Fend
```

Sensor conveyor programming

Typically, two tasks are used to operate a sensor conveyor. One task waits for a part to trip the sensor and add it to the conveyor queue. The other task checks for parts in the Pickup Area of the conveyor queue. When a part is in the Pickup Area, the robot is commanded to pick up the part and place it to the specified position.

This sample program automatically recovers when the robot tracks the workpiece that is out of the Pickup Area.

```
Function main
  Motor On
  Power High

  Speed 30
  Accel 30, 30

  Xqt ScanConveyor               'Task that registers queues
  Xqt PickParts                  'Task that tracks parts (queue)
Fend

Function ScanConveyor
  Double lpulse1                 'Previous latch pulse
    lpulse1 = Cnv_LPulse(1)      'Registers the latch pulse as lpulse1
    Do
      'Registers a part as a queue only when it passes the sensor
      If lpulse1 <> Cnv_LPulse(1) Then
        Cnv_QueAdd 1, Cnv_Point(1, 0, 0)
        lpulse1 = Cnv_LPulse(1)    'Updates lpulse1
      EndIf
    Loop
Fend

Function PickParts
  OnErr GoTo ErrHandler
  Integer ErrNum
  Cnv_Mode 1,1             'Selects the tracking mode
  WaitParts:
  Do
    'Waits until a part (queue) enters the Pickup Area
    Wait Cnv_QueLen(1, CNV_QUELEN_PICKUPAREA) > 0
```

```
    'Starts tracking the part
    'When using the SCARA robot
    Jump Cnv_QueGet(1)
    Wait 0.1                    'The robot moves at the same speed as the
conveyor for the Wait time specified to this part
    Jump P1                     'Moves the picked part to a specified place
    Cnv_QueRemove 1, 0          'Clears the picked part (queue)
  Loop
  'Clears the parts (queue) in the downstream side from the Pickup Area
  'Automatically recovers from the error '"The specified queue data is outside
the set area"
  ErrHandler:
    ErrNum = Err
    If ErrNum = 4406 Then
      Cnv_QueRemove 1, 0
      EResume WaitParts
'Displays an error other than "The specified queue data is outside the set area"
    Else
      Print "Error!"
      Print "No.", Err, ":", ErrMsg$(Err, 1)
      Print "Line :", Erl(0)
      'User error occurred
      Error 8000
    EndIf
Fend
```

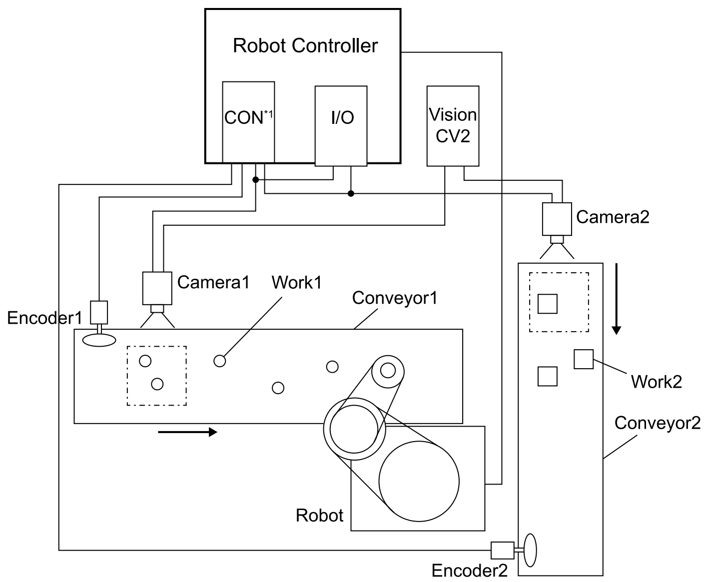
17.21 Multiple Conveyors

Epson RC+ supports multiple logical conveyors and robots. You can use multiple robots with one conveyor.

This section describes a conveyor system that uses one robot with two or more conveyors.

Conveyor tracking for several conveyors

This section describes a conveyor system where one robot picks up "Part 1" from Conveyor 1 and puts the picked parts above "Parts 2" on Conveyor 2 as shown in the figure below. In this conveyor system, each conveyor needs one encoder and camera (sensor).



*1: Connector for conveyor tracking or PG Board

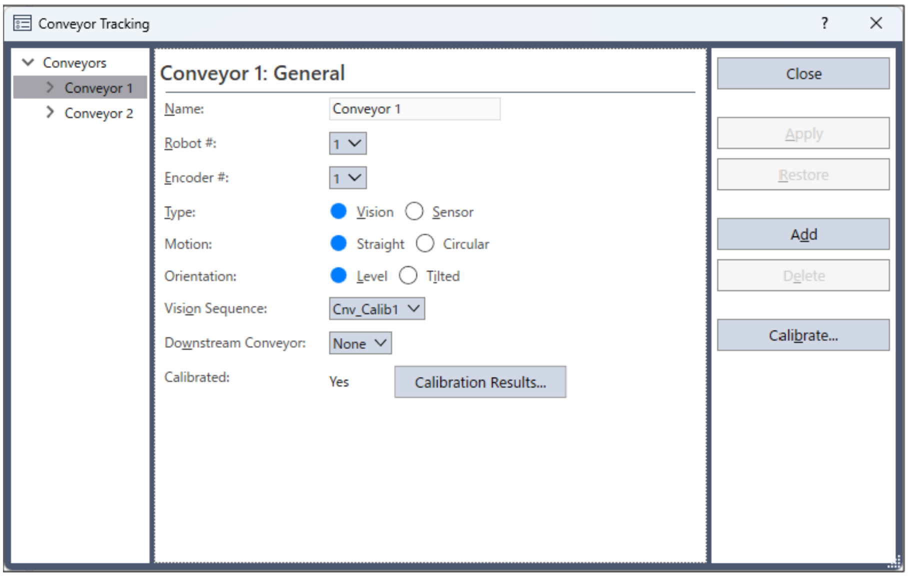
How to use several conveyors

Usage is described below.

1. See and create conveyors 1 and 2. (Set the robot in the upstream side to the Conveyor 1.)

   Creating Conveyors in a Project

2. For the [Encoder] and [Vision Sequence], set the different number and sequence for each conveyor 1 and 2.



3. Calibrate Conveyor 1.

4. Check the operation while referring to either of the following:
   - **Vision Conveyors**
   - **Sensor Conveyors** - "Check the motion"

5. Calibrate Conveyor 2.

6. Check the operation of Conveyor 2.

A sample program is shown below.

This sample program automatically recovers when the robot tracks the workpiece that is out of the Pickup Area.

```
Function main
  Motor On
  Power High

  Speed 30
  Accel 30, 30

  Xqt ScanConveyorStrobed  'Task that registers queues
  Xqt PickParts            'Task that tracks parts (queue)
Fend
Function ScanConveyorStrobed
  Integer i, j, numFound, state, trigger1, trigger2
  Real x, y, u
  Boolean found

  trigger1 = 10 'Allocates pin10 of controller I/O to conveyor 1
  trigger2 = 11 'Allocates pin11 of controller I/O to conveyor 2
  Off trigger1; Off trigger2
  'Turns OFF the camera trigger and encoder latch
  Do
    VRun FindParts1     'Searches for parts on the conveyor
```

```
    On trigger1            'Turns ON the camera trigger and encoder latch I/O
    Do
      VGet FindParts1.AcquireState, state
    Loop Until state = 3
    VGet FindParts1.Parts.NumberFound, numFound
    'Registers the part that has been shot as a queue
    For i = 1 To numFound
      VGet FindParts1.Parts.CameraXYU(i), found, x, y, u
      Cnv_QueAdd 1, Cnv_Point(1, x, y)
    Next i
    Off trigger1           'Turns OFF the camera trigger and encoder latch I/O
    Wait 0.1

    'Registers the parts (queue) of the Conveyor 2
    'Searches for parts on the conveyor
    VRun FindParts2
    On trigger2            'Turns ON the camera trigger and encoder latch I/O
    Do
      VGet FindParts2.AcquireState, state
    Loop Until state = 3
    VGet FindParts2.Parts.NumberFound, numFound
    'Registers the part that has been shot as a queue
    For j = 1 to numFound
      VGet FindParts2.Parts.CameraXYU(j), found, x, y, u
      Cnv_QueAdd 2, Cnv_Point(2, x, y)
    Next j
    Off trigger2           'Turns OFF the camera trigger and encoder latch I/O
    Wait 0.1
  Loop
Fend
Function PickParts
  OnErr GoTo ErrHandler
  Integer ErrNum

  MemOff 1;  MemOff 2      'Turns OFF the memory I/O

  Jump P1

  Do
    'Tracking of the Conveyor 1
    WaitPickup1:
    'Turns ON the memory I/O when the Conveyor 1 tracking phase starts
    MemOn 1                'Turns ON the memory I/O 1
    'Clears the parts (queue) in the downstream side from the downstream limit
    Do While Cnv_QueLen(1, CNV_QUELEN_DOWNSTREAM) > 0
      Cnv_QueRemove 1, 0
    Loop
    'Waits until a part (queue) enters the Pickup Area
    Wait Cnv_QueLen(1, CNV_QUELEN_PICKUPAREA) > 0

    'Starts tracking the part
    'When using the 6-axis robot
    Jump3 Cnv_QueGet(1):Z(0):U(90):V(0):W(180)
    'When using the SCARA robot
    Jump Cnv_QueGet(1)
    Wait 0.1 'The robot moves at the same speed as the conveyor for the Wait time
specified to this part
    'Clears the picked part (queue)
    Cnv_QueRemove 1, All
    MemOff 1               'Turns OFF the memory I/O 1

    'Tracking of the Conveyor 2
    WaitPickup2:
    MemOn 2                'Turns ON the memory I/O 2
    'Waits until a part (queue) enters the Pickup Area
```

```
    Wait Cnv_QueLen(2, CNV_QUELEN_PICKUPAREA) > 0

    'Starts tracking the part
    'When using the 6-axis robot
    Jump3 Cnv_QueGet(2):Z(0):U(90):V(0):W(180)
    'When using the SCARA robot
    Jump Cnv_QueGet(2)          Wait 0.1 'The robot moves at the same speed as
the conveyor for the Wait time specified to this part
    Wait 0.1 'The robot moves at the same speed as the conveyor for the Wait time
specified to this part
    'Clears the picked part (queue)
    Cnv_QueRemove 2, All
    MemOff 2               'Turns OFF the memory I/O 2

    Jump P1
  Loop
  'Clears the parts (queue) in the downstream side from the Pickup Area
  'Automatically recovers from the error '"The specified queue data is outside
the set area"
  ErrHandler:
  ErrNum = Err
  If ErrNum = 4406 Then
    If MemSw(1) = On Then
      Cnv_QueRemove 1
      EResume WaitPickup1
    EndIf
    If MemSw(2) = On Then
      Cnv_QueRemove 2
      EResume WaitPickup2
    EndIf
    'Error other than "The specified queue data is outside the set area"
    'Is displayed
  Else
    Print "Error!"
    Print "No.", Err, ":", ErrMsg$(Err, 1)
    Print "Line :", Erl(0)
    'User error occurred
    Error 8000
  EndIf
Fend
```

17.22 Multi-robot Conveyor

Epson RC+ supports multiple logical conveyors and robots. You can use multiple robots with one conveyor, or multiple robots with multiple conveyors.

This section describes a conveyor system that uses two or more robots with one conveyor and a conveyor system that uses one robot with two or more conveyors.

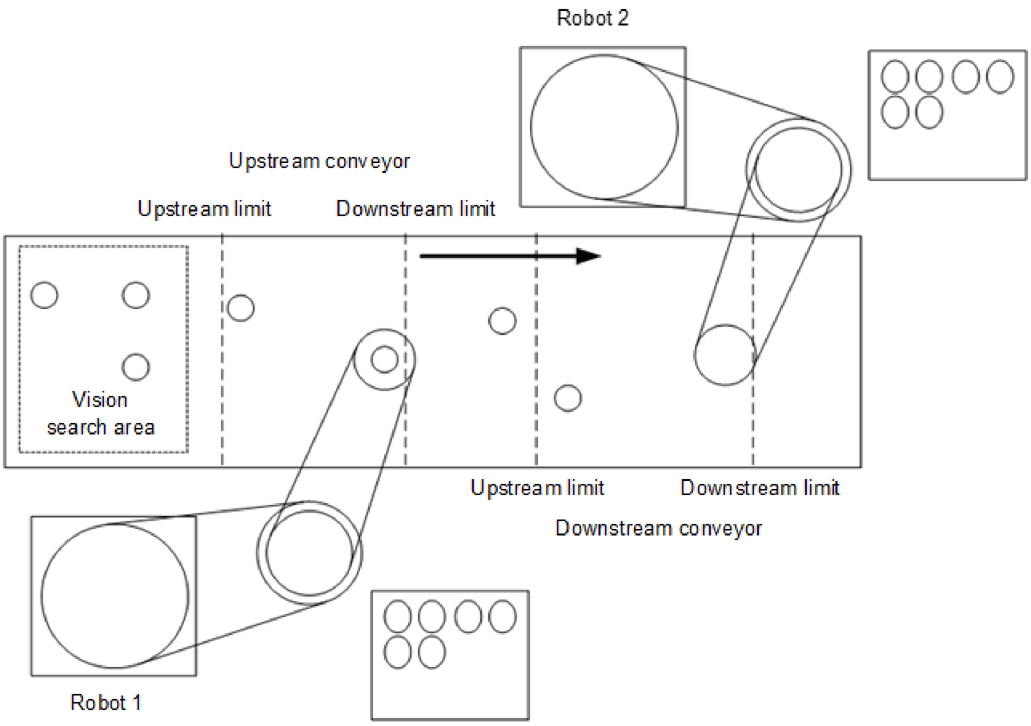
For details on multi robots, refer to the following manual.

"Robot Controller Manual"

Multi-robot conveyor

The multi-robot system uses two or more robots with one conveyor as shown below. In this system, the second robot (downstream) picks up the parts that the first robot (upstream) failed to pick up.

Although the system uses several robots, it uses only one camera (sensor), encoder, and conveyor.



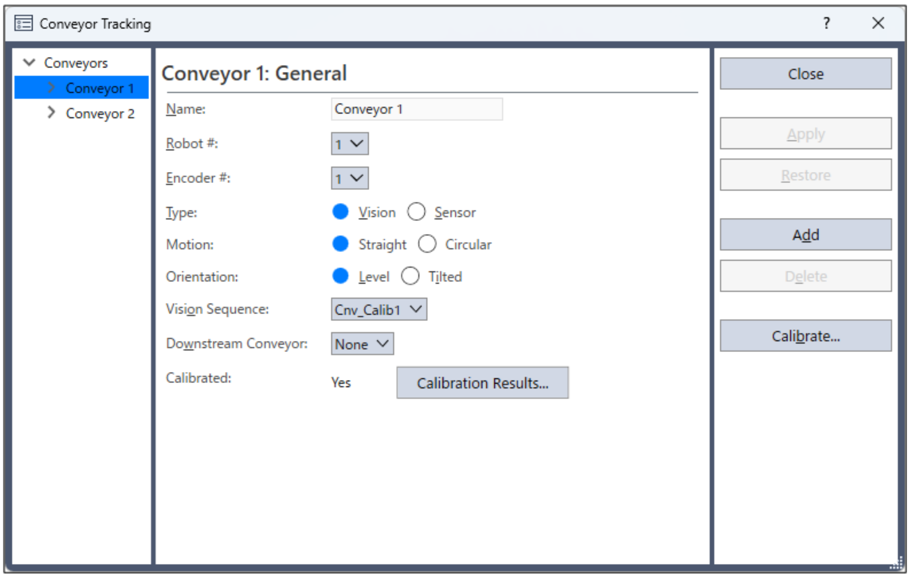
How to use multi-robot conveyor

To use the multi-robot conveyors, set the upstream and downstream conveyors. Instructions for using multi-robot conveyors are described below.

1. See and create conveyors 1 and 2. (Set the upstream-side robot to Conveyor 1.)

   Creating Conveyors in a Project

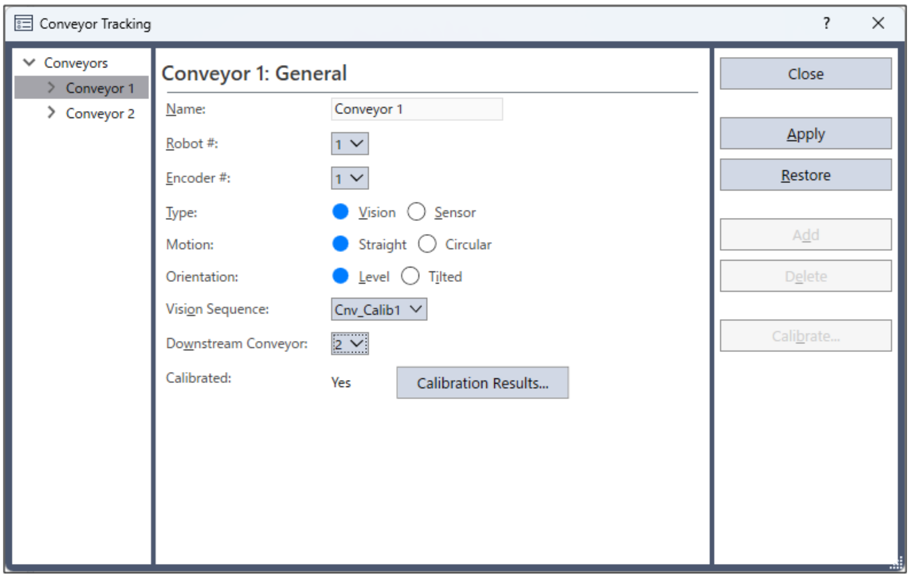
2. For [Encoder] and [Vision Sequence], set the same number and sequence for both conveyors 1 and 2.



3. Calibrate Conveyor 1.

4. Check the operation while referring to either of the following:
   - **Vision Conveyors**
   - **Sensor Conveyors** - "Check the motion"

5. Set [Downstream Conveyor] to “2”.



6. Calibrate Conveyor 2.

7. Check the operation of Conveyor 2.

   i. Clear the all queue data registered to each conveyor.

```
>Cnv_QueRemove 1,all
>Cnv_QueRemove 2,all
```

   ii. Place the part in the vision search area.

   iii. Execute the program “ScanConveyorStrobed(ScanConveyor)” and register a queue.

   iv. Halt the program “ScanConveyorStrobed” and move the conveyor until the part enters the Pickup Area.

   v. Stop the program “ScanConveyorStrobed” and move the conveyor until the part enters the Pickup Area of the conveyor 2.

   vi. Execute the command below to move the queue from conveyor 1 to conveyor 2.

```
>Cnv_QueMove 1,0
```

   vii. Pick up the part.

```
>Jump Cnv_Queget(2)
```

   viii. Check if the robot end effector is above the center of a part. If the robot end effector is not above the center of the part, perform the calibration again.

   ix. Move the conveyor and check if the robot follows the part. At this point, the end effector will be off the center of part but this is not a problem.

   x. Stop the tracking motion.

~~~
```
>Cnv_AbortTrack
```
~~~

8. A sample program is shown below.

```
Function main
  Xqt ScanConveyorStrobed  'Task that registers queues
  Xqt PickParts1 'Task for the upstream robot to track the parts (queue)
  Xqt PickParts2 'Task for the downstream robot to track the parts (queue)
Fend

Function ScanConveyorStrobed
  Integer i, numFound, state, trigger
  Real x, y, u
  Boolean found

  trigger = 10 'Allocates pin10 of controller I/O
  Off trigger 'Turns OFF the camera trigger and encoder latch I/O
  Do

    'Searches for parts on the conveyor
    VRun FindParts
    On trigger  'Turns ON the camera trigger and encoder latch I/O
    Do
      VGet FindParts.AcquireState, state
    Loop Until state = 3
    VGet FindParts.Parts.NumberFound, numFound
    'Registers the found part to the queue of conveyor 1
    For i = 1 To numFound
      VGet FindParts.Parts.CameraXYU(i), found, x, y, u
      Cnv_QueAdd 1, Cnv_Point(1, x, y)
    Next i
    Off trigger 'Turns OFF the camera trigger and encoder latch I/O
    Wait 0.1
  Loop
Fend

Function PickParts1
  OnErr GoTo ErrHandler
  Integer ErrNum

  Robot 1
  Motor On
  Power High
  Speed 30
  Accel 30, 30

  Jump P1

 WaitParts:
  Do
    'Waits until a part (queue) enters the Pickup Area
    Wait Cnv_QueLen(1, CNV_QUELEN_PICKUPAREA) > 0

    'Starts tracking the part
    'When using the 6-axis robot
     Jump3 Cnv_QueGet(1):Z(0):U(90):V(0):W(180)
    'When using the SCARA robot
    Jump Cnv_QueGet(1)
    Wait 1 'The robot moves at the same speed as the conveyor for the Wait time
specified to this part
    Jump P1 'Moves the picked part to a specified place
    'Clears the picked part (queue)
    Cnv_QueMove 1, 0
  Loop
  'Moves the parts (queue) in the downstream side than the Pickup Area of
conveyor1 'to the conveyor 2
  ErrHandler:
```

```
    ErrNum = Err
    If ErrNum = 4406 Then
      Cnv_QueRemove 1, 0
      EResume WaitParts
      'When an error except the conveyor tracking motion range error occurs,
      'Error is displayed
    Else
      Print "Error!"
      Print "No.", Err, ":", ErrMsg$(Err, 1)
      Print "Line :", Erl(0)
      'User error occurred
      Error 8000
    EndIf
Fend

Function PickParts2
  OnErr GoTo ErrHandler
  Integer ErrNum

  Robot 2
  Motor On
  Power High
  LoadPoints "robot2.pts"
  Speed 30
  Accel 30, 30

  Go P1

  WaitParts:
  Do
    'Waits until a part (queue) enters the Pickup Area
    Wait Cnv_QueLen(2, CNV_QUELEN_PICKUPAREA) > 0
    'Starts tracking the part
    'When using the 6-axis robot
    Jump3 Cnv_QueGet(2):Z(0):U(90):V(0):W(180)
    'When using the SCARA robot
    Jump Cnv_QueGet(2)
    Wait 1 'The robot moves at the same speed as the conveyor for the Wait time
specified to this part
    Jump P2 'Moves the picked part to a specified place
    'Clears the picked part (queue)
    Cnv_QueRemove 2, 0
  Loop

  'Clears the picked part (queue)

  'Clear the parts (queue) in the downstream side than the Pickup Area of
conveyor 2
  'Automatically recovers from the error '"The specified queue data is outside
the set area"
  ErrHandler:
  ErrNum = Err
  If ErrNum = 4406 Then
    Cnv_QueRemove 2, 0
    EResume WaitParts
    'Error other than "The specified queue data is outside the set area"
    'is displayed
  Else
    Print "Error!"
    Print "No.", Err, ":", ErrMsg$(Err, 1)
    Print "Line :", Erl(0)
    'User error occurred
    Error 8000
```

```
  EndIf
Fend
```

17.23 Abort Tracking

There are some situations when you want to abort tracking a part that moves out of the Pickup Area while the robot is tracking the part. In this case, use the Cnv_AbortTrack command in a separate task that monitors the conveyor queue.

```
Function MonitorDownstream
  Robot 1
  Do
    If Cnv_QueLen(1, CNV_QUELEN_DOWNSTREAM) > 0 Then
      Cnv_AbortTrack 0
    EndIf
    Wait .1
  Loop
Fend
```

17.24 Conveyor Tracking with 6-Axis Robot

When you use a 6-axis robot in a conveyor tracking system, you need to set the values of U, V, and W.

For this, use the Cnv_QueGet command.

The following shows the case where the robot end effector moves toward a part during the pickup.

```
Go Cnv_Queget(Conveyor number,[Index]):U(90):V(0):W(180)
```

An example of using Jump3 is shown below.

```
Jump3 P1,Cnv_QueGet(1):Z(**):U(90):V(0):W(180),
Cnv_QueGet(1):U(90):V(0):W(180)
```

KEY POINTS

When setting the Z(**) height, be aware of the following:

- Home position of Z in the tracking coordinate is the calibration position.
- To raise the Z height in the tracking coordinate, offset in a positive (+) direction.
- To lower the Z height in the tracking coordinate, offset in a minus (-) direction.
- Robot coordinate P1 can be converted to the conveyor coordinate and displayed.

  ```
  > print P1@cnv1
  ```

17.25 Tracking Mode

There are three tracking modes: quantity-priority mode accuracy-priority mode, and variable speed conveyor mode. The mode can be selected by the Cnv_Mode command.

Tracking mode selection is only available for linear conveyors. For circular conveyors, quantity-priority mode is only available.

17.25.1 Quantity-priority mode

Quantity-priority mode prioritizes reducing time to catch up with the workpiece (queue) over the picking accuracy. This mode is suitable for the conveyor tracking system in which space between the workpieces is narrow.

> **KEY POINTS**
>
> When the quantity-priority mode is selected, tracking delay (situation in which the Manipulator does picking motion at the posterior part of the workpiece to the direction of the conveyor motion) may occur. When you correct the tracking delay, write the program to add the offset as follows.
>
> ```
> Go Cnv_Queget(Conveyor number,[Index])+X(**)
> ```
>
> X indicates the direction of the conveyor motion. ** indicates correction amount [mm]. For the details of offset, refer to the following manual:
>
> "SPEL"+ Language Reference - Go"

17.25.2 Accuracy-priority mode

Accuracy-priority mode is the mode to correct the tracking delay. Accuracy-priority mode improves the picking accuracy while it takes more time to catch up with the workpiece. This mode is suitable for the conveyor tracking system for small workpieces.

When using the accuracy-priority mode, it is recommended to perform the tracking delay acquisition operation in advance. For details, see the following points.

Accuracy-priority mode - Acquiring tracking delay

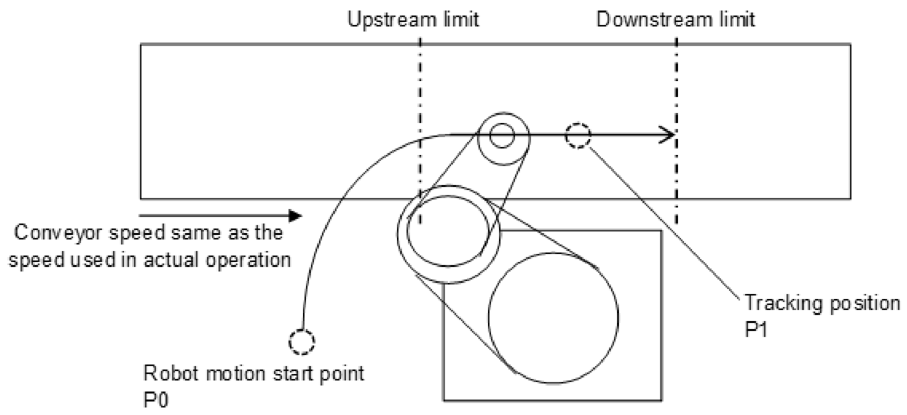
17.25.3 Accuracy-priority mode - Acquiring tracking delay

When using the conveyor tracking accuracy priority mode, you can acquire and compensate for robot tracking delays to accurately track a workpiece.

The following tracking delay acquisition operation should be performed in advance to acquire tracking delays. Failing to set the offset value in advance may adversely affect accuracy and elapsed time.

Acquiring the tracking delay

When acquiring the tracking delay, the robot moves from the motion start point to the downstream limit as shown in the diagram below. This acquires the offset amount [mm] in the direction the conveyor is traveling.



This motion can be executed by running the following sample program.

Configure the following settings beforehand.

- Perform conveyor calibration
- Set Accel, Speed, Tool, and other parameters to the same settings as that used for actual operation
- Teach the robot the start position for the conveyor tracking motion as P0
- Teach the robot the following two points required to perform the motion
  - Point 1: The robot motion start point
  - Point 2: The point tracked on the conveyor
- Move the conveyor at the same speed as that used for actual operation

KEY POINTS

- The tracking delay is acquired based on the conveyor width-direction and height coordinates at the tracking position taught to the robot. Configure settings to closely match actual operation.
- The offset value changes based on the conveyor speed, robot acceleration, speed, and orientation. Therefore, the tracking delay should be reacquired when any of these settings are changed.
- As this program runs on a virtual queue, an actual workpiece is not required.

```
Function Cnv_Adjust_measure

  'Move to robot motion start point
  Motor On
  Go P0

  Power High
  Speed 100
  Accel 100, 100
  Cnv_Accel 1, 2000

  'Check conveyor operation
  If Cnv_Speed(1) < 0.1 Then
    Print "Conveyor not operational"
    Exit Function
  EndIf

  'Register virtual workpiece to queue
  Cnv_QueRemove 1, All                   'Clear queue
  Cnv_Trigger (1)                        'Latch conveyor pulse
  Cnv_QueAdd 1, XY(0, CY(P1@Cnv(1)), CZ(P1@Cnv(1)), CU(P1@Cnv(1)), 0, 0) /CNV(1)
```

```
   'Register virtual workpiece to queue based on P1

  Wait Cnv_QueLen(1, CNV_QUELEN_PICKUPAREA) > 0
  ' Wait until queue enters the Pickup Area
  Cnv_Adjust 1, On                       'Turn offset acquisition flag On

  'Perform motion
  Go CnvQueGet(1,0)                     'For SCARA robots
  'Go CnvQueget(1,0):U(90):V(0):W(180)  ' For 6-axis robots

  Do
  Wait 0.02
  Loop Until (CX(RealPos@CNV(1)) >= Cnv_Downstream(1))
  ' Wait until downstream limit is reached

  Go here                                'Stop robot
  Cnv_QueRemove 1, All                   'Reset queue
  Cnv_Adjust 1, Off                      'Turn offset acquisition flag Off
  motor off

  'Output results of offset value acquired
  If Cnv_AdjustGet(1, 0) = 2 then
  Print "Unable to correctly acquire offset value"
  Else
    Print "Operation results=", Cnv_AdjustGet(1, 0)
    Print "Offset amount =", Cnv_AdjustGet(1, 1)

  EndIf
Fend
```

KEY POINTS

- The offset value is reset when turning the controller off. Use Cnv_AdjustSet to set the offset value acquired within the program in use. To store the offset value acquired, add the following program to save the file to the Project folder.

```
Integer fileNum; String filename$
fileNum = FreeFile
filename$ = "File name"
AOpen filename$ As #fileNum
Print #fileNum,Cnv_AdjustGet(1,1)
Close #fileNum
```

- If the operation results are not "1", the robot may not have been able to reach the workpiece within the Pickup Area, and offset value acquisition may have exceeded 100 seconds. Check the Accel and Speed settings, upstream and downstream limit settings, the motion start point, and the conveyor speed.

17.25.4 Variable speed conveyor mode

The variable speed conveyor mode compensates for the delay in robot tracking in response to changes in conveyor speed.

It is suitable for conveyor systems where the conveyor may stop and continue is in progress while contact is made with the workpiece. For example, screw tightening applications are conceivable.

When stopping and continuing the conveyor in variable speed conveyor mode, set the limit of acceleration and deceleration after conveyor tracked and it is recommended that the operation used to determine the offset value be performed in advance. For details, see the next section, "Setting the offset value for variable speed conveyor mode".

> **KEY POINTS**
>
> If any of the following processes are performed when the conveyor is being stopped or while it is stopped, conveyor tracking will be terminated and cannot be continued.
>
> - Open the safety door connected to the controller.
> - Press the Emergency Stop button connected to the controller.
> - Press the Pause button or execute the Pause command.

17.25.5 Setting the offset value, acceleration and deceleration for variable speed conveyor mode

The robot's tracking delay relative to conveyor speed changes depends on the conveyor speed, conveyor acceleration/deceleration, robot model used, Inertia setting, Weight setting, and other factors.

Therefore, it is necessary to set the offset value according to the operating environment, and the limit of acceleration and deceleration after conveyor tracked. In order to improve the robot's tracking delay, it is necessary to adjust the offset value and set the optimum value.

The amount of robot tracking delay is acquired with the Cnv_PosErr function, the offset value is set with Cnv_PosErrOffset, and the limit of acceleration and deceleration after conveyor tracked is set with Cnv_AccelLim.

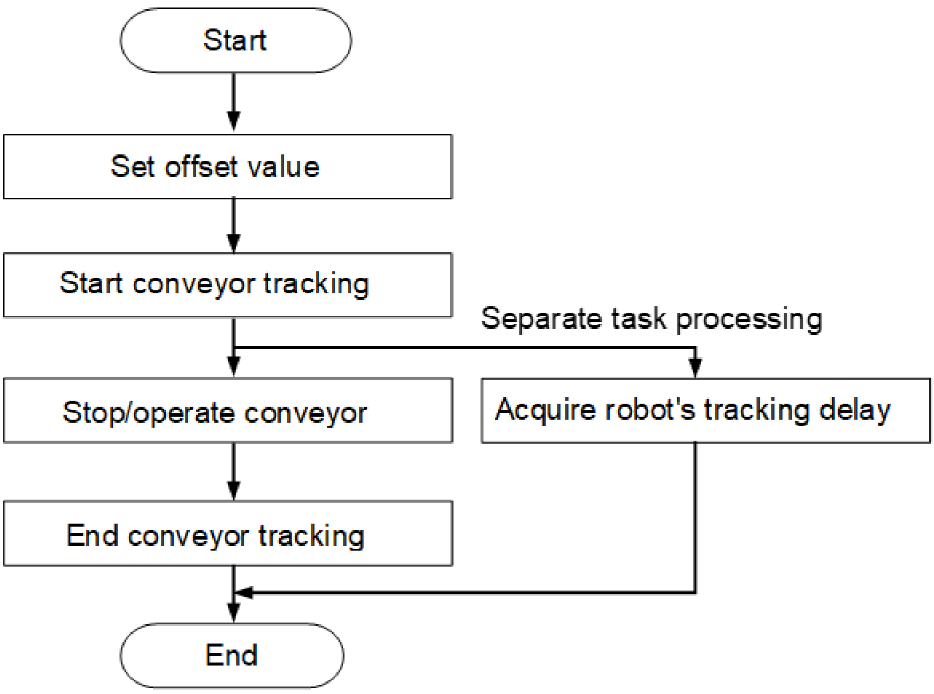
> **KEY POINTS**
>
> When adjusting the offset value, set Cnv_Fine to “0”. If the value is not “0”, the amount of tracking delay cannot be acquired correctly with the Cnv_PosErr function.
>
> In the quantity-priority mode and accuracy-priority mode, it is not possible to acquire and set offset values.
>
> When acquiring the offset value, check the accuracy-priority mode - Acquiring tracking delay" setting. For a working program, use the sample program described below.

Set the limit to a value greater than the conveyor acceleration and deceleration when stopping or operating the conveyor. A rough guideline for the set value is about twice the conveyor acceleration and deceleration.

If the limit is too high, robot motion gets oscillatory due to variation of the conveyor speed or noise. If it is lowered too much, the robot will not stop tracking the conveyor even it stops, and it may move out of the operating area of the robot. In that case, set a tracking abort line or program to stop tracking at the downstream limit.

After setting the limit of acceleration and deceleration, follow the procedure below to set the offset value and acquire the robot's tracking delay.

In order to find the optimal offset value, change the offset value and perform multiple times.




```
Integer fileNum                         'Declare a file number
Function Cnv_PosErr_measure
  Motor On
  Go P0                                 'Move to robot motion start point
  Power High
  Speed 100
  Accel 100, 100
  Cnv_Accel 1, 2000
  Cnv_Fine 1,0                          'Fine setting
  Cnv_Mode 1,2                          'Variable speed conveyor mode
  Cnv_PosErrOffset 1,10                 'Set offset value
  ' Check conveyor operation
  If Cnv_Speed(1) < 0.1 Then
    Print "Conveyor not operational"
    Exit Function
  EndIf

  ' Register virtual workpiece to queue
  Cnv_QueRemove 1, All                  'Clear queue
  Cnv_Trigger (1)                       'Latch conveyor pulse
  Cnv_QueAdd 1, XY(0, CY(P1@Cnv(1)), CZ(P1@Cnv(1)), CU(P1@Cnv(1)), 0, 0) /CNV(1)
   'Register virtual workpiece to queue based on P1

  Wait Cnv_QueLen(1, CNV_QUELEN_PICKUPAREA) > 0
  ' Wait until queue enters the Pickup Area
  Xqt CnvPosErrTest                     'Start acquiring offset values in
separate task

  'Perform motion
  Go Cnv_QueGet(1,0)                    'For SCARA robots
  'Go Cnv_Queget(1,0):U(90):V(0):W(180)  ' For 6-axis robots

  Do
    Wait 0.02
  Loop Until (CX(RealPos@CNV(1)) >= Cnv_Downstream(1))
  ' Wait until downstream limit is reached
  Go here                               'Stop robot
  Cnv_QueRemove 1, All                  'Reset queue
  Wait 0.5
  Quit CnvPosErrTest                    'End offset value acquisition
```

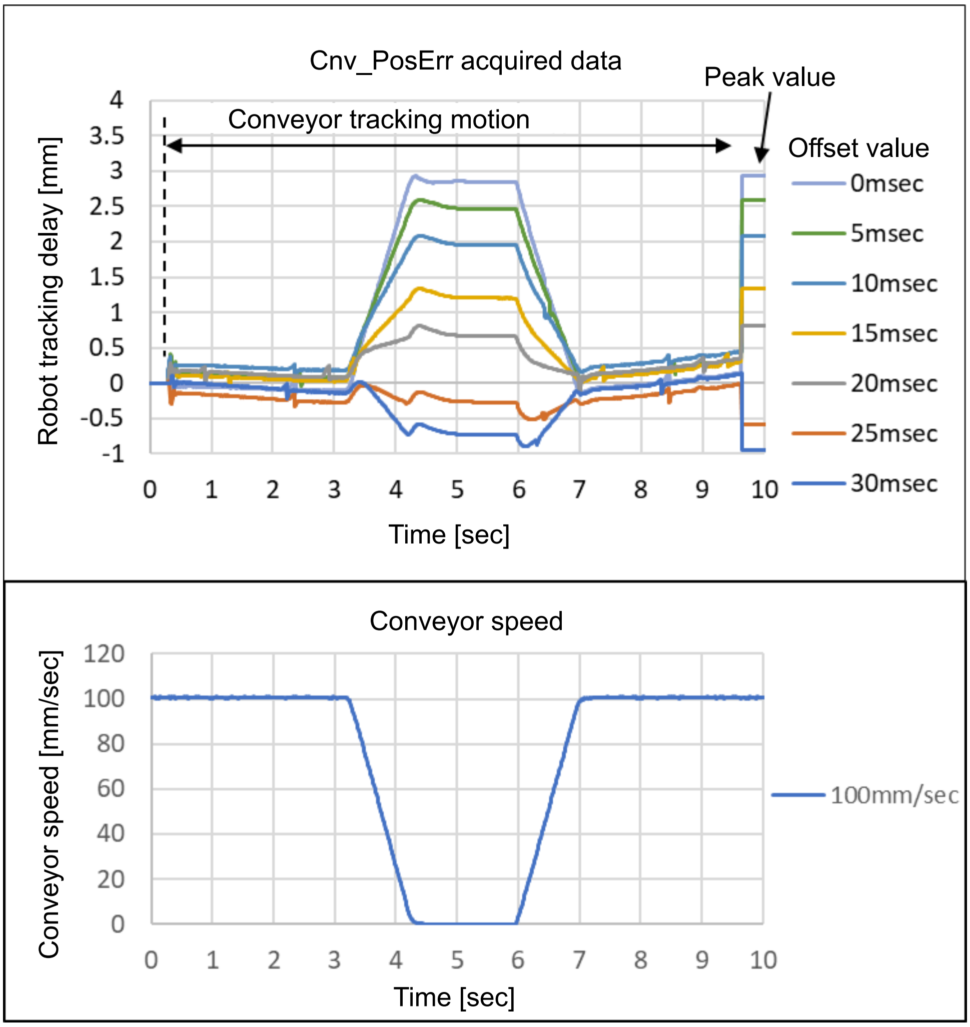
```
  motor off

Fend

Function CnvPosErrTest
  fileNum = FreeFile                      'Acquire a file number
  WOpen "poserr.csv" As fileNum          'csv filename
  Print #fileNum, "Time[sec],Cnv_PosErr[mm],Cnv_Speed[mm/s]"
  TmReset 0
  Do
    Print #fileNum, Tmr(0), ",", Cnv_PosErr(1), ",",
    Cnv_Speed(1)
    Wait 0.01
  Loop
Fend
```

A "poserr csv" file will be created in the project folder of Epson RC+ 8.0. Open the file in a spreadsheet or other software and create a line graph or scatter plot.

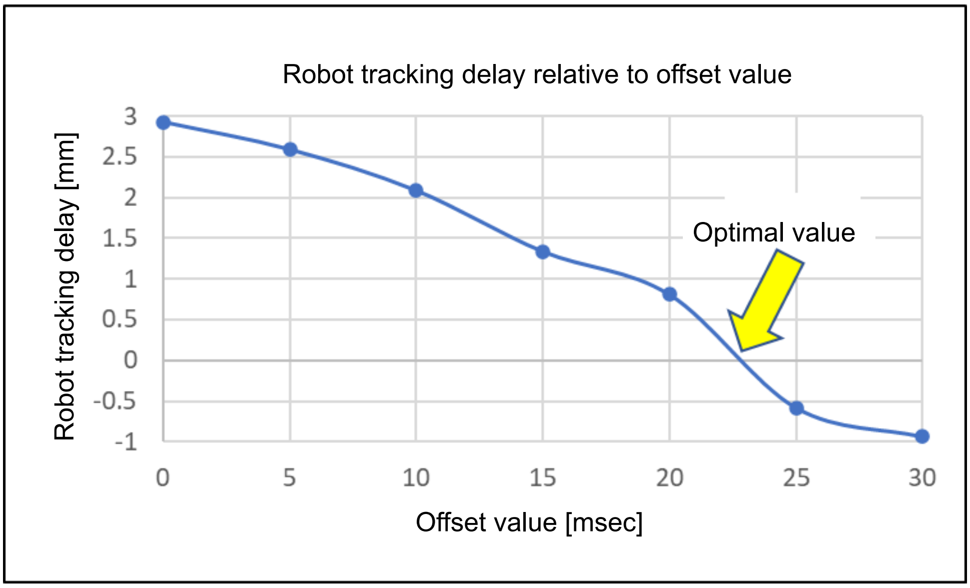
If data is acquired multiple times with different offset values, a graph like that shown below can be produced.

After a conveyor tracking operation, the return value of the Cnv_PosErr function will be the Cnv_PosErr peak value acquired during said operation. It is useful to use these peak values when creating graphs.

A positive tracking delay means that the robot is advancing too far downstream from the workpiece.



Plotting the robot's tracking delay against the offset value allows the optimal offset value to be set.



From the results shown above, add the following offset value setting to the conveyor tracking program:

```
Cnv_PosErrOffset 1, 22.7  'Offset value
```

CAUTION

- The offset values described in this procedure are for reference. Depending on the operating environment and the offset value that is set, the operation may not be successful or may cause vibration.
- If the robot performs an action different from the setting, immediately press the emergency stop button.

17.26 How to shorten the picking cycle time

As examples, the following methods can be used to shorten the picking cycle time:

- -Use the Arch command
- -Use the Cnv_Accel command

KEY POINTS

The followings are the points to consider when using the Cnv_Accel command.

- Maximum Cnv_Accel value is 5000 mm/s^2.
- If the Cnv_Accel setting value is 0 or exceeds 5001, the default value (2000 mm/s^2) will be set.
- If the acceleration error occurs, greater Cnv_Accel value cannot be specified. Decrease the Cnv_Accel value or decrease Accel or AccelS.

17.27 Manipulator Posture

Manipulator posture during the tracking motion is always the default posture regardless of the posture at the conveyor tracking calibration. To specify the posture for the tracking, write a program as shown in the following example.

Example: tracks the workpiece with Lefty arm position

```
jump Cnv_Queget(Conveyor number,[Index])/ L
```

KEY POINTS

During tracking motion, singularity avoiding function cannot be used. Therefore, set the positions of the Manipulator and the conveyor so that the Manipulator does not pass through the singularity.

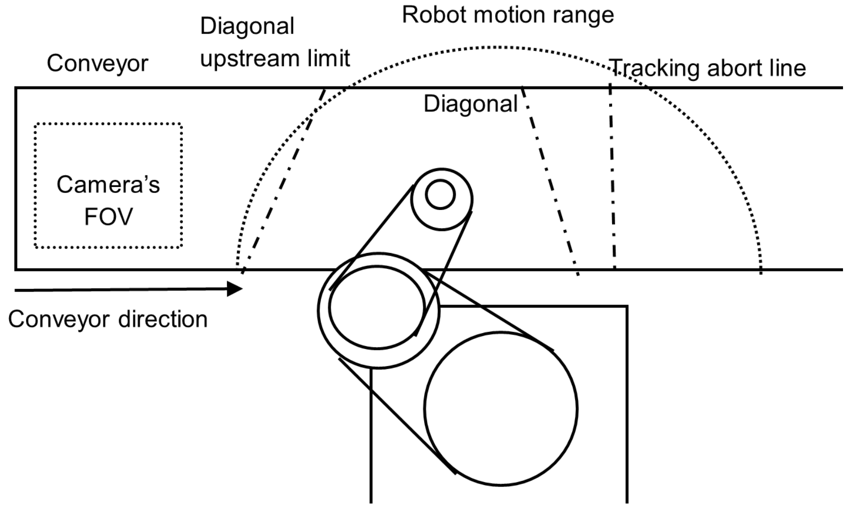
17.28 Tracking Abort Line

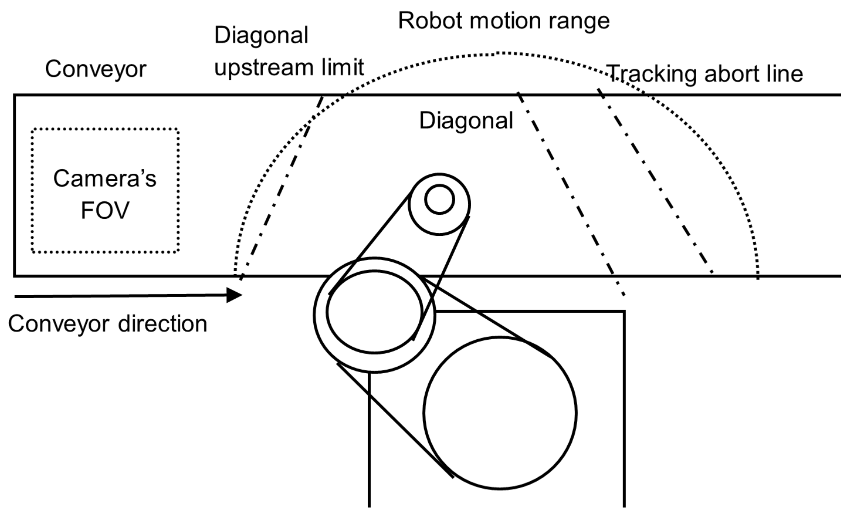
The tracking abort lines cancel or abort the robot's tracking motion in following cases:

- When it is predicted before the robot starts tracking that it catches up the workpiece beyond the abort tracking line.
- When the workpiece passes the tracking abort line while the robot performs the tracking motion. At this time, there is a function to detect whether the Z axis is descending for picking and to ascend in the height direction for safety. This height can be set arbitrarily.

KEY POINTS

- When you set the tracking abort line, the out of motion range error will not occur during conveyor tracking.
- If errors such as the out of motion range error occurs even though the tracking abort line is set, it is possible that the robot motion range has been reached during deceleration. Set the line more upstream than the current position.
- The circular conveyor does not support the setting of the tracking abort line.

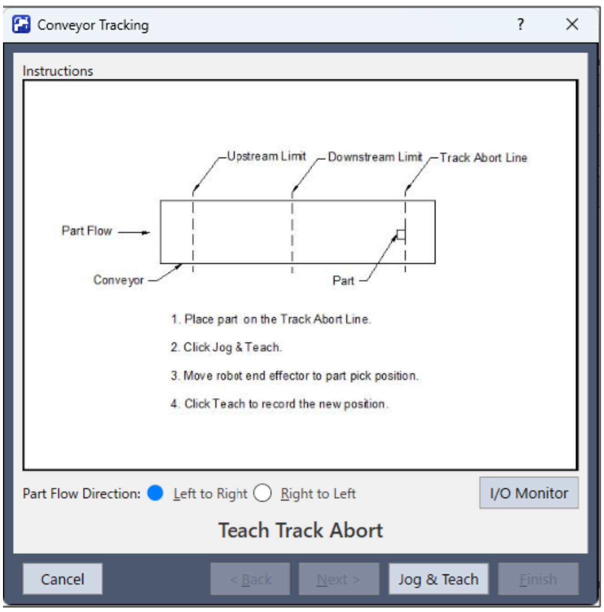




17.28.1 How to set a tracking abort line

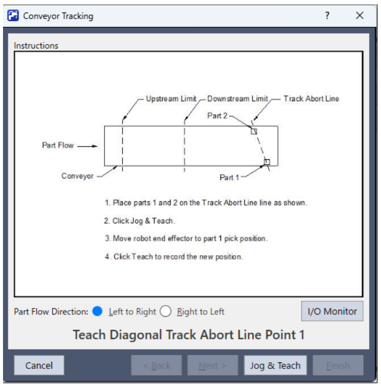
Set a tracking abort line in the following way.

1. Select [Tools]-[Conveyor Tracking].
2. Select the conveyor to configure.
3. Select [Track Abort Line].
4. Check the [Enable tracking abort line] checkbox.
   i. Click [Teach]. The GUI will be displayed.



   ii. If you want to enable the diagonal tracking abort line, check the [Diagonal Track Abort Line] checkbox.

Click [Teach]. The GUI will be displayed.

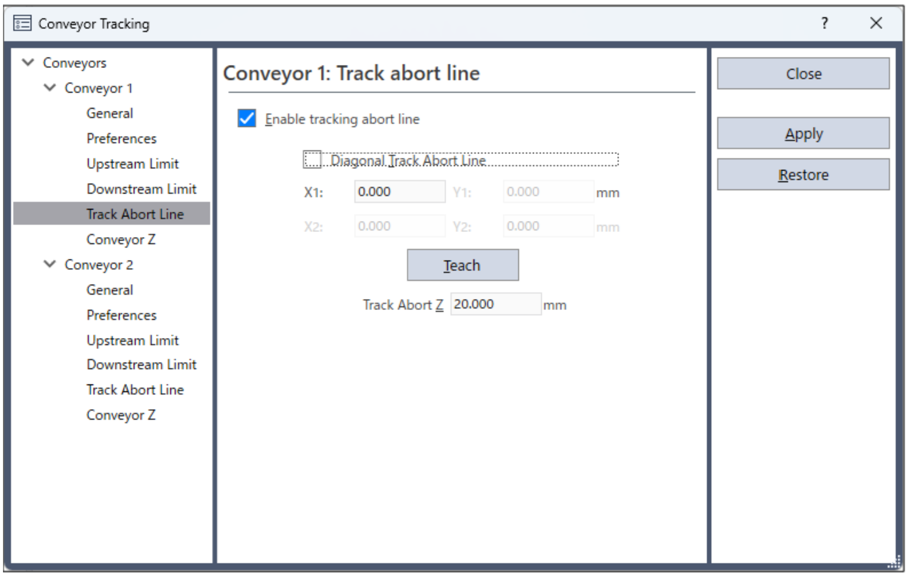


2. Follow the steps in the window and finish the setting.

17.28.2 How to set the Z rising height

The default rising height during tracking abort is 10 mm. You can change the height in the following way.

1. Select [Tools]-[Conveyor Tracking].
2. Select the conveyor to configure.
3. Select [Track Abort Line].
4. Set the rising height and click the [Apply] button.



KEY POINTS

If the out of motion range error occurs while the hand rises, decrease the rising height.

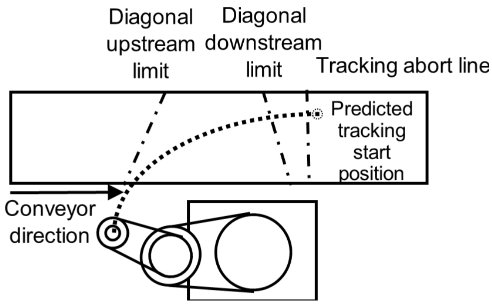
17.28.3 How to check the tracking abort state

You can check the tracking state in relation to the tracking abort line using the Cnv_Flag function.

A Cnv_Flag return value of “0” indicates normal operation, while “other than 0” indicates that tracking has been canceled or aborted. If the return value is other than "0", perform the following adjustments.

When 1:

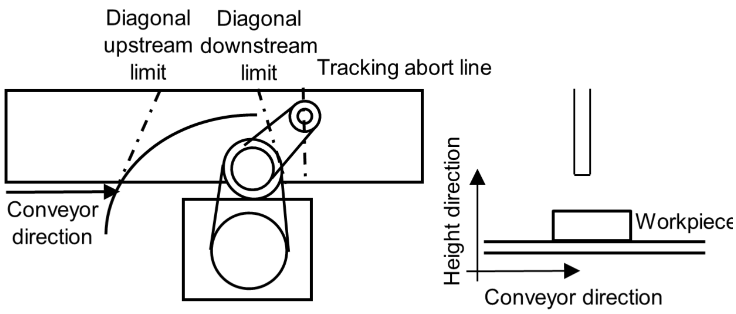
Tracking has been canceled as the workpiece is expected to exceed the tracking abort line before tracking commences.



Motion start may be delayed due to the downstream limit position setting. Set a downstream limit that is further toward the upstream side than the current position.

When 2:

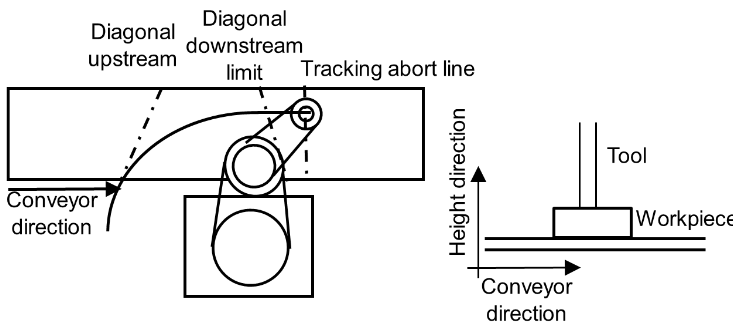
Tracking has been aborted as the tracking abort line was crossed during tracking (before pickup).



Motion start and completion could be delayed due to an improper downstream limit position and robot standby position. Set a downstream limit that is further toward the upstream side than the current position. Alternatively, bring the robot standby position closer to the downstream limit.

When 3:

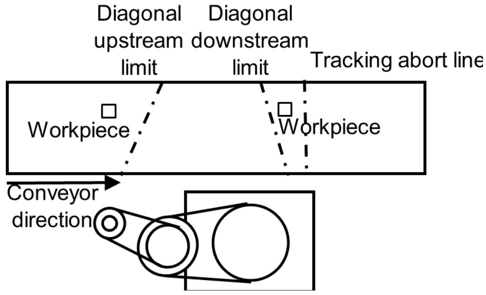
Tracking has been aborted as the tracking abort line was crossed during tracking (during pickup).



Motion start and completion could be delayed due to an improper downstream limit position and robot standby position, or an improper pickup time. Set a downstream limit that is further toward the upstream side than the current position. Alternatively, bring the robot standby position closer to the downstream limit. Try reducing the workpiece pickup time.

When 4:

Tracking has been canceled as the workpiece was outside the Pickup Area during motion command execution.



Refer to the sample program to put the robot on standby until the workpiece exceeds the upstream limit.

If this still occurs even when performing the above, the number of workpieces in the work flow may exceed the robot's processing capacity, causing the workpieces to exceed the downstream limit. Try performing the following adjustments.

- Reduce the number of workpieces
- Increase acceleration using Cnv_Accel
- Setup a downstream conveyor

> **KEY POINTS**
>
> If tracking motion is canceled or aborted, the program will continue to execute the subsequent instruction without stopping.

17.28.4 Program

If the tracking abort line is configured, the error 4406 does not occur. When you set the tracking abort line, use Cnv_Flag as in the following program. Do not use this program when the abort line is not used.

> **KEY POINTS**
>
> - Use the program 2 after the downstream limit is configured in the program 1 and Cnv_Flag do not return 2 and 3.
> - When using the program 1, the robot can operate without an error even though the downstream setting is not adequate since the robot aborts the tracking motion. However, abort of tracking increases the cycle time. It is recommended to adjust the downstream line if you are using the program 1.

Program 1

```
Function RB1
  'Moves to the waiting position P0
  Jump P0
  Do
    'Waits until the workpiece passes the upstream limit
    Wait Cnv_QueLen(1, CNV_QUELEN_PICKUPAREA) > 0
    Jump Cnv_QueGet(1)          'Executes the command to operate conveyor
tracking

    'When the state is normal, operates the picking
   If Cnv_Flag(1) = 0 Then
      On Vacuum1                'Vacuum ON
      Wait 0.1

      'If the workpiece crosses the tracking abort line during picking, causing
```

```
tracking to stop,
       release the pickup-failed workpiece.
       If Cnv_Flag(1) = 3 Then
          Jump P2              'Moves to the position to release the pickup-
failed workpiece
          Off Vacuum1          'Releases the workpiece
          Wait 0.1
          Jump P0              'Moves to the waiting position P0

       'The picked workpiece is moved to the placing position P1
       Else
          Cnv_QueRemove 1, 0    'Deletes the picked queue
          Jump P1              'Moves to the position to release the workpiece
          Off Vacuum1          'Releases the workpiece
          Wait 0.1
       EndIf


'If the operation is canceled because it is expected to cross the tracking abort
line, the queue is deleted.
    ElseIf Cnv_Flag(1) = 1 Then
       Cnv_QueRemove 1, 0       'Deletes the queue data


'If the operation is canceled because the workpiece at the time of executing '
the operation command is outside the Pickup Area, the queue is deleted.

    ElseIf Cnv_Flag(1) = 4 Then
       Cnv_QueRemove 1, 0       'Deletes the queue data

    'If the tracking motion is aborted since the workpiece passed the tracking
abort line,
    'the queue is deleted.
    ElseIf Cnv_Flag(1) = 2 Then
       Cnv_QueRemove 1, 0       'Deletes the queue data
       Jump P0                  'Moves to the waiting position P0
    EndIf
  Loop
Fend
```

Program 2

```
Function RB1

  'Moves to the waiting position P0
  Jump P0
  Do
    'Waits until the workpiece passes the upstream limit
    Wait Cnv_QueLen(1, CNV_QUELEN_PICKUPAREA) > 0
    Jump Cnv_QueGet(1)             'Starts tracking

'When the state is normal, operates the picking
    If Cnv_Flag(1) = 0 Then
       On Vacuum1               'Vacuum ON
       Wait 0.1
       Cnv_QueRemove 1, 0    'Deletes the picked queue
       Jump P1                  'Moves to the position to release the workpiece
       Off Vacuum1              'Releases the workpiece
       Wait 0.1

'If the operation is canceled because it is expected to cross the tracking abort
line, the queue is deleted.
ElseIf Cnv_Flag(1) = 1 Then
       Cnv_QueRemove 1, 0       'Deletes the queue data
```

```
    'If the operation is canceled because the workpiece at the time of executing
    'the operation command is outside the Pickup Area, the queue is deleted.
    ElseIf Cnv_Flag(1) = 4 Then
       Cnv_QueRemove 1, 0       'Deletes the queue data
    EndIf
  Loop
Fend
```

17.29 Tips for Accuracy Improvement of Conveyor Tracking

17.29.1 Overview

This section provides the tips for improving the robot’s performance to handle the workpieces in the Conveyor Tracking, which uses the vision system to detect the workpieces.

Process of Accuracy Improvement

Prepare the conveyor tracking in the following steps.

1. System construction
2. Vision calibration
3. Conveyor calibration
4. Check of the workpiece detection accuracy and the detection rate
5. Check of the workpiece handling accuracy

In order to improve the handling accuracy, proper preparations and adjustments in each step are necessary. The following sections describe tips for accuracy improvement for each step.

Subsequent descriptions use the SCARA robot. Please note that the tips are common in SCARA and 6-Axis robots.

17.29.2 Tips for System Construction

Tool Settings

In order to realize precise pickups, the end effector of the robot needs to pick up the workpieces correctly with a tool such as a vacuum pad attached.

To pick up the workpiece accurately by the tool, it is necessary to configure the tool in [Tools] page.

Eccentricity of the tool causes a constant pickup position gap. Make sure to configure the tool settings.

Also, adjustment by the tool settings is not effective for the improper tool such as a randomly-distorted vacuum pad due to the degenerated rubber. Make sure to use the proper tool.

KEY POINTS

- For more details on tool settings, see below.

  [Tools]-[Robot Manager]-[Tools] Page

- For details on tool number selection, see below.

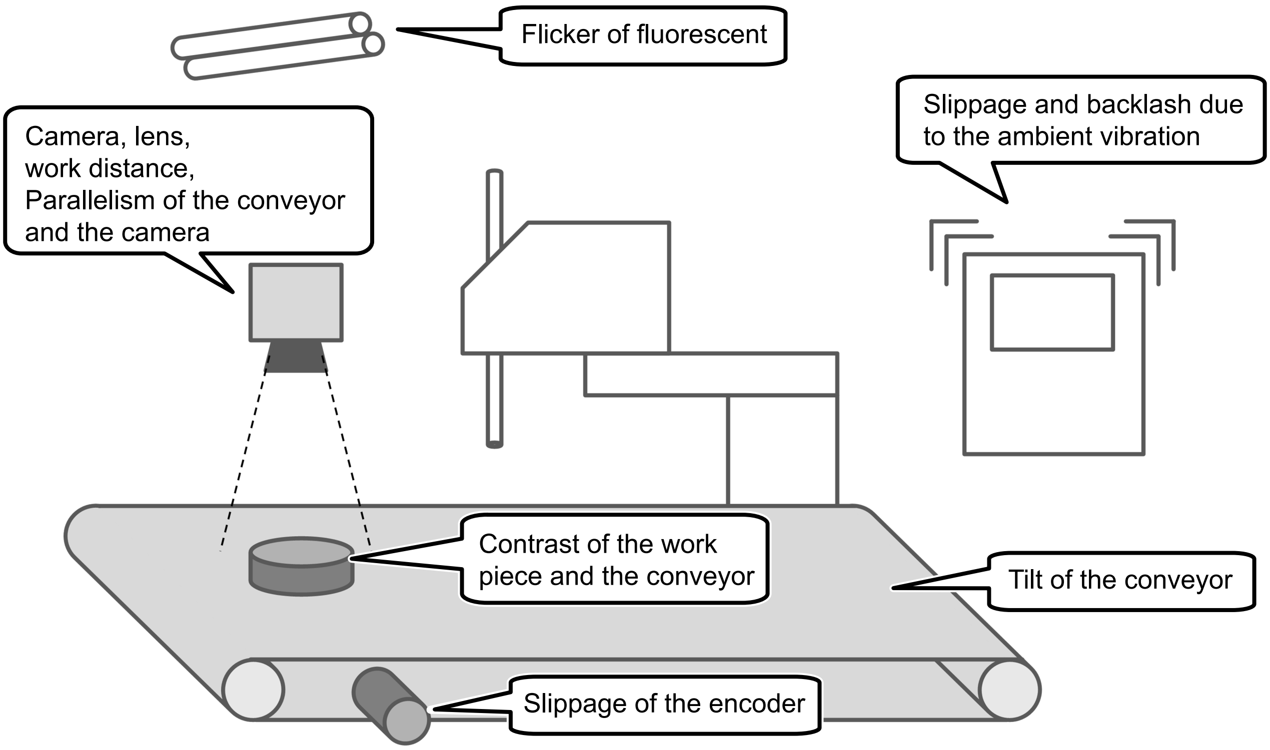
  [Tools]-[Robot Manager]-[Jog & Teach] Panel

- After adjusting the tool, make sure to check the operation and ensure that the calibration result is accurate.

Installation and Environment

In order to realize precise pickups, the camera and the conveyor should be installed in the proper environment and calibrated properly.

Install the system with attention to the following points.



Important Points for Installation of the Conveyor System

- Use a proper camera and lens. Configure a work distance (distance from the lens to the object) properly. Also, ensure parallelism of the camera and the conveyor in order not to cause distortion of the field of view (FOV).

- Ensure horizontality of the horizontal conveyor to match the positions of the robot coordinate system and the conveyor coordinate system.

  For the tilted conveyors, calibrate the tilt properly.

- If the encoder slips, pulse of the conveyor movement cannot be counted correctly.

- If the contrast of the workpiece and the conveyor is low, edges of the workpiece are difficult to be detected.

- Ambient vibration and source of the impact may cause slippage and backlash of the camera, the conveyor, and the workpieces. It may result in decrease in accuracy.

- General fluorescent lights have periodic flickers and may affect the work detection. Consider using fluorescent lights specialized for image processing or the LED lighting system.

KEY POINTS

Select and install the appropriate cameras and lenses to realize work detection which satisfies required pickup accuracy. Required work detection accuracy should be threefold of the required pickup accuracy. For details on modification of the FOV for accuracy improvement, see below.

Tips for Vision Calibration - "Field of View of Cameras"

17.29.3 Tips for Vision Calibration

Field of View of Cameras

A large field of view increases mm/pixel (length of 1 pixel) value and decreases detection accuracy of the workpieces.

If the following values from the calibration result do not satisfy the required accuracy,

- XmmPerPixel (X mm of one pixel)
- YmmPerPixel (Y mm of one pixel)

consider the following methods.

- Reinstall the camera and the workpiece to shorten the work distance (distance from the lens to the object)
- Use a high resolution camera
- Use a high resolution lens (e.g. our mega pixel lens) or a long-focal-length lens.

KEY POINTS

For details on the vision calibration, refer to the following manual.

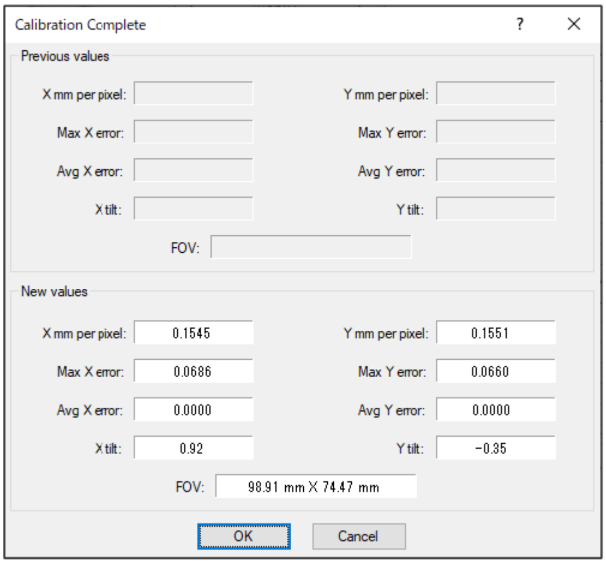
"Vision Calibration" in "Vision Guide 8.0 Software"

Deviation and Tilt of the Field of View

If Error (deviation) or Tilt values displayed in the calibration result were more than "1", it can be considered that calibration was not done properly.

For details, refer to the following manual.

"Vision Guide Software-[Calibration Complete Dialog Box"



Dialog displaying the calibration results

Reference Point Detection

For the vision calibration, use a proper combination of the reference point and the vision object; for instance, use a perfect circle as a reference point and detect it by the Blob object. In addition, it is necessary to perform a calibration with the “aperture” and “focus” of the camera adjusted to the workpiece.

- Adjust the camera aperture not too bright and not too dark in order to detect edges and marks of the workpiece.
- Adjust the focus to see the workpiece sharply. Blurs may affect the detection rate and accuracy.

KEY POINTS

- If the workpiece is thick and the top face does not come into the focus when the focus is placed on the conveyor, adjust the focus to the top face and set the reference point at the same height to calibrate.
- For details on reference points, refer to the following manual:

  "Vision Guide 8.0 Software- Reference Points and Camera Points"
- For details on vision objects, refer to the following manual:

  "Vision Guide 8.0 Software- Vision Objects"

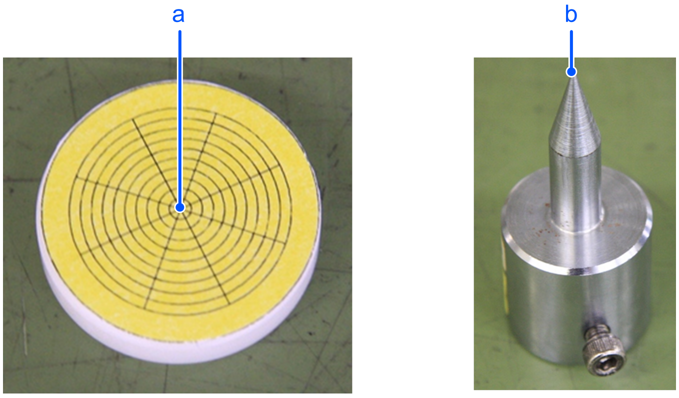
17.29.4 Tips for Conveyor Calibration

Workpiece and Tool

In order to realize precise pickups, it is necessary to perform a correct teaching in the conveyor calibration. To move the center of the end effector to the feature point (e.g. the center point) of the workpiece detected by the camera, it is recommended, for example, to use a workpiece and a tool as described below.

workpiece : Teaching point (a) is easy to be found

Tool : Tip point (b) is easy to be found (Make sure to configure the tool settings.)



Example of the workpiece and the tool used in the conveyor calibration

> **KEY POINTS**
>
> Perform a model teaching by overlaying the model origin of the Corr or Geom object on the teaching point, and adjust the positions of the camera coordinate system and the conveyor coordinate system correctly. If the teaching point is the center of balance of the workpiece (a perfect circle or a square), the center can be detected as the model origin by the Blob object.

Z Value Adjustment

The workpiece and the tool used in the conveyor calibration can have a different height from those used in actual workpiece handling. By adjusting the Z value after replacing the tool and the workpiece, errors related to the Z offset can be solved.

In the following cases, Z value adjustment is effective.

- Tool cannot reach and pick up the workpiece. (Z offset is too large)
- Robot collides with the workpiece and damages it. (Z offset is too small)

Like the above cases, redoing of the whole conveyor calibration process may not be required. If there is a problem in the pickup height, adjust the Z value as a first step.

> **KEY POINTS**
>
> For details on the Z value adjustment, see below.
>
> **Adjusting the Z value**

17.29.5 Troubleshooting for Workpiece Detection

Teaching of Pickup Position

In order to realize precise pickups, the pickup position of the workpiece should be detected properly as a model origin. To compensate the constant pickup position gap in workpiece handling due to the gap between the pickup position and the model origin, following methods are effective.

- Perform a model teaching by overlaying the model origin of the Corr or Geom object on the teaching point and set the CameraX and CameraY as the pickup position.

- To set the center of balance as a pickup position, detect it as a model origin by the Blob object and set the CameraX and CameraY as the pickup position.

* CameraX: X coordinate of the detected workpiece position in the camera coordinate system

* CameraY: Y coordinate of the detected workpiece position in the camera coordinate system

> **KEY POINTS**
>
> For details on vision objects, refer to the following manual:
>
> "Vision Guide 8.0 - Software- Vision Objects"

If the workpiece Cannot Be Detected in the Search Area

If the workpiece which is in the search area cannot be detected and an image processing error occurs, it can be improved by adjusting the vision properties. See the following points.

- Adjust the camera's exposure time

  Long exposure time affects the work detection since it may blur the image of the moving workpiece. Use the ExposureTime property to shorten the exposure time.

- Adjust the shape score value

  If the work detection rate is unstable, it may be improved by adjusting the Accept property of the vision object.

> **KEY POINTS**
>
> For details on the vision properties, refer to the following manual.
>
> "Vision Guide 8.0 Properties and Results Reference"

If Work Detection Does Not Satisfy the Required Accuracy

If the work detection does not satisfy the required accuracy, it can be improved by adjusting the vision properties. See the following points.

- Adjust the camera's exposure time

  Long exposure time affects the work detection since it may blur the image of the moving workpiece. Use the ExposureTime property to shorten the exposure time.

- Adjust the camera's field of view

  Wide field of view increases the length of 1 pixel and decreases the detection accuracy. Check the XmmPerPixel and YmmPerPixel values.

> **KEY POINTS**
>
> Example: If approximately 0.5 mm of the image blur at the 100 mm/sec of the conveyor speed is acceptable, set the exposure time to 5 msec.

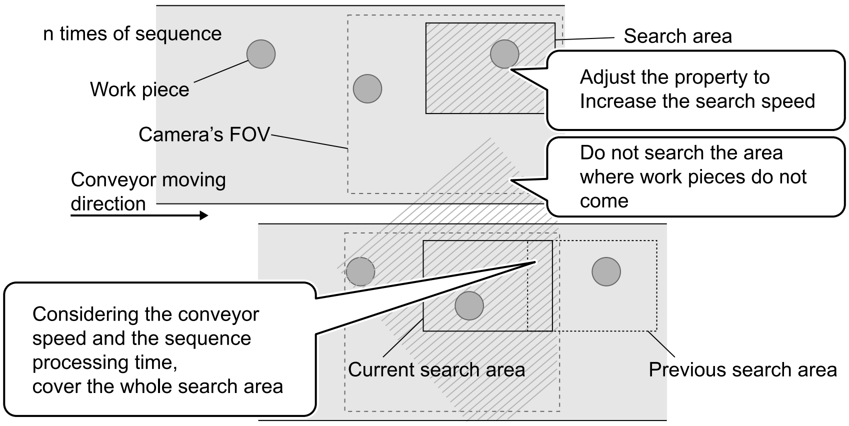
KEY POINTS

For details on the vision properties, refer to the following manual.

"Vision Guide 8.0 Properties and Results Reference"

Tips for the case if the image processing cannot be done in time

If the image processing cannot be done in time, it can be improved by adjusting the search area and the vision properties. See the following points.



Tips for the case if the image processing cannot be done in time

- Adjust the search window of the object

  A large search window increases time to execute the vision objects. Adjust the search window as small as possible by eliminating the area where workpieces do not come in.

- Adjust the number of the objects to be detected

  When you want to detect only one workpiece at a time, setting the NumberToFind property to “1” may reduce the execution time.

- Adjust the range of the expected scale

  If there is no great variability in size of the workpieces, set the ScaleEnable property to “False”. If there is a small variability, narrow the ranges of the ScaleFactorMax and the ScaleFactorMin properties as much as possible.

- Adjust the range of the angled detection

  If there is no great variability in angle among the workpieces, set the AngleEnable property to “False”. If there is a small variability, narrow the range of the AngleRange property as much as possible.

- Adjust the timeout period

  A process is aborted when the image processing time is considered to exceed the timeout period. If image processing times vary, both the detection rate and the execution time may be improved by shortening the Timeout property.

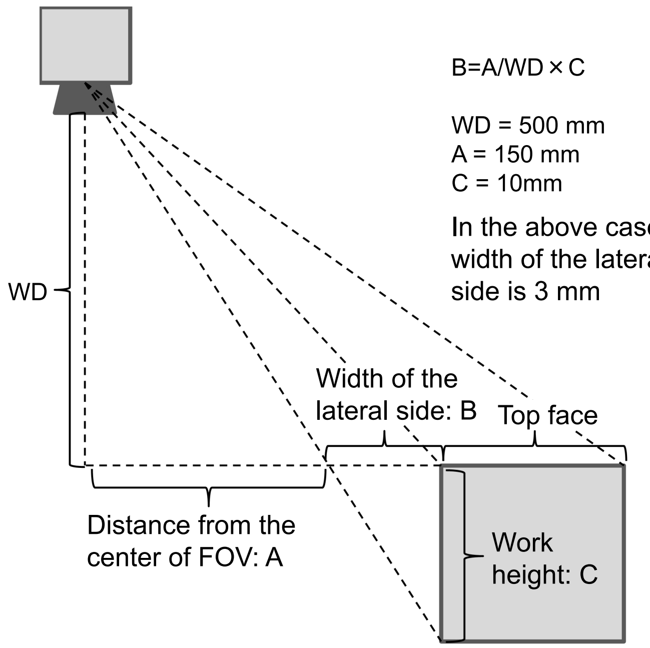
KEY POINTS

For details on the vision properties, refer to the following manual.

"Vision Guide 8.0 Properties and Results Reference"

When Using A Thick Workpiece

If the workpiece is thick, the camera's FOV includes the lateral side of the workpiece as shown in the figure below. If the top face and the lateral side of the workpiece have similar color, those two faces may be detected as a single top face of the workpiece. Pay attention to this influence especially when using the thick workpieces.



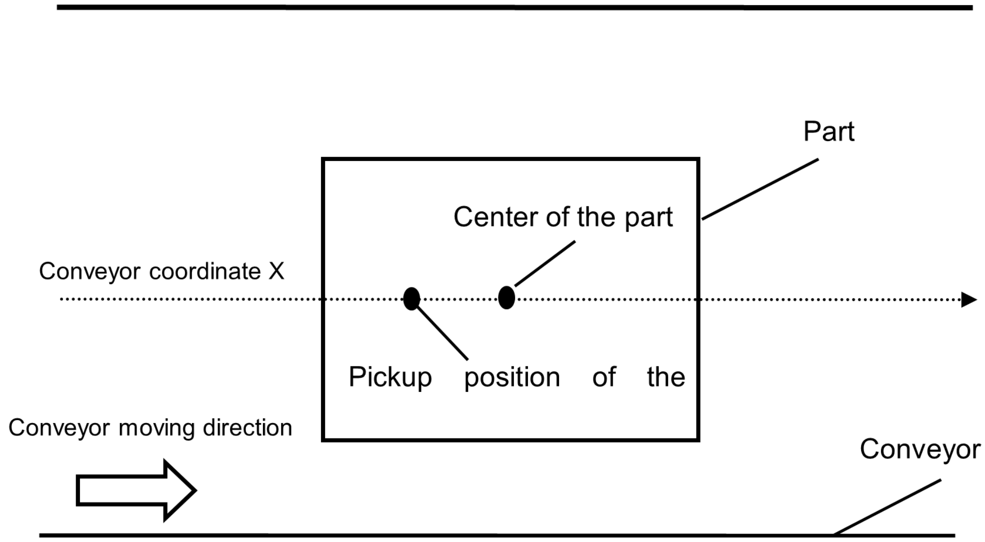
Influence of detecting the lateral side of the workpiece

> **KEY POINTS**
>
> This influence can be decreased by increasing the work distance or replacing the lens with a one with a long focal distance and a narrow angle of view.

17.29.6 Offset

Pickup of moving parts

With conveyor tracking, the pickup position of the robot may deviate from the center of the part as shown below. This gap occurs as a result of accuracy errors of the vision system calibration, tool calibration, and tracking.



Followings are countermeasures for the problem.

1. Feed the part with an angle near 0°degree. Then, pick up the part.
2. Measure the gap between the center of the part and the robot's pickup position.
3. Repeat the steps 1 and 2 for five times and calculate the average.
4. Set the average calculated in step 3 to the program as follows:

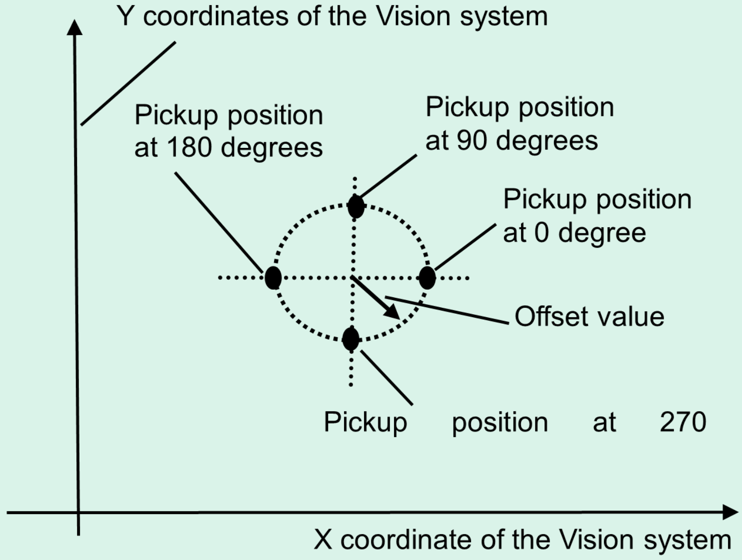
```
> Jump Cnv_QueGet(1) +X(**)
```

5. Feed the part with an angle near 90°degree. Then, pick up the part.
6. If the gap is large, fine-tune the value set in the step (4).
7. Feed the part with an angle near 0°degree. Then, pick up the part.
8. If the gap is large, fine-tune the value set in the step (6).
9. Repeating steps 6 to 8 may improve picking accuracy.

KEY POINTS

If the gap between the center of the part and the pickup position can be measured by using the vision system, compensate the "Offset" as follows:

1. Feed the part with an angle near 0 degree. Then, pick up the part.
2. Take the image of the picked part by the camera and record X and Y coordinates.
3. Repeat the steps 1 and 2 for five times and calculate the average.
4. Feed the part with an angle near 90°degree. Then, pick up the part.
5. Take the image of the picked part by the camera and record X and Y coordinates.
6. Repeat the steps 5 and 6 for five times and calculate the average.
7. Feed the part with an angle near 180°degree. Then, pick up the part.
8. Take the image of the picked part by the camera and record X and Y coordinates.
9. Repeat the steps 7 and 8 for five times and calculate the average.
10. Feed the part with an angle near 270 degree. Then, pick up the part.
11. Take the image of the picked part by the camera and record X and Y coordinates.
12. Repeat the steps 11 and 12 for five times and calculate the average.
13. Plot the values in the steps 3, 6, 9, and 12 as shown in the figure below, and calculate the offset value.



14. Set the offset value to the program as follows:

```
> Jump Cnv_QueGet(1) +X(offset)
```

KEY POINTS

Depending on the offset value, the upstream limit may be exceeded and error 4406 may occur.

```
Wait Cnv_QueLen(1, CNV_QUELEN_PICKUPAREA) > 0
Jump Cnv_QueGet(1) -X(offset)
```

The error can be avoided by doing either of the following:

- Set the wait time before Jump command.
- Set the "offset" when registering the queue and not to register when executing the Jump command.

17.30 Dispense Application on Conveyors

17.30.1 Overview

Conveyor Tracking corresponds CP motion commands and Path motion and can be used in dispense application.

The following motion commands correspond:

- Move : Linear motion
- Arc : XY plane circular interpolation
- Arc3 : 3D circular interpolation
- CVMove : Free curve interpolation

CP motion and Path motion can be executed while following a workpiece. Dispensing speed of CP motion and Path motion are set in AccelS and Speeds.

```
'Setting of dispensing speed
SpeedS 50          'Dispensing speed 50mm/s
AccelS 1000 1000
```

KEY POINTS

- When executing a dispensing at a constant speed, add a pre-motion to reach the dispensing speed. When you use an optional analog I/O board, it can adjust the amount of dispensing according to the speed.
- When using the Arc and Arc3 commands for conveyor tracking, use the Arc syntax (1) and Arc3 syntax (1), which specify the midPoint, as described in the SPEL+ Language Reference.
- Arc syntax (2) to (4) and Arc3 syntax (2), which do not specify the midPoint, are not available for conveyor tracking.

17.30.2 Setting a Target Point

When executing CP motion and Path motion in Conveyor Tracking, specify a target point by using conveyor queue data detected in Vision system or points.

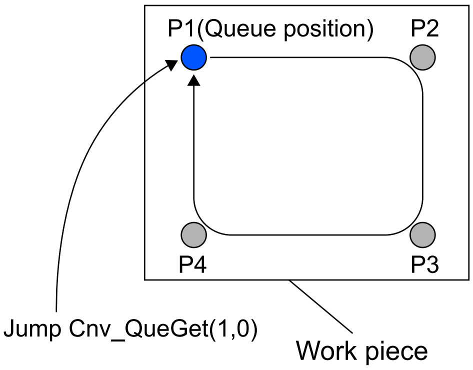
To use points:

Teach points in the dispensing path using a workpiece placed on the conveyor. Do not move the conveyor during teaching.

Specify the coordinate differences between points for CP motion and Path motion. If the angle of the workpiece is different from that of the teaching, calculate the angle difference and specify the coordinates.

The following is a sample program that uses CP motion and Path motion on a square shape.

Teach in the order of P1, P2, P3, P4, P1.



Converts point coordinates to conveyor coordinates (using conveyor 1)

```
P101 = P1 @CNV1;  P102 = P2 @CNV1
P103 = P3 @CNV1;  P104 = P4 @CNV1

Jump Cnv_QueGet(1,0)      'Follows the queue position
Move Cnv_QueGet(1,0)+X(CX(P102)-CX(P101))+Y(CY(P102)-CY(P101)) CP
Move Cnv_QueGet(1,0)+X(CX(P103)-CX(P101))+Y(CY(P103)-CY(P101)) CP
Move Cnv_QueGet(1,0)+X(CX(P104)-CX(P101))+Y(CY(P104)-CY(P101)) CP
Move Cnv_QueGet(1,0)
```

To use CVMove command:

```
Converts point coordinates to conveyor coordinates (using conveyor 1)
P101 = P1 @CNV1;  P102 = P2 @CNV1
P103 = P3 @CNV1;  P104 = P4 @CNV1
Curve "MyFile", O, 2, 4, P(101:104)  'Creates a Curve file

Jump Cnv_QueGet(1,0)      'Follows the queue position
CVMove "MyFile"
```

To use conveyor queue data detected in Vision system:

Register the position of points P1 to P4 in the diagram of "To use points:" to Vision system as conveyor queue data. Converting coordinates as in "To use points:" or calculating the coordinate differences between points and the angle of a workpiece are not required.

To use CVMove command, create Curve in the same way as "To use points:"

```
Jump Cnv_QueGet(1,0)      'Follows the queue position
Move Cnv_QueGet(1,1) CP
Move Cnv_QueGet(1,2) CP
Move Cnv_QueGet(1,3) CP
Move Cnv_QueGet(1,0)
```

17.30.3 Adjusting an Amount of Dispensing

Use the optional analog I/O board to output analog voltages according to the speed of the robot during Conveyor Tracking.

When you use a dispenser with a function to output the dispensing amount, the dispensing amount can be adjusted according to the speed of the robot.

For connection and use of the analog I/O board, refer to the following manuals:

- "Robot Controller RC800-A Manual - Analog I/O Board"
- "Robot Controller RC700 Series Manual - Analog I/O Board"
- "Robot Controller RC700-D Manual - Analog I/O Board"
- "Robot Controller RC700-E Manual - Analog I/O Board"
- "Robot Controller RC90 Series Manual - Analog I/O Board"

The SPEL+ command of the analog I/O board used for Conveyor Tracking is AIO_Set.

For details, refer to the following manual.

"SPEL+ Language Reference"

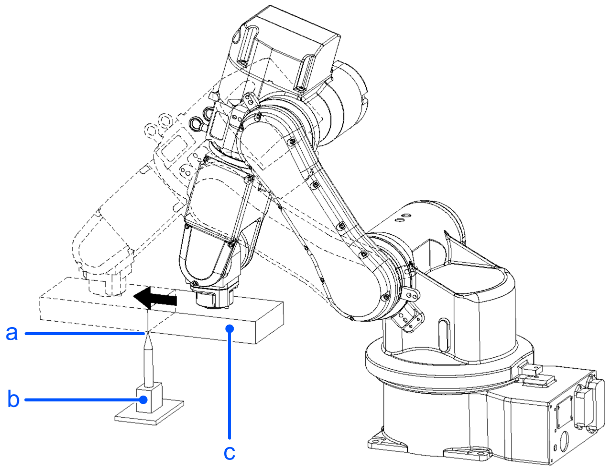
KEY POINTS

AIO_Set cannot be used for circular conveyors.

18. ECP Motion

18.1 Overview

An ECP (external control point) motion is when the robot arm holding a part follows a specified trajectory (part's edges, etc.) using an outside fixed tool.



Symbol	Description
a	External control point (ECP)
b	Outside fixed tool
c	Workpiece

The ECP option supports the following:

- ECP definition by ECPSet statement and selection by ECP statement
- ECP motion commands (additional functions of Move, Arc3, Curve, and CVMove commands)
- Teaching with ECP jogging

This option is available for SCARA (including RS series), Cartesian and 6-axis robots (including N series). Also, it can be used with multi-robot systems.

Up to 15 ECP coordinate systems can be defined.

18.1.1 How to move the arm with ECP motion

In the following paragraphs, the process for moving the 6-axis robot arm with ECP motion is explained as an example.

1. External control point settings

The ECP (external control point) is a coordinate system data used for defining the robot position and orientation at a processing point on the tip of the outside fixed tool.

The ECP should be defined based on the robot coordinate system or desired local coordinate system.

For example, when a drawing shows that the ECP is located at X=300, Y=300, Z=300 based on the robot coordinate system, specify it as shown below.

```
ECPSet 1,XY(300,300,300,0,0,0)    ' Defines ECP No.1
```

When you have no ECP location data, you can specify it by teaching.

As an example, attach the tool of which you know the data precisely and bring the tip of the tool close to the ECP and then teach its position anywhere as P0. Then, specify the ECP using P0 coordinate data as shown below.

```
ECPSet 1,P0 :U(0) :V(0) :W(0)     ' Defines ECP No.1
```

The orientation data (U, V, W) were set to 0 in the above examples. In these cases, the orientation in the ECP coordinate system is equal to that in the reference robot coordinate system.

You can specify U, V, and W coordinates in the ECP coordinate system. However, this data is valid only during the tangential correction mode ON in the Curve statement and ECP jog motion.

2. Teaching

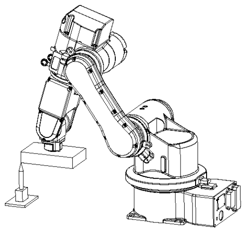
Teach the point data while moving the robot arm holding the actual part. In this section, the part is assumed to be a rectangular solid and the arm is moved straight so that it touches one side of the part of the ECP specified in the previous section 1. Setting the ECP.

For details on teaching, see below.

[Tools]-[Robot Manager]-[Jog & Teach] Panel

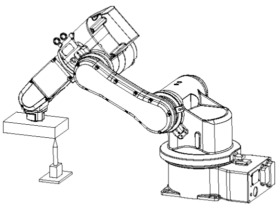
2-1 Teaching the motion start point

Move the arm to the motion start point. Teach it as P1.



2-2 Teaching the motion end point

Move the arm to the motion end point. Teach it as P2.



ECP Jog Mode:

The ECP jog mode is an additional jog mode used for teaching besides the Joint, World, and Tool jog modes.

> **KEY POINTS**
>
> The ECP jog mode is based on the selected ECP coordinate system.

3. Executing Motion

To move the arm with ECP motion, add the "ECP" parameter to a motion command.

```
ECP 1        'Select ECP
Go P1        'Moves the arm to the motion start point
Move P2 ECP 'ECP Execute
```

Use the Arc3 command to move the arm in an arc trajectory with the fixed tool. Use the Curve and CVMove commands to move the arm in cubic spline curves.

19. Distance Tracking Function

19.1 Overview

Distance tracking function controls the robot so that a constant distance can be kept between the robot and the workpiece. Distance sensor which is connected to the analog I/O board (optional) is used. To use the function, analog I/O board (optional) is required.

Select one axis from the following as the direction to control.

- Tool coordinate system: X axis, Yaxis, Zaxis
- ECP coordinate system: X axis, Yaxis, Zaxis

ECP coordinate system can be selected only when the ECP (external control point motion) option is enabled.

Specify the axis to be controlled by AIO_TrackingSet.

The distance tracking function is available for SCARA robots (including RS series manipulators) and 6-Axis robots (including N series manipulators). Also, it can be used with multi-robot systems.

Carefully note the following when using the distance tracking function in multi-robot systems.

A: Two robots : Up to two distance tracking sensors are available. Two robots can use each distance tracking sensor to execute the distance tracking function simultaneously.

B: Three robots : One distance tracking sensor is available. One robot can use the distance tracking function. However, three robots can use the one distance tracking sensor and execute the distance tracking function in order by switching it.

C: More than four robots : The distance tracking function is not available.

When using the distance tracking function in multi-robot systems, you cannot connect the force sensor.

> ⚠ CAUTION
>
> We recommend using a laser displacement meter for a distance sensor. For the specifications of the distance sensor, please check its manual thoroughly. Improper use of the sensor may result in abnormal motions of the robot.

For connection and use of the analog I/O board, refer to the following manuals:

- "Robot Controller RC800-A Series Manual" - Analog I/O Board
- "Robot Controller RC700 Series Manual - Analog I/O Board"
- "Robot Controller RC700-D Manual - Analog I/O Board"
- "Robot Controller RC700-E Manual - Analog I/O Board"
- "Robot Controller RC90 Series Manual - Analog I/O Board"

19.1.1 Distance Tracking Accuracy

For the accuracy achieved by this function, see the after mentioned experiment results.

However, the accuracy of the distance tracking function may differ depending on the robot model, speed, and workpiece shape.

Experiment conditions

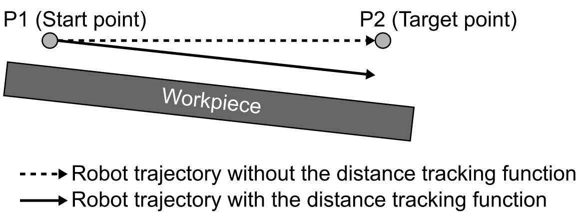
- Robot: Vertical 6-axis robot C4L

- Laser displacement meter: 2 types (see table below for specifications)

	Medium speed/medium accuracy laser displacement meter	High speed/high accuracy laser displacement meter
Measured distance (mm)	20 to 30	7.2 to 8.8
Spot diameter (um)	Approx. 25×1200	Approx. ⌀20
Repeatability (um)	1	0.02
Sampling cycle	0.33, 1, 2, 5 ms (4 levels available)	20, 50, 100, 200, 500, 1000 us (6 levels available)
Light source (Laser class)	Class 2	Class 1

Experiment environment

Start point and target point have been taught in advance.



- Without the distance tracking function: Robot moves in a straight line from the start point to the target point.
- With the distance tracking function: The robot moves on a trajectory to keep a constant distance to the workpiece like the blue arrow (robot trajectory) shown in the above figure.

Experiment results

Values of the distance tracking accuracy are the width of variations in the distance measured values between the start point and end point of the distance tracking function. Two types of the laser displacement meter are used. (See the tables below for the experiment results.)

Distance tracking accuracy of medium speed/medium accuracy laser displacement meter

Robot SpeedS(mm/s)	Robot AccelS(mm/s*s)	Workpiece tilt (mm)		
		5 deg.	10 deg.	15 deg.
10	100	±0.03	±0.04	±0.06
30	300	±0.06	±0.09	±0.14
50	500	±0.09	±0.15	±0.32
100	1000	±0.15	±0.30	±0.48

Distance tracking accuracy of high speed/high accuracy laser displacement meter

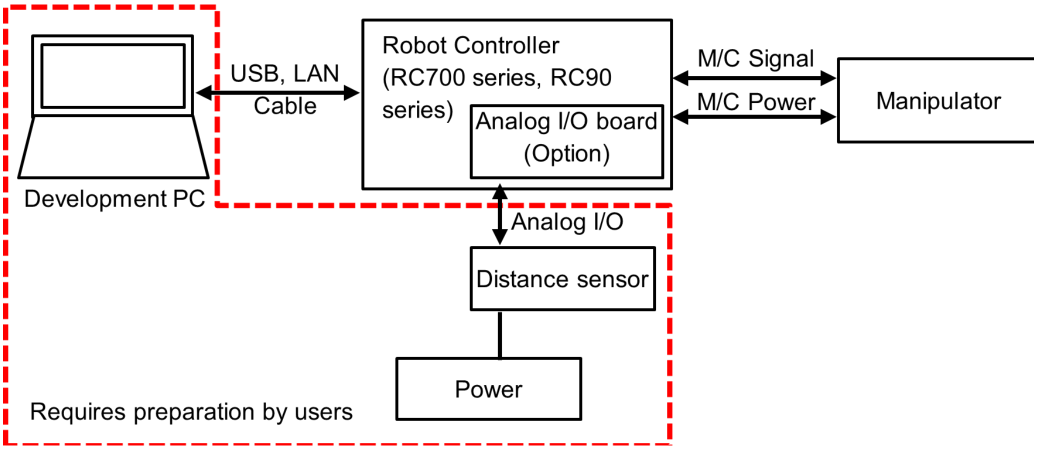
Robot SpeedS(mm/s)	Robot AccelS(mm/s*s)	Workpiece tilt (mm)		
		5 deg.	10 deg.	15 deg.
10	100	±0.02	±0.04	±0.05
30	300	±0.04	±0.06	±0.13

Robot SpeedS(mm/s)	Robot AccelS(mm/s*s)	Workpiece tilt (mm)		
		5 deg.	10 deg.	15 deg.
50	500	±0.06	±0.11	±0.20
100	1000	±0.13	±0.20	±0.35

19.2 Connection Example

This section describes the connection example of the distance tracking function.

19.2.1 Basic Connection Example



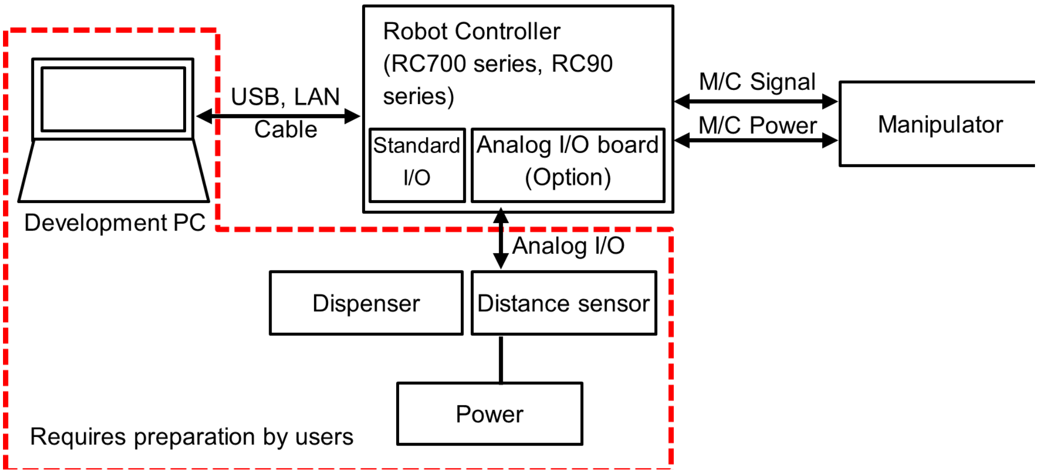
KEY POINTS

The customer must prepare the following items:

- Power (select depending on the distance sensor to be used.)
- Distance sensor (e.g. laser displacement meter)
- Development PC

19.2.2 Connection Example for Dispense Application

Distance tracking function is available for dispense applications. To realize the dispense application with high accuracy, it is important to keep a constant needle gap (distance between tip of the needle and the workpiece). Constant needle gap can be realized with the distance tracking function. The figure below shows an example connection for dispense application.



KEY POINTS

The customer must prepare the following items:

- Power (select depending on the distance sensor and dispenser to be used.)
- Distance sensor (e.g. laser displacement meter)
- Dispenser
- Development PC

19.3 Command

List of SPEL+ commands for the distance tracking function.

- AIO_TrackingSet: Sets the distance tracking function
- AIO_TrackingStart: Starts the distance tracking function
- AIO_TrackingEnd: Ends the distance tracking function
- AIO_TrackingON Function: Returns status of the distance tracking function

For details on commands, refer to the following manual:

"Epson RC+ 8.0 SPEL+ Language Reference"

19.4 Steps to adjust parameters

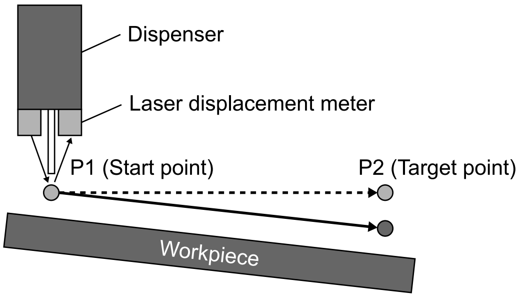
The accuracy of the distance tracking function may differ depending on the robot model, speed, and workpiece shape.

Therefore, you need to set parameters depending on the working environment when using the distance tracking function.

To improve the accuracy of the distance tracking function, adjust parameters and set a proper value.

Parameters to be set are ProportionalGain, IntegralGain, and DifferentialGain. These are parameters for AIO_TrackingStart.

In the steps to adjust parameters, it is assumed to use a flat metallic plate as a workpiece in the dispense application as shown below.

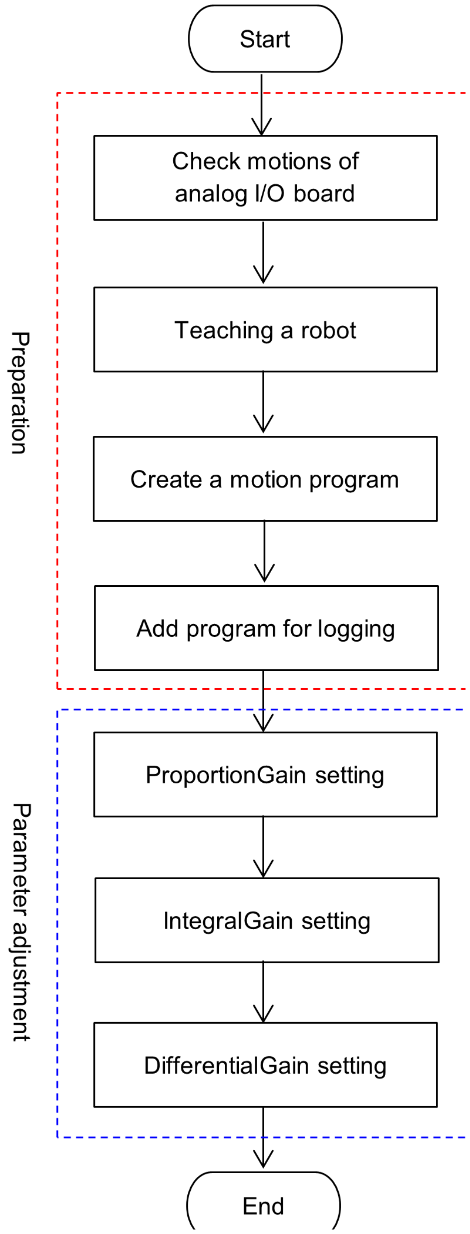


Steps to adjust parameters are below:

Statuses at the beginning are as follows:

- Dispenser: Connection and settings have completed.
- Laser displacement meter: Connected to analog I/O.

Describe in order of the preparation for parameter adjustment and parameter adjustment.



> ⚠ CAUTION
>
> - The parameters used in this step are reference values. Please note that the operation may not be successful or the motion may be vibratory depending on the set parameters and some operating conditions.
> - If the robot performs an action different from the setting, immediately press the emergency stop button.

19.4.1 Check Motions of Analog I/O Board

The following steps describe how to check the motion of the analog I/O board.

1. Make sure that the analog I/O board and the laser displacement meter (distance sensor) are connected properly.

   For connection and use of the analog I/O board, refer to the following manuals:
   - "Robot Controller RC800-A Series Manual" - Analog I/O Board
   - "Robot Controller RC700 Series Manual - Analog I/O Board"

- "Robot Controller RC700-D Manual - Analog I/O Board"
- "Robot Controller RC700-E Manual - Analog I/O Board"
- "Robot Controller RC90 Series Manual - Analog I/O Board"

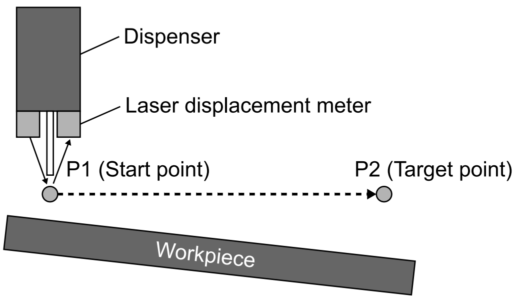
2. Execute the following command on the command window.

```
>Print AIO_In (channel number of analog I/O board)
```

3. Output voltage from the laser displacement meter is displayed. Check the displayed value and the value measured by the laser displacement meter. If those values are the same, the analog I/O board works properly.

19.4.2 Teaching a Robot

Teach the start and target point of the distance tracking function.



1. Move the robot to a position where the laser displacement meter will be within the measurable range.
2. Set a robot position and orientation within the range and teach as the start point (P1).

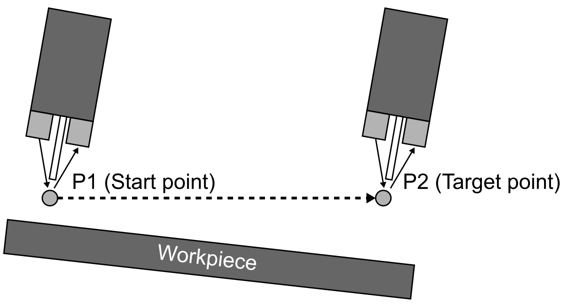
   When using the distance tracking function for a dispense application, be sure to check that the needle gap at the start point is same as the recommended gap value of the dispenser.

3. Move the robot to the target point.
4. Teach a moved point as the target point (P2).

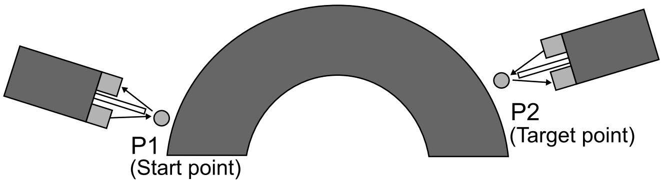
Sometimes the laser displacement meter cannot measure the tilt of workpiece due to workpiece types (especially mirrored object), its tilt angle, and types of the laser displacement meter. In that case, place the bottom of a chassis of the laser displacement meter in parallel with the workpiece surface.

For layout of the distance sensor (laser displacement meter) and target workpiece, follow the specifications of each distance sensor.

Ex: When placing the workpiece and chassis of the laser displacement meter in parallel



For the arc-shaped workpiece as shown above, teach the trajectory of the arc shape by using Move or Arc command before executing the distance tracking function.



19.4.3 Create a Motion Program

Create a motion program for the distance tracking function.

Program example:

Move the robot from P1 to P2 by using the distance tracking function.

Set the needle tip position of the dispenser for Tool coordinate system. However, the dispenser does not move until the parameter adjustment ends. Parameters of AIO_TrackingSet are examples. Be sure to set parameters depending on the working environment.

```
Function AIOTrackingSample
  ' -------------Robot configuration ---------------
  Motor On
  Power High
  SpeedS 30
  AccelS 300, 300
  Tool 1
  ' -------------Motion part ---------------
  Move P1                                 'Move to start point
  AIO_TrackingSet 1, -1, 0, -3, 3, 0, 2  ' Sets the distance tracking function
  Wait 2
  AIO_TrackingStart 1, 10, 0, 0          ' Starts the distance tracking function
  Move P2                                 'Starts the distance tracking function
  AIO_TrackingEnd                         ' Ends the distance tracking function
  Wait 2
  Motor Off
Fend
```

Set the default values for parameters of AIO_TrackingStart as follows:

- ProportionalGain: 10
- IntegralGain: 0
- DifferentialGain: 0

19.4.4 Add a Program for Distance Sensor Logging

To adjust the parameters (ProportionalGain, IntegralGain, DifferentialGain), you need to check the measured data of the laser displacement meter during execution of the distance tracking function.

Measured data of the laser displacement meter can be acquired by the following sample program.

Add ★ to the program created below.

Create a Motion Program

```
____________________________________________________________
Integer fileNum                          ' ★ Declare a file number
Function AIOTrackingSample
'=============================================================
```

```
' Program to record the measured value of the distance sensor during execution of
the distance tracking function.
'============================================================
'------------- Robot configuration ---------------
  Motor On
  Power High
  SpeedS 30
  AccelS 300, 300
  Tool 1
  '--------- Create a CSV file for recording ------------------
  fileNum = FreeFile                        ' ★ Acquire a file number
  WOpen "AIO_Monitor.csv" As fileNum       ' ★ Save to Project folder
  ' ------------- Motion part ---------------
  Move P1                                   ' Move to start point
  Xqt AIO_Monitor                           ' ★ Start to record the measured value
by distance sensor
  AIO_TrackingSet 1, -1, 0, -3, 3, 0, 2  ' Sets the distance tracking function
  Wait 2AIO_TrackingStart 1, 10, 0, 0     ' Starts the distance tracking function
  Move P2                                   ' Starts the distance tracking function
  AIO_TrackingEnd                          ' Ends the distance tracking function
  Wait 2
  Quit AIO_Monitor                          ' ★ Quit recording the values by
distance sensor
  Close #fileNum                            ' ★ Close CSV
  Motor Off
Fend

____________________________________________________________
Function AIO_Monitor                        ' ★
'========================================================
' Called by AIOTrackingSample.
' Keep recording values input to Ch1 of analog I/O board to CSV.
'========================================================
  Do                                        ' ★
    Print #fileNum, AIO_In(1)               ' ★
    Wait 0.002                              ' ★
  Loop                                      ' ★
Fend                                        ' ★
____________________________________________________________
```

19.4.5 ProportionalGain Setting

This section describes how to execute the program created below and adjust ProportionalGain.

Add a Program for Distance Sensor Logging

1. Test run in low speed

   Execute the program created below in low speed (10 mm/s or less).

   Add a Program for Distance Sensor Logging

   Set SpeedS to 10 or less and AccelS to 100 or less.

Make sure that the robot moves to the target point and the program works properly. Since the value of ProportionalGain is small, the robot moves straight to the target point. Be sure to move the robot in the environment with no obstacles between the motion start point and target point.

When 4603: out of range error occurs:

Since the value of ProportionalGain is small, "4603: Out of range error" may occur. If the error occurs, increase the value of ProportionalGain by 10.

2. Test run in actual speed

Since it was confirmed that the program works properly at (1), execute the program in the desired working environment. Set robot speed and acceleration to desired value.

When 4603: out of range error occurs:

Parameter adjustment is required. Refer to the following tips to adjust parameters, and then check the motion again.

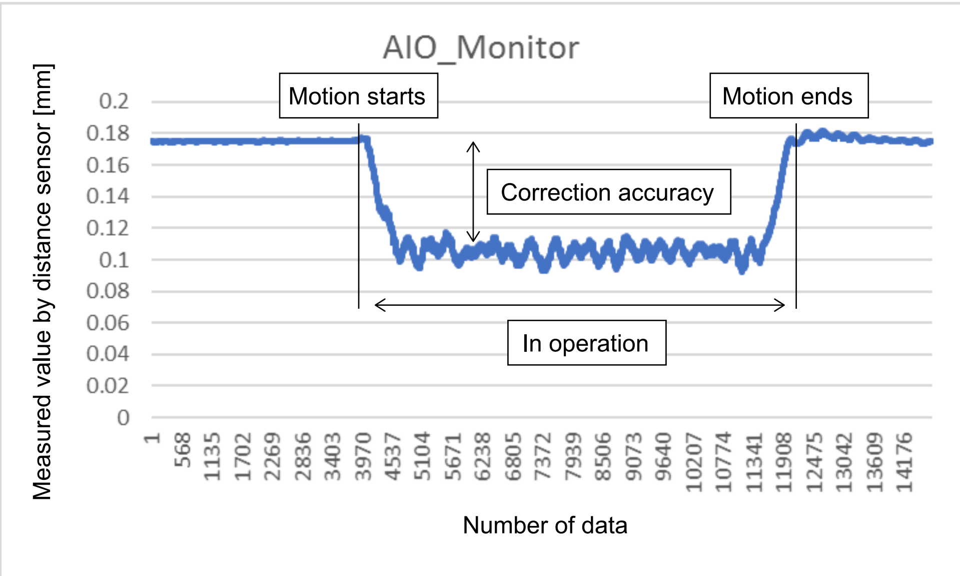
- Value of ProportionalGain is small. Increase the current value by 10.
- Robot speed is too fast. Move the robot in 100mm/s or less.

3. Check the motion results

“AIO_Monitor.csv” is created in the project folder of Epson RC+ 8.0. Open the file in spreadsheet software and create a line graph or a scatter plot by all data of column A.

Graph shown below will be created. Check the correction accuracy on the graph. In case of the following graph, the correction accuracy is approx. 70um.

If the correction accuracy is within the target accuracy, parameter adjustment ends.



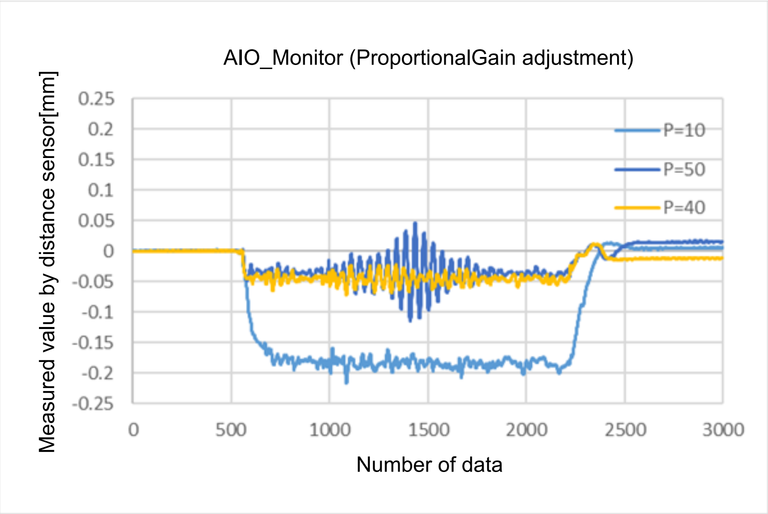
4. Adjustment of ProportionalGain

If the correction accuracy does not achieve the target value, adjustment of ProportionalGain is required.

ProportionalGain is a parameter to set strength of correction. Adjust value of ProportionalGain and repeat the program execution to calculate a proper value.

Be sure to increase the value of ProportionalGain gradually. Changing the value to a larger one at one time is extremely hazardous and the robot may move unintentionally.

When adjusting ProportionalGain, keep IntegralGain and DifferentialGain to "0".



When adjusting ProportionalGain, the correction accuracy is improved.

However, if increasing the value too much, the robot motion will be vibratory. (Refer to the upper graph: P=50)

ProportionalGain value without robot vibrations and the best correction accuracy is an optimum value. (Refer to the upper graph: P=40)

If the target correction accuracy is not achieved even if ProportionalGain is adjusted, you need to adjust IntegralGain.

19.4.6 IntegralGain Setting

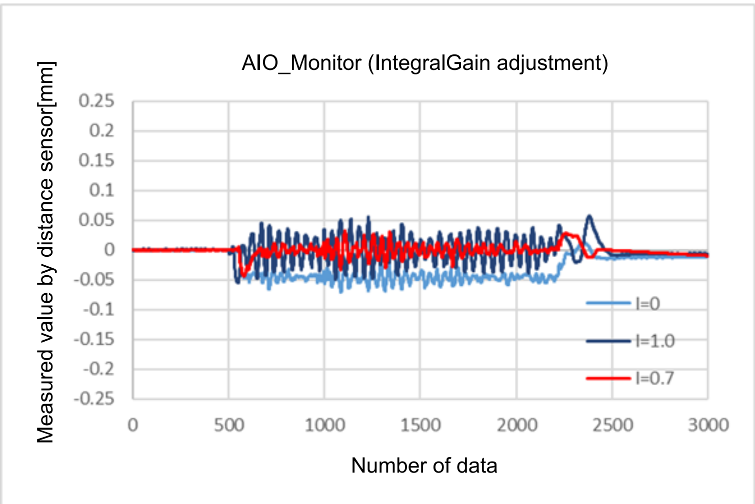
IntegralGain is a parameter to clear an offset between the target value.

Adjust value of IntegralGain and repeat the program execution to calculate a proper value.

Be sure to increase the value of IntegralGain gradually. Changing the value to a larger one at one time is extremely hazardous and the robot may move unintentionally.

When adjusting IntegralGain, keep ProportionalGain to the value calculated below and DifferentialGain to "0".

ProportionalGain Setting



When adjusting IntegralGain, offset between the target value is cleared.

However, if increasing the value too much, the robot motion will be vibratory. (Refer to the upper graph: I=1.0)

IntegralGain value without robot vibrations and the best correction accuracy is an optimum value. (Refer to the upper graph: I=0.7)

19.4.7 DifferentialGain Setting

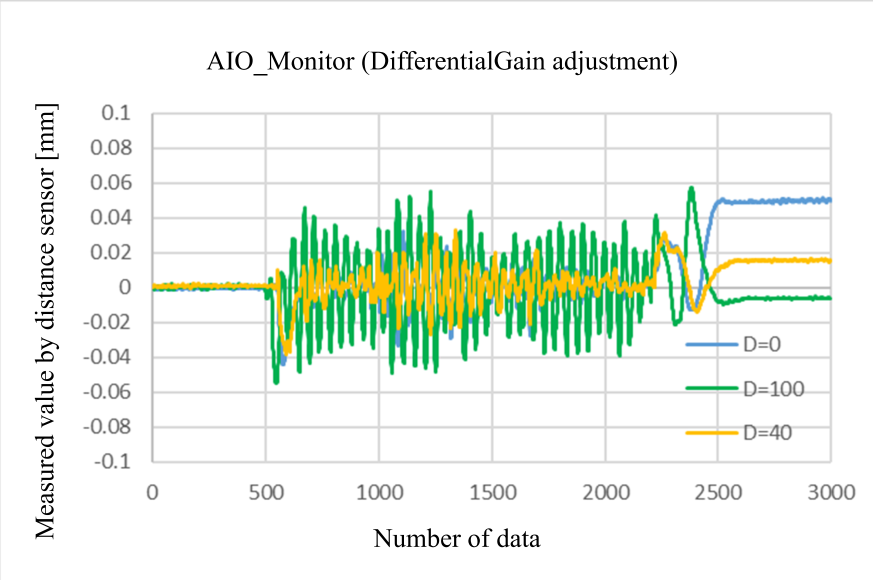
DifferentialGain is a parameter to improve the responsiveness of correction.

Adjust value of DifferentialGain and repeat the program execution to calculate a proper value.

Increase the value of DifferentialGain gradually. Changing the value to a larger one at one time is extremely hazardous and the robot may move unintentionally.

When adjusting DifferentialGain, enter the value calculated below.

- ProportionalGain: **ProportionalGain Setting**
- IntegralGain: **IntegralGain Setting**



When adjusting DifferentialGain, the responsiveness of correction is improved.

However, if increasing the value too much, the robot motion will be vibratory. (Refer to the upper graph: D=100)

DifferentialGain value without robot vibrations and the best correction accuracy is an optimum value. (Refer to the upper graph: D=40)

Now, gain adjustment is completed.

19.5 Example of Dispense Application

The following describes a program example when using the distance tracking function for dispense applications.

⚠ CAUTION

- The parameters used in this step are reference values. Please note that the operation may not be successful or the motion may be vibratory depending on the set parameters and some operating conditions.
- If the robot performs an action different from the setting, immediately press the emergency stop button.

19.5.1 Basic Example

It is a program using the distance tracking function when the robot moves from P1 to P2. Dispenser is connected to output No.1 of the standard I/O.

For details on standard I/O connections, refer to the following manuals:

- "Robot Controller RC800-A Series Manual" - Analog I/O Board
- Robot Controller RC700 Series Manual - I/O Connector
- "Robot Controller RC700-D Manual - I/O Connector"
- "Robot Controller RC700-E Manual - I/O Connector"
- "Robot Controller RC90 Series Manual - I/O Connector"

```
Function AIOTrackingSample
  '------------- Robot configuration -------------
  Motor On
  Power High
  SpeedS 30
  AccelS 300, 300
  Tool 1
  '------------- Motion part -------------
  Move P1                                 'Move to start point
  AIO_TrackingSet 1, -1, 0, -3, 3, 0, 2  ' Sets the distance tracking function
  AIO_TrackingStart 1, 10, 0, 0          ' Starts the distance tracking function
  Move P2 !D1; On 1; D99; Off 1!         ' Move to target point, start and end of
dispenser application
  AIO_TrackingEnd                        ' Ends the distance tracking function
  Motor Off
Fend
```

19.5.2 Example with Application Amount Control

It is a program example when controlling the application amount depending on the robot speed.

This program can prevent amassing fluid at the start point, end point, and corners.

When using this function, a dispenser with "external input of the application amount" is required.

For adjustment procedures of application amount and connection method, refer to the manual of the dispenser to be used.

```
Function Main
  '------------- Robot configuration ---------------
  Motor On
```

```
  Power High
  SpeedS 30
  AccelS 300, 300
  Tool 1

  AIO_Set 1, On, RealTCPSpeed, 100, 0    ' Start the analog output of the robot
speed
  ' ------------- Motion part ---------------
  Move P1                                'Move to start point
  AIO_TrackingSet 1, -1, 0, -3, 3, 0, 2  ' Sets the distance tracking function
  AIO_TrackingStart 1, 10, 0, 0          ' Starts the distance tracking function
  Move P2 !D1; On 1; D99; Off 1!         ' Move to the end point
  AIO_TrackingEnd                        ' Ends the distance tracking function
  AIO_Set 1, Off                         ' End the analog output of the robot
speed
  Motor Off
Fend
```

20. Real-Time I/O

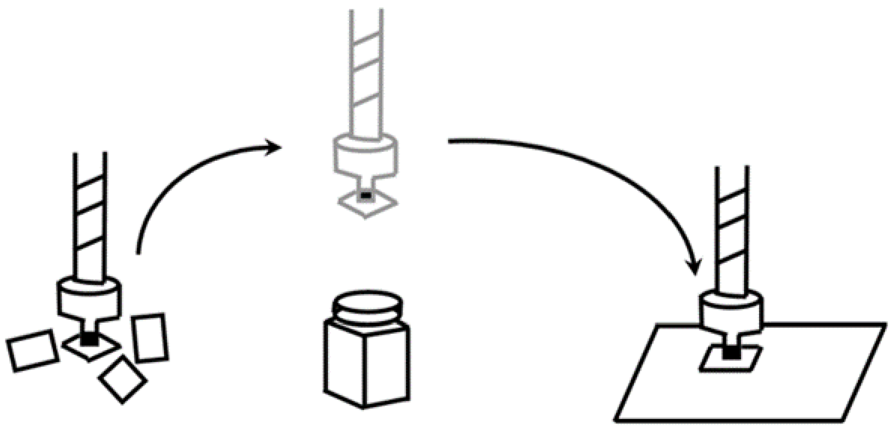
This function can only be used with the controller RC700 series/RC800 series.

20.1 Overview

Real-time I/O is a feature that allows you to input trigger signals into the R-I/O connector of the robot controller so that you can latch and acquire the robot position at high speed while it is operating.

An example of an application using real-time I/O is "Picture on the fly": This synchronizes the robot position detection and the vision position detection, and performs part pickup, alignment, and assembly without stopping the robot.

With the real-time I/O feature, you can reduce the robot stop time for vision image acquisition that is necessary for traditional vision applications.



20.2 Specifications

R-I/O Connector

The RC700 series/RC800 series robot controller has an R-I/O connector that is used to connect the real-time I/O trigger input signals. An R-I/O input is a special input interface that monitors the signals at higher speed than the standard I/O inputs. There are two trigger input signals on each of the Control Unit and Drive Units. For example, set the transmission type sensor so that it reacts when the robot passes the camera acquisition point and use the R-I/O connector so that the R-I/O input is detected at the moment shutter clicked.

For details on hardware (connection connectors and circuits), refer to the following manual:

"Robot Controller Manual - I/O Remote Settings"

Real-time I/O commands

There are special commands provided for use the real-time I/O. The following are basic descriptions of these commands.

For details, refer to the following manual.

"SPEL+ Language Reference"

LatchEnable

> This command is used to enable or disable the latch function of the robot position information with the real-time I/O. When LatchEnable On executes, it enables the robot position latch function using the trigger input signals connected to the R-I/O connector. After the latch is enabled, it is possible to latch the number of consecutive latches (up to 4 times) specified by SetLatch. To repeatedly latch the robot position, execute LatchEnable Off and then execute LatchEnable On again. To use the command repeatedly, it requires 60 ms minimum interval for each command processing time. It is not necessary to consider the command executing time.

SetLatch

Specifies the real-time input port that you connected the trigger input signal to, the input logic, and the number of consecutive latches. The table below shows the port numbers you can specify. Specify the port number that the robot using R-I/O is connected. If the other ports are specified, an error will occur. One robot cannot wait for the trigger signals from multiple ports.

RC700 series

		Point	Port number
Control Unit	INPUT	2 points	24, 25
Drive Unit 1	INPUT	2 points	56, 57
Drive Unit 2	INPUT	2 points	280, 281
Drive Unit 3	INPUT	2 points	312, 313

RC800 series

		Point	Port number
Control Unit	INPUT	4 points	24, 25, 26, 27

Execution of SetLatch requires approximately 40 ms for processing.

LatchState Function

This function confirms whether the selected robot's position information is completely latched. After it confirms that the latch has been done, it acquires the position information using the LatchPos Function.

LatchPos Function

This function returns the robot position information latched by the trigger input. Executing the LatchPos Function needs approximately 15 ms for processing.
To return Tool 0 and Arm 0 positions: Set the WithoutToolArm parameter when using the "Picture on the fly" application.

RobotPos Vision Sequence Property

When acquiring the parts place position by using the RobotPlacePos result, set the robot position in image capturing to this property before acquiring the RobotPlacePos result.
Also, set the RobotPos sequence property to set the robot coordinates of the image acquisition position to calculate the part position when you use a mobile camera system.
In any of the above cases, the system can calculate the correct part position by using the position acquired by LatchPos Function in this property.
For details, refer to the following manual.
"Vision Guide 8.0 Properties and Results Reference"

Latch accuracy

The following is the theoretical sampling time used to latch the position information.

		Sampling time [μsec]
Control Unit	4-axis robot	32
	6-axis robot	32
Drive Unit *	4-axis robot	32
	6-axis robot	21

- RC700 series only

You can get a rough idea of latch accuracy from the robot speed (parts moving speed) at the latch trigger input and the sampling time. For real accuracy, you must have a margin on the required accuracy because time delay and variation in the hardware may affect. The latch accuracy will improve as the robot moves slower at the trigger input.

Latched position accuracy [mm] = Robot speed [mm/sec] × Sampling time [sec]

20.3 Usage

1. Basic Example

The following program is a sample to connect any trigger signal to the R-I/O connector of the controller, latch the robot position information while it is operating at the trigger input, and show the latched position information.

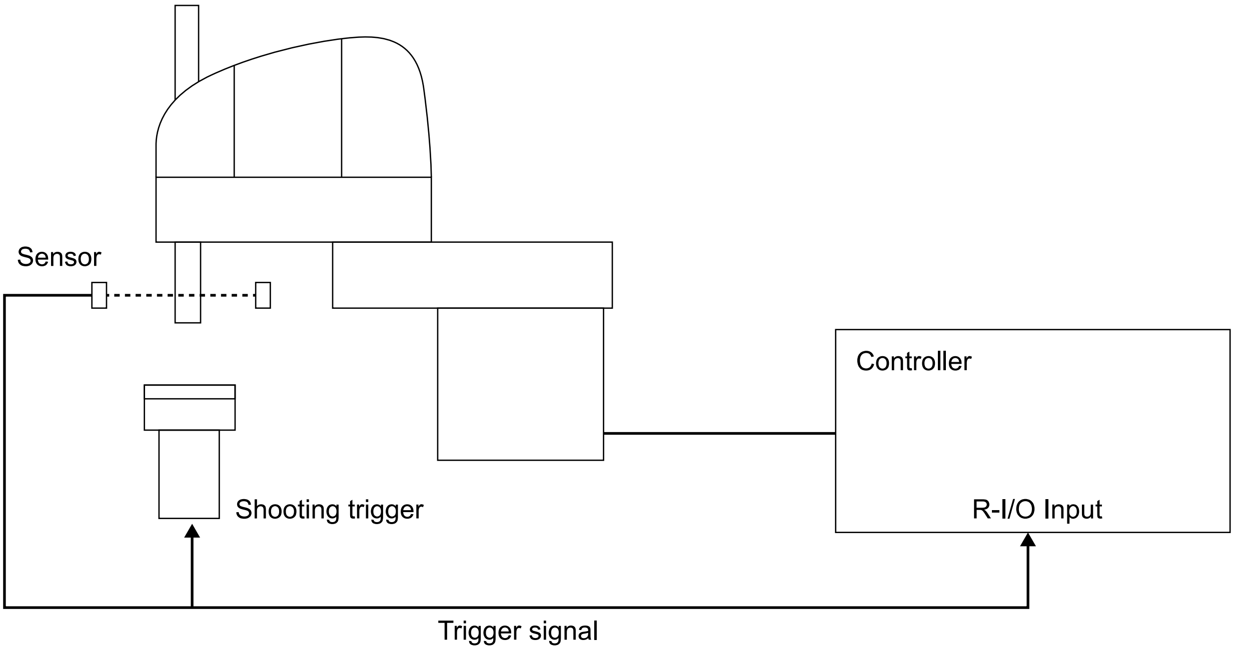
```
Function Main
  Motor On
  Power High

  Speed 50; Accel 50, 50
  SpeedS 500; AccelS 5000

  Go P0                                'Start position
  SetLatch SETLATCH_PORT_CU_0,
SETLATCH_TRIGGERMODE_LEADINGEDGE, 4
  LatchEnable On                       'Latch enabled
  Move P1                              'Starts operation, inputs trigger while the
operation

  Wait LatchState = True               'Confirmed the latch is complete
  P3 = LatchPos(WithoutToolArm, 1)     'Retrieves latch position 1
  P4 = LatchPos(WithoutToolArm, 2)     'Retrieves latch position 2
  P5 = LatchPos(WithoutToolArm, 3)     'Retrieves latch position 3
  P6 = LatchPos(WithoutToolArm, 4)     'Retrieves latch position 4
  LatchEnable Off                      'Disable the latch

  Print P3                             'Displays latch position 1
  Print P4                             'Displays latch position 2
  Print P5                             'Displays latch position 3
  Print P6                             'Displays latch position 4
Fend
```

Example to omit parameters:

```
Function Main
  Motor On
  Power High

  Speed 50; Accel 50, 50
  SpeedS 500; AccelS 5000

  Go P0                                'Start position
  SetLatch SETLATCH_PORT_CU_0,
SETLATCH_TRIGGERMODE_LEADINGEDGE
  LatchEnable On                       'Latch enabled
  Move P1                              'Starts operation, inputs trigger while the
operation

  Wait LatchState = True               'Confirmed the latch is complete
  P3 = LatchPos                        'Acquire the latched position
  LatchEnable Off                      'Disable the latch

  Print P3                             'Show the latched position
Fend
```

2. Example with Vision system

This is an example that uses the robot end effector to handle parts, passes above the external fixed upward camera acquisition point without stopping, and assembles the parts with an appropriate position correction.



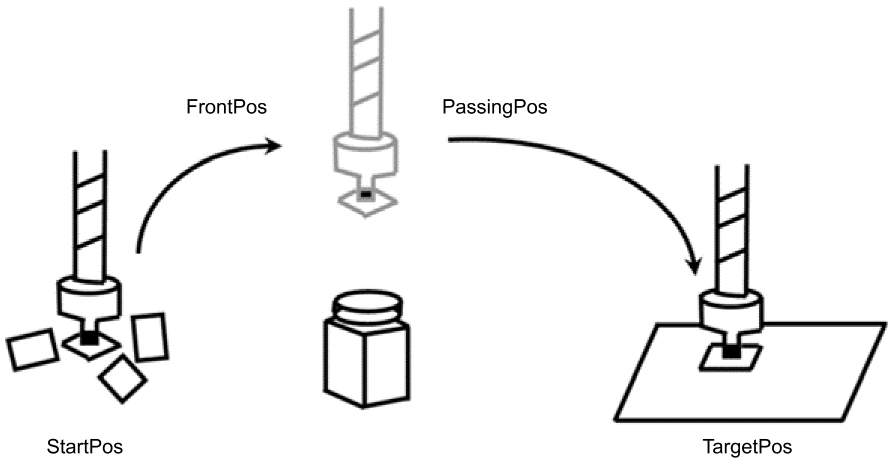
This system has a transmission type sensor that outputs the trigger signal when the robot end effector handles a part and passes the camera acquisition point. Then, it connects the sensor output with both the R-I/O and the camera trigger input for external tuning and synchronizes the latched robot position information and the camera image. It calculates the part position error and offsets the position comparing the robot position information from the camera image to the robot position information from the real-time I/O.

In this case, the robot vision system must be calibrated as the upward fixed camera. Also, by registering the parts place position in advance, robot position information can be acquired for precise parts placement by RobotPlacePos result. Parts place position can be set in the CalRobotPlacePos property wizard.

For details on the camera trigger signal connection and the vision calibration, refer to the following manuals:

"Vison Guide 8.0"

The following program is a sample.




```
Function Main
  Robot 1
  Motor On
  Power High
```

```
  Speed 100
  Accel 100, 100

  Jump InitPos                                    'Moves to the initial point
  Wait 1.0

  SetLatch 24, SETLATCH_TRIGGERMODE_LEADINGEDGE  'Sets the latch condition

  MemOff 0
  Xqt PictureOnFly_Camera                         'Starts the shooting task

  Jump StartPos C0                                'Moves to the part feed point
  Wait 0.5

  LatchEnable On                                  'Starts waiting the latch

  MemOn 0                                         'Enables the shooting

  Jump FrontPos C0 CP                             'Moves close to the camera
  Go PassingPos CP                                'Passes over the camera

  Go TargetPos :Z(-70) CP                         'Moves over the assembly point

  Wait MemSw(1) = On                              'Waits until the image
processing is complete
  Wait LatchState = True                          'Waits for position latch
'completion
  LatchEnable Off                                 'Disables the position latch
  Jump ExactTargetPos C0 LimZ (-70)               'Moves to the assembly point
  Wait 0.5

  Jump InitPos                                    'Moves to the initial point
  Wait 0.5

  Motor Off

Fend
```

Function to execute from work image capturing to work place acquisition

```
Function PictureOnFly_Camera

  'Vision Result variable
  Integer AcqStat         'Strobe imaging completion flag
  Boolean Found           'Work detection status

  Wait MemSw(0) = On      'Waits the imaging start flag
  MemOff 1                'Clears the imaging completion flag
  MemOff 0                'Clears the imaging start flag
  AcqStat = 0             'Clears the imaging start flag

  VRun PictureOnFly_i


  Do Until AcqStat = 3    'Wait for strobe
    VGet PictureOnFly_i.AcquireState, AcqStat
  Loop

  'Check the work detection
  VGet PictureOnFly_i.Geom01.Found, Found

  If Found = False Then
    Print "Work NotFound"
    Pause
  EndIf
```

```
  Wait LatchState = True  'Waits for the trigger

  'Sets the image capturing position (trigger position) to the Vision
  VSet PictureOnFly_i.RobotPos, LatchPos (WithoutToolArm)

  'Acquires the robot position
  VGet PictureOnFly_i.Geom01.RobotPlacePos, Found, ExactTargetPos

  MemOn   1               'Changes the camera imaging flag
Fend
```

21. Additional Axis

21.1 Overview

You can attach up to two additional drive axes (per manipulator) which can operate in association with the manipulator. The position data of the additional axis is saved with the robot point data. The additional axis can move simultaneously with the manipulator by motion commands and you can design an application using a traveling axis (manipulator on the straight axis) with simple programming.

KEY POINTS

If you want to operate the manipulator and drive axis separately, you need to define the additional axis as another manipulator using the multi-manipulators feature.

CAUTION

When you use the additional axis as traveling axis and mount a manipulator(s) on the axis, the reaction force of manipulator(s) is put on the traveling axis. Therefore, you should limit the acceleration/deceleration speed with the Accel setting so that it is within the allowable inertia of traveling axis. In addition, the manipulator may swing widely at the positioning and possibly break the additional axis.

21.2 Specifications

Types of additional axis

The supported additional axis is the PG axis, controlled by the pulse generator board. However, note that the PG axis has some limitations.

Limitations of a PG additional axis:

- Synchronizes with the manipulator to start motion but not to finish.
- Does not support Path motion with CP On and Pass. Stops for every motion.
- Does not go through the CVMove series of points.
- Calibration is necessary using the MCAL command. Cannot operate the additional axis and the robot together until the calibration is complete. If movement of the PG additional axis is "0", and Go and Move are executed to the point where only the robot moves, the robot will move singly.

Number of additional axis

Up to two additional axes are available for each of the SCARA robot series (including RS series), Cartesian coordinate robot, 6-axis robot (including N series), and Joint type robot. However, the number of axes you can add is determined by how many axes are available with your controller.

Position data management

The additional axes are allocated to Joint #8 and #9 for all robot types. The position data are shown in the S and T coordinate values of point data of the manipulator to which you add the additional axes.

The additional axis as Joint #8 is called the additional S axis and Joint #9 is the additional T axis.

The coordinate values of additional axes are saved with the robot point data but don't have any effect on the robot coordinate system.

How to operate

The additional axis can move simultaneously with the manipulator. (synchronous start / stop) However, if you use the PG axis, it doesn't synchronize with manipulator to finish and operate by the different acceleration/deceleration speed from the manipulator. See below for details on motion commands.

Also, you can operate the additional axis and manipulator separately by proper management of the point data. However, you cannot operate separately both of them in arbitrary timing. In this case, use the multi-manipulators function and set the drive axis as another manipulator.

Command specification

Pulse, Go, BGo, TGo, Pass
: The additional axis can operate in association with the manipulator motion. However, if you use the PG axis, it synchronizes only to start the motion and a motion command completes when both the manipulator and axis finish each motion. In addition, if the PG additional axis has a travel distance, the Path motion with CP On and Pass are prohibited and the axis moves with CP Off automatically.

Move, BMove, Tmove
: The additional axis can operate in association with the manipulator motion. However, if you use the PG axis, it synchronizes only to start the motion and a motion command completes when both the manipulator and axis finish each motion. In addition, if the PG additional axis has a travel distance, the Path motion with CP On is prohibited and the axis moves with CP Off automatically.

Arc, Arc3
: The additional axis can operate in association with the manipulator motion. It doesn't go through the specified midPoint and directly goes to the end point. If you use the PG axis, it synchronizes only to start the motion and a motion command completes when both the manipulator and axis finish each motion. In addition, if the PG additional axis has a travel distance, the Path motion with CP On is prohibited and the axis moves with CP Off automatically.

CVMove
: The additional axis can operate in association with the manipulator motion. If you use a servo axis for the additional axis, for each of the S and T axis it creates a curve going through the S and T coordinates specified by a series of point data. However, if you use the PG axis for the additional axis, it doesn't go through the series of points and directly goes to the end point. Also, it synchronizes only to start the motion and a motion command completes when both the manipulator and axis finish each motion. In addition, if the PG additional axis has a travel distance, the Path motion with CP On is prohibited and the axis moves with CP Off automatically.

Jump
: The additional axis executes PTP motion in association with the manipulator horizontal motion. However, if you use the PG axis, it synchronizes only to start the motion and a motion command completes when both the manipulator and axis finish each motion. In addition, if the PG additional axis has a travel distance, the Path motion with CP On is prohibited and the axis moves with CP Off automatically.

Jump3, Jump3CP
: The additional axis can operate in association with the manipulator depart / span / approach motion. However, if you use the PG axis, it synchronizes only to start the motion and a motion command completes when both the manipulator and axis finish each motion. In addition, if the PG additional axis has a travel distance, the Path motion with CP On and Pass are prohibited and the axis moves with CP Off automatically.

JTran, PTran
: The additional axis can operate separately by specifying as Joint #8, #9.

Example:

```
> JTran 8, 90     ' Move the additional S axis by 90 mm
> PTran 9, 10000  ' Move the additional T axis by 10000 pulse
```

21.3 Usage

Additional axis configuration

For the instruction of configuring the additional axis, see below.

Configuration of Additional Axes

If you use the PG axis for the additional axis, you need to set the PG parameters. For details on PG parameters, refer to the following manual:

"Robot Controller Option PG Motion System Manual"

Point data usage

This example specifies the position data of manipulator and additional ST axes and substitutes them to the point data.

```
P1 = XY(10, 20, 30, 40) :ST(10, 20)          ' SCARA robot
P1 = XY(10, 20, 30, 40, 50, 60) :ST(10, 20)  ' 6-axis robot
```

This example specifies the position data of manipulator and additional ST axes and executes a PTP motion.

```
Go XY(10, 20, 30, 40) :ST(10, 20)
Go XY(10, 20, 30, 40, 50, 60) :ST(10, 20)
```

This example specifies the position data of additional ST axes individually.

```
P1 = XY(10, 20, 30, 40) :S(10) :T(20)
P1 = XY(10, 20, 30, 40) :S(10)
P1 = XY(10, 20, 30, 40) :T(20)
```

This example omits the robot position assignment XY( ) and specifies only the additional axis position. Then, the point data is defined so that the manipulator doesn't move (undefined).

```
P1 = ST(10, 20)
Go P1             ' Only additional axis moves and the manipulator remains at the
current position
```

This example operates only the additional axis.

```
Go ST(10, 20)     ' Only the additional axis moves
```

This example omits the additional axis position assignment ST( ) and specifies only the manipulator position. Then, the point data is defined so that the additional axis doesn't move (undefined).

```
P1 = XY(10, 20, 30, 40)
Go P1             ' Only the manipulator moves and the additional axis remains at
the current position
```

This example operates the manipulator only.

```
Go XY(10, 20, 30, 40)  ' Only the manipulator moves
```

This example calculates the additional axis coordinate value using a point operator expression.

```
P1 = XY(10, 20, 30, 40, 50, 60) :ST(10, 20)
P2 = P1 + S(10) + T(20)  'Add the offset amount to the additional ST axes for P1
```

Note that you cannot use the point operator for undefined points.

```
P1 = XY(10, 20, 30, 40, 50, 60)
P2 = P1 + S(10) + T(20)                        ' Error (ST are undefined for P1
and you cannot use the point operator)
P1 = XY(10, 20, 30, 40, 50, 60) +ST(10, 20)     '  Error
P1 = XY(10, 20, 30, 40, 50, 60) +S(10) +T(20)   ' Error
Go ST(10, 20) + X(10)                          ' Error (XY are undefined and you
cannot use the point operator)
```

This example shows the additional ST axes coordinate values retrieved from the point data.

```
Print CS(P1),  CT(P1)
```

Pallet motion

When you specify a pallet with the point data including the position data of additional axis, the position data of additional axis is also calculated by the pallet operator. If you use the additional axis as traveling axis, you can define a wide range pallet than for a single manipulator.

Also, if you want to use the additional axis not as traveling axis and exclude the additional axis position from the pallet operator, define the pallet with the point data that clears the additional axis position data.

22. Absolute Accuracy Calibration

Each function and applicable models of the absolute accuracy calibration are as follows.

Function \ Model		GX4-A, GX8-A,GX4-B, GX8-B	Other Models
Arm Length Calibration	Option for limited models	OK	N/A
Area distortion correction	Standard function	OK	OK
Joint Accuracy Calibration	Standard function for limited models	OK	N/A

22.1 Overview

Differences in ideal robot performance and actual robot performance are caused by mechanical errors and the robot's structure. Absolute accuracy calibration compensates for these differences to ensure that the actual position and trajectory of the robot match the coordinates and trajectory specified.

Using absolute accuracy calibration to improve robot accuracy is expected to provide the following benefits.

- Reduce the number of teaching points required
- High-precision assembly
- Reduction of start-up time during restoration

22.2 Arm Length Calibration

22.2.1 Overview

Arm length calibration is one of the absolute accuracy calibration functions. It is used to measure the actual length of each robot arm, and correct any deviation between the intended robot position and the actual robot position.

22.2.2 Replacing Parts Requiring Arm Length Calibration Remeasurements

Arm length measurements need to be performed again when replacing the following parts.

- Reduction gear unit replacement
- Ball screw spline unit replacement

For more information of parts, contact a supplier of your region.

22.2.3 Arm Length Measurements

Arm length calibration measurements require precise measurements that cannot be performed by the user. When purchasing an arm length calibration license, arm length calibrations are completed prior to delivery based on factory arm length measurements.

We provide an arm length calibration remeasurement service for users replacing parts. Please inquire with your supplier.

22.2.4 Enabling and Disabling Arm Length Calibration

Arm length calibration can be enabled and disabled using the following commands.

```
ArmCalib On | Off
```

⚠ CAUTION

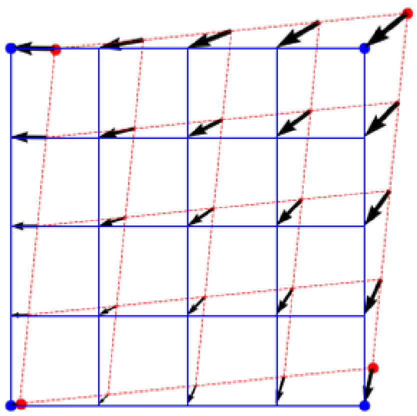
- Enabling and disabling arm length calibration will cause teaching positions to become misaligned. Perform the position teaching procedure again.
- Enabling arm length calibration will render parts of the movable range described in the manual as inaccessible. This is because of differences in the logical robot arm length and the actual robot arm length.

22.3 Area distortion correction

22.3.1 Overview

The area distortion correction function corrects the positions of points when reference points on a drawing and the positions of those points that the robot is actually taught are different. Corrections are made within the area bounded by the selected reference points.

Using the area distortion correction function, teaching can be omitted for points within the area bounded by the reference points.



🔴 : Reference point positions on the drawing

🔵 : Actually taught positions

22.3.2 Command

Here is a list of SPEL+ commands for the area distortion correction function.

- AreaCorrectionSet: Sets and displays a correction area
- AreaCorrectionDef Function: Returns correction area settings
- AreaCorrectionClr: Deletes a correction area
- AreaCorrection Function: Returns corrected points
- AreaCorrectionInv Function: Returns corrected points to their original condition
- AreaCorrectionOffset Function: Returns positions relatively displaced from corrected points
- AreaCorrectionSet: Sets and displays a correction area

For details on commands, refer to the following manual:

"Epson RC+ 8.0 SPEL+ Language Reference"

22.3.3 How to use

Setting reference points

The area distortion correction function works inside the area set as the correction area. Therefore, reference points must be set to surround the operating point. The reference points are set for the robot.

For the reference points, use points whose positions relative to each other are precisely known. For example, use a reference hole of the device or a point with a tight position tolerance. Since corrections are made using the correspondence of taught points, the use of inaccurate reference positions may result in inaccurate correction results.

Increasing the number of reference points may improve accuracy.

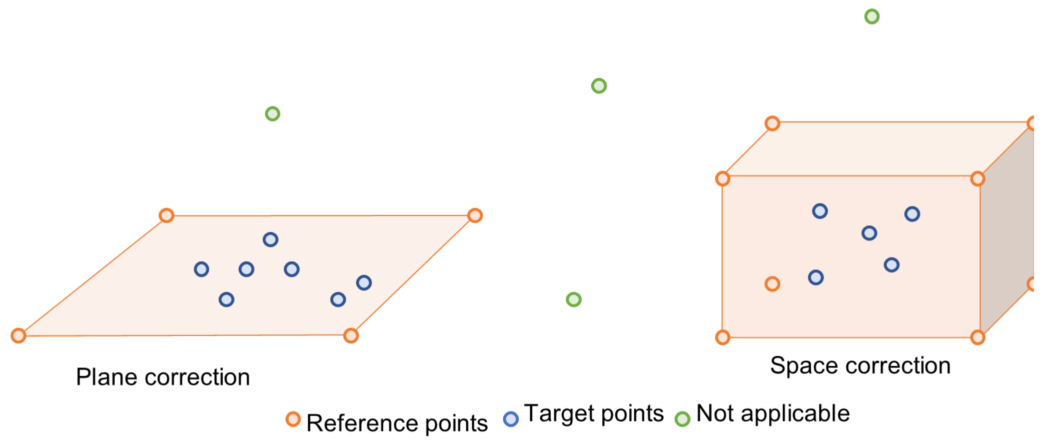
You can use Plane correction and Space correction.

Plane correction

Plane correction allows you to make corrections to points on a plane composed of the points you have selected as reference points. If plane correction is selected, place the refPointList on a plane. A minimum of three reference points is required.
When plane is selected as the correction type, the effect of the correction is reduced at points vertically distant from the plane selected for Kind (the type of correction). Set the correction area at an appropriate height or, if a reference point can be established in the height direction, specify space for Kind (the type of correction).

Space correction

Space correction allows you to make corrections to points in three-dimensional space composed of the points you have selected as reference points. When selecting space correction, make sure that the refPointList surround the area to be corrected. A minimum of four reference points is required.
Save the positions of the reference points on the drawing as point data. Point numbers must be continuous. For example, if you choose four reference points, prepare four contiguous areas in the point file.

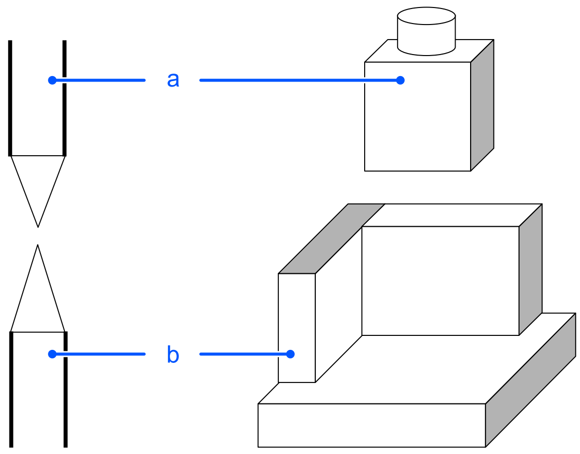


To correct point data using the area distortion correction function with a vertical 6-axis robot (including N series), the tool coordinate system's Z axis at the point to be corrected must match the tool coordinate system's Z axis at the reference point of the correction area. The angle formed by the tool coordinate system's Z axis can be acquired by specifying COORD_Z_PLUS as the axis number for the DiffToolOrientation function.
With a SCARA robot (including RS series), correction is effective in any orientation.
If you want to maintain high accuracy even at a point where the orientation has rotated in the tool coordinate system's Z-axis direction from the reference point, it is effective to add the rotated orientation point to the reference points.

Reference point teaching

When teaching, use a method that will ensure as much accuracy as possible. For example, use a device-side reference point and a tool as shown in the figure below.

- Reference point: Teaching point is easy to be found
- Tool: Tip point is easy to be found



Correction of operating points

After teaching has been performed for all reference points, use AreaCorrectionSet to set the correction area. Set P1 to P4 as reference point positions on the drawing. Also, set P11 to P14 as the points where the reference points were actually taught. To set up a plane correction for correction area 1, set as follows:

```
AreaCorrectionSet 1, P(1:4), P(11:14), MODE_PLANE
```

If you want to apply the correction to operating point P20 in the correction area, write the following:

```
Go AreaCorrection(P20, 1)
```

For more information about each command, see the following manual:

"Epson RC+ 8.0 SPEL+ Language Reference"

⚠ CAUTION

- Set the correction area for each tool. If the correction is performed using a tool different from the tool taught by area number, the position may be inaccurate.
- The correction area is valid until the controller power is turned off.

  To enable the correction area, execute AreaCorrectionSet on the points recorded in the point file.

22.3.4 During restoration

Using the area distortion correction function, teaching points can be omitted, potentially reducing the time required for reteaching that accompanies restoration. To omit teaching points during restoration, the following conditions must be met:

- Reference points are set.
- Teaching is performed for the reference points prior to restoration.
- Points to be restored are inside the correction area, and the correction is valid.

If only corrected point data is available, or if an actually taught point is used as an operating point, use the AreaCorrectionInv function to return to the point that existed before correction.

Teach the reference points again and create new point data. Apply the AreaCorrectionSet function to these teaching points to create a new correction area.

For the points prior to correction, make a correction using the newly defined correction area. The position will be closer to the original position than if using the uncorrected points directly.

An example is shown below.

```
' Assume it is defined by correction area 1 before restoration

P21 = AreaCorrectionInv(P121,1)  'P121 is a point created by teaching
P22 = AreaCorrectionInv(P122,1)  'P122 is the point after conversion

' Reteach the reference points to P101 to P104
' Newly set P1 to P4 and P101 to P104 as correction areas

AreaCorrectionSet 2, P(1:4), P(101:104), MODE_PLANE

' Apply new correction areas
' (P121, P122, and P123) are used as operating points

P121 = AreaCorrection(P21, 2)
P122 = AreaCorrection(P22, 2)
P123 = AreaCorrection(P23, 2)
```

22.3.5 When reconfiguration of the area is needed

The area must be set up again when the following operations have been performed:

- Reduction gear unit replacement
- Ball screw spline unit replacement
- AC servo motor replacement
- Timing belt replacement
- Calibration
- Body installation

22.4 Joint Accuracy Calibration

22.4.1 Overview

Joint accuracy calibration is one of the absolute accuracy calibration functions. It is used to measure the joint accuracy of each axis, and correct any errors.

22.4.2 Replacing Parts Requiring Joint Accuracy Calibration Remeasurements

Joint accuracy measurements need to be performed again when replacing the following parts.

- Reduction gear unit replacement
- AC servo motor replacement

- Timing belt replacement

For more information of parts, contact a supplier of your region.

22.4.3 Joint Accuracy Measurements

Joint accuracy calibration contains factory settings. Joint accuracy calibration remeasurements for part replacing shall be performed by personnel who has taken a proper training.

For details, refer to the following manual.

"Safety Manual - Training"

Trajectory accuracy is improved within the range specified in the joint accuracy calibration wizard. Calibrate joint accuracy at points actually in use. Joint calibration is performed across all motion ranges at the factory.

⚠ CAUTION

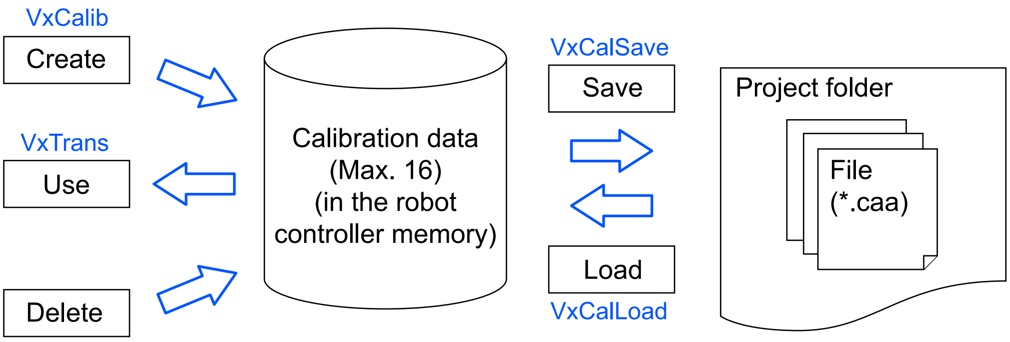
- Do not change joint accuracy calibration settings unless absolutely necessary. Changing these settings unnecessarily may adversely impact trajectory accuracy.
- Trajectory accuracy may drop outside the range specified in the joint accuracy calibration wizard. To improve accuracy across the entire range of motion, specify a joint accuracy calibration range to cover the entire motion range.

23. Calibration of Commercial Vision Sensor and Robot

23.1 Overview

When using the commercially available vision sensors or image processing systems, instead of our Vision Guide, it is necessary to calibrate the image processing result (image coordinate system, camera coordinate system) with the robot coordinate system. This chapter describes the calibration procedure.

The figure below shows commands and functions related to calibration and behaviors of data and files.



The vision calibration data can be created in the following steps.

1. Camera Installation
2. Create the image processing sequence for calibration(in each vision sensor)
3. Teach the robot position for calibration at necessary parts
4. Perform image processing at necessary parts and acquire the image processing result.
5. Execute calibration (VxCalib command)
6. Save calibration data (VxCalSave command)

KEY POINTS

If you are using our Vision Guide option, refer to the "Vision Guide manual". Calibration with the Vision Guide option can be configured by the wizard easily.

CAUTION

We cannot answer the questions regarding communication settings and usage for commercial vision sensors. Please contact the manufacture directly.

23.2 Specifications

Calibration data/Calibration file

Up to 16 calibration data can be saved to the robot controller at the same time.

If you are using more than 16 calibration data, load them from the file and save to the file.

Up to 16 files can be created. Be careful not to exceed the maximum number of files.

Camera installation

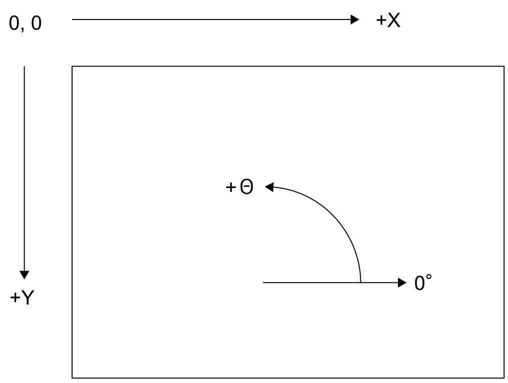
Following seven camera mounting types are supported. See details below.

Camera Installation

1. Standalone
2. Fixed downward
3. Fixed upward
4. Mobile camera on the Joint #2
5. Mobile camera on the Joint #4
6. Mobile camera on the Joint #5
7. Mobile camera on the Joint #6

Image coordinate system

Following image shows the adapted image coordinate system. The unit is pixel.



23.3 Camera Installation

Camera installation method can be selected for each calibration data. The data set required for calibration differs depending on mounting types. Note that the wrong setting may result in improper calibration.

Epson RC+ 8.0 supports the following camera installations:

Camera Installation	Description
Standalone	Camera can be installed anywhere. Camera does not have a relation with the robot. With this method, position information in the Robot coordinate system cannot be acquired. However, it can be converted from the image coordinate system to the camera coordinate system. That is, simple length measurement can be performed.
Fixed downward	Camera and target objects do not move and is looking down into robot work envelope. The camera acquires position information in the Robot coordinate system. The camera must be installed vertically to the XY plane of the specified coordinate system. (Angle gap may result in poor accuracy) Specified coordinate systems are Robot coordinate system and Local coordinate system. Uses nine reference points.
Fixed upward	Camera does not move and is looking up into a portion of the robot work envelope. For example, this installation method is used to check the position of the object which is carried by the robot. It does not require a reference point. The calibration target is on the end effector or the object held by the robot.

Camera Installation	Description
Mobile camera on the Joint #2	Camera is mounted on Arm #2 on SCARA robot or Joint #2 on Cartesian robot. Uses one reference point.
Mobile camera on the Joint #4	Camera is mounted on Arm #4 on SCARA robot or Joint #4 on Cartesian robot. Uses one reference point.
Mobile camera on the Joint #5	Camera is mounted on Joint 5 on 6-Axis robot. Uses one reference point.
Mobile camera on the Joint #6	Camera is mounted on Joint 6 on 6-Axis robot. Uses one reference point.

23.4 Reference Points

Reference points are important points to be used to calibrate the relation of the image coordinates and the Camera or Robot coordinate systems.

Each calibration scheme requires one or more reference points. The methods for teaching these points vary according to the camera mounting and orientation.

For Standalone camera calibration, you manually enter the coordinate values of the reference points into the system.

For all other camera calibrations, you teach the reference points using the robot.

23.5 Reference Points for Mobile Camera

This scheme requires one reference point. Also, TowRefPoint parameter can be specified. If TowRefPoint parameter is True, a pair (two points) of the position data is required for the reference point. Each position data includes two points whose U-axis values are 180 ° apart in the specified coordinate system. By the TwoRefPoint function, the system can determine more precise position of the reference position in the robot coordinate system. However, this function is not necessary if the robot tool is defined accurately.

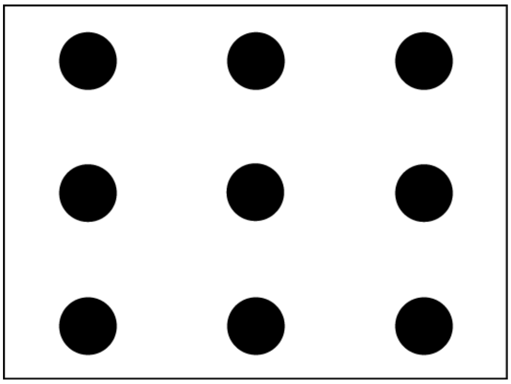
For the reference point, the taught points where the robot jogged can be used.

Here are some examples of taught reference points:

- A part or calibration target in the robot work envelope.
- A hole somewhere in the work envelope that a tool mounted on the robot end effector can be slipped into.

23.6 Reference Points for Fixed Camera

The 'Fixed Downward' and 'Standalone' calibration schemes require a calibration target plate or sheet that contains nine targets.



Fixed Camera Calibration Targets

For 'Fixed Downward' calibrations, the targets could be holes in a plate that a rod on the robot end effector can be slipped into. The distances between the targets do not have to be exact.

For Standalone cameras, a pattern sheet can be used. The horizontal and vertical distances between the targets must be known.

23.7 Command List

Following table shows the commands and functions related to Vision calibration.

For details, refer to the following manual.

"SPEL+ Language Reference"

Command name	Function
VxCalib Statement	Creates calibration data for the vision system.
VxCalDelete Statement	Deletes the calibration data.
VxCalLoad Statement	Loads the calibration data from a file.
VxCalInfo Function	Returns the calibration completion status and the calibration result.
VxCalSave Statement	Saves the calibration data to a file.
VxTrans Function	Converts pixel coordinates to robot coordinates and returns the converted point data

24. Installing Controller License

When you purchase licenses with your system, the licenses have already been installed on your system before shipment. Of course, you can purchase licenses separately.

License forms and enabling procedures vary by Controller series. See the corresponding procedure.

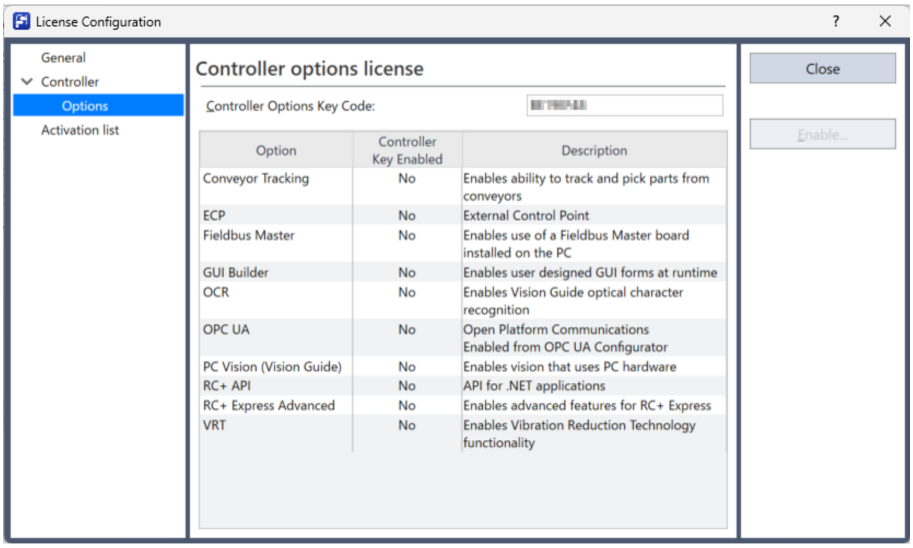
24.1 Checking License configuration

In the Epson RC+ 8.0 menu, select [Setup]-[License Configuration]. The window shown below appears. You can check and enable the license according to the Controller you are connected to.

Item	Description
Controller Connection Status	Displays the connection status of the controller.
Authentication List Status	Displays the status of the authentication list. Used to enable multiple licenses at once.

24.2 License Configuration for RC700, RC90, T, and VT Series

If you connect an RC700, RC90, T, or VT series Controller and then select [Controller]-[Options] from the [License Configuration] window, the window shown below appears. You can see which options are enabled for your system.



Item	Description
Controller Options Key Code	Shows the options key code for the connected Controller. Requires purchase of a license.
Options	Name of the option.
Controller Key Enable	Indicates that the option is enabled in the controller.
Description	A brief note for each option.

24.2.1 To enable an option license on site

1. Connect to the Controller for which you want to enable the option.

2. From the [License Configuration] window, select [Controller]-[Options].

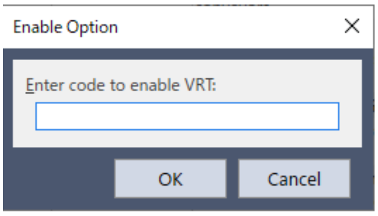
3. Record the Controller options key code shown in the window.

4. Call the supplier of your region from whom you purchased the options key code.

5. You will receive a code to enable the option from the supplier of your region.

6. Select the option to enable, and then click the [Enable] button.

7. Enter in the code you received from the supplier of your region.



> **KEY POINTS**
>
> The code is case sensitive.

24.2.2 If the DMB Board or CF card is replaced

If the DMB board or CF card is replaced due to malfunction, all configured options will be disabled. Follow the procedure in To enable an option license on site to configure the option licenses again.

* As for RC700, RC90, VT series, if the DMB board or CF card is replaced, the previous code for enabling the option cannot be used.

24.3 License Configuration for RC800 Series

To enable the RC800 series Controller license, use one the methods described below. Select an appropriate method based on your network environment.

- Online authentication

  If your PC has Epson RC+ 8.0 installed, is connected to a Controller, and can connect to the Internet, you can enable the license online.

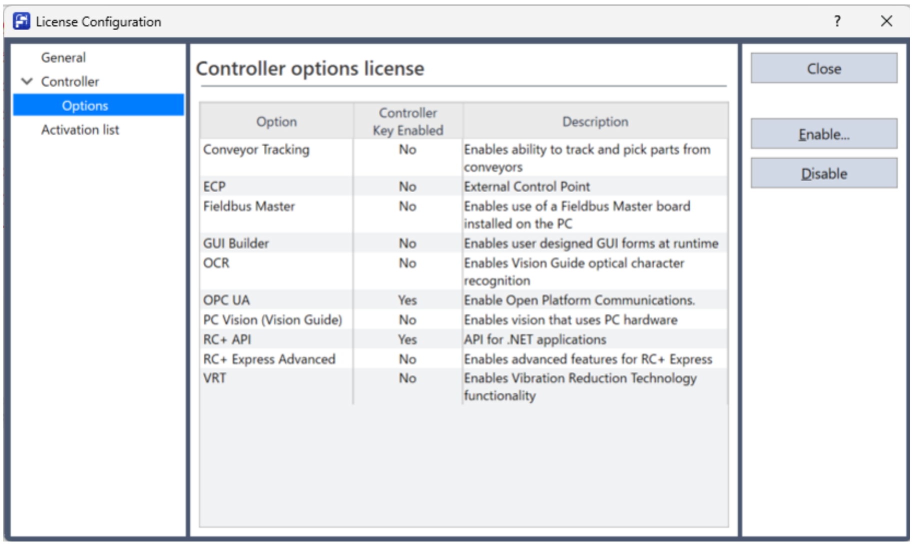
- Offline authentication (authenticate one license at a time)

  If your PC has Epson RC+ 8.0 installed, is connected to a Controller, but cannot connect to the Internet, you can enable the license offline. Acquire and transfer the authentication key file from another browser terminal.

- Offline authentication (authenticate multiple licenses together)

  If your PC has Epson RC+ 8.0 installed, is connected to a Controller, but cannot connect to the Internet, you can enable multiple licenses together. Prepare a separate PC with Internet access and Epson RC+ 8.0 installed.

If you connect an RC800 series Controller and then select [Controller]-[Options] from the [License Configuration] window, the window shown below appears. You can see which options are enabled for your system.

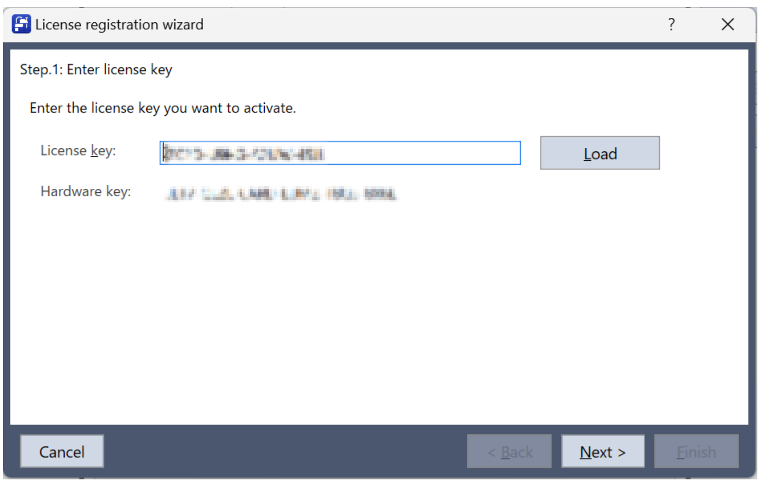


Item	Description
Options	Name of the option.
Controller Key Enable	Indicates that the option is enabled in the controller.
Description	A brief note for each option.

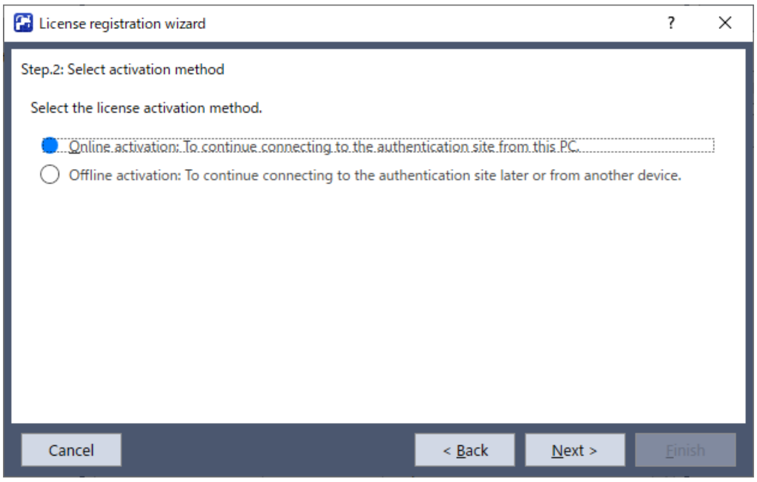
24.3.1 Enabling an option license (online authentication)

If your PC has Epson RC+ 8.0 installed, is connected to a Controller, and can connect to the Internet, use this authentication method.

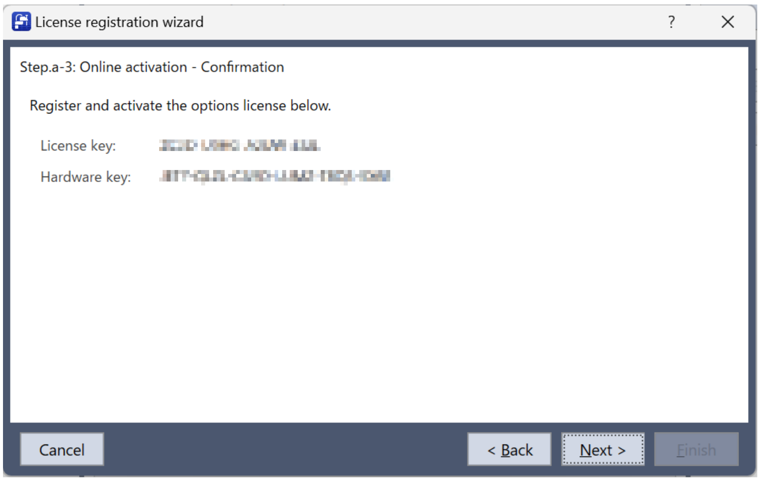
1. Purchase an option-enabling license from the supplier of your region.
2. Connect to the Controller for which you want to enable the option.
3. From the [License Configuration] window, select [Controller]-[Options].
4. Click the [Enable] button.
5. Enter the license key in the text box. If you acquired the license certificate as a file, click the [Load] button to specify the file.



6. Select [Online Authentication] and click [Next].



7. Check the displayed license key and click [Next].



8. Check the results window and click [Finish].

9. Close the [License Configuration] window.

   The controller will be automatically restarted.

24.3.2 Enabling an option license (offline authentication for one license at a time)

If your PC has Epson RC+ 8.0 installed, is connected to a Controller, but cannot connect to the Internet, use this authentication method. This method authenticates one license at a time.

Use a PC that can connect to the Internet. In addition, there must be a means of transferring files between two PCs.

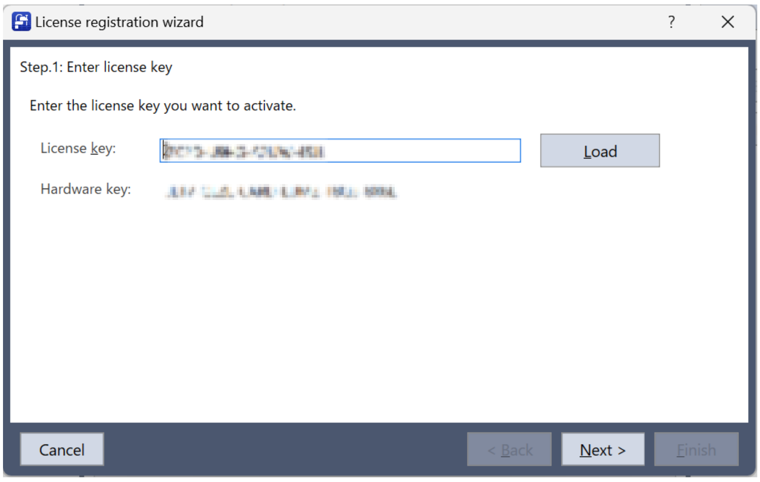
1. Purchase an option-enabling license from the supplier of your region.

2. Connect to the Controller for which you want to enable the option.

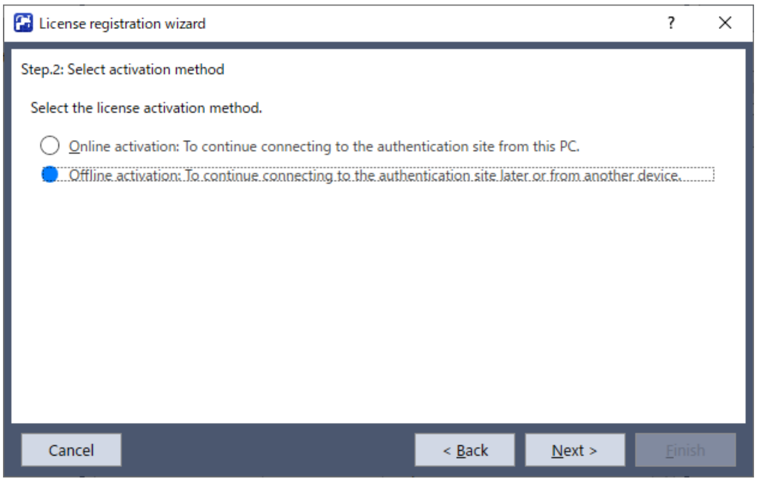
3. From the [License Configuration] window, select [Controller]-[Options].

4. Click the [Enable] button.

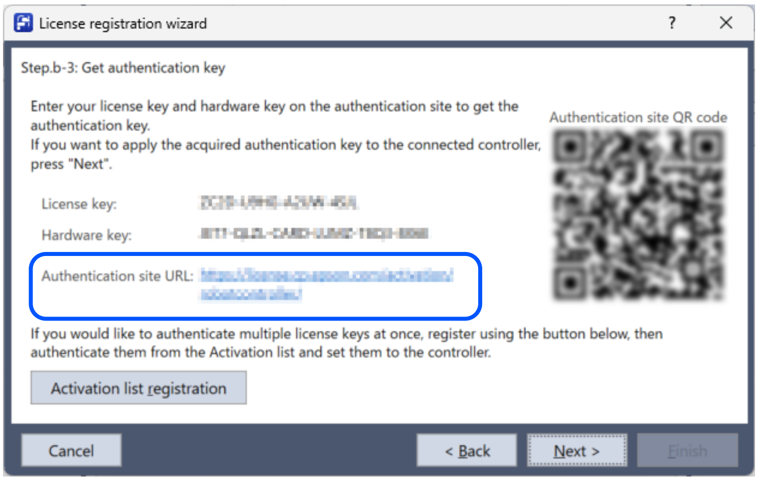
5. Enter the license key in the text box. If you acquired the license certificate as a file, click the [Load] button to specify the file.



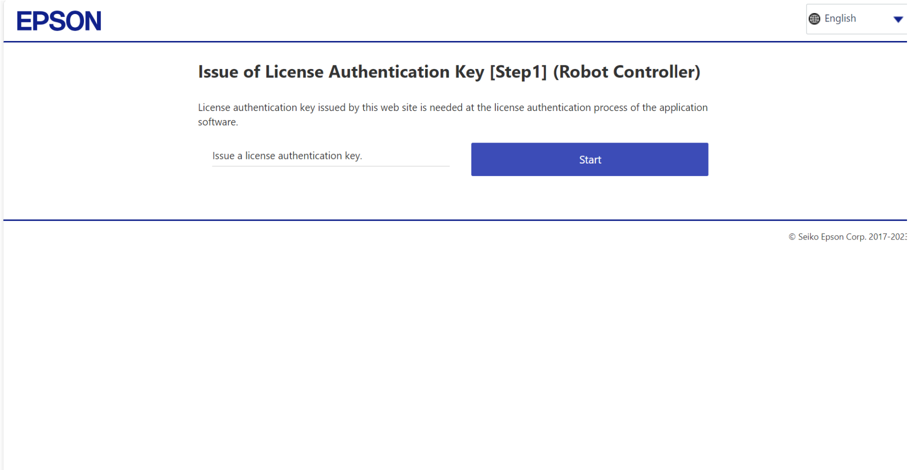
6. Select [Offline Authentication] and click [Next].



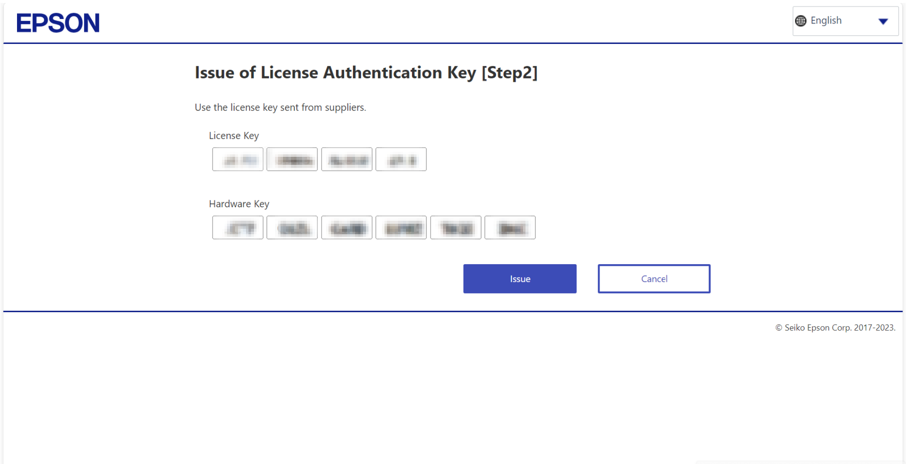
7. Access the displayed URL, using a browser on a PC with Internet access.



8. Click [Start] on the Web page that appears.

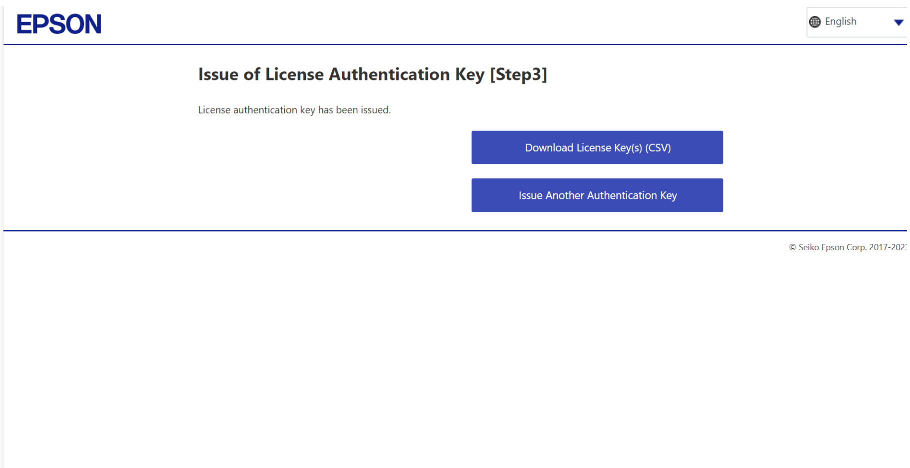


9. Enter the license key and hardware key shown in the window in step 7, and click [Issue].



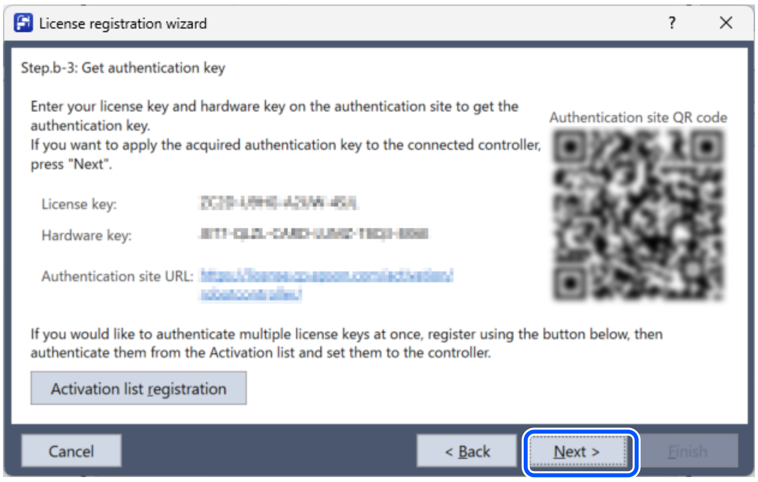
10. Click [Output in CSV] to save the authentication key file.

> **KEY POINTS**
>
> If you click [Continue Authentication] without saving the authentication key file, you will need to enter the key again.

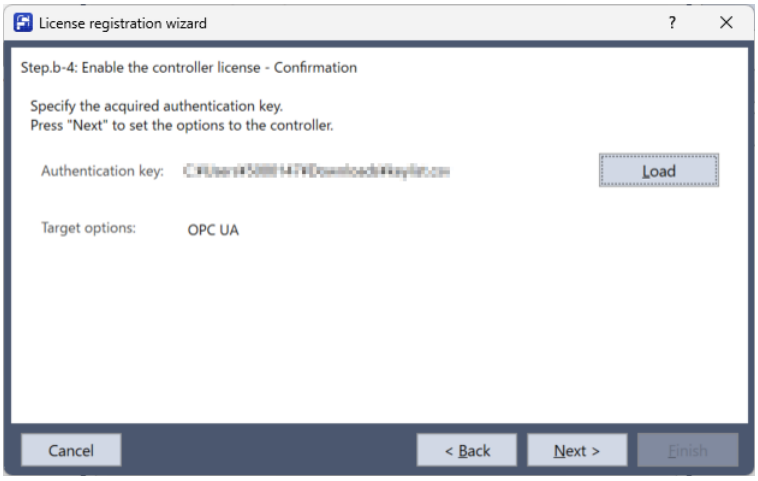


11. Using a USB flash drive or similar device, transfer the authentication key file to the PC connected to the Controller.

12. From the window displayed by the PC connected to the Controller, click [Next].



13. Click the [Load] button and specify the transferred authentication key file. Check the displayed option and click [Next].



14. Check the results window and click [Finish].

15. Close the [License Configuration] window.

    The controller will be automatically restarted.

24.3.3 Enabling option licenses (offline authentication of multiple licenses together)

If your PC has Epson RC+ 8.0 installed, is connected to a Controller, but cannot connect to the Internet, use this authentication method. This authentication method uses an authentication list file to collectively authenticate multiple licenses.

Prepare a PC with Internet access and install Epson RC+ 8.0. In addition, there must be a means of transferring files between two PCs.

Select [Authentication List] from the [License Configuration] window to display the following window showing the authentication status of the license.

Item	Description
License key	Displays license keys registered in the authentication list.
Serial #	Displays the serial number of the license-enabling Controller.
Status	Displays the authentication status of the license key.
Output Authentication List	Outputs the authentication list to a file.
Load Authentication List	Loads the authentication list from a file.
Execute Authentication	Registers the license key, which is registered in the list, at the authentication site.
Enable Execution	Registers the license key, which is authenticated at the authentication site, to the Controller and enables it.
Remove	Deletes an item from the authentication list.

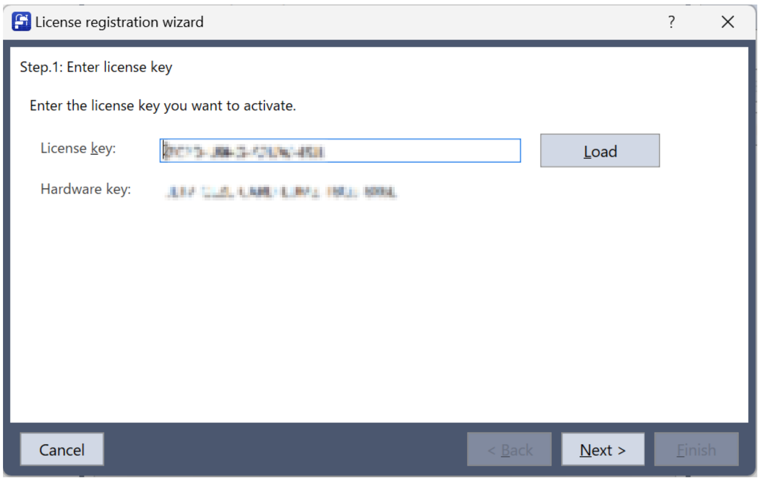
Purchasing a license

Purchase an option-enabling license from the supplier of your region.

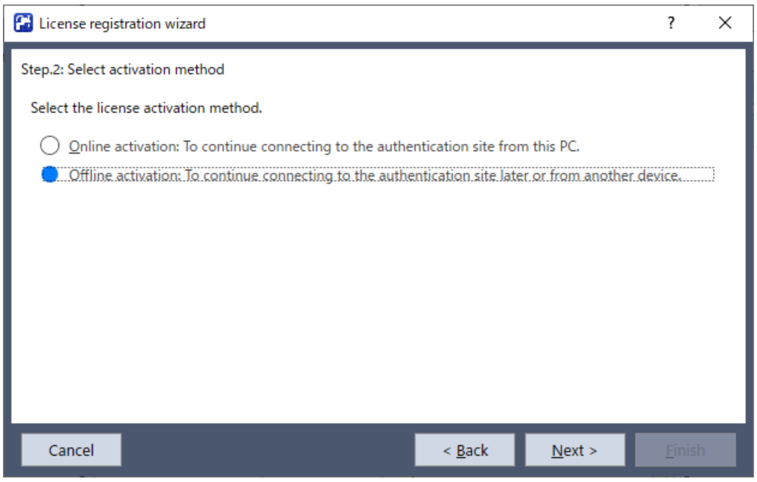
Creating an authentication list file

Perform the following steps from Epson RC+ 8.0 on a PC connected to the Controller.

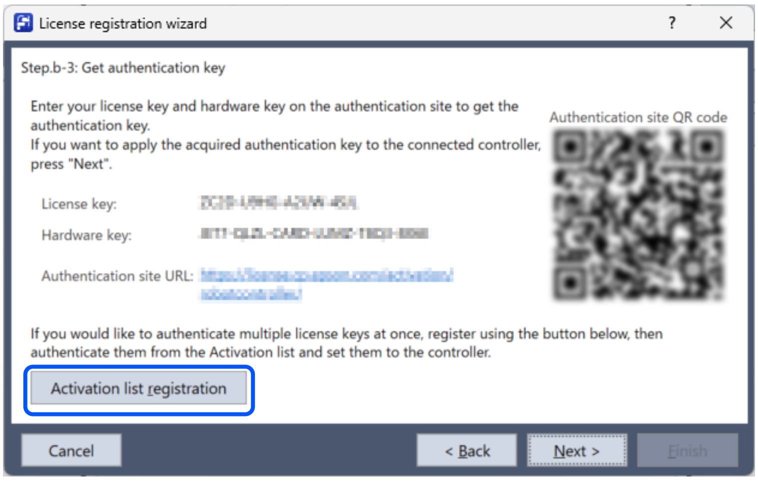
1. Connect to the Controller for which you want to enable the option.
2. From the [License Configuration] window, select [Controller]-[Options].
3. Click the [Enable] button.
4. Enter the license key in the text box. If you acquired the license certificate as a file, click the [Load] button to specify the file.



5. Select [Offline Authentication] and click [Next].



6. Click [Register Authentication List].



7. Check the results window and click [Finish].

8. From the [License Configuration] window, select [Authentication List].

9. Check the list of license keys and serial numbers displayed.

10. Click [Output Authentication List] to save the authentication list file.

Registration to authentication site

Perform the following steps from Epson RC+ 8.0 on a PC with Internet access:

1. Using a USB flash drive or similar device, transfer the authentication list file saved in "Creating an authentication list" to a PC with Internet access.

2. From the [License Configuration] window, select [Authentication List].

3. Click [Load Authentication List] to load the authentication list file.

4. Check the list of license keys and serial numbers displayed and click [Execute Authentication].

5. Confirm that the authentication status has been updated.

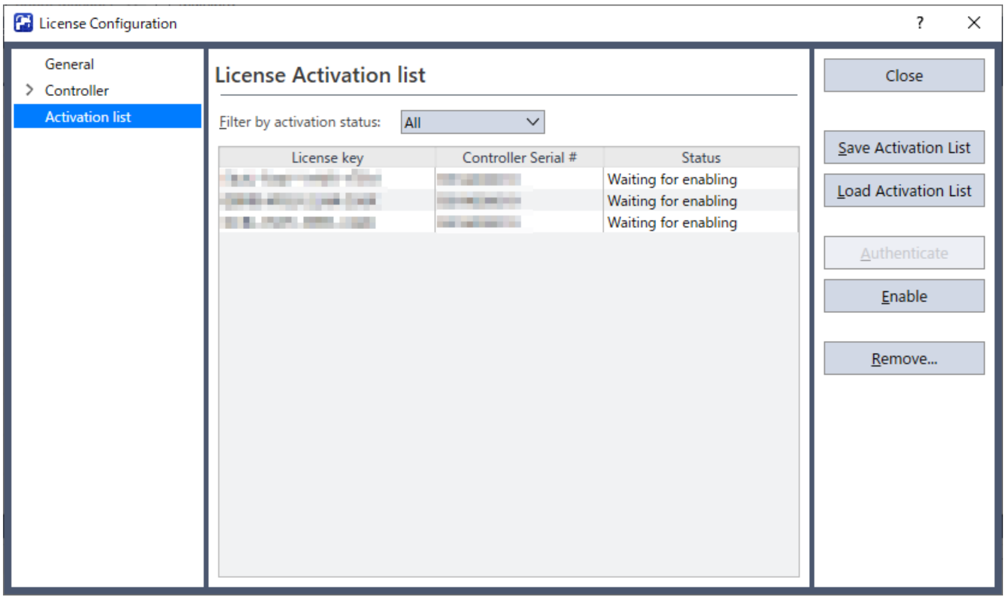
6. Click [Output Authentication List] to save the list to file.

   After saving, you can clear the list by clicking [Remove].

Enabling to Controller

Perform the following steps from Epson RC+ 8.0 on the PC connected to the Controller:

1. Using a USB flash drive or similar device, transfer the authentication list file saved in "Registration to authentication site" to the PC connected to the Controller.
2. From the [License Configuration] window, select [Authentication List].



3. Click [Load Authentication List] to load the authentication list file.
4. Check the list of license keys and serial numbers displayed and click [Enable Execution].
5. Connect to the Controller for which you want to enable the option.
6. Select the license key you want to enable. Selecting [All] will collectively enable the license keys corresponding to the currently connected Controllers.
7. Check the option name displayed and click [Next].
8. Check the results window and click [Finish].
9. Close the [License Configuration] window.

   The controller will be automatically restarted.

24.3.4 If the MAIN Board or SD card is replaced

If the MAIN board or SD card is replaced due to malfunction, all licenses configured on the controller will be disabled. Reconfigure by following the "option license configuration" procedures for each license.

* With the RC800 series, a previously acquired license key can continue to be used when the MAIN board or SD card is replaced.

25. Appendix

25.1 Appendix A: Software License Agreement

Information regarding the software license agreement can be found at the following folder before installation.

Epson_RC+****\EULA_OSSLicenses (****: RC+ version)

25.2 Appendix B: Epson RC+ 8.0 Software

Epson RC+ 8.0 can be used in the following operating systems.

- Windows 10 64-bit version (version1607 or later)
- Windows 11 64-bit version

(except Windows 10 (S mode), Windows 10 IoT Core, Windows 11 SE)

25.2.1 Epson RC+ 8.0 Software Installation

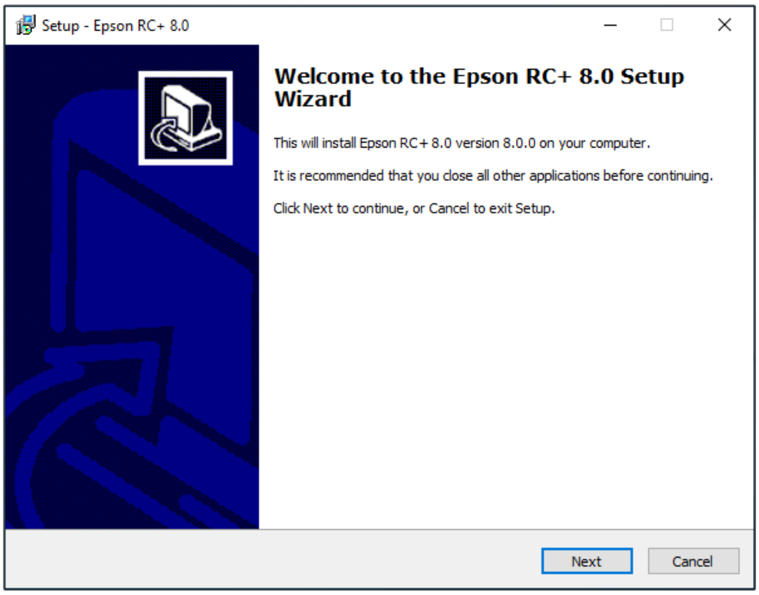
The Epson RC+ 8.0 software needs to be installed on your development PC.

First, make sure your PC is connected to the Internet.

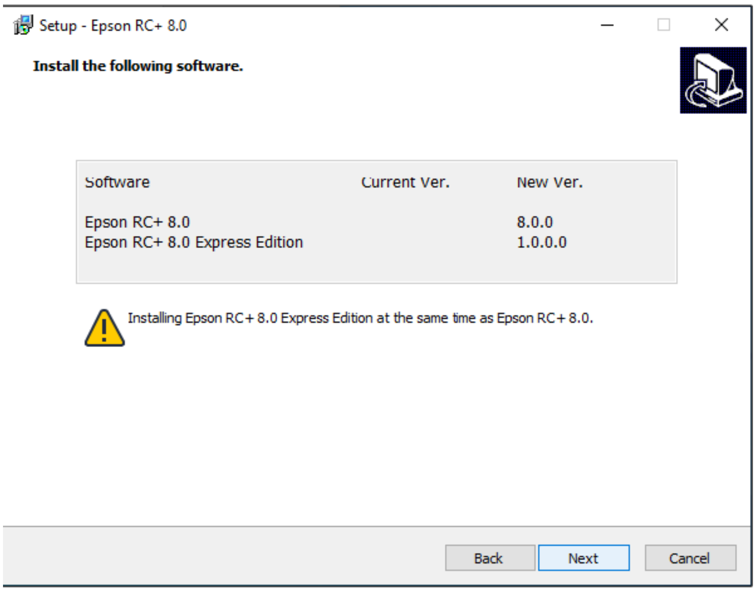
1. Insert the software disk attached to the product into your PC. Run EpsonRobotSoftwareInstallerSetup.exe on the disk and follow the on-screen instructions to begin installation.

> **KEY POINTS**
>
> - For installation instructions for Epson Robot Software Installer, refer to the Epson Robot Software Installer manual on your disk.
>
>   If you save and install the Epson RC+ setup file from the Internet, the latest software and manuals will be installed.
> - During installation, it will be displayed in the language according to your settings in [Control Panel]-[Clock and Region]-[Region]-[Formats].
>
>   However, depending on the country you selected at the Region setting, it may be displayed in English.

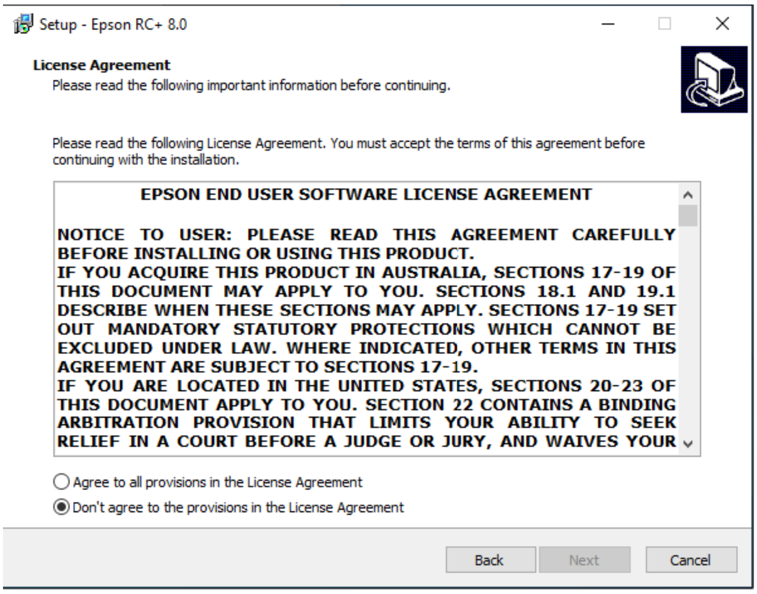
2. Check the RC+ version you want to install and click the [Next] button.



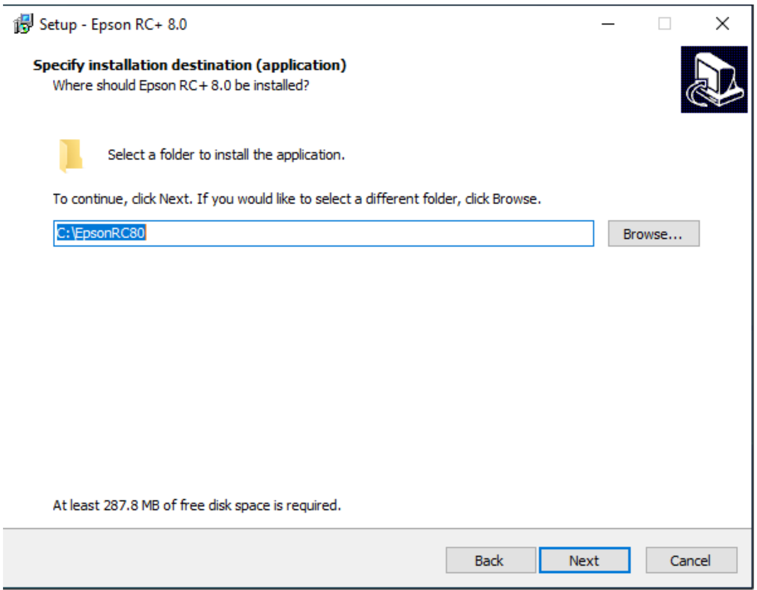
3. The window shown below appears. Check the content and click the [Next] button.



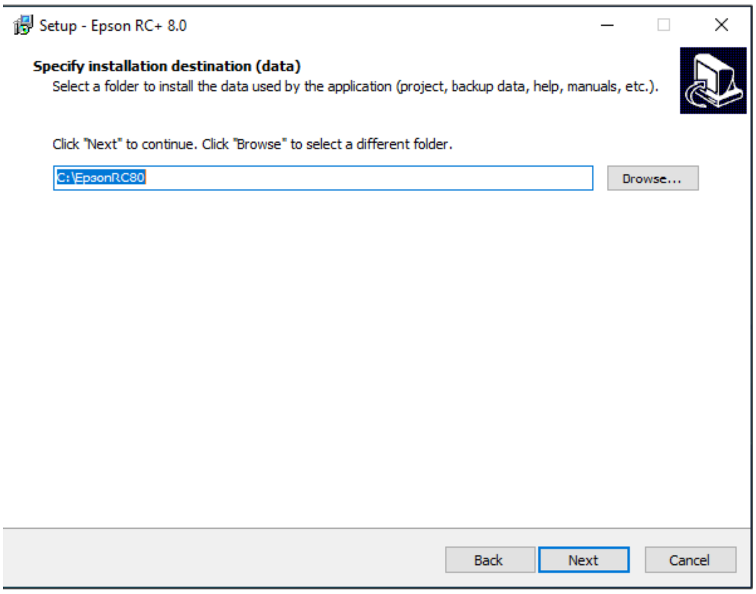
4. After reviewing the software license agreement, select “I accept all terms of the license agreement” and click the [Next] button.



5. Specify where to install the Epson RC+ 8.0 software. If necessary, click the [Browse] button to change the installation location.

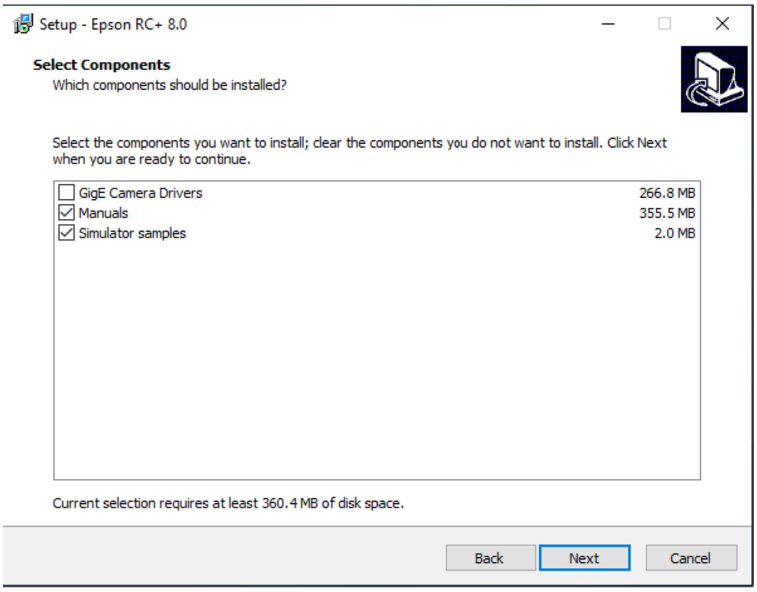


6. Specify where to install the project and other data. If necessary, click the [Browse] button to change the installation location.

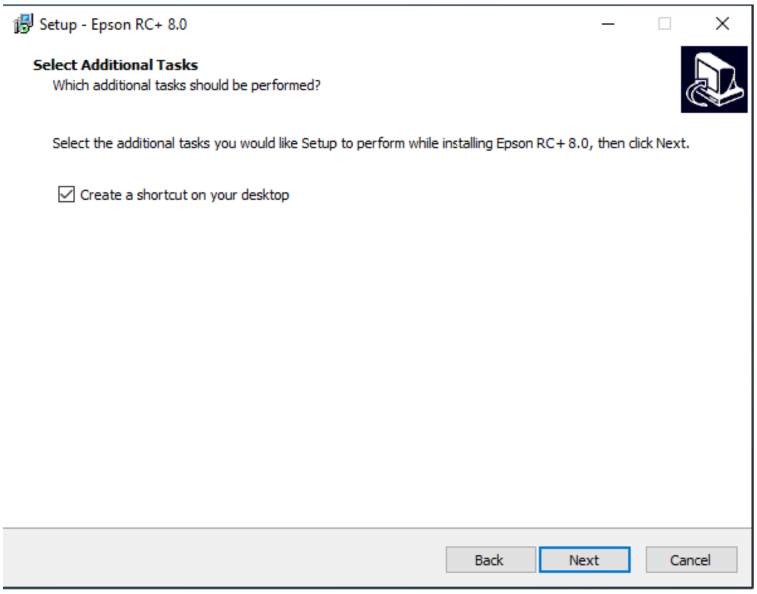


7. The dialog for selecting the options to be installed will be displayed.

   Check the options you want to install and click the [Next] button.

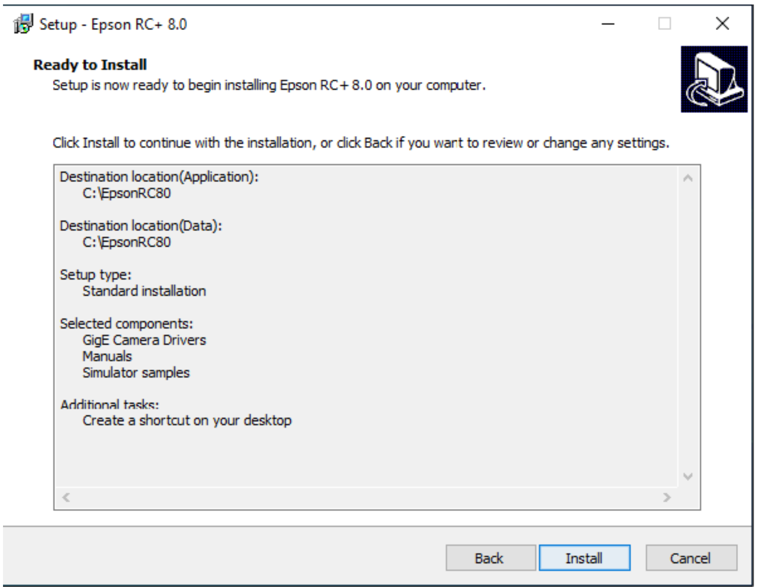


8. Select the tasks to execute and click the [Next] button.

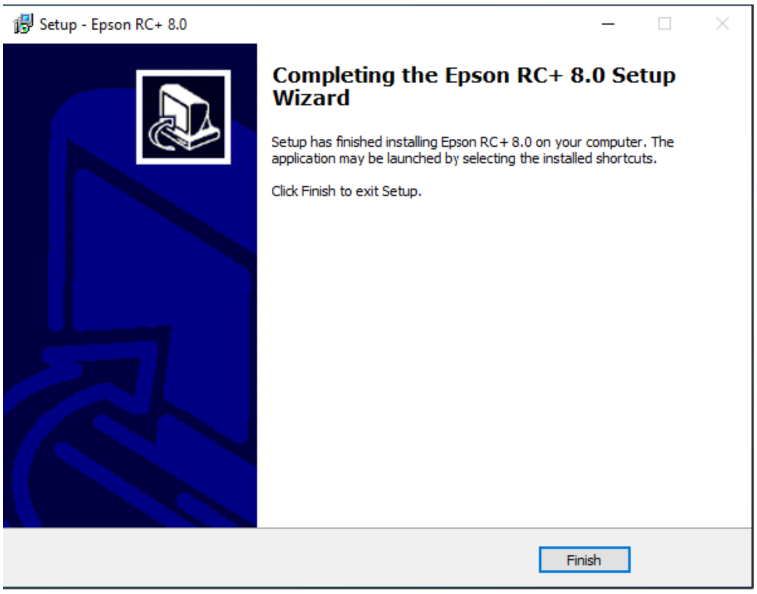


9. The dialog to review the settings will be displayed.

   If you are satisfied with the settings, click the [Next] button.



10. After the installation is completed, the following screen will be displayed.



KEY POINTS

The system may restart.

25.2.2 Epson RC+ 8.0 SoftwareUpdate

KEY POINTS

Make sure that Epson RC + 8.0 is updated by a user with Administrator right.

Save the latest version of Epson RC+ from Epson Robot Software Installer and update according to the menu.

For software downloads, refer to the following manual.

"Epson Robot Software Installer"

25.3 Appendix C: Simulator Functions List of Unsupported Manipulator Models

Simulator functions do not support the following Manipulator models.

X5 series Manipulators

All models do not support the simulator functions.

There is no alternative model.

G6 series Manipulators

Protected-models as described in the table below do not support the simulator functions.

G6-451D-II	G6-551D-II	G6-651D-II
G6-451DR-II	G6-551DR-II	G6-651DR-II

G6-451DW-II	G6-551DW-II	G6-651DW-II
G6-451P-II	G6-551P-II	G6-651P-II
G6-451PR-II	G6-551PR-II	G6-651PR-II
G6-451PW-II	G6-551PW-II	G6-651PW-II
G6-453D-II	G6-553D-II	G6-653D-II
G6-453DR-II	G6-553DR-II	G6-653DR-II
G6-453DW-II	G6-553DW-II	G6-653DW-II
G6-453P-II	G6-553P-II	G6-653P-II
G6-453PR-II	G6-553PR-II	G6-653PR-II
G6-453PW-II	G6-553PW-II	G6-653PW-II

When connecting to Virtual Controller, the following models can be set as an alternative model. However, external dimensions and motion range of arms may differ.

- Standard model: G6-***S*-II
- Cleanroom model: G6-***C*-II

For more details, refer to the following manual.

"SCARA Robot G series Manipulator Manual"

G10 series Manipulators

Protected-models as described in the table below do not support the simulator functions.

G10-651D-II	G10-851D-II
G10-651DR-II	G10-851DR-II
G10-651DW-II	G10-851DW-II
G10-651P-II	G10-851P-II
G10-651PR-II	G10-851PR-II
G10-651PW-II	G10-851PW-II
G10-654D-II	G10-854D-II
G10-654DR-II	G10-854DR-II
G10-654DW-II	G10-854DW-II
G10-654P-II	G10-854P-II
G10-654PR-II	G10-854PR-II
G10-654PW-II	G10-854PW-II

When connecting to Virtual Controller, the following models can be set as an alternative model. However, external dimensions and motion range of arms may differ.

- Standard model: G10-***S*-II
- Cleanroom model: G10-***C*-II

For more details, refer to the following manual.

"SCARA Robot G series Manipulator Manual"

G20 series Manipulators

Protected-models as described in the table below do not support the simulator functions.

G20-851D-II	G20-A01D-II
G20-851DR-II	G20-A01DR-II
G20-851DW-II	G20-A01DW-II
G20-851P-II	G20-A01P-II
G20-851PR-II	G20-A01PR-II
G20-851PW-II	G20-A01PW-II
G20-854D-II	G20-A04D-II
G20-854DR-II	G20-A04DR-II
G20-854DW-II	G20-A04DW-II
G20-854P-II	G20-A04P-II
G20-854PR-II	G20-A04PR-II
G20-854PW-II	G20-A04PW-II

When connecting to Virtual Controller, the following models can be set as an alternative model. However, external dimensions and motion range of arms may differ.

- Standard model: G20-***S*-II
- Cleanroom model: G20-***C*-II

For more details, refer to the following manual.

"SCARA Robot G series Manipulator Manual"