

EPSON

EPSON RC+

Ver.4.2

User's Guide

Rev.2

EM086S1714F

EPSON RC+ Ver.4.2

User's Guide

Rev.2

Copyright © 2007-2008 SEIKO EPSON CORPORATION. All rights reserved.

FOREWORD

Thank you for purchasing our robot products. This manual contains the information necessary for the correct use of the EPSON RC+ software.

Please carefully read this manual and other related manuals when using this software.

Keep this manual in a handy location for easy access at all times.

WARRANTY

The robot and its optional parts are shipped to our customers only after being subjected to the strictest quality controls, tests and inspections to certify its compliance with our high performance standards.

Product malfunctions resulting from normal handling or operation will be repaired free of charge during the normal warranty period. (Please ask your Regional Sales Office for warranty period information.)

However, customers will be charged for repairs in the following cases (even if they occur during the warranty period):

1. Damage or malfunction caused by improper use which is not described in the manual, or careless use.
2. Malfunctions caused by customers' unauthorized disassembly.
3. Damage due to improper adjustments or unauthorized repair attempts.
4. Damage caused by natural disasters such as earthquake, flood, etc.

Warnings, Cautions, Usage:

1. If the robot or associated equipment is used outside of the usage conditions and product specifications described in the manuals, this warranty is void.
2. If you do not follow the WARNINGS and CAUTIONS in this manual, we cannot be responsible for any malfunction or accident, even if the result is injury or death.
3. We cannot foresee all possible dangers and consequences. Therefore, this manual cannot warn the user of all possible hazards.

TRADEMARKS

Microsoft, Windows, Windows logo, Visual Basic, and Visual C++ are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

Other brand and product names are trademarks or registered trademarks of the respective holders.

TRADEMARK NOTATION IN THIS MANUAL

Microsoft® Windows® 2000 Operating System

Microsoft® Windows® XP Operating System

Throughout this manual, Windows 2000 and Windows XP refer to the respective operating systems. In some cases, Windows refers generically to Windows 2000 and Windows XP

NOTICE

No part of this manual may be copied or reproduced without authorization.

The contents of this manual are subject to change without notice.

Please notify us if you should find any errors in this manual or if you have any comments regarding its contents.

INQUIRIES

Contact the following service center for robot repairs, inspections or adjustments.

If service center information is not indicated below, please contact the supplier office for your region.

Please prepare the following items before you contact us.

- Your controller model and its serial number
- Your manipulator model and its serial number
- Software and its version in your robot system
- A description of the problem

SERVICE CENTER



MANUFACTURER & SUPPLIER

Japan & Others

SEIKO EPSON CORPORATION

Suwa Minami Plant
Factory Automation Systems Dept.
1010 Fujimi, Fujimi-machi,
Suwa-gun, Nagano, 399-0295
JAPAN

TEL : +81-(0)266-61-1802

FAX : +81-(0)266-61-1846

SUPPLIERS

North & South America

EPSON AMERICA, INC.

Factory Automation/Robotics
18300 Central Avenue
Carson, CA 90746
USA

TEL : +1-562-290-5900

FAX : +1-562-290-5999

E-MAIL : info@robots.epson.com

Europe

EPSON DEUTSCHLAND GmbH

Factory Automation Division
Otto-Hahn-Str.4
D-40670 Meerbusch
Germany

TEL : +49-(0)-2159-538-1391

FAX : +49-(0)-2159-538-3170

E-MAIL : robot.infos@epson.de

SAFETY PRECAUTIONS

Installation of robots and robotic equipment should only be performed by qualified personnel in accordance with national and local codes. Please carefully read this manual and other related manuals when using this software.

Keep this manual in a handy location for easy access at all times.



 WARNING	■ This symbol indicates that a danger of possible serious injury or death exists if the associated instructions are not followed properly.
 CAUTION	■ This symbol indicates that a danger of possible harm to people or physical damage to equipment and facilities exists if the associated instructions are not followed properly.

Table of Contents

Introduction	1
Welcome to EPSON RC+	1
System Overview	2
RC420 Controller	2
RC520 Controller	2
Software	2
Hardware	3
System Block Diagram	3
Options	4
SeikoRCS Users	4
SPEL for Windows Users	4
Documentation	4
 Safety	 5
Overview	5
Definitions	5
Robot Power	5
Safeguard	6
Safeguard Position Recovery	7
Operation Modes	8
Start Mode	8
Changing Operation Mode	8
Emergency Stop	9
Teach Control Device	9
Safety-related Requirements	10
Installation and Design Precautions	11
Designing a Safe Robot System	11
Robot System Installation, Start-up, and Testing	13
Precautions regarding Robot Operation	14
General Precautions	14
Automatic Operation	14
Teaching Robot Points	15
Return to Automatic Operation	15
Program Verification	15
Troubleshooting	15
Maintenance	16
Backup of Projects and System	16
Environmental Requirements	17
Precautions for Transport and Storage	18
End User Instruction Manual	18
End User Training	18

Getting Started	19
Hardware Installation	19
Software Installation	19
Installing the software options key	20
Installation for offline development	20
PC Configuration	20
Windows Security Administration	21
Writing your first program	22
 Operation	 25
System Power Up Procedure	25
Starting EPSON RC+	25
Startup sequence	25
Start Up Configuration	26
Start Mode	26
Start Mode Dialog	26
Start Up Mode: Program / Debug	27
Start Up Mode: Production / Operator	27
Start Up Mode: Production / Remote	27
Start Up Mode: Production / OP	27
Auto Start	28
Windows Log In	28
Command Line Options	29
Running without video monitor, keyboard, mouse	31
Problems during Start up	31
The SPEL Driver Manager	32
System Shutdown Procedure	32
 The EPSON RC+ GUI	 33
GUI Overview	34
Project Explorer Pane	34
Status Pane	35
Status Bar	35
Online Help	35
Toolbars	36
To create a new tool bar	36
To add new commands to an existing tool bar	36
File Menu	37
New Command (File Menu)	37
Open Command (File Menu)	37
Save Command (File Menu)	38
Save As Command (File Menu)	38
Restore Command (File Menu)	38
Rename Command (File Menu)	38
Delete Command (File Menu)	39
Import Command (File Menu)	39
Print Command (File Menu)	40
Exit Command (File Menu)	42
Edit Menu	42
Undo Command (Edit Menu)	42
Cut Command (Edit Menu)	42
Copy Command (Edit Menu)	42
Paste Command (Edit Menu)	42
Find Command (Edit Menu)	43
Find Next Command (Edit Menu)	44
Replace Command (Edit Menu)	44
Select All Command (Edit Menu)	45

Indent Command (Edit Menu).....	45
Outdent Command (Edit Menu).....	45
Comment Block Command (Edit Menu).....	45
Uncomment Block Command (Edit Menu).....	45
Go To Definition Command (Edit Menu)	46
Renumber Command (Edit Menu)	46
Strip Line Numbers Command (Edit Menu)	46
Project Menu.....	47
New Command (Project Menu).....	47
Open Command (Project Menu).....	48
Edit Command (Project Menu).....	49
Robot Parameters Command (Project Menu).....	52
Save Command (Project Menu)	64
Save As Command (Project Menu).....	64
Rename Command (Project Menu)	65
Copy Command (Project Menu).....	66
Delete Command (Project Menu).....	67
Compile File Command (Project Menu)	67
Build Command (Project Menu)	68
Rebuild Command (Project Menu).....	68
Run Menu	69
Start Command (Run Menu)	69
Operator Window Command (Run Menu).....	69
Control Device Command (Run Menu)	70
Step Into Command (Run Menu)	70
Step Over Command (Run Menu).....	70
Walk Command (Run Menu)	71
Resume Command (Run Menu)	71
Abort All Command (Run Menu)	71
Toggle Breakpoint Command (Run Menu).....	71
Clear All Breakpoints Command (Run Menu).....	72
Display Variables Command (Run Menu)	72
Instant Watch Command (Run Menu).....	72
Call Stack Command (Run Menu)	73
Tools Menu.....	74
Robot Control Panel Command (Tools Menu).....	75
Jog and Teach Command (Tools Menu)	76
Task Manager Command (Tools Menu).....	81
I/O Label Editor Command (Tools Menu)	83
I/O Monitor Command (Tools Menu).....	85
Monitor Window Command (Tools Menu).....	86
Macros Command (Tools Menu)	88
Maintenance Command (Tools Menu).....	89
Force Monitor Command (Tools Menu)	90
Audit Viewer Command (Tools Menu).....	91
User Error Editor Command (Tools Menu).....	91
Vision Command (Tools Menu).....	92
Conveyor Tracking Command (Tools Menu)	92
Setup Menu.....	93
System Configuration Command (Setup Menu)	93
Preferences Command (Setup Menu).....	95
Options Command (Setup Menu).....	102
Window Menu	103
Cascade Command (Window Menu).....	103
Tile Horizontal Command (Window Menu)	103
Tile Vertical Command (Window Menu).....	103
Arrange Icons Command (Window Menu)	103
Close All Command (Window Menu).....	103
1, 2, 3 Command (Window Menu).....	103

Help Menu	104
Contents Command (Help Menu)	104
Search for Help On Command (Help Menu)	105
About EPSON RC+ Command (Help Menu)	106
The SPEL⁺ Language	107
Overview	108
Program structure	108
What is a SPEL ⁺ program?	108
Calling functions	108
Line Numbers	109
Commands and statements	109
Function and variable names	109
Data types	110
Operators	110
Working with variables	111
Variable scopes	111
Local variables	111
Module variables	111
Global variables	112
Global Preserve variables	112
Arrays	113
Initial values	113
Clearing variables	113
Working with strings	114
Working with files	115
Multi-statements	116
Labels	116
Comments	116
Error handling	117
Multi-tasking	118
Chaining	119
Using multiple robots	120
Robot coordinate systems	122
Overview	122
Robot Coordinate System of Cartesian Robot	122
Robot Coordinate System of SCARA Robot	124
Robot Coordinate Systems for 6-Axis Robots	125
Local Coordinate Systems	126
Tool Coordinate Systems	127
ECP Coordinate Systems (Option)	128
Robot arm orientations	129
SCARA robot arm orientations	129
6-Axis robot arm orientations	130
Robot motion commands	132
Homing the robot	132
Point to point motion	132
Linear motion	132
Joint motion	133
Controlling position accuracy	133
Curves	133
CP Motion Speed / Acceleration and Tool Orientation	134
PTP Speed / Acceleration for Small Distances	134
Working with robot points	135
Defining points	135
Referencing points by name	135
Referencing points with variables	135
Using points in a program	136
Importing points in a program	136

Saving and loading Points	136
Point attributes.....	136
Extracting and setting point coordinates.....	138
Alteration of points.....	138
Input and output control.....	139
Hardware I/O	139
Memory I/O	139
I/O Commands	139
Using Traps.....	140
Handling Emergency Stop	141
Maintenance commands	141
Predefined constants	142
Calling Functions in Dynamic Link Libraries	144
Creating a DLL	145
Step 1: Decide on functions and parameters.....	145
Step 2: Create a DLL project.....	145
Step 3: Call the DLL function from SPEL ⁺	149
Step 4: DLL Debug Test Cycle	149
More information	149

Building SPEL⁺ Applications 151

Designing Applications	151
Creating the simplest application	151
Application Layout.....	151
Auto start at power up	152
Managing Projects	152
Overview	152
Creating a new project.....	154
Configuring a project.....	154
Building a project.....	156
Making copies of projects	157
Backing up a project.....	157
Editing Programs	157
Program rules	157
Typing in program code	157
Syntax Help	158
Syntax Errors.....	159
Renumbering	159
Removing line numbers.....	159
Editing Points	160
Running and Debugging Programs.....	162
The Run Window	162
Debugging	164
The Operator Window	167
Operator Window Configuration.....	168
Configuring start up for Operator Window	168
Using Remote Control	168

Motion Systems 169

RC420 Motion System	169
RC420 Drive Module Software Configuration.....	169
Saving drive module configuration	170
Loading drive module configuration	171
RC520 Motion System	171
RC520 Drive Unit Software Configuration.....	171
Saving drive unit configuration	173
Loading drive unit configuration	173
PG Motion System.....	174

Robot Configuration	175
Adding a Robot	175
Adding Standard Robots	175
Adding Pulse Generator Robots	176
Deleting a Robot	176
Changing Robot Parameters	177
Saving a robot configuration	178
Loading a robot configuration	178
Moving a robot between systems	178
I/O Systems.....	179
Standard Digital I/O	179
Overview	179
Input Response Time	179
Installing ISA I/O Boards	180
ISA I/O Board Software Configuration	180
Ethernet I/O.....	183
Overview	183
Ethernet I/O Hardware Installation.....	184
Assigning rack IP addresses	185
Ethernet I/O Software Configuration.....	185
Ethernet I/O Addressing in SPEL ⁺	187
SPEL ⁺ Ethernet I/O Commands	188
Using Standard Counters	188
Using Quadrature Encoder Counters	189
Using Quadrature Encoder Counters with Index	189
Fieldbus I/O	190
Remote Control	191
Overview.....	191
Auto Cycle Remote Control.....	191
Teach Mode with Remote Control.....	192
Debugging Remote Control	192
Remote Input Handshake Timing	193
Selecting Program Groups	193
Selecting Robots	193
Remote Inputs	194
Remote Outputs	195
Remote Safeguard Position Recover.....	196
Remote Auto Recover	196
Remote Manual Recover	196
Operation of RecoverReqd and RecoverInCycle Outputs	196
Remote Control Software Configuration	197
RS-232 Communications	199
RS-232 Software Configuration.....	199
RS-232 Commands	200
TCP / IP Communications.....	201
TCP/IP Setup	201
Ethernet Hardware.....	201
IP Addresses	201
Configuring Windows TCP/IP	202
LMHOSTS File Configuration	204
Testing Windows TCP/IP setup.....	204
TCP/IP Software Configuration.....	205

TCP/IP Commands	206
TCP/IP Example	206
SPEL⁺ Options	207
Force Sensing	211
Overview	211
Specifications.....	212
Installation	212
Installing the force transducer circuit board	212
Mounting the force transducer.....	213
Mounting tooling to force sensor.....	215
Connecting the force transducer	216
Routing the transducer cable	216
Software Configuration	216
Force Sensing Commands	218
Using the Force Sensing Trigger	219
Stopping Motion along Z axis	219
Stopping Motion along X or Y axes.....	219
Using Torque Limiting	220
Security.....	221
Overview	221
Installation	221
Security Configuration.....	221
General Tab	222
Users Tab.....	223
Groups Tab.....	224
Group Rights	224
Security Audit.....	225
SPEL ⁺ Security Commands	225
Conveyor Tracking	227
Overview	227
Conveyor Tracking Processes.....	228
Tracking conveyor system.....	228
Indexed Conveyor System.....	228
Hardware Installation.....	229
PG Board Specifications.....	229
DIP Switch Settings.....	231
Jumper Settings	231
Signal Connections.....	231
Encoder Input Circuit	236
Conveyor encoder configuration	237
Verifying new encoder operation	237
Conveyor Tracking Commands	238
Creating conveyors in a project	239
Configure conveyors.....	240
Vision conveyors	241
Vision conveyor camera and lighting.....	241
Vision calibration sequence.....	241
Vision conveyor calibration.....	241
Vision conveyor programming.....	247
Sensor conveyors	249
Sensor conveyor calibration	249
Sensor conveyor programming.....	253

Multiple conveyors and robots.....	254
To pick up the same parts with multiple robots.....	254
To pick up different part types with multiple robots	255
Abort tracking.....	255
ECP Motion	257
Overview.....	257
How to move the arm with ECP motion.....	258
Installing EPSON RC+ Options	261
Software License Agreement	263
Appendix A: PC Configuration.....	267
Configure the PC Control Unit	267
Installing EPSON RC+ Software.....	268
Adding / Removing Options.....	269
Reserving Hardware Resources	270
Turn off automatic restart after system failure.....	273
Installing LAN Driver.....	275
Updating Graphic Driver and Setting Graphic Accelerator	279
Setting Time and Date	284
Setting Screen Resolution	286
Installing Optional Device Port (COM3).....	288
Installing RC420 UPS Drivers.....	292
Adding and Removing Memory	298
Precaution for Epson ProSix PS Series robots	298
Settings at Shipment	299
Appendix B: Changes from previous SPEL versions	301
Changes from SPEL to SPEL ⁺	301
New Commands in SPEL ⁺	304
Obsolete commands	307
SRC-3xx Error Codes	309
Appendix C: Converting SPEL for Windows projects.....	311
Appendix D: Ethernet I/O Hardware	313
Index.....	315

Introduction

Welcome to EPSON RC+

Welcome to the EPSON RC+ Robot Control System. EPSON RC+ is a robotic cell control system that utilizes the open architecture of the PC and Microsoft Windows.

EPSON RC+ features:

- PC based control using Microsoft Windows.
- Integrated application development environment.
- SPEL⁺. A powerful, easy to use BASIC-like programming language that supports multi-tasking, robot motion control, I/O control, and networking.
- Multiple robots on one system.
- I/O systems including Digital I/O boards, Ethernet I/O, and Fieldbus I/O.
- TCP/IP and RS-232 communications.
- Vision Guide. Integrated vision robot guidance.
- VB Guide. Enables you to control the system using standard programming environments including Microsoft Visual Basic and Microsoft Visual C++.
- Force Sensing option allows the robot to use torque limiting and torque/force sensing and measurement.
- 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 or indexed conveyors using vision or sensors.
- PG Motion System option allows you to use third party motors and drives to control auxiliary equipment such as XY tables, slides, etc.
- ECP option supports CP motion relative to a fixed point.

System Overview

The EPSON RC+ Robot Control System contains several standard and optional components that enable you to control a robotic work cell. There are two controller types supported by EPSON RC+: RC420 and RC520.

RC420 Controller

The RC420 Controller combines the PC and motion drive system in one compact box. It can control one 4 axis robot.

The RC420 standard system contains the following items:

- PC based controller running Windows, the EPSON RC+ software, and SPEL+ drivers.
- Built in motion drive system. The motion drive system can control up to four motors, or typically one robot.
- Optional I/O expansion boards.

RC520 Controller

The RC520 Controller is an industrial passive backplane PC combined with up to 3 drive units.

The RC520 standard system contains the following items:

- PC running Windows which runs the EPSON RC+ software and SPEL+ drivers.
- One Motion Interface Board (MIB) installed in the PC for each drive unit.
- One to three RC520 Motion Drive Units. Each drive unit can control up to four motors, or typically one robot. Also, each drive unit has 16 DC inputs and 16 DC outputs.
- Optional I/O expansion boards for the PC.

EPSON RC+ can be used in several configurations. Each configuration uses a PC as the core controller, with EPSON RC+ software.

- **Robot Control System.** In this configuration, one or more motion drive units are used to drive one or more robots.
- **Stand Alone Control System.** In this configuration, the PC is used as a cell controller without any robots. An special standalone MIB board must be installed in the PC to run SPEL+ tasks.
- **Vision Control System.** In this configuration, the PC is used as a standalone vision controller.
- **Offline Development System.** EPSON RC+ can run on any PC running Windows XP without any additional hardware. In this mode, you can edit programs and build projects to check syntax on an offline computer.
- **Simulation.** EPSON RC+ can run programs without a robot and drive unit. An MIB board must be installed in the PC. This mode allows you to run and debug programs without a robot or drive unit.

Software

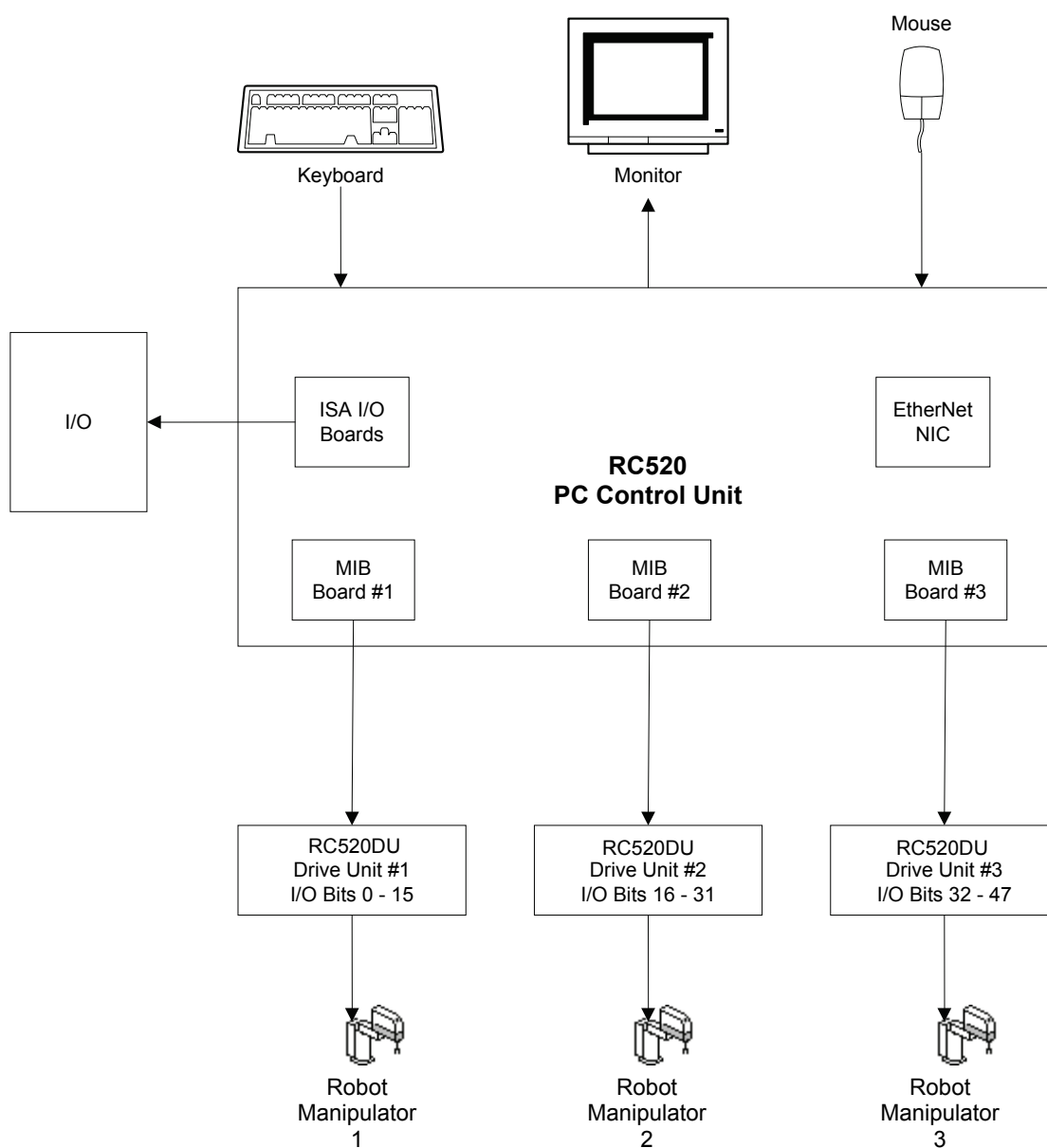
The system includes EPSON RC+ software, which comes pre-installed with Windows. The EPSON RC+ software package includes all required development, runtime, and option software on one CD. You can purchase options with the product or enable them in the field.

Hardware

Refer to the Controller manual for details on the controller hardware.

System Block Diagram

The EPSON RC+ system is based on an industrial Pentium III PC using a passive back plane to allow for expansion. The system can support up to three Drive Units, allowing you to control multiple robots from one PC. You can also configure the system to use multiple robots with one or two axes on a single drive unit, allowing for greater expansion.



Options

Several EPSON RC+ options are available. Some options require a software key. One key supports all options. Most options can be enabled in the field after purchasing from your distributor.

See the chapter *Installing Options* for details on how to enable an option.

SeikoRCS Users

If you have used SeikoRCS 3.x software, you will find that EPSON RC+ is very similar.

You can convert your SeikoRCS project by opening the project. EPSON RC+ will copy the entire project from the \SeikoRCS\Project directory to the \EpsonRC\Project directory. You can then use the project in the EPSON RC+ environment.

SPEL for Windows Users

If you have used SPEL for Windows 1.x or 2.x software, you will find that EPSON RC+ is very similar.

Many new commands have been added to the SPEL⁺ language, which replaces SPEL. Also, several commands were made obsolete or replaced. For details, see Appendix B: *Changes from previous versions*.

You can convert your SPEL for Windows project by opening the project. EPSON RC+ will copy the files to a new directory and optionally convert the programs. For details, see Appendix C: *Converting projects from previous versions*.

Documentation

All documentation is provided in hardcopy format and also on the PC in Acrobat PDF format. To view manuals on the PC, click Start - Programs - EPSON RC+ and click on the manual you want to view.

EPSON RC+ User's Guide

This manual contains detailed information for the entire system.

SPEL⁺ Language Reference Manual

This manual contains detailed information for the entire SPEL⁺ language.

Vision Guide Manuals

These manuals contain information for using the Vision Guide option.

VB Guide Manual

This manual contains information for using the VB Guide option.

Robot Manual

This manual contains detailed information for the robot(s) you have purchased. Each series of robots has its own manuals.

Safety

Overview

This chapter explains the important safety requirements for robotic systems using EPSON RC+.

Installation of robots and robotic equipment should only be performed by qualified personnel in accordance with national and local codes. Please read and understand this entire chapter before using your EPSON RC+ system.

Remember that safety is the most important consideration when designing and operating any robotic system.

Definitions

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 status. During operation, when the safeguard circuit opens, the system will switch to operation-prohibited status.

The robot will operate at low speed and torque in the restricted status (low power). In the unrestricted status (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 status (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 status of the robot will be set to either the restricted (low power) status or the operation-prohibited status. The system will not change to the unrestricted (high power) status if the appropriate procedures are not followed.

When the system is in restricted (low power) status or operation-prohibited status, 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.

Safeguard

To ensure safe operation of the robotic work cell, you must install a safety system using safety doors, light curtains, safety floor mats, etc.



WARNING

There is a safeguard input circuit in the EMERGENCY connector on the RC520 Drive Unit (or RC420 Controller) that connects with the safety device interlock switch. In order to protect those working with the robot be sure that the interlock switch is connected and working properly.

If a closed safeguard is open during robot motion, the robot stops immediately and enters pause status. All robot motors are turned off. This applies to all robots connected to the controller. The descriptions below explain how the safeguard input works.

Safeguard closed: The safeguard input is turned ON. The robot can automatically operate in unrestricted (high power) status.

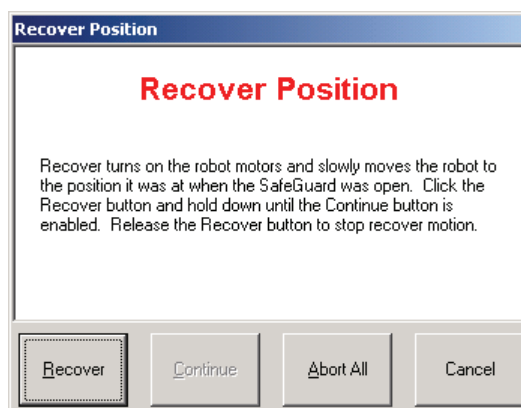
Safeguard open: The safeguard input is turned OFF, and the interlock function operates. The robot stops immediately, motors are turned off, and further operation is impossible until either the safeguard is closed or Teach mode is turned ON and the enable circuit is engaged.

For further details on the safeguard and interlock, refer to the section *Installation and Design Precautions* later in this chapter. For detailed wiring instructions, refer to the *Setup & Operation EMERGENCY Connector* in the controller manual.

Safeguard Position Recovery

During automatic operation, when the safeguard is open, the robot pauses and the robot motors are turned off. At this point, the robot joints can be moved by hand. After the safeguard is closed and Cont is executed, operation is determined by the SPEL Option *Auto recover position after safeguard* preference.

- Auto recover on** The motors will automatically turn back on, and the robot will move slowly to the position it was in when the safeguard was opened, then normal cycle will continue.
- Auto recover off** For the EPSON RC+ Run window and Operator window, the following dialog will be displayed with a Recover button when the operator clicks Cont.



The operator must press and hold the Recover button until the motors have turned on and the robot has moved back to the position it was in when the safeguard was opened. If the operator releases the Recover button during recover motion, the robot will stop. When the recover motion has completed, the operator can then click the Cont button to continue normal cycle.



It is strongly recommended that operators do not move the robot when the safeguard is open. If for some reason operators will move the robot by hand when the safeguard is open, ensure that the robot can safely move back to its original position without obstructions. When the robot moves to the position it was in when the safeguard was opened, the robot does not retrace the trajectory that was moved by hand. In this case, the SPEL Option *Auto recover position after safeguard* preference should be turned off.

Operation Modes

The operation mode is defined as the single control point for EPSON RC+, therefore you cannot control EPSON RC+ in more than one operation mode at the same time.

There are two basic types of operation modes: AUTO and TEACH. AUTO operation modes allow you to execute 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.

There are 4 AUTO operation modes:

- | | |
|-------------------------|---|
| (1) Program/Debug | This mode allows you to develop your projects. It is the default start mode. |
| (2) Production/Operator | This mode starts the system and displays the Operator Window. During startup, the operation mode can be changed using a password. |
| (3) Production/Remote | This mode starts the system for remote control. |
| (4) Production/OP | This mode starts the system for control from the OP500RC Operator Pendant. |

Start Mode

The Start mode specifies the AUTO operation mode for EPSON RC+ when it starts.

You can set one AUTO operation mode from the following modes (1) - (4).

- (1) Program/Debug
- (2) Production/Operator
- (3) Production/Remote
- (4) Production/OP

For information on how to change the start mode, refer to the *Operation* chapter.

Changing Operation Mode

You can change from any AUTO operation mode (1 - 4) to TEACH mode. To change the current AUTO operation mode to TEACH, set the TEACH / AUTO Switch to the TEACH position.

When the TEACH / AUTO Switch is changed back to AUTO, the operation mode is returned to the preceding AUTO operation mode (1 - 4).

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

The AUTO operation mode cannot be changed to any operation modes except the TEACH mode after the system has started. To change the operation mode, set the start mode again and restart EPSON RC+.

For more information, refer to the *Operation* chapter.

Emergency Stop

The RC520 Drive Unit and RC420 controller are 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.



CAUTION

The path that the robot will follow from the time the emergency stop switch is pressed until the device stops, as well as the stop position itself, cannot be positively determined. In many cases, the stop position will not exceed the target position for the operation prior the emergency stop. Depending on the robot's loading condition and operation speed, overruns are inevitable. Taking this into consideration be sure the layout for the peripheral equipment includes extra space.

In addition, the RC520 Drive Unit is equipped with an emergency stop output terminal that is interlocked with the robot's emergency stop system. This can be used to control peripheral equipment.

For detailed wiring instructions, refer to the section *EMERGENCY Connector* in the controller manual.

Teach Control Device

Operators can use a TEACH control device to operate the robot in the Teach operation mode.

Construct the TEACH control device using the following switches and connect it to the OPTIONAL DEVICE connector on the front panel of the controller.

- TEACH / AUTO mode switch
- Emergency stop switch
- 3 position enable switch (dead-man)

For further details on the internal circuit of TEACH control device, refer to *Setup & Operation TEACH Control* in the controller manual.

Safety-related Requirements

Specific tolerances and operating conditions for safety are contained in the manuals for the robot, controller and other devices. Be sure to read those manuals as well.

Robot systems safety standard and other examples are given in this chapter. Therefore, to ensure that safety measures are complete, please refer to the other standards listed as well.

(Note: The following is only a partial list of the necessary safety standards.)

EN775	European Standard; Manipulating industrial robots - Safety.
ANSI/RIA R15.06	American National Standard; Industrial Robots and Robot Systems - Safety Requirements.
IEC204-1 (EN60204-1)	Safety of machinery - Electrical equipment of machines Part 1. Specification for general requirements.
EN292-1,-2	Safety of machinery - Basic concepts, general principles for design Part 1. Basic terminology, methodology Part 2. Technical principles and specifications
EN418	Emergency stop equipment, functional aspects - principles for design.
EN953	General requirements for design and construction of guards.
EN55011	Limits and methods of measurement of radio disturbance characteristic of industrial scientific and medical (ISM) radio frequency equipment
EN61000-6-2	Electromagnetic compatibility (EMC) Part 6-2. Generic standards - Immunity for industrial environments
EN1050	Safety of machinery Principles for risk assessment
EN954-1	Safety of machinery-Safety related parts of control systems

Installation and Design Precautions

Designing a Safe Robot System

It is important to operate robots safely. It is also important for robot users to give careful consideration to the safety of the overall robot system design.

This section summarizes the minimum conditions that should be observed when using EPSON robots in your robot systems.

Please design and manufacture robot systems in accordance with the principles described in this and the following sections.

Environmental Conditions

Carefully observe the conditions for installing robots and robot systems that are listed in the “Environmental Conditions” tables included in the manuals for all equipment used in the system.

System Layout

When designing the layout for a robot system, carefully consider the possibility of error between robots and peripheral equipment. Emergency stops require particular attention, since a robot will stop after following a path that is different from its normal movement path. The layout design should provide enough margin for safety. Refer to the manuals for each robot, and ensure that the layout secures ample space for maintenance and inspection work.

When designing a robot system to restrict the area of motion of the robots, do so in accordance with the methods described in each robot manual. Utilize both software and mechanical stops as measures to restrict motion.

Install the emergency stop switch at a location near the operation unit for the robot system where the operator can easily press and hold it in an emergency.

Do not install the controller at a location where water or other liquids can leak inside the controller. In addition, never use liquids to clean the controller.

Disabling Power to the System using lock out / tag out

The power connection for the robot drive units should be such that it can be locked and tagged in the off position to prevent anyone from turning on power while someone else is in the safeguarded area. For further details, refer to the section *Procedure of Lockout/Tagout* in the chapter *Safety Precautions* in the controller manual.

End Effector Design

Provide wiring and piping that will prevent the robot end effector from releasing the object held (the work piece) when the robot system power is shut off.

Design the robot end effector such that its weight and moment of inertia do not exceed the allowable limits. Use of values that exceed the allowable limits can subject the robot to excessive loads. This will not only shorten the service life of the robot but can lead to unexpectedly dangerous situations due to additional external forces applied to the end effector and the work piece.

Design the size of the end effector with care, since the robot body and robot end effector can interfere with each other.

Peripheral Equipment Design

When designing equipment that removes and supplies parts and materials to the robot system, ensure that the design provides the operator with sufficient safety. If there is a need to remove and supply materials without stopping the robot, install a shuttle device or take other measures to ensure that the operator does not need to enter a potentially dangerous zone.

Ensure that an interruption to the power supply (power shutoff) of peripheral equipment does not lead to a dangerous situation. Take the measures that not only prevent a work piece held from

being released as mentioned in “End effector Design” but that also ensure peripheral equipment other than the robots can stop safely. Verify equipment safety to ensure that, when the power shuts off, the area is safe.

Remote Control

To prevent operation by remote control from being dangerous, start signals from the remote controller are allowed only when the control device (CtrlDev) is set to REMOTE, Teach mode is OFF, and the system is commanded to accept remote signals. For the safety of the overall system, however, safety measures are needed to eliminate the risks associated with the start-up and shutdown of peripheral equipment by remote control.

Emergency Stop

Each robot system needs equipment that will allow the operator to immediately stop the system's operation. Install an emergency stop device that utilizes emergency stop input from the controller and all other equipment.

During an emergency stop, the power that is supplied to the motor driving the robot is shut off, and the robot is stopped by dynamic braking.

The emergency stop circuit should also remove power from all external components that must be turned off during an emergency. Do not assume that the robot controller will turn off all outputs if configured to. For example, if an I/O card is faulty, the controller cannot turn off a component connected to an output. The emergency stop on the controller is hardwired to remove motor power from the robot, but not external power supplies.

Safeguard System

To ensure safety, a safeguard system should be installed for the robot system.

When installing the safeguard system, strictly observe the following points:

Refer to each robot manual, and install the safeguard system outside the maximum space. Carefully consider the size of the end effector and the work pieces to be held so that there will be no error between the moving parts and the safeguard system.

Manufacture the safeguard system to withstand calculated external forces (forces that will be added during operation and forces from the surrounding environment).

When designing the safeguard system, make sure that it is free of sharp corners and projections, and that the safeguard system itself is not a hazard.

Make sure that the safeguard system can only be removed by using a tool.

There are several types of safeguard devices, including safety doors, safety barriers, light curtains, safety gates, and safety floor mats. Install the interlocking function in the safeguard device. The safeguard interlock must be installed so that the safeguard interlock is forced to work in case of a device failure or other unexpected accident. For example, when using a door with a switch as the interlock, do not rely on the switch's own spring force to open the contact. The contact mechanism must open immediately in case of an accident.

Connect the interlock switch to the safeguard input of the drive unit's EMERGENCY connector. The safeguard input informs the robot controller that an operator may be inside the safeguard area. When the safeguard input is activated, the robot stops immediately and enters pause status, as well as either operation-prohibited status or restricted status (low power status).

Make sure not to enter the safeguarded area except through the point where the safeguard interlock is installed.

The safeguard interlock must be installed so that it can maintain a safe condition until the interlock is released on purpose once it initiates. The latch-release input is provided for the EMERGENCY connector on the Controller to release the latch condition of the safeguard interlock. The latch release switch of the safeguard interlock must be installed outside of the safeguarded area and wired to the latch-release input.

It is dangerous to allow someone else to release the safeguard interlock by mistake while the operator is working inside the safeguarded area. To protect the operator working inside the safeguarded area, take measures to lock out and tag out the latch-release switch.

Presence Sensing Device

The above mentioned safeguard interlock is a type of presence sensing device, since it indicates the possibility of somebody being inside the safeguard system. When separately installing a presence sensing device, however, perform a satisfactory risk assessment and pay thorough attention to its dependability.

Here are precautions that should be noted:

- Design the system so that when the presence sensing device is not activated or a dangerous situation still exists that no personnel can go inside the safeguard area or place their hands inside it.
- Design the presence sensing device so that regardless of the situation the system operates safely.
- If the robot stops operating when the presence sensing device is activated, it is necessary to ensure that it does not start again until the detected object has been removed. Make sure that the robot cannot automatically restart.

Resetting the Safeguard

Ensure that the robot system can only be restarted through careful operation from outside the safeguard system. The robot will never restart simply by resetting the safeguard interlock switch. Apply this concept to the interlock gates and presence sensing devices for the entire system.

Robot System Installation, Start-up, and Testing

Installation

When installing the robot and robot system, follow the instructions contained in each of the robot and robot system manuals.

Start-up and Functional Testing

If the safeguard system is not ready at the time of start-up and functional testing, specify an area to install the safeguard system (as a temporary measure) and then begin.

During start-up and functional testing, do not allow workers inside the safeguard area until the safeguard function is activated.

Before start-up and functional testing, carefully read the related manuals and obtain a good understanding of safety-related precautions.

Before supplying the robot and robot system with power for the first time, verify the items listed below.

Items to check before supplying with power

- Prescribed bolts are securely tightened to the robot.
- Electrical connections are set up correctly, and power supply conditions (including voltage, frequency, and error level) are within the specified range.
- Compressed air source (if applicable) is properly connected.
- Peripheral devices are properly connected.
- Safety device is equipped with an interlock switch, and it functions properly.
- Operating environment conditions conform to the conditions specified in the robot and controller manuals.

Items to check after supplying with power

- Start/stop, mode selection, and other functions work properly.
- Moving axes operate normally, and that the area of motion is limited as stipulated in the specifications.
- Emergency stop circuit functions correctly.
- Power supply can be shut off.
- TEACH operation mode is functioning properly.
- Safety device and interlock switch function correctly.
- Other safeguards (if applicable) are installed correctly in their prescribed locations.
- Robot operates accurately in restricted status (low power status).
- Robot operates properly under rated loads and at maximum speed.

Restarting after a Change

When restarting the robot system after its hardware or software has been corrected or serviced, strictly observe the following:

- Before supplying the system with power, check the locations where the hardware was modified.
- Test the functions of the robot system to make sure that it operates correctly.

Precautions regarding Robot Operation

General Precautions

Before operation, become familiar with the location of all emergency stop buttons.

During an emergency, always press the nearest emergency stop button. There should never be any emergency stops in the system that do not operate.

After an emergency, do not restore the emergency stop circuit until it has been determined that the entire system is safe to restart.

If your robot is a 6-Axis type, record the pulse values of the reference points used for calibration. For details, refer to the section *Setup & Operation 3.6 Setting the Reference Points for Calibration* in the 6-Axis manipulator manual.

Automatic Operation

Ensure that system automatic operation is enabled only while the following requirements are being met:

- Emergency stop switches are installed in the prescribed location and operate correctly.
- No personnel are inside the safeguarded area of the system.
- Safety procedures that are established separately for the robot system (if applicable) are being followed.

Teaching Robot Points

If possible, teaching should be performed with no personnel inside the safeguarded area.

Teach mode can be used to allow the robot to be jogged or moved at slow speed when the safeguard is open. Before going inside the safeguarded area, robot operators that need to move the robot under servo control must switch Teach mode to ON by using the mode selector switch of the optional device. Operators then carry the optional device while inside the safeguarded area. As a result, the operation mode cannot be changed from outside the safeguarded system while somebody is inside the safeguarded area.

Auto Mode

The robot cannot be jogged under power with the safeguard circuit open and the motors will be turned off. However, the robot can be moved by hand to a position with the safeguard circuit open and the position can then be taught.

Teach Mode

The robot can be jogged or moved at slow speed as long as the three position enable (dead man) switch is engaged.

Please observe the following guidelines for teaching points:

- Robot operators must receive training that utilizes the same type of robot. Before teaching, the operator should be thoroughly familiar with teaching procedures.
- Before teaching, remove all errors and malfunctions.
- Before the robot operator goes inside the safeguard system, confirm that the robot motors go off when the safeguard is open and that emergency stop switches are functioning correctly.
- The robot operator should visually check the robot system and safeguard system interior to ensure that there are no potential hazards.
- Design the system in such a way that prevents the overall robot system from being started up from any location while the operator is inside the safeguard area.
- If there is a possibility of a dangerous situation arising from the operation of a device other than the robot, such as an actuator, take steps to prevent such operation or ensure that these devices can only be controlled by the teaching operator.

Return to Automatic Operation

If there are safety devices that have been temporarily disabled for system inspection or other reasons, always return them to their original status before restarting automatic operation.

Program Verification

If it is necessary to verify a program while the system is in unrestricted (high power) status, first make sure that no personnel are inside the safeguarded area.

Troubleshooting

Troubleshoot from outside the safeguard system. If that is not possible, strictly observe the requirements below.

- Operators responsible for troubleshooting should be trained and qualified to perform such work.
- Establish work safety procedures to minimize the danger of that operators inside the safeguard system will be exposed to.

Maintenance

In order to keep the robot and robot system operating safely, maintenance (and inspection) is important. Adequately trained personnel should perform the procedures required to do the maintenance work safely. Make sure that maintenance is performed according to the instructions in the robot and controller manuals (maintenance editions).

If maintenance is required inside the safeguard system, take the following precautions:

- Shut off the power supply using lockout / tagout to prevent anyone from turning ON the robot power supply by mistake. For further details, refer to the section *Procedure of Lockout/Tagout* in the chapter *Safety Precautions* in the controller manual.
- If the robot system power supply cannot be shut off, strictly observe the following:
 - (1) Visually inspect the robot system to ensure that there are no conditions that could lead to a malfunction.
 - (2) If it is discovered that the robot system is damaged or malfunctioning, perform the required repairs and retest it before allowing the operator to go inside the safeguard system.
- Grant full control of the robot and robot system to those performing maintenance and/or repairs inside the safeguard system.
- Ensure that the robot system does not respond to any remote control devices.
- Ensure that all emergency stop devices are functioning correctly.
- Before starting the robot system in automatic operation, return all temporarily disabled safety devices to their original enabled status.
- Do not use tweezers or other metal tools to aid in battery replacement. This could cause a battery short. Replace a battery using only the specified type and be careful to observe the polarity of the battery.

Backup of Projects and System

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

To backup project data, select Copy from the EPSON RC+ Project menu. Refer to the section *Copy Command (Project Menu)* in the chapter *The EPSON RC+ GUI* in this manual.

To backup system data, select Maintenance from the EPSON RC+ Tools menu and execute MKVER. Refer to the section *Maintenance Command (Tools Menu)* in the chapter *The EPSON RC+ GUI* in this manual.



CAUTION

- If your system cannot be restored using SETVER, you must restore all drive unit and robot calibration parameters (HOFS, MCOFS, CALPLS) before operating the robot(s). Failure to do so will cause the robot(s) to move to incorrect positions.



In addition to making backup copies of the project and system data, you must also record the configuration data for the following:

- Start Up
- Security

Record the configurations for the items above as follows:

1. Select Setup | System Configuration to open the System Configuration dialog.
2. Select the tab for the item to record.
3. Record the configuration data shown on the tab page.

After restoring the backup data, configure the settings for the items recorded above.

Environmental Requirements

In order to optimize the PC and drive unit performance for safety, they must be placed in an environment which satisfies the following conditions.

Ambient temperature	5 to 40°C (with minimal variation)
Ambient relative humidity	20 to 80 % (with no condensation)
Electrostatic noise	Less than 6kV (without mouse and keyboard)
Environment	<ul style="list-style-type: none"> • Install indoors only. • Place in a well-ventilated area. • Keep away from direct sunlight. • Keep away from dust, oil, salinity, metal powder or other contaminants. • Keep away from flammable or corrosive liquid and gas. • Keep away from water. • Keep away from shocks or vibrations. • Keep away from sources of electronic noise. <p>If the controller must be used in an environment that does not fulfill the above mentioned conditions, take adequate countermeasure. For example, the controller may be covered with a box having a cooling mechanism.</p> <p>The controller is not designed for clean-room specification. If it is to be installed in a clean room, it needs to be covered with a box with either ventilation or cooling mechanism.</p>
Base table	Use a base table that is at least 100 mm off the floor. Placing the controller directly on the floor could allow dust penetration leading to malfunction.
Space	<p>Allow at least 50 mm on each side.</p> <p>There must be room in front of the controller so that the entire controller can be pulled outward. There must also be room behind the controller so that one can attach and remove cables and boards.</p> <p>Do not block airflow to or from the fan.</p>

NOTE



The PC and drive units must be used in the horizontal position.

Precautions for Transport and Storage

Transport and store the manipulator and controller in a temperature range of -25° to +55°C.

Humidity within 10 to 90 % is recommended.

Do not shock or shake the manipulator or controller during transport.

When condensation occurs on the manipulator or controller during transport or storage, turn ON the power supply only after the condensation has dried.

When the manipulator, and controller are used for the robot system again after the long-term storage, perform a test run to verify that they work properly, and then operate them thoroughly.

End User Instruction Manual

Be sure that the robot system instruction manual supplies a list of all the equipment included in the system (such as the robots, associated equipment, and safety devices) as well as a description of how to use each.

Be sure to provide the following in the manual:

- An easy to follow explanation of the robot system and how to install it, as well as a step by step summary of the system installation and external power supply connections.
- A description of all hazards and how to avoid them.
- A description (including interconnection diagrams) of the safety devices, interacting functions, and safeguard's interlocking function against hazardous conditions, in particular, the safeguard's interlocking function for devices installed to perform interactively.
- Precise instructions regarding usage of the system.

End User Training

Be sure those in charge of safety management confirm that the operators who program, operate, and maintain the robot and robot system obtain proper training and have the expertise to conduct the work safely.

Training should include at least the following:

- Study of regulation safety procedures, and safety-related recommendations by robot manufacturers and system designers.
- Clear explanation of the work involved.
- Description of all control equipment required for the work and their functions.
- Explanation of potential hazards involved in the work.
- Work safety procedures and specific methods of avoiding potential hazards.
- Safety device and interlock function testing and confirmation methods are working properly.

Getting Started

This chapter contains instructions for setting up an EPSON RC+ system. It is recommended that first time users first read the preceding Safety chapter, then read through this chapter to get more familiar with the system.

Contents

- Hardware Installation
- Software Installation
- Writing your first program
- Documentation

Hardware Installation

The EPSON RC+ system hardware comes pre-configured at the factory.

For instructions on hardware installation, refer to the manual for the controller you are using:

- RC420 Controller manual.
- RC520 Controller manual.

Software Installation

EPSON RC+ is installed at the factory before shipment with the specified configuration. However, if you need to add an option, upgrade, or re-install, refer to the section *Installing EPSON RC+ Software* in *Appendix A* for instructions on installing EPSON RC+.

Installing the software options key

Included with the EPSON RC+ Software is a software key device that enables options. Without this device, the options you have installed will not be enabled. All options are enabled on one key. Options can also be enabled in the field. See the chapter *Installing Options* for details.

The software options key is installed by simply plugging the male end of the key into the parallel port connector on the PC Control Unit.



CAUTION

Do not plug the options key into the Option Device connector or any other port other than the parallel port. Permanent damage can occur.

Installation for offline development

When installing for offline development, such as a laptop computer, do not select the SPEL Runtime Drivers option, since this requires an MIB board. In offline mode, no software options key is required for editing programs and building projects.

If you are using VB Guide, you may install it and use the VB Guide controls in your VB Guide project. But you cannot run the project.

PC Configuration

Refer to *Appendix A* and configure Date, Time, and screen resolution.

Windows Security Administration

When EPSON RC+ is installed, a user with Administrator rights must install it. Normally this is done at the factory before shipment.

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



User Name and Password at shipment

The RC520 and RC420 robot controllers are configured at factory as shown below:

User Name: EPSON RC User (with Administrator authority)

Password: epson

When the computer name, user name, or password is changed in RC420, the UPS service will not automatically shut down the robot controller in the case of a power failure despite its setting. After you change the computer name, user name, or password, configure the UPS service again so that it will automatically shut down the robot controller in the case of a power failure. For details, refer to the section *Setup & Operation 10.6.3 Setting up the Automatic Shutdown at the Time of a Power Failure* in the RC420 robot controller manual.



For users who use the RC420 robot controller with UPS

The RC420 robot controller is configured at factory as shown below:

Computer Name: Serial No. (SN*****: ***** denotes a five-digit number and it is unique to each controller.)

User Name: EPSON RC User

Password: epson

When the computer name, user name, or password is changed, the UPS service will not automatically shut down the robot controller in the case of a power failure despite its setting. After you change the computer name, user name, or password, configure the UPS service again so that it will automatically shut down the robot controller in the case of a power failure. For details, refer to the section *Setup & Operation 10.6.3 Setting up the Automatic Shutdown at the Time of a Power Failure* in the RC420 robot controller manual.

To provide security within the EPSON RC+ environment, a Security software option is available. This option allows you to manage EPSON RC+ users and audit development activity. See the chapter *Security Option* for details.

Writing your first program

Follow these instructions to create a simple application program so that you will become more familiar with the EPSON RC+ development environment.

1. Start EPSON RC+

Double-click the EPSON RC+ icon on the desktop.

2. Create a new project

- a. Select New from the Project menu.
- b. Type in a name for a project, for example, FirstApp.
- c. Choose **Ok** to create the new project.

When the new project is created, a program called Main.prg is created. You will see a window open with the title Main.prg 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. You can use upper or lower case characters.

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

4. Run the program

Press **F5** to run the program. (F5 is the hot key for the Start selection of the Run menu). You will see the Status window showing the build operation.

Your program is compiled and loaded into memory. If there are no errors during build, the Run window will appear.


Press the **Start** button on the Run window to run the program.

You should see the following displayed on the Run window:

```
-> main started as Task 1
This is my first program.
-> All Tasks Stopped
```

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

5. Teach robot points

Click on the Jog and Teach button  on the toolbar. You will see the Jog and Teach dialog box.

Look at the lower left hand corner of the screen. You will see a drop down list box called **Point File Name**. The current selection is robot1.pnt. This is the default robot point file for your application. Any points that you teach will be saved with this filename.

Click on the **Robot Control Panel** toolbar button . Click on the **Motor On** button to turn on the robot motors. You will be prompted to confirm the operation. Answer Yes to continue.

If the **MCal** button is enabled, press it now to calibrate the robot. You will be prompted to confirm the operation. Answer **Yes** to continue. Then you will see a status window appear and the robot should start calibrating. Type any key, or click the **Stop** button if you need to abort the MCal command. The MCal button is not enabled if you are using an absolute encoder robot.

Exit the Robot Control Panel by clicking on the **Close** button.

Press the **Teach P0** button.

Jog the robot by pressing the +Y jog button. Hold the button down to continue jogging. Let go when the robot is about half way out in the work envelope.

Jog the robot down by pressing the **-Z** button.

Now change the current point to P1 by clicking on the right arrow on the **Point #** slider until the point number is 1.

Press the **Teach P1** button. You will see a confirmation message to teach the point. Answer Yes.

Press the **+X** button to jog the robot in the +X direction.

Change the current point to P2 by clicking on the right arrow on the **Point #** slider until the point number is 2.

Press the **Teach P2** button. You will see a confirmation message to teach the point. Answer Yes.

Leave the jog screen by pressing the **Close** button located in the lower right corner of the screen. You will see a message saying that changes have been made. Click **Yes** to save the changes.

6. Modify the program to include robot motion commands

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

```
Function main
    Print "This is my first program."
    Jump P1
    Jump P2
    Jump P0
Fend
```

Run the program by pressing **F5** and then click on the **Start** button on the Run window. The robot should jump to each of the points you taught.

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

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

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

Run the program by pressing **F5** and then click on the **Start** button on the Run window. The robot should jump 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 system configuration on floppy disk. This is easy to do with EPSON RC+. It is important that you keep regular backups of your applications on external media such as floppy disks.

Follow these steps to backup the project and system configuration:

1. Insert a blank floppy disk in the floppy drive.
2. From the Project menu, select Copy.
3. Change the Destination Drive to A:.
4. Click OK. The project will be copied to the floppy disk.
5. From the Tools menu, select Maintenance.
6. Click on the MKVER button.
7. Enter the MKVER name "sample".
8. Select the A: drive.

9. Click OK. The system configuration will be backed up on the floppy disk.

Now that you have written your first program, you should read the chapter *Building SPEL⁺ Applications*.

Operation

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

- System Power Up Procedure
- Starting EPSON RC+
- The SPEL Driver Manager
- System Shutdown Procedure

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 all Drive Units, I/O devices, PC Control Unit, and monitor.
3. Start the EPSON RC+ software.

Starting EPSON RC+

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

Start Method 1

1. Double click on the EPSON RC+ robot icon located on the Windows desktop to start EPSON RC+.

Start Method 2

1. Click the Windows Start Button.
2. Select the EPSON RC+ Program Group.
3. Select EPSON RC+.

Start Method 3

1. Configure EPSON RC+ to start automatically after Windows starts. The details are described later in this chapter in the section *Start Up Configuration*.

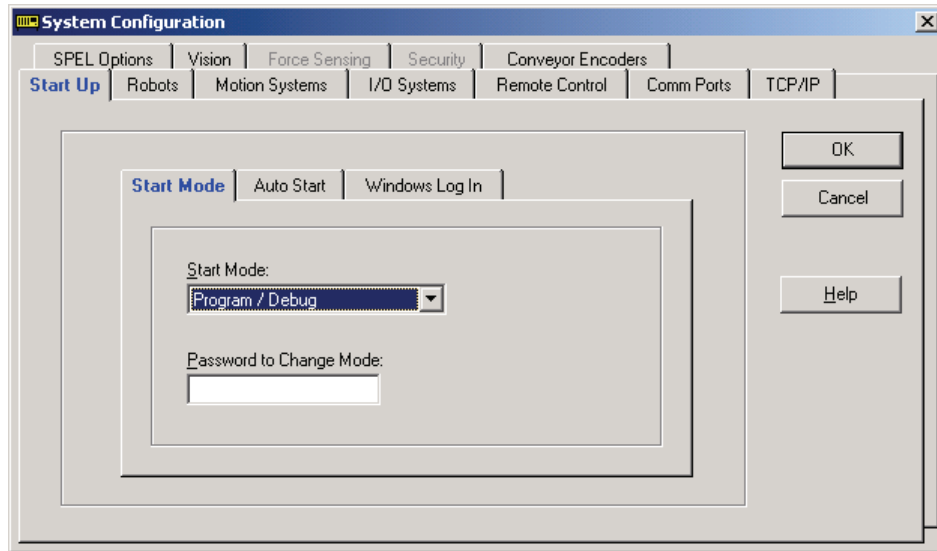
Startup sequence

When EPSON RC+ starts, it reads initial settings for the current user and local system from the Windows registry.

If there are no project files specified on the startup command line, then the last project that was opened will be opened at startup time. The program files that were last opened will be opened again.

Start Up Configuration

To configure start up, select System Configuration from the Setup menu. The Start Up tab has several settings.



Start Mode

There are three basic start modes:

- | | |
|------------------------------|---|
| Program / Debug | This mode allows you to develop your projects. This is the default start up mode. |
| Production / Operator | This mode starts the system and displays the Operator Window. |
| Production / Remote | This mode starts the system for remote control. |
| Production / OP | This mode starts the system for control from the Operator Pendant. |

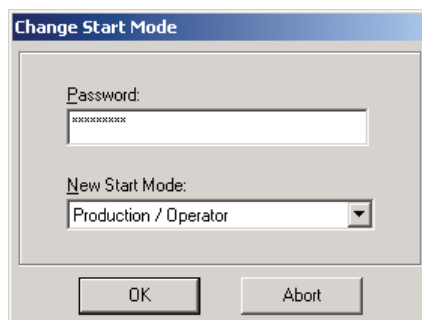
Start Mode Dialog

When the start mode is not set for Program / Debug, then a dialog is displayed at start up that allows you to change the start up mode using a password. After several seconds, if the **Change Start Mode** button has not been clicked, the system will initialize and the Operator Window will be displayed.

You can disable this start up dialog using command line options described later in this section.



If you click the **Change Start Mode** button, another dialog will be displayed, as shown below:



To change the start up mode, you must supply a password, or you can abort start up all together.

This allows authorized personnel to enter Program / Debug mode temporarily to make changes or adjustments.

NOTE



When you change mode from this dialog, it is only temporary. The next time EPSON RC+ runs, the original start mode setting will be used. To permanently change the start mode, you must change it from the Setup | System Configuration | Start Up tab.

Start Up Mode: Program / Debug

Program / Debug is the default start mode. This is the EPSON RC+ development environment, from which you can:

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

Start Up Mode: Production / Operator

The Production / Operator mode can be used as a simple operator interface for production. In this mode, the Operator Windows is displayed. You can configure the functions that are available to the operator.

For more details, see the Operator Window section in the chapter *Building Applications in SPEL⁺*.

Start Up Mode: Production / Remote

The Production / Remote mode is used to automatically prepare EPSON RC+ to accept remote commands. When EPSON RC+ started in Production / Remote mode, the Operator window is displayed with no operator buttons to allow any diagnostic messages to be viewed. The system control device (CtrlDev) is set for Remote.

In Production / Remote mode, you can run without monitor, keyboard, and mouse connected to the PC Control Unit if you also configure Auto Start and Windows Log In, as described below. Refer to the section *Running without video monitor, keyboard, mouse* later in this chapter.

Start Up Mode: Production / OP

The Production / OP mode is used to automatically prepare EPSON RC+ to accept commands from the Operator Pendant. When EPSON RC+ started in Production / OP mode, the Operator window is displayed with no operator buttons to allow any diagnostic messages to be viewed. The system control device (CtrlDev) is set for OP.

In Production / OP mode, you can run without monitor, keyboard, and mouse connected to the PC Control Unit. Refer to the section *Running without video monitor, keyboard, mouse* later in this chapter. Programs can be selected and started from the pendant.

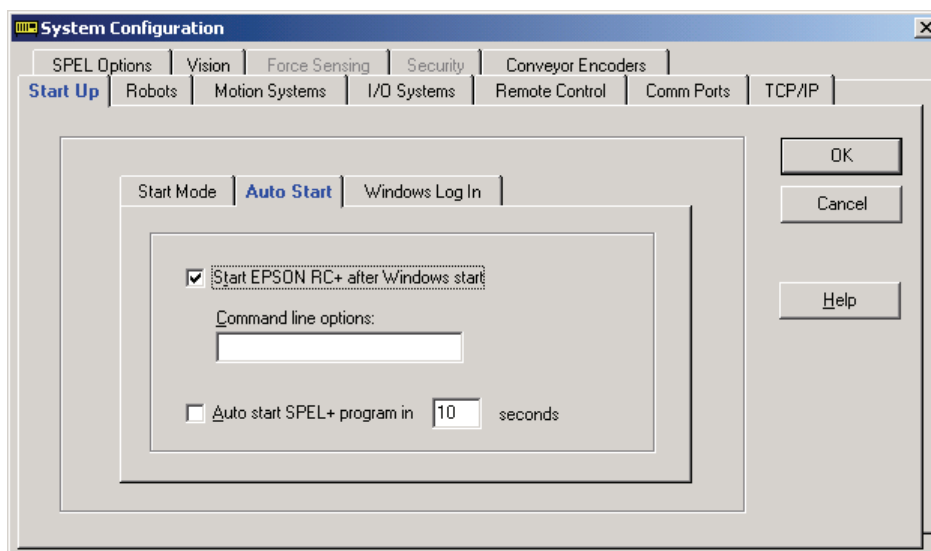
Auto Start

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

From the Auto Start tab, set the **Start EPSON RC+ after Windows start** check box.

You can specify EPSON RC+ command line options (/opr, /nosplash, etc.) in the Command line options text box. Refer to the section *Command Line Options* in the chapter *Operation* for details.

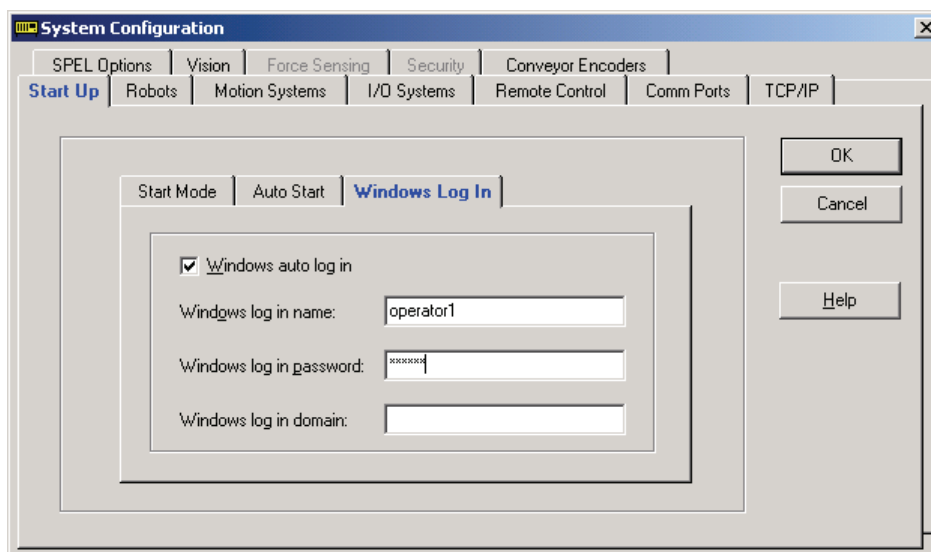
Check the **Auto start SPEL+ program** check box to start the main SPEL+ program after the specified number of seconds. This only works when the start mode is Production / Operator.



Windows Log In

You can configure automatic Windows log in from EPSON RC+. From the Windows Log In tab, type in the name and password of the user logging in. Set the **Windows auto log in** check box. Optionally, you can supply a domain, if required.

You must have Windows Administrator rights to set log in. After making changes, you must reboot the system the first time. After the first reboot, Windows log in will be automatic for boot or log out.





For users who use the RC420 robot controller with UPS

The RC420 robot controller is configured at factory as shown below:

Computer Name: Serial No. (SN*****: ***** denotes a five-digit number and it is unique to each controller.)

User Name: EPSON RC User

Password: epson

When the computer name, user name, or password is changed, the UPS service will not automatically shut down the robot controller in the case of a power failure despite its setting.

After you change the computer name, user name, or password, configure the UPS service again so that it will automatically shut down the robot controller in the case of a power failure. For details, refer to the section *Setup & Operation 10.6.3 Setting up the Automatic Shutdown at the Time of a Power Failure* in the RC420 robot controller manual.

Command Line Options

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

- Start EPSON RC+ for a specified project.
- Change the EPSON RC+ start up mode without a dialog.
- Suppress warnings.
- Disable splash screen

Examples of command line options are:

- Command input from Windows Start menu | Run
- Target property of desktop shortcut

Starting EPSON RC+ for a specific project

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

`ERC40.EXE [drive:project_name]`

drive:project_file The drive letter and name of a project.

Example

Open project *myapp* on drive C: at startup:

`ERC40.EXE c:myapp`

Change the EPSON RC+ start up mode

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

To start in Program / Debug mode (no password required)

`ERC40.EXE /PROG`

To start in Production / Operator mode

`ERC40.EXE /OPR | /OPRAS`

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

If only the OPR flag is supplied, EPSON RC+ will open the project from the last session and display the operator window. EPSON RC+ will only be visible in the Windows Task Manager. When the operator window is closed, EPSON RC+ will be terminated. /OPRAS automatically starts the project after a 10 second count down. The auto start can be aborted during the count down.

If the Security Option is enabled, you can also use the /LOGIN option to automatically log the operator into the system. See the description below.

Example

Open project myapp on drive C and display the operator window:

```
ERC40.EXE c:myapp /OPR
```

To start in Production / Remote mode

```
ERC40.EXE /REMOTE
```

To start in Production / OP mode

```
ERC40.EXE /OP
```



The robot drive unit(s) should be ON before starting EPSON RC+ with the OPR, OPRAS, REMOTE, or OP command line options. If EPSON RC+ cannot communicate with the robot drive unit, then an error dialog will be displayed and the operator window will not be opened.

For more details, see the Operator Window section in the chapter *Building Applications in SPEL⁺*.

Login

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

```
ERC40.EXE /LOGIN "userID", "password"
```

This is especially useful when you are starting in operator mode. You can log in the operator at start up.

If the *userID* or *password* are invalid, a login dialog will be displayed.

Suppress warnings

You can suppress the additional warning messages using the following syntax:

```
ERC40.EXE /NOWARNINGS
```

Or

```
ERC40.EXE /NW
```

Disabling the EPSON RC+ splash window

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

```
ERC40.EXE /NOSPLASH
```

Or

```
ERC40.EXE /NS
```

Running from the Windows Run Box

You can specify a command (e.g. ERC40.EXE C:myapp) from the Windows Start menu | Run | Open text box.

Making startup icons for your projects

You can create icons that automatically start EPSON RC+ for different projects and start modes.

1. Create a new shortcut to ERC40.EXE on your desktop by selecting the EPSON RC+ icon, then right clicking and selecting Create Shortcut..
2. Right click on the icon and select Properties. In the command line text box, type ERC40.EXE followed by the name of the project file you want to open as described above. Add any other option to the command line, such as /OPR, /PROG, /REMOTE, etc.

Running without video monitor, keyboard, mouse

When using Production / Remote or Production / OP modes, you may want to use the system without monitor, keyboard, and mouse connected to the PC Control Unit.

To use RC+ without monitor, keyboard, and mouse, follow these guidelines:

- Configure Auto Start and Windows Login. Verify automatic operation before removing the monitor, keyboard and mouse.
- Do not use MsgBox, InputBox, or other means of operator input in your programs.
- To shutdown the system, you must execute the ShutDown command using an input to your program.
- If a critical system error occurs, such as emergency stop circuit failure or safeguard circuit failure, the system will appear to be hung, because a critical message is being displayed. In this case, you must connect a monitor, keyboard, and mouse to correct the problem.

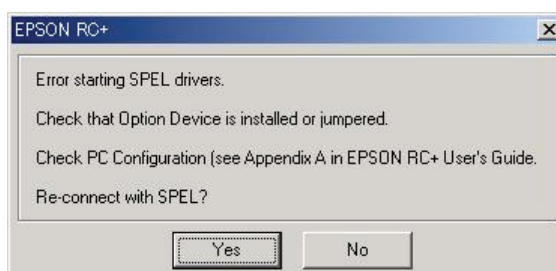
Problems during Start up

When EPSON RC+ is started for use SPEL+ Runtime (which can include one or more robots), it checks if the SPEL runtime drivers are running. If not, the drivers are started. After several seconds, you may see the following dialog when using an RC520 controller:



SPEL Driver startup error: RC520

When the RC420 controller is used, the following dialog will be displayed:



SPEL Driver startup error: RC420

For the RC520, the most common causes of this dialog are that power is off for one or more drive units, or that one or more cables are not properly seated. Make sure drive unit power is on.

After clicking Yes, EPSON RC+ attempts to restart the SPEL runtime drivers. If the dialog appears again, please check the MIB board(s) jumpers and cables (RC520) or Drive Module DIP Switch settings (RC420), then try again.

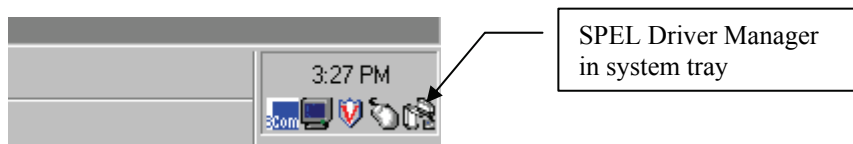
If you answer no to the dialog above, you can continue to work in the EPSON RC+ environment, but if you try to execute any commands involving a robot, or attempt to run SPEL+ programs, you will see this dialog again.

The SPEL Driver Manager

When EPSON RC+ is first started, a program called SPEL Driver Manager is started and an icon is installed in the system tray. When EPSON RC+ is exited, the SPEL drivers remain running. This allows EPSON RC+ to start up faster on the next use. Also, the robot drive motors will remain on between EPSON RC+ sessions and MCal will not be required again for incremental encoder robots.

The SPEL Manager can be used to shutdown or restart the SPEL drivers. If EPSON RC+ is not running, then only the Shutdown menu item will be enabled.

To use the SPEL Manager, right click on the robot icon in the system tray and select a menu item.



System Shutdown Procedure

It is very important to shutdown the system properly. Files on the PC may be damaged if the PC is turned off before shutdown.

Follow this procedure to shutdown the system:

1. Stop all tasks. You cannot shut down while tasks are running. If you attempt to, you will get an error message and system shutdown will be aborted.
2. Shutdown the EPSON RC+ software by executing File | Exit.
3. Shutdown Windows.

RC420 Controller:

4. Remove power from the controller and monitor.
5. Remove power from I/O devices.

RC520 Controller:

4. Remove power from the PC and monitor.
5. Remove power from the drive units and I/O devices.

The EPSON RC+ GUI

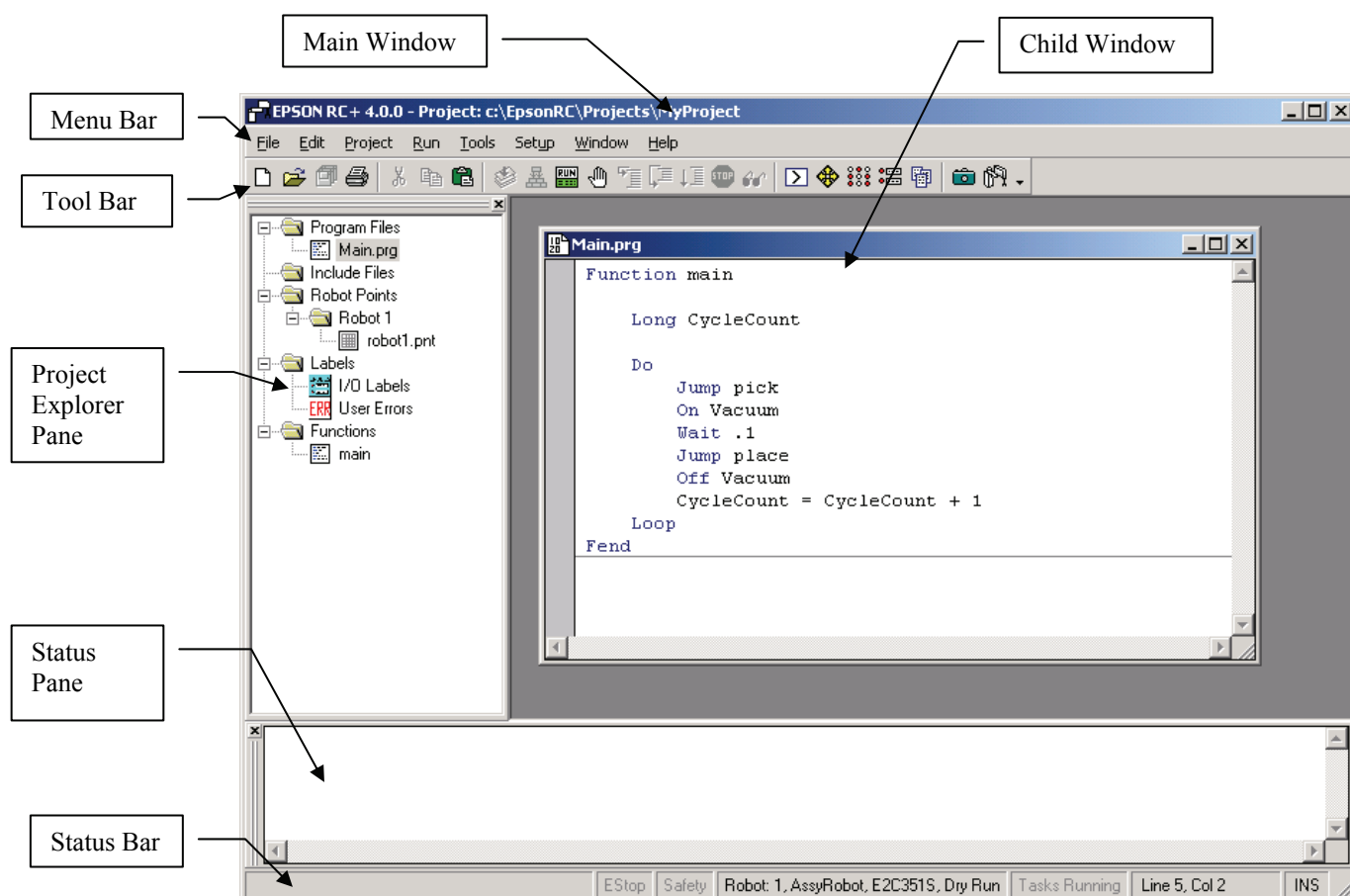
The chapter contains information on the EPSON RC+ GUI. After an overview and discussing online help and toolbars, the remainder of the chapter is divided into sections that follow the menu system.

Contents

- Overview
- Online Help
- Toolbars
- File Menu
- Edit Menu
- Project Menu
- Run Menu
- Tools Menu
- Setup Menu
- Help Menu

GUI Overview

EPSON RC+ is a multiple document interface (MDI) application. There is one main parent window and several child windows which can be opened simultaneously. The main window has a menu bar, tool bar, and status bar, as shown below. In addition, there is a Project Explorer pane and Status pane.



Only one session of EPSON RC+ can be running at one time. If you try to start EPSON RC+ while it is already running, it will be restored from minimization (if necessary) and brought to the foreground.

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.

To open a file or jump to a function, double-click on the item.

To hide the Project Explorer, right click on the bar above the pane or the toolbar, then uncheck the Project Explorer selection.

To show the Project Explorer, right click on the bar above the pane or the toolbar, then check the Project Explorer selection.

To resize the Project Explorer, move the mouse cursor over the right side of the pane, then drag the pane right or left to the desired width.

You can move the Project Explorer pane to either the left or right side of the main window. To move the pane, click down on the bar above the pane, then drag to either the left or right side of the main window and release the mouse button.

Status Pane

The status pane displays status messages, such as project build status.

To hide the Status pane, right click on the bar on the left side of the pane or the toolbar, then uncheck the Status selection.

To show the Status pane, right click on the bar on the left side of the pane or the toolbar, then check the Status selection.

To resize the Status pane, move the mouse cursor over the top edge of the pane, then drag the top edge up or down.

The Status pane is always located at the bottom of the main window and cannot be moved.

Status Bar

The status bar located at the bottom of the main window displays the following:

Message area	Syntax error for the current line, system messages.
Emergency Stop status	Indicates if emergency stop is active.
Safeguard status	Indicates if one or more safeguard circuits is open.
Current robot	Shows current robot number, name, and model along with Dry Run status.
Tasks running status	Indicates that one or more tasks are running.
Current Line and Column	When a program editor window is active, the current line and column are displayed.
INS / OVR status	Indicates insert or overtype mode.

Online Help

EPSON RC+ has an extensive context sensitive help system.

There are several methods to get help.

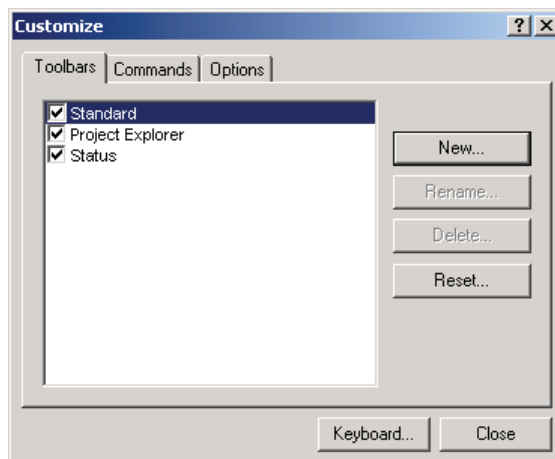
- Select **Contents** from the Help menu to browse help topics.
- Select **Search for Help On...** from the Help menu to search for a specific topic.
- When editing programs, press F1 with the caret in the keyword of interest.
- When errors are displayed, press F1 with the caret on the error line.
- When a dialog is open, press F1 or click the Help button.

Toolbars

You can create your own toolbars or you can add tool bar buttons to the standard tool bar.

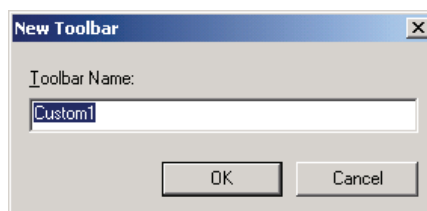
To customize toolbars, right click on the main tool bar and select **Customize**.

The following dialog will appear.



To create a new tool bar

1. Click the **New** button.



2. Type in the name of the new tool bar and click **OK**.
3. A blank toolbar will be created.
4. Now select the **Commands** tab.
5. Select the category for the commands you want to add to your toolbar.
6. Drag the commands you want onto the new toolbar.
7. Click **Close**.

To add new commands to an existing tool bar

1. Open the **Customize** dialog by right clicking on the main toolbar and selecting **Customize**.
2. Click the **Commands** tab.
3. Select the category for the commands you want to add to the toolbar.
4. Drag the commands you want onto the toolbar.
5. Click **Close**.

File Menu

The EPSON RC+ File Menu includes commands for managing and printing files in the current project.

New Command (File Menu)

The New command is used to add new files to the current project. When the New command is selected, the Project Edit dialog is opened. You create new files from the Project Editor. Please refer to that section for details.

Shortcuts


Toolbar: 

Open Command (File Menu)

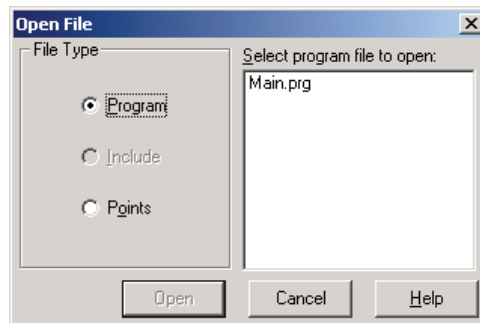
Open one or more files in the current project for editing. You can open either program source files, include files, or robot points files.

If there is a file in the current project directory (as shown in the Edit Project dialog box) and the file is not in the current project, you will not be able to open the file. You must add the file to the project before you can open it. This also pertains to include files and point files.

Shortcuts

Toolbar: 

Keys: CTRL+O



Dialog Box Options

Option	Description
Program	Select this radio button to show a list of program file names in the files list box.
Include	Select this radio button to show a list of include file names in the files list box.
Points	Select this radio button to show a list of point file names in the files list box.
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 / deselect any file. The Shift key allows you to select a group of files.
Open	Opens the selected file(s).
Cancel	Cancels the open operation.

Tip: You can also double click on a file name in the **Select file to open** list box to open the file without having to choose the **OK** button.

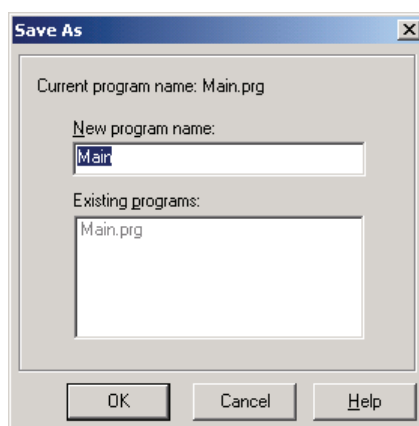
Save Command (File Menu)

The Save command writes the current file to disk. The current file can be a program file, include file, points file, I/O label editor, etc. This command is disabled if the current file does not need to be saved.

Save As Command (File Menu)

Save the program, include file, or point file in the currently active window with a new file name. The original file will be left in the project directory unchanged. The new name will be used throughout the current project in place of the old name.

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



Restore Command (File Menu)

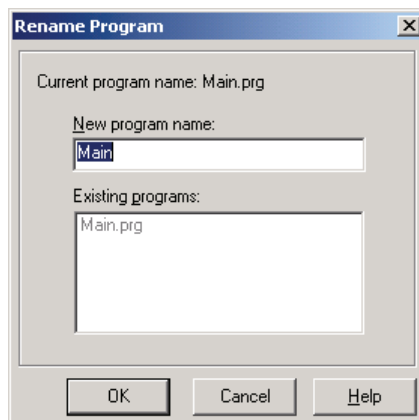
Restores the currently active program, include file, I/O labels, vision sequence data, or points from disk.

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

You will be prompted to confirm this operation.

Rename Command (File Menu)

Use Rename to change the name of the program, include file, or point file you are currently editing.



To rename a file

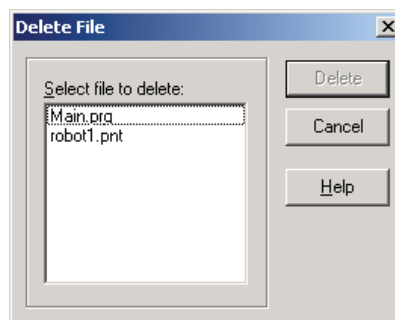
Make the file you want to rename the current active window by clicking anywhere on the program window, or selecting it from the list in the Window Menu, or by opening the file with the Open command from the File Menu.

Type in a new name for the file and click **OK**. Refer to the existing files list to see what other programs are in the current project directory. The new file name you enter cannot be the same as any other file in the project directory. You will get an error message if you enter a new name that is already being used.

If you use **Rename** on an include file, you must rename the file in each of your #include statements that refer to it.

Delete Command (File Menu)

This command allows you to delete a file in the current project directory. You can delete program files, include files, and point files. The file does not have to be in the project make list to delete it.



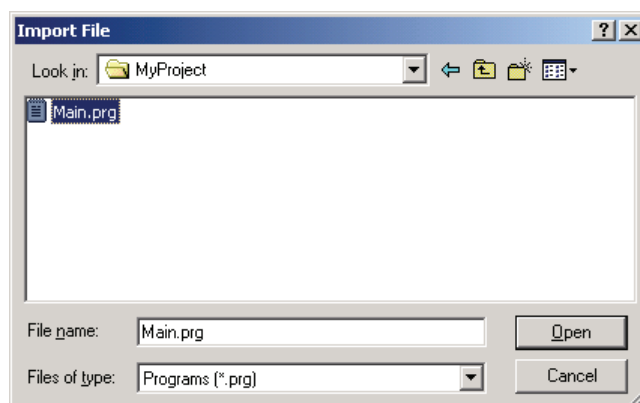
Dialog Box Options

Option	Description
Select file to delete	Click on the file name you want to delete. This file list displays all .PRG, .INC, and .PNT files in the current project directory.
Delete	Delete 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.
Cancel	Cancels the delete operation.

Import Command (File Menu)

Import a program or point file into the current project. Use this command to import programs and points from other EPSON RC+ projects.

- Program file names for importing must have a .PRG extension.
- Include file names for importing must have a .INC extension.
- Point file names for importing must have a .PNT extension.
- Input labels must have the file name INPLABEL.TXT.
- Output labels must have the file name OUTLABEL.TXT.
- Memory labels must have the file name MEMLABEL.TXT.
- Ethernet I/O labels must have the file name ENETLABEL.TXT.
- User errors must have the file name USERERRORS.TXT.
- Macros must have the .MAC extension.



To import a file

1. Select the file type from the **File Type** list box.
2. Select the drive, directory, and file name of the program you want to import.
3. Press **OK** to continue. If a file is being overwritten, you will be prompted to continue. The file will then be copied to the current project's directory. If you are importing a program, include file, or point file that is not in the current project, the Edit Project dialog box will open, with the name of the file you are importing in the File Name text box. Add the file(s) to appropriate group.
4. Press **OK** to save the changes to the project.

NOTE




If Program Groups are enabled and you do not put the imported program in any program groups, you will not be able to open the file for editing. If you just want to open an imported file for copy and paste, then create a program group to put the program in.

Print Command (File Menu)

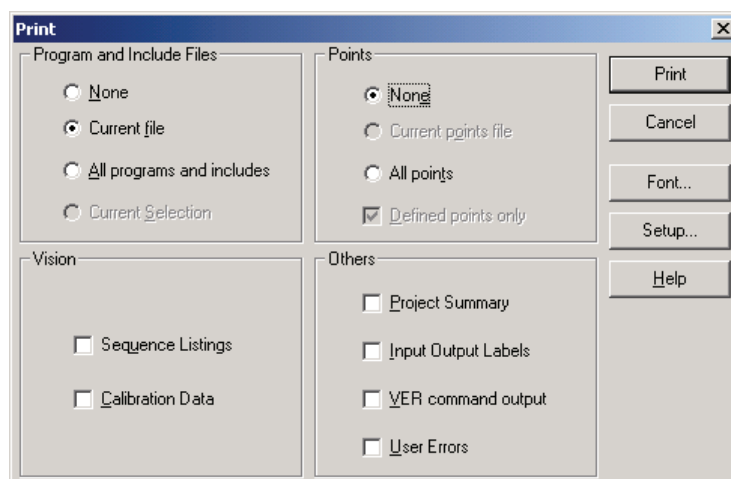
This command opens the Print dialog box. From this dialog, you can print one or more program or point files in the current project. You can also print out the robot controller configuration.

Each document is printed with a header that includes the file name, date and time, and page number.

Shortcuts

Toolbar: 

Keys: CTRL+P



Dialog Box Options

Option	Description
Program Group	
None	Select this radio button if you do not want to print any program source files.
Current file	Select this radio button to print the currently active program document.
All programs and includes	All program files in the project will be printed.
Current selection	The selected text in the currently active program document will be printed.
Points Group	
None	Select this radio button if you do not want to print any point files.
Current file	Select this radio button to print the currently active point document.
All points	Print all point files.
Defined points only	When this check box is on, only points that are defined will be listed.
Vision Group	
Sequence listings	Prints listings for all vision sequences in the current project. If Vision Guide is not installed, this option will not be available.
Calibration data	Prints listings for all vision camera calibration schemes in the current project. If Vision Guide is not installed, this option will not be available.
Others Group	
Project Summary	Select this check box to print a summary of the programs and points used in the current project.
Input Output Labels	Select this check box to print a listing of the all of the I/O labels used in the project.
Ver command output	Prints a listing for the Ver command. You will be prompted for a title that will be added to the top of the printout.
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.
Buttons	
Print	Print the selected files. This button will be dimmed if nothing is selected to be printed.
Font...	Opens a dialog for selecting the printer font. The selected font is saved for subsequent printing.
Setup...	Opens a dialog for selecting which printer to use. You can also select Portrait or Landscape mode.
Cancel	Closes the dialog box.

Exit Command (File Menu)

Exits from EPSON RC+.

If you are running a program from the Run Window, you will see a message that a program is running and you will not be allowed to exit. You must abort all tasks first before you can exit.

If there are any open program or point files that have not been saved, you will be prompted to save these files with Yes, No, or Cancel.

If you select **Yes**, then all unsaved files in the project will be saved.

If you select **No**, then the program will exit without saving the files.

If you select **Cancel**, then the exit command will be aborted.


Edit Menu

The EPSON RC+ Edit Menu includes commands for editing program files.

Undo Command (Edit Menu)

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

Shortcuts

Toolbar: 

Keys: CTRL+Z

Cut Command (Edit Menu)

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

Shortcuts


Toolbar: 

Keys: CTRL+X

Copy Command (Edit Menu)

Copies the current selection into the Clipboard.

Shortcuts

Toolbar: 

Keys: CTRL+C

Paste Command (Edit Menu)

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

Shortcuts


Toolbar: 

Keys: CTRL+V

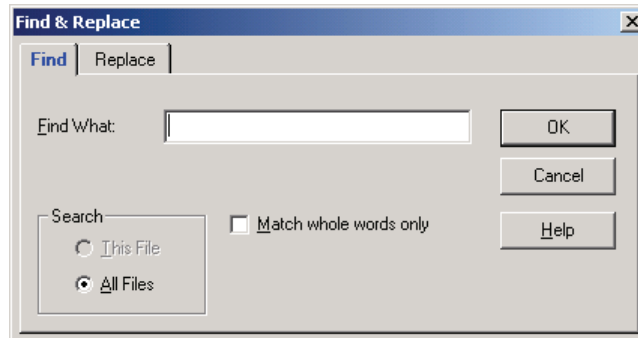
Find Command (Edit Menu)

Find a text string in the current program or all programs in the project. The first time you execute this command, the dialog box will be centered over the main window. If you reposition it, then the next time Find is executed, the dialog will appear where you last positioned it.

Shortcuts

Toolbar: 

Keys: CTRL+F



Dialog Box Options

Option	Description
Find What	Type the text you want to search for. If any text was selected when you chose the Find command, it will be displayed here. If no text was selected, then the text from the last Find will be displayed. You are limited to one line of text. If you select more than one line before executing Find, then only the first line will be displayed here. Tabs are represented by "\t".
This File	Search only in the current program file.
All Files	Search all files in the project.
Match whole words only	Search for the full word by itself and not as part of another word. This check box is not available if the search string contains more than one word.
OK	Start the search. If the text is found in a file that is not open, then it will be opened.
Cancel	Closes the dialog box without performing the search.

Find Next Command (Edit Menu)

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

Shortcuts

Toolbar: 

Keys: F3 or CTRL+G

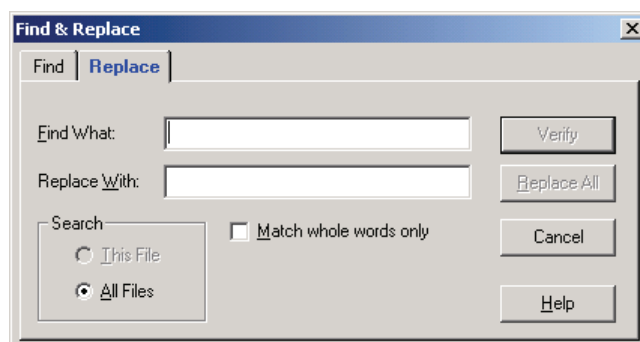
Replace Command (Edit Menu)

Search for a text string and replace it with new text. The first time you execute this command, the dialog box will be centered over the main window. If you reposition it, then the next time Replace is executed, the dialog will appear where you last positioned it.

Shortcuts

Toolbar: 

Keys: CTRL+R




Dialog Box Options

Option	Description
Find What	Type the text you want to search for. If any text was selected when you chose the Replace command, it will be displayed here. If no text was selected, then the text from the last Find will be displayed.
Replace With	Type the replacement text.
This File	Search only in the current program file.
All Files	Search all files in the project.
Match whole words only	Search for the full word by itself and not as part of another word.
Verify	Start the search. If the text is found, a dialog box will appear asking if you want to replace the text or not. Choose Yes to replace the text, No to continue the search, Cancel to abort.
Replace All	Replace all occurrences without any verification.
Cancel	Closes the dialog box without performing the search.

Select All Command (Edit Menu)

Select entire program file or entire point file. You can then execute Cut or Copy.

Shortcuts


Toolbar: 

Keys: CTRL+A

Indent Command (Edit Menu)

Move the selected text block one tab to the right.

Shortcuts


Toolbar: 

Keys: TAB

Outdent Command (Edit Menu)

Move the selected text block one tab to the left.

Shortcuts


Toolbar: 

Keys: Shift+Tab

Comment Block Command (Edit Menu)

Comments out the selected block of lines by adding the comment character to the beginning of each line. To use, select one or more lines to be commented. Then select Comment Block from the Edit menu, or toolbar, or right click and select Comment Block from the pop-up menu. A comment character will be added to the beginning of each of the selected lines.

Shortcuts

Toolbar: 

Uncomment Block Command (Edit Menu)

Removes leading comment character from the selected block of lines. To use, select one or more lines to be uncommented. Then select Uncomment Block from the Edit menu, or toolbar, or right click and select Uncomment Block from the pop-up menu. The first comment character from each of the selected lines will be removed.

Shortcuts

Toolbar: 

Go To Definition Command (Edit Menu)

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 an identifier in a program window, then select Go To Definition from the Edit menu or toolbar. You can also right click on the identifier and select Go To Definition from the pop-up menu.

If the identifier is a function name or variable, the program window where it is declared will be opened and the current line number will be set.

If the identifier is an point label, the point file will be opened and the current row for the label will be displayed.

If the identifier is an I/O label, the I/O label editor will be opened and the current row for the label will be displayed.

If the identifier is a user error label, the User Error editor will be opened and the current row for the label will be displayed.

Shortcuts

Toolbar: 


Renumber Command (Edit Menu)

Renumber the currently active program. A dialog box is displayed that shows the starting line number and the increment. If the current program has line numbers, then the starting line number will equal the line number for the first line of the program.

The default line number increment is determined by the setting in Preferences: Editor tab. GoTo and GoSub line numbers are automatically adjusted. If you change the increment, then the line number increment preference is set to this value.

If **Auto renumber before save** is checked in the Preferences: Editor tab, then your program will automatically be renumbered every time you save it.

Shortcuts


Toolbar: 

Strip Line Numbers Command (Edit Menu)

Use this command to strip all line numbers from the currently active program window. This is useful for converting program files that were written using line numbers.

The line numbers used in GoTo, GoSub, and OnErr statements will be replaced with labels.

Shortcuts

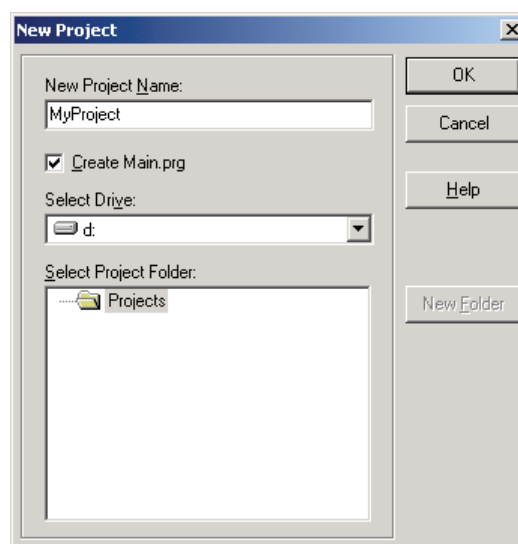
Toolbar: 

Project Menu

The EPSON RC+ Project Menu includes commands for managing and building projects.

New Command (Project Menu)

The New command is used to create a new EPSON RC+ project. Projects can be on any disk drive on the system. They are stored in the \epsonrc\projects directory on the selected drive. Subdirectories can also be created.

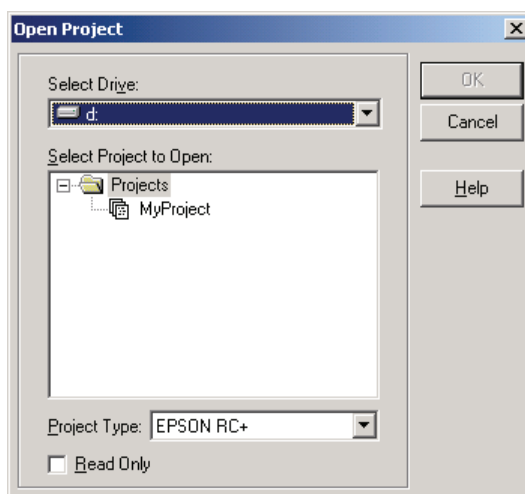


Dialog Box Options

Option	Description
New Project Name	Type in a new name for the project. The name can include alphanumeric characters along with underscores.
Select Project Folder	This is a list of folders and projects on the selected drive. If you click on a name in this list, it will be displayed in the New Project Name text box. 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.
Select Drive	Select the desired disk drive for the new project.
New Folder	Click this button to create a new folder in the currently selected folder.
Create Main.prg	Turn this check box on if you want Main.prg to be created when the project is created.
OK	Choose OK to create the new project. If Create Main.prg was checked, then it will be opened.
Cancel	Choose Cancel to abort creating a new project.
Help	Click help for instructions.

Open Command (Project Menu)

Use this command to open a EPSON RC+ application project. When a project is opened, the previous project is closed out. You will be prompted to save changes.



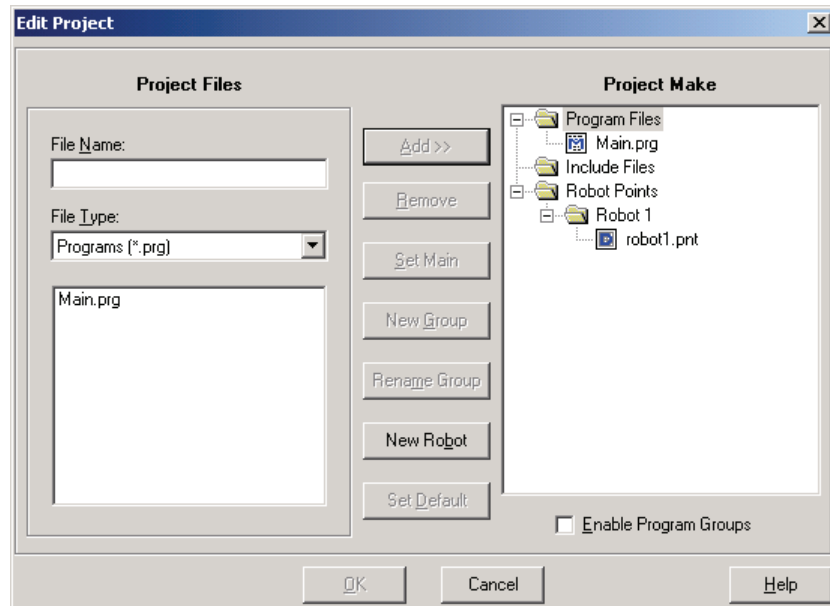
Dialog Box Options

Option	Description
Select Drive	Select the desired disk drive for the project you want to open.
Select Project to Open	Select a project name from the list box. To open a folder, double click on the folder or click the + box located to the left of the folder.
Project Type	This drop down list only appears if projects created with SeikoRCS or SPEL ⁺ for Windows are found on the selected drive. You can then select EPSON RC+, SeikoRCS, or SPEL for Windows projects. You can open SeikoRCS and SPEL for Windows projects and convert them for use with EPSON RC+. The project files are automatically copied to the EPSON RC+ project directory. You will be prompted to convert the project or open for read-only.
OK	Open the selected project.
Cancel	Cancel the operation.

If you are opening a SPEL for Windows project, you will get a message asking to convert the project or open for read only. If the project was using more than one program group with separate point files, you will get a warning message and no point files will be added to the converted project. The point files will remain in the new project directory. This is done for safety because point files are no longer associated with program groups, and therefore points are not automatically loaded when a group is started, either from the Run Window, Operator Window, or Chain command. You must add the point files to the project and then add LoadPoints statements in your project to load the correct point files.

Edit Command (Project Menu)

This Edit command is used to define which program source files and point files are to be used in the current project. It is divided into two main sections: The left section contains a list of files in the current project directory. You can select which files to view from the **File Type** list box. The right section contains a project make tree that includes program groups, include files, and robot points.



The files shown in the file list are in the current project directory. Before you can use a file in the project, you must put it into the project make tree using the Add button.

What is a program group?

A program group is a collection of one or more program files that contain SPEL⁺ functions.

Each group is an executable program containing functions. During project build, all of the programs in a group are compiled and linked into one OBJ file.

Each group has one main program. The first function in the main program is started when the group is started. You can set any program in a group to be the main program.

Program groups are optional. By default, a new project does not use program groups.

To create a new program

1. Select the program group that you want the new program to be in by clicking on its folder in the project make tree.
2. Type the name of the program in the **File Name** text box in the Program Files section.
3. Click the **Add >>** button next to the program files list. You will get a message asking if its OK to create the new file. Answer **OK** to create the file and put it in the group you selected in step 1.

To enable / disable program groups

Click the **Enable Program Groups** check box.

To add an existing program source file to a program group

1. Ensure that program groups are enabled by turning on the **Enable Program Groups** check box.

2. Select the group you want to add a program to by clicking on its folder in the project make tree.
3. Select the program name you want to add to the group by clicking on a program name in the program files list box.
4. Click the **Add >>** button

Or

Double click on the program name in the file list box.

The file will be added to the list of programs for the group you selected.

If the file is already in the group, you will get an error message.

To create a new include file

1. Type the name of the include file in the **File Name** text box. The name of the include file can also be the same name as a program or program group.
2. Click the **Add >>** button. You will get a message asking if its ok to create the new file. Click **OK** to create the file and put it in the project make tree.

To add an existing include file to the project

1. Select the include file name you want to add to the project by clicking on an include file name in the file list box.
2. Click the **Add >>** button.

Or

Double click on the include file name in the file list box.

The file will be added to the list of include files for project make.

Editing program groups

The right half of the dialog box contains a project make tree. Each program group can contain one or more program files that can specify several SPEL⁺ functions. .

You can create and delete program groups. You can put the same program in more than one group.

To open or close a program group folder

Double click on the group folder icon or click on the "+" button next to the folder.

When you open a group folder, a list of programs will be shown below the folder indicated by program icons.

To close the folder, double click on the folder icon, or click the "-" button next to the folder.

To remove a program group

1. Select a group folder with the mouse or by using the up/down arrow keys.
2. Click the **Remove** button. You will be prompted to confirm the operation.

To remove a program from a program group

1. Select the program you want to remove from the group list.
2. Click the **Remove** button.

To create a new program group

1. Click the **New Group** button. A dialog box will prompt you for a new program group name.
2. Click **OK** to create the new program group.

To set the main program in a group

1. Click the program you want to be the main program.
2. Click the **Set Main** button. The program will be moved to the top of the group and the icon will change to a program icon with an "M".

Robot Points

Each robot on the system has its own set of point files in the project. There must be at least one point file for each robot being used. Point files cannot be shared between robots.

One point file is designated as the default by using the **Set Default** button.

To add a robot

Click on the **New Robot** button to add a robot to the current project.

To add a new robot point file

1. Click the robot folder where you want to add the point file.
2. Enter the name of the point file you want to create. Use the PNT extension.
3. Click the **Add >>** button. You will get a message asking if its ok to create the new file. Click **OK** to create the file and put it in the project make tree

To add an existing robot point file

1. Click the robot folder where you want to add the point file.
2. Select Points from the **FileType** list box.
3. Select the point file you want to add from the list.
4. Click the **Add >>** button.

To set default robot point file

1. Click the point file you want to be the default.
2. Click the **Set Default** button. The point file will be moved to the top of the list and the icon will change to a point file icon with a "D".

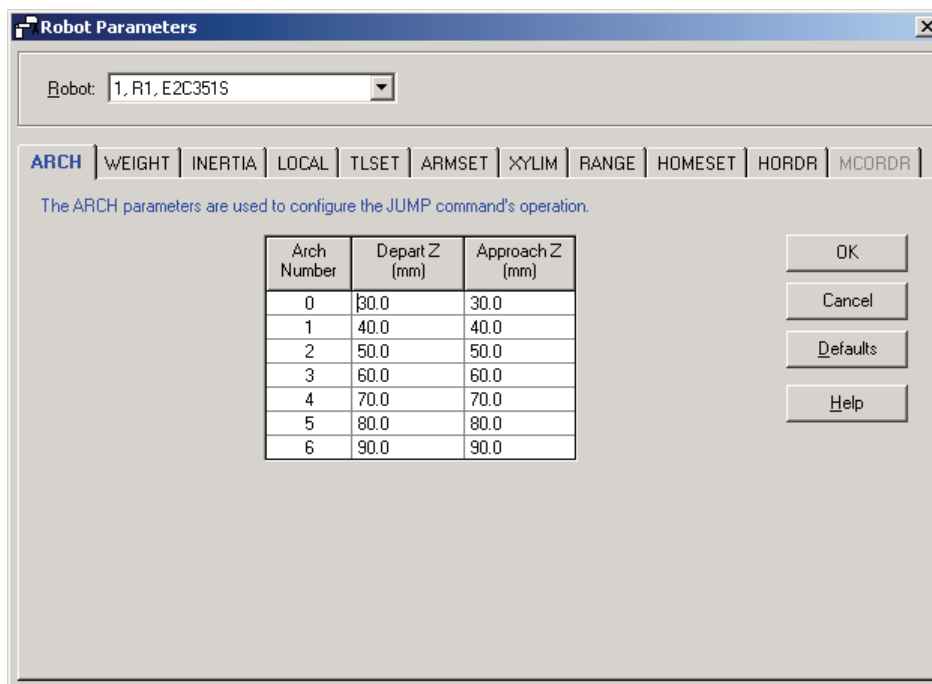
Robot Parameters Command (Project Menu)

This command opens the Robot Parameters dialog. This dialog contains several tabs that are used to set parameters for all robots on the system for the current project.

Arch Tab: Robot Parameters

This tab allows you to configure the depart Z and approach Z settings in the Arch table. Arch is used for the Jump, Jump3, and Jump3CP motion commands.

There are seven different setting pairs in the Arch table. Each setting pair is one row in the grid.



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.

Press the TAB key to move to the next cell.

Dialog Box Options

Option	Description
Defaults	Click the defaults button to display factory default settings.
OK	Choose OK to save the new Arch settings.
Cancel	Exit without saving changes.

Weight Tab: Robot Parameters

This tab is for changing the Weight parameters for the current robot.

Robot Parameters

Robot: 1, R1, E2C351S

ARCH **WEIGHT** INERTIA LOCAL TLSET ARMSET XYLM RANGE HOMESET HORDR MCORDR

Configure payload parameters

Weight: End effector + part to be carried.

Length: Optional distance from rotational center of joint 2 to center of gravity of end effector and part.

Weight: 1.00 ☒ Kg ☐ Lbs.

Length: 125 millimeters

OK
Cancel
Defaults
Help

Equivalent program statement:
WEIGHT 1, 125

Dialog Box Options

Option	Description
Weight	Type in the new weight of the payload on the robot.
Kg/Lb	Choose which unit the weight is represented in: kilograms or pounds.
Length	Type in the new length.
Defaults	Click the defaults button to display factory default settings.
OK	Choose OK to save the new settings.
Cancel	Exit without saving changes.

Inertia Tab: Robot Parameters

This tab is for changing the Inertia parameters. The tab is disabled if the current robot does not support the Inertia command.

The screenshot shows the 'Robot Parameters' dialog box with the 'INERTIA' tab selected. At the top, a dropdown menu shows 'Robot: 1, r1, E2C351S'. Below the tabs (ARCH, WEIGHT, INERTIA, LOCAL, TLSET, ARMSET, XYLM, RANGE, HOMESET, HORDR, MCORDR), the text 'Configure inertia parameters' is displayed. The main area contains two labels: 'Load inertia: End effector + part to be carried.' and 'Eccentricity: Optional distance from rotational center of joint 4 to center of gravity of end effector and part.' Below these are two input fields: 'Load inertia: 0.005 kgm²' and 'Eccentricity: 0.000 millimeters'. To the right are four buttons: 'OK', 'Cancel', 'Defaults', and 'Help'. At the bottom, a text box labeled 'Equivalent program statement:' contains the text 'INERTIA 0.005, 0'.

Dialog Box Options

Option	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.
Defaults	Press the defaults button to display factory default settings.
OK	Choose OK to save the new settings.
Cancel	Exit without saving changes.

Local Tab: Robot Parameters

This tab allows you to define local coordinate systems for a robot. When the tab is selected, the current values are displayed.

A grid is used to display all of the values for the locals you can define.

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 and the local will be defined when you press the **OK** button.

Robot Parameters

Robot: 1, r1, E2C351S

ARCH | WEIGHT | INERTIA | **LOCAL** | TLSET | ARMSET | XYLM | RANGE | HOMESSET | HORDR | MCORDR

Define LOCALs.

Local Number	X (mm)	Y (mm)	Z (mm)	U (deg)	V (deg)	W (deg)
1						
2						
3						
4						
5						
6						
7						
8						
9						
10						

OK
Cancel
Help

Navigating the grid

Use the TAB key to move to the next field. Use the arrow keys or the mouse to move to any field.

For more information on locals, see the description for the Local command.

Dialog Box Options

Option	Description
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). For 6-axis robots only.
W	Rotation angle of the local about the base X axis (yaw). For 6-axis robots only.
OK	Save changes.
Cancel	Cancel changes and exit.

TLSet Tab: Robot Parameters

This tab allows you to define tool settings for a robot. When the tab is selected, the current values are displayed.

A grid is used to display all of the values for all three tools you can define.

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 tool, then the remaining fields will be set to zero and the tool will be defined when you press the **OK** button.

Robot: 1, r1, E2C351S

ARCH | WEIGHT | INERTIA | LOCAL | **TLSET** | ARMSET | XY LIM | RANGE | HOMESET | HORDR | MCORDR

Define TOOLS.

Tool Number	X (mm)	Y (mm)	Z (mm)	U (deg)	V (deg)	W (deg)
1						
2						
3						

OK
Cancel
Help

Navigating the grid

Use the TAB key to move to the next field. Use the arrow keys or the mouse to move to any field.

For more information on tools, see the description for the TLSet command.

Dialog Box Options

Option	Description
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). For 6-axis robots only.
W	Rotation angle of the tool about the X axis (yaw). For 6-axis robots only.
OK	Save changes.
Cancel	Cancel changes and exit.

ArmSet Tab: Robot Parameters

This tab allows you to define Arm settings for a robot. When the tab is selected, the current ArmSet values are displayed. The tab is disabled if the current robot does not support the ArmSet command.

A grid is used to display all of the values for all three arm configurations you can define.

When an arm is undefined, then all fields for that tool 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 press the **OK** button.

Navigating the grid

Use the TAB key to move to the next field. Use the arrow keys or the mouse to move to any field.

For more information on arm parameters, see the description for the ArmSet command in the SPEL⁺ Language Reference.

Dialog Box Options

Option	Description
Parm 1 - Parm 5	These parameters depend on the robot type. See the description for the ArmSet command for details.
OK	Save changes.
Cancel	Cancel changes and exit.

ECPSet Tab: Robot Parameters

This tab allows you to define ECP (external control point) settings for a robot. When the tab is selected, the current values are displayed.

A grid is used to display all of the values for all ECPs you can define.

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, then the remaining fields will be set to zero and the ECP will be defined when you press the **OK** button.

Robot Parameters

Robot: 1, R1, PS3-AS00

ARCH | WEIGHT | INERTIA | LOCAL | TLSET | ARMSET | **ECPSET** | XYLM | RANGE | HOMESET | HORDR | MCORDR

Define ECPs

ECP Number	X (mm)	Y (mm)	Z (mm)	U (deg)	V (deg)	W (deg)
1						
2						
3						
4						
5						
6						
7						
8						
9						
10						

OK
Cancel
Help

Navigating the grid

Use the TAB key to move to the next field. Use the arrow keys or the mouse to move to any field.

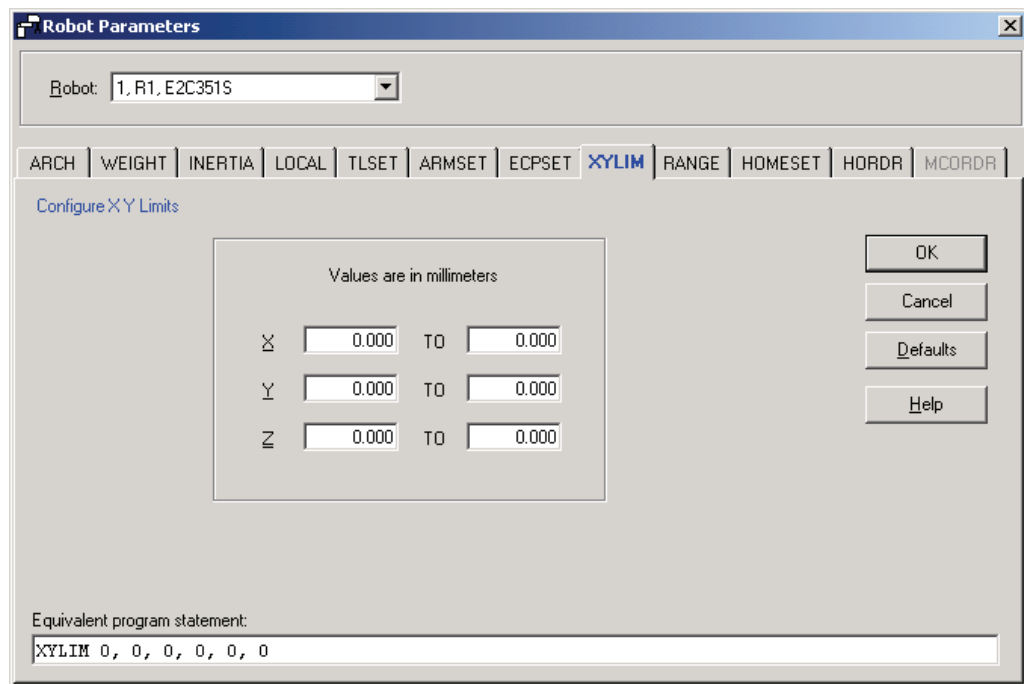
For more information on ECP, see the description for the ECPSet command.

Dialog Box Options

Option	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). For 6-axis robots only.
W	Rotation angle of the ECP about the X axis (yaw). For 6-axis robots only.
OK	Save changes.
Cancel	Cancel changes and exit.

XYLim Tab: Robot Parameters

This tab allows you to configure limits for X, Y and Z motion in the robot envelope.

**Dialog Box Options****Option Description**

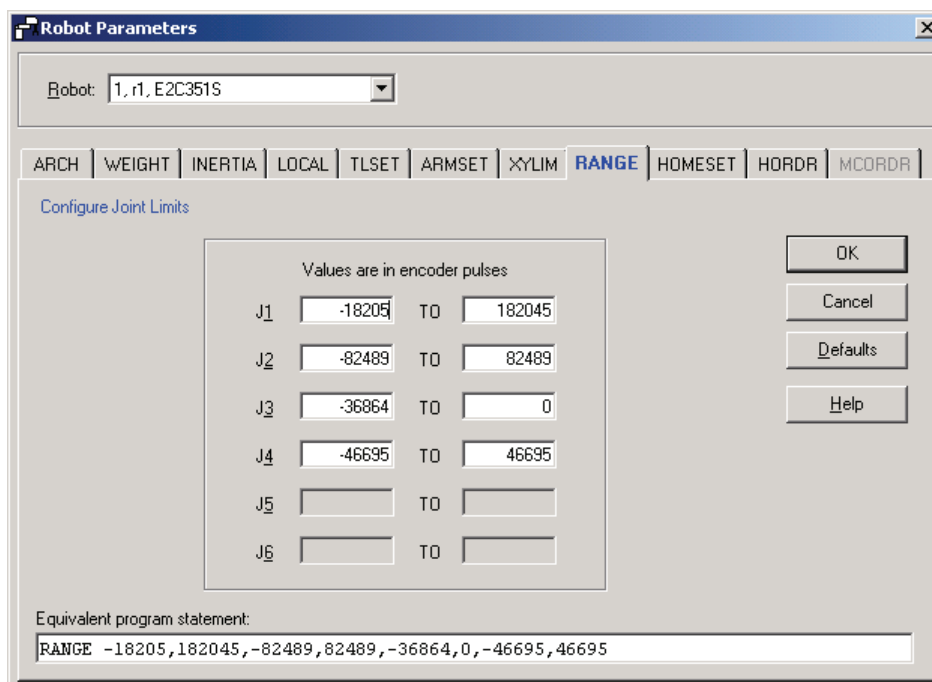
X, Y, Z Type in the minimum and maximum X, Y, and Z limit values. Setting both values to zero disables the limits.

OK Save changes.

Cancel Cancels changes and exits.

Range Tab: Robot Parameters

This tab allows you to configure the robot joint limits.

**Dialog Box Options**

Option	Description
--------	-------------

Range	Type in the minimum and maximum encoder pulse values for each joint.
OK	Save changes.
Cancel	Cancels changes and exits.

HomeSet Tab: Robot Parameters

HomeSet allows you to configure the optional user home position used by the Home command.

When you select the HomeSet tab, the current HomeSet position is read from the robot controller and displayed in the text boxes. If the HomeSet position has never been defined, then the text boxes will be blank.

To define the HomeSet position, you can enter an encoder position value for each of the four robot joints in the text boxes, or you can click on the **Read Current Position** button to read the current encoder position values.

The screenshot shows the 'Robot Parameters' dialog box with the 'HOMESET' tab selected. The 'Robot' dropdown menu is set to '1, r1, E2C351S'. The 'HOMESET' tab is active, displaying the text 'Configure joint locations for HOME command'. A central box contains the text 'Values are in encoder pulses' and six input fields for joints J1, J2, J3, J4, J5, and J6, each with a '0' entered. To the left of this box is a 'Read Current Position' button. To the right are 'OK', 'Cancel', and 'Help' buttons. At the bottom, a text area labeled 'Equivalent program statement:' contains the text 'HOMESET 0, 0, 0, 0'.

Robot
1, r1, E2C351S

ARCH | WEIGHT | INERTIA | LOCAL | TLSET | ARMSET | XYLM | RANGE | **HOMESET** | HORDR | MCORDR

Configure joint locations for HOME command

Read Current Position

Values are in encoder pulses

J1	J2	J3	J4	J5	J6
0	0	0	0		

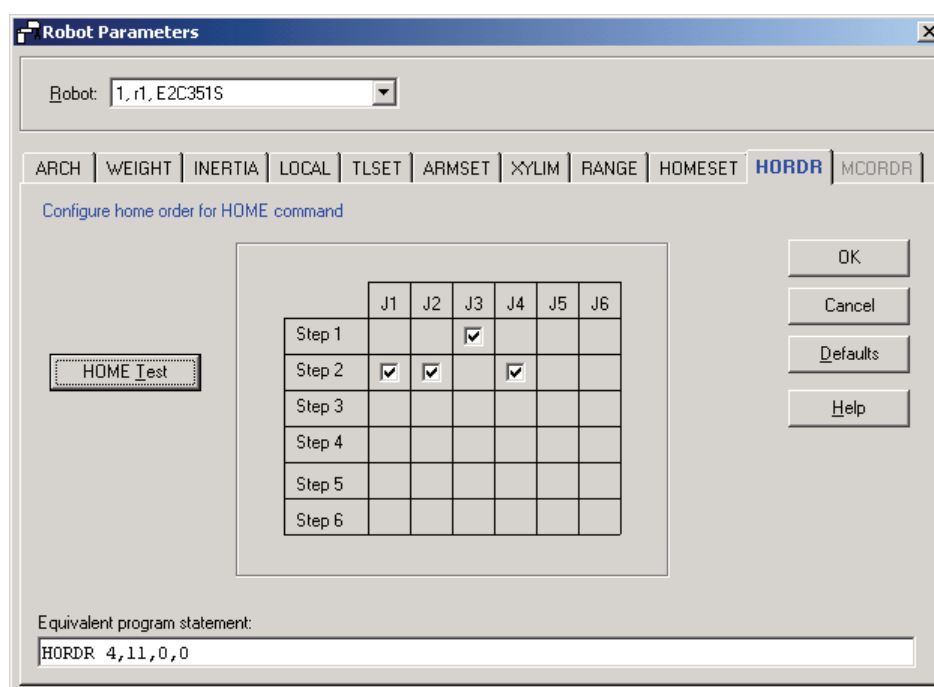
OK
Cancel
Help

Equivalent program statement:
HOMESET 0, 0, 0, 0

Hordr Tab: Robot Parameters

This tab lets you configure Hordr (home order) for the robot.

When the tab is first selected, the current settings in the robot controller are read and displayed.

**Changing home order**

The tab contains a grid of check boxes that allow you to configure the order in which the robot joints go to their home positions.

Each row of the grid is a home step and each column is a robot joint. One home step can include zero, one, or more joints. When a step contains more than one joint, then all of the joints in the step will go to home simultaneously.

To change which step a joint homes in, "uncheck" the box for the joint you want to change. Then check the new step for the joint.

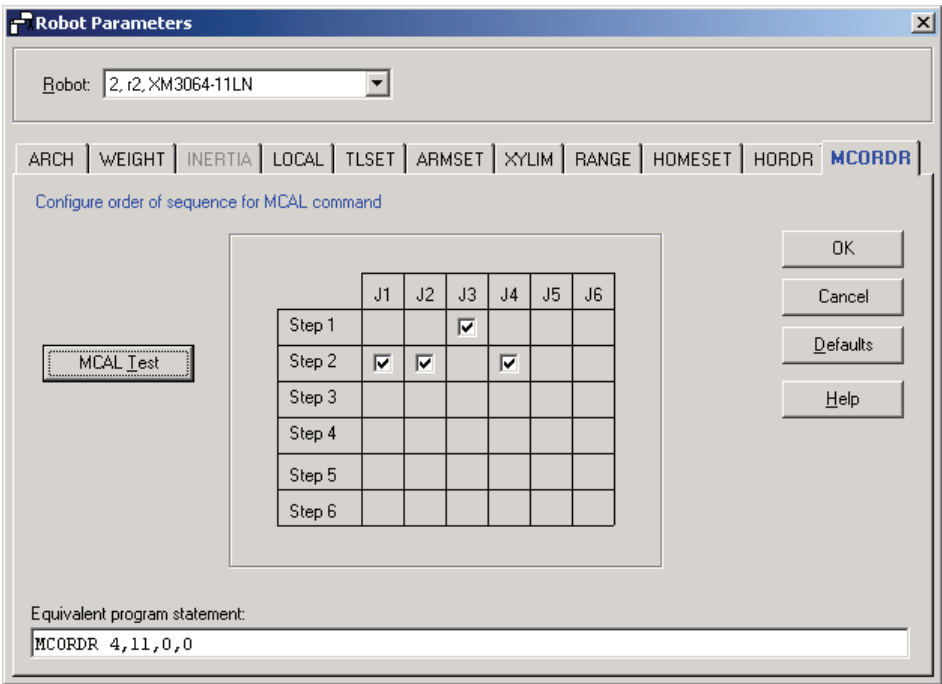
Tab options

Option	Description
Defaults	Press the default button to display factory default home order.
Equivalent statement	This read only text box shows the equivalent home order statement that can be used in a program or from the Monitor Window.
OK	Choose OK to save your changes in the robot controller system memory and on the PC with the current project. If the Z joint is not in the first home step by itself, you will get a warning.
Cancel	Exit without making changes.

MCordr Tab: Robot Parameters

This tab lets you configure MCordr (home order) for the MCal command.

When the tab is first selected, the current settings in the robot controller are read and displayed.



Changing MCal order

The tab contains a grid of check boxes that allow you to configure the order in which the robot joints go to their MCal positions.

Each row of the grid is an MCal step and each column is a robot joint. One MCal step can include zero, one, or more joints. When a step contains more than one joint, then all of the joints in the step will calibrate simultaneously.

To change which step a joint calibrates in, "uncheck" the box for the joint you want to change. Then check the new step for the joint.

Tab options

Option	Description
Defaults	Press the restore default button to display factory default MCal order.
Equivalent statement	This read only text box shows the equivalent MCal order statement that can be used in a program or from the Monitor Window.
OK	Choose OK to save your changes in the robot controller system memory and on the PC with the current project. If the Z joint is not in the first MCal step by itself, you will get a warning.
Cancel	Exit without making changes.

Save Command (Project Menu)

This command saves all currently open program and point files in the current project that need to be saved. This menu selection is grayed out if nothing needs to be saved.

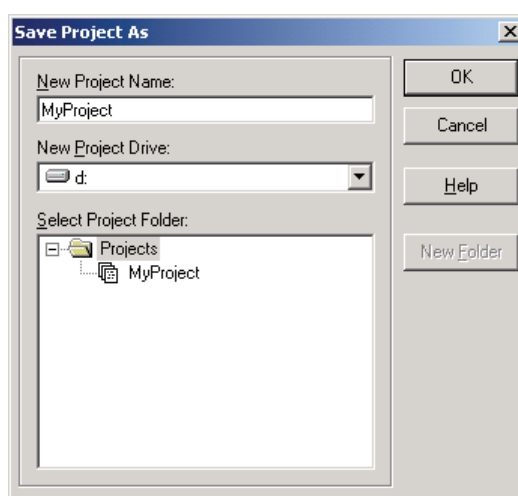
Shortcuts

Toolbar: 

Tip: It's a good idea to save frequently while you are editing programs or points. Just click the disk button on the toolbar to save all of your files.

Save As Command (Project Menu)

Saves all files in the current project to a new drive and/or project name. The current project will be preserved.

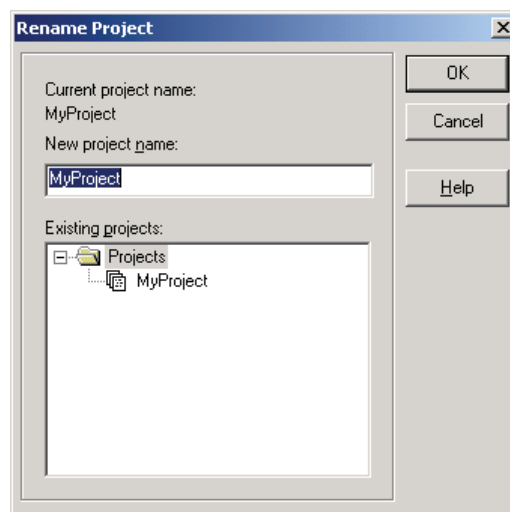


Dialog Box Options

Option	Description
New Project Name	Type in a new name for the project. The name can include alphanumeric characters along with underscores. The maximum number of characters is eight. You can use the same name as the current project if you select a drive that is not the same as the current project's drive.
New Project Drive	Drive for the new project location.
OK	Commit the new name and drive.
Cancel	Cancel the operation.

Rename Command (Project Menu)

Renames the current project. Before rename, all open point files must be closed. You will be prompted to close the files before the Rename Project dialog box appears.

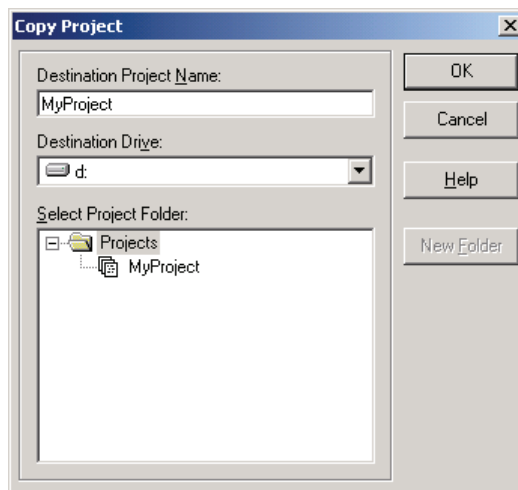


Dialog Box Options

Option	Description
New Project Name	Type in a new name for the project. The name can include alphanumeric characters along with underscores.
Existing projects	This list box shows other projects on the selected drive. The new name you choose cannot be one of the names in this list.
OK	Renames the project.
Cancel	Cancels the operation.

Copy Command (Project Menu)

Copies all files in the current project to a specified drive and project name. You can use the current project name for the destination name if you select a new drive. You can also specify a new name for the destination project.

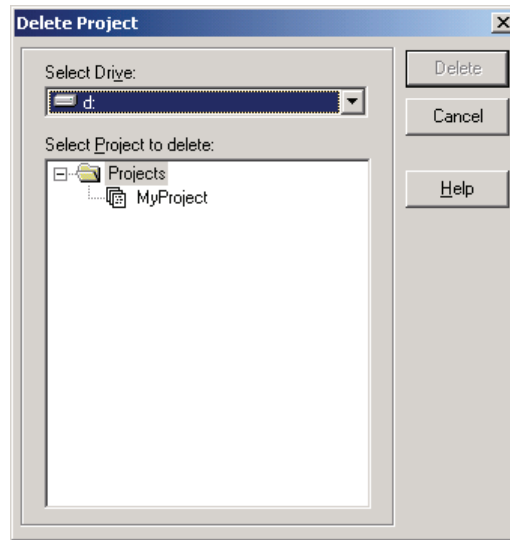


Dialog Box Options

Option	Description
Destination Project Name	Type in a name for the new copy of the project. The name can include alphanumeric characters along with underscores. The maximum number of characters is eight. You can use the same name as the current project if you select a drive that is not the same as the current project's drive.
Destination Drive	Drive for the project copy.
OK	Start the copy process.
Cancel	Cancel the operation.

Delete Command (Project Menu)

This command deletes an entire project. All files in the project directory will be destroyed.



Dialog Box Options

Option	Description
Select Drive	Select drive for the project to delete.
Select Project	Select a project to delete from the list.
Delete	Delete the project. You will be prompted to confirm the operation.
Cancel	Cancel the operation.

Compile File Command (Project Menu)


This selection will compile the program in the currently active program source window.

The Status window displays the results of the compile.

The source file is sent to the robot controller and then compiled.

If there are any errors, they will be displayed in the Status Window.

Shortcuts

Toolbar: 

NOTE




This command is not required to build the project. It is useful for checking for compilation errors in one file before building an entire project.

Build Command (Project Menu)

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 program.

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.

Shortcuts

Toolbar: 

Keys: CTRL+B

Rebuild Command (Project Menu)

Rebuilds the entire current project. All program files are re-compiled and linked.

Run Menu


The EPSON RC+ Run Menu includes commands for running and debugging programs.

Start Command (Run Menu)

Opens the Run window to run a program in the current project. Before opening the Run window, you are prompted to save all files if there are any unsaved files. Then the project is built, if required. If there are any errors during build, the Run window will not be opened.

After the Run window is open, you must click on the Start button to initialize program execution.

Shortcuts

Toolbar: 

Keys: F5

Operator Window Command (Run Menu)

Opens the Operator Window to run the current project. Before opening the Operator window, you are prompted to save all files if there are any unsaved files. Then the project is built, if required. If there are any errors during build, the Operator Window will not be opened.

If the project is ready to run (last build was successful), then EPSON RC+ will change to an icon and the Operator Window will be opened.

For more information, see the Operator Window section in the chapter *Building Applications in SPEL⁺*.

Shortcuts

Keys: SHIFT+F5

Control Device Command (Run Menu)

This dialog allows you to set which device can execute commands, such as starting tasks, motor on/off, etc.



When you want to use remote control in Program / Debug mode, you must set the control device to Remote. See Using Remote Control for details. To debug programs that use the OP500RC Operator Pendant, set the control device to OP.

If the Run window is open, its control buttons will change according to the control device setting. When the control device is set to Remote, the Run window Start button will change the Enable Remote. This allows you to debug remote control. When the control device is set to OP, the Run window Start button will change to Enable OP.

When the control device is Remote or OP, the Operator Window buttons are disabled and remote control or OP commands are accepted.

The setting is preserved between sessions.

Shortcuts

Toolbar: 

Step Into Command (Run Menu)

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

Shortcuts

Toolbar: 

Keys: F10, F8

Step Over Command (Run Menu)

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

Shortcuts


Toolbar: 

Keys: F11, Shift+F8

Walk Command (Run Menu)

Execute lines until after the next motion command or output command, depending on the *Walk stops for output commands* preference on the SPEL Options tab in Setup | System Configuration.

Shortcuts


Toolbar: 

Keys: F12

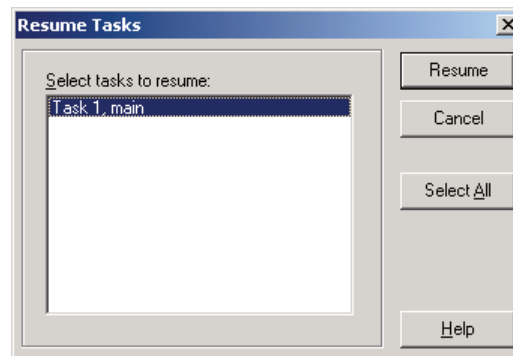
Resume Command (Run Menu)

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 is in halt mode.

Shortcuts

Toolbar: 

Keys: F7




Dialog Box Options

Option	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.

Abort All Command (Run Menu)

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

Shortcuts

Toolbar: 

Toggle Breakpoint Command (Run Menu)

Sets the current line as a breakpoint or returns the current line to normal.

Shortcuts

Toolbar: 

Keys: F9

Clear All Breakpoints Command (Run Menu)

Clears all breakpoints.

Shortcuts

Keys: Ctrl+Shift+F9


Display Variables Command (Run Menu)

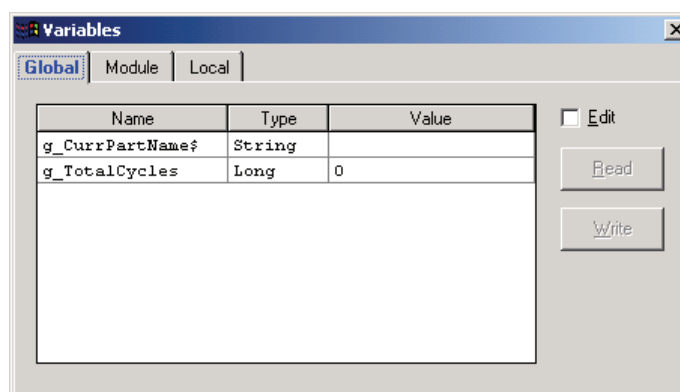
Displays a dialog box that shows the values for all variables in robot controller memory and all system variables.

You can change the values for each of the variables displayed by typing the new value in the Value column and then click the **Write** button. As you type in new values, the text color changes to red, indicating that the value is new and as not been written.

When an array is displayed, the first element is shown. You can change which element to view by typing in the desired array subscript and then clicking the **Read** button.

Shortcuts

Toolbar: 

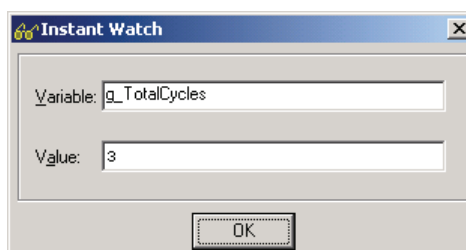


Instant Watch Command (Run Menu)

Displays a dialog box that shows the value of the variable at the cursor location.

Shortcuts

Toolbar: 



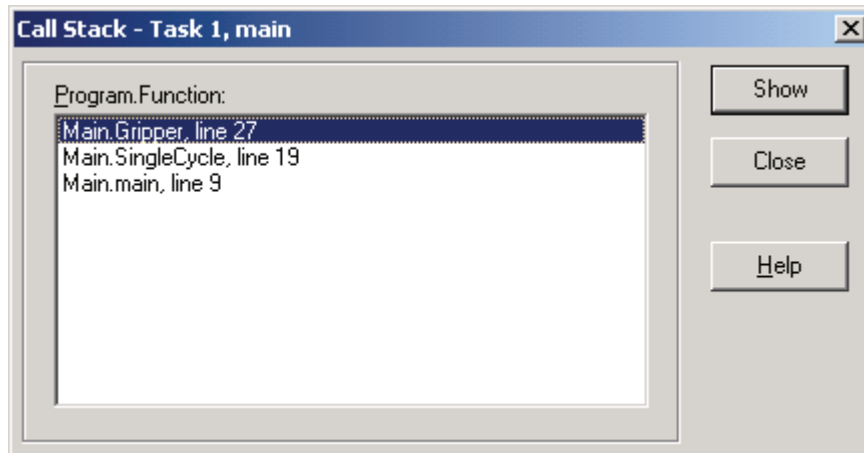
Open Instant Watch

To open Instant Watch, set a breakpoint in a program or halt a program from the Task Manager. Then click on the Instant Watch toolbar button. If the caret in the current program editor window was on a variable name, that variable will be displayed along with its current value. Otherwise, a message will be displayed stating that there was no variable to display.

You must close the Instant Watch dialog before you can step or resume the program.

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 bottom function is the task function.

Each row in the list shows the program, function, and line number. The line number displayed is the physical line number in the program file, not the user line number.

You can view the code for any of the function calls in the list by selecting a function, then clicking [Show](#). The program window for the function you selected is then displayed and the line of the function call is marked by a yellow arrow in the editor left margin.

Shortcuts

Toolbar: 

Tools Menu

EPSON RC+ has several GUI tools to aid in 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:

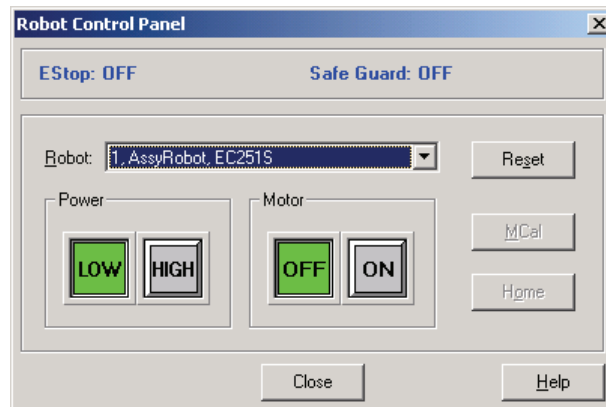
- **Robot Control Panel**
Controls motor power and selects current robot.
- **Jog and Teach**
Jog the current robot and teach points.
- **Task Manager**
Monitor and control task execution.
- **I/O Label Editor**
Define and edit I/O labels.
- **I/O Monitor**
Monitor I/O status.
- **Memory I/O Monitor**
Monitor memory bit status.
- **Monitor Window**
Execute SPEL commands directly.
- **Maintenance**
Perform maintenance operations.
- **Force Monitor**
Monitor force sensors.
- **Audit Viewer**
View security audit information.
- **User Error Editor**
Define and edit user errors.
- **Conveyor Tracking**
Define, edit, and calibrate conveyors.

Robot Control Panel Command (Tools Menu)

The robot control panel contains buttons for basic robot operations, such as turning motors on/off or homing the robot. While the panel is displayed, it monitors ESTOP and SAFEGUARD status and displays their status.

Shortcuts

Toolbar: 



Status Indicators

Indicator	Description
ESTOP	Indicates if ESTOP has occurred. Once an ESTOP occurs, you must execute a Reset to clear it.
SAFEGUARD	Indicates whether the SAFEGUARD input is on or off.

Controls

Control	Description
Robot drop down list	Allows you to select the current robot for subsequent operations. This list contains all of the robots on the system.
Reset button	Resets the robot servo system and ESTOP condition.
Close button	Closes the robot control panel.
Motor On button	Turns on all robot motors for the current robot. You will be prompted to confirm this operation.
Motor Off button	Turns off all robot motors for the current robot.
MCal button	Mechanically homes the robot. This command must be performed after motors are turned on for the first time since the controller was powered on. This is only for robots with incremental encoders.
Home button	Sends the robot to the position specified with the HomeSet command.
Power Low button	Puts servo system in low power mode.
Power High button	Puts servo system in high power mode.

Jog and Teach Command (Tools Menu)

Overview

The Jog & Teach window is primarily used for teaching robot points.

You can configure how the Jog & Teach window can be viewed in the development environment from the Setup | Preferences | Jog tab.

Jog & Teach MDI window The Jog & Teach window is displayed as a child window along with the other child windows inside the EPSON RC+ development environment parent window.

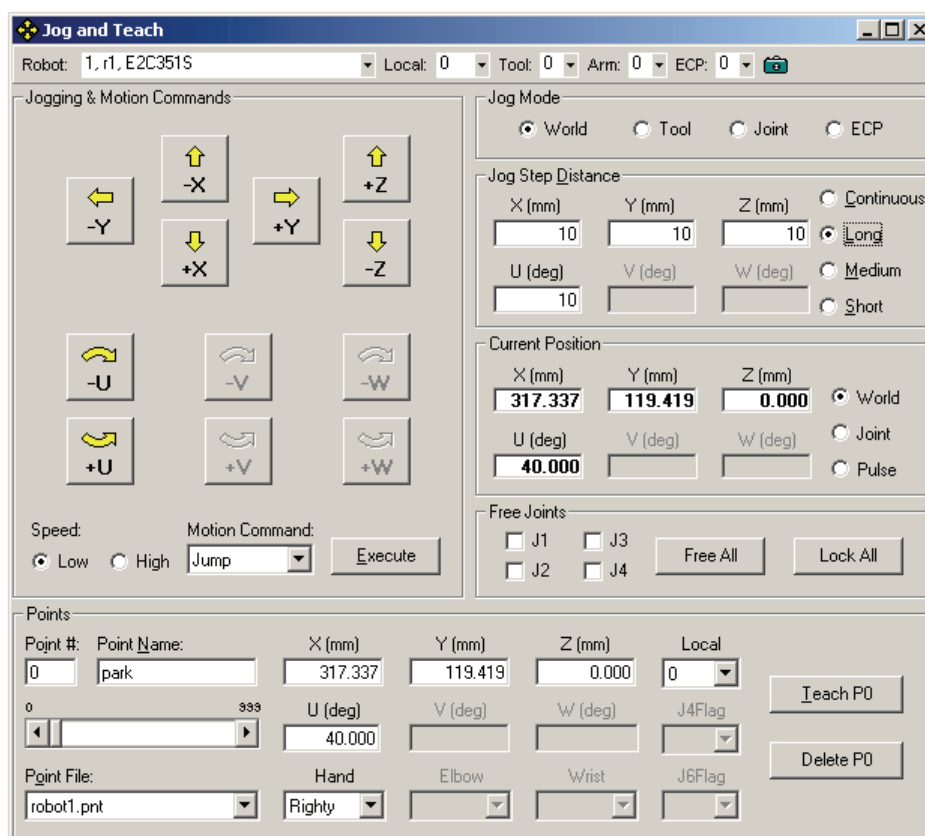
Jog & Teach Dialog The Jog & Teach window is displayed as a modal dialog which is displayed in the foreground over the development environment parent window.

Before you can open the Jog & Teach window, you must have a project open and there must also be at least one point file in the project for each robot on the system. A message will be displayed if you attempt to open Jog & Teach and one or more robot point files need to be created. You can add point files with the Edit command (Project menu).

You can jog the robot in World, Tool, or Joint modes. You can also execute motion commands supported by the selected robot.

Shortcuts

Toolbar: 



Jog Controls

The Jog & Teach window contains several controls, described below.

Robot

This drop down list is used to select the current robot.

Local

This drop down list is used to select the current Local for jogging and 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

This drop down list is used to select the current Tool for jogging and teaching. Only Tools that have been defined are shown in the list.

Arm

This drop down list is used to select the current Arm for jogging and teaching. Only Arms that have been defined are shown in the list. Arms are not used with 6-axis robots.

ECP

This drop down list is used to select the current ECP for jogging. Only ECPs that have been defined are shown in the list. ECPs are only allowed if the External Control Point option has been activated.

Vision

The Vision tool bar button is available when Vision Guide is installed. Click this button to toggle the video display on the Jog & Teach window.

Jogging & Motion Commands

This group contains buttons that are used to jog the current robot throughout the work envelope.

The jog buttons can be actuated with the mouse or keyboard. You can also execute motion commands. The commands you can execute depend on the robot type.

You can jog one step at a time by clicking on a button and releasing. Or you can step continuously by holding the button down. See *How to jog robot* for details

You can change the orientation of the buttons by clicking on the Preferences button on the toolbar.

The jog buttons that are displayed depend on the Jog mode. For World, Tool, and ECP jogging, the X, Y, Z, U, V, W buttons appear (V and W are enabled only for 6-Axis robots). For Joint jogging, the joint buttons appear, labeled J1 - J6.

The X, Y, and Z buttons jog the robot in Cartesian space.

The U buttons rotate the tool coordinate system about the Z axis. This is also known as *roll*.

For 6-Axis robots, the V buttons rotate the tool coordinate system about the Y axis. This is also known as *pitch*. The W buttons rotate the tool coordinate system about the X axis. This is also known as *yaw*.

The motion command buttons can be disabled by selecting Preferences from the toolbar. You can also specify that a confirmation message is displayed each time a motion operation is executed.

Speed

The speed for jogging and motion commands can be changed by selecting Low or High. When the Jog & Teach window is open, the speed is set to Low. Jogging is always in low power mode. The speeds and accelerations associated with the jog speed settings are shown below.

Jog 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 mm/sec ²	40 mm/sec ²
	Continuous Joint	*	10 mm/sec ²	20 mm/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 mm/sec ²	40 mm/sec ²
	Continuous Joint	*	10 mm/sec ²	20 mm/sec ²
	Step	Default PTP speed	Default PTP acceleration	Default PTP deceleration

*Continuous joint speed and acceleration depends on robot model

Jog Distance

This group contains text boxes that are used to specify the distance that each axis moves when its corresponding jog button is pressed. There are radio buttons for selecting Continuous, Long, Medium, and Short jog distances. When Continuous is selected, the robot is jogged in continuous mode. When Long, Medium, or Short are selected, the robot is jogged in step mode.

To change a jog distance, first select the distance to be changed, then type in the new value. Short jog distance can be from 0 to 10 mm, Medium distance can be from 0 to 100 mm, and Long distance can be any value. You are prevented from typing in values that are out of range.

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

Current Position

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, and **Pulse** displays the current encoder pulse count for each joint.

Jog Mode

This group contains four option buttons for selecting the jog mode.

- World Mode:** Jogs the robot along the X, Y, Z axes in the current local, tool, arm, and ECP. For robots with 4 DOF, you can also jog U (roll). For robots with 6 DOF, you can jog U (roll), V (pitch), and W (yaw). This is the default setting.
- Tool Mode:** Jogs the robot in the coordinate system defined by the current tool.
- Joint Mode:** Jogs each joint of the robot. A separate set of jog buttons will appear when using joint mode when using non-Cartesian robots.
- ECP Mode:** Jogs the robot along the axes of the coordinate system defined by the current external control point. Coordinates are World coordinates.

Free Joints

Contains check boxes for each active robot joint. When you free a joint, you can move it by hand to teach a location. The Free All button is used to free all axes. The Lock All button is used to lock all axes. This group is not available for 6-Axis robots.

Current Arm Orientation

The current Hand, Elbow, Wrist, J4Flag, and J6Flag values are displayed. This group is only displayed for 6-Axis robots.

Points

This group shows the current point number and name for Teach and motion commands. You can either type in the desired point number, or you can use the slider to select it.

For SCARA robot points, you can specify:

Hand: Righty (default) or **Lefty**

For 6-Axis robot points, you can specify:

Hand: Righty (default) or **Lefty**

Elbow: Above (default) or **Below**

Wrist: No Flip (default) or **Flip**.

J4Flag: 0 (default) or **1**.

J6 Flag: 0 (default) or **1 - 127**.

There are text boxes for the coordinates of the current point. You can type in or edit the coordinates of a point. Changes are made to the points loaded in memory.

Use the **Teach** button to read the current robot position into the coordinates of the current point.

See *How to teach points* for more information.

How to jog

In the upper left hand corner of the jog screen, you will see a control group called Jogging and Motion Commands that contains jog buttons. In the World, 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. In continuous mode, when a jog button held down, the robot moves continuously using linear interpolated motion.



Jog motion in step mode is PTP (point to point) motion. It is difficult to predict jog motion trajectory. 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.



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, before you can continue jogging.

To jog

Select the jog mode: World, Tool, Joint, or ECP.

Check the jog speed by looking at the value for Speed. If necessary, change it.

Check jog distance by looking at the values in the **Jog Distance** control group. Select **Continuous**, **Long**, **Medium**, and **Short** jog distance. You can type in the desired jog distance when Continuous is not selected.

Click on one of the jog buttons with the left mouse button. If you hold the mouse button down, the robot will continue to jog.

You can also use the keyboard to jog the robot. See Jog and Teach Keys.

Tip: You can change the orientation of the jog buttons for each robot by selecting Preferences: Jogging tab from the Setup menu or the Preferences button on the Jog & Teach toolbar. This will allow you to align your PC monitor with the orientation of the robot.

How to teach points

Teaching a robot point means that you are defining a location stored in memory for the robot to move to.

Follow these steps to teach points from the Jog and Teach screen:

1. Select the point file you are teaching points for by using the **Point File Name** drop down list box in the **Points** Control group in the lower left corner of the screen.
2. Select the point number you want to teach in the **Points** control group in the lower left corner of the screen. You can either type in a point number in the point number text box, or you can use the slider.
3. If desired, you can type a name for the current point in the **Point Name** text box. This is optional, but recommended. Try to use a descriptive name so that its easier to recall what the point is used for or where it is. You can also change point names by using the Point Editor.
4. Jog the robot to the desired position or free some or all axes and manually move the robot into position.
5. Click on the **Teach** button. This will retrieve the robot's current position and display it in the coordinates for the point.
6. As an alternative to clicking the **Teach** button, in the **Points** control group you can type in the coordinates of the point.

Saving or aborting your work

Jog & Teach MDI Child

When Jog & Teach is used as an MDI child, use the File menu to select Save or Restore for the current point file. You can also execute Project | Save or click the Save all files toolbar button.

Jog & Teach Dialog

When you close the jog dialog, you will be prompted if you want to save your changes. Answer Yes to make your changes permanent or No to abort all changes.

You can also click the **Save** button to save the changes made without closing the jog screen.

Teach Mode

You can jog and move the robot at slow speed with the safeguard open by using Teach mode. If the OP500RC pendant is enabled, Teach mode can only be used from the pendant. Otherwise, Teach mode can only be used when the Jog & Teach dialog is open.

To enable Teach mode, you must wire a Teach Mode key switch and a 3 position dead man switch to the Option Device connector on the EPSON RC+ PC, or purchase the OP500RC pendant.

When Teach mode is ON, you must engage the Enable (dead man) circuit to move the robot or turn outputs on and off. If you attempt to jog or change an output Enable disengaged, a message box will be displayed.

Task Manager Command (Tools Menu)

Description

The Task Manager dialog box allows you to Halt (suspend), resume, and Quit (abort) tasks.

To open the Task Manager

Select Task Manager from the Tools menu.

Or

Click on the  button on the toolbar.

Operation

The Task Manager is used for suspending, resuming, stepping, and stopping tasks.

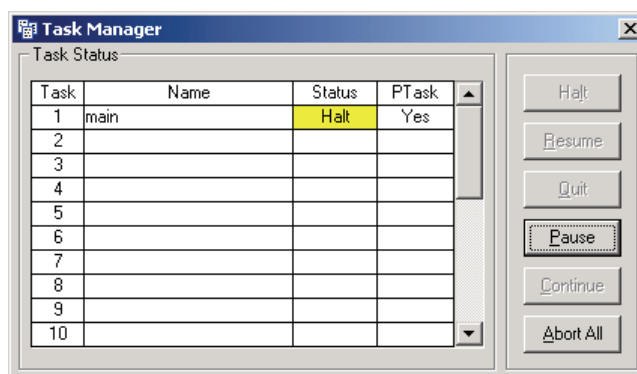
When Task Manager is started, you will see a grid containing status information for 32 tasks. There are four items shown for each task: Task Number, Task Name, Status, and Type.

Task Number Number of task from 1 to 32.

Task Name Name of the function that was started as a task.

Status Current task status: Run, Halt, Pause, Quit, Safety, Step.

PTask The PTask field indicates whether a task will suspend when a system pause occurs. If PTask is No, then the task was started with the NoPause option and will not suspend when a pause occurs.



To Halt, Step, Walk, and Resume a task

Use the mouse to click on the desired task in the task list. You can click on any field to select a task. The current task status must be Run before it can be halted. The Halt button will become active after you have selected the task.

After a task has been halted, the source code will be displayed and the next step will be indicated. You can either click on the **Resume** button to resume execution, or executed Step Into, Step Over, or Walk from the Run menu.

To Pause and Continue tasks

Pause allows you to "suspend" all tasks where pause is enabled.

Click on the **Pause** button to pause tasks for which pause is enabled. The robot will decelerate to a stop immediately.

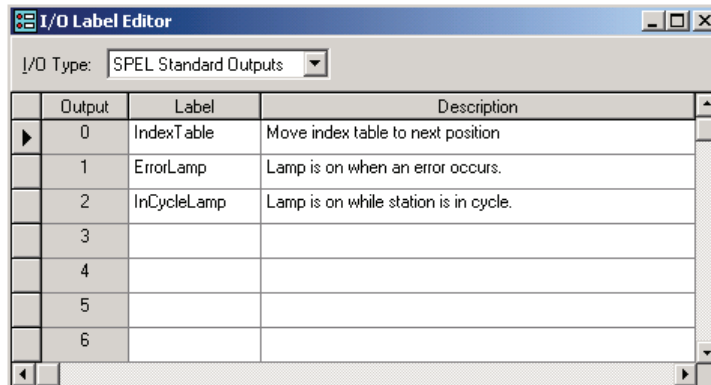
After executing Pause, click on **Continue** to resume all suspended tasks.

Dialog Box Options

Option	Description
Halt button	Suspends the selected task. This temporarily stops the task so that it can be started again with the Resume button. Halt can only be executed when a task is running (status is Run). When Halt is executed, the Resume button will be enabled. If a motion command is executing with Halt occurs, the motion will be completed before the task reaches the Halt state.
Resume button	After one or more tasks have been suspended with the Halt button, clicking on Resume will make the halted tasks continue where they left off.
Quit button	This 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.
Pause button	This pauses all tasks that have pause enabled. After pause, you must use either Continue or Abort All .
Continue button	This button continues all tasks that were paused with the Pause button.
Abort All button	This button aborts all tasks.

I/O Label Editor Command (Tools Menu)

The I/O label editor lets you define names for inputs, outputs, and memory I/O for each project. The labels can be used in your programs, from the Monitor window, or in macros.



The I/O Label Spreadsheet

When you select I/O Label Editor from the Tools menu, a window opens that contains a spreadsheet editor.

The first column heading shows the input type you are editing, which can be selected from the drop down list located at the top left corner of the window: SPEL Standard Inputs, SPEL Standard Outputs, SPEL Memory I/O, Ethernet I/O, and Fieldbus I/O.

The second column contains the labels for each bit. You can type in up to 16 characters for a label. Label characters can be alphanumeric or underscore.

For SPEL Standard Inputs, SPEL Standard Outputs, and SPEL Memory I/O, the third column contains an optional description that you can enter which describes more about the I/O point.

For Ethernet I/O, the third column shows the type of I/O point and the fourth column contains the description.

For Fieldbus I/O, the first columns shows the device ID, the second column shows the device name, the third columns shows the bit number, the fourth column shows the label, and the fifth column shows the description.


If you add a description to an I/O point, then the first line of the description will be displayed as a ToolTip on the I/O monitor.

To add or edit a label

Select the type of I/O you want to label. After you select the I/O type, the spreadsheet will be redrawn to display the labels for that type. The number of rows in the spreadsheet equals the number of bits available for the type you have selected. For memory I/O labels, do not add the "\$" prefix when you create the label. This prefix must be used in a program statement when you use the label.

Use the mouse to scroll through the spreadsheet and put the cursor in the Name field next to the bit number that you want to add a label to. Type in the name, which can be up to 16 alphanumeric characters without any spaces.

After adding labels, save the changes by executing Save from the File Menu or by clicking on the

Save Project  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.

To cut and paste labels and descriptions

You can cut and paste labels and descriptions by selecting with the mouse, then executing Copy, Cut, and Paste from the 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 on the left and execute either the Cut or Copy command from the Edit menu. When selecting multiple rows, use the shift or control key.
2. Select the I/O type, if required. For example, you may want to cut labels from Inputs and put them in Ethernet I/O.
3. Select the row where you want to start the paste by clicking the row selector on the left of the row.
4. Execute the Paste command from the Edit menu.

I/O Monitor Command (Tools Menu)

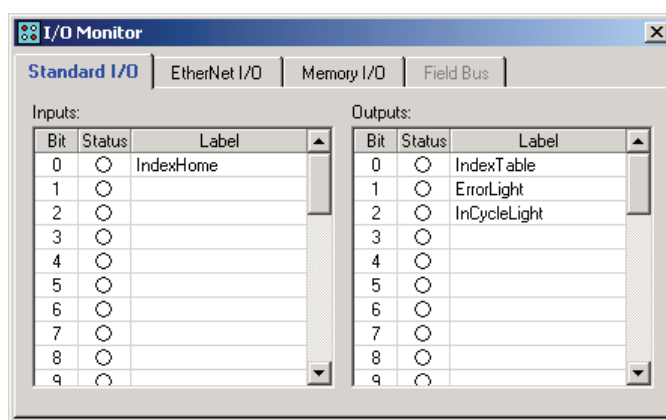
The I/O Monitor dialog lets you monitor all system inputs and outputs. The inputs are displayed on the left side of the dialog and the outputs on the right. Labels that have been defined using the I/O Label Editor are displayed next to each bit.

After the dialog has been opened, the input and output status for the currently I/O page is constantly updated.

The I/O monitor will always be displayed on top of other child windows, such as program windows and point windows.

You can also turn outputs on and off by double clicking on the output LEDs.


If a description has been entered for an I/O channel in the I/O Label Editor, then a ToolTip will be displayed when the mouse pointer is over the channel. The ToolTip shows the first line of the description.



To open the I/O Monitor

Use one of the following methods to open the I/O monitor:

Select I/O Monitor from the Tools menu.

Click on the  tool bar button.
Type CTRL + I.

Using the I/O Monitor

Select the tab for the type of I/O you want to monitor.

Scroll through the grids to locate the desired inputs or outputs to monitor.

Input status is automatically updated.

To turn an output off or on, double click on the LED graphic for the desired output.

You can resize the I/O monitor in the vertical direction to show more points. Move your mouse pointer to the lower right corner of the window to get a size handle, then click down and drag the window down or up to the desired size.

Monitor Window Command (Tools Menu)

The monitor window emulates an ASCII terminal that communicates directly with the robot controller. You can execute SPEL immediate commands from this window and view the results.

There are also some special commands required for setup of the robot controller that must be executed from the Monitor window. See Maintenance Commands.

To open the Monitor window

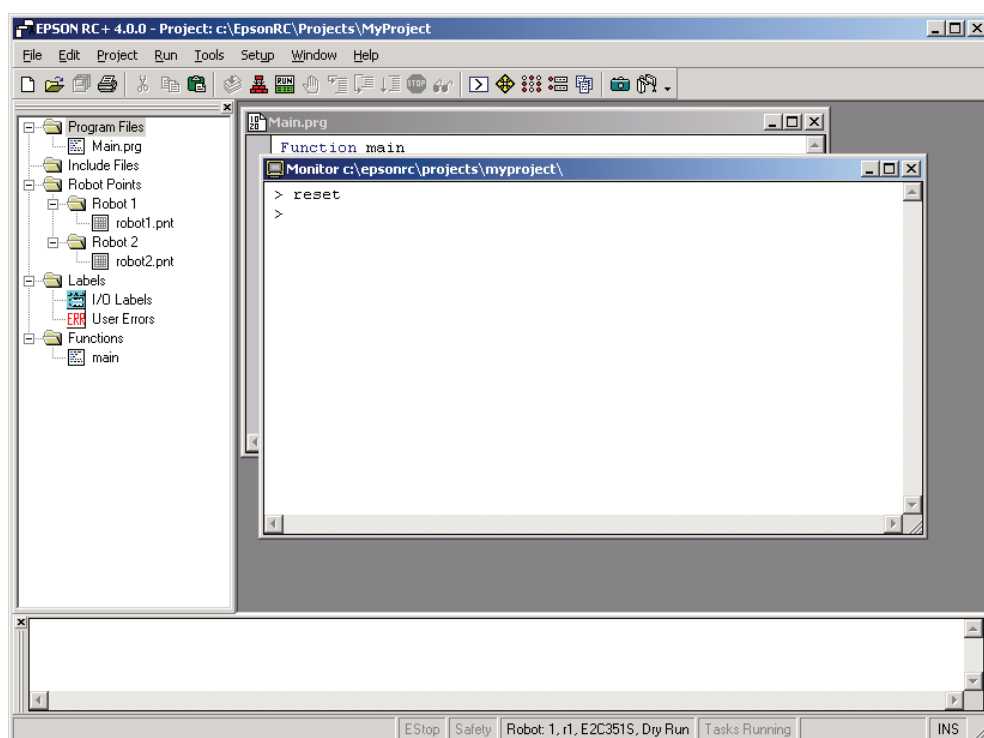
Select Monitor from the Tools menu

Or

Click on the  button on the toolbar.

Or

Type CTRL+M



To execute SPEL⁺ commands from the Monitor 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. The cursor can be anywhere on the line when you press Enter.
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.


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 Enter.

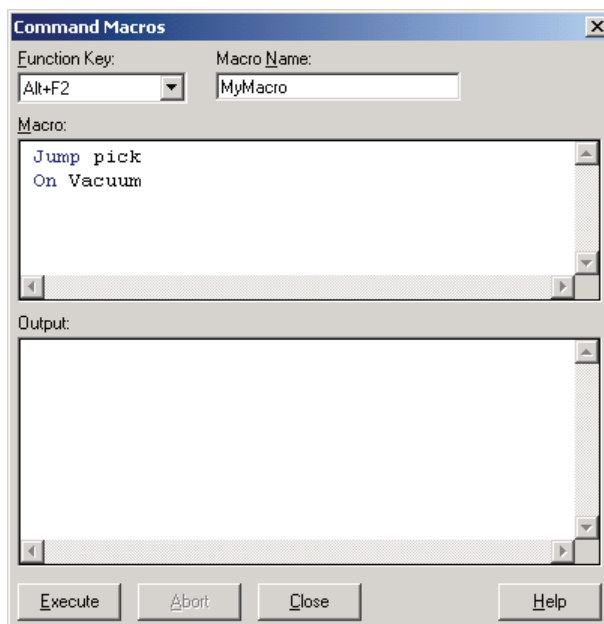
Monitor Window Keys

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+W	Re-display last command line after the prompt.
Ctrl+X	Execute Cut command. Cut current selection and put in Clipboard.
Ctrl+Z	Undo last change.
Ctrl+Home	Go to the top of the window.
Ctrl+End	Go to last prompt at end of the window.
?	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.

Macros Command (Tools Menu)

You can create SPEL command macros using the Macro Editor. Macros consist of one or more SPEL statements. 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.

Select Tools | Macros  to open the Command Macros dialog.



Type the macro statements in the Macro text box.

Click **Execute** to run the macro.

Click **Close** to close the dialog. You will be prompted to save the macros you have created or changed.

To open a macro and execute it, type Alt+function key to open the macro. Then click **Execute** to run it. Macros never execute by pressing the function key. The separate execute step is provided for safety, since macros can move the robot and control I/O.

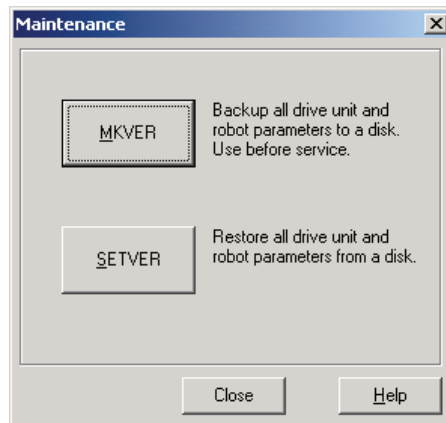
Maintenance Command (Tools Menu)

Select Maintenance from the Tools menu to open the Maintenance dialog.

From the Maintenance dialog, you can save and restore the complete system configuration using the MKVER and SETVER commands.


Before servicing the system, you should execute MKVER and store the system configuration on a floppy disk.

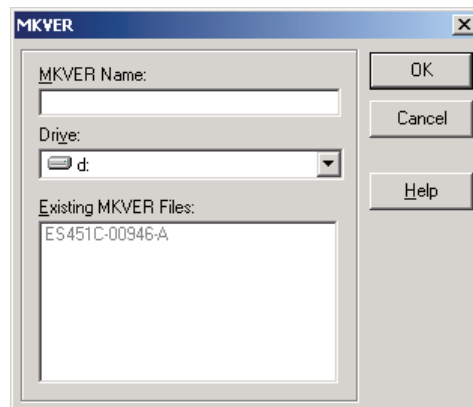
If required, you can use SETVER to restore previously stored data.



MKVER

Use MKVER to save system configuration.

1. Select Tools | Maintenance .
2. Click on the **MKVER** button to open the MKVER dialog.

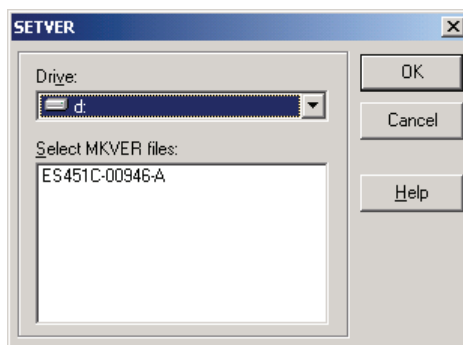


3. Type in a unique name to identify the system in the MKVER Name text box.
4. Select the disk drive where you want to save the information.
5. Click OK.

SETVER

To restore system configuration:

1. Select Tools | Maintenance.
2. Click on the SETVER button to open the SETVER dialog.



3. Select the drive where the information is stored.
4. Select the configuration from the MKVER files list box.
5. Click **OK** to restore the system information.




Restore the saved system configuration only for the same system.

SeikoRCS Compatibility: If MKVER files are found in the SeikoRCS directories on the same drive, then SeikoRCS files will be denoted by (SeikoRCS) after each name in the list. You can use SeikoRCS files on the EPSON RC+ system.

Force Monitor Command (Tools Menu)

This menu selection is available when the Force Sensing Option is enabled.

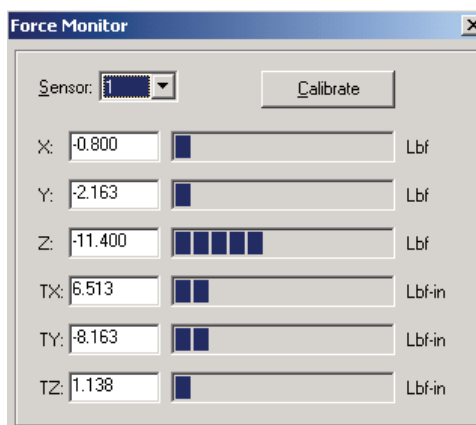
To open, select Tools | Force Monitor .

After the dialog has been opened, the status for all force sensing axes is constantly updated.

The Force Monitor will always be displayed on top of other child windows, such as program windows and point windows.

Use the sensor drop down list to select the sensor you want to monitor.

Click the Calibrate button to zero out all axes.



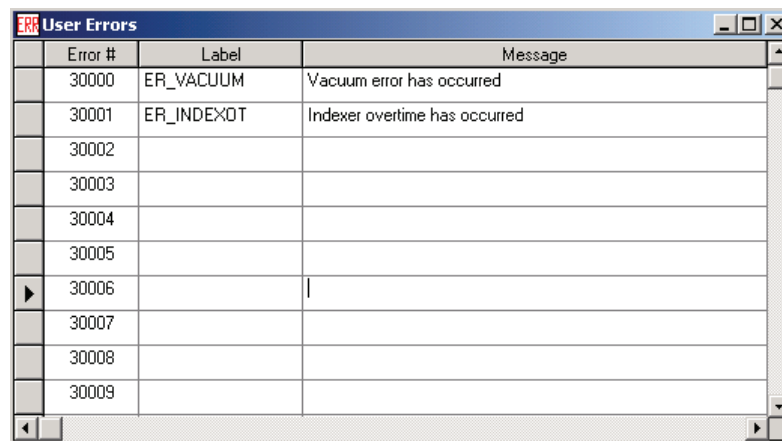
Audit Viewer Command (Tools Menu)

This menu selection is available when the Security Option is enabled. A dialog is opened showing session data for all EPSON RC+ users on the system or network.

See the chapter *Security Option* for details.

User Error Editor Command (Tools Menu)

The User Error Editor allows you to define user errors for the current project. If a user error is caught in an error handler, then the ErrMsg\$ function will return the error message defined in the editor. The label associated with an error can be used in any integer expression.



User error numbers are fixed from 30000 to 30999.

Labels can be up to 16 characters in length.

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.

Some user error examples:

Error #	Label	Message
30000	ER_VACUUM	Vacuum error has occurred.
30001	ER_INDEXOT	Indexer overtime has occurred.


In your program code, use the Error statement to generate a user error. For example:

```
On Vacuum
Wait Sw(VacOn), 1
If Tw = 1 Then
    Error ER_VACUUM
EndIf
```

The user error information is stored in the current project directory in a file called userErrors.txt.

You can use the Import command 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 Project  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.

Vision Command (Tools Menu)

This menu selection is available when the Vision Guide Option is enabled. When this command is executed, the Vision Guide Dialog is displayed.

See the Vision Guide manuals for details.

Conveyor Tracking Command (Tools Menu)


This menu selection is available when the Conveyor Tracking Option is enabled. A dialog is opened showing the conveyors for the current project.

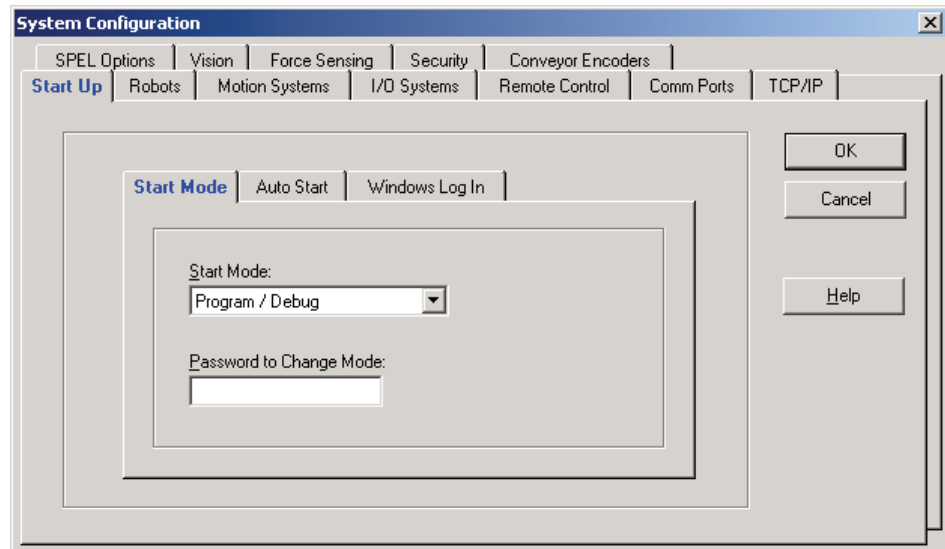
See the chapter *Conveyor Tracking* for details.

Setup Menu

System Configuration Command (Setup Menu)

The System Configuration command opens a dialog with several tabs. Each tab corresponds to a component of the system.

To open the System Configuration dialog, select Setup | System Configuration .




- **Start Up Tab**
The Start Up tab allows you to configure the start mode, operator auto start, and Windows log in. See the chapter *Operation* for details.
- **Robots Tab**
The Robots tab is where you add or delete robots on the current system. See the chapter *Robot Configuration* for details.
- **Motion Systems Tab**
This tab is used to configure the motion systems that drive the robots. See the chapter *Motion Systems* for details.
- **I/O Systems Tab**
The I/O Systems tab is where you setup up I/O hardware. See the chapter *I/O Systems* for details.
- **Remote Control Tab**
This tab is used to configure Remote Control I/O. See the chapter *Remote Control* for details.
- **Comm Ports Tab**
The Comm Ports tab is used to configure RS-232 communication ports. See the chapter *RS232 Communications* for details.
- **TCP/IP Tab**
This tab configures TCP/IP communications. See the chapter *TCP/IP Communications* for details.
- **SPEL Options Tab**
This tab is used to set SPEL Runtime options. See the chapter *SPEL Options* for details.

- **Vision Tab**
This tab is used to configure Vision Guide hardware.
- **Force Sensing Tab**
This tab is used to configure the Force Sensing option. See the chapter *Force Sensing* for details.
- **Security Tab**
This tab is used to configure the Security option. See the chapter *Security* for details.
- **Conveyor Encoders Tab**
This tab is used to add / delete encoders used for Conveyor Tracking. See the chapter *Conveyor Tracking* for details.

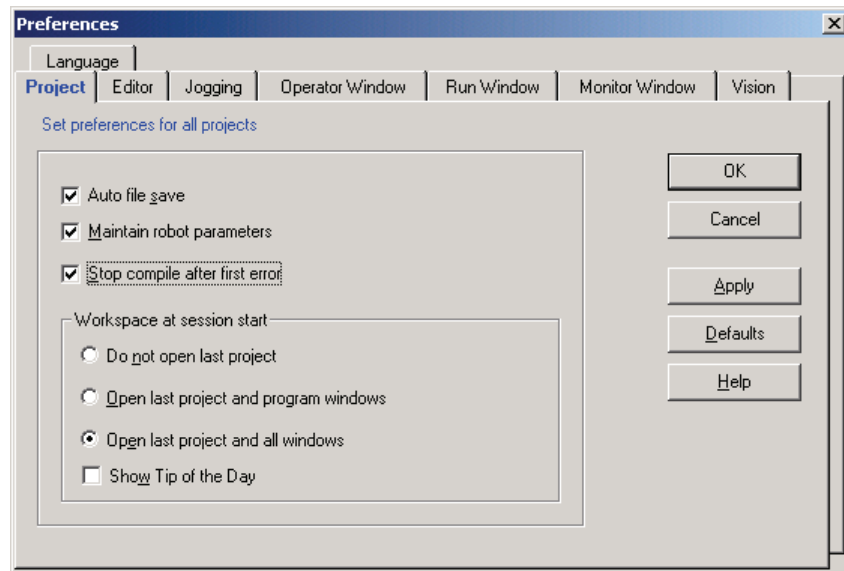
Preferences Command (Setup Menu)

The Preferences command opens a dialog that contains several tabs that are used to configure user preferences.

To open the Preferences dialog, select Setup | Preferences .

Project Preferences

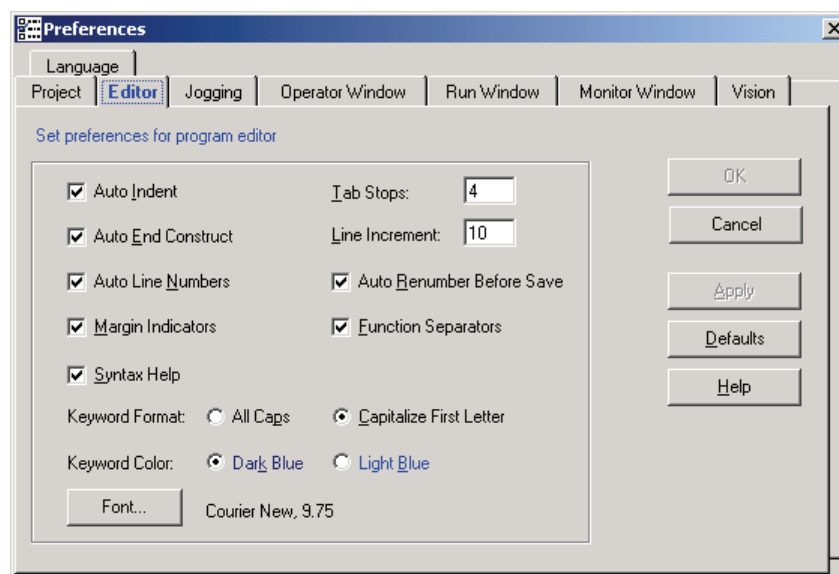
From this tab, you can configure your preferences for all projects.



Option	Description
Auto file save	Checking this box will cause EPSON RC+ 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. Default is On.
Maintain robot parameters	Checking this box will cause all robot parameters to be stored with projects and sent to the controller when a project is opened and built. Robot parameters include Hordr, MCordr, HomeSet, Weight, Local, TLSet, Range, XYLim, Arch, ArmSet, ECPSet. Default is Off.
Stop Compile after first error	Stops compile after first error occurs. This makes it easier to see the first error in the status window and allows you to fix one error at a time. Default is On.
Do not open last project	If this radio button is selected, the last project will not be opened when EPSON RC+ is started.
Open last project and program windows	If this radio button is selected, the last project will be opened and any program windows that were previously opened will be opened.
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.
Show Tip of the Day	If this check box is on, the Tip of the Day dialog will be displayed when EPSON RC+ is started.

Editor Preferences

This tab lets you configure your preferences for the program editor windows.



Option	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 If, Else, For, Select, Case and While statements. Default is On.
Auto End Construct	Check this box if you want EPSON RC+ to add the end construct statement for a loop construct. For example, if you enter a For statement, then a Next statement will be added automatically. Default is On.
Auto Line Numbers	Check this box if you want a line number to be inserted after carriage return is pressed. A line number will only be inserted if the current line has a line number on it. If you are not using line numbers, then numbers will not automatically be inserted. Default is On.
Auto Renumber Before Save	Check this box if you want programs to automatically be renumbered each time you save. If you are not using line numbers, then this option has no effect. Default is On.
Margin Indicators	Check this box to display a margin on the left side. This margin is used to indicate lines with breakpoints, current step line, current execution line. Default is On.
Function Separators	Check this box to display a line after each Fend statement. Default is On.
Syntax Help	Check this box to enable the Syntax Help window. The Syntax Help window displays syntax for a keyword after it has been typed. Default is On.
Tab Stops	Type in the number of columns to move for the TAB key. Default is 4.
Line Number Increment	Enter the line number increment for auto line numbers and renumber. Default is 10.
Keyword Format	Enabled when Auto Format is checked. Select All CAPS or Capitalize first letter. Default is Capitalize first letter.

Keyword Color

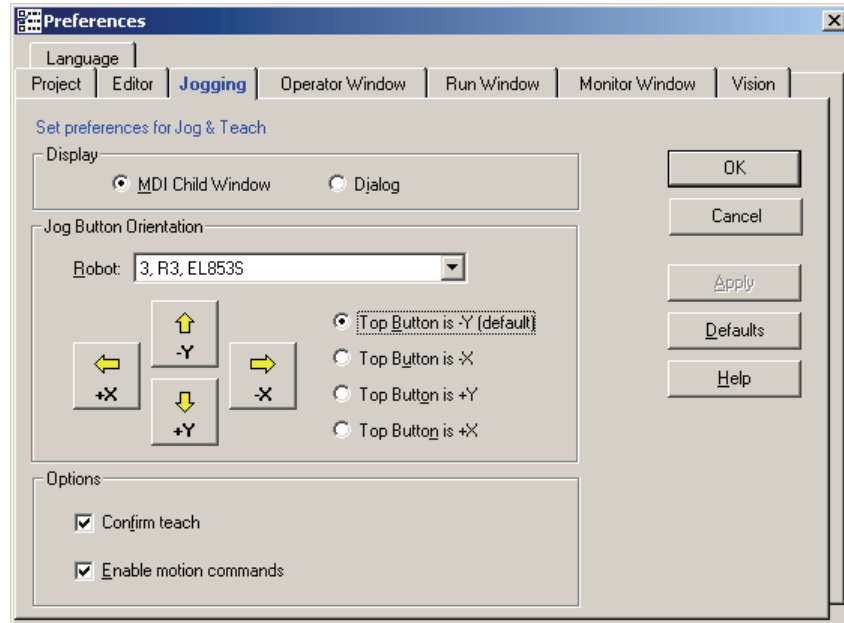
Enabled when Auto Format is checked. Select Dark Blue or Light Blue. Default is Dark Blue.

Font

Click on the Font button to open the fonts dialog. Choose the font you desire for the editor. The monitor window also uses the editor font. The current font name and size is displayed next to the Font button.

Jogging Preferences

This tab lets you setup up the Jog and Teach dialog.

**Display**

The Jog & Teach dialog can be displayed as an MDI child window (default) or as a dialog. When displayed as an MDI child, the Jog & Teach window is displayed in the MDI document area and can remain open while you work with other windows and dialogs. When displayed as a dialog, you can only work with the Jog & Teach controls until you close the dialog. When using screen resolutions less than 1024 x 768, the Jog & Teach MDI window will not fit in the MDI document area. In this case, only the Dialog mode is allowed.

Setting Jog Button Orientation

The jog button orientations can be changed for each robot on the system. This is useful for “aligning” your PC monitor with each robot’s Cartesian coordinate system. Align the buttons so that each robot moves in the direction of the arrows.

Before changing orientation, you must first select the desired robot using the **Robot** drop down list.

You can change the orientation of the jogging buttons and arrow keys for the X and Y axes by clicking on one of the following radio buttons:

Top button is -Y (Default)

Top button is -X

Top button is +Y

Top button is +X

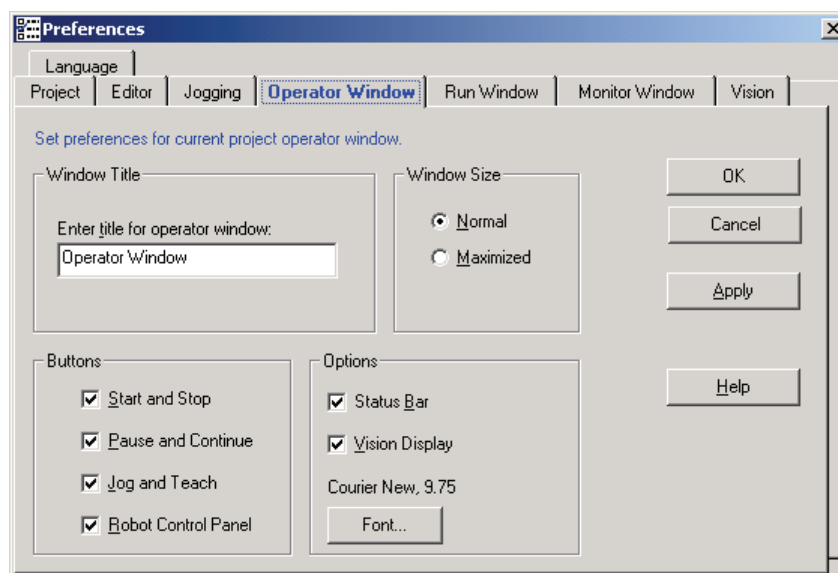
You can also click on one of the buttons to change it to the top button position instead of selecting a radio button.

Other Jogging Preferences

Option	Description
Confirm teach	Check this box if you want a confirmation prompt each time you press the Teach button.
Enable motion commands	Check this box if you want to execute motion commands (Go, Jump, etc.) from the Jog & Teach window.

Operator Window Preferences

This tab allows you to configure the Operator Window. The configuration is saved separately for each project.



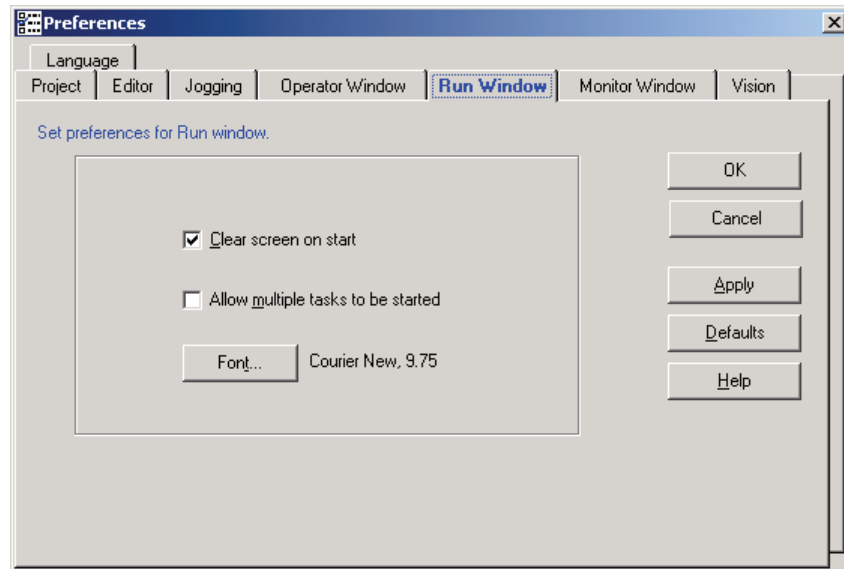
Option	Description
Window Title	Type in the title that you want to appear at the top of the operator window.
Window Size	Choose Normal (normal size, centered on screen) or Maximized (full screen).
Start and Stop buttons	Check this box if you want the Start and Stop buttons to be displayed. These buttons allow the operator to start and stop from the operator window.
Pause and Continue buttons	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.
Jog button	Check this box if you want the Jog button to be displayed. This will allow the operator to open the jog dialog from the operator window.
Robot Control Panel	Check this box if you want the Robot button to be displayed. This will allow the operator to open the robot control panel dialog from the operator window.
Status Bar	Check this box if you want the status bar displayed at the top of the window. The status bar shows Emergency Stop, Safe Guard, and Date/Time.
Vision Display	Check this box if you want a video window to be displayed along with the text window. This option is dimmed if Vision Guide is not installed and enabled.

Font

Click on the Font button to open the fonts dialog. Choose the font you desire for the operator window. The current font name and size is displayed next to the Font button.

Run Window Preferences

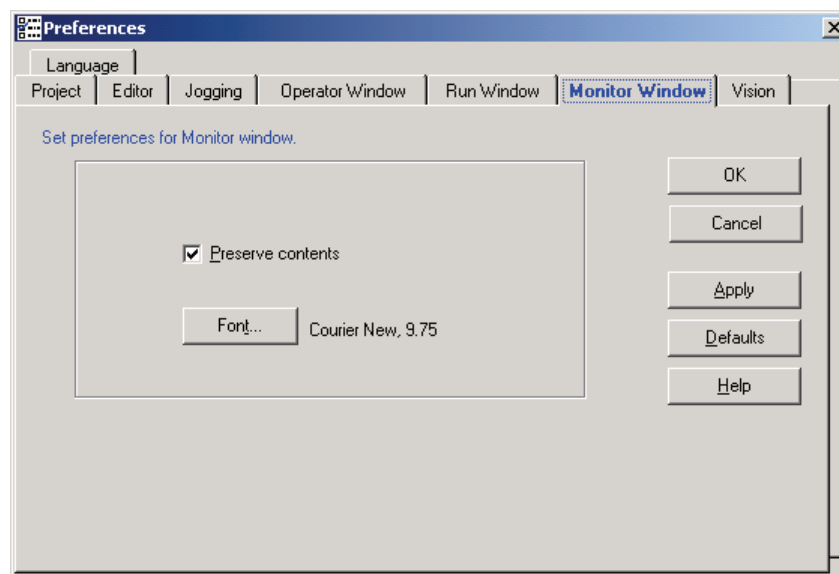
This tab allows you to change preferences for the Run Window.



Option	Description
Clear screen on start	Checking this option will cause the Run Window text pane to be cleared each time the Start button is clicked.
Allow multiple tasks to be started	Checking this option will allow you to start more than one task from the Run window. The Start button will not be disabled after starting a task.
Font	Click on the Font button to open the fonts dialog. Choose the font you desire for the Run window. The current font name and size is displayed next to the Font button.

Monitor Window Preferences

This tab allows you to change preferences for the Monitor Window.



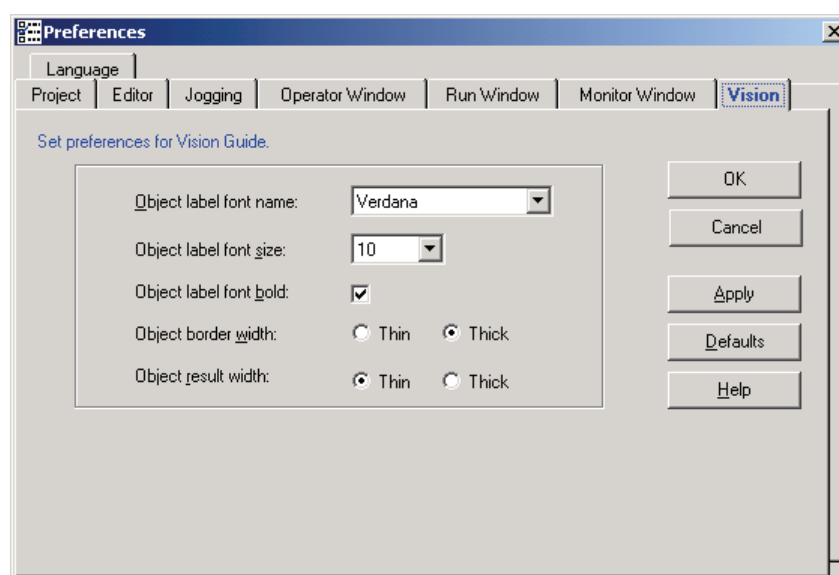
Option	Description
Preserve contents	Checking this option will cause the monitor window to preserve it's contents between sessions.
Font	Click on the Font button to change the font for the Monitor window.

Vision Preferences

This tab allows you to change preferences for the Vision Guide option.



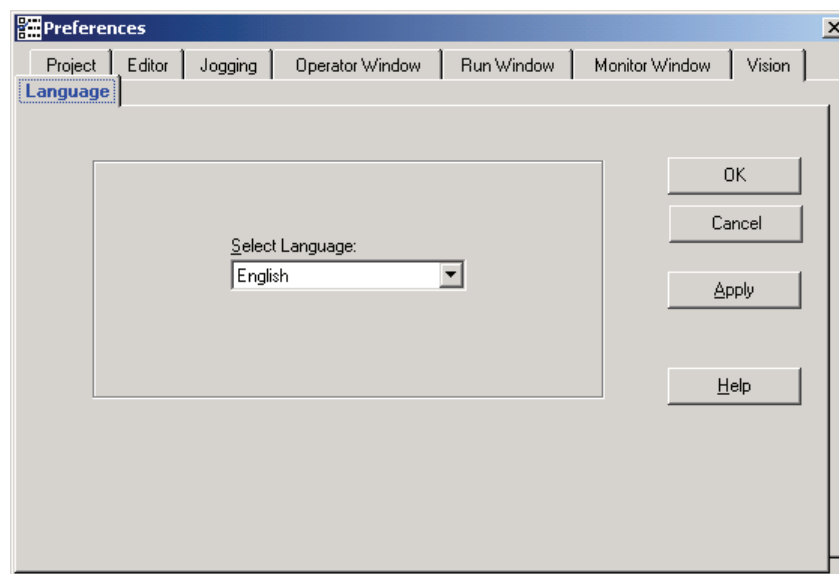
In previous versions of SeikoRCS and SPEL for Windows, the Camera settings were set on this dialog. In EPSON RC+, the vision hardware can be configured from the Setup | System Configuration | Vision tab.



Option	Description
Object label font name	Select the desired font name for the object labels used on the Vision Guide window, Run window vision display, Operator window vision display, and Calibration cycle vision display. All fonts shown are TrueType so that they can be scaled to any size.
Object label font size	Select the desired font size for the object labels used on the Vision Guide window, Run window vision display, Operator window vision display, and Calibration cycle vision display.
Object label font bold	Checking this option will make the font for the object labels bold.
Object border width	Sets the width for the object search window.
Object result width	Sets the width for the object result graphic.

Language Preferences

This tab allows you to change the EPSON RC+ GUI language.



When EPSON RC+ is installed on a Windows system using a Western language, then English and German are available. If it is installed on a Windows system using an Eastern language, then English and Japanese are available.

After selecting the desired language, you must re-start EPSON RC+.

Note that if you change the language to a language that does not match the Windows language, then some system dialogs will be in the Windows language. For example, if you select German for EPSON RC+, but you are using English Windows, then messages boxes and some file navigation dialogs will be in English.

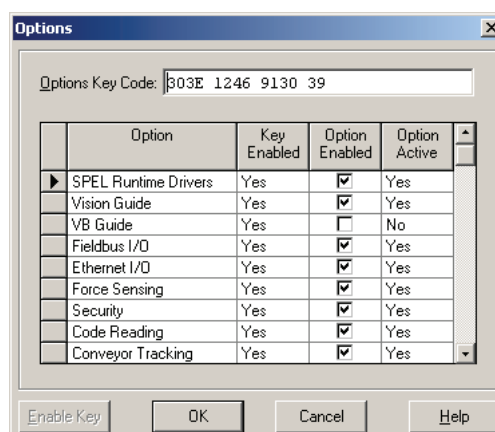
Options Command (Setup Menu)

This dialog allows you to view / enable software options on the current system.

EPSON RC+ uses a parallel port key to enable options on the system.

If an option is not enabled, you can enable it by purchasing it from your distributor. When you call to purchase, you must give the **Options Key Code** to the operator. You will then be given a code to enable the option for the current software options key.

After receiving the code, click the **Enable Key** button and enter the code. The option you purchased should now be enabled.




See the chapter *Installing EPSON RC+ Options* for more details.

Window Menu

Cascade Command (Window Menu)

Use Cascade to show all of the currently open files in windows of the same size, stacked one on top of another.


Shortcuts

Toolbar: 

Tile Horizontal Command (Window Menu)

Use Tile Horizontal to evenly display all open windows horizontally.


Shortcuts

Toolbar: 

Tile Vertical Command (Window Menu)

Use Tile Vertical to evenly display all open windows vertically.


Shortcuts

Toolbar: 

Arrange Icons Command (Window Menu)

Arrange the icons for all child windows that have been minimized.

Shortcuts

Toolbar: 

Close All Command (Window Menu)

Closes all child windows.

Shortcuts

Toolbar: 

1, 2, 3 Command (Window Menu)

A listing 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.

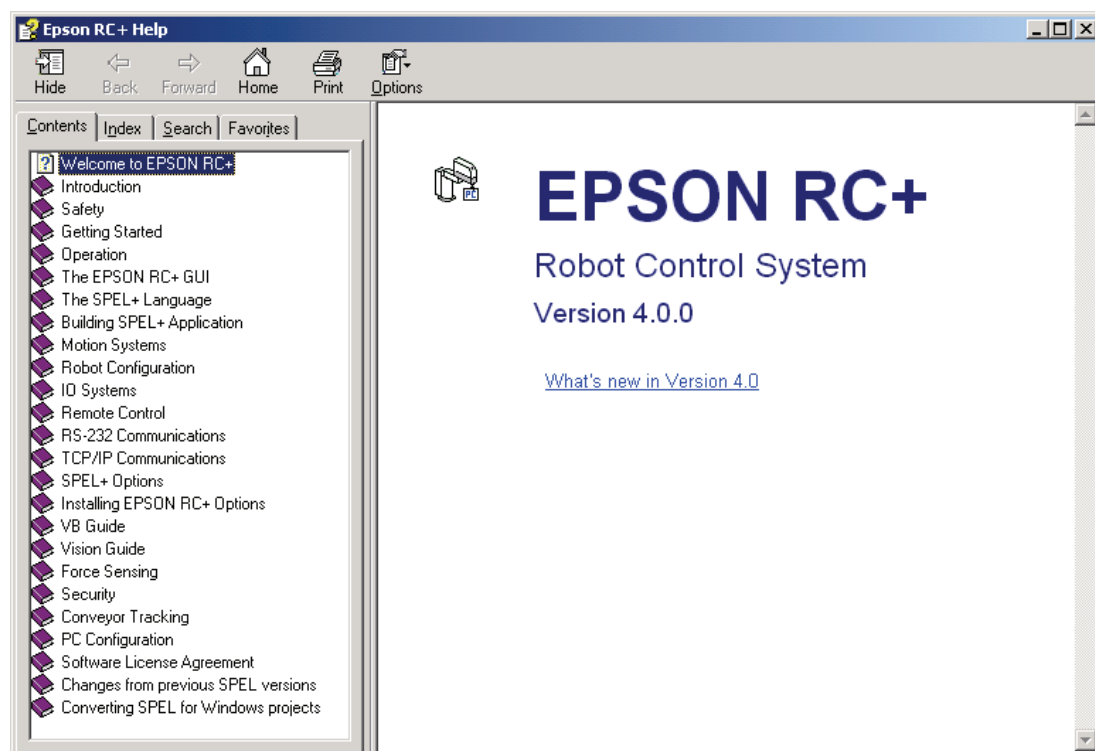
Help Menu

Contents Command (Help Menu)

This command opens the Table of Contents for the EPSON RC+ on-line help system.

Shortcuts


Toolbar: 

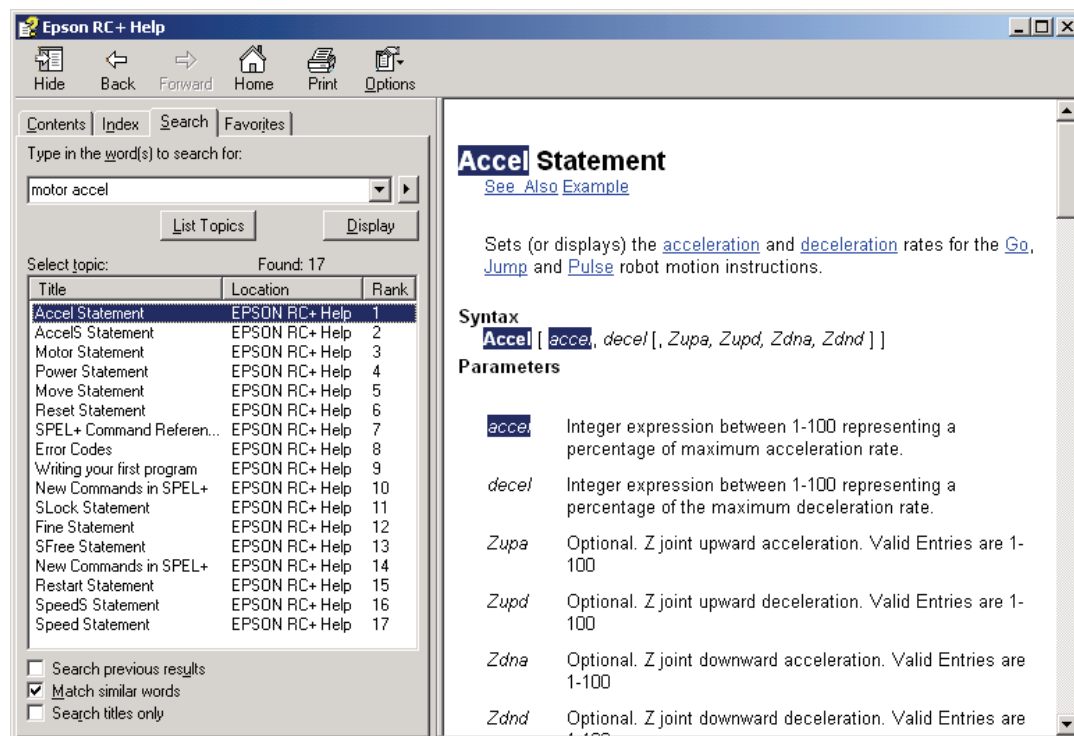


Search for Help On Command (Help Menu)

This command opens the help system search dialog.

Shortcuts

Toolbar: 



About EPSON RC+ Command (Help Menu)

The About command displays a dialog box showing the current version of the EPSON RC+ software, along with copyright and license information. When calling technical support about EPSON RC+, you should report the version you are using from this dialog.



The SPEL⁺ Language

This chapter contains information about the SPEL⁺ Language.

Contents

- Overview
- Program structure
- Commands and statements
- Function and variable names
- Data types
- Working with variables
- Working with strings
- Working with files
- Labels
- Comments
- Error handling
- Multi-tasking
- Chaining
- Robot motion commands
- Input and output control
- Working with robot points
- Using Traps
- Handling Emergency Stop
- Maintenance commands
- Predefined constants

Overview

SPEL⁺ is a BASIC-like programming language that runs on Windows. It supports multitasking, motion control, I/O control, machine vision, force sensing, conveyor tracking.

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 Monitor window.

Program structure

What is a SPEL⁺ program?

A SPEL⁺ program is a text file that contains one or more function definitions. Every program file has a .PRG extension and resides in the project directory on the PC.

A function definition begins with the Function statement and ends with the Fend statement. All source statements to be executed by a function must be included within the body of the function.

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
```

Calling functions

You can call a user function or DLL function by using the Call statement. The function must reside in any program file in the same program group. To get a return value, you must use the function as an argument in a statement. 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)

Call MyDllFunc(1, 2)
MyDllFunc 1, 2
```


Line Numbers

In the EPSON RC+ programming environment, you can choose whether to use line numbers in your program or not. If you want to use line numbers in your programs, then make sure that the first line starts with a number.

The following program contains line numbers:

```
MAIN.PRG

0010 Function Main
0020   Call func1
0030   ...
0100 Fend

0110 Function func1
0120   Jump pickpnt
0130   ...
0140 Fend
```

Commands and statements

Commands and statements consist of a SPEL+ instruction followed by the parameters for that instruction.

A command is executed immediately. You can execute commands from the Monitor window or from the Macros dialog box.

Statements can be used only in programs.

Statements can include more than one SPEL+ instruction by using a semi-colon (;) to separate instructions.

The maximum length for a statement is 200 characters. This includes the line number.

Function and variable names

Function and variable names can include up to 32 alphanumeric characters and the underscore character. 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 a numeric digit or underscore.

String variables must have an additional dollar sign ('\$') suffix, as shown in the example below:

```
Function Test
  String modname$
  Print "Enter model name:"
  Line Input modname$
  Print "model is ", modname$
Fend
```

Data 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	1 byte	TRUE or FALSE
Byte	1 byte	-128 to +127
Double	8 bytes	-1.79E+308 to 1.79E+308
Integer	2 bytes	-32768 to +32767
Long	4 bytes	-2147483648 to +2147483647
Real	4 bytes	-3.40E+38 to 3.40E+38
String	255 bytes	All ASCII characters

Operators

The following table shows the different operators for the SPEL+ language.

Keyword or Symbol	Description
+	Addition
-	Subtraction
*	Multiplication
/	Division
**	Exponentiation
=	Equal
>	Greater than
<	Less than
>=	Greater than or equal to
<=	Less than or equal to
<>	Not equal
And	Performs logical and bitwise AND operation.
Mod	Returns the remainder obtained by dividing a numeric expression by another numeric expression.
Not	Performs logical or bitwise negation of the operand.
Or	Performs the bitwise Or operation on the values of the operands.
Xor	Performs the bitwise Xor operation on the values of the operands.

Working with variables

Variable scopes

There are three different scopes for variables in SPEL⁺:

- Local
- Module
- Global

Local variables

Local variables are available to all statements in the same function. Functions using local variable names do not 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, Integer, Long, 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
```

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, Integer, Long, Real, Double, String

One way to indicate that a variable is module level is to precede the name with "m_", as shown in the example below.

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 m_IntVar1
Fend
```

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, Integer, Long, Real, Double, String) at the beginning of the program before any Function statements:

One way to indicate that variables are global is to precede the name with "g_", as shown in the example below.

Group: MAINGRP

Program: MAIN.PRГ

```
Global Integer g_TotalCycles
```

```
Function main
```

```
    Call LoadPart
    ...
    ...
End
```

Group: MAINGRP

Program: LOADPART.PRГ

```
Function LoadPart
```

```
    Jump pick
    On gripper
    Wait .1
    Jump place
    Off gripper
    Wait .1
    g_TotalCycles = g_TotalCycles + 1
End
```

For more information, see Data Types.

Global Preserve variables

You can preserve global variable values by using the optional Preserve parameter when you declare global variables.

Preserved variables are stored on disk under the globals subdirectory in the current project directory. Each preserved variable has its own file. Whenever a preserved variable is assigned a value, the new value is saved to the disk file. The time overhead to assign a preserved variable is approximately 1 - 2 milliseconds. If the variable is an array, only the element changed is updated in the file using random-access.

Preserved variables are restored during project open or after project build.

If the data type of a preserved variable is changed, or the number of dimensions is changed, then the variable will be erased (values are cleared).

To clear a preserved variable, use the Erase statement.

Arrays

You can declare local, program, and global arrays with up to three dimensions for all data types.

To declare an array, use this syntax:

dataType name (dimension1 [, dimension2 [, dimension3]])

SPEL⁺ arrays are zero based. The first element is referenced with a value of zero.

Dimensions can range from 0 to 2147483646. When using large arrays, out of memory errors may occur, depending on how much memory is in the system.

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 dimensions 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 upper bound of an array dimension.

```
Integer i, a(10)

For i = 1 to UBound(a)
  a(i) = i
Next i
```

You can also assign one entire array to another, as long as the data type and dimensions are the same.

```
Integer a(10), b(10)

b = a
```

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.

Clearing variables

Use Erase to clear all of the elements of an array at run time. You can also clear all variables by using Erase All. This is useful after executing Chain or Restart. You must also use Erase to clear Global Preserve variables.

Working with strings

A string in SPEL⁺ is a set of ASCII characters 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.

You can declare string arrays with up to three dimensions.

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.
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.
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.
Trim\$	Returns the specified string with left and right spaces removed.
UCase\$	Returns the specified string in upper case characters.
Val	Converts a string to a number.

Working with 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	Checks if a folder exists.
FreeFile	Returns an unused file handle.
Input	Inputs one or more variables from a file.
Kill	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 for reading.
Seek	Sets the current file pointer.
WOpen	Opens a file for writing.
Write	Writes out a variable at the current file pointer without appending a line terminator.
WriteBin	Writes binary data.

Before using a file, you must open it with one of three open commands: AOpen, BOpen, ROpen, or WOpen. You specify a file number in the open statement. File number can be from 30 to 63.

Here is an example to save a text file and read it.

```
Function SaveData(ByRef data$() As String)
    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(ByRef data$() As String)
    Integer fNum, i

    fNum = FreeFile
    ROpen "c:\mydata\data.txt" As #fNum
    Input #fNum, i
    Redim data$(i)
    For i = 0 To UBound(data$)
        Input #fNum, data$(i)
    Next i
    Close #fNum
Fend
```

Multi-statements

A program statement can contain several statements separated by semi-colons. The total length of a multi-statement program line cannot exceed 200 characters.

For example:

```
Function Test
    Pass P1; Pass P2; Go P3      ' Multi-statement
Fend
```

Even though multi-statements are supported (mainly for compatibility), it is strongly recommended that they not be used. Multi-statements can make your code more difficult to read and debug.

Labels

A program label 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:    ' This is a program label.
    Print "Program aborted"
Fend
```

Comments

Use comments to add notes to your programs. An apostrophe character (') starts a comment.

Examples of comments

```
Function Main
    ' ***** Main Demo Program *****
    Xqt conveyor      ' Start up the task for conveyor
    Do
        Print "Press ENTER to run demo cycle"
        Print "Press CTRL+C to quit"
        Input dummy
        Call demo      ' Execute the demo function
    Loop              ' Return to start of main loop
```


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 below shows the program instructions that are used for error handling.

Statement Purpose

OnErr	Use the OnErr statement to define the location of the error handling routine.
EClr	Use the EClr statement to clear the current error status. This is normally used in 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.

User Errors

You can define your own error messages by using the User Error Editor which is available from the Tools menu. See the section on User Error Editor in the chapter *EPSON RC+ Gui* for details.

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 Restart statement to start the program from the top.

```
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
      Restart
  Send
Fend
```

Multi-tasking

For some applications, you may want to control other equipment besides the robot, such as a 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 tasks running simultaneously. A task is a function that has been started by the system or by the **Xqt** statement. When Program Groups are enabled, in order to utilize multi-tasking in SPEL⁺, all of the functions used at one time must be included in the same group.

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. You can also specify that a task will not pause by a pause statement or signal by supplying the **NoPause** parameter.

The table below shows the program instructions that are used for multitasking.

Statement	Purpose
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 that are configured to pause.
Cont	Continue all tasks that are in the pause state.

One example for starting another task is to run a conveyor system for the robot work cell.

Program Group: MAINGRP

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
```

Chaining

Use the Chain statement in your programs to start another program group in the project. When the **Chain** statement executes, it loads the points for the program group being chained to and then starts the main program of the group.

Program Groups must be enabled to use Chain. You can enable Program Groups from the Edit Command in the Project Menu.

Here is a simple example that uses Chain.

Program Group: MAINGRP

Program Name: MAIN.PRG

Function Main

```
mainLoop:
    Print "Main Menu"
    Print "1) Run widget 1"
    Print "2) Run widget 2"
    Print "3) Quit"
    Input choice
    Select choice
        Case 1
            Chain "widget1"
        Case 2
            Chain "widget2"
        Case 3
            End
        Default
            GoTo mainLoop
    Send
Fend
```

Program Group: WIDGET1

Program Name: WID1.PRG

```
Function Widget1
    Print "Executing widget 1"
    Chain "MAINGRP"          ' Start main group again
Fend
```

Program Group: WIDGET2

Program Name: WID2.PRG

```
Function Widget2
    Print "Executing widget 2"
    Chain "MAINGRP"          ' Start main group again
Fend
```

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.

Here is an example where two robots run simultaneously, each with its own task.

```
Function main
  Xqt Robot1
  Xqt Robot2
Fend

Function Robot1
  Robot 1
  Speed 50
  Do
    Jump pick
    On gripper1
    Wait .1
    Jump place
    Off gripper1
    Wait .1
  Loop
Fend

Function Robot2
  Robot 2
  Speed 50
  Do
    Jump pick
    On gripper2
    Wait .1
    Jump place
    Off gripper2
    Wait .1
  Loop
Fend
```

Here is another example where two robots run simultaneously, using the same function for each task. Note that the Xqt statements pass the robot number and the RobotTask function executes the Robot statement according to the robotNum parameter.

```
Function main
  Xqt RobotTask(1)
  Xqt RobotTask(2)
Fend

Function RobotTask(robotNum As Integer)
  Integer gripper

  If robotNum = 1 Then
    gripper = 10
  Else
    gripper = 11
  EndIf

  Robot robotNum
  Speed 50
  Do
    Jump pick
    On gripper
    Wait .1
    Jump place
    Off gripper
    Wait .1
  Loop
Fend
```

Robot coordinate systems

Overview

In this section we discuss the coordinate systems for different types of robots supported in SPEL⁺.

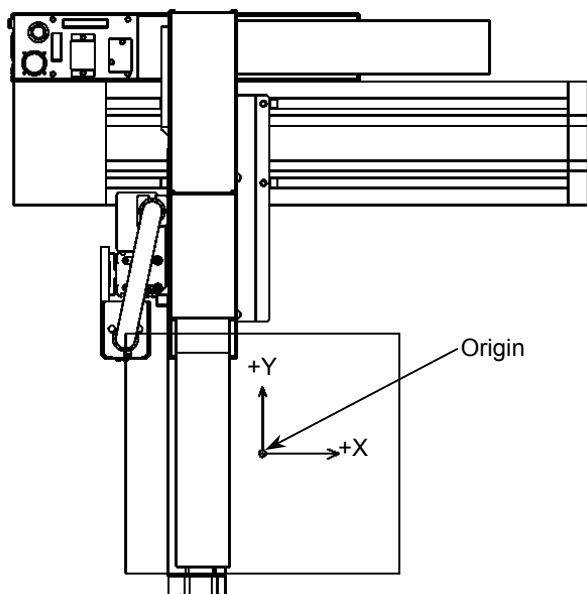
The following coordinate systems are used in SPEL⁺:

Robot Coordinate System	This is the native coordinate system of the robot. This is also known as the default base coordinate system.
Local Coordinate System	This is a user defined coordinate system located somewhere within the working envelop.
Tool Coordinate System	This is the coordinate system of the tool mounted on the robot end-effector.

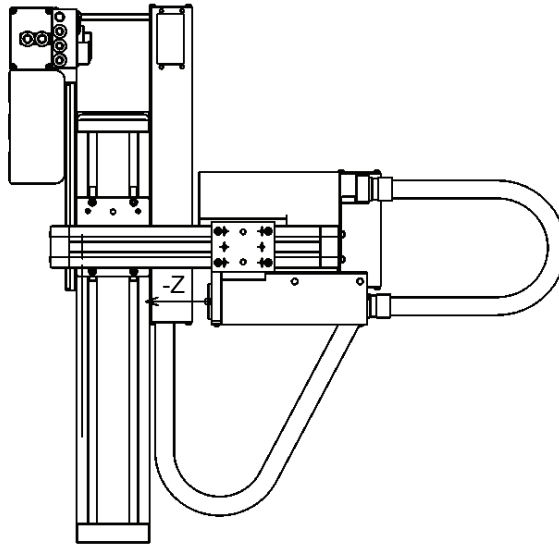
Robot Coordinate System of Cartesian Robot

For the EZ Modules X4 series, the robot coordinate system differs depending on the robot model and mounting direction. Refer to the EZ Modules X4 series manipulator manual for details of each robot coordinate system.

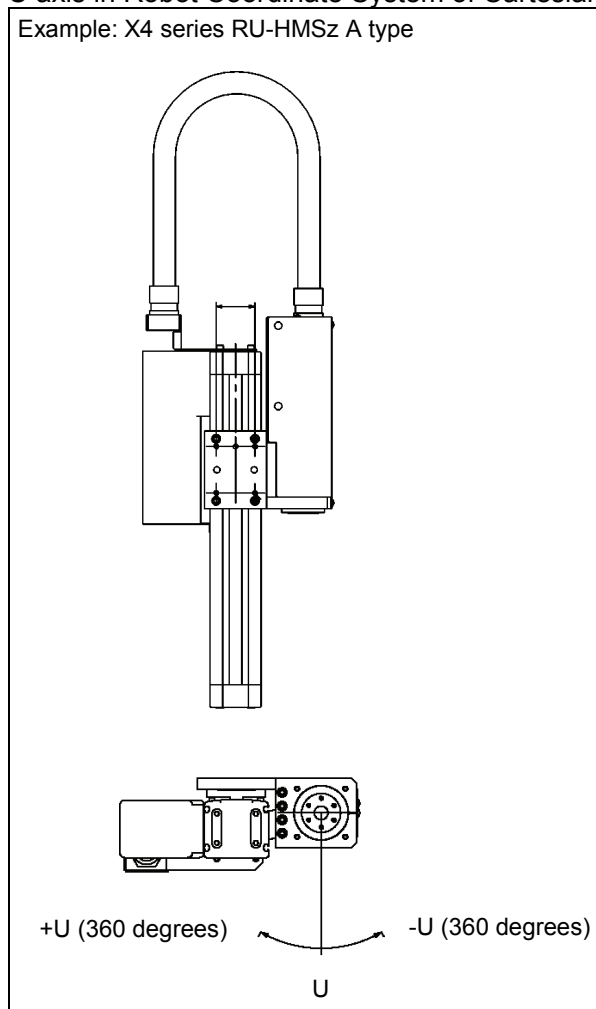
Example: X4 series RU-HMSz A type



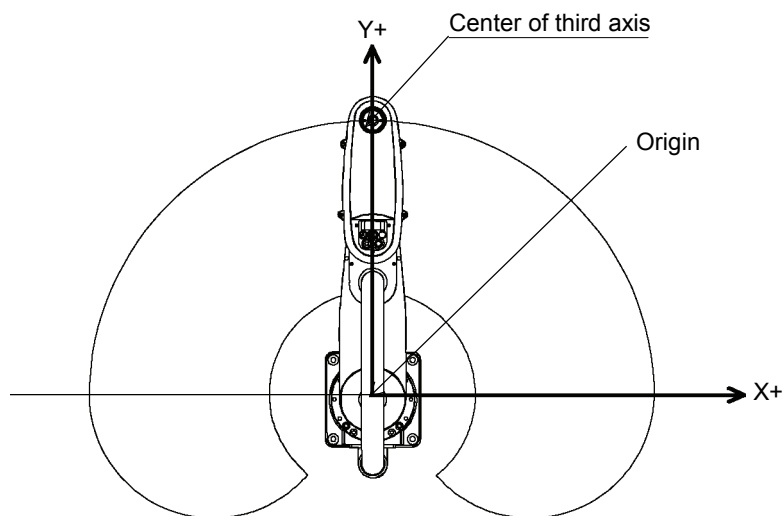
Z axis in Robot Coordinate System of Cartesian Robot
Example: X4 series RU-HMSz A type



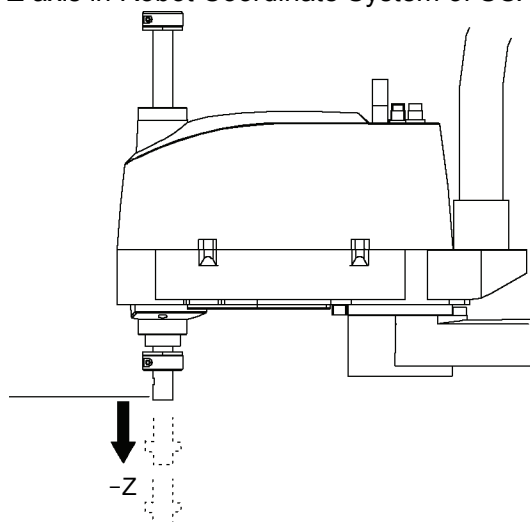
U axis in Robot Coordinate System of Cartesian Robot
Example: X4 series RU-HMSz A type



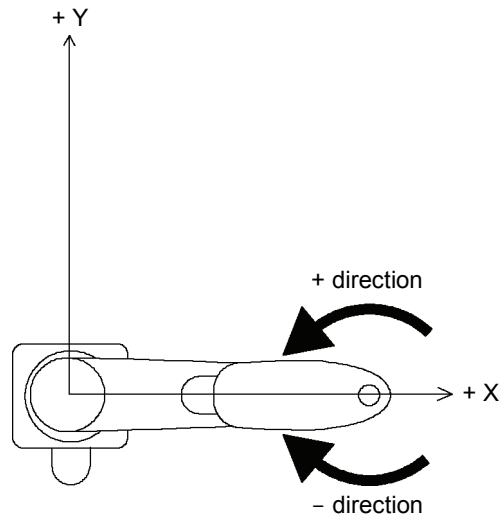
Robot Coordinate System of SCARA Robot



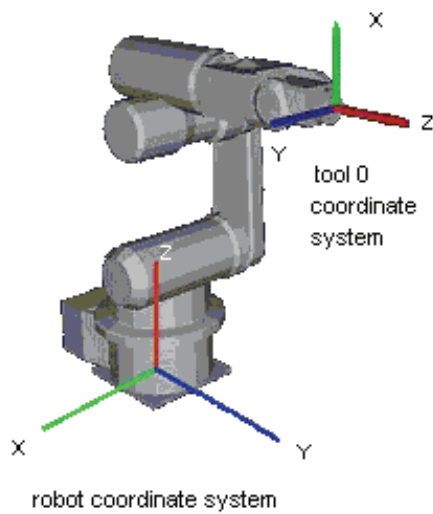
Z axis in Robot Coordinate System of SCARA Robot



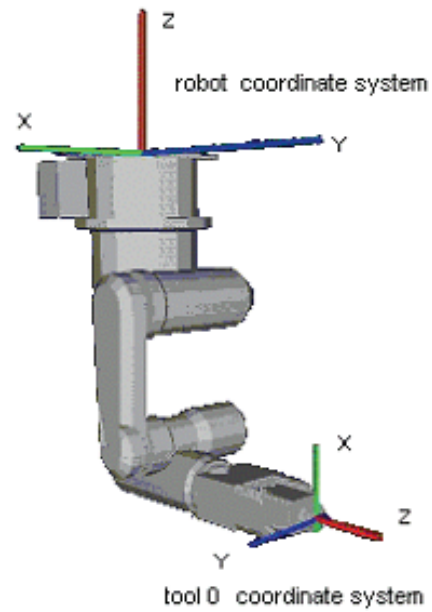
U axis in Robot Coordinate System of SCARA Robot

**Robot Coordinate Systems for 6-Axis Robots**

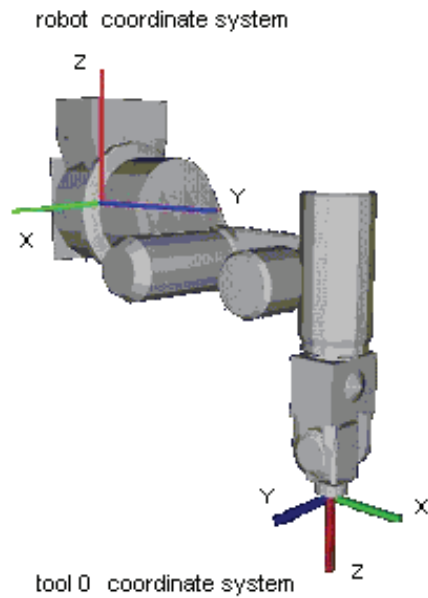
Floor Mounting



Ceiling Mounting



Side (Wall) Mounting



Local Coordinate Systems

With SPEL+, a maximum of 15 local coordinate systems can be defined. SPEL+ correlates robot coordinate systems and local coordinate systems by defining in advance the relative positional relationship of the local coordinate system from the robot coordinate system, assigning local numbers (1 to 15), and then assigning the local numbers to coordinate system attributes (local).

To define a local coordinate system, use the Local statement

Tool Coordinate Systems

Point data is defined by the position and orientation of the tool coordinate system with respect to some reference rectangular 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.

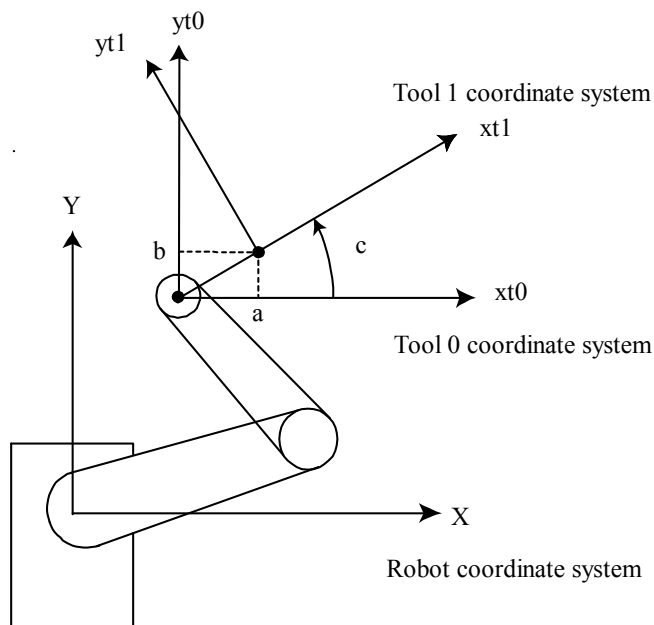
You can also define and use your own tool coordinate systems. To define the tool coordinate systems, use Tlset.

The default TOOL 0 coordinate systems are defined as follows according to the robot type.

SCARA and Cartesian Tool 0 coordinate system

The origin of TOOL 0 for SCARA and Cartesian robots is the center of the forth 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 the figure below.)

TOOL 0 coordinate system can rotate as the fourth joint rotates.



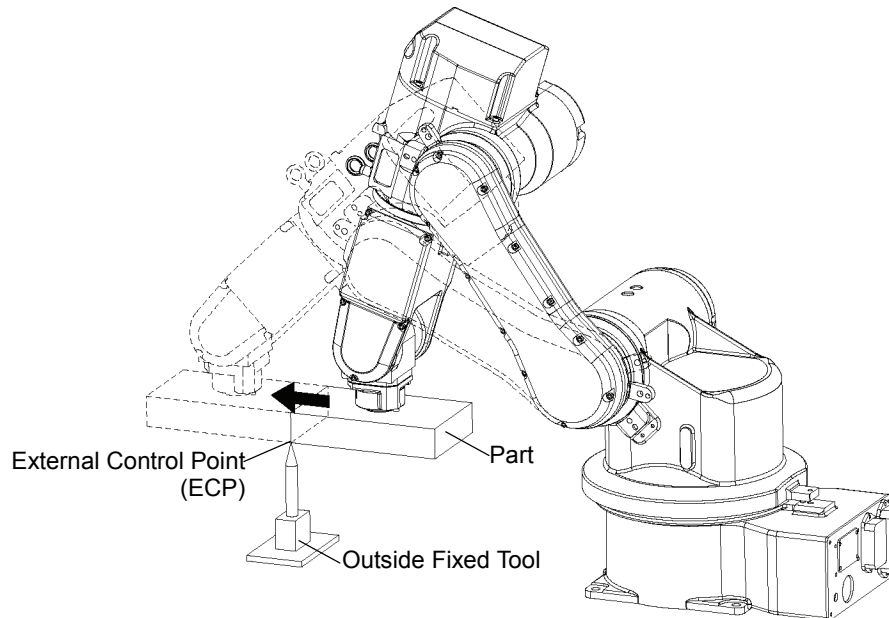
6-Axis Tool 0 coordinate system

For table mounting, the origin of TOOL 0 is the flange side center of the sixth joint. In TOOL 0, the tool Z axis is perpendicular to the sixth joint flange. (See the figure in the previous section *Robot Coordinate Systems*). The TOOL 0 coordinate system moves as the 6-Axis robot changes its orientation.

For ceiling mounting and wall mounting robots, the TOOL 0 coordinate systems are defined as shown in the figures in the section *Robot Coordinate Systems*.

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.



Use the ECPSet statement for defining an ECP coordinate system. A maximum of 15 ECP coordinate systems can be defined.

The following commands are available for optional ECP:

- Move command
- Arc3 command
- Curve and CVMove commands
- ECP jog motion

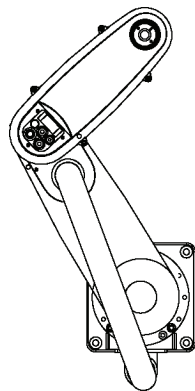
For details, refer to the chapter ECP Motion.

Robot arm orientations

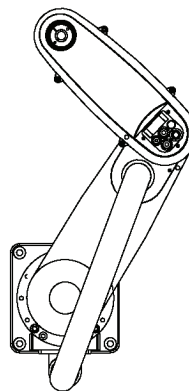
When developing a robot a program, it is necessary to specify the point data taught for a particular arm orientation. If you fail to do so, the position can deviate slightly depending on the arm orientation, which in turn can cause the arm to follow an unexpected path, resulting in interference with 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.

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 below.



Lefty arm orientation



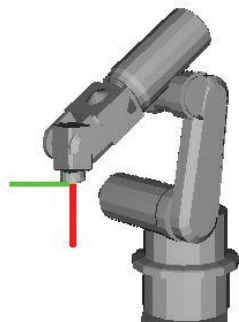
Righty arm orientation

Examples of moving to the same point using Lefty and Righty arm orientations

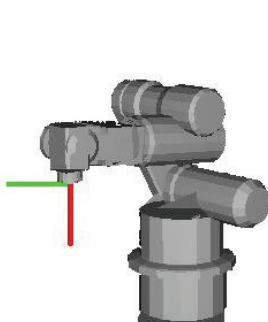
6-Axis robot arm orientations

The 6-Axis robot can be operated in various arm orientations within a given work envelope as shown below:

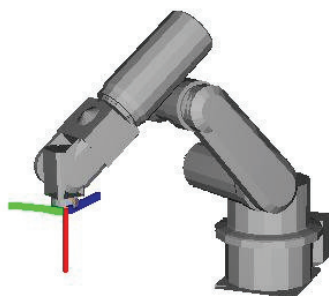
Righty hand orientation



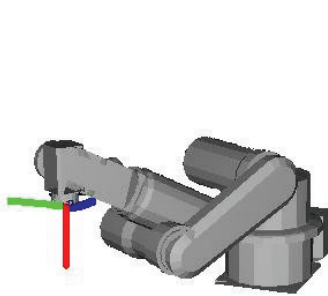
Lefty hand orientation



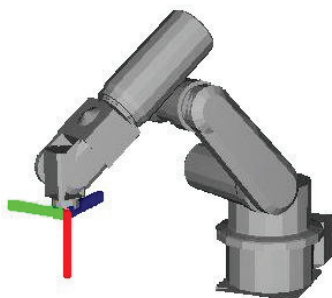
Above elbow orientation



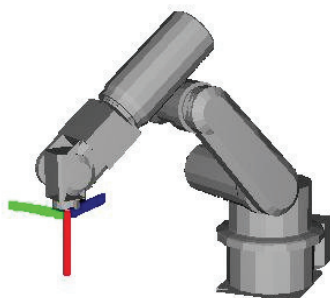
Below elbow orientation



NoFlip wrist orientation



Flip wrist orientation



To specify orientation for 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), and NF (NoFlip wrist orientation) or F (Flip wrist orientation).

There are eight available orientations as shown below, however, the 6-Axis robot cannot be operated in all of the orientations depending on 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 are 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 ($-180 < \text{the forth joint angle} \leq 180$) or J4F1 ($\text{the forth joint angle} \leq -180$ or $180 < \text{the forth joint angle}$).

To specify the J6Flag, add a forward slash (/) followed by J6F0 ($-180 < \text{the sixth joint angle} \leq 180$), J6F1 ($-360 < \text{the sixth joint angle} \leq -180$ or $180 < \text{the sixth joint angle} \leq 360$), or J6Fn ($-180 \cdot (n+1) < \text{the sixth joint angle} \leq 180 \cdot n$ or $180 \cdot n < \text{the sixth joint angle} \leq 180 \cdot (n+1)$).

Robot motion commands

SPEL⁺ includes several commands for controlling the robot from your programs.

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.



For incremental encoder robots only: After the robot controller is first powered up, the robot position must be calibrated. Use the Mcal command to calibrate the robot. You can set the order for which the robot joints go to their calibrated positions by using the MCodr command. Mcal needs to be used only once after the SPEL runtime drivers have been started.

Point to point motion

Point to point (PTP) commands move the robot from its current position to a specified point. Motion 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. The Arch table settings determine the Jump profile.
Jump3	Jump to a point in 3 dimensions.
Jump3Cp	Jump to a point in 3 dimensions using CP motion.
Pass	Move near one or more points.
TGo	Move directly to a point in a tool coordinate system.

Linear motion

Linear motion commands move the robot from its current position to a specified point in a *straight line*.

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.

Joint motion

Command	Description
---------	-------------

JTran	The JTran command can be used to move one joint of the robot to position specified in degrees or millimeters, depending on the joint type. The speed and acceleration are the same as for point to point motion commands.
PTran	The PTran command can be used to move one joint of the robot to an encoder pulse position. The speed and acceleration are the same as for point to point motion commands.
Pulse	The Pulse command can be used to move all joints of the robot to encoder pulse positions. The speed and acceleration are the same as for point to point motion commands.

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.

Curves

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	Creates a file containing a path specification.
CVMove	Executes a path specified by Curve.

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, moving the arm by normal CP motion commands will cause an increase in the variation of speed, acceleration and deceleration of tool orientation. In some cases, an error will occur.

To prevent these situations, add the ROT parameter to the CP motion commands. The arm will be moved based on the specified angular velocity and acceleration/deceleration of the main axis regarding the orientation variation.

The angular velocity and acceleration/deceleration of the main axis regarding the orientation variation should be specified with the SpeedR and AccelR commands in advance.

For example:

```
SpeedR 50          ' degree/sec
AccelR 200, 200 ' degree/sec2
Move P1 ROT
```

NOTE



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    BMove
Arc      TMove
Arc3     Jump3CP
```

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. See PTPBoost and PTPBoostOK in the *SPEL+ Language Reference* manual for more details.

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).

The X, Y, and Z coordinates of a point are specified in millimeters. The U, V, and W coordinates are specified in degrees.

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 Jog & Teach window.

Defining points

You can define points in a program statement, with the Jog and Teach Tool, or at the Monitor Window.

In a program statement or at the Monitor window, you can assign coordinates to a point, or define a point that is the current robot arm position.

```
P1 = XY(200, 100, -25, 0)      ' Assign coordinates to point P1
pick = XY(300, 200, -45, 0)   ' Assign coordinates to point pick
P10 = P*                       ' Assign a point to current position
```

Referencing points by name

You can assign names to point numbers so you can refer to points by name in a program. Assign names from the point editor (see Editing Points) or the Jog and Teach Tool. Names must be unique for each point number when used in the same point file.

```
For i = 0 To 10
  Go pick
  Jump place
Next i
```

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
```



Although you can define points at the Monitor window for test purposes, it is recommended that all points be defined in a program or with the Jog and Teach Tool. Points defined at the Monitor window will be cleared from memory when you build a project or run a program unless you execute SavePoints.

Using points in a program

When running programs, the default point file for each robot is loaded at start up. You can also define points in the program, or load them using the LoadPoints statement.

```
Function main
  Integer i

  LoadPoints "modell1.pnt"
  For i = 0 To 10
    Jump pick
    Jump place
  Next i
Fend
```

Importing points in a program

When running programs, you can import points into the current project using the ImportPoints statement.

```
Function main
  Integer i

  ImportPoints "c:\models\modell1.pnt", "robot1.pnt"
  LoadPoints "robot1.pnt"
  For i = 0 To 10
    Jump pick
    Jump place
  Next i
Fend
```

Saving and loading Points

Use LoadPoints to load a point file in the current project for the currently selected robot. You can optionally specify the **Merge** parameter to combine points in a file with points that have already been loaded. You cannot load a point file from a different robot than is currently active.

Use SavePoints to save the points for the current robot. If the point file is not in the current project, it will be added automatically.

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         ' Righty 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 are rotated 360 degrees. To distinguish these points, the J4Flag and J6Flag point attributes are provided. These flags allow you to specify a position range for joint 4 and joint 6 for a given point.

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
```

Extracting and setting point coordinates

Use the CX, CY, CZ, CU, CV, and CW commands to get a coordinate of a point or 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
```

Alteration of points

There are several ways of modifying a point without reteaching 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

Input and output control

Hardware I/O

There are 16 DC inputs and 16 DC outputs on a standard drive unit. By purchasing expansion I/O boards, you can expand I/O up to 368 inputs and 368 outputs. You can also add Ethernet I/O and Fieldbus I/O. See *I/O Systems* chapter for details.

Memory I/O

There are 64 bytes (512 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.



In previous versions, I/O bit numbers or labels used for memory bits used the "\$" prefix. For example, On \$1 turned on memory bit 1. Off \$cycDone turned off the memory bit defined in the I/O label editor called "cycDone". This syntax is still supported, but has been replaced with the Mem commands.

I/O Commands

Command Description

In	Reads one byte (eight bits) of input data. Used for SPEL Standard I/O inputs.
InW	Reads one word (sixteen bits) of input data. Used for SPEL Standard I/O inputs.
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. Used for SPEL Standard I/O outputs.
On	Turns on one output bit. Used for SPEL Standard I/O outputs.
Out	Sets/reads one byte (eight bits) of output data. Used for SPEL Standard I/O outputs.
OutW	Sets/reads one word (sixteen bits) of output data. Used for SPEL Standard I/O outputs.
Oport	Reads the status of one output bit. Used for SPEL Standard I/O outputs
InBCD	Reads one byte of 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. Used for SPEL Standard I/O inputs.

Other I/O commands are available when I/O options are purchased. The commands for I/O options are described in their respective sections in the *I/O Systems* chapter.

Using Traps

Traps enable a function to be called when a certain event occurs. There are several types of traps. Four traps are user defined and are triggered by an input condition. The remaining traps are system defined. The following table describes the traps.

Trap	Description
Trap 1 - 4	Triggered by an input condition, either from a hardware input or a memory bit. These traps are level triggered (not edge triggered), so you should wait until the condition is false before rearming the trap, otherwise an infinite loop will occur. User traps can use GoTo, GoSub, or Call.
Trap Emergency	Triggered when Emergency Stop occurs. First, all tasks are stopped, then the function is called. Chain or Restart may be called after waiting for EStop to be cleared in the trap function.
Trap Error	Triggered when an error occurs that has not been caught by a user error handler. First, all tasks are stopped, then the function is called. Chain or Restart may be called.
Trap SgOpen	Triggered when the safeguard circuit is open. Must be rearmed inside the trap function.
Trap SgClose	Triggered when the safeguard circuit is closed. Must be rearmed inside the trap function.
Trap Pause	Triggered when the system enters the Pause state. Must be rearmed inside the trap function.
Trap Abort	Triggered when all tasks have been stopped by the user or by the system. First, all tasks are stopped, then the function is called.

Note that all traps except for Emergency, Error, and Abort must be rearmed before they will fire again. To rearm a trap, execute Trap at the end of the trap function.

You should keep trap functions short and avoid continuous loops. You cannot execute motion commands in trap functions.

```
Function main
  ' Arm the trap
  Trap 1 Sw(1) = On Call Sw1Trap
  Do
    RunCycle
  Loop
Fend

Function Sw1Trap
  On 1, 2
  ' Wait for trap condition to clear
  Wait Sw(1) = Off
  ' Rearm the trap
  Trap 1 Sw(1) = On Call Sw1Trap
Fend
```


Handling Emergency Stop

When Emergency Stop occurs, all tasks are stopped. You can catch this condition using Trap Emergency to wait for the EStop to be cleared, then restart your main program.

```
Function main
  ' Arm the trap
  Trap Emergency Call EStopTrap
  Do
    RunCycle
  Loop
Fend

Function EStopTrap
  Print "Emergency Stop"
  Print "Clear the Emergency Stop to restart"
  Do
    Wait 1
    Reset
  Loop Until Not EstopOn
Fend
```

Maintenance commands

There are maintenance commands in SPEL⁺ used for robot controller initialization and configuration. Maintenance commands should be executed from the Monitor Window.

Command	Description
---------	-------------

Ver	Displays robot setup parameters. You can print out the Ver listing. See Print Command (File Menu).
------------	--

Verinit	Initialize robot setup parameters.
----------------	------------------------------------

You can also execute MKVER and SETVER from the Tools | Maintenance dialog box. See the chapter *EPSON RC+ GUI* for details.

Predefined constants

There are several predefined constants for use in SPEL⁺ programs. At project build time, the values for these constants are substituted for the constant name.

Constant Name	Value	Usage
TRUE	-1	Boolean expression
FALSE	0	Boolean expression
CR	CHR\$(13)	String expression
LF	CHR\$(10)	String expression
CRLF	CHR\$(13) + CHR\$(10)	String expression
TAB	Tab character	String expression
PI	3.1415926535898	Numeric expression
MB_OK	0	MsgBox flags
MB_OKCANCEL	1	MsgBox flags
MB_ABORTRETRYIGNORE	2	MsgBox flags
MB_YESNOCANCEL	3	MsgBox flags
MB_YESNO	4	MsgBox flags
MB_RETRYCANCEL	5	MsgBox flags
MB_ICONSTOP	16	MsgBox flags
MB_ICONQUESTION	32	MsgBox flags
MB_ICONEXCLAMATION	48	MsgBox flags
MB_ICONINFORMATION	64	MsgBox flags
MB_DEFBUTTON1	0	MsgBox flags
MB_DEFBUTTON2	256	MsgBox flags
IDOK	1	MsgBox return
IDCANCEL	2	MsgBox return
IDABORT	3	MsgBox return
IDRETRY	4	MsgBox return
IDIGNORE	5	MsgBox return
IDYES	6	MsgBox return
IDNO	7	MsgBox return
CNV_QUELEN_ALL	0	Cnv_QueLen
CNV_QUELEN_UPSTREAM	1	Cnv_QueLen
CNV_QUELEN_PICKUPAREA	2	Cnv_QueLen
CNV_QUELEN_DOWNSTREAM	3	Cnv_QueLen
FORCE_LESS	0	Force sensing
FORCE_GREATER	1	Force sensing
FORCE_XFORCE	1	Force sensing

Constant Name	Value	Usage
FORCE_YFORCE	2	Force sensing
FORCE_ZFORCE	3	Force sensing
FORCE_XTORQUE	4	Force sensing
FORCE_YTORQUE	5	Force sensing
FORCE_ZTORQUE	6	Force sensing

Calling Functions in Dynamic Link Libraries

SPEL⁺ allows you to call functions in Dynamic Link Libraries (DLLs). You can create your own DLLs to create additional commands for the SPEL⁺ language (see next section).



CAUTION

As with any DLL, care must be taken that parameters are passed correctly and that DLL functions do not corrupt memory by overwriting variable space, etc. It is not recommended that beginners create DLLs for SPEL+. Also, you must handle multithreading situations if you call a DLL from more than one task at the same time.

Use the Declare statement to declare a DLL function call. You must use the Declare statement once for each library call you want to make. It is recommended that you create one program file in your project that contains your Declare statements for each category of functions. You may also want to create SPEL⁺ wrapper functions that encapsulate the DLL calls to simplify calling from other SPEL functions you have written.

Here's an example that calls the Windows API to play wave file.

File SOUND.PRG

```
Declare PlaySound, "winmm.dll", "PlaySound", (name$ As String,  
module As Long, dwFlags As Long) As Long  
  
#define SND_FILENAME &h20000  
  
' SPEL+ wrapper function  
Function PlayWaveFile(fileName$ As String)  
  
    PlaySound fileName$, 0, SND_FILENAME  
Fend
```

Creating a DLL

This section describes how to add a command to SPEL⁺. You can use any development environment capable of creating standard Windows DLL libraries. The examples in this section use Microsoft Visual C++ 6.0 in Microsoft Visual Studio.

Step 1: Decide on functions and parameters

You need to plan the functions and parameters to use for your DLL. By default, parameters are passed by value (By Val), which means that they are not changed when the function returns, since only the value was passed. Values passed by reference can be changed by the DLL function.

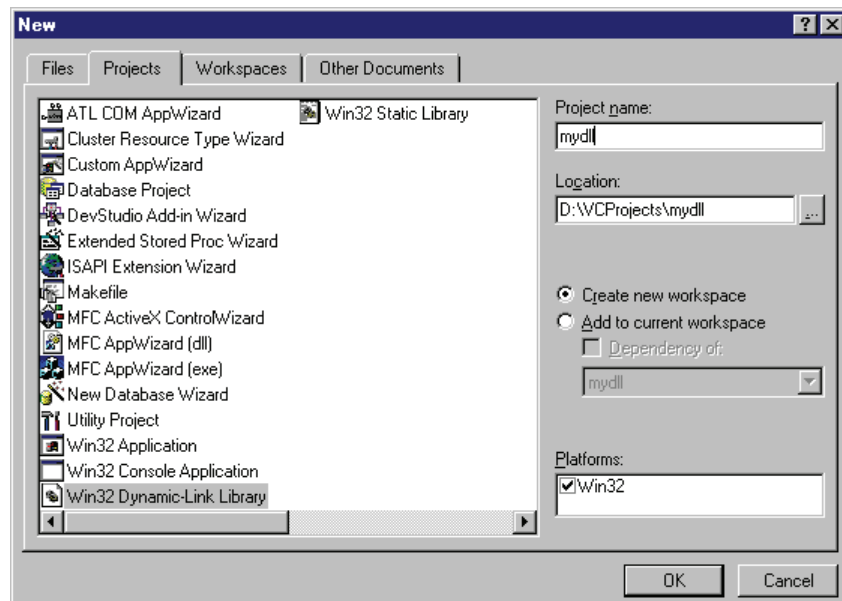
You can pass numbers and strings by value or reference. Number variables can also be arrays.

NOTE In the current release of EPSON RC+, you cannot pass 2 byte integer values or variables to a DLL. Always use the Long data type for integer parameters.

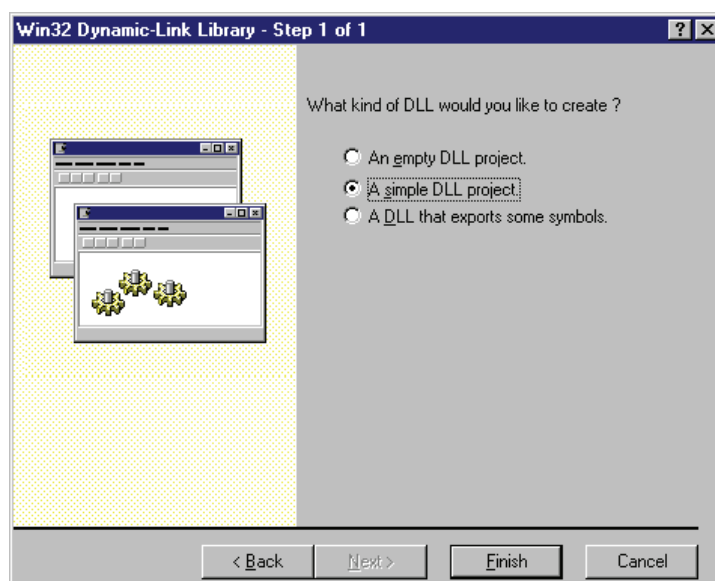
In this example, we will create a function called MyDllAdd that takes two long numbers passed by value and returns a long.

Step 2: Create a DLL project

1. Start Visual C++.
2. Select File | New.
3. Select Win32 Dynamic Link Library.



4. Select the "A simple DLL project" radio button.



5. Click finish to create the project.
6. Open the mydll.cpp file.

7. Add a function called MyDllAdd as shown below. Note that you must use the `_stdcall` keyword (or equivalent) so that Pascal calling convention is used.

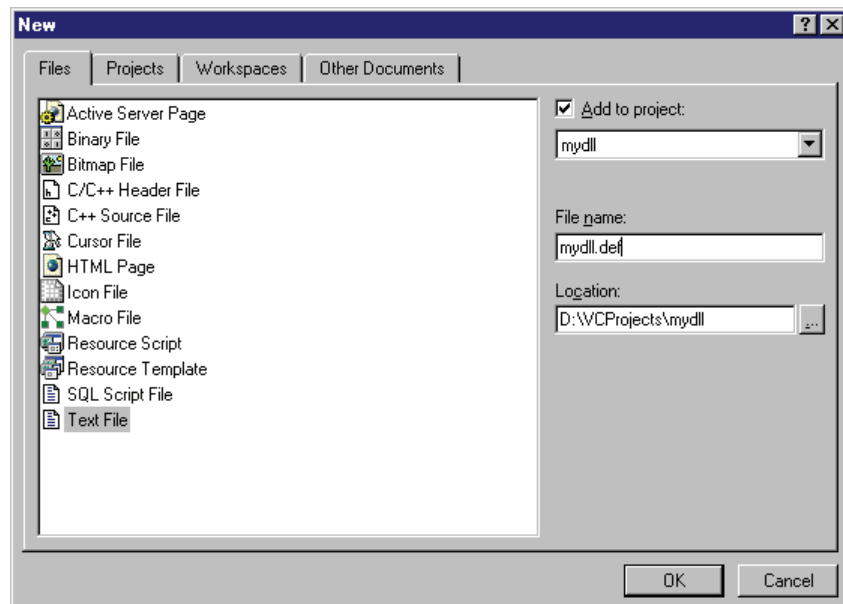
```
// mydll.cpp : Defines the entry point for the DLL application.
//

#include "stdafx.h"

BOOL APIENTRY DllMain( HANDLE hModule,
                      DWORD  ul_reason_for_call,
                      LPVOID lpReserved
                      )
{
    return TRUE;
}

// Add Function
long _stdcall MyDllAdd(long a, long b)
{
    return a + b;
}
```

Now you must export the function in a DEF file. Select File | New and create a text file in the project called mydll.def.



Open mydll.def.

Add the following lines:

```
EXPORTS
```

```
MyDllAdd
```

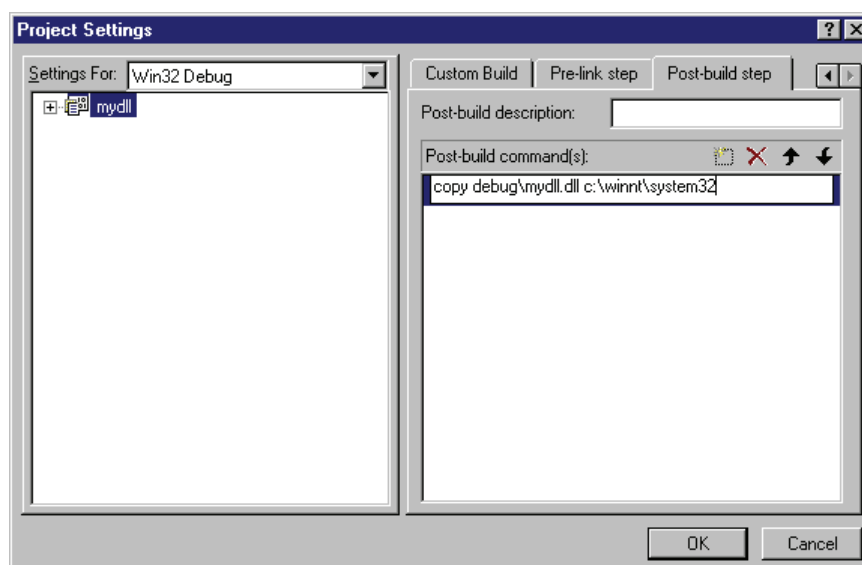
To make it easier to test your DLL, it is recommended that you add a post build command to copy the DLL to the Windows system32 directory so SPEL+ can access it easily.

Select Project | Settings.

Navigate to the Post-build Step tab.

Add the following step:

```
copy debug\mydll.dll c:\winnt\system32
```



8. Press F7 to build the DLL.

Step 3: Call the DLL function from SPEL⁺

You can now try your DLL function from SPEL⁺.

Start EPSON RC+.

Create a new project called MyDllTest.

Declare the DLL function and call it.

```
Declare MyDllAdd, "mydll.dll", "MyDllAdd", (a As Long, b As Long) As Long
```

```
Function main
```

```
    Long result
```

```
    result = MyDllAdd(2, 3)
```

```
    Print "result = ", result
```

```
End
```

Step 4: DLL Debug Test Cycle

You can easily make changes in your DLL, build it, and try it without having to close VC++ or EPSON RC+. Make sure that you abort all tasks before making changes. SPEL⁺ will unload all DLLs in use after all tasks are aborted.

Follow these steps:

1. Abort All tasks.
2. Switch to VC++.
3. Make your program changes and build (press F7).
4. Switch back to EPSON RC+ and run your project.

More information

See the description for the Declare statement in the on-line help or SPEL+ Language Reference manual. It contains information on how to work with various types of parameters.

Building SPEL⁺ Applications

Designing Applications

Creating the simplest application

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 robot1.pnt 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 Jog and Teach Tool.
4. Run the program by selecting Start from the Run menu or by pressing F5 (the shortcut key for the Start command).

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.

Operator interface

Operator Window

Use the operator window provided with EPSON RC+.

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.

Operator Pendant

Use the OP500RC Operator Pendant. With the OP500RC, operators can run programs and teach points. See the OP500RC Manual for more information.

Windows Application using VB Guide

Use the VB Guide Option along with a Windows development tool such as Visual Basic or Visual C++. See the VB Guide Manual for more information.

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. Will you be running the same program for different parts, requiring different points? Then you should configure the EPSON RC+ project to use separate point files for each group. Otherwise, configure the project to use the same points for all program groups.

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 effector 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. The robot controller supports up to 368 inputs and 368 outputs of standard I/O. You must purchase additional I/O boards if you need more than 16 inputs or outputs. You can also use Ethernet I/O and Fieldbus I/O.

Peripherals

You can connect RS232 peripherals to the robot controller serial ports, such as bar code readers, etc. You can use Comm1 or Comm2. You can also add more ports by installing an optional RS232 expansion board. See the chapter RS-232 Communications.

You can use TCP/IP to connect peripheral equipment. See the chapter TCP/IP Communications.

Auto start at power up

Your application can automatically log in a Windows user and start your SPEL+ project after Windows boots.

See the section *Start Up* in the chapter *Operation* for details..

Managing Projects

Overview

What is an EPSON RC+ Project?

An EPSON RC+ project is a collection of SPEL+ program files, robot point files, and setup parameters 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 code, robot configuration, system variables, and point definitions in one project, it's easy to open a project and begin running or editing. Also, it's 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.

There are also commands for copying and renaming projects, making it easy to create new projects from previous versions and for backing up projects to ZIP or floppy disks.

What's in an EPSON RC+ project?

Each project has its own directory on disk. This keeps all of the files for each project separate from each other. The project directories are under the \EpsonRC\Projects directory on any drive. Sub directories may also be created.

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+. You should never edit this file. Doing so may cause errors when you open the project. This file has a .PMK extension.

Program source files

A program source 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) on the PC or called from other functions.

Program groups

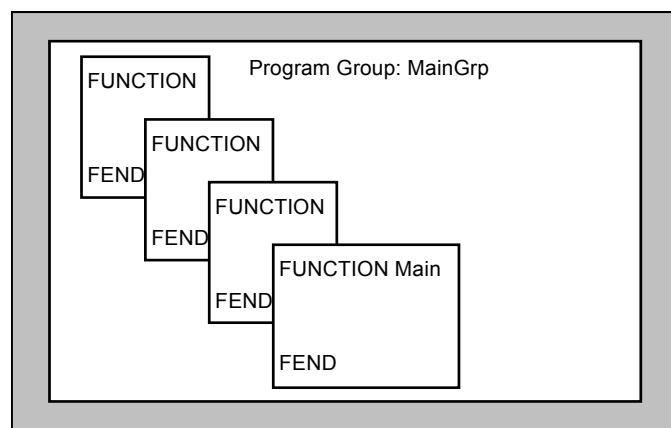
By default a project has no program groups. You can enable groups from the Project | Edit dialog.

Each project can contain one or more program groups. Each program group contains one or more program source files. Each program group has a main program that is executed first when the group is started. You can execute a group from within a program by using the Chain command.

Programs must be grouped together in SPEL+ in order to allow executing more than one function (task). To run an executable in SPEL+, all functions must be in the same object file. Think of each program group as an executable file.

There is one predefined group in each project: MainGrp. You cannot delete this group, but it can be empty. It can also be renamed.

The diagram below shows a program group.



Point files

A point file contains a list of robot points. The file extension is PNT. Each point describes a Cartesian coordinate in the robot work envelope. Point files are associated with project robots. Each robot in a project has its own set of point files. You cannot use the same point file for more than one robot.

Robot Configuration file

All of the robot setup parameters except for Hofs, MCofs, and CalPls are stored in a file in the project directory with a .RCF extension. Hofs, MCofs, and CalPls can be saved in separate files for each robot that you run your projects with by using MkVer from the Tools | Maintenance dialog.

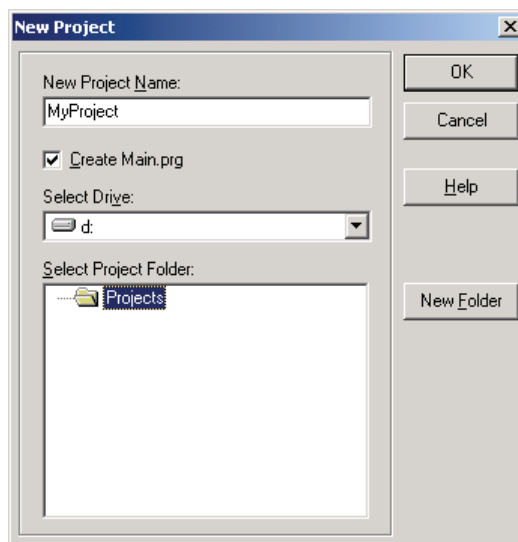
The Default Project

When you create a new project, one program (Main.prg) and one point file (Robot1.pnt) are created for you. Program Groups are disabled. For some users, this is all they need to get started. They don't have to configure the project. They can just type in a program, teach points using the Jog and Teach Tool, and run the program.

If you prefer that main.prg not be created for a new project, you can disable the feature by clicking the **Create Main.Prg** check box off on the New Project dialog.

Creating a new project

Projects can be created on any drive on your system. Projects reside in the \EpsonRC\Projects directory. You can also create subdirectories to organize different types of projects.



To create a new project

1. Select New Project from the Project Menu. The New Project dialog box will appear.
2. Select the disk drive where you want the project to be stored on.
3. Select the project directory or create a new directory by clicking the New Folder button after selecting the parent directory.
4. Type in the name for the new project.
5. If you want to create Main.prg automatically, turn on the Create Main.prg check box.
6. Choose **Ok** to create the project.

Configuring a project

Each application project you create must be configured properly before you can run the application.

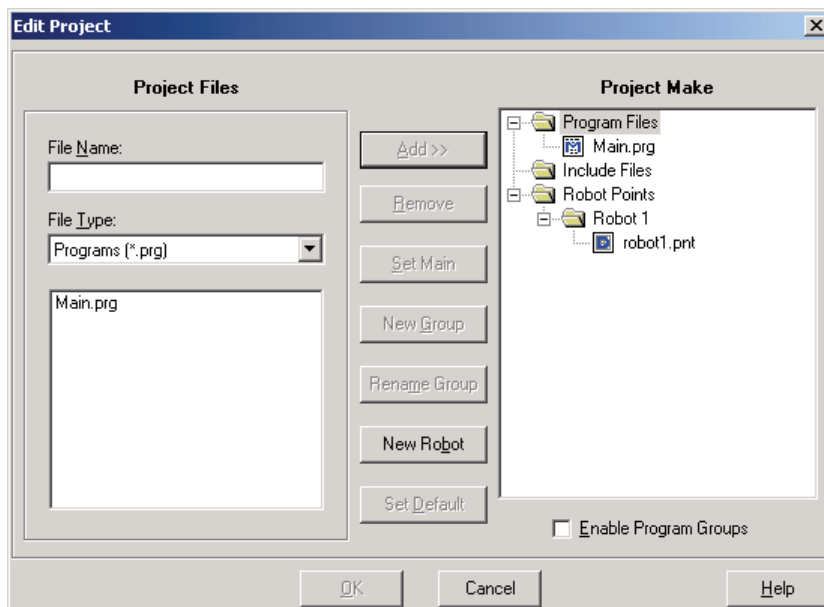
If you are just writing simple applications with one program and one point file, you won't have to configure program groups or point preferences, since they are automatically created for you to include one program and one point file when you execute the New command from the Project menu with the Create MAIN.PRg check box set to on.

For more complicated applications, you need to decide on what you want to accomplish and then configure the program groups and points preferences accordingly. See Designing Applications for details.

There are two commands in the Project menu that allow you to configure a project: Edit and Robot Parameters.

Editing a project

Select Edit from the Project menu to open the Edit Project dialog. From this dialog, you configure which programs, include files, and point files are used in the current project.

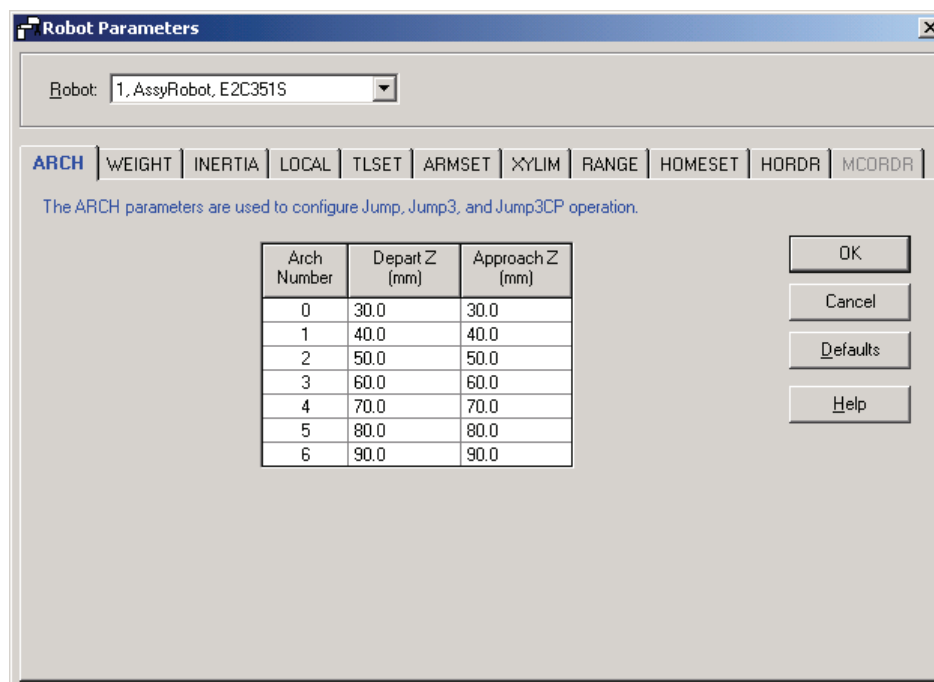


If you are using more than one robot, you need to add robots to the project by using the **New Robot** button. Then add at least one point file for each robot.

For details on Project | Edit, see the Edit Command (Project Menu) in the chapter *EPSON RC+ GUI*.

Project robot parameters


Several parameters for each robot in the current project can be changed from the Project | Robot Parameters dialog. These parameters can optionally be saved with the project by checking the **Maintain Parameters** check box on the Project tab of the Setup | Preferences dialog.



Building a project

Before you can run any of the code in your application, you must build the project.


To build your application project

Select Build from the Project Menu or click on the Build button  on the toolbar.

Or

Select Rebuild from the Project Menu. This will rebuild the entire project.

Or

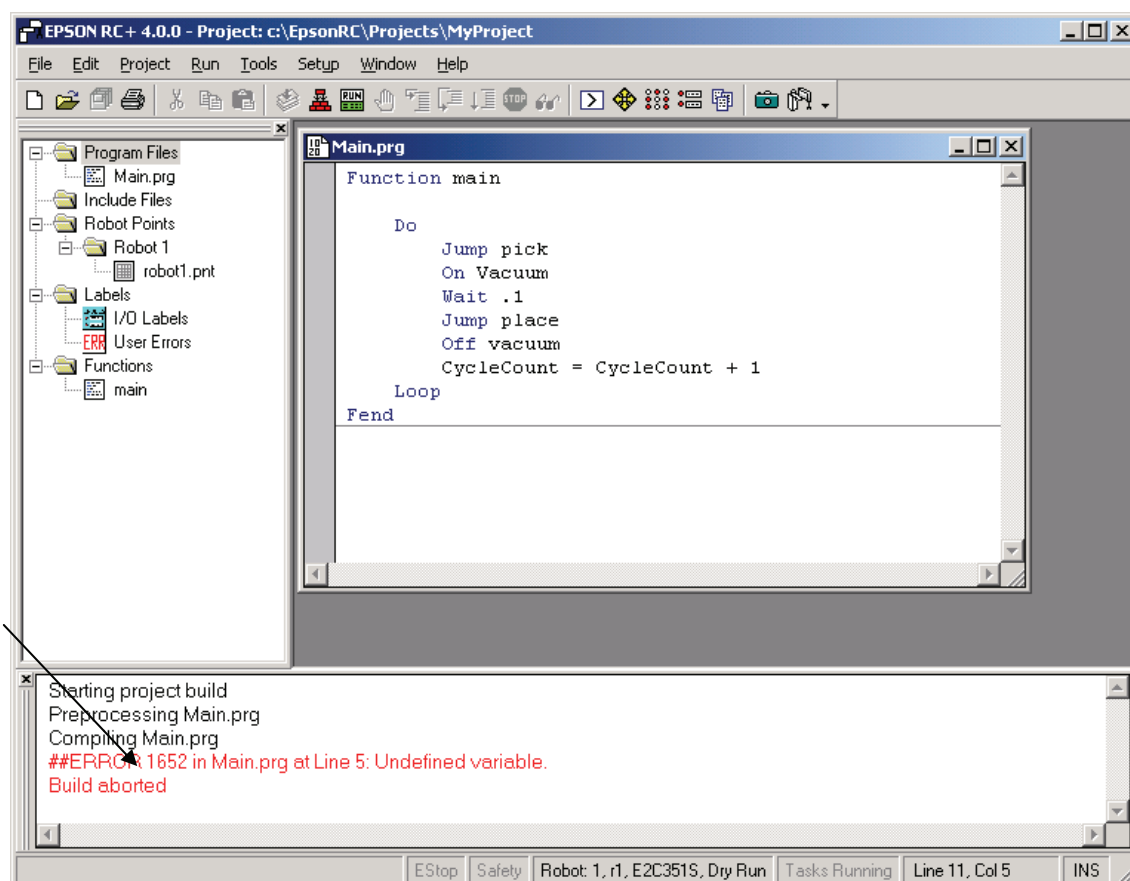
Select Start from the Run Menu or click on the Run button  on the toolbar. The project will be built before the Run window appears.

Or

Select Operator Window from the Run Menu. The project will be built before the Operator window appears.

Status Window

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 name, and line number. Double click on the line with the error to go directly to the source code that caused the error.

Making copies of projects

Use the Copy Project command in the Project Menu to copy all of the file in the current project to another disk drive or directory. You can also save the project under a different name. This command is useful for making backups or for transferring a project to a floppy disk so that it can be loaded on another machine.

Use the SaveAs command in the Project Menu to save the current project under a different name or drive. This command is useful for making a test version of a project without making changes to the original project.

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 disk drive. Typically you will want to copy the project to a floppy disk drive.

To restore a project from a floppy disk (or different hard drive), open the project on the floppy, then use the Copy Project command to copy the project to the original disk

Editing Programs

Before you can edit a program, it must be in the current project and opened in a program editor window. When you first create a new project, this is done for you automatically if the **Create MAIN.PRg** box is checked in the New Project dialog box. You can open several program editors during one session.

To open a program for editing

1. Select Open from the File Menu.
2. Select the file(s) you want to open.
3. Choose **OK** to open the file.

Program rules

A program can contain one or more SPEL⁺ Function definitions.

Each line in a program can optionally start with a line number. Lines can also be blank. You can insert any number of blank lines to separate subroutines and functions, if desired.

Maximum length for each line is 200 characters, including the line number, if used.

Maximum line number is 32767. You will get an error if you try to run a program with a line number that is too large. If you don't use line numbers, you can create a program within the controller's program memory capacity.

If you are executing functions from another program file in the project, the program must be in the same program group.

Typing in program code

Each line can optionally start with a line number. If the **Auto Line Number** feature is enabled (default), line numbers will be created for you automatically for each new line added to the end of the file.

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 semi-colons and commas.

When you insert new lines and you are using line numbers, then you must supply the line number. Type the number. Then when you press SPACE or TAB, the number will be formatted with leading zeros if necessary.

Consider using mixed case or lower case for variables and function names instead of all CAPS. This could 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 While statements.

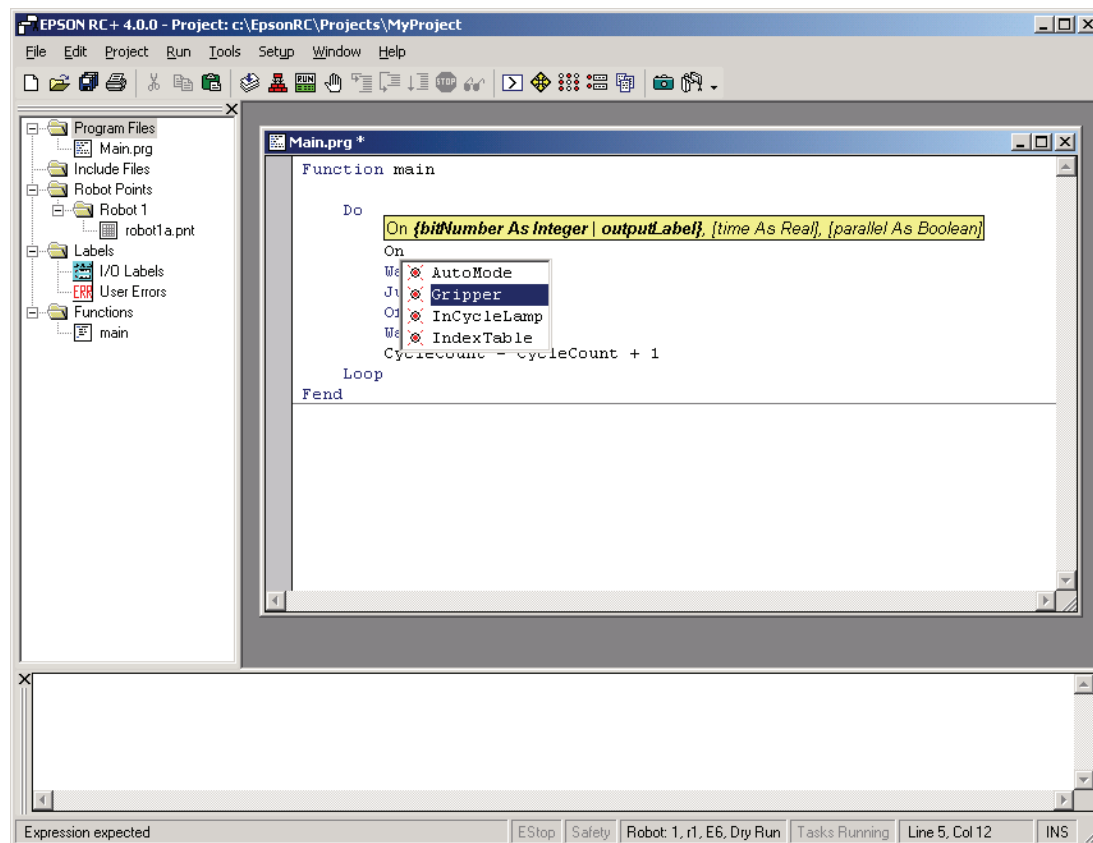
```
For i = 1 To 10
    Jump P(i)
    Jump P0
Next i
```

Use the **Auto End Construct** feature to automatically add the end construct statement. For example, when you enter a For statement and press Enter, a Next statement is automatically created with an indented blank line above it.

Syntax Help

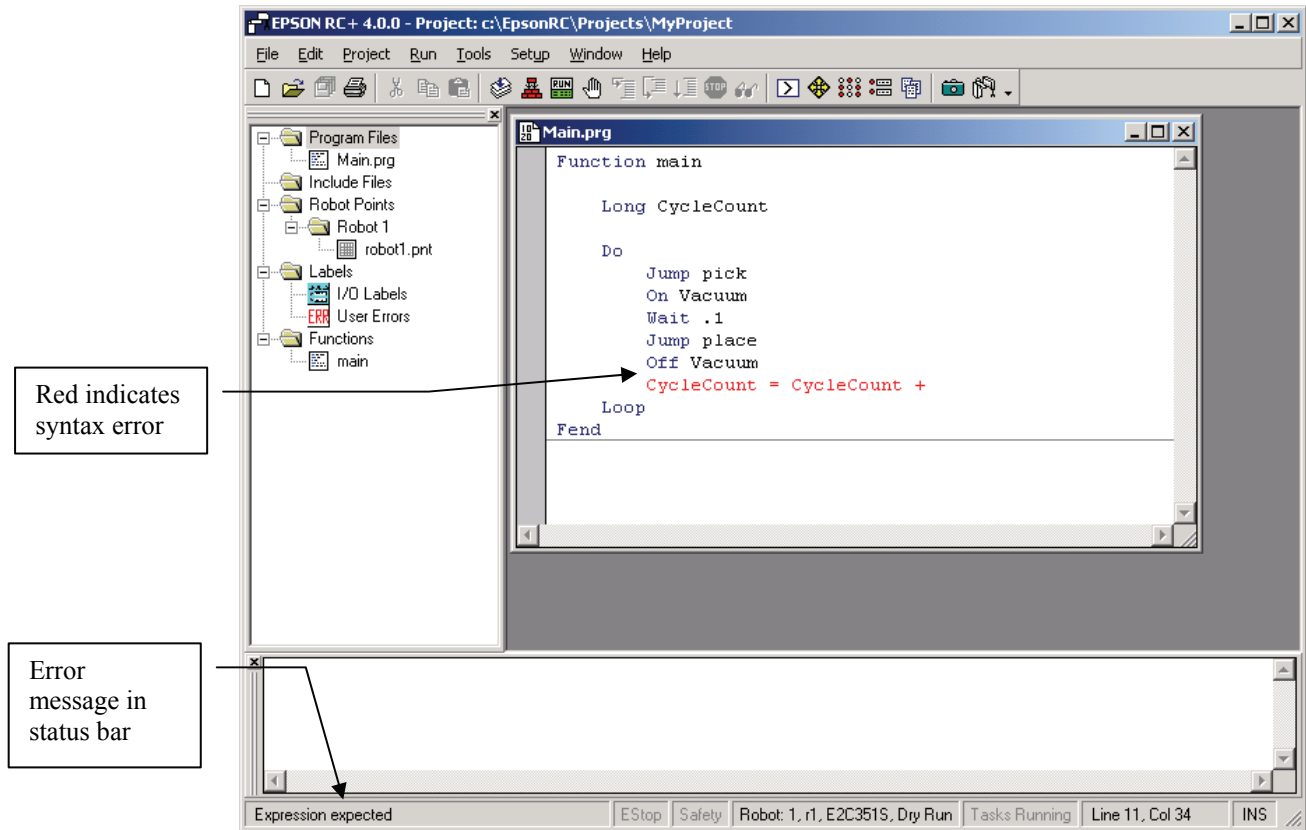
When you type in a SPEL keyword or user function, the syntax help window will appear to show the syntax of the statement or function. After the statement is entered, the syntax helper will automatically close, or you can press the Esc key to close it. You can enable / disable Syntax Help from the Setup | Preferences | Editor tab.

A list box will be displayed for some parameters as you type. To select a value in the list, use the up and down arrow keys, or type the first few characters, to highlight the desired item, then press Tab to select the item. 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 addition to Tab, you can use a comma or period to select an item. In the example shown below, the first parameter of the On statement can be an output label, so a list of output labels in the current project is displayed.



Syntax Errors

When a syntax error is detected, the line with the error will be displayed in red. If the caret is placed on the line with the error, then a brief message will be displayed on the status bar. For example, in the program shown below, the message "Expression expected" is displayed on the status bar.



Renumbering

Select Renumber from the Edit menu to renumber the program.

The line increment used can be set with the Preferences: Editor tab command from the Setup Menu.

You can also turn on **Auto renumber before save** in the editor preferences so that every time you save a program file, it will be renumbered.

If a program was originally created without line numbers, you can use the Renumber command to add numbers to the program.

Removing line numbers

Select Strip line numbers from the Edit menu to remove line numbers from a program. Line numbers that are referred to by GoTo, GoSub, and On Error will be replaced with labels.

Editing Points

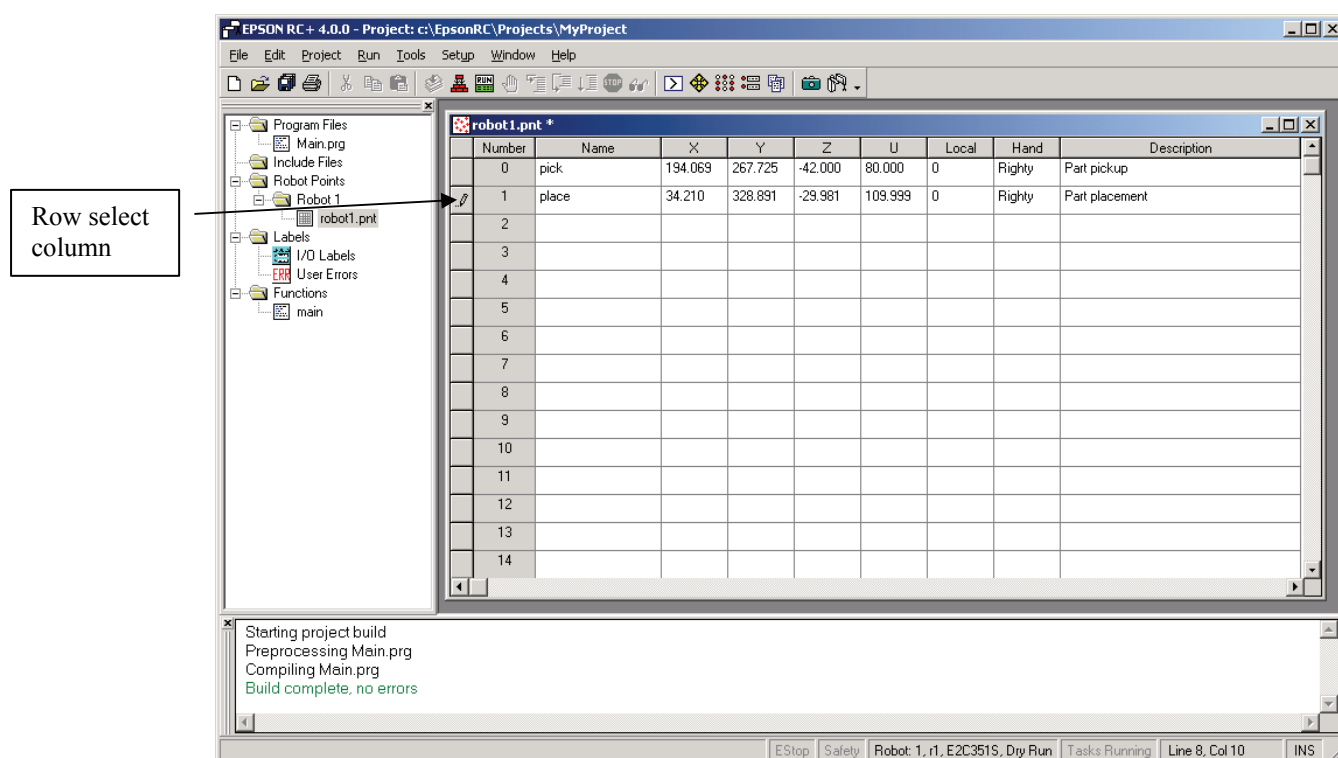
EPSON RC+ provides a spreadsheet editor for robot points. You can define new points or cut, copy, and paste points from one point file to another, including between projects.

Robot teach point definitions are kept in files on your PC and in the robot controller. EPSON RC+ always keeps files in the robot controller up to date with the files on the PC.

To open a point file for editing

1. Select Open from the File Menu to show the Open dialog box.
2. Choose the **Points** radio button. You will see a list of point file names in the files list box.
3. Select the point file you want to edit by clicking on the name.
4. Press the **Open** button to open the file. You will see a spread sheet window for the point file you selected.

The robot points spread sheet window



The spreadsheet window contains one row for each point in the file. The spreadsheet always contains rows for all points, even if they are not defined. The cells for an undefined point are blank.

Row select column

This is the first column on the left. Click on this column to select a row.

Point number column

Point number. Range is from 0 to the max. number of points.

Name column

Name of the point.

Coordinate columns

Coordinates in millimeters.

Local number column

Local number drop down list. Range is from 0 to 15.

Hand column

Drop down list with two values of robot orientation: LEFTY, and RIGHTY.

To select one or more rows

Click on the row select column (first column on the left) to select a row. 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.

To select all rows

Execute Select All from the Edit menu, or type Ctrl+A.

To define a new point

Using the mouse put the spreadsheet cursor anywhere on the row of the point you want to define and start entering 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, put the cursor in the Name column and type in a name for the point.

Press the TAB key to move to the X coordinate column. Type a coordinate value, then press Enter. You will see zeros automatically entered for all of the other coordinates of the point. This indicates that the point as been defined.

To delete a point

Select the row containing the point and cut it by selecting Cut from the Edit menu or by typing Ctrl+X.

To cut and paste points

1. Select one or more rows and execute either the Cut or Copy command from the Edit menu.
2. Select the row where you want to start the paste.
3. Execute the Paste command from the Edit menu.

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.

To run a program


Select Start 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 program group and also which program to run in the selected group.

The Run Window

Description

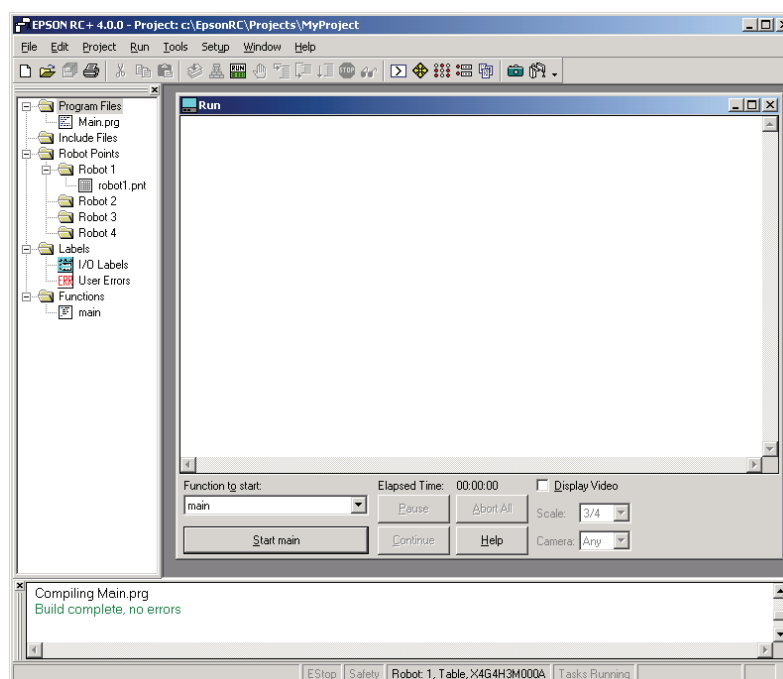
The Run window includes controls for running the programs in the current project.

To open the Run window

Select Start from the Run menu, or click on the Run button  on the toolbar. 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 double click on the Control menu in the upper left hand corner of the window.



Run window controls

Control Description

Text area	<p>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.</p> <p>If an error occurs while running a program, the error number, 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. You can also get help on the error that occurred by putting the cursor anywhere on the line and pressing F1.</p>
Program group list	<p>Select the program group you want to run a program from. Note: Groups must be enabled for the current project for this list to be visible. See the Edit Command in the Project Menu.</p>
Function list	<p>Select a function in the current program group to start.</p>
Elapsed time display	<p>Each time you Start a program with the Start button, the timer is reset to zero and begins timing the cycle. When all tasks have been stopped, the timing stops. This is useful for overall cycle time measurement. Time is displayed in seconds.</p>
Start button	<p>Starts the function shown in the function drop down list.</p>
Pause button	<p>Pause 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.</p>
Continue button	<p>Continue paused tasks. Operation depends on the setting for the SPEL Option <i>Auto recover position after safeguard</i> preference. See the section <i>Safeguard Position Recovery</i> in the chapter <i>Safety</i> for more information.</p>
Abort All button	<p>Stops all tasks. If the robot is executing a motion command when this button is pressed, the robot will decelerate to a stop.</p>
Display Video check box	<p>Check this box to display video. You can check video on or off anytime, including while tasks are running. Not available if the Vision Guide option is not enabled.</p>
Scale dropdown list	<p>Select the video scale from this list. You can change scale anytime, including while tasks are running. The scale is saved between sessions. Not available if the Vision Guide option is not enabled.</p>
Camera dropdown list	<p>Select the camera to display from this list. The default is Any, so every vision sequence is displayed when VRun executes. If you select a camera number, then only sequences that use that camera will be displayed. You can change the camera to display anytime, including while tasks are running. This setting is saved between sessions. Not available if the Vision Guide option is not enabled.</p>
CTRL+C keys	<p>Same as Abort All button.</p>

Debugging

EPSON RC+ supports source level debugging. You can set breakpoints and step through your source code. You can also suspend a task using the Task Manager Halt command and then step through the code.

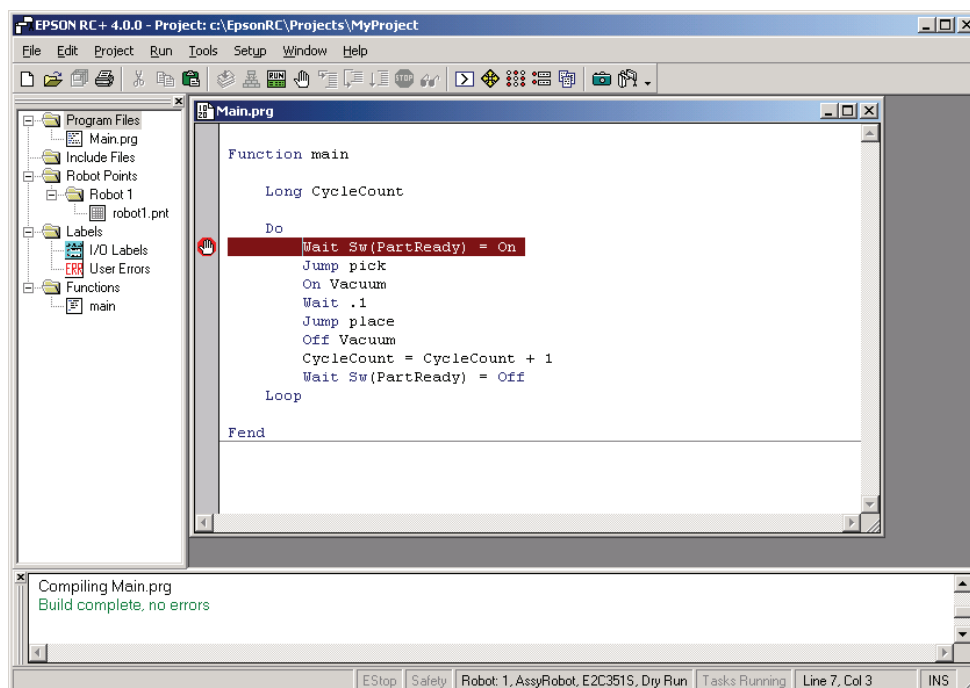
Setting and clearing breakpoints

Open the program where you want to set a breakpoint, then click on the line where you want to stop. Use one of the following methods to set a breakpoint:

- If Margin Indicators are enabled, then click in the margin next to the line. You will see a breakpoint symbol next to the line.
Or
- Type F9.
Or
- Select Toggle Breakpoint from the Run menu.

Execute one of the methods above to clear a breakpoint, or select Clear All Breakpoints from the Run menu.

You cannot set a breakpoint on non-executing statements, such as `#define`, `#include`, or blank lines. You can set a breakpoint on a comment line if it is inside a function.



After stopping at a breakpoint, you can set/clear other breakpoints.

Stepping through a program

There are two 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.

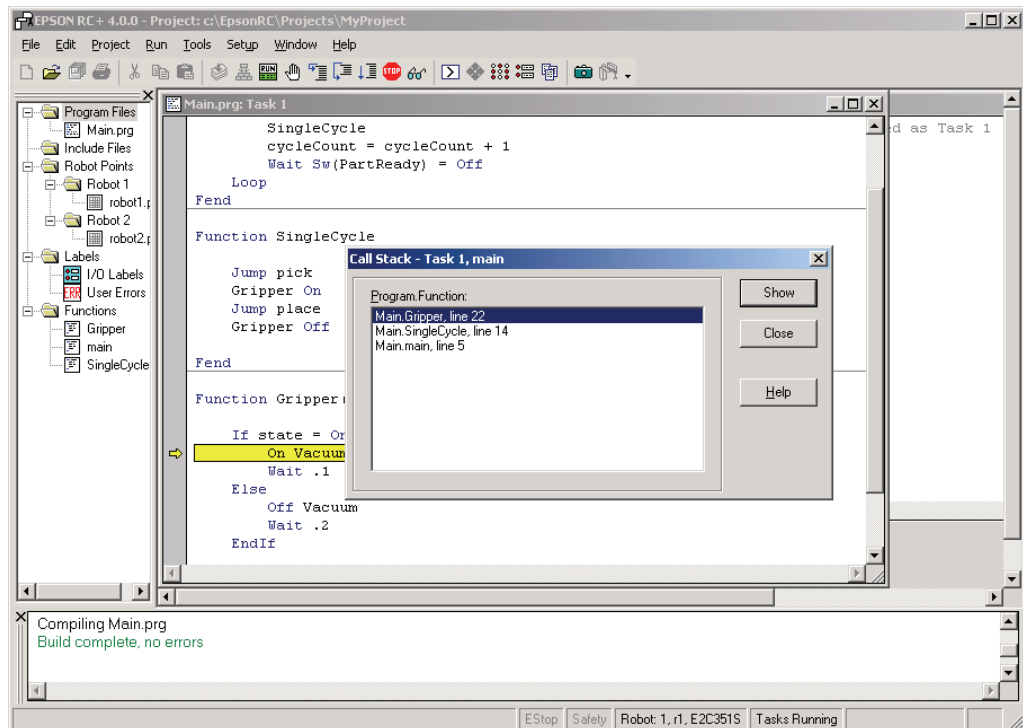
Step Over steps through each line but when a Call statement is encountered, the function in the statement is executed completely.

To step through code, you must either set a breakpoint and run until the breakpoint is reached, or suspend a task from the Task Manager.

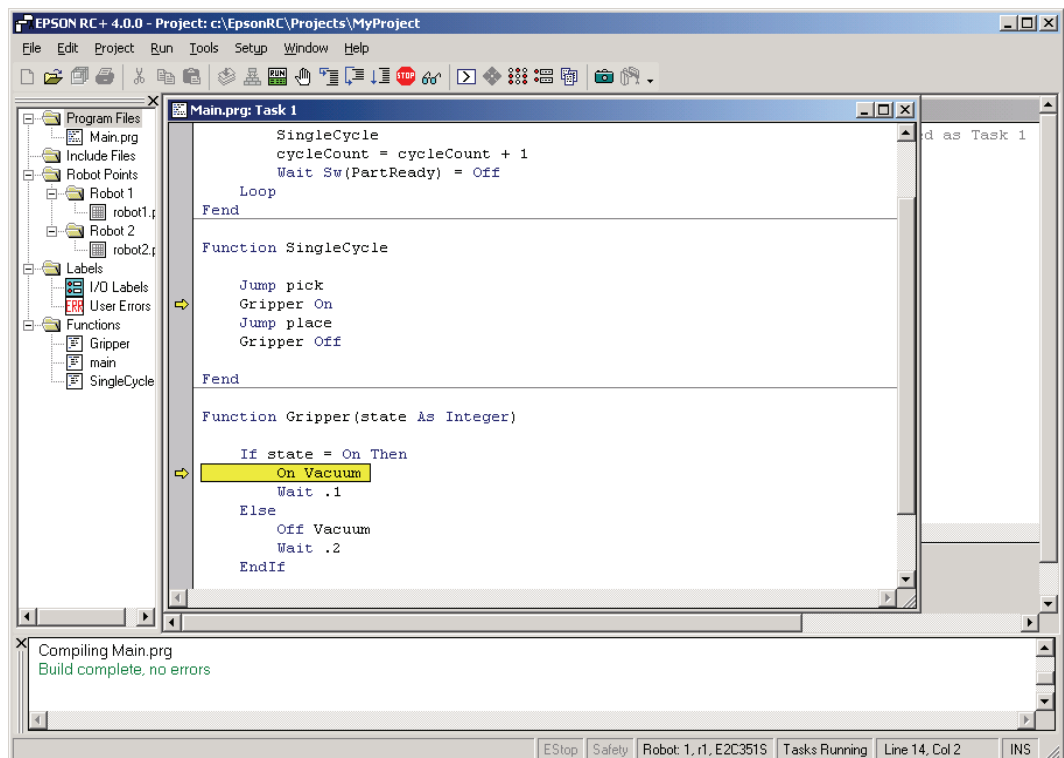
Viewing the Call Stack

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.



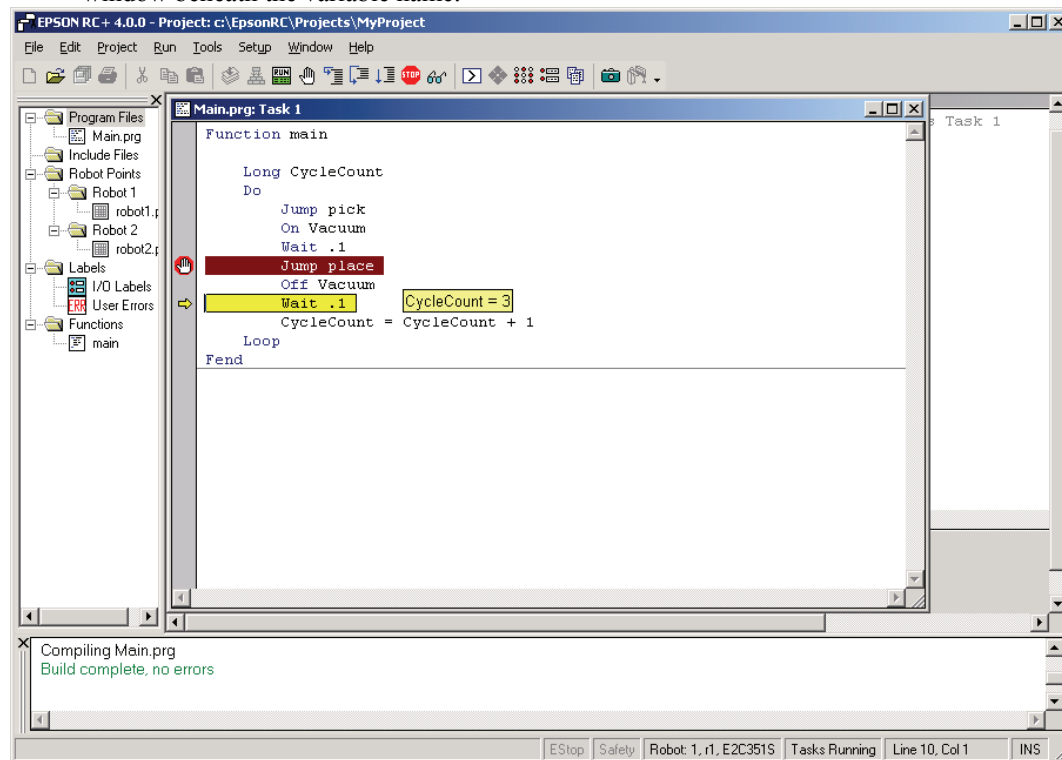
After you select a function in the Call Stack list, the function will be displayed in a program window and an arrow in the left margin will point to the line where the next function in the call stack is being called. In the example below, the arrow in the SingleCycle function is pointing to the Gripper On statement to indicate that Gripper was called from SingleCycle.



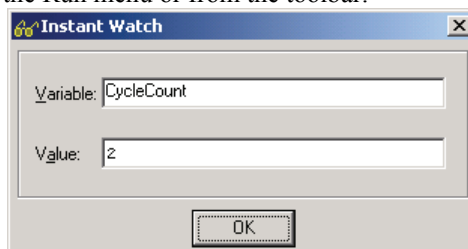
Displaying variables

To view variable values, you can do one of the following:

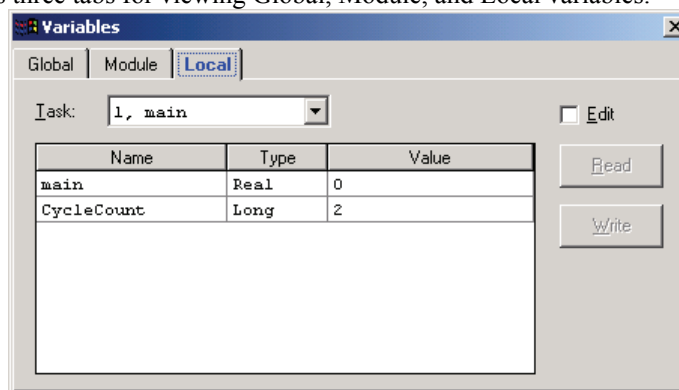
1. When a task is halted by halt or breakpoint, you can view the value for a variable by moving the mouse cursor over the variable name. The value will be displayed in a tool tip type window beneath the variable name.



2. When a task is halted by halt or breakpoint, click anywhere on a variable name, then select Instant Watch from the Run menu or from the toolbar.



3. Select Display Variables from the Run menu to display the variable display dialog. This dialog has three tabs for viewing Global, Module, and Local variables.

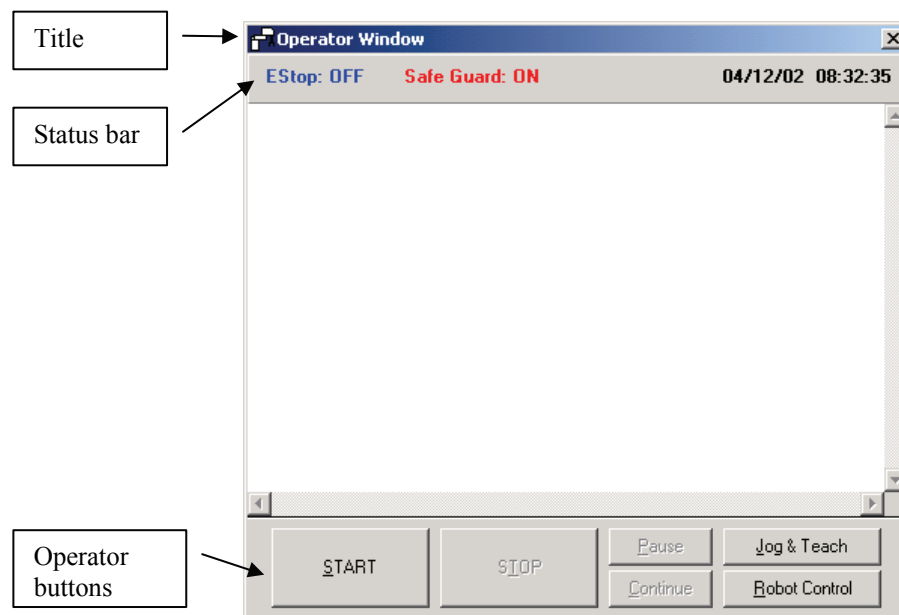


You can change the value of a variable by checking the Edit check box, then type in the new value in the value column. Next, click the Write button to change the variable. When the Edit box is

checked, the variable values are not automatically updated. You can click the Read button to update all values.

The Operator Window

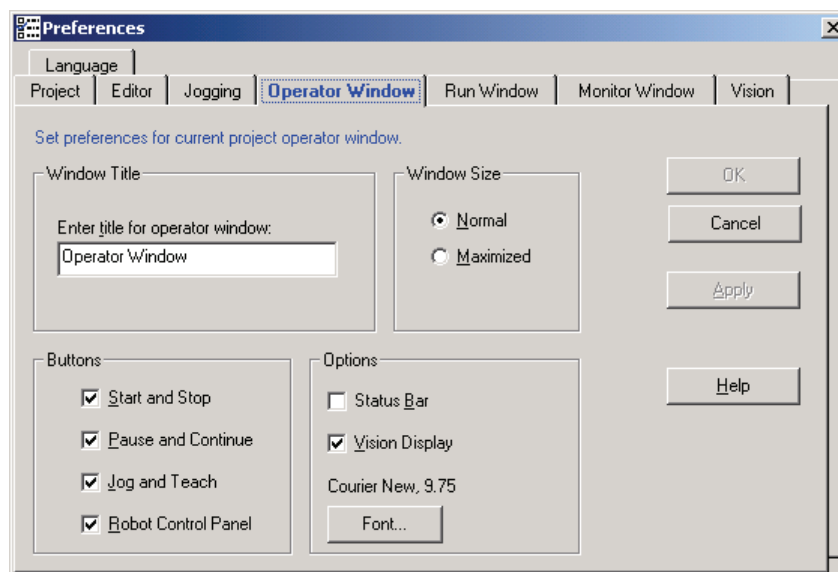
The Operator Window can be used as a simple interface for operators. You can configure EPSON RC+ to open only the Operator Window when started. In addition, when Remote Control is being used, the Operator Window will be displayed, with all buttons hidden.



Control	Description
START button	Start the first function in the main program of the first group.
STOP button	Stops all tasks.
Pause button	Pause all tasks that are enabled for pause.
Continue button	Continue paused tasks. Operation depends on the setting for the SPEL Option <i>Auto recover position after safeguard</i> preference. See the section <i>Safeguard Position Recovery</i> in the chapter <i>Safety</i> for more information.
Jog & Teach button	Opens the Jog and Teach dialog.
Robot Control button	Opens the Robot Control Panel.

Operator Window Configuration

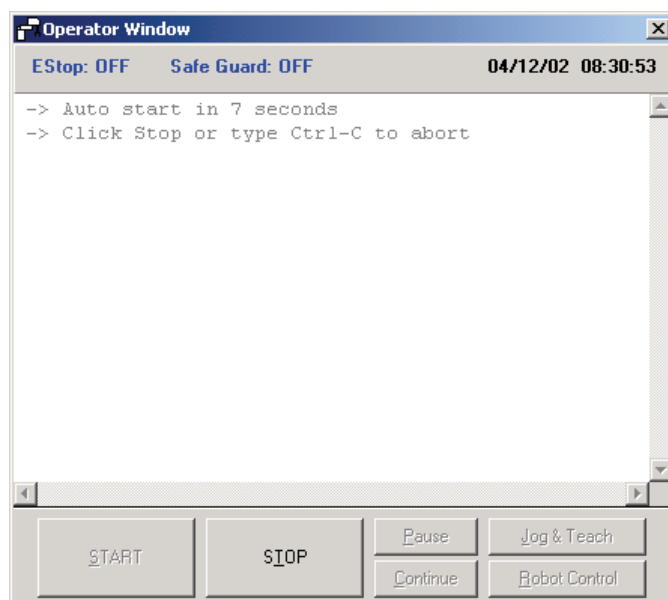
You can configure the Operator Window from the Operator Window tab in Setup | Preferences.



See the chapter *EPSON RC+ GUI* for more details.

Configuring start up for Operator Window

The system can be configured to automatically log into Windows and start EPSON RC+ in one of the operator modes. You can also configure the Operator Window to auto start the program. See the section *Start Up Configuration* in the chapter *Operation* for details.



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 the chapter *Remote Control* for details.

Motion Systems

EPSON RC+ supports the motion systems listed below. Refer to the appropriate section for the system you are using.

- RC420 Motion System
- RC520 Motion System
- PG Motion System

RC420 Motion System

The RC420 Motion System is built into the RC420 Controller. For details on the RC420 hardware and maintenance, please refer to the RC420 Controller manual.

RC420 Drive Module Software Configuration

The RC420 Drive Module is configured at the factory before shipment.

NOTE



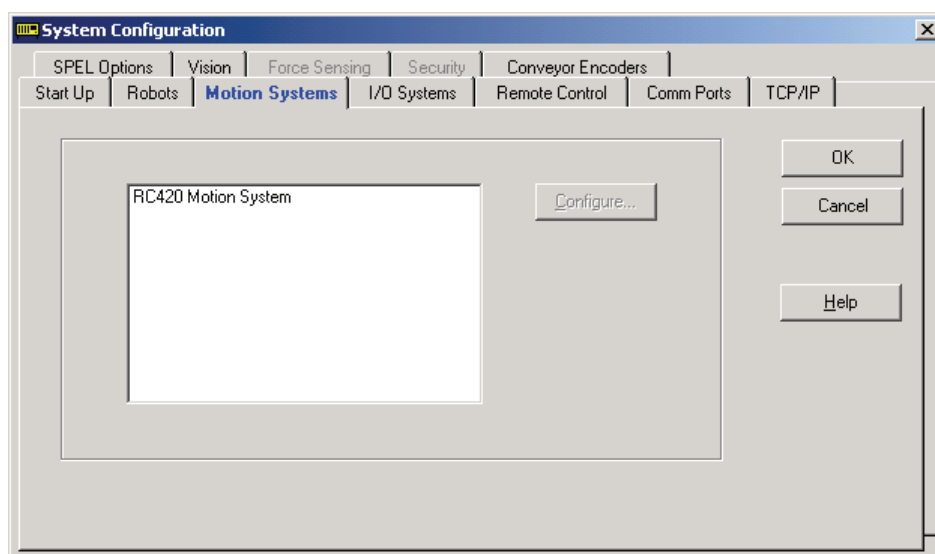
You must properly configure the drive module so that the Robot can be added to the system. It is very important that all information is correct, especially Servo power settings.

Configuring the drive module

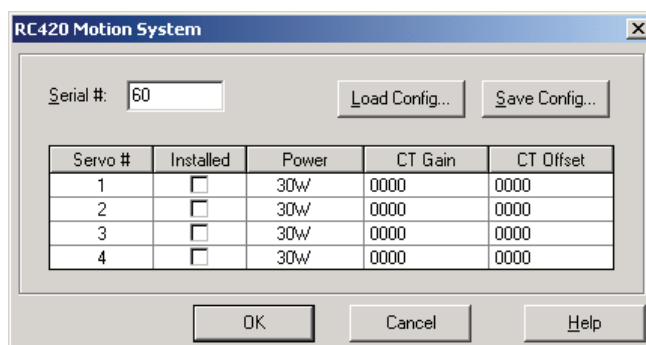
Follow these steps to configure the drive module:

1. If you have not already done so, execute MKVER from the Tools | Maintenance menu to backup the current configuration.
2. Make sure that the jumpers are set correctly on the MIB board.
3. Start EPSON RC+.
4. Select System Configuration from the Setup menu.

- Click the **Motion Systems** tab.



- Click on the **RC420 Motion System** item in the list box.
- Click the **Configure** button. The RC420 Motion System dialog will be displayed as shown below.



- Enter the Gain and Offset parameters for each amplifier. These parameters are printed on each amplifier and can be observed by removing the Controller top cover. If you have a configuration file, you can use the **Load Config** button to load the file.
- Click **OK** to close the dialog.
- You can now click on the Robots tab to add a robot to the system. Or you can click **OK** now to save the settings so you can add the robot later.

Saving drive module configuration

You can save the configuration data for the drive module in a file. The file is created with a DUD extension. This file contains all of the amplifier settings for the drive module.

Follow these steps to save configuration:

- Select System Configuration from the Setup menu.
- Click the Motion Systems tab.
- Click on the **RC420 Motion System** item in the list box.
- Click the Configure button. The RC420 Motion System dialog will be displayed.
- Ensure that the serial number of the Controller is correct. Change it if necessary.
- Click the **Save Config** button. Browse to a floppy disk or network directory and click Save.

Loading drive module configuration

Follow these steps to load configuration:

1. Select System Configuration from the Setup menu.
2. Click the **Motion Systems** tab.
3. Click on the **RC420 Motion System** item in the list box.
4. Click the **Configure** button. The RC420 Motion System dialog will be displayed.
5. Click the **Load Config** button. Browse to the desired file and click Open.



If you attempt to load a drive unit configuration that has different power settings, you will receive an error message if a robot is using the drive unit.

RC520 Motion System

The EPSON RC520 Motion System consists of a Motion Interface Board (MIB) that mounts in an ISA slot in the PC and a drive unit that contains up to four amplifiers. Up to three MIBs can be used in one system.

For details on the RC520 hardware and maintenance, please refer to the RC520 Controller manual.

RC520 Drive Unit Software Configuration

The EPSON RC+ system can support up to three RC520 Drive Units. Normally, there is one robot per drive unit. There are two types of drive units: RC520DU and RC520DU6. The RC520DU Drive Unit can contain up to 4 servo amplifiers. The RC520DU6 Drive Unit is used exclusively for 6-axis manipulators and contains 6 servo amplifiers. Each amplifier controls one robot motor. For each Drive Unit, there must be one MIB (Motion Interface Board) in the PC.

All Drive Units are configured at the factory before shipment.

The Drive Unit 1 configuration cannot be disabled.



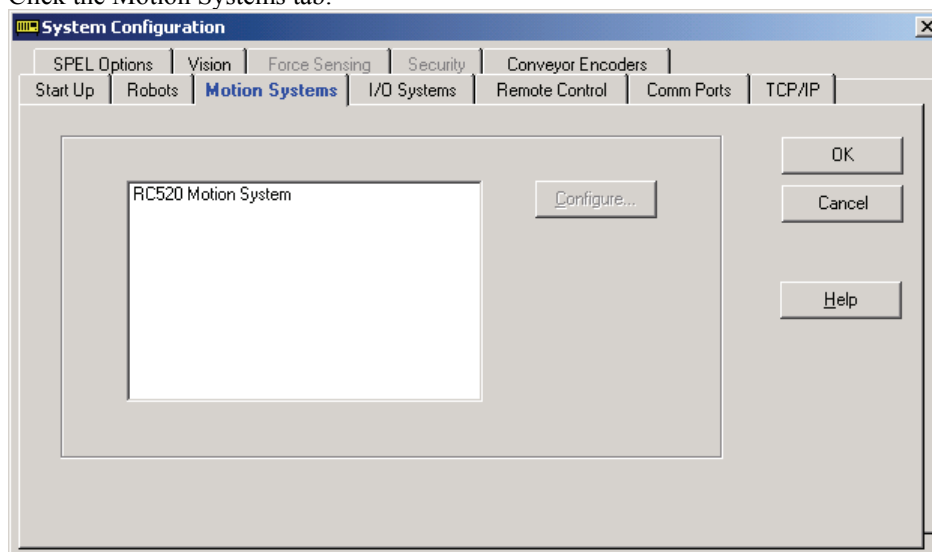
You must properly configure the drive units attached to the PC so that Robots can be added to the system. It is very important that all information is correct, especially Servo power settings.

Adding a Drive Unit

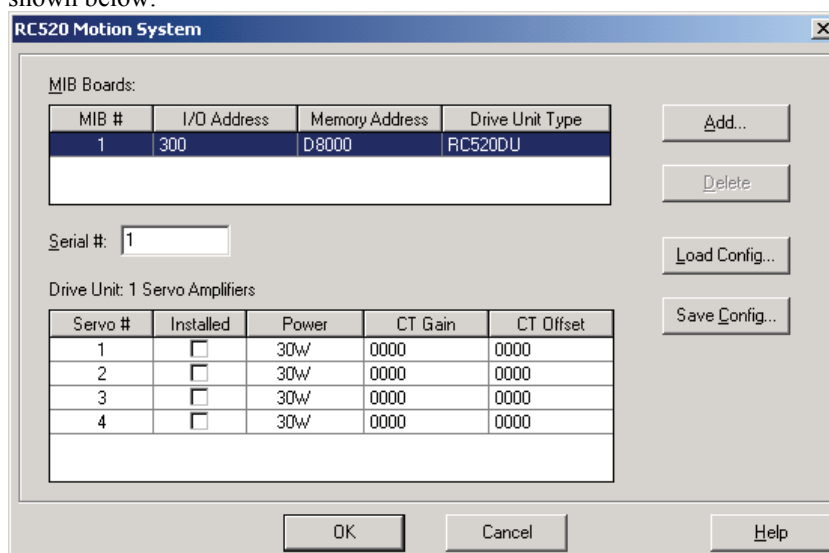
Follow these steps to install an additional Drive Unit:

1. If you have not already done so, execute MKVER from the Tools | Maintenance menu to backup the current configuration.
2. Install the Drive Unit and MIB board. Make sure that the jumpers are set correctly on the MIB board.
3. Start EPSON RC+.
4. Select System Configuration from the Setup menu.

5. Click the Motion Systems tab.



6. Click on the RC520 Motion System item in the list box.
7. Click the Configure button. The RC520 Motion System dialog will be displayed as shown below.



8. Click the **Add** button to add a new drive unit.
9. The **I/O Address** and **Memory Address** fields for the new drive unit will be displayed. These are the addresses that the corresponding MIB board should be set to.
10. Click in the Drive Unit Type field, then click the arrow to display the selections. Select RC520DU or RC520DU6. Check the drive unit for the correct model number.
11. Click the Installed check box for each amplifier that is installed.
12. Click in the Power field for each amplifier, then click the arrow to display the selections. Select the power used for each amplifier (units are in watts).
13. Enter the Gain and Offset parameters for each amplifier. These parameters are printed on each amplifier and can be observed by removing the drive unit top cover. If you have a configuration file, you can use the **Load Config** button to load the file that includes these settings.
14. Click **OK** to close the dialog.

15. You can now click on the Robots tab to add a robot to the system that will use the drive unit you have just added. Or you can click **OK** now to save the settings for the new drive unit so you can add a robot later.

Removing a Drive Unit

Follow these steps to remove a Drive Unit:

1. Start EPSON RC+.
2. Select System Configuration from the Setup menu.
3. Click the Robots tab.
4. Select the robot that was using the drive unit.
5. Click the **Delete** button to remove the robot.
6. Click the Motion Systems tab.
7. Click on the **RC520 Servo System** item in the list box.
8. Click the **Configure** button.
9. Select the drive unit number you want to remove.
10. Click the **Delete** button.
11. Click **OK** to close the dialog then OK again to save the settings.
12. Shutdown the system and remove the MIB board from the PC. Refer to the Controller manual for details.

Saving drive unit configuration

You can save the configuration data for each drive unit in a file. The file is created with a DUD extension. It contains all of the amplifier settings for one drive unit.

Follow these steps to save configuration:

1. Select System Configuration from the Setup menu.
2. Click the **Motion Systems** tab.
3. Click on the **RC520 Motion System** item in the list box.
4. Click the **Configure** button. The Motion System dialog will be displayed.
5. Ensure that the serial number of the drive unit is correct. Change is if necessary.
6. Click the **Save Config** button. Browse to a floppy disk or network directory and click Save.

Loading drive unit configuration

Follow these steps to load configuration:

1. Select System Configuration from the Setup menu.
2. Click the **Motion Systems** tab.
3. Click on the **RC520 Motion System** item in the list box.
4. Click the **Configure** button. The **RC520 Motion System** dialog will be displayed.
5. Click the **Load Config** button. Browse to the desired file and click Open.



If you attempt to load a drive unit configuration that has different power settings, you will receive an error message if a robot is using the drive unit.

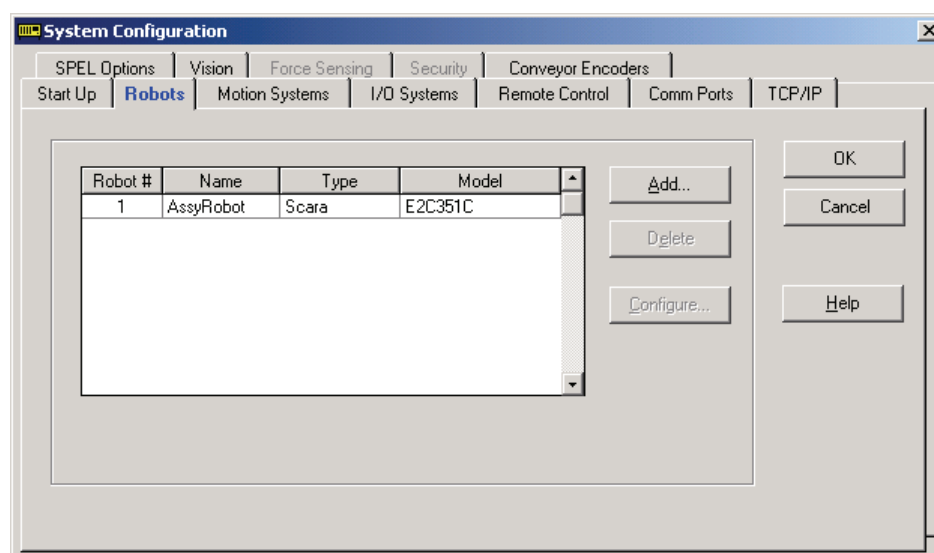
PG Motion System

The PG (Pulse Generator) Motion System is an option. For instructions on using the PG Motion System, refer to the PG Motion System Manual.

Robot Configuration

This chapter contains information for adding, modifying, and deleting robots installed on the system.

Robots are configured from the Robots tab on the System Configuration dialog.



Adding a Robot

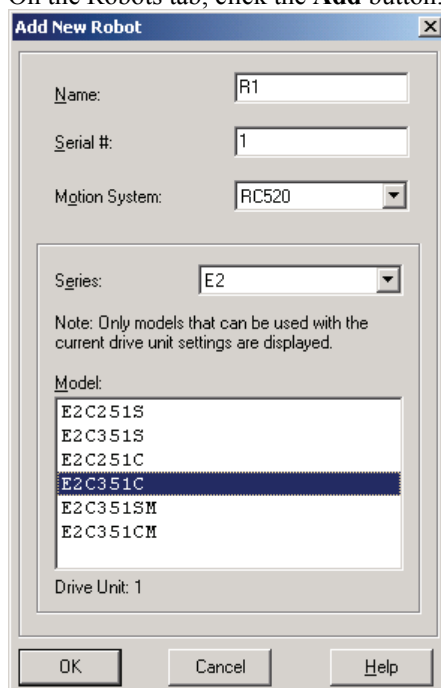
Adding Standard Robots

Before adding a standard EPSON robot to an RC520 system, you must add a Drive Unit to the system to control the robot. For RC420 controllers, the controller itself is considered to be a drive unit, since the amplifiers are built in to support one robot. See Drive Unit Software Configuration in this chapter for details. The Drive Unit must have amplifiers with the proper wattage for the robot you want to add.

To add a robot, follow these steps:

1. Select System Configuration from the Setup menu.

2. On the Robots tab, click the **Add** button. The following dialog is displayed.

The image shows a Windows-style dialog box titled "Add New Robot". It contains several input fields: "Name:" with the text "R1", "Serial #:" with the text "1", and "Motion System:" with a dropdown menu showing "RC520". Below these is a "Series:" dropdown menu showing "E2". A note states: "Note: Only models that can be used with the current drive unit settings are displayed." Below the note is a "Model:" list box containing the following items: "E2C251S", "E2C351S", "E2C251C", "E2C351C" (which is highlighted), "E2C351SM", and "E2C351CM". At the bottom of the list box area, it says "Drive Unit: 1". At the very bottom of the dialog are three buttons: "OK", "Cancel", and "Help".

3. Type in a name for the new robot and enter the serial number from the robot's nameplate. Any serial number can be used, but it is recommended that you use the number that is stamped on the robot.
4. Select the RC520 or RC420 motion system, depending on controller type. If there are no other motion systems installed, then the correct RC520 or RC420 motion system will already be selected.
5. Select the desired robot series. After you select a series, all models for that series for the amplifier wattages specified previously will be shown in the Model list.
6. Select the robot model you want to add. If you don't see the robot you want to add, then the Drive Unit does not have the correct amplifiers installed, or the Drive Unit is configured incorrectly. Click Cancel and check the Drive Unit configuration.
7. Click **OK**. The SPEL Runtime drivers will restart and the new robot should be installed.
8. Verify the new robot is installed by selecting Robot Control Panel from the Tools menu. You should see the new robot in the Robot drop down list.

Adding Pulse Generator Robots

If you have purchased the PG Motion System Option, you can add user defined robots. See the PG Manual for details.

Deleting a Robot

To delete a robot, follow these steps:

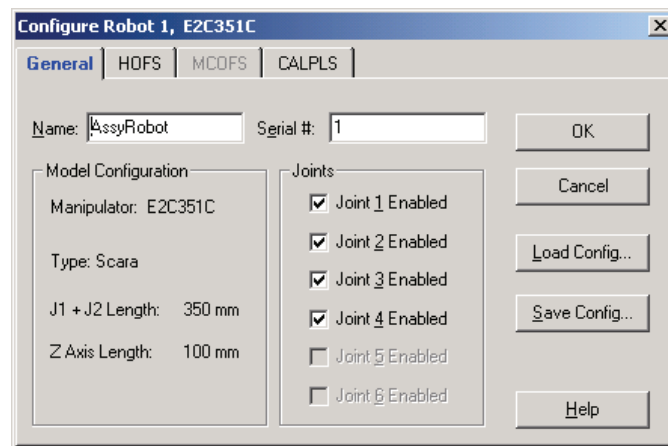
1. Select System Configuration from the Setup menu.
2. On the Robots tab, select the robot you want to delete from the list.
3. Click the **Delete** button.
4. Click **OK**. The SPEL Runtime drivers will be restarted.

Changing Robot Parameters

The following parameters for the robot can be changed from the Configuration dialog:

- **Enable/Disable Joints**
You can disable one or more joints. On robots with ball screw Z axis, you must disable both joints 3 and 4 together.
- **MCofs** (for robots with incremental encoders only)
MCofs are the MCal calibration offsets and are normally generated by the MCOrg command. From this dialog, you can change the parameters manually.
- **Hofs**
Hofs are the joint home offsets. 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 offsets. 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 / encoder.

These are one-time settings for each robot. Additional robot parameters can be set from the Robot Parameters selection in the Project menu.



To change robot parameters, follow these steps:

1. Select System Configuration from the Setup menu.
2. On the Robots tab, select the robot to configure.
3. Click the Configure button.
4. Click the General tab to enable/disable joints, or click Hofs, MCOfs or CalPls to change these values.
5. Click OK to make the changes permanent.

Saving a robot configuration

You can save and load individual robot configuration files. This is useful for moving a robot from one system to another. When you save configuration, a file is created with an MPD file extension. This file contains Hofs, Mcofs, and CalPIs values.

To save a robot configuration:

1. Select System Configuration from the Setup menu.
2. On the Robots tab, select the robot to save configuration for.
3. Click the Configure button.
4. Click the General tab.
5. 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.
6. Click the Save Config button. Browse to a destination directory and click Save.

Loading a robot configuration

To save a robot configuration:

1. Select System Configuration from the Setup menu.
2. On the Robots tab, select the robot to configure.
3. Click the Configure button.
4. Click the General tab.
5. Click the Load Config button. Browse to the desired MPD file and click Open.

Moving a robot between systems

To move a robot between systems:

1. Save the robot configuration as described in the previous section *Saving a robot configuration* to a floppy disk or network directory.
2. Physically move the robot to the new system and connect the cables.
3. Add the robot as described previously in *Adding a Robot*. Note that the drive unit to be used by the robot has the correct motor power amplifiers. If the amplifier power is wrong, you will not be able to add the robot.
4. Load the previously saved robot configuration (MPD) file.

I/O Systems

The EPSON RC+ system supports different types of I/O systems, including local digital I/O, and Ethernet I/O.

The chapter contains instructions for hardware and software configuration for each I/O system.

Standard Digital I/O

Overview

The Standard ISA Digital I/O Boards provide local digital I/O on the PC Control Unit. Each card uses optical isolation and supports 32 inputs and 32 outputs. EPSON RC+ supports up to 10 boards.



NOTE

There are two types of Standard ISA Digital I/O Boards:

NPN Type: This type does NOT conform to CE requirements.

PNP Type: Use this type for CE conformance.

Input Response Time

For fastest input response time, use the Wait statement instead of polling. The Wait statement causes the task to sleep until an interrupt occurs that detects the input condition and wakes the task immediately to continue execution.

For example, this statement will cause the task to sleep until myinput turns on.

Do

```
Wait Sw(myinput) = On
```

```
' process after input is detected
```

Loop

The following code polls the input in a loop and can sometimes miss the input, depending on how long the input is on. Also, because of the tight loop, more CPU time is used.

Do

```
If Sw(myinput) = On Then
```

```
' process after input is detected
```

```
EndIf
```

Loop

Installing ISA I/O Boards

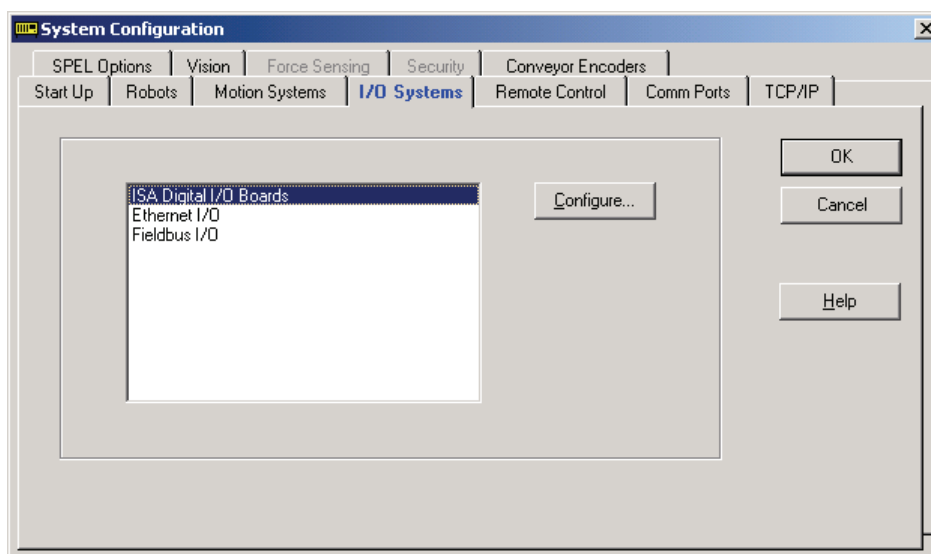
To install the I/O boards, please refer to the RC520 Hardware manual. The I/O bits 0 through 47 are reserved for the three drive units, DU1 through DU3. The input and output bit numbers are assigned as follows. For the Standard ISA I/O boards, the I/O bit numbers start with 48.

Input Bit #	Output Bit #	Applicable Hardware
0 to 15	0 to 15	Drive Unit 1 (DU1)
16 to 31	16 to 31	Drive Unit 2 (DU2)
32 to 47	32 to 47	Drive Unit 3 (DU3)
48 to 79	48 to 79	1st Standard ISA I/O board
80 to 111	80 to 111	2nd Standard ISA I/O board
⋮	⋮	⋮
336 to 367	336 to 367	10th Standard ISA I/O board

ISA I/O Board Software Configuration

You can add up to 10 ISA I/O boards on the system. Each standard I/O board provides 32 DC inputs and 32 DC outputs.

Before you configure the system software, install the I/O boards in the PC Control Unit as described the RC520 Hardware manual.

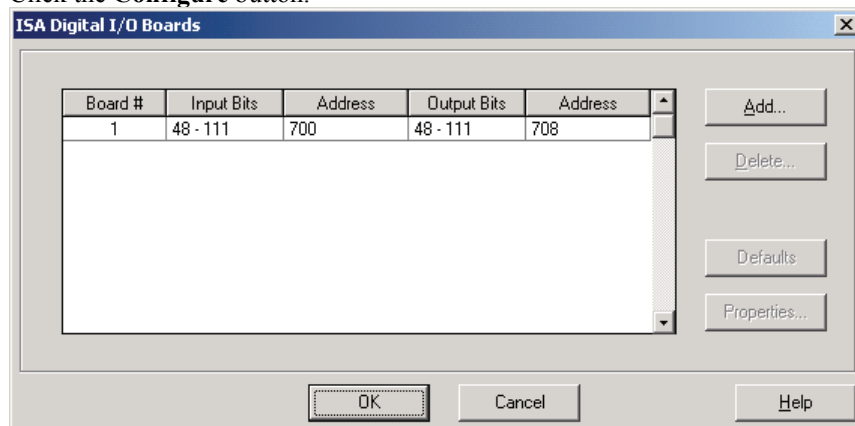


Adding an I/O Board

Follow these steps to add an I/O board to the system.

1. Set the jumpers for the board you want to install. Use the default address settings.
2. Install the board in the PC.
3. Start EPSON RC+.
4. Select System Configuration from the Setup menu.
5. Click the I/O Systems tab.
6. Click on the ISA Digital I/O Boards item in the list.

7. Click the **Configure** button.



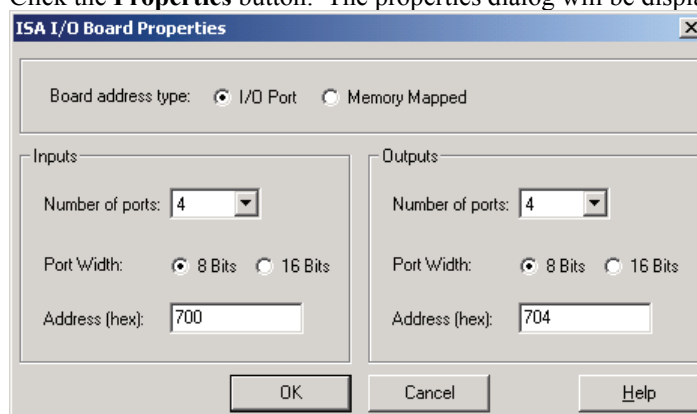
8. Click the **Add** button.
9. The address should match the jumper settings on the board. Default settings are recommended.
10. Click **OK** to close the dialog. Then click OK again to close the System Configuration dialog and save the new changes. The SPEL Runtime drivers will be restarted.

Setting properties for custom I/O boards

When using ISA I/O boards other than the standard boards from EPSON, you must configure the properties for each board.

To set properties:

1. Start EPSON RC+.
2. Select System Configuration from the Setup menu.
3. Click the I/O Systems tab.
4. Click on the ISA Digital I/O Boards item in the list.
5. Click the **Configure** button.
6. Select the desired board in the list.
7. Click the **Properties** button. The properties dialog will be displayed.



8. Set the properties for the inputs and outputs on the board.

Removing an I/O board

Follow these steps to remove an I/O board from the system.

1. Start EPSON RC+.
2. Select System Configuration from the Setup menu.
3. Click the I/O Systems tab.
4. Click on the ISA Digital I/O Boards item in the list.
5. Click the **Configure** button.
6. Select the last I/O board in the list. You can only remove the last board.
7. Click the Delete button to remove the board from the list.
9. Click **OK** to close the dialog. Then click OK again to close the System Configuration dialog and save the new changes. The SPEL Runtime drivers will be restarted.

Ethernet I/O

Overview

The Ethernet I/O option is an integrated I/O system using the Opto22 SNAP Ethernet I/O system. This system allows you to connect I/O racks with standard Ethernet 10BaseT hubs, switches, and cables. The EPSON RC+ software key Ethernet I/O Option must be enabled to use this option.



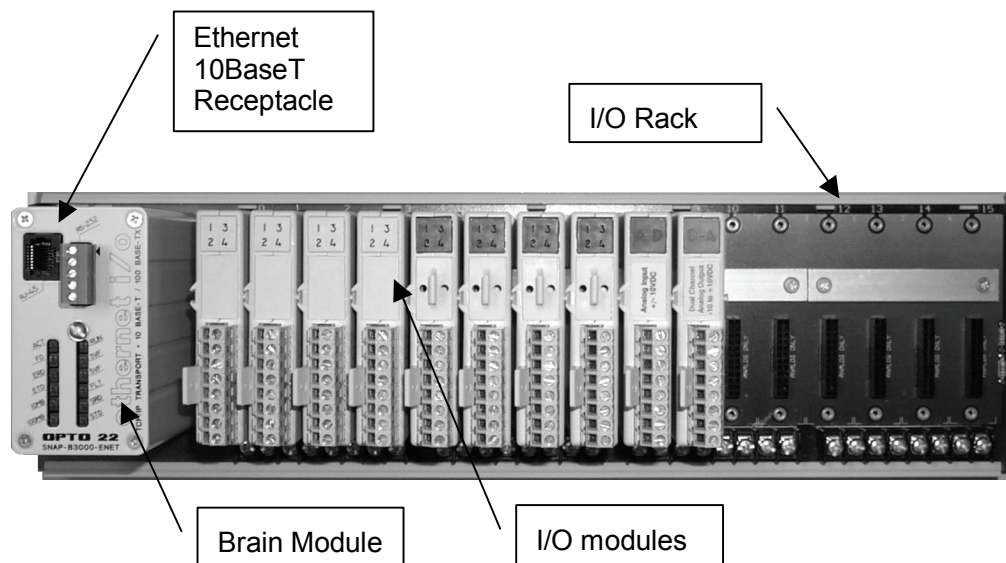
It is assumed that the user has prior knowledge of using and configuring Ethernet networks before using the EPSON RC+ Ethernet I/O option. The information contained in this manual explains how to interface with the Ethernet I/O via the network using EPSON RC+.

Response times for Ethernet I/O can vary and depend on several factors, including network load, number and types of devices, number of SPEL+ tasks, etc. When the fastest and most consistent response times are required, please use EPSON Standard digital I/O, which incorporates interrupt driven inputs and outputs.

Each I/O system consists of one or more I/O racks. Each rack has 1 Brain module, and can contain up to 16 I/O modules, depending on the size of the rack. Digital Input and Output modules typically contain 4 points. Analog modules have 1, 2, or 4 channels.

The brain module from each rack is a TCP/IP server. One 10BaseT cable connects each Brain to an Ethernet hub or switch.

For a complete list of the hardware supported, refer to *Appendix D*.

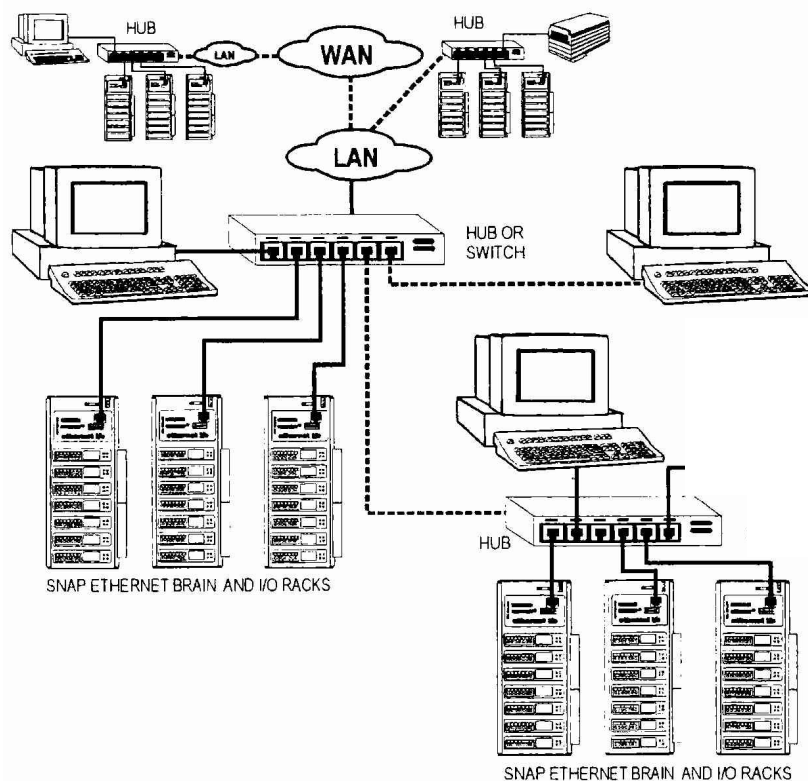


RCSSNAP-B16MC Rack

Ethernet I/O Hardware Installation

Each I/O rack's brain module must be connected to an Ethernet network using 10BaseT cable to a hub. If your PC is not connected to a network and you have one rack, you can use the crossover cable that is supplied with the system to connect directly to the rack without the need for a hub.

The figure below shows a network with Ethernet I/O racks connected to Ethernet hubs.



Assigning rack IP addresses

The IP address for each brain module is fixed. If you are using DHCP (Dynamic Host Configuration Protocol) on your network, then you need to assign fixed IP addresses for the brain modules on the DHCP server. You need to plan what IP addresses you will assign to each brain module.

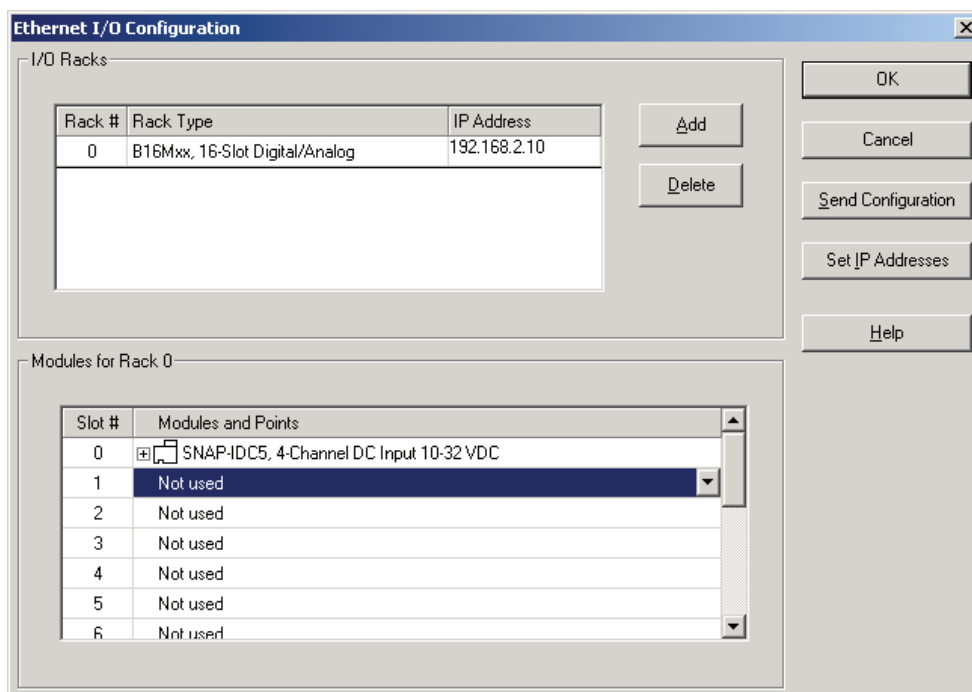
If there is a network administrator at your facility, you should discuss your addressing scheme with him. He may assign addresses for your racks.

Here is an example addressing scheme that uses the standard class C subnet 192.168.1. This will allow one PC and 253 racks on one sub net. If you use more than one subnet, you will need a router or a PC with multiple network cards to act as a router. The PC address is 192.168.1.1.

Rack	IP Address
0	192.168.1.2
1	192.168.1.3
2	192.168.1.4
...	...
253	192.168.1.254

Ethernet I/O Software Configuration

To configure Ethernet I/O, select System Configuration from the Setup menu. Click the I/O Systems tab, then select Ethernet I/O from the list and click the Configure button. The dialog shown below will be displayed.



Option	Description
Add	Click this button to add a rack.
Delete	Click this button to delete the last rack. It will be dimmed when the last rack is not selected or there are no racks installed.
Send Configuration	Sends the current configuration to the currently selected rack's brain module and writes it in flash memory.
Set IP Addresses	Configures rack brain module IP addresses.

Adding racks

To add a rack:

Click the **Add** button. A default rack, B16Mxx will be added to the rack list.

If you will be using a different type of rack, click in the Rack Type field for the new rack, then click the drop down arrow to show the list of available racks.

Setting Brain IP addresses

Before a rack can be used on the system, its brain module must be assigned an IP address. This operation needs to only be performed once, unless you want to change the IP address again later.

When you purchase a rack, it has an IP address of 0.0.0.0, which is invalid and indicates that the address has not yet been configured. When the rack is powered up, it sends special messages over the network to assign it an address. This is called the BOOTP protocol.

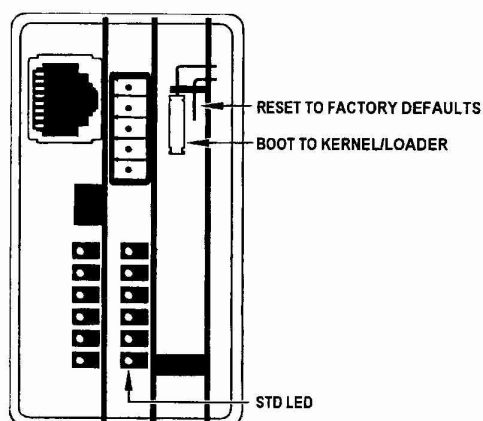
Follow these steps:

1. Make sure your rack is connected to an Ethernet hub on the same subnet as your PC, or use a crossover 10BaseT cable to connect your PC's Ethernet adapter with the brain.
2. Click the **Set IP Address** button. The BootP Utility will start.
3. Click the **Listen** button.
4. Power up the rack.
5. You should see the rack listed.
6. Now assign the IP address and gateway address for the rack. Note the IP address used.
7. Exit the BootP Utility.
8. Type in the IP address in the IP Address field for the rack you just configured.

After clicking the Listen button: if you don't see your rack in the list, you may need to reset the brain, as described below:

To force a rack to request a new address, you must reset the brain module by following these steps:

1. Remove power to the brain.
2. Remove the cover.
3. Remove the "Boot to kernel/loader" jumper.
4. Install the "Reset to factory defaults" jumper.
5. Power up the brain.
6. Wait for the STD LED to stop flashing.
7. Remove the "Reset to factory defaults" jumper.
8. Install the "Boot to kernel/loader" jumper.



Adding modules to a rack

After the racks have been added and configured, you must select the type of module installed in each slot.

1. First, select the rack you want to configure.
2. Click on the module slot to configure.
3. Click the drop down arrow to show the list of available modules.
4. Select the module that is installed.
5. Repeat for all other modules.

Sending configuration

After you have configured a rack, you can send the configuration data to the rack to burn it's flash memory. If you know the rack is already configured, you can skip this step.

1. First, select the rack you want to send data.
2. Click on the **Send Configuration** button.

Ethernet I/O Addressing in SPEL⁺

All of the Ethernet I/O commands and labels use a numbering scheme that includes the rack number along with the point number or port number.

Rack numbers start at 0.

Rack	Point Numbers	Port Numbers
0	0 - 63	0 - 15
1	1000 - 1063	1000 - 1015
2	2000 - 2063	2000 - 2015
n	n000 - n063	n000 - n015

SPEL⁺ Ethernet I/O Commands

All Ethernet I/O commands begin with the same prefix: "ENetIO_". Here is a list of all of the commands. For details, please see the online help or SPEL⁺ Language Reference Manual.

ENetIO_AnaGetConfig	Returns analog input or output module configuration values.
ENetIO_AnaIn	Reads one analog input or output channel.
ENetIO_AnaOut	Sets the value for one analog output.
ENetIO_AnaSetConfig	Configures analog input or output module.
ENetIO_ClearLatches	Clears both the on and off digital latches for one input point.
ENetIO_DigGetConfig	Returns the configuration for a digital input.
ENetIO_DigSetConfig	Sets the configuration for a digital input.
ENetIO_In	Reads one digital input or output module (4 points).
ENetIO_IONumber	Returns the bit number of the specified Ethernet I/O label.
ENetIO_Off	Turns off one digital output.
ENetIO_On	Turns on one digital output.
ENetIO_Oport	Returns the status of one digital output.
ENetIO_Out	Sets one digital output module (4 points).
ENetIO_ReadCount	Returns the current counter value for the specified input.
ENetIO_Sw	Returns the status of one digital input.
ENetIO_SwLatch	Returns the status of one digital input on or off latch.

Using Standard Counters

Any digital input on an analog/digital rack can be used as a standard counter.

1. Call ENetIO_DigSetConfig to configure the input as a standard counter using the value 1.
2. Call ENetIO_ReadCount to read and optionally reset the counter.

For example:

```
Function EnableCounters(ByVal enable As Boolean)
    If enable Then
        config = 1
    Else
        config = 0
    EndIf
    ENetIO_DigSetConfig 0, enable
    ENetIO_DigSetConfig 1, enable
Fend

Long count0, count1

' Read counter for digital input 0, but do not reset it
count0 = ENetIO_ReadCount(0, FALSE)

' Read counter for digital input 1 and reset it
count1 = ENetIO_ReadCount(1, TRUE)
```


Using Quadrature Encoder Counters

Some input modules can be used to read quadrature encoders, such as the SNAP-IDC5Q.

The SNAP-IDC5Q can read two separate encoders with optional index (Z phase) counters.

To use an encoder counter:

1. Call `ENetIO_DigSetConfig` to configure either the A or B phase input as a quadrature counter with a value of 4. You only need to configure one of the inputs.
2. Call `ENetIO_ReadCount` to read and optionally reset the counter.

Using Quadrature Encoder Counters with Index

You can also count the encoder index (Z phase) channel by using a separate input module. It is recommended that you use a fast input module for counting the index pulse such as the SNAP-IDC5FAST.

To use an encoder counter with index counter:

1. Call `ENetIO_DigSetConfig` to configure either the A or B phase input as a quadrature counter with index using the value `xxgg0004`, where `xx` is the bit number of the index (Z phase) input using a separate module and `gg` is 01 for leading edge or 02 for trailing edge index trigger. You only need to configure one of the inputs. The index channel will automatically be configured in this format: `qqgg0041`, where `qq` is the encoder input and `gg` is the index trigger edge.
2. Call `ENetIO_ReadCount` for either the A or B channel to read and optionally reset the encoder counts.
3. Call `ENetIO_ReadCount` for the index channel to read and optionally reset the index counts.

For example, with a SNAP-IDC5Q in slot 0 and SNAP-IDC5FAST in slot 1, encoder 1 A and B phases are connected to inputs 0 and 1 of the IDC5Q. Encoder 1 Z phase is connected to input 0 of the IDC5FAST.

```
' xxgg0004 = 04010004
ENetIO_DigSetConfig 0, &H04010004

' Read encoder count for encoder 1
encoderCount = ENetIO_ReadCount(0, FALSE)

' Read index count for encoder 1
indexCount = ENetIO_ReadCount(4, FALSE)
```

Fieldbus I/O

Fieldbus I/O is an option. For instructions on using the Fieldbus I/O option, refer to the Fieldbus I/O Manual.

Remote Control

Overview

The EPSON RC+ system can be controlled by external equipment using Remote Control. The remote equipment can execute several commands, including Motor On/Off, Start, Pause, Continue, Abort, Select Robot, Select Group.

There are two steps required for remote control:

1. Configure Remote Control inputs and outputs using the Remote Control tab on the System Configuration dialog from the Setup menu.
2. Set the control device to remote.

When the control device is remote, then robot commands and task commands can only be issued by the remote device.

One typical use for remote control is 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, TasksRunning, Error, etc. The buttons are connected to remote inputs to control motor power and start programs.

Another use for remote control is control from a PLC. When using a PLC, you will need to be familiar with the handshake required to use remote inputs. See details below.

Auto Cycle Remote Control

To run in auto cycle with remote control:

1. The system Start Up Mode should be set to **Production / Remote**. See the section *Start Up Mode* in the chapter *Operation* for details.
2. The system should also be configured to auto log into Windows and start EPSON RC+ during Windows start. See the section *Start Up Mode* in the chapter *Operation* for details.
3. The host device (e.g. PLC) can wait for the AutoMode remote output to turn on before issuing remote commands.
4. Now the remote input commands will be accepted.

Teach Mode with Remote Control

When using Teach Mode with remote control, when Teach mode is ON, no remote input commands can be used. Remote status outputs will still operate.



WARNING

Remote status outputs (such as MotorOn, PowerHigh, etc.) will operate when Teach Mode is ON, even when the enable switch (dead man) is disengaged. Therefore, DO NOT use remote status outputs to drive any devices that cause motion or any other safety hazard.

You can monitor the Teach Mode status and the EnableOn status. EnableOn will be ON when Teach Mode is ON and the operator has engaged the enable switch (dead man).

Debugging Remote Control

You can debug programs using Remote Control from the EPSON RC+ development environment.

To run programs by remote control for debugging:

1. Set control device. Select Control Device from the Run menu, then select Remote.



Or, you can change to remote from the monitor window:

```
>ctrldev remote
```

2. Open the Run Window and click Enable Remote I/O.
3. Now the remote input commands will be accepted.

You can set breakpoints and print messages to the run window.

Remote Input Handshake Timing

Whenever a remote input is actuated, the remote device can monitor that the command has been accepted and completed.

1. Turn on a remote Input
2. Wait for CmdRunning remote output to turn on.
3. Wait for CmdRunning to turn off.
4. Check Error remote output for error condition.

If an error occurs, the Error output will turn on. To clear the error, you must turn on the Reset input. No other inputs will be accepted when there is an error condition.

When tasks are running (after starting a program group), certain inputs are disabled, as indicated in the next section. The TasksRunning remote output can be used to monitor this condition.

Selecting Program Groups

By default, when the Start input is turned on, program group 1 is started. You can select other program groups by using the SelGroup inputs. A maximum of 16 program groups can be selected.

Program groups are numbered in the order that they appear in the project editor, starting with 1.

To select a program group:

1. Turn on the SelGroup0 - SelGroup3 remote inputs that correspond with the desired group number. This binary number is zero based and is equal to the group number - 1.
2. Turn the SelGroup input on and off to select the new group.

You can monitor the current group number using the CurrGroup0 - CurrGroup3 remote outputs.

Selecting Robots

For multi-robot systems, you must select a robot before executing robot command inputs and status outputs, such as Motor On and Power High.

To select a robot:

1. Turn on the SelRobot0 - SelRobot3 remote inputs that correspond with the desired robot number. This binary number is zero based and is equal to the robot number - 1.
2. Turn the SelRobot input on and off to select the new robot.

You can monitor the current robot number using the CurrRobot0 - CurrRobot3 remote outputs.

Remote Inputs

MotorOn	Turns motors on for the current robot. The MotorOff input must be off. Disabled when tasks are running.
MotorOff	Turns motors off for the current robot and disables MotorOn. Disabled when tasks are running.
PowerHigh	Sets high motor power mode for the current robot. The motors must be on and the PowerLow input must be off. Disabled when tasks are running.
PowerLow	Turns off high power mode for the current robot and disables PowerHigh. Disabled when tasks are running.
MCal	Executes MCal command. Disabled when tasks are running.
Home	Executes Home command. Disabled when tasks are running.
Start	Starts the current program group. Disabled when tasks are running.
Pause	Pauses all tasks that are configured to pause. Enabled only when tasks are running.
Cont	Continues tasks that have been paused. Enabled only when tasks are running.
AbortAll	Aborts all tasks or the current command.
Reset	Resets the controller and error condition. This is the only command accepted when the Error output is turned on.
SelRobot	Selects the current robot for robot commands, according to the 4 inputs SelRobot0 to SelRobot3. The robot number is 0 based, with 0 corresponding to Robot 1 (default).
SelRobot0	Robot Bit 0 (value = 1).
SelRobot1	Robot Bit 1 (value = 2).
SelRobot2	Robot Bit 2 (value = 4).
SelRobot3	Robot Bit 3 (value = 8).
SelGroup	Selects the current program group to be started with the Start input according to the 4 inputs SelGroup0 to SelGroup3.
SelGroup0	Group Bit 0 (value = 1).
SelGroup1	Group Bit 1 (value = 2).
SelGroup2	Group Bit 2 (value = 4).
SelGroup3	Group Bit 3 (value = 8).
Recover	Initiates a position recovery after the safeguard has been closed. If the SPEL Option <i>Auto recover position after safeguard</i> is on, the robot will slowly move to the position it was in when the safeguard was open. If the option is off, then the robot will move only while this input remains on. If this input goes off during recover motion, then the robot will stop and recover will not be complete.
ShutDown	Shuts down RC+ and Windows. Disabled when tasks are running.
Reboot	Shuts down RC+ and reboots Windows. Disabled when tasks are running.

Remote Outputs

MotorOn	Indicates that motors are on for the current robot.
PowerHigh	Indicates that motor power is set to high for the current robot.
MCalComplete	Indicates that MCal has been completed for the current robot. If the robot uses absolute encoders, then this output will always be on.
AtHome	Indicates that the current robot is at its home position.
CmdRunning	Indicates that a remote input command is in cycle.
TasksRunning	Indicates that tasks are running in the controller.
Paused	Indicates that tasks have been paused.
SafeGuard	Indicates that the safeguard circuit is open.
EStopOn	Indicates that emergency stop has occurred.
Error	Indicates that an error had occurred. You must execute a Reset input to clear the error condition.
AutoMode	Indicates that the system is in auto mode and remote input commands will be accepted.
TeachMode	Indicates that the system is in Teach mode.
EnableOn	Indicates that the enable switch (dead man) is on with the system in Teach mode.
CurrRobot0	Robot Bit 0 (value = 1).
CurrRobot1	Robot Bit 1 (value = 2).
CurrRobot2	Robot Bit 2 (value = 4).
CurrRobot3	Robot Bit 3 (value = 8).
CurrGroup0	Group Bit 0 (value = 1).
CurrGroup1	Group Bit 1 (value = 2).
CurrGroup2	Group Bit 2 (value = 4).
CurrGroup3	Group Bit 3 (value = 8).
MotorMode	Indicates the current motor setting. Sometimes the motors are turned off by the system, such as when the safeguard is opened. But MotorMode signals the current user setting.
PowerMode	Indicates the current power setting. Sometimes the power is set to low by the system. But PowerMode signals the current user setting.
RecoverReqd	Indicates that a recover position can be executed after the safeguard has been closed.
RecoverInCycle	Indicates that recover position is in cycle.

Remote Safeguard Position Recover

When the safeguard is opened while tasks are executing, all tasks are paused, all robots decelerate to stop, and all robot motors turn off. In some cases, the robot position will change when the motors go off. Also, some robot models may be moved by hand while the safeguard is open.

After the safeguard is closed, the robot motors must be turned on again and moved slowly to the positions they were at when the safeguard was opened. This is called safeguard position recovery.

In EPSON RC+, you can configure the system for auto recover or manual recover by setting the SPEL Option *Auto recover position after safeguard*.

Remote Auto Recover

For auto position recovery, the SPEL Option *Auto recover position after safeguard* must be ON.

After the safeguard is closed, if Recover input is ON, then recover is executed. Motion does not stop if recover input is removed during recover motion.

After the safeguard is closed, if Cont input is ON, then recover is executed first if required, then all paused tasks will resume.

Remote Manual Recover

For manual position recovery, the SPEL Option *Auto recover position after safeguard* must be OFF.

After the safeguard is closed, if Cont input is ON and recover is required, no action occurs.

After the safeguard is closed, if Recover input is ON, recover is executed. Motion will stop if the recover input is removed before recover motion has completed.

After recover has completed, the Cont input can be used to resume all paused tasks.

Operation of RecoverReqd and RecoverInCycle Outputs

During recover motion, RecoverReqd output turns OFF and RecoverInCycle output turns ON. If recover is complete, RecoverInCycle turns OFF. If recover is not complete, the RecoverReqd turns ON again, and RecoverInCycle turns OFF.

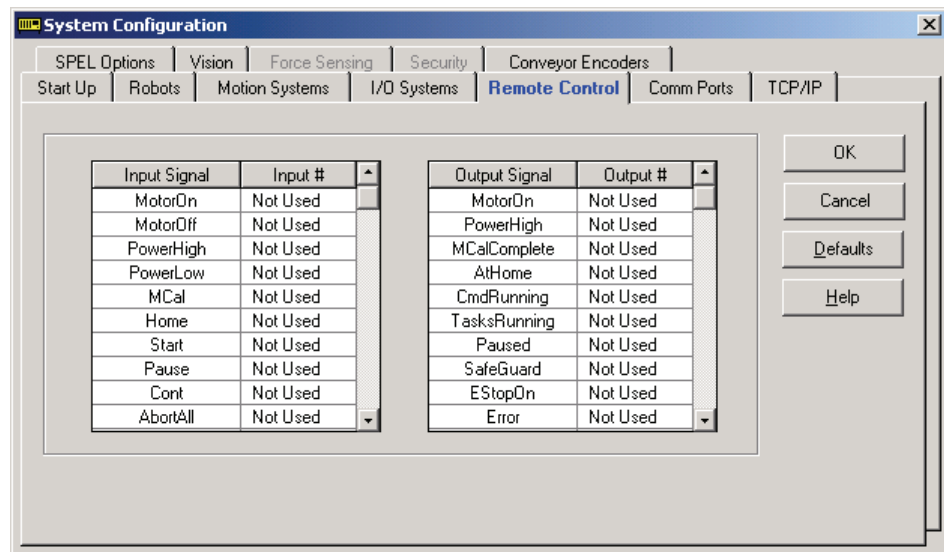
Remote Control Software Configuration

Any digital input or output can be configured for a system function. For details on using Remote Control, see Using Remote Control.

To assign I/O for remote control use:

1. Select System Configuration from the Setup menu and select the **Remote Control** tab.
2. For each input or output you want to use for remote control, click on the Input # or Output # cell next to the desired bit, then select the bit number.
3. Click **OK** to save the new settings.

To clear all remote settings, click the **Defaults** button.



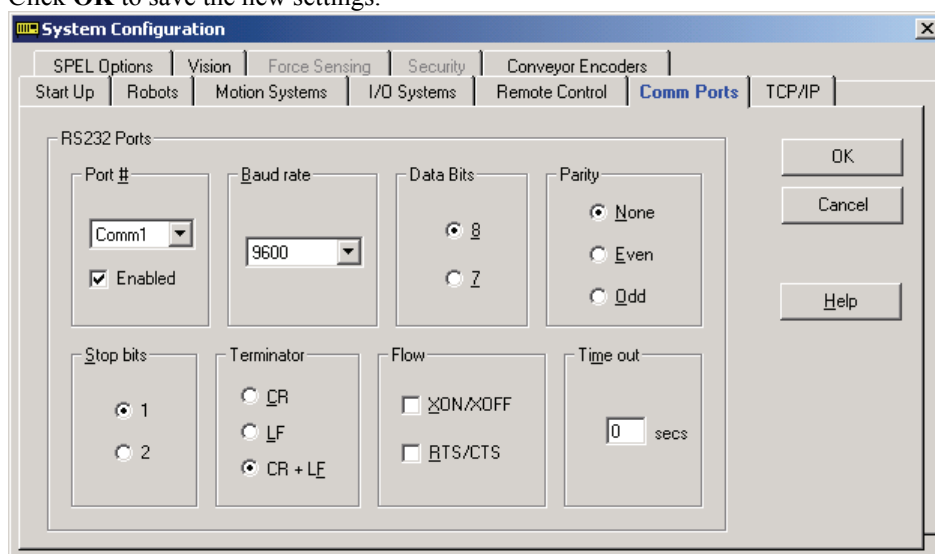
RS-232 Communications

EPSON RC+ supports standard Windows RS-232 ports: Comm1-Comm2 and Comm5 – Comm20. Comm3 and Comm4 are reserved for EPSON RC+. To use Comm5 – Comm20, you need to purchase an RS-232 board with standard Windows drivers.

RS-232 Software Configuration

To configure an RS-232 port:

1. Select System Configuration from the Setup menu and select the **Comm Ports** tab.
2. Select the port you want to configure from the **Port #** drop down list.
3. Click the **Enabled** check box to enable the port.
4. Change the other settings as desired.
5. Click **OK** to save the new settings.



RS-232 Commands

Here is a list of all of the commands associated with RS-232 communications. For details, please see the online help or SPEL⁺ Language Reference Manual.

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.
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.
Write #	Sends characters out of the port.

TCP / IP Communications

EPSON RC+ supports up to twenty TCP/IP ports that allow peer to peer communications. This chapter contains instructions on using TCP/IP, including IP addresses, Windows TCP/IP configuration, and SPEL⁺ examples.



When the Windows firewall is activated, you cannot use TCP/IP communications. Make sure to disable the firewall. Firewall is disabled at shipment.

TCP/IP Setup

Before you can use TCP/IP communications, you must configure your total system. This section describes basic configuration.

Ethernet Hardware

The controller CPU includes a built in Ethernet interface with an RJ45 connector accessible from the controller rear panel. It supports 10BaseT (10 Mbps) and 10BaseTX (100 Mbps).

IP Addresses

You need to decide how devices on your system will be addressed.

- **DHCP (Dynamic Host Configuration Protocol)**
This system can be used if a server is running DHCP on the network. Clients automatically get their IP addresses from the server. In this configuration, host names must be used, not IP addresses. Normally, DHCP is used in larger systems where a system administrator is managing the network. In this case, you should contact your administrator to plan your network layout. DHCP is typically installed on a PC running Windows NT/2000 Server or a router. Even though you see a DHCP tab in Windows on the controller, it is not the DHCP service. It is only used to communicate with DHCP.
- **LMHOSTS File**
You can configure an LMHOSTS file on one computer on the system to be the master. The LMHOSTS file contains host names and their corresponding IP addresses. All other PCs on the system can use an INCLUDE statement in their LMHOSTS file that includes the master file.
- **Hard Coded IP addresses**
IP addresses can be used. This system is the easiest to setup, but not to maintain, since all computers on the system have a fixed address. For small systems where IP addresses will not change, this is a good choice.

If you are not using DHCP, make sure that you use the correct network address and subnet mask in your configuration. For PCs to communicate, they must either be on the same subnet or point to a gateway that routes to other subnets. For small systems, it is best to use one subnet.

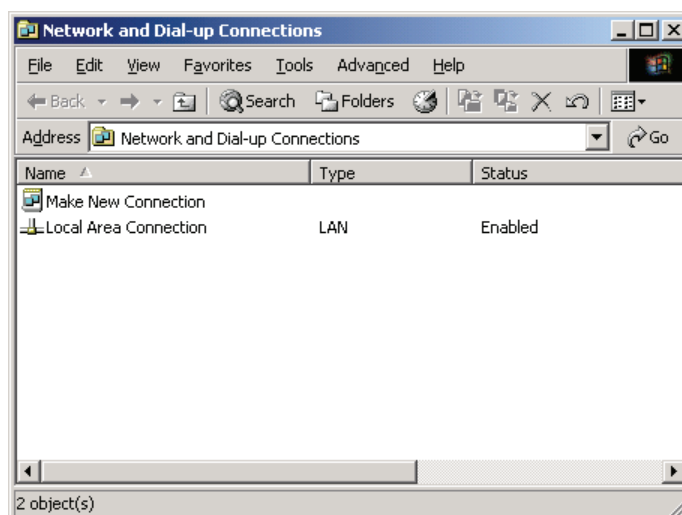
The following table shows a typical IP address configuration.

Host Name	IP Address	Subnet	Subnet Mask
PC1	192.168.1.1	192.168.1	255.255.255.0
PC2	192.168.1.2	192.168.1	255.255.255.0

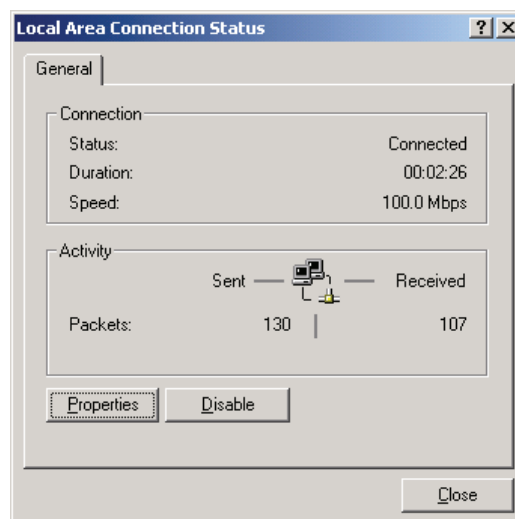
In this example, the network address (subnet) is 192.168.1. With a subnet mask of 255.255.255.0, there can be 254 hosts on this subnet (0 and 255 cannot be used).

Configuring Windows TCP/IP

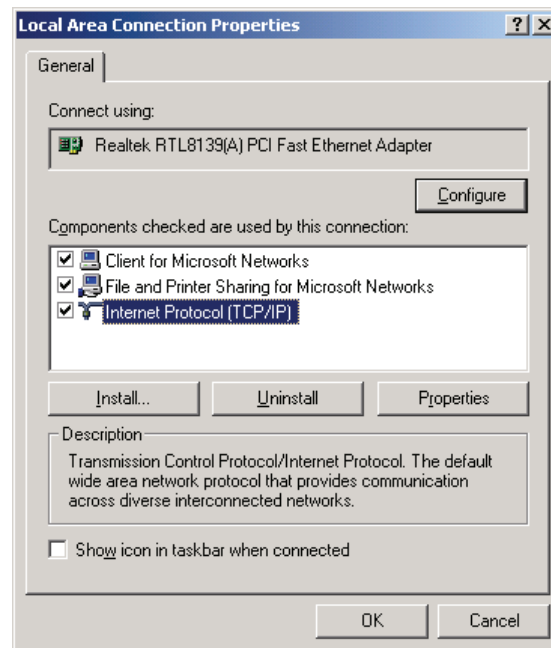
From the Control Panel, double click the **Network and Dial-up Connections** icon to open the following window.



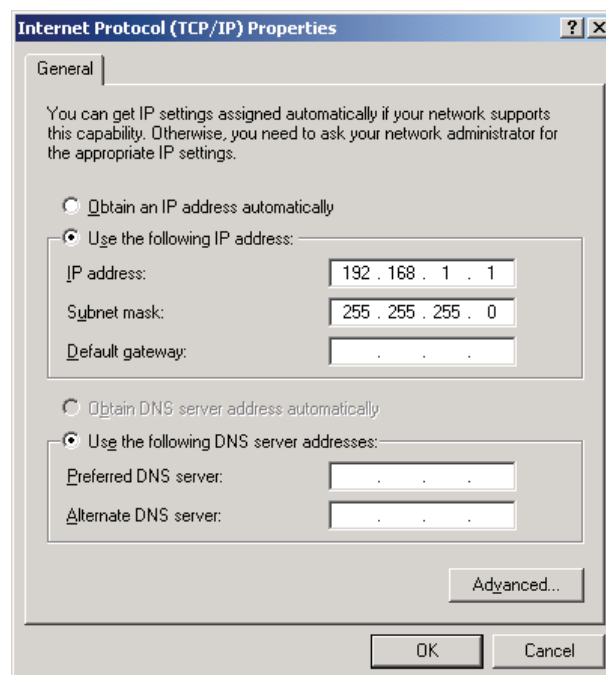
Double-click on **Local Area Connection**.



Click the **Properties** button.



Next, click on Internet Protocol (TCP/IP) and then click the **Properties** button. The following dialog is displayed:



Click on the **Use the following IP address** radio button.

Enter the IP address and the subnet mask. Default Gateway is optional and not needed if all computers are on the same subnet and there is no router.

Click OK to save changes.

LMHOSTS File Configuration

If your system is not using DHCP and you want to use meaningful names instead of IP addresses, then you can configure an LMHOST file on each PC. Optionally, you can setup one PC to contain the master file, and the other PCs can include it.

To create an LMHOSTS file:

Open a Command Window in Windows.

```
cd \winnt\system32\drivers\etc.
```

```
edit LMHOSTS.
```

Type in the following:

192.168.1.1	PC1	PRE#
192.168.1.2	PC2	PRE#

Save the file. Now you should be able to use the host names instead of IP addresses. The PRE# keyword tells the system to load this address in the name cache at start up.

To setup a master LMHOSTS File, designate one PC on the network to contain the master. For example:

Master LMHOSTS file on PC1:

192.168.1.1	PC1	PRE#
192.168.1.2	PC2	PRE#
192.168.1.2	PC3	PRE#

All other PCs can include the master LMHOSTS file in their own LMHOSTS file:

192.168.1.1	PC1	PRE#
#INCLUDE \\pc1\winnt\system32\drivers\etc\lmhosts		

Note that the first IP address must be supplied for PC1, the master, so that the INCLUDE statement can find the master lmhosts file on PC1.

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:\>
```

If you are using DHCP or LMHOSTS files, ping your own computer by name:

```
C:\>ping pc1
Pinging pc1 [192.168.1.1] with 32 bytes of data:
Reply from 192.168.1.1: bytes=32 time<10ms TTL=128
Reply from 192.168.1.1: bytes=32 time<10ms TTL=128
Reply from 192.168.1.1: bytes=32 time<10ms TTL=128
Reply from 192.168.1.1: bytes=32 time<10ms TTL=128
C:\>
```


If not using DHCP or LMHOSTS, ping your own IP address:

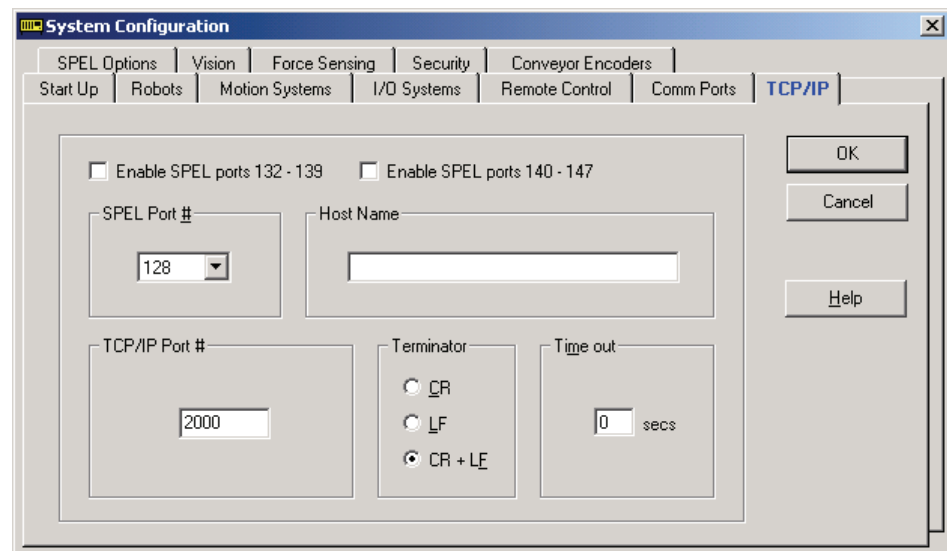
```
C:\>ping 192.168.1.1
Pinging 192.168.1.1 with 32 bytes of data:
Reply from 192.168.1.1: bytes=32 time<10ms TTL=128
Reply from 192.168.1.1: bytes=32 time<10ms TTL=128
Reply from 192.168.1.1: bytes=32 time<10ms TTL=128
Reply from 192.168.1.1: bytes=32 time<10ms TTL=128
C:\>
```

Now ping another computer on the network. Use the name if you are using DHCP or LMHOSTS, otherwise use the IP address:

```
C:\>ping pc2
Pinging pc2 [192.168.1.2] with 32 bytes of data:
Reply from 192.168.1.2: bytes=32 time<10ms TTL=128
Reply from 192.168.1.2: bytes=32 time<10ms TTL=128
Reply from 192.168.1.2: bytes=32 time<10ms TTL=128
Reply from 192.168.1.2: bytes=32 time<10ms TTL=128
C:\>
```

TCP/IP Software Configuration

You can configure TCP/IP settings in a SPEL⁺ program using the SetNet command. You can also configure settings from the TCP/IP tab on the System Configuration dialog.



For two PCs running EPSON RC+ to communicate with TCP/IP, they must use each other's host name and must also use the same TCP/IP port number.

If you are using hard coded IP addresses, you can use the IP address in the host name.

TCP/IP Commands

Here is a list of all of the commands associated with TCP/IP communications. For details, please see the online help or SPEL⁺ Language Reference Manual.

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.
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.
Write #	Sends characters out of the port.

TCP/IP Example

This example shows how to configure 2 PCs running EPSON RC+ to communicate using TCP/IP. Note that for the OpenNet command, one PC is the server, and the other is a client.

PC #1:

Port: #128

Host Name: PC2

TCP/IP Port: 1000

```
Function tcpip1
  String reply$
  OpenNet #128 As Server
  WaitNet #128
  Do
    Print "Sending data to host 1"
    Print #128, "Data from host 1"
    Input #128, reply$
    Print "Received reply: ", reply$
  Loop
Fend
```

PC #2:

Port: #128

Host Name: PC1

TCP/IP Port: 1000

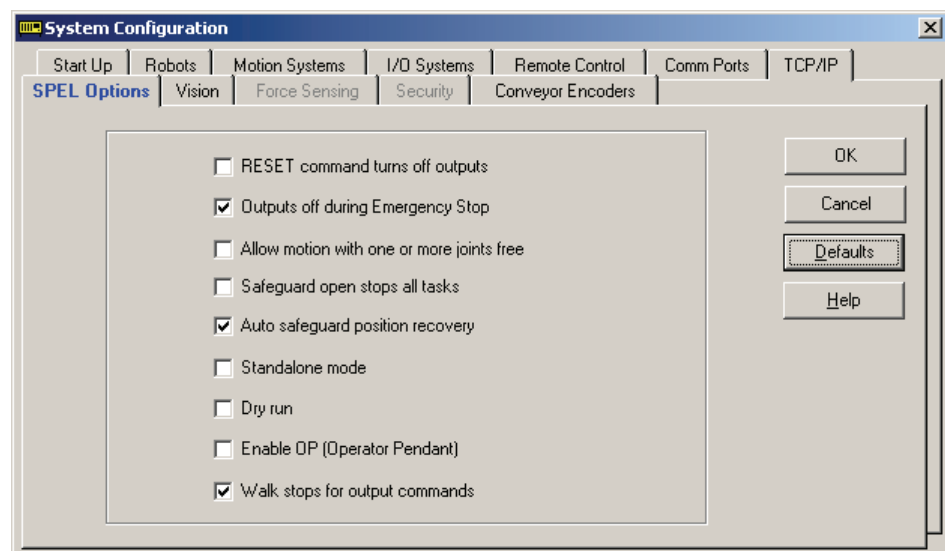
```
Function tcpip2
  String data$
  OpenNet #128 As Client
  WaitNet #128
  Do
    Input #128, data$
    Print "Received '", data$, "' from host 1"
    Print #128, "Received data"
  Loop
Fend
```

SPEL⁺ Options

There are several options switches for the SPEL system.

To configure SPEL options:

1. Select System Configuration from the Setup menu and select the **SPEL Options** tab.
2. Change settings as desired by clicking the check boxes.
3. Click **OK** to save the new settings.



The option switches are as follows:

RESET command turns off outputs

When this option is turned on, all outputs other than remote control outputs will be turned off when a Reset instruction is executed. The default setting is off.



NOTE

The outputs of the standard I/O, expansion I/O, Ethernet I/O, and Fieldbus I/O (option) are included in the “Outputs” mentioned in the above option switches (“RESET command turns off outputs” and “Outputs off during Emergency Stop”). However, the outputs of the memory I/O are not included in the “Outputs.” Therefore, the outputs of the memory I/O are not turned off by RESET command execution or during Emergency Stop.

Outputs off during Emergency Stop

When this option 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. The default setting is on.

NOTE



You should always remove all power to output devices when emergency stop occurs. Even if EPSON RC+ turns off outputs, the I/O board could be malfunctioning.

Allow motion with one or more joints free

When this option is turned on, motion commands can be executed after SFree has been used to free one or more joints. The default setting is off.

Safeguard open stops all tasks

Check this option to cause all tasks to stop when the safeguard is open. When checked and the safeguard opens while tasks are running, all robots will decelerate to a stop, all robot motors will be turned off, and all tasks will stop. No pause will occur but the SGOpen trap will fire. This option can be used in applications where pause / continue are not required and the program must be restarted after safeguard is open. This is similar to emergency stop, but the robots decelerate to a stop before motors are turned off.

Auto safeguard position recovery

This option allows you to specify how to recover robot position after the safeguard has been opened and the robot has moved. It is grayed out if *Safeguard open stops all tasks* is checked.

- Auto recover on** The motors will automatically turn back on, and the robot will move slowly to the position it was in when the safeguard was opened, then normal cycle will continue. This is the default setting.
- Auto recover off** For the EPSON RC+ Run window and Operator window, a dialog will be displayed with a Recover button when the operator clicks Cont. The operator must press and hold the Recover button until the motors have turned on and the robot has moved back to the position it was in when the safeguard was opened. If the operator releases the Recover button during recover motion, the robot will stop. When the recover motion has completed, the operator can then click the Cont button to continue normal cycle.

See the section *Safeguard Position Recovery* in the chapter *Safety* for more information.

Standalone mode

Check this box to use the system (RC520 only) in standalone mode. When checked, the system will operate without any robots or motion systems. This mode allows the system to run as an I/O controller. You must use the standalone MIB board and PC emergency stop / safeguard connector to use this mode.

Dry run

This option allows you to run programs without using any drive units (for RC520) or without connecting the controller to a robot (for RC420). All program statements will work. Motion statements will execute approximately the same amount of time as when connected to a robot. You must have at least one MIB board installed to use dry run.

Enable OP (Operator Pendant)

When checked, the OP500RC Operator Pendant is enabled. Before checking this box, the pendant should be connected to the Optional Device connector on the controller.

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. The default setting is on.

NOTE

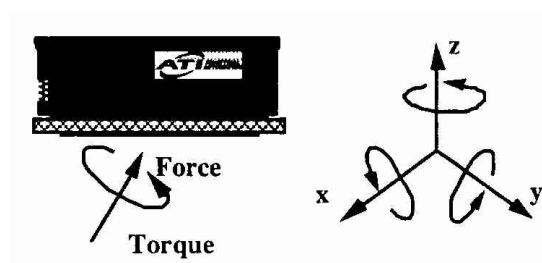


When *Walk stops for output commands* is checked, the Walk command will only stop for standard I/O output commands On, Off, Out, OutW. It will not stop for any other I/O statements, including Ethernet I/O and Fieldbus output statements.

Force Sensing

Overview

The EPSON RC+ Force Sensing Option allows you to integrate force sensing in your applications. The force sensor is typically mounted on the robot's U axis. The sensor has 6 axes: ForceX, ForceY, ForceZ, TorqueX, TorqueY, TorqueZ.



With this option you can do the following:

- Read one or all 6 force/torque sensing axes values.
- Set triggers for motion commands.
- Limit the robot's torque for one or more axes.
- Use multiple force sensors in the same application.



EC Series robot with Gamma force sensor.

Specifications

EPSON RC+ supports ATI force sensors using ISA and PCI interface boards.

For specifications on force sensors, please see the ATI website:

<http://www.ati-ia.com/sensors.htm>

Installation

Your EPSON RC+ software key must have the Force Sensing Option enabled. If you purchased the option with your system, then the option will already be installed and configured.

You can also purchase the Force Sensing Option and install it in the field. See the chapter *Installing EPSON RC+ Options* for details.

Installing the force transducer circuit board

If you are adding Force Sensing in the field, you must install the board(s) in your system, then run the EPSON RC+ setup and select the Force Sensing Option to install the drivers. There are two types of boards supported: ISA and PCI. You can use both types of boards on the same system.

ISA Board Installation

For ISA force sensing boards, you must set the DIP switch S1 for the I/O port address of each board before installing it.

The following table shows the address settings.

Board	Address	S1 (9 - 3)
1	280h	1 0 1 0 0 0 0
2	288h	1 0 1 0 0 0 1
3	290h	1 0 1 0 0 1 0

NOTE



Each ISA board must be connected to its corresponding transducer. Ensure that the number on the transducer matches the number on the mounting flange of the board.

PCI Board Installation

For PCI force sensing boards, you must first install the National Instruments DAQ drivers that came with the board, then install the board.

To install the National Instruments DAQ drivers:

1. Run the NI-DAQ driver installer for Traditional DAQ drivers.
2. Accept defaults for each step of the installation wizard.
3. Shutdown the system.
4. Install the board(s).
5. Start the system.
6. Run the National Instruments Measurement & Automation Explorer program once to register the ActiveX server and verify that the board(s) that were installed are recognized.

You do not need to install the ATI software that came with the force transducer.

NOTE



For PCI force sensing boards, the calibration for the transducer must be loaded into memory. EPSON RC+ handles this when you import the calibration data file as described in the section *Software Configuration* later in this chapter. The calibration data file can be located on the CD that came with the force transducer.

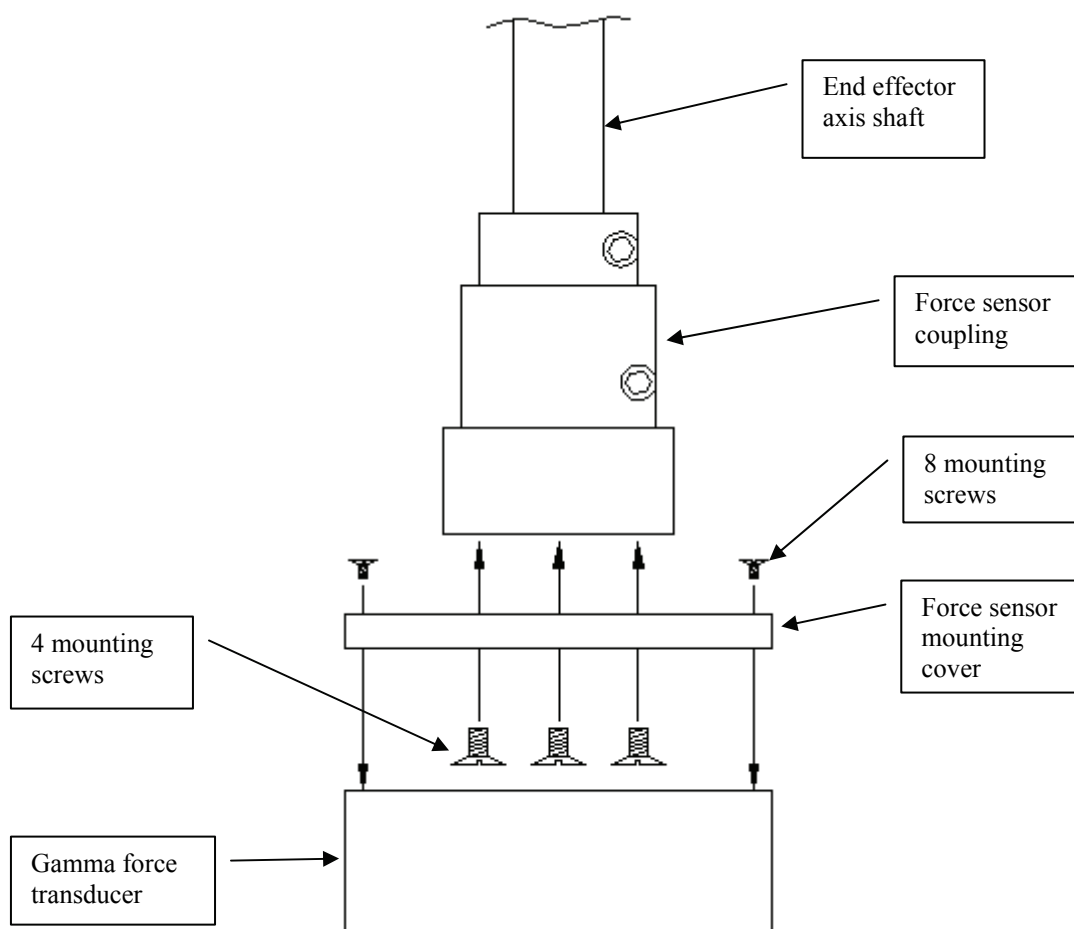
Mounting the force transducer

The Force Sensing option comes with an end effector mounting flange for your robot model that the force sensor is mounted to.

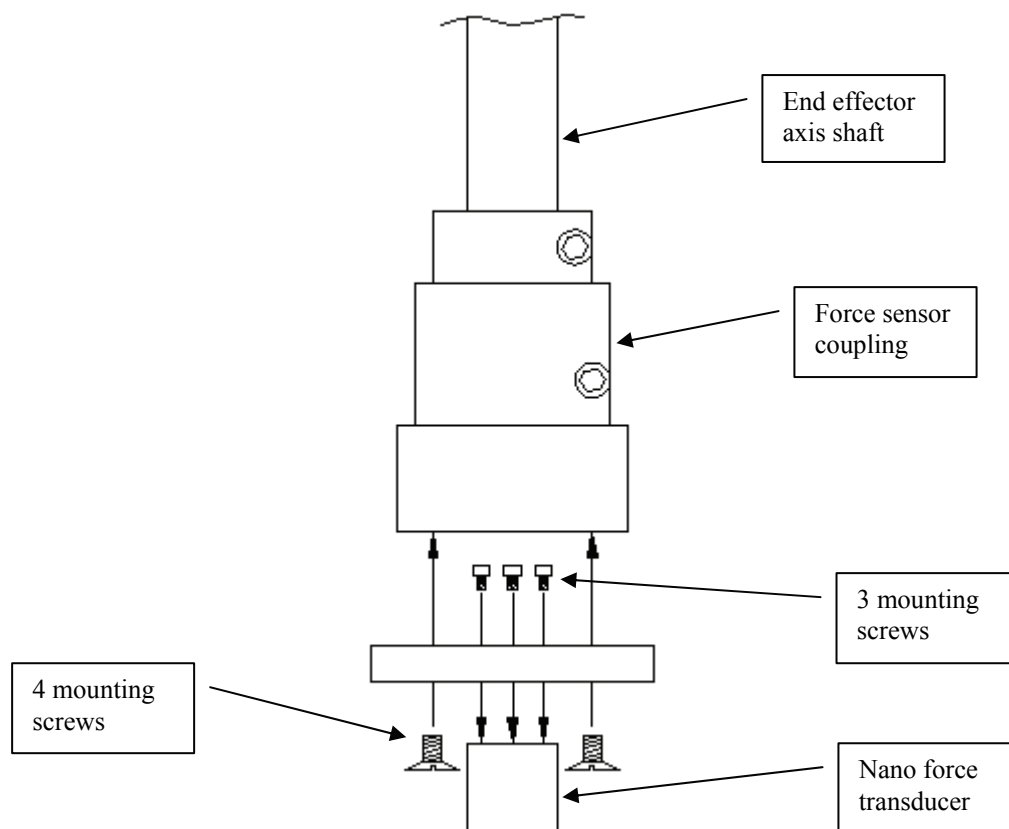
To mount the force transducer to the robot:

1. Remove the top cover of the transducer.
2. Remove the robot's end effector axis coupling and mount it to the transducer cover.
3. Install the transducer cover / coupling assembly onto the transducer.
4. Mount the end effector flange to the bottom side of the transducer.
5. Install the entire assembly on the end effector axis.

The following figures show mounting for gamma and nano transducers.



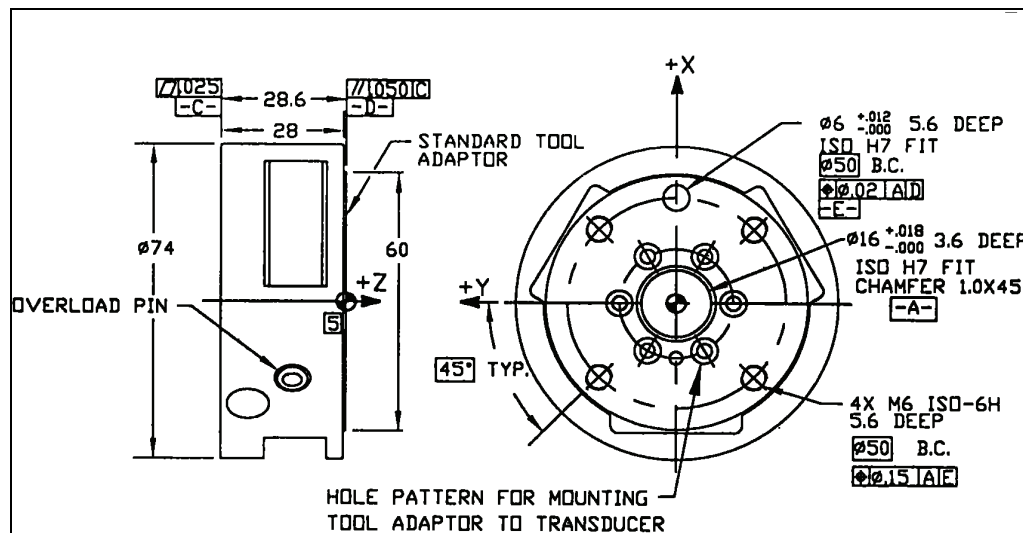
Mounting the Gamma Force Sensor



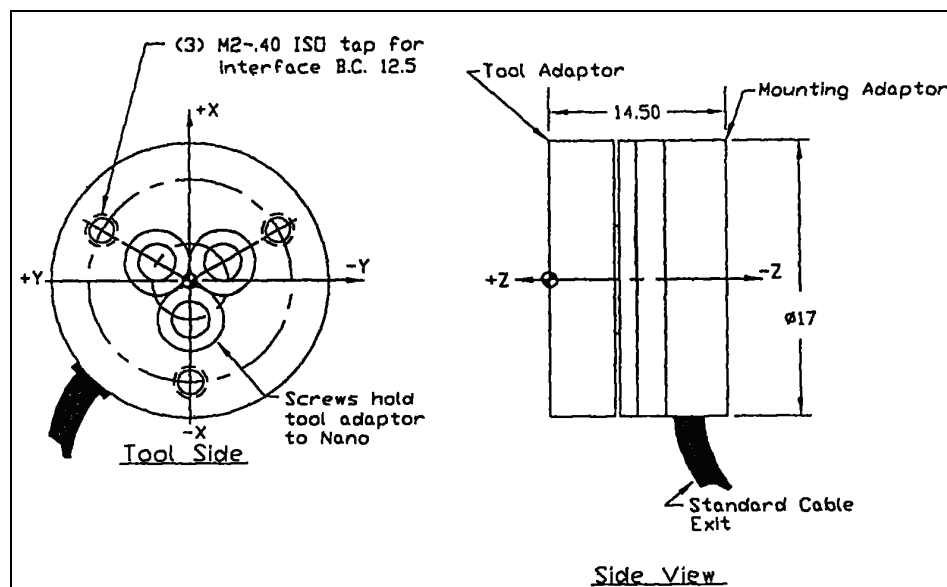
Mounting the Nano Force Sensor

Mounting tooling to force sensor

The following diagrams show the tooling mounting dimensions for Gamma and Nano force transducers.



Tool mounting for Gamma transducer



Tool mounting for Nano transducer

Connecting the force transducer

Use the cable supplied with the transducer to connect it to the PC board. The Nano transducer connects to an external interface box which in turn connects to the PC board.



For ISA force sensing boards, the serial number on the sensor must match the serial number on the PC board. This is important for sensor calibration to work correctly.



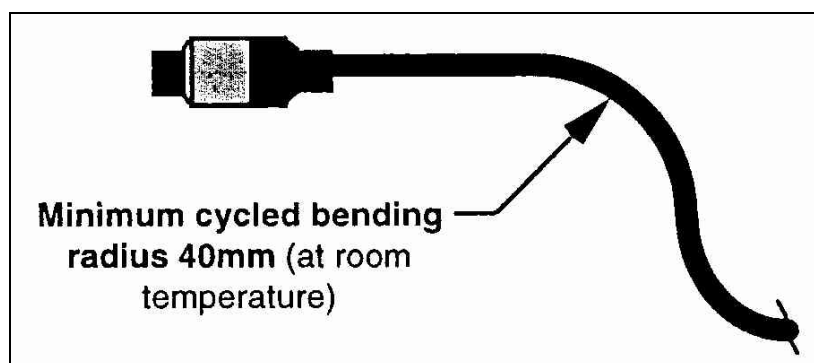
CAUTION

Make sure power is off before connecting or disconnecting the force transducer. Protect transducer from electro-static discharge. Do not touch the internal electronics or connector pins.

Routing the transducer cable

The transducer cable must be routed so that it is not stressed, pulled, kinked, cut, or otherwise damaged throughout the full range of motion. If the cable will rub other cables during cycling, use a plastic spiral wrap to protect it.

When the cable is cycling below the minimum bending radius the cable may fail due to fatigue. A small radius can be used if the cable is not being moved.

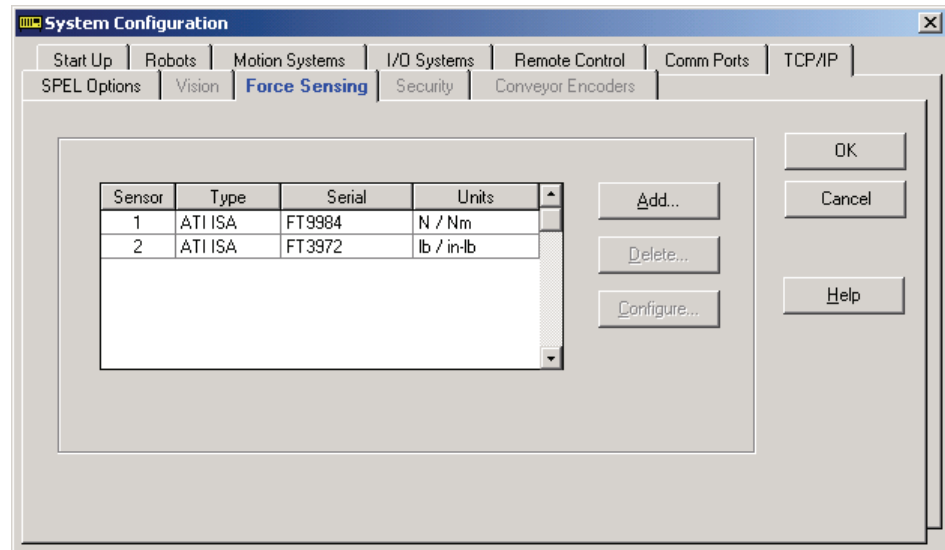


Software Configuration

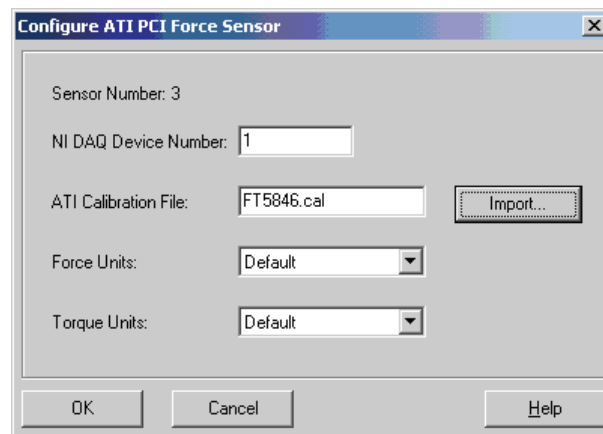
To configure EPSON RC+ Force Sensing:

1. Start EPSON RC+, then select Setup | System Configuration
2. Click on the Force Sensing tab. If the tab is disabled, then the software key has not been enabled.

3. ISA force sensors are automatically detected and should be displayed in the list of sensors.

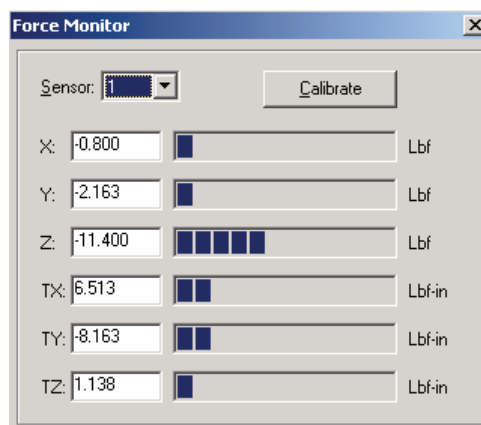


4. To add a PCI board, click the Add button. Otherwise, go to step 8.



5. Enter the NI DAQ device number. This is assigned by the National Instruments software. To view NI DAQ device numbers, run the National Instruments Measurement & Automation Explorer.
6. Click the Import button to import a calibration file for the sensor. This can be found on the CD that came with the sensor. Navigate to the calibration file whose name includes the serial number of the sensor. Click Open, and the file will be copied to the EpsonRC\force directory.
7. Leave the force and torque units at the default setting to use the native units. The actual units will be displayed in the sensor list after you click OK. Or, you can select the desired units.
8. Click OK to accept the new sensor.

9. From the Tools menu, select Force Monitor. This will open the Force Monitor window.



10. Apply pressure to the sensor. You should see the values change on the Force Monitor window. If you are using multiple sensors, change the sensor number on the monitor and verify that each sensor is working.

Force Sensing Commands

All Force Sensing commands begin with the same prefix: "Force_". Here is a list of all of the commands. For details, please see the online help or SPEL⁺ Language Reference Manual.

Force_Calibrate	Zeros out all axes for the current sensor.
Force_ClearTrigger	Clears all trigger conditions for the current sensor.
Force_GetForce	Returns the current value for one axis for the current sensor.
Force_GetForces	Returns the current values for all axes for the current sensor in an array.
Force_Sensor	Sets / returns the current sensor for the current task.
Force_SetTrigger	Sets /displays the force limit triggers for the current sensor.
Force_TC	Sets / returns the current torque control mode for the current robot.
Force_TCLim	Sets / returns the current torque limits for the current robot.
Force_TCSpeed	Sets / returns the current speed after torque limit is reached for the current robot.

Using the Force Sensing Trigger

You can configure the system to stop the robot after the force sensing trigger has been activated. You can set the trigger to activate when one or more force sensing axes reaches a preset limit. You use the Till command to check the trigger condition during motion.

Stopping Motion along Z axis

Use a trigger on the ZForce axis to stop the robot during Z axis motion.

For example:

```
' Set the force trigger to fire when force on Z axis  
' is less than -10 pounds
```

```
Force_ClearTrigger  
Force_SetTrigger FORCE_ZFORCE, -10, FORCE_LESS  
Till Force  
Jump P1  
Speeds 1  
Move P2 Till
```

You can combine other conditions with Force in the Till command:

```
Till Sw(1) = On Or Force
```

You can combine other force/torque conditions by calling Force_SetTrigger more than once. It's a good idea to clear all triggers first before setting them.

```
Force_ClearTrigger  
Force_SetTrigger FORCE_ZFORCE, -10, FORCE_LESS  
Force_SetTrigger FORCE_XFORCE, 5, FORCE_GREATER
```

Stopping Motion along X or Y axes

Use a trigger on the XForce, XTorque, YForce, YTorque axes to stop the robot during Z axis motion. You need to align the force sensor by rotating the robot's U axis. The X and Y axes of the force sensor are marked on the transducer.

For example:

```
' Set the force trigger to fire when torque or force on X axis  
' is less than -10 pounds
```

```
Force_ClearTrigger  
Force_SetTrigger FORCE_XFORCE, -10, FORCE_LESS  
Force_SetTrigger FORCE_XTORQUE, -10, FORCE_LESS  
Till Force  
Jump P1  
Speeds 1  
Move P2 Till
```

Using Torque Limiting

Three commands are provided in the Force Sensing option to limit torque for each robot axis: Force_TC, Force_TCLim, and Force_TCSpeed. These commands allow you to control the amount of torque for each robot joint motor. When one or more joints exceed the torque limit, the robot will attempt to move at the torque limit speed and will not servo out if motion cannot be continued.

The torque limit is specified in percentage of total torque.

```
' Limit motor torque on Z axis to 10 percent
Force_TCLim -1, -1, 10, -1
```

```
' Set speed after reaching torque limit
Force_TCSpeed 5
```

```
' Turn on torque limiting
Force_TC On
```

```
Go P1
```

It is important to note that once torque limiting is on, you must remove the robot from the torque limiting situation before you turn torque limiting off. Otherwise a servo error will occur. For example, after moving to a point under torque limiting, executing Force_TC Off will cause a servo error. This is correct because when torque limiting goes off, the robot is trying to reach the final position.

Security

Overview

The EPSON RC+ Security Option allows you to manage EPSON RC+ users and also monitor usage.

When the Security Option 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 points, use Jog and Teach, and use the Monitor Window. When a user attempts to do something that he does not have a right for, a "Permission denied" message will be displayed.

Each login session is recorded in a Microsoft Access compatible data base. An Audit Viewer is included that allows you to view each session's activity.

Users can login to EPSON RC+ with a name and password. Optionally, EPSON RC+ can use the Windows user name to log in automatically.

Installation

Your EPSON RC+ software key must have the Security Option enabled. You can enable your software key by calling your distributor and purchasing the Security Option. See the chapter *Installing EPSON RC+ Options* for details.

Run the EPSON RC+ installation from the CD and check the Security Option. Normally, this is installed at the factory if the Security Option is purchased with the system.

Security Configuration

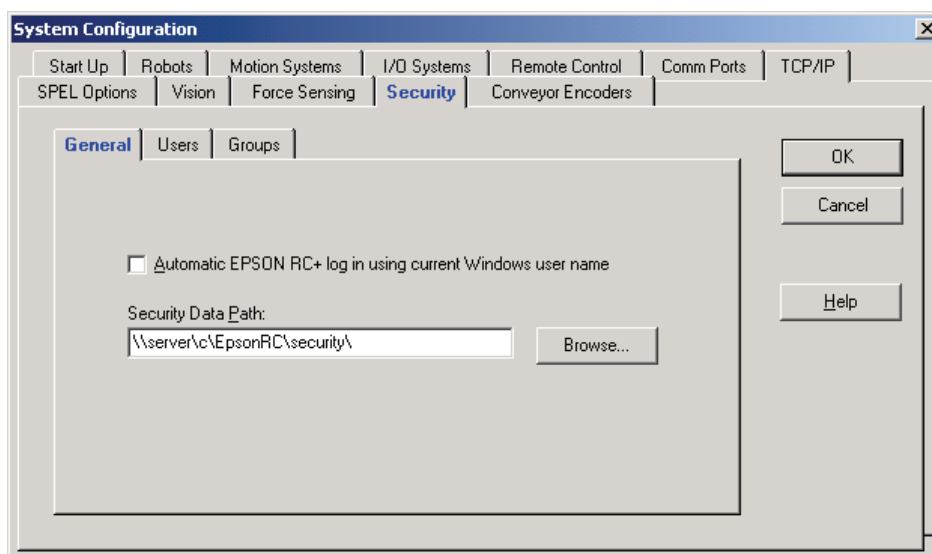
EPSON RC+ requires a path for security files. If you have more than one system on a network, it is recommended that you setup the security files path for all systems to reside on one server on the network.

To administer EPSON RC+ security:

1. Start EPSON RC+.
2. Select Setup | System Configuration.
3. Click the Security tab. If the Security tab is disabled, then check your software key.
4. On the General tab, type in the path for your security files or click the Browse button.
5. Click the Users tab.
6. For each user on your systems, click the New button. Each new user belongs to the Guest group by default. Click in the group field, then click the drop down button to select the desired group.

General Tab

This tab allows you configure security. It will be disabled if the Security Option is not activated, or if the current user does not have Administrator rights.



Automatic log in using current Windows user name

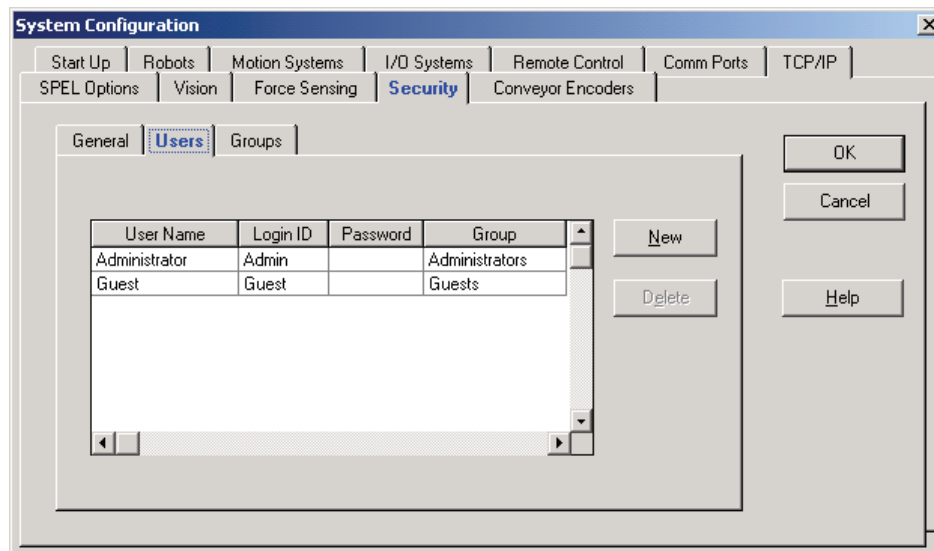
Check this box if you want EPSON RC+ to use the current Windows log in ID. When this feature is active, you will not see a log in dialog when EPSON RC+ starts, unless EPSON RC+ cannot find the user in the Security system.

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.

Users Tab

This tab allows you to add and remove EPSON RC+ 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. By default, there is no password.



To add a user

1. Click the [New](#) button.
2. A new user will be added to the end of the list.
3. Click in the Group field for the new user.
4. Click the drop down button in the field and select the group for the user. See below to create groups.

To delete a user

1. Click in any field in the row for the user you want to delete.
2. Click the [Delete](#) button.
3. You will get a confirmation message to delete the user.

To change a user's group

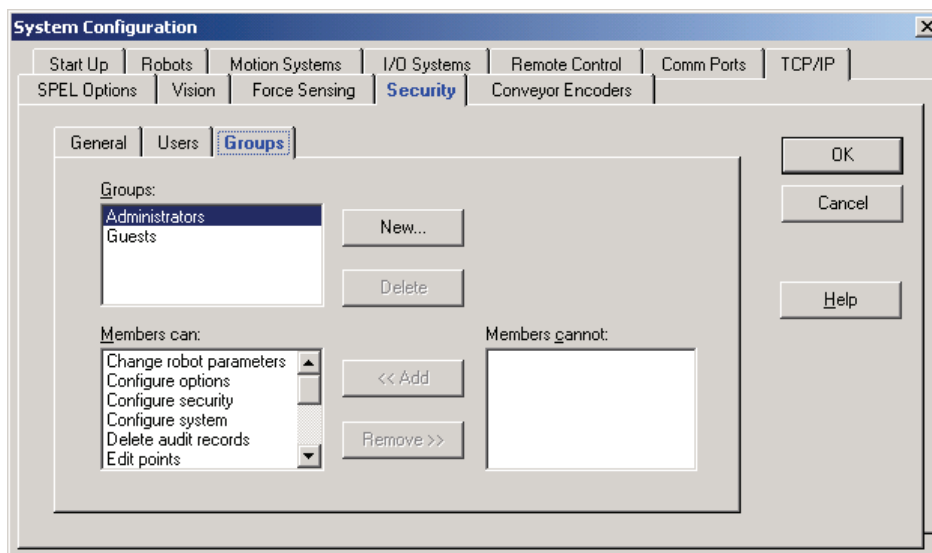
1. Click in the Group field for the user you want to change.
2. Click the drop down button in the field and select the new group.

Editing Name, LogID and Password

1. Click in the field you want to change.
2. Edit the field.
3. All fields are not case sensitive.

Groups Tab

This tab allows you to configure user groups. Every EPSON RC+ 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 **New** button.
2. Type in the name for the group.

To delete a group

1. Click on the group you want to delete.
2. Click the **Delete** button.
3. You will get a confirmation message to delete the group.

Changing rights for groups

1. Select the group you want to change rights for. Note that you cannot change rights for Administrators or Guests.
2. To add rights, click one or more rights in the Users Cannot list. Hold down the Ctrl key to select more than one individual right, or hold down the shift key to select an entire group. Click the <<Add button to add the new right(s) for the group.
3. To remove rights, click one or more rights in the Users Can list.
4. Click the Remove>> button to remove the rights.

Group Rights


The list below shows the rights that are available for user groups. Remember that Administrators have all rights and guests have no rights.

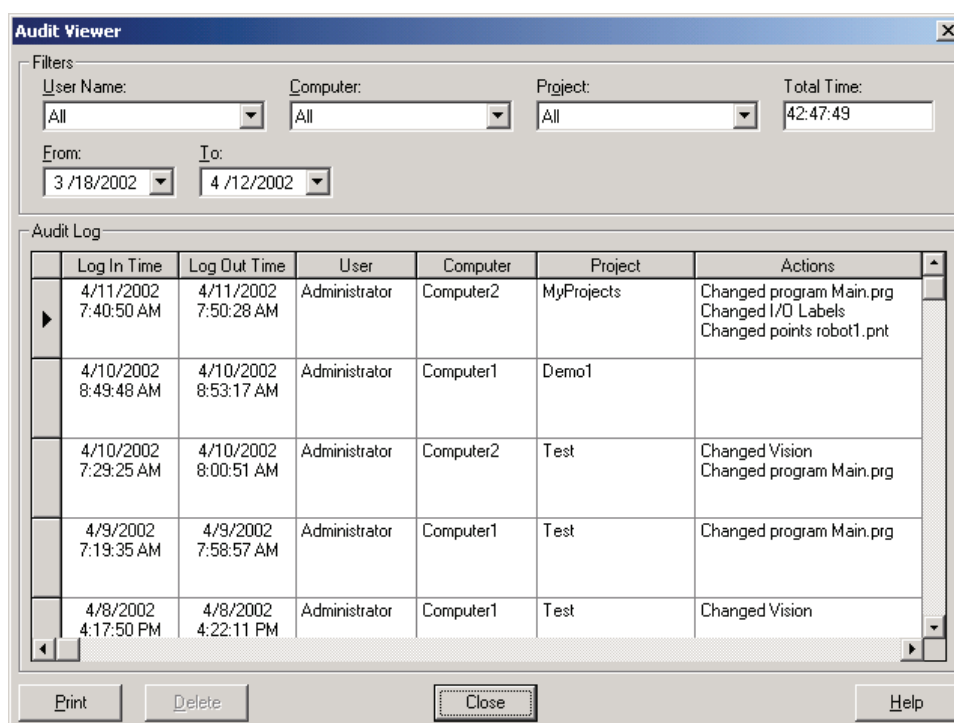
Right	Description
Change robot parameters	Users can open the Robot Parameters dialog and change settings.
Configure options	Users can change option settings in Setup Options.
Configure security	Users can configure the security system.
Configure system	Users can configure the entire EPSON RC+ system.
Delete audit records	Users can delete audit records in Tools Audit Viewer.

Edit points	Users can change points in the point editor.
Edit programs	Users can edit programs.
Edit project	Users can edit projects.
Edit vision	Users can change vision parameters.
Jog and teach	Users can open the Jog and Teach dialog.
Teach points	Users can teach and delete points from the Jog and Teach dialog.
Turn on outputs	Users can turn outputs on or off.
Change memory I/O	Users can turn memory I/O bits on or off.
Use monitor window	Users can open the monitor window and execute commands.
View audit log	Users can view the Audit log.

Security Audit

When the Security Option is enabled, EPSON RC+ will keep track of who logs in to the system and actions performed.

Activity is stored in a Microsoft Access compatible data base in the security data path. To view the audit log, select Audit Viewer  from the Tools menu.



SPEL⁺ Security Commands

Here are the SPEL⁺ commands that are enabled with the Security Option. For details, please see the online help or SPEL⁺ Language Reference Manual.

LogIn	Logs in the application as another user at runtime.
GetCurrentUser\$	Returns the login ID of the current user.

Conveyor Tracking

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+ 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 a 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.

A total of 16 physical conveyors can be defined on each system. A physical conveyor has one encoder whose signals are received by an ISA encoder board.

Up to 16 logical conveyors can be defined in each project. A logical conveyor is defined by a physical conveyor, a robot, and vision or sensor configuration.

Multiple robots and multiple conveyors are supported. For example, 2 robots can be used to pick parts from one physical conveyor.

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.

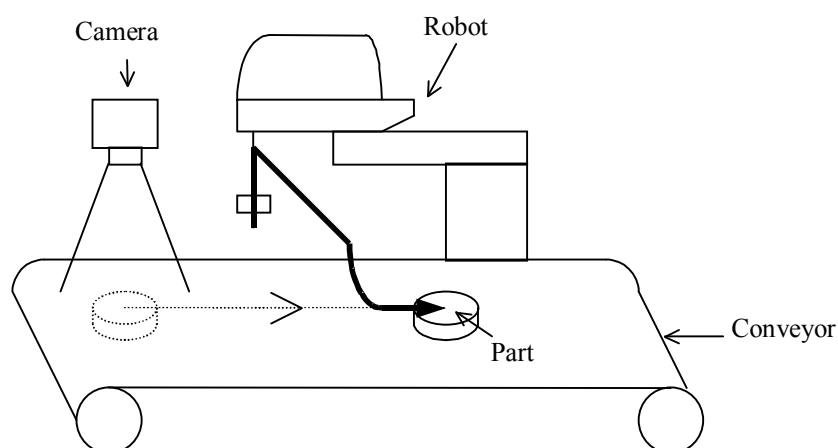


Fig 1.1 Tracking Conveyor System

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.
4. After finding and picking up all parts, conveyor moves by the specified distance again.

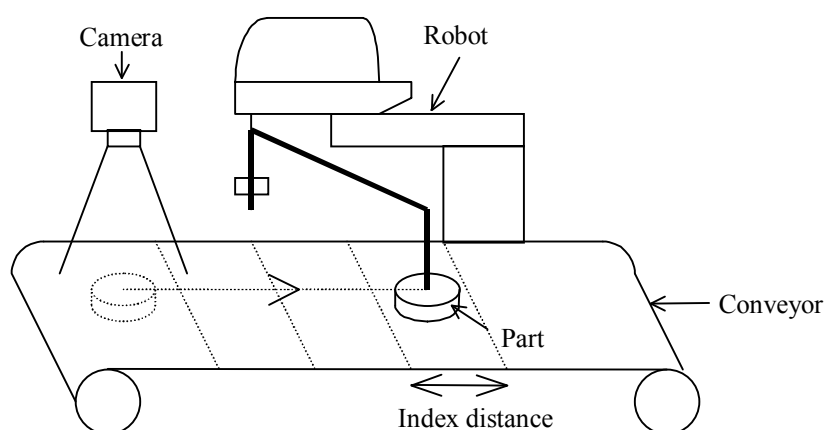


Fig 1.2 Indexed Conveyor System

Hardware Installation

To use conveyor tracking, you must install encoders for each physical conveyor on the system. Each encoder is wired to a single channel on a PG (Pulse Generator) board. Each board can accommodate up to 4 encoders. A trigger input is also provided for each encoder to latch position, such as when used with a strobed vision camera.

PG Board Specifications

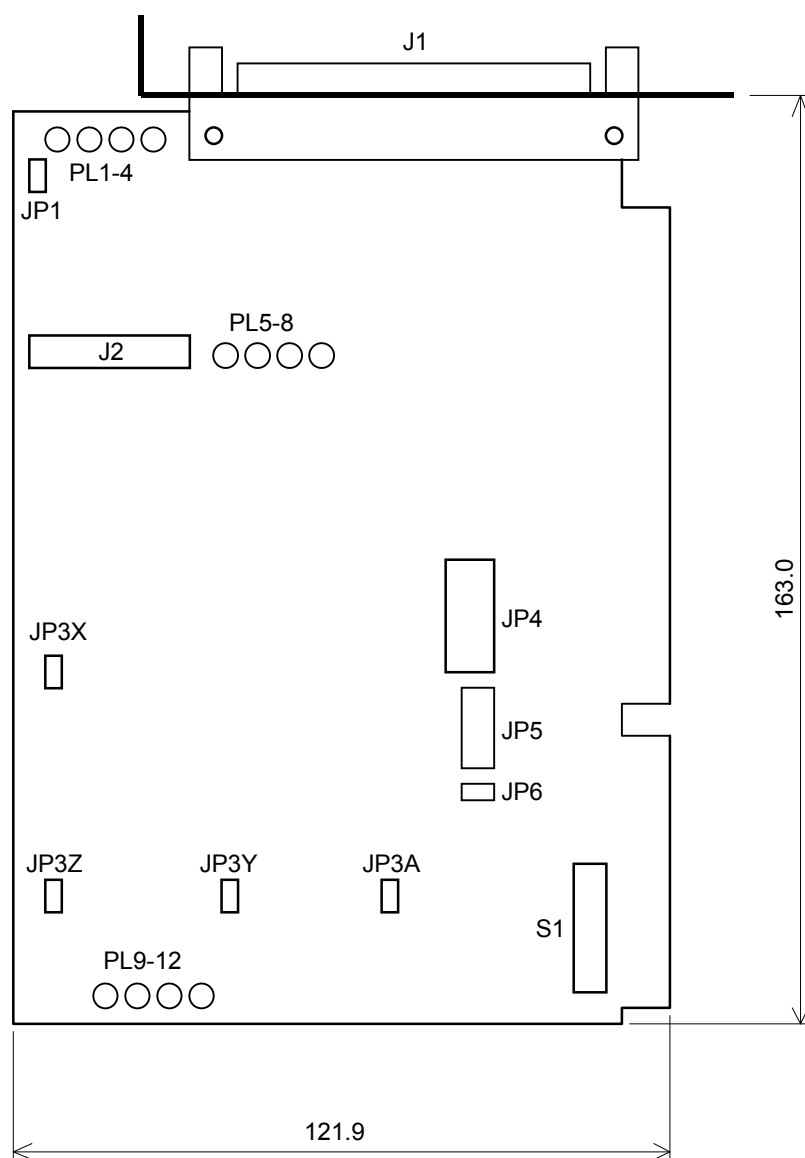
The table below shows the specification for the PG board.

Board Name	C-863
Compatible Controller	RC520, RC420
Bus Specification	ISA
Board Extension Capability	4 boards maximum
Encoder channels	4 channels / board
Encoder Type	ABZ phase differential input (RS-422 line receiver)
Input Pulse Rate	1 PPS ~ 3.3 MPPS
Input Signal	Conveyor pulse latch input
Board Address	Set by DIP switch according the board number. (See DIP Switch Settings later in this chapter).
connector	DX10A - 100S (Hirose Electric Co.,Ltd.)
Power Supply	24V \pm 2V 140mA or 5V \pm 0.25V 700mA (Jumper selectable)

The following encoder models have been tested:

OMRON	E6B2-CWZ1X
SUNX	OLE-38-2000L05
TAMAGAWA	TS5312N512-2000C/T

The figure below shows the layout of the PG board.



DIP Switch Settings

The board address is set by DIP switch (S1) on the PG Board. S1 must be set according to the board number, as shown in the following table.

Board #	Address	DIP S1								
		1 (A7)	2 (A8)	3 (A9)	4 (A10)	5 (A11)	6 (A12)	7 (A13)	8 (A14)	9 (A15)
1	500h	On	Off	On	Off	On	On	On	On	On
2	580h	Off	Off	On	Off	On	On	On	On	On
3	600h	On	On	Off	Off	On	On	On	On	On
4	900h	On	Off	On	On	Off	On	On	On	On

On = 0, Off = 1

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 Controller and keep a written record of the address setting and the board No.

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.

Jumper Settings

JP1 sets the voltage of the external power supply used for the encoder(s).

JP1 Open: +24VDC \pm 2V

JP1 Closed: +5VDC \pm 0.25V

The remaining jumpers are reserved and should not be changed.

JP3 X, Y, Z, A: All jumpers installed

JP4: No jumper

JP5: No jumper

JP6: Jumper installed

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

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 Counter1	60	-	-	Not used
11	In	TRG2	Trigger input for Counter2	61	-	-	Not used
12	In	TRG3	Trigger input for Counter3	62	-	-	Not used
13	In	TRG4	Trigger input for Counter4	63	-	-	Not used
14	In	EXTV	External power supply for Input circuit(*1)	64	In	EXTV GND	External power supply GND for Input circuit (*1)
15	In	EXTV	External power supply for Input circuit(*1)	65	In	EXTV GND	External power supply GND for Input circuit (*1)
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
21	-	-	Not used	71	-	-	Not used
22	-	-	Not used	72	-	-	Not used
23	-	-	Not used	73	-	-	Not used
24	-	GND	GND	74	-	GND	GND
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 Counter1	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	-	GND	GND	90	-	GND	GND
41	In	+A2	Phase +A signal for Counter 2	91	In	+A4	Phase +A signal for Counter 4

Pin	Dir	Signal	Description	Pin	Dir	Signal	Description
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 Counter4
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
47	-	-	Not used	97	-	-	Not used
48	-	-	Not used	98	-	-	Not used
49	-	-	Not used	99	-	-	Not used
50	-	GND	GND	100	-	GND	GND

*1 Set the jumper JP1 according to the external power supply voltage.

<Pins No. 25 through 30, 41 through 46, 75 through 80, 91 through 96>

Connect the pin No. shown above with encoder output (+A, -A, +B, -B, +Z, -Z).

<Pins No. 10 through 13>

When the conveyor pulse is latched by external signal, connect the pin No. shown above with latch signal. Exactly when the signal is turned OFF to ON, the encoder pulse is latched.

<Pins No. 14, 15, 64 and 65>

When using the pin No. 10 through 13, connect external power with the pin No. shown above.

When not using the pin No. 10 through 13, it is not necessary to connect external power with the pin No. shown above.

Signal Assignments: PG board connector terminal block 1

The signals on the 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.

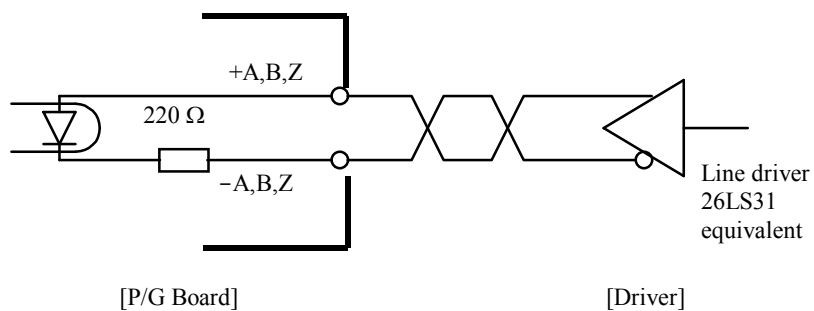
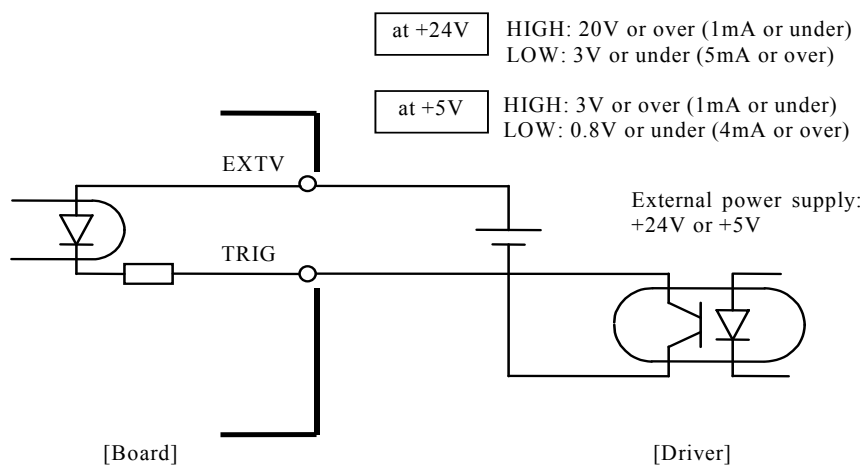
Pin	Signal	Description	Pin	Signal	Description
1 (16)	-	Not used	26 (32)	-	Not used
2 (17)	-	Not used	27 (33)	-	Not used
3 (18)	-	Not used	28 (34)	-	Not used
4 (19)	-	Not used	29 (35)	-	Not used
5 (20)	-	Not used	30 (36)	-	Not used
6 (21)	-	Not used	31 (37)	-	Not used
7 (22)	-	Not used	32 (38)	-	Not used
8 (23)	-	Not used	33 (39)	-	Not used
9 (24)	GND	Ground	34 (40)	GND	Ground
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
16 (31)	-	Not used	41 (47)	-	Not used
17 (48)	-	Not used	42 (49)	-	Not used
18 (9)	-	Not used	43 (50)	GND	Ground
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)	EXTV GND	External power supply ground

Signal Assignments: PG board connector terminal block 2

The signals on the 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)	GND	Ground	34 (90)	GND	Ground
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	Ground
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)	EXTV GND	External power supply ground

Encoder Input Circuit

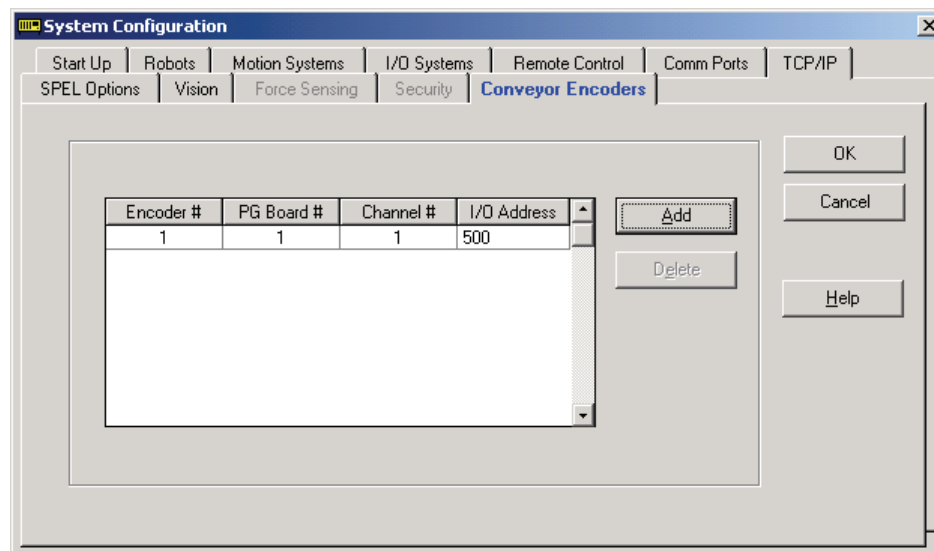


Conveyor encoder configuration

Before you can create any conveyors in a project, you must first add conveyor encoders to the system. Each physical conveyor must have an encoder.

First, you must install one PG board for every four encoders in the PC Control Unit and wire the encoders to the board(s). Please refer to the Hardware Installation section of this chapter for details.

To define system encoders in EPSON RC+, select Setup | System Configuration and click the **Conveyor Encoders** tab.



Click the **Add** button to add an encoder. Note that all encoders must be added sequentially.

You can delete the last encoder in the list. Select it, then click the **Delete** button.

Verifying new encoder operation

After wiring one or more new encoders and adding them to RC+ (as described in the previous section), follow these steps to verify operation.

1. Start RC+.
2. Create a new project called TestCnv.
3. From Tools | Conveyor Tracking, add a new sensor conveyor for each encoder. Make sure that conveyor 1 uses encoder 1, conveyor 2 uses encoder 2, etc. It is not necessary to calibrate the conveyors.
4. Now you can use the Cnv_Pulse function to read pulses from each encoder from a program or from the monitor window.

For example, execute this print statement from the monitor window to read the pulses from encoder 1. Then move the conveyor and execute the command again.

```
>? cnv_pulse(1)
```

Or you can use a simple program as shown below. Start the program, and move the conveyor. The Cnv_Pulse values should change as the conveyor is moved.

```
Function main
Do
    Print Cnv_Pulse(1)
    Wait .5
Loop
End
```

Conveyor Tracking Commands

All Conveyor Tracking commands begin with the same prefix: "Cnv_". Here is a list of all of the commands. For details, please see the online help or SPEL⁺ Language Reference Manual.

Cnv_AbortTrack	Aborts a motion command to a conveyor queue point.
Cnv_DownStream	Returns the downstream limit for the specified conveyor.
Cnv_Fine	Sets / returns the value of Cnv_Fine for one conveyor.
Cnv_Name\$	Returns the name of the specified conveyor.
Cnv_Number	Returns the number of a conveyor specified by name.
Cnv_Point	Returns a robot point in the specified conveyor's coordinate system derived from sensor coordinates.
Cnv_PosErr	Returns deviation in current tracking position compared to tracking target.
Cnv_Pulse	Returns the current position of a conveyor in pulses.
Cnv_QueueAdd	Adds a robot point to a conveyor queue.
Cnv_QueueGet	Returns a point from the specified conveyor's queue.
Cnv_QueueLen	Returns the number of items in the specified conveyor's queue.
Cnv_QueueList	Displays a list of items in the specified conveyor's queue.
Cnv_QueueMove	Moves data from upstream conveyor queue to downstream conveyor queue.
Cnv_QueueReject	Sets / returns and displays the queue reject distance for a conveyor.
Cnv_QueueRemove	Removes items from a conveyor queue.
Cnv_QueueUserData	Sets / returns and displays user data associated with a queue entry.
Cnv_RobotConveyor	Returns the conveyor being tracked by a robot.
Cnv_Speed	Returns the current speed of a conveyor.
Cnv_Trigger	Latches current conveyor position for the next Cnv_QueueAdd statement.
Cnv_Upstream	Returns the upstream limit for the specified conveyor.



To track a part as the conveyor moves, you must use Cnv_QueueGet in a motion command statement. For example:

```
Jump Cnv_QueueGet(1) ' this tracks the part
```

You cannot assign the result from Cnv_QueueGet to a point and then track it by moving to the point.

```
P1 = Cnv_QueueGet(1)
Jump P1 ' this does not track the part
```

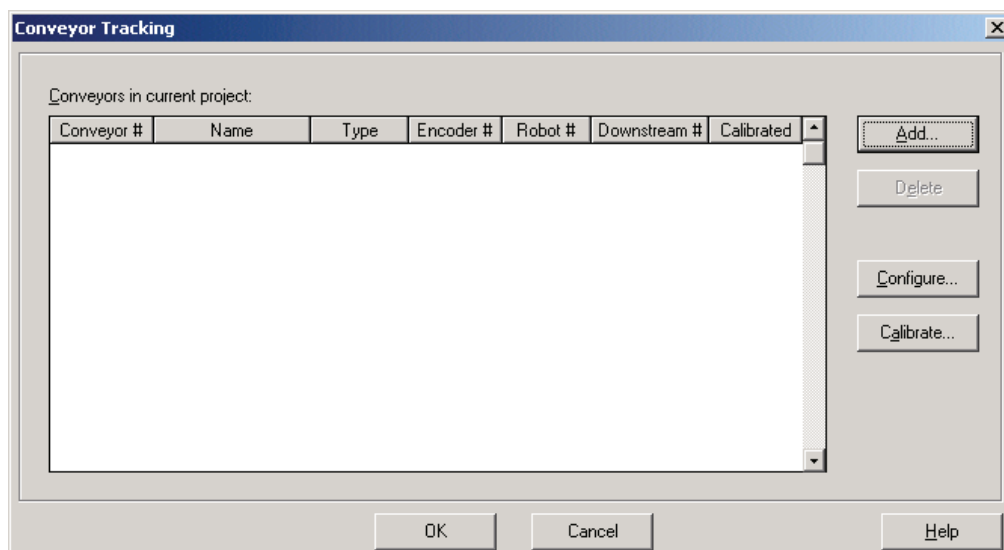
When you assign the result from Cnv_QueueGet to a point, the coordinate values correspond to the position of the part when the point assignment was executed.

Creating conveyors in a project

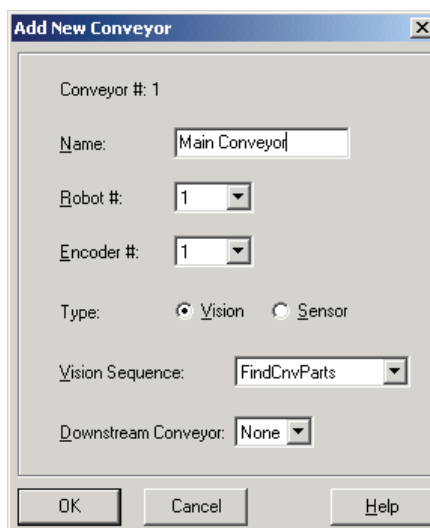
Conveyors are configured for each EPSON RC+ project. Up to 16 conveyors can be created per project. A conveyor is a logical entity that combines a robot with one or more conveyors.

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, select Tools | Conveyor Tracking to open the Conveyor Tracking configuration dialog.



To add a conveyor, click the **Add** button. The following dialog will appear.



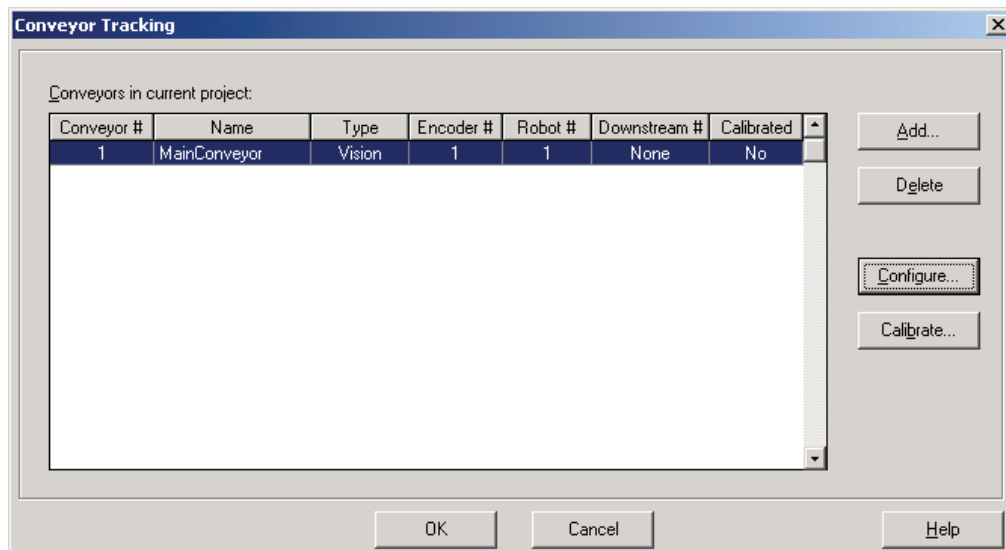
Enter a name for the conveyor, then specify the robot number, encoder number, and type.

If there will be another conveyor defined as a downstream conveyor, you can select it. Note that it must already be defined.

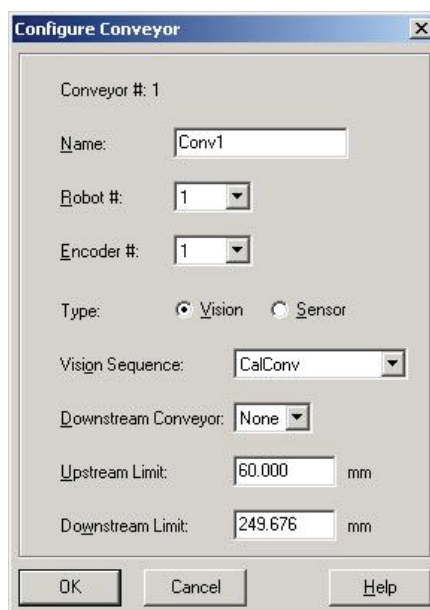
Configure conveyors

After a conveyor has been created, you can change its parameters.

1. Select Tools | Conveyor Tracking.
2. Click on the conveyor you want to change.
3. Click the Configure button. You can also double click the conveyor.



4. The following dialog will be displayed. Edit any one of the configuration options.



5. Click OK to save changes.
6. If you changed Robot #, Encoder #, Type, or Vision Sequence, then you need to calibrate the conveyor again.

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 and programming.

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 an RS170 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. This method is more expensive.

If you are using multiple asynchronous reset cameras in multiple tasks, you must use SyncLock to lock the vision system during VRun and waiting until the picture is acquired.

For example:

```
SyncLock 1      ' Lock vision for this task
VRun FindPart
On strobe, .2
Do
  VGet FindPart.AcquireState, state
Loop Until state = 3
SyncUnlock 1    ' Unlock vision
```

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 must be capable of finding three calibration marks or parts. You must use marks or parts that can be aligned with the robot end effector.

Vision conveyor calibration

Follow these steps to calibrate a vision conveyor:



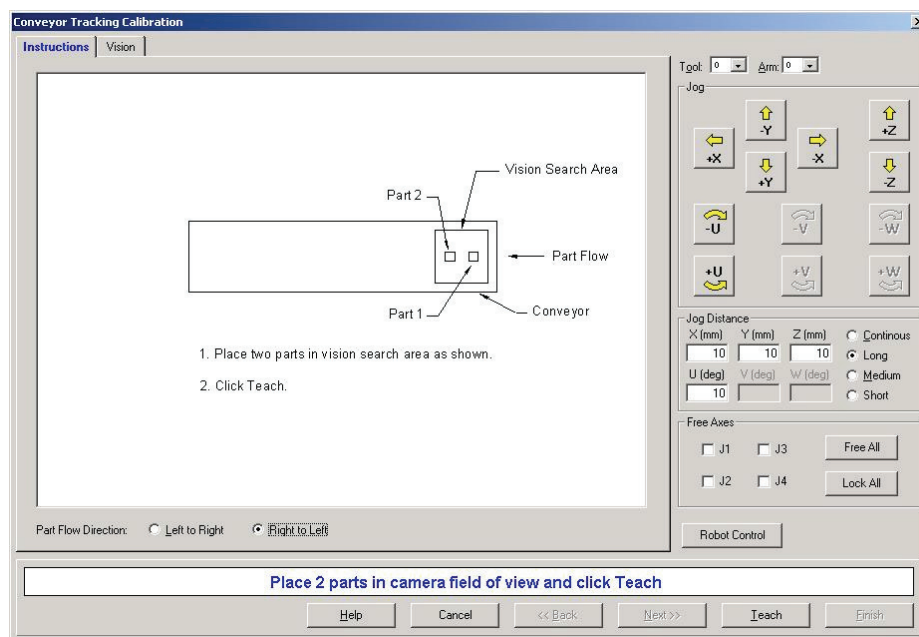
When teaching part positions with the robot during calibration, it is important to position X, Y, and Z of each point accurately. The conveyor is calibrated in X, Y, Z, U, V and W.

1. Select Tools | Conveyor Tracking.
2. Select the conveyor you want to calibrate.
3. Click the **Calibrate** button. The Conveyor Tracking Calibration wizard will appear.

- 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.

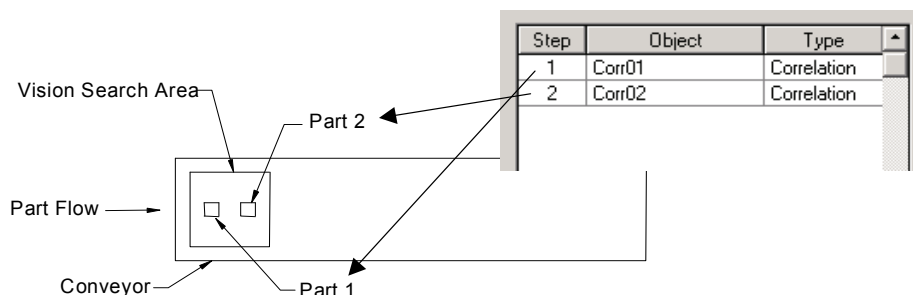
Select the Part Flow Direction to best match the conveyor you are calibrating. The instruction pictures will change according to the setting. Part Flow Direction is only used to aid in the instructions. It has no affect on the calibration.

For the first wizard step, place two parts on the conveyor as shown in the wizard instruction picture so that they are within the search areas for each of the parts in the vision calibration sequence.



The vision calibration sequence must find both parts. When using a simple sequence that uses one object for each part, the first object of the sequence must be taught by the robot as part 1. The second object of the sequence must be taught by the robot as part 2.

The two parts can be anywhere in the field of view. However, 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.



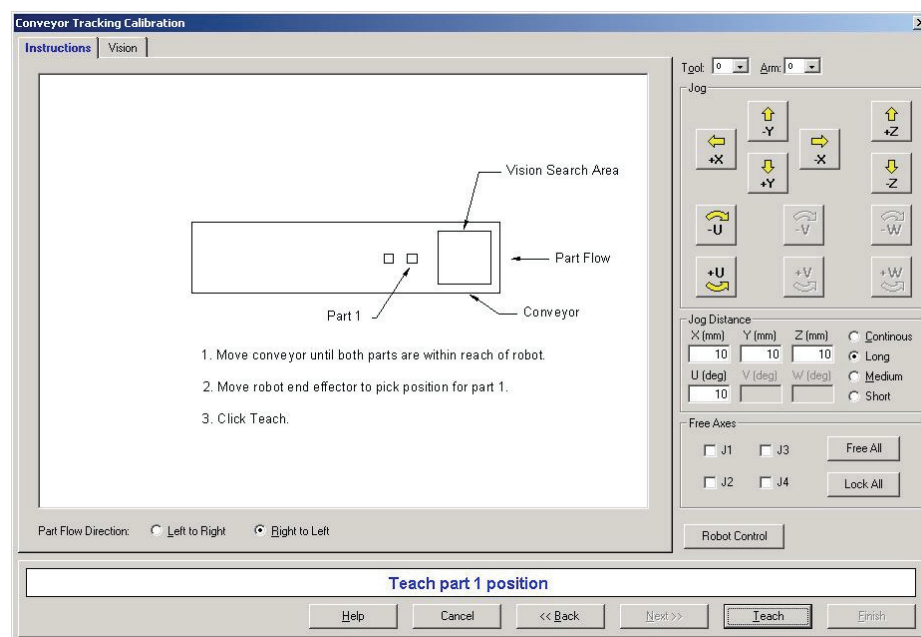
If the vision sequence used to locate the parts is more complex, then the last two objects of the sequence must have the location for parts 1 and 2. The object that finds part 1 must be the second to last object. The object that finds part 2 must be the last object. For

example, the following sequence uses two Point objects for the locations of part 1 and part 2. Step 5 locates part 1 and step 6 locates part 2.

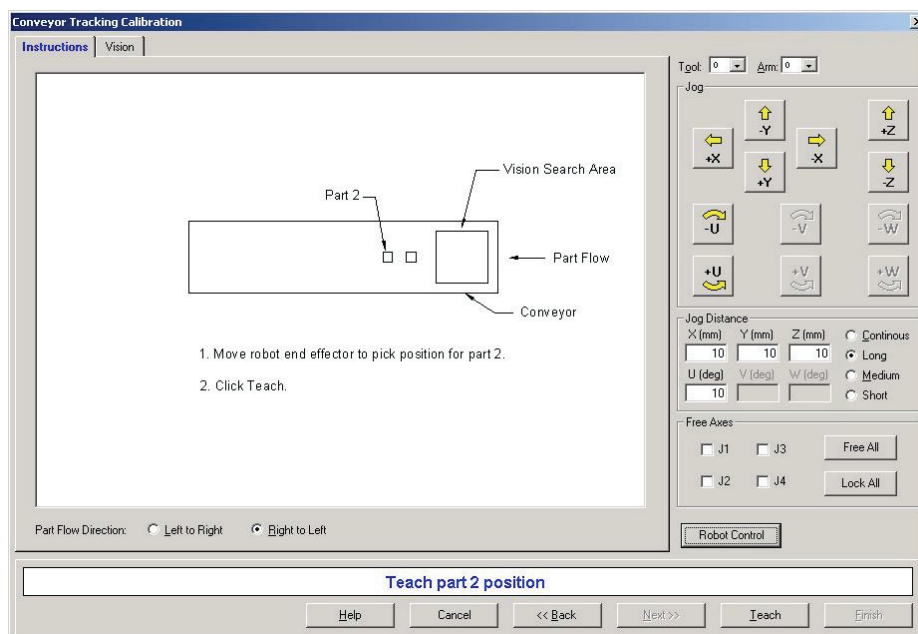
Step	Object	Type	Description
1	Corr01	Correlation	Locates feature on part 1
2	Frame01	Frame	Single point frame using Corr01 as origin
3	Corr02	Correlation	Locates feature on part 2
4	Frame02	Frame	Single point frame using Corr02 as origin
5	Point01	Point	Point in Frame01 that locates part 1.
6	Point02	Point	Point in Frame02 that locates part 2.

The last two steps of the sequence should correspond with the locations of Part 1 and Part 2. Click the Vision tab to see live video. The camera orientation may not be the same as the picture. Use the camera video to ensure that the parts are within the correct search area for each. Click the **Teach** button. If the Vision tab is selected when you click **Teach**, you will see the vision sequence graphical results displayed. In this case, the wizard will not advance to the next step and you must click the Next button to view the next step. This allows you to click **Teach** more than one time in case you want to adjust the parts.

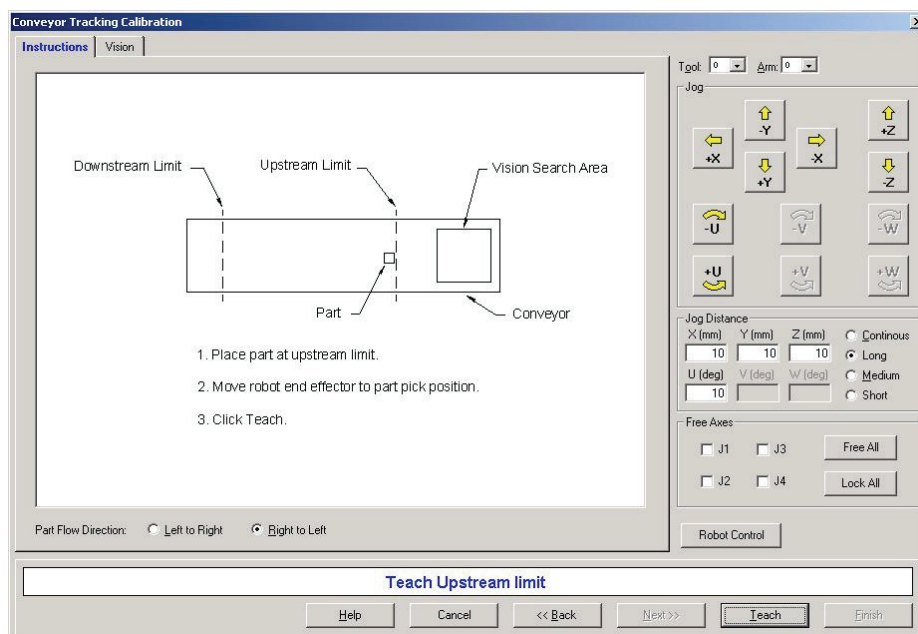
5. Move the conveyor by hand until both parts are within reach of the robot. Do not move the parts, only the conveyor. Move the robot end effector to the pick position for Part 1, as shown in the instruction picture. Click the **Teach** button.



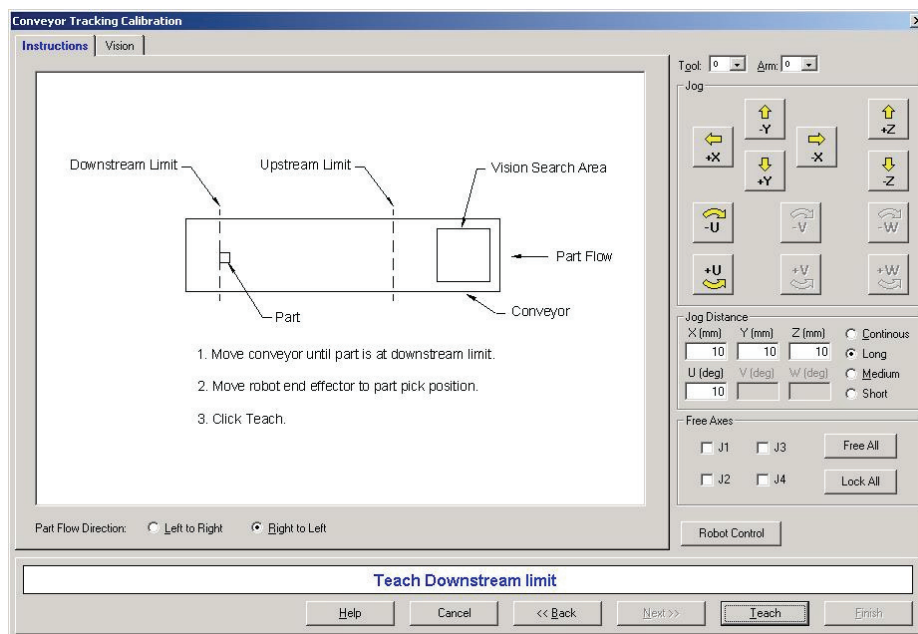
- Move the robot end effector to the pick position for Part 2, as shown in the instruction picture. Click the **Teach** button.



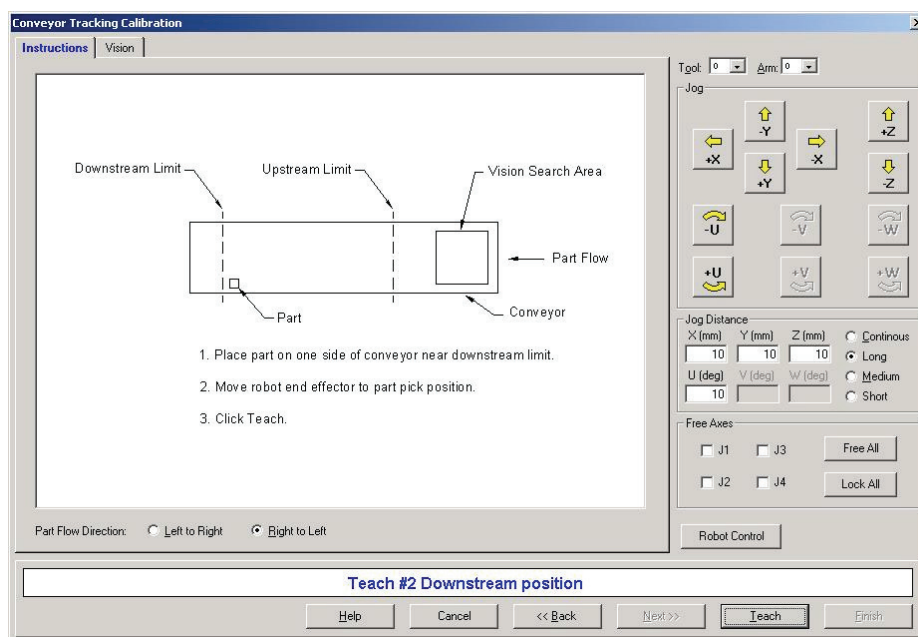
- Now move or place the part at the upstream limit. The upstream limit marks the beginning of the area where the robot will pick up parts. Move the robot end effector above the part. In this step, only the robot X coordinate is used, so Y and Z are not critical. Click the **Teach** button.



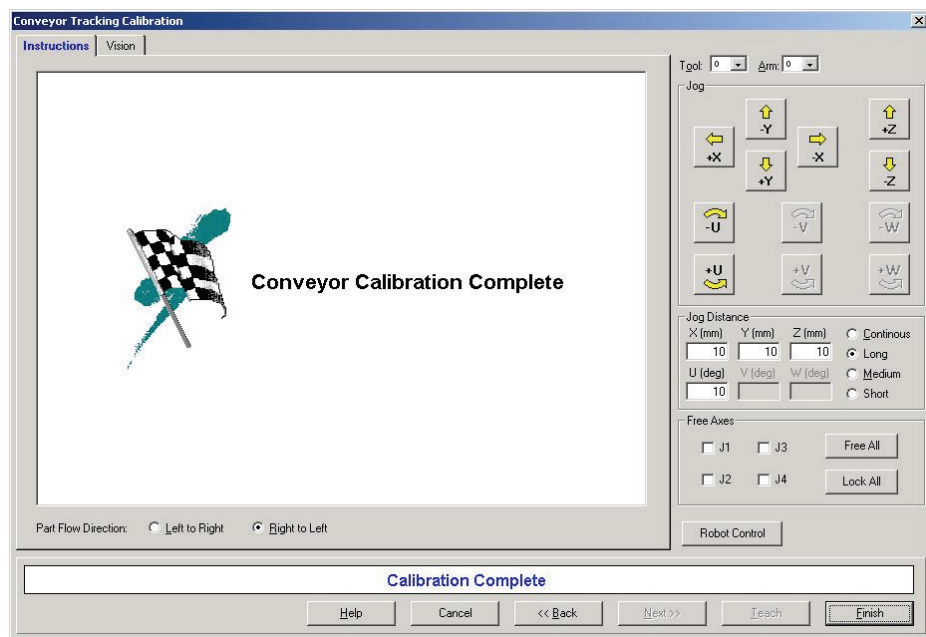
8. Now move the conveyor so the part is now at the downstream limit. Do not move the part, only the conveyor. The downstream limit marks the end of the area where the robot will pick up parts. Move the robot end effector to the part pick position. Click the **Teach** button.



9. 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. Move the robot end effector to the part pick position. Click the **Teach** button.



10. The calibration complete picture will be displayed. Click the **Finish** button.



Vision conveyor programming

Typically, two tasks are used to operate a vision conveyor. One task find parts with the vision system and adds them to the conveyor queue. The other task checks for parts in the conveyor queue that are in the pickup area. When a part is in the pickup area, the robot is commanded to pick up the part and place it.

The following example shows two tasks. The scanning task uses the vision system to find parts and add them to the conveyor queue. There are two examples for the scanning task.

ScanConveyorNonStrobed does not use a strobe lamp and hardware trigger. In this case, Cnv_Trigger must be called before running the vision sequence. ScanConveyorStrobed uses a strobe lamp and hardware conveyor trigger. PickParts waits for parts to be present in the pickup area and commands the robot to pick and place each part.

If you are using an asynchronous reset camera and strobe, then the strobe trigger should also be wired to the trigger on the PG board. In this case, the vision sequence RuntimeAcquire property must be set to Strobed.

```
Function main
  Xqt ScanConveyorNonStrobed
  Xqt PickParts
Fend

Function ScanConveyorNonStrobed
  Integer i, numFound
  Real x, y, u
  Boolean found
  Do
    Cnv_Trigger 1
    VRun FindParts
    VGet FindParts.Parts.NumberFound, numFound
    For i = 1 to numFound
      VGet FindParts.Parts.CameraXYU(i), found, x, y, u
      Cnv_QueueAdd 1, Cnv_Point(1, x, y)
    Next i
    Wait .1
  Loop
Fend

Function ScanConveyorStrobed
  Integer i, numFound, state
  Real x, y, u
  Boolean found
  Do
    VRun FindParts
    Do
      VGet FindParts.AcquireState, state
    Loop Until state = 3
    VGet FindParts.Parts.NumberFound, numFound
    For i = 1 to numFound
      VGet FindParts.Parts.CameraXYU(i), found, x, y, u
      Cnv_QueueAdd 1, Cnv_Point(1, x, y)
    Next i
    Wait .1
  Loop
Fend

Function PickParts
  Do
    Do While Cnv_QueueLen(1, CNV_QUELEN_DOWNSTREAM) > 0
      Cnv_QueueRemove 1, 0
    Loop
    If Cnv_QueueLen(1, CNV_QUELEN_PICKUPAREA) > 0 Then
      Jump Cnv_QueueGet(1)
      On gripper
```

```
        Wait .1
        Jump place
        Off gripper
        Wait .1
        Cnv_QueRemove 1, 0
    EndIf
Loop
Fend
```

Sensor conveyors

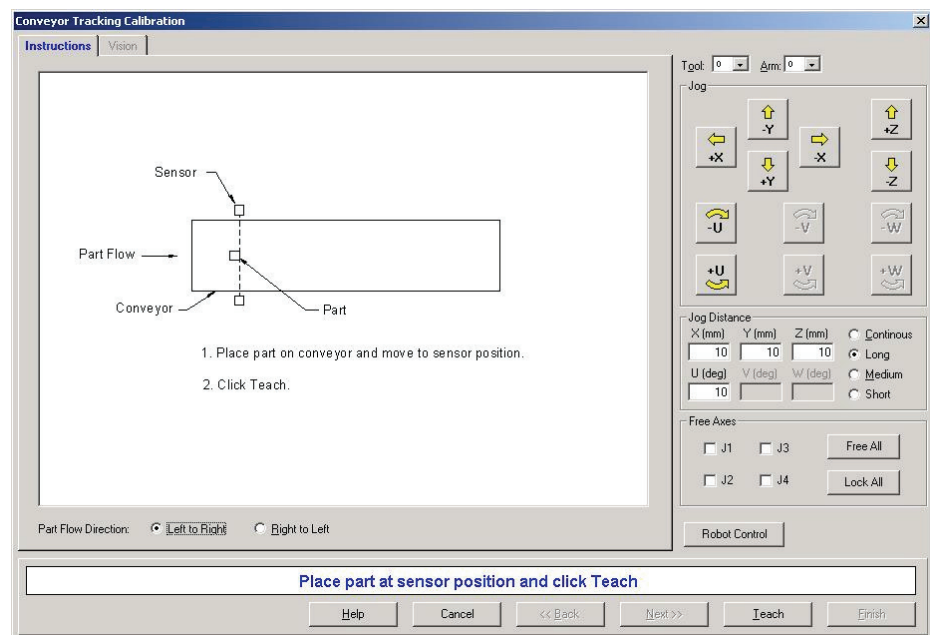
Sensor conveyor calibration

Follow these steps to calibrate a sensor conveyor:

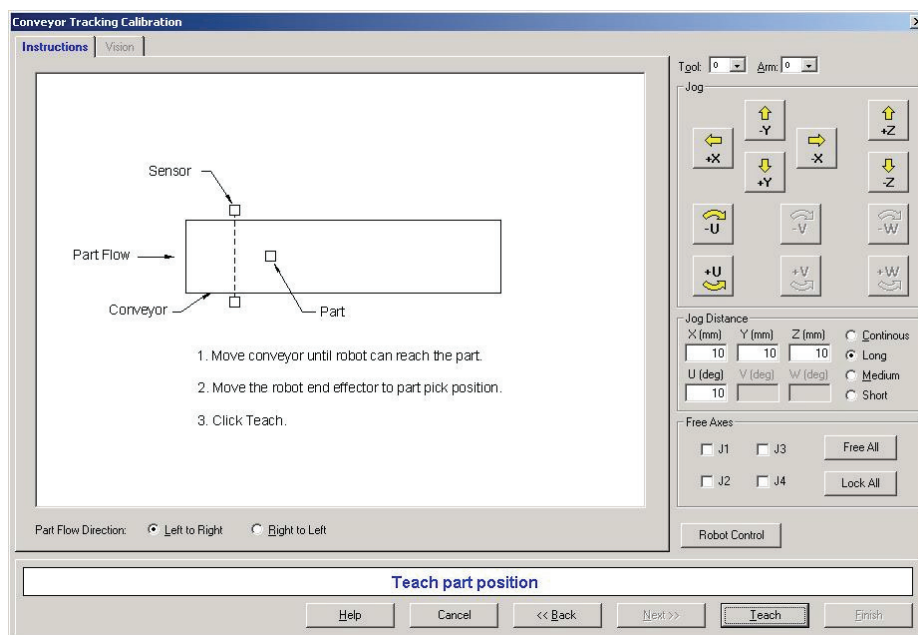


When teaching part positions with the robot during calibration, it is important to position X, Y, and Z of each point accurately. The conveyor is calibrated in X, Y, Z, U, V and W.

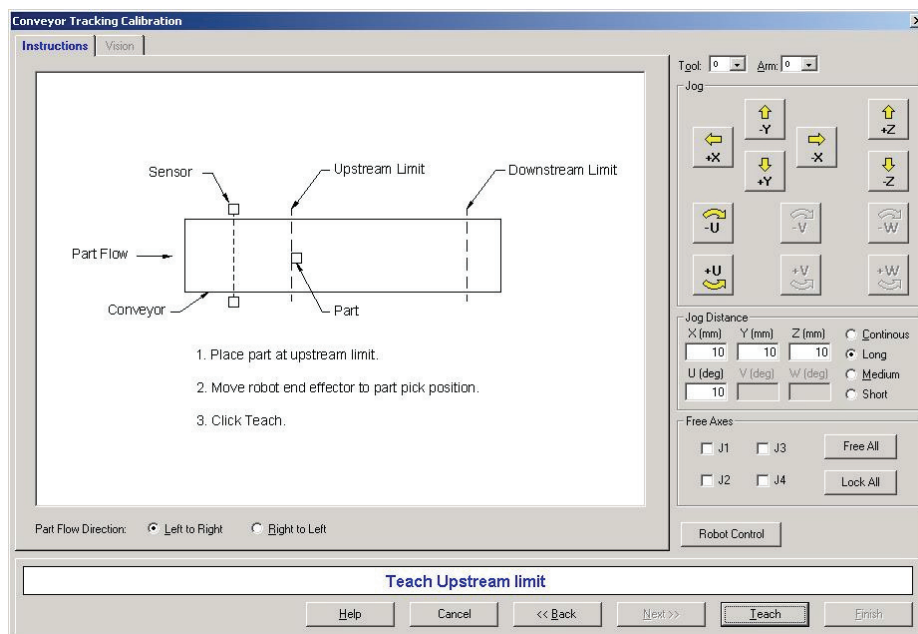
1. Select Tools | Conveyor Tracking.
2. Select the conveyor you want to calibrate.
3. Click the **Calibrate** button. The Conveyor Tracking Calibration wizard will appear.
4. Follow the instructions for each step. Before you can proceed to the next step, you must click the **Teach** button. You can go 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 affect on the calibration.
6. For the first wizard step, place a part on the conveyor and move the conveyor toward the sensor until the sensor just turns on. Click the **Teach** button.



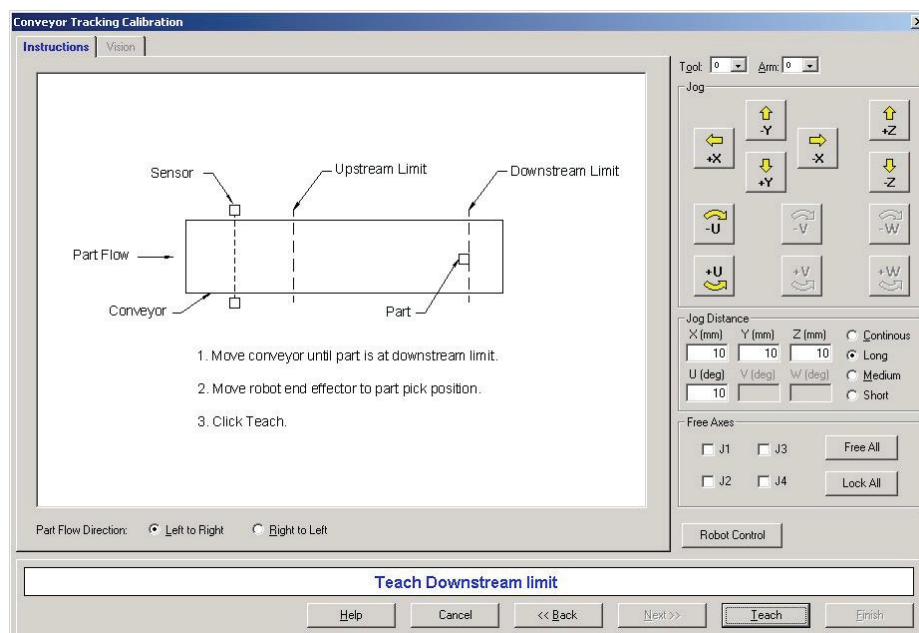
7. Move the conveyor by hand until the part is within reach of the robot. Do not move the part itself, only the conveyor. Move the robot end effector to the part pick up position. Click the **Teach** button



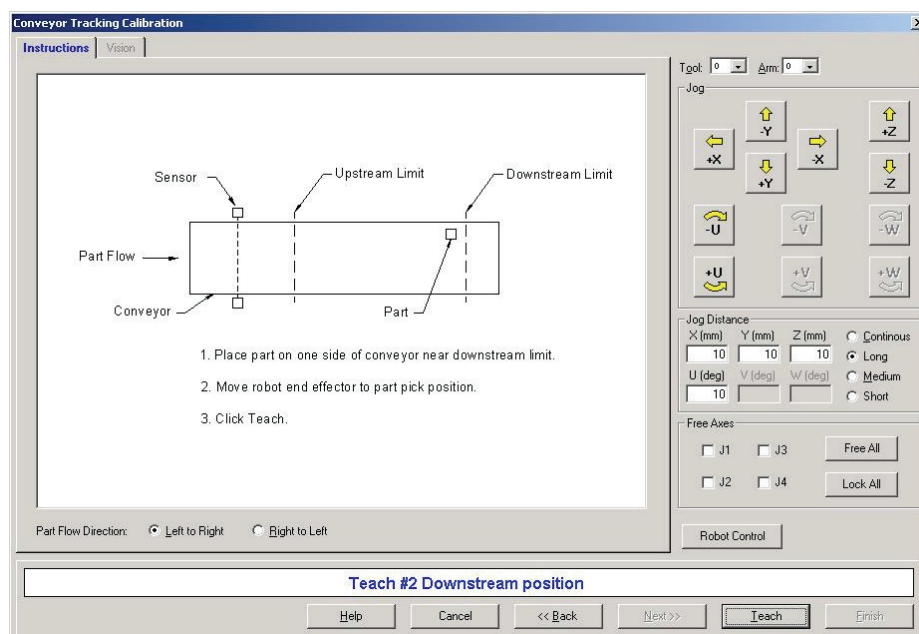
8. Now move or place the part at the upstream limit. The upstream limit marks the beginning of the area where the robot will pick up parts. Move the robot end effector to the part pick position. Click the **Teach** button.



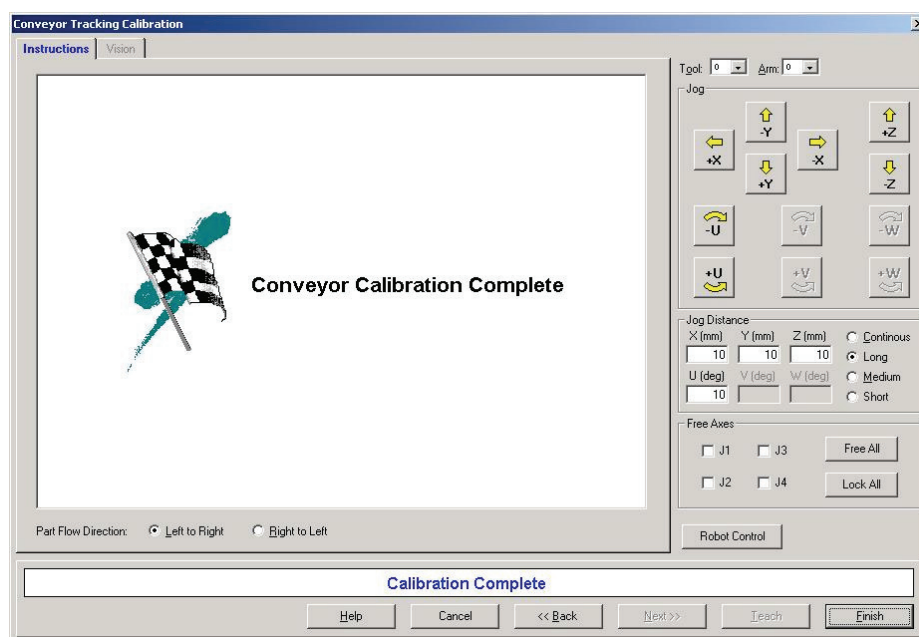
9. Now move or place the part at the downstream limit. The upstream limit marks the end of the area where the robot will pick up parts. Move the robot end effector to the part pick position. Click the **Teach** button.



10. 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. Move the robot end effector to the part pick position. Click the **Teach** button.



11. The calibration complete picture will be displayed. Click the **Finish** button.



Sensor conveyor programming

Typically, two tasks are used to operate a sensor conveyor. One task wait for a part to trip the sensor and add it to the conveyor queue. The other task checks for parts in the conveyor queue that are in the pickup area. When a part is in the pickup area, the robot is commanded to pick up the part and place it.

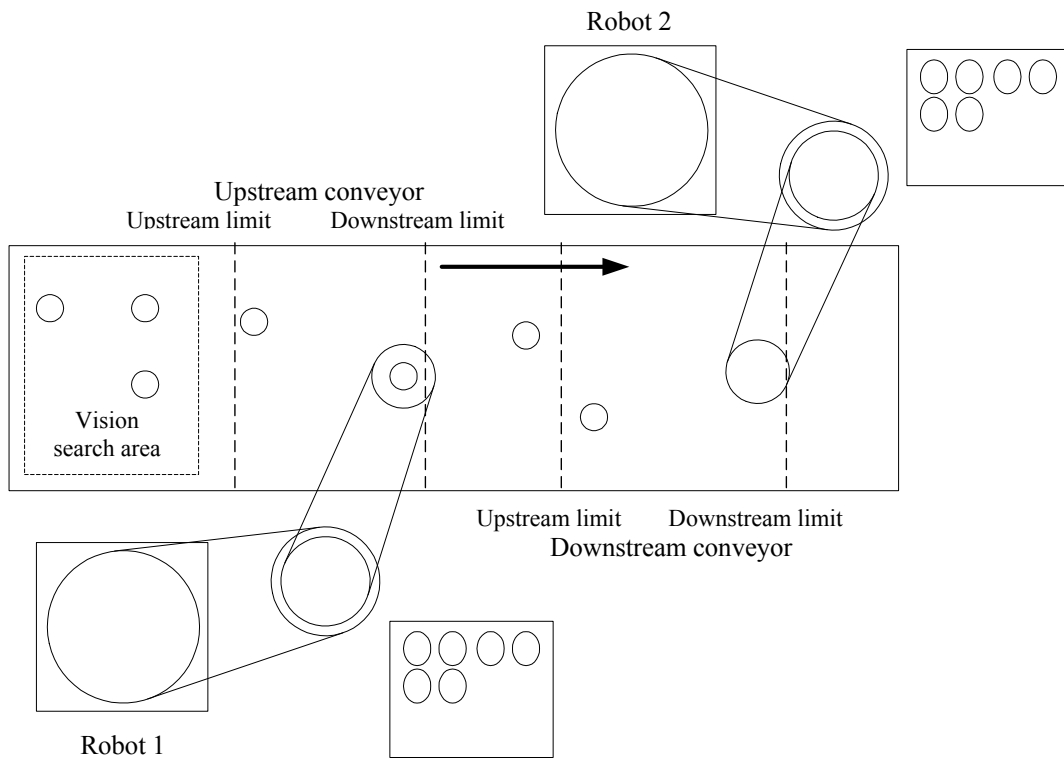
```
Function main
  Xqt ScanConveyor
  Xqt PickParts
Fend

Function ScanConveyor
  Do
    Wait Sw(sensor) = On
    Cnv_Trigger 1
    Cnv_QueueAdd 1, Cnv_Point(1, 0, 0)
    Wait Sw(sensor) = Off
  Loop
Fend

Function PickParts
  Do
    Do While Cnv_QueueLen(1, CNV_QUELEN_DOWNSTREAM) > 0
      Cnv_QueueRemove 1, 0
    Loop
    If Cnv_QueueLen(1, CNV_QUELEN_PICKUPAREA) > 0 Then
      Jump Cnv_QueueGet(1)
      On gripper
      Wait .1
      Jump place
      Off gripper
      Wait .1
      Cnv_QueueRemove 1, 0
    EndIf
  Loop
Fend
```

Multiple conveyors and robots

EPSON RC+ supports multiple logical conveyors and robots. You can use multiple robots on one conveyor, or multiple robots with multiple conveyors. For example, using two robots on one physical conveyor, you can create a logical conveyor for each robot. Parts that are not picked up by the first robot can be picked up by the second robot.



To pick up the same parts with multiple robots

When both robots are handling the same type of parts, you will want to move the parts in the conveyor 1 downstream to the conveyor 2 upstream. This allows parts that were not picked up by robot 1 to be picked up by robot 2.

In this example, we will use two robots on one conveyor.

Configure a conveyor for each robot: conveyor 1 for robot 1 and conveyor 2 for robot 2.

Set the downstream conveyor for conveyor 1 to 2.

In the processing code for robot 1, use `Cnv_QueueMove` to move parts in conveyor 1 downstream area to conveyor 2 upstream area:

```
If Cnv_QueueLen(1, CNV_QUELEN_DOWNSTREAM) > 0 Then
    Cnv_QueueMove 1
EndIf
```

To pick up different part types with multiple robots

When both robots are handling different types of parts, you can create a conveyor for each robot. When the vision system finds parts, it adds the part positions to the correct queue, depending on the part type.

Abort tracking

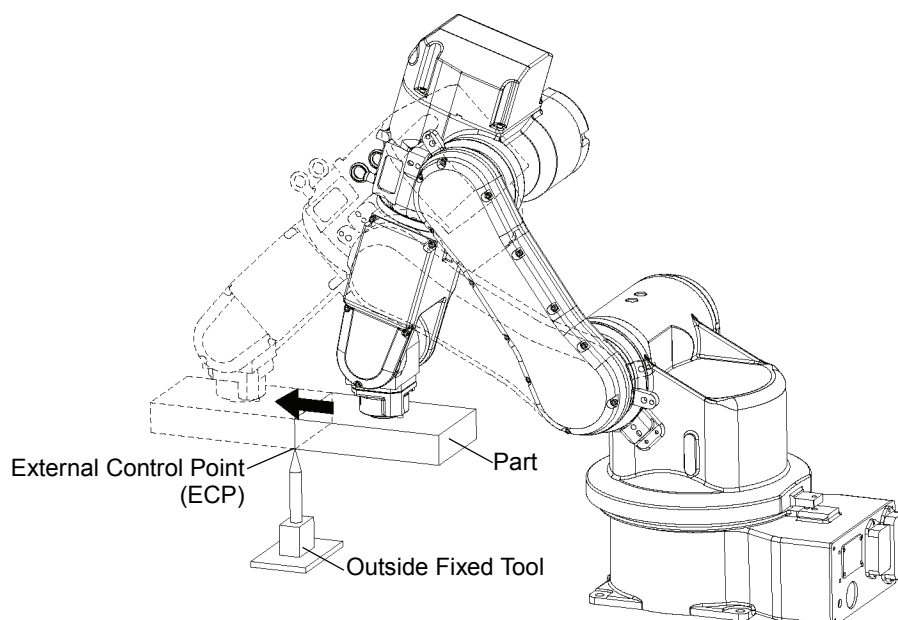
There are some situations when you want to abort tracking a part that moves out of the pickup area while the robot is tracking the part. In this case, use the `Cnv_AbortTrack` command in a separate task that monitors the conveyor queue.

```
Function MonitorDownstream
  Robot 1
  Do
    If Cnv_QueLen(1, CNV_QUELEN_DOWNSTREAM) > 0 Then
      Cnv_AbortTrack 0
    EndIf
    Wait .1
  Loop
Fend
```


ECP Motion

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.



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, Cartesian and 6-Axis robots. Also, it can be used with multi-robot systems.

Up to 15 ECP coordinate systems can be defined.

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. Setting the ECP

The ECP (external control point) is 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.

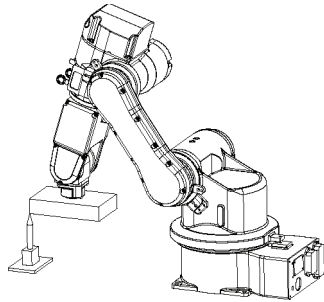
2. Teaching

Teach the point data while moving the robot arm holding the actual part. In this section, the part is assumed a rectangular solid and the arm is moved straight so that it touches one side of the part to the ECP specified in the former section *1. Setting the ECP*.

For details of teaching, refer to the section *Jog and Teach Command (Tools Menu)* in the chapter *The EPSON RC+ GUI*.

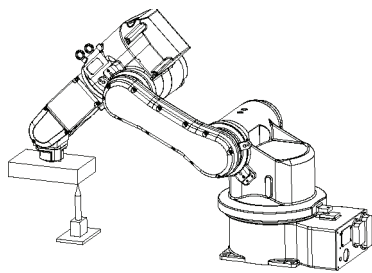
2-1 Teaching the motion start point

Move the arm to the motion start point and teach it as P1.



2-2 Teaching the motion end point

Move the arm to the motion end point and teach it as P2.



NOTE

ECP Jog Motion:

The ECP jog motion is added to available jog motions for teaching besides Joint, World, and Tool jog motions.

The ECP jog motion is based on the selected ECP coordinate systems.

3. Executing Motion

To move the arm with ECP motion, add an “ECP” parameter to a motion command.

```
Go P1           ' Moves the arm to the motion start point
Move P2 ECP     ' Executes ECP motion
```

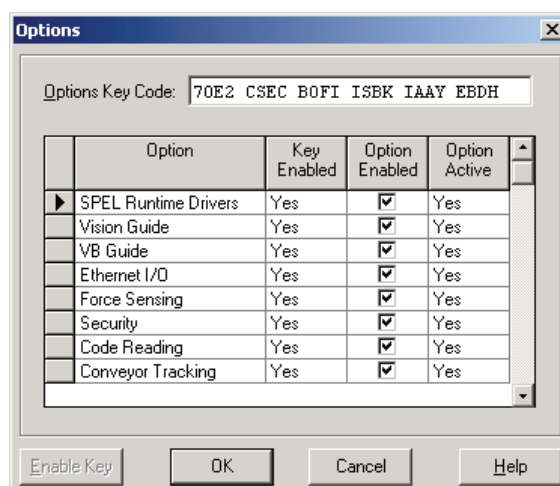
Use the Arc3 command to move the arm in an arc trajectory with the fixed tool. Use the Curve and CVMove commands to move the arm in cubic spline curves.

Installing EPSON RC+ Options

When you purchase options with your system, the options are already installed on your system. However, you can enable additional options on site.

The EPSON RC+ system uses a software key on the PC parallel port to enable options on the system.

To see what options are enabled on your system, select Setup | Options. The following dialog will be displayed, showing which options are enabled.

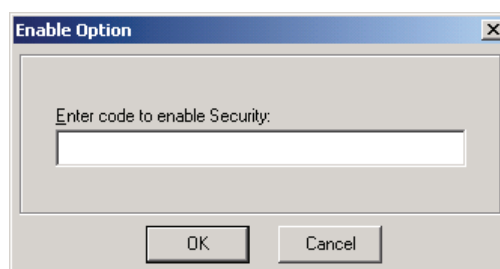


Options	Description
Key Enabled	Indicates that the software key has been enabled for the option.
Option Enabled	Allows you to temporarily disable an option.
Option Active	Indicates that the option is functioning. This field could say No if there was trouble at start up (for example, the option was not installed, or a file was missing.).

To enable an option on site

1. Write down the **Options Key Code** from the Options dialog.
2. Call your distributor to purchase the option.
3. You will receive a code to enable the option from your distributor.
4. Select the option to enable on the grid, and then click the Enable button.

5. Enter in the code you received from your distributor.



Enabling an option

When adding the following options, you must also install the software from the EPSON RC+ Installation CD.

- Vision Guide
- VB Guide
- Force Sensing
- Security
- Conveyor Tracking
- Ethernet I/O
- Fieldbus I/O
- PG Motion System

See Getting Started for instructions on running the installation. Be sure to check the option(s) you have enabled.

To temporarily disable an option

Use the Option Enabled check box to enable/disable an option. This is useful for using a software key on another system where an option's hardware is not installed.

Software License Agreement

THIS IS A CONTRACT. CAREFULLY READ ALL THE TERMS AND CONDITIONS CONTAINED IN THIS AGREEMENT. INSTALLING THE SOFTWARE (EPSON RC+) INDICATES YOUR ACCEPTANCE OF THESE TERMS AND CONDITIONS. IF YOU DO NOT AGREE TO THESE TERMS AND CONDITIONS, SIMPLY DO NOT INSTALL OR USE THE EPSON RC+ SOFTWARE.

LICENSE

SEIKO EPSON CORPORATION (the "Licensor") hereby grants to you (the "Licensee") a non-exclusive and transferable right to use the EPSON RC+ Software program and documentation (the "Licensed Materials"). You may only use this software on one (1) CPU. You may not use, copy, or modify the Licensed Materials, in whole or in part, except as expressly provided for by this agreement.

OWNERSHIP

As the Licensee you own the magnetic or other physical media on which the Licensed Materials are recorded or fixed, but the Licensor retains sole and exclusive title to and ownership of the Licensed Materials recorded on the original disk and all subsequent copies regardless of the form or media in or on which the original and other copies may exist. By paying the fee required for this license, you do not become the owner of the Licensed Materials, but are entitled to use the Licensed Materials according to the terms of this agreement. You acknowledge that the Licensed Materials are not your property and understand that giving away or selling copies of the Licensed Materials is theft.

TERM

This license is effective until terminated. You may terminate this license by destroying the Licensed Materials together with any backup copies that may have been made. This license will also terminate if you fail to comply with any term or conditions of this Agreement. You agree upon such termination to destroy the Licensed Materials together with any copies which may have been made.

BACKUP AND TRANSFER

The Licensed Materials are copyrighted and contain proprietary information and trade secrets of the Licensor. Unauthorized copying, modifying or reproducing of the Licensed Materials, even if modified, merged or included with other software, is expressly forbidden. You may make one (1) archival copy of the Licensed Materials for the sole purpose of backing up the Licensed Materials. In no event does the limited copying or reproduction permitted hereunder include the right to decompile, disassemble or electronically transfer the Licensed Materials, or translate the Licensed Materials into another language. You may sell your license rights in the software to another party. If you sell your license rights in the Licensed Materials you must at the same time transfer the documentation and the backup copy or destroy the backup copy. You cannot sell your license rights in the Licensed Materials to another party unless that party also agrees to the terms and conditions of this agreement.

PROTECTION AND SECURITY

You agree not to deliver or otherwise make available the Licensed Materials or any part thereof, including, without limitation, the object code, to any person other than the Licensor or its employees, except for purposes specifically related to your use of the Licensed Materials on one (1) CPU, without the prior written consent of the Licensor. You agree to use your best efforts and take all reasonable steps to safeguard the Licensed Materials to ensure that no unauthorized person shall have access thereto and that no unauthorized copy, publication, disclosure or distribution thereof, in whole or in part, in any form, shall be made. You recognize that the Licensed Materials contain valuable confidential information and trade secrets.

LIMITED WARRANTY

The only warranty the Licensor makes to you in connection with this license to use the Licensed Materials is that the media on which the Licensed Materials are recorded will be replaced without charge, as long as the original diskette(s) are returned to the licensor, with satisfactory proof of date of purchase, within ninety (90) days of purchase. This warranty is limited to the Licensee and is not transferable. The foregoing warranty does not extend to any Licensed Materials that have been damaged as a result of accident, misuse or abuse.

EXCEPT FOR THE LIMITED WARRANTY DESCRIBED ABOVE, THESE LICENSED MATERIALS ARE PROVIDED "AS IS". THE ENTIRE RISK AS TO THE RESULTS AND PERFORMANCE OF THE LICENSED MATERIALS IS ASSUMED BY YOU. THE LICENSOR DOES NOT WARRANT, GUARANTEE OR MAKE ANY REPRESENTATIONS REGARDING THE USE OF, OR THE RESULTS OBTAINED WITH THE LICENSED MATERIALS IN TERMS OF CORRECTNESS AND RELIABILITY OR LEGALITY. THE ABOVE IS THE ONLY WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED. INCLUDING, WITHOUT LIMITATION, LOSS OF PROFITS OR INABILITY TO USE THE LICENSED MATERIALS, EVEN IF THE LICENSOR OR SUCH OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. YOU AGREE THAT YOUR EXCLUSIVE REMEDIES, AND THE LICENSOR'S OR SUCH OTHER PARTY'S ENTIRE LIABILITY WITH RESPECT TO THE LICENSED MATERIALS SHALL BE AS SET FORTH HEREIN, AND IN NO EVENT SHALL THE LICENSOR'S OR SUCH OTHER PARTY'S LIABILITY FOR ANY DAMAGES OR LOSS TO YOU OR ANY OTHER PARTY EXCEED THE LICENSED FEE PAID FOR THE LICENSED MATERIALS.

ENTIRE AGREEMENT

This Agreement represents the entire agreement between Licensor and Licensee with respect to the License and use of the Licensed Materials.

SEVERABILITY

If any provision or a portion of this agreement is determined to be invalid under any applicable law, it shall be deemed omitted and the remaining provisions and partial provisions of this agreement shall continue in full force and effect.

NOTICE

Any notice or other communication relating to this license agreement to be sent to the Licensor must be mailed by certified mail to your distributor.

Appendix A: PC Configuration

Configure the PC Control Unit

Follow these instructions to setup the PC Control Unit for use with the SPEL Runtime Drivers and RC520 Drive Units. Normally, this is already done at the factory before shipment. However, you may also need to replace the hard disk.



You should always use the PC Control Unit that was purchased from EPSON. It was specifically configured and tested to meet UL, CE mark, performance, and safety requirements.

1. Set IRQ 5 and IRQ 10 in the PC BIOS to ISA (not PCI/ISA). This reserves IRQ 5 for the optional devices such as the OP500RC pendant and IRQ 10 for the first MIB board. For the RC420 controller, set IRQ 5, IRQ10 and IRQ11 in the PC BIOS to ISA (not PCI/ISA). This reserves IRQ11 for COM4.
2. Install Windows. During installation, format the entire hard disk. Create one NTFS partition and install in the default C:\WINDOWS directory.
3. Install the latest Windows service pack that has been tested with EPSON RC+. See the readme.txt file on the CD for information.
4. When using a network, install the LAN driver. Refer to *Installing LAN Driver* in the appendix.
5. Update the graphic driver and setup the graphic accelerator. Refer to *Updating Graphic Driver and Setting Graphic Accelerator* in the appendix.
6. Set time, date, and screen resolution. Refer to the appropriate sections in this appendix.
7. If an optional device is being used, such as the OP500RC pendant, then refer to *Installing Option Device Port COM3* in this appendix.
8. Install EPSON RC+. Refer to *Installing EPSON RC+ Software* in this appendix.
9. Reserve memory, IRQ, and port addresses for the hardware you are using by using the Windows Add New Hardware wizard in the Control Panel. See the section *Reserving Hardware Resources* later in this appendix.

Installing EPSON RC+ Software

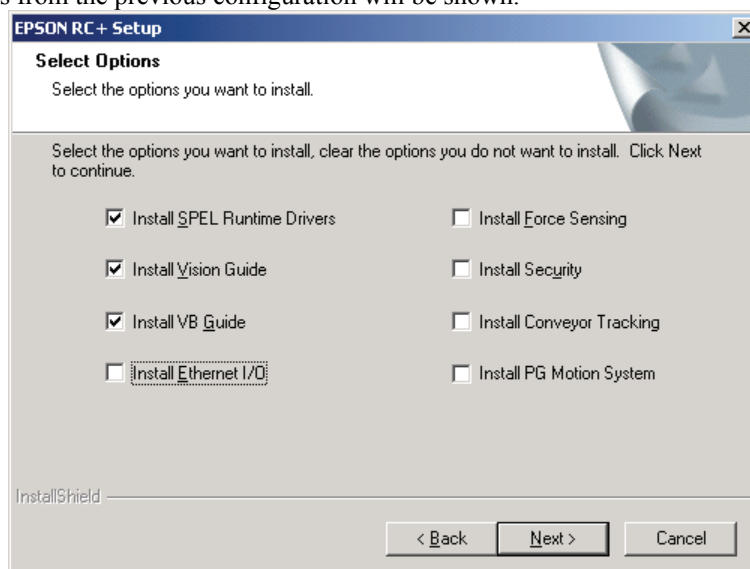
The EPSON RC+ software comes preinstalled at the factory, so software installation is not required. However, in the future you may want to install a version upgrade or an option that was not previously installed. All software for EPSON RC+ is provided on one CD that ships with the system.



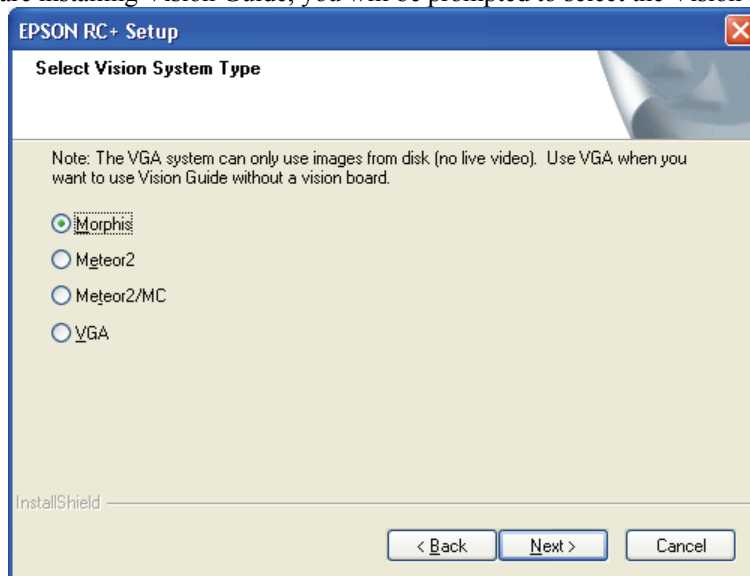
If you are upgrading from a previous version, you should execute MKVER from Tools | Maintenance to back up robot configurations (as a precaution). Then install the new version without uninstalling the previous version.

To install a new version:

1. Insert the EPSON RC+ Setup CD in the CD drive.
2. Follow the instructions provided on the screen for entering your name and company.
3. Select which drive to install EPSON RC+.
4. You will then be prompted to select which options to install. If you are upgrading, the settings from the previous configuration will be shown.



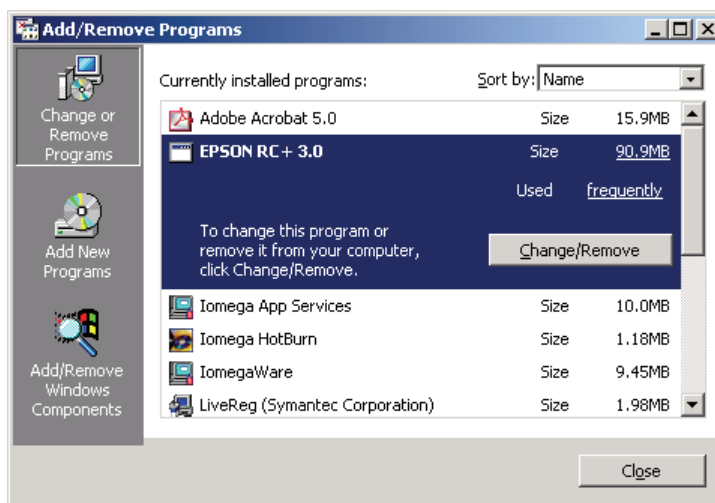
5. If you are installing Vision Guide, you will be prompted to select the Vision System type.



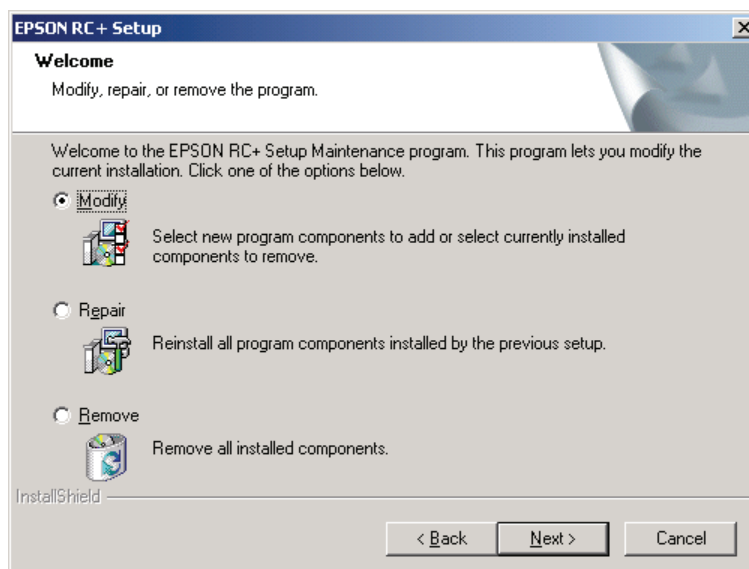
6. After files are installed, you must restart the system.

Adding / Removing Options

You can add or remove options by running Add / Remove Programs from the Windows Control Panel or by running the setup program again from the CD.



The following dialog will be displayed.



Select **Modify**, then click **Next**.

Check or uncheck the options you want to add/remove.

Follow the remaining instructions to make the modifications.

Reserving Hardware Resources

After Windows has been installed for the first time, you must reserve resources for the hardware you will be using on your system. When configured at the factory, this is already preformed.

When using ISA board (MIB or expansion I/O board) for the system, you must reserve the resource of each ISA board. The table below shows the resource names that correspond with the boards installed.

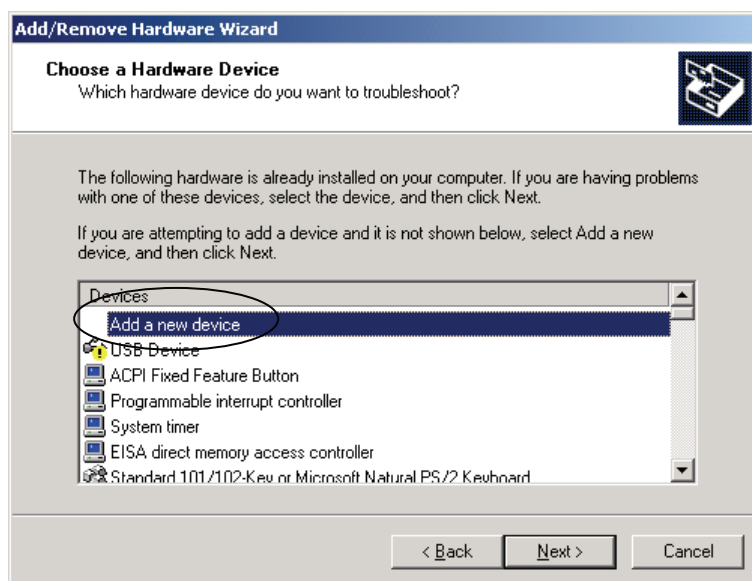
Board	Resource Name
MIB 1-3	SPEL RESOURCE MIB 1 - 3
Expansion I/O	SPEL RESOURCE IO 1 - 10



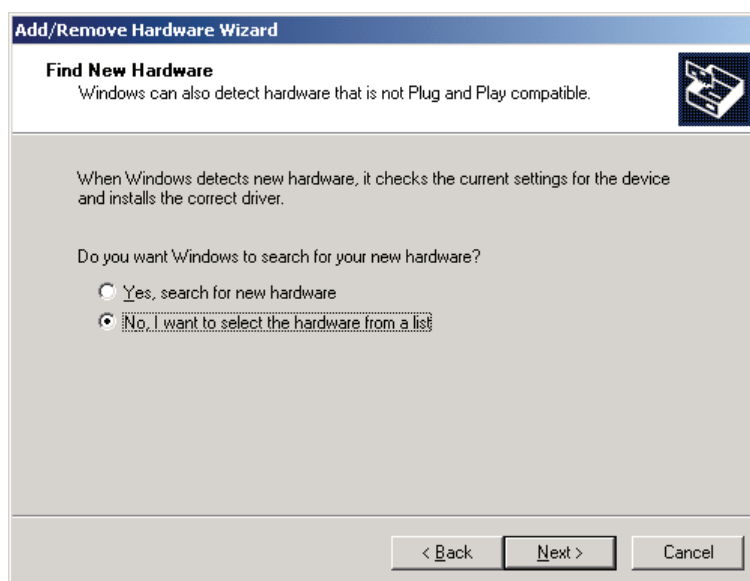
When using RC420, make sure to reserve “SPEL RESOURCE Hardware Monitor” resource in addition to the resource above.

Follow these steps to reserve resources for each ISA board used on the system:

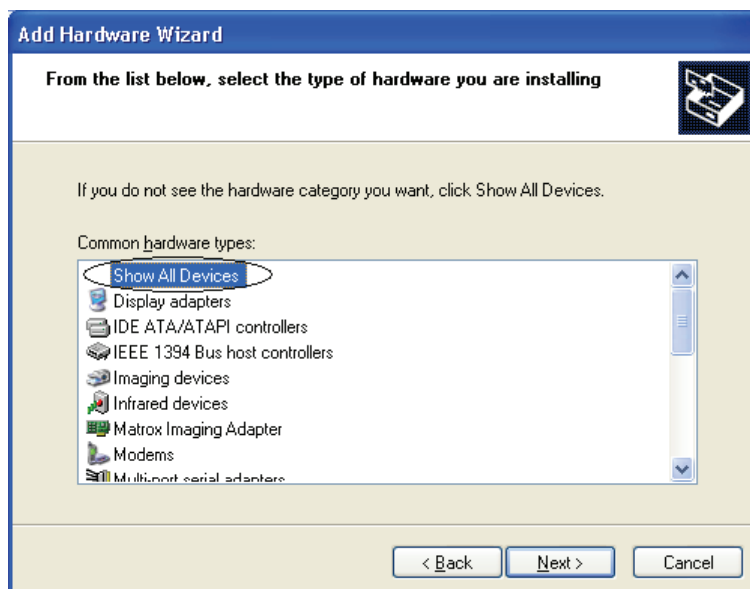
1. Open the Control Panel and select **Switch to Classic View**.
2. Double click **Add hardware**.
3. Click **Next** on Add Hardware Wizard dialog.
4. Select **Yes, I have already connected the hardware**, then click **Next**.
5. Select **Add a new hardware device**, then click **Next**.



6. Select **Install the hardware that I manually select from a list**, then click **Next**.

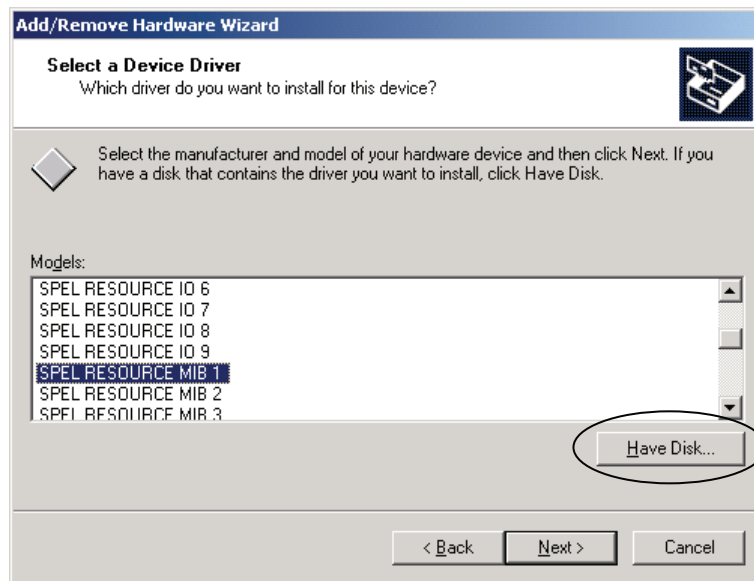


7. Select **Show All Devices**, then click **Next**.



8. Click **Have disk...** button.
9. Enter target path for the spel.inf file. Normally this is c:\EpsonRc\spel. Then click **OK**.

10. Select a hardware item from the list, e.g. **SPEL RESOURCE MIB 1**, then click **Next**.



When reserving “SPEL RESOURCE Hardware Monitor” resource for RC420, select **SPEL RESOURCE Hardware Monitor**, then click **Next**.

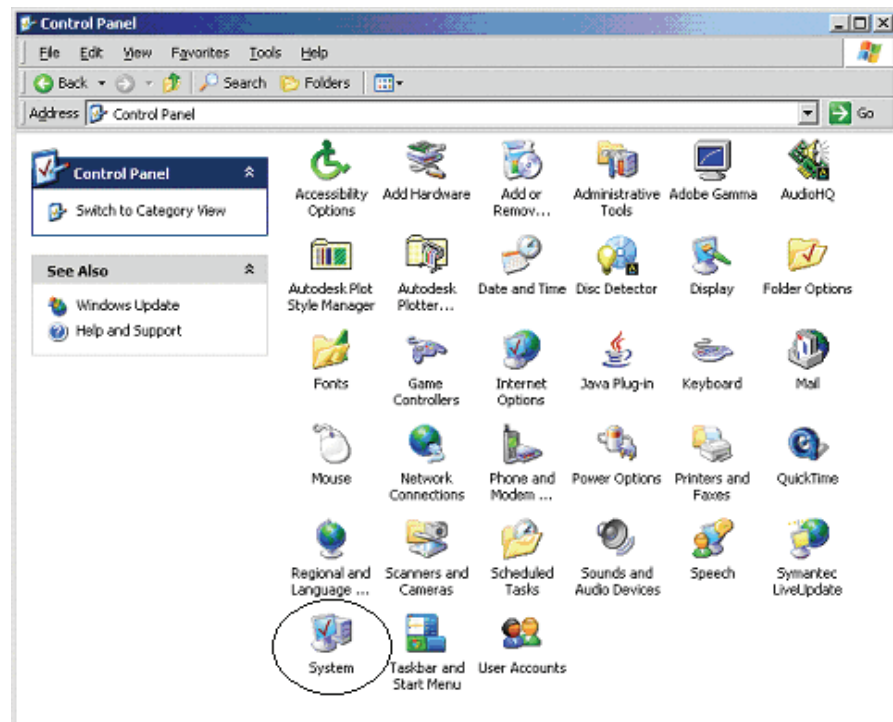
11. Click the **Next** button in the wizard dialog to finish the installation. Do not reboot at this time if you have more resources to reserve.
12. Repeat for additional hardware items.
13. When finished, reboot the system.

Turn off automatic restart after system failure

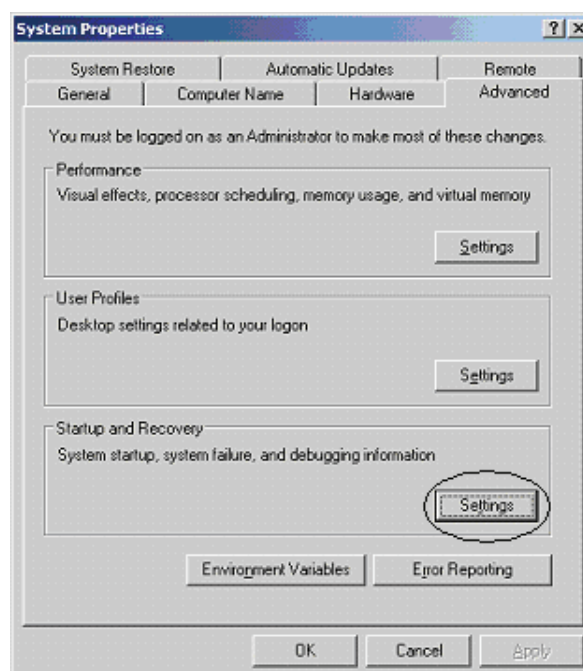
By default, Windows will automatically restart after a system fatal error has occurred. This makes it difficult to know what error has occurred if the system cannot be restarted. By disabling automatic restart, the fatal error will remain on the display until the system is manually restarted.

Perform these steps to disable automatic restart after system failure:

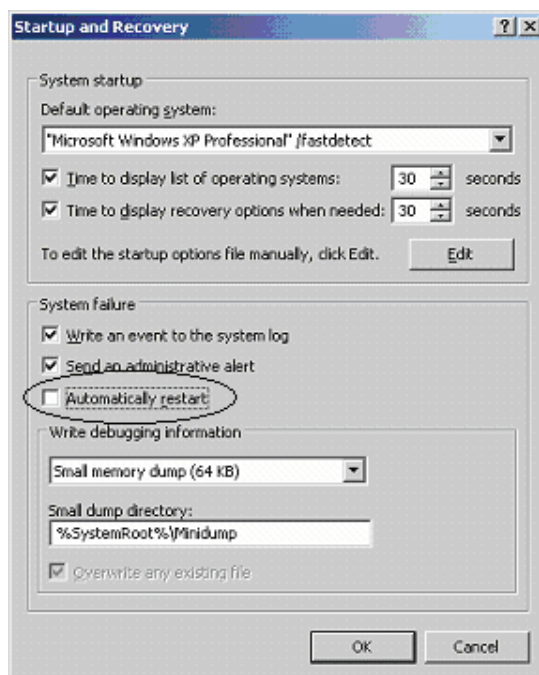
1. Open the Control Panel and select **Switch to Classic View**.
2. Double-click the **System** icon.



3. Click the **Advanced** tab, then click the **Settings...** button in the **Startup and Recovery** group.



4. Uncheck the **Automatically restart** check box in the **System failure** group.



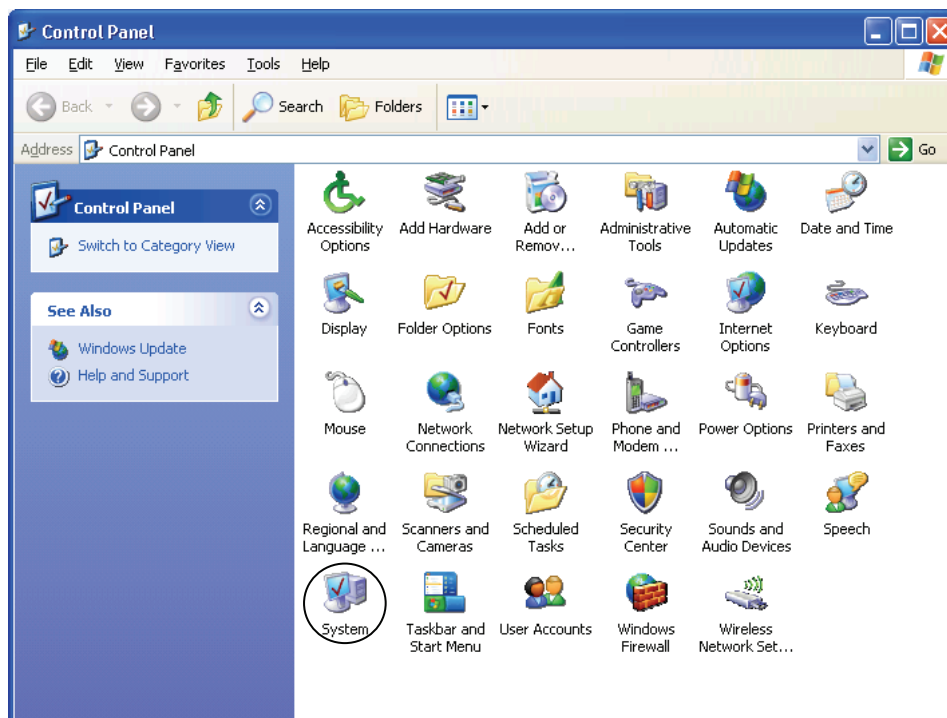
Installing LAN Driver



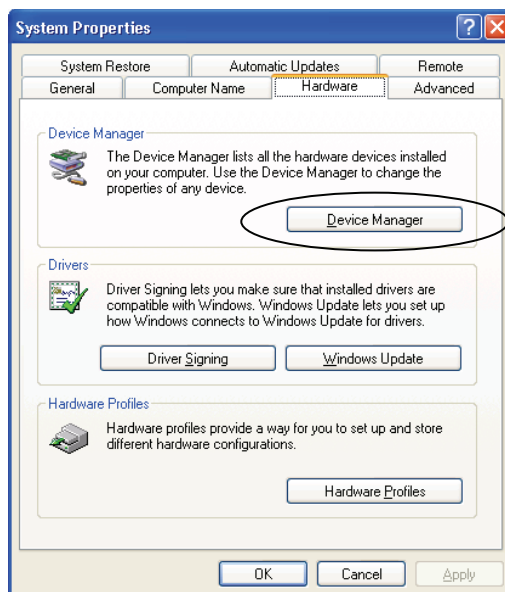
When using a network, make sure to install the LAN driver.
When using a TCP/IP communication, make sure to install the LAN driver.

Follow these instructions to install the LAN driver.

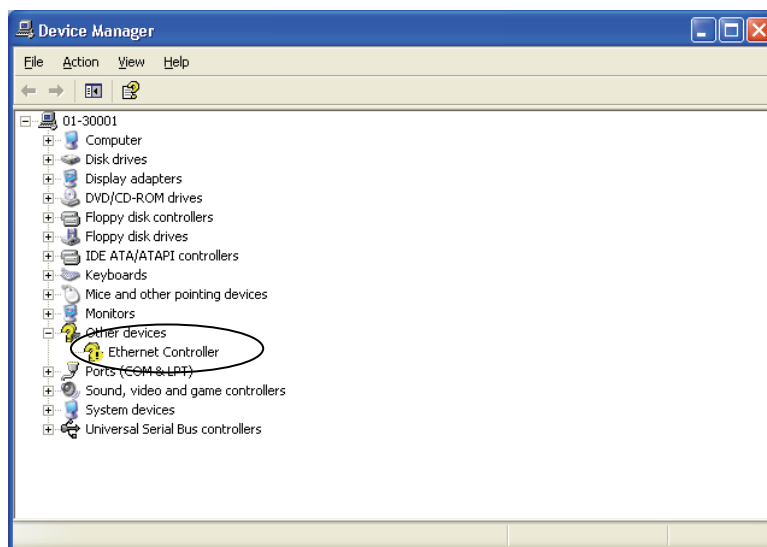
1. Insert the driver disk (DRIVERS CD-ROM Ver.1.10) to the CD-ROM drive.
2. Open the Control Panel and select **Switch to Classic View**.
3. Double-click the **System** icon.



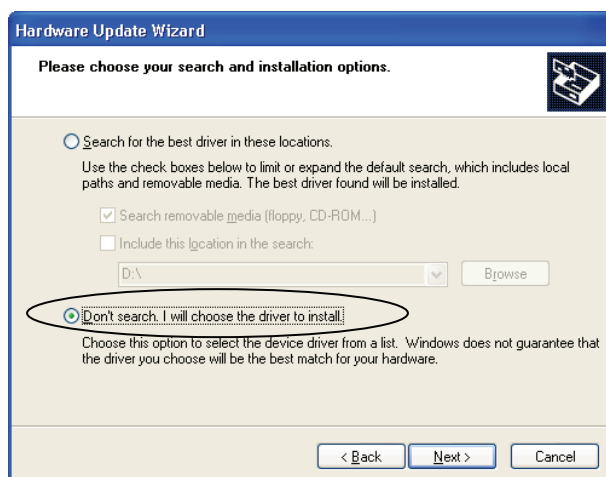
4. Select the **Hardware** tab and click **Device Manager** in the Device Manager group.



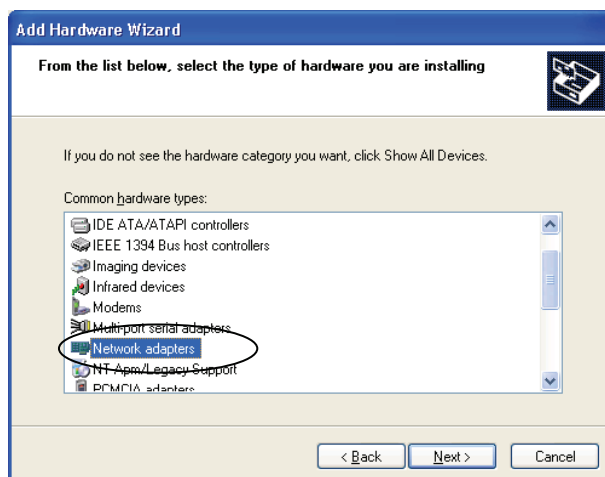
5. Right-click on the **Ethernet Controller** icon in the **Other devices**.



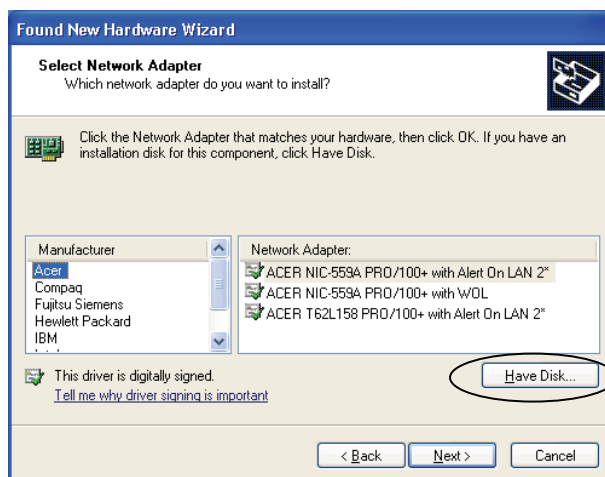
6. Click **Update Driver ...**.
7. Select **No, not this time** in the Hardware Update Wizard, then click **Next**.
8. Select **Install from a list or specific location**, then click **Next**.
9. Select **Don't search. I will choose the driver to install.**, then click **Next**.



10. Select **Network adapters**, then click **Next**.



11. Click **Have disk...** button.

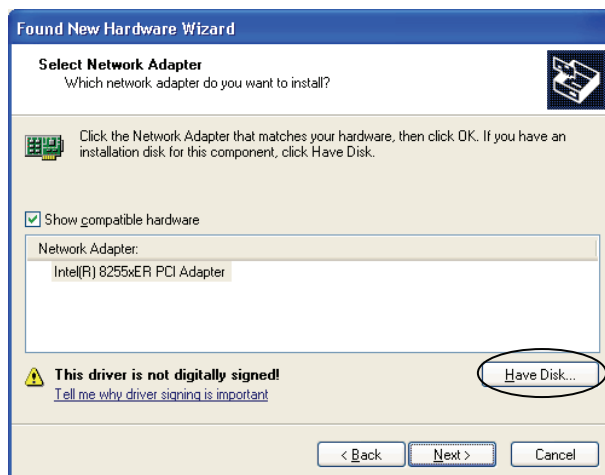


12. Enter or browse for the target path for the Net559ER.INF file located in D:\SPI_6941\LAN\82559ER\WinXP. Then click **OK**.

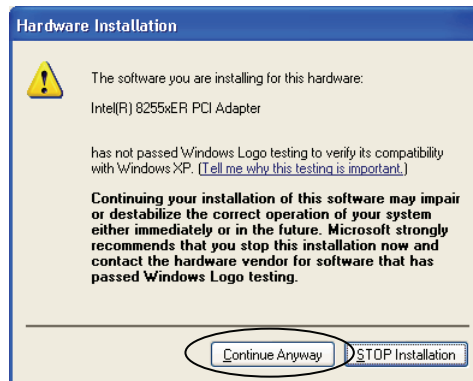


Make sure to specify the file for Windows XP.

13. Click **Next** on Found New Hardware Wizard dialog.



14. Click **Continue Anyway**.



15. Click **Finish** and finish the installation.

16. Remove the driver disk (DRIVERS CD-ROM Ver.1.10) from the CD-ROM driver.

17. Reboot the system when prompted.

Updating Graphic Driver and Setting Graphic Accelerator

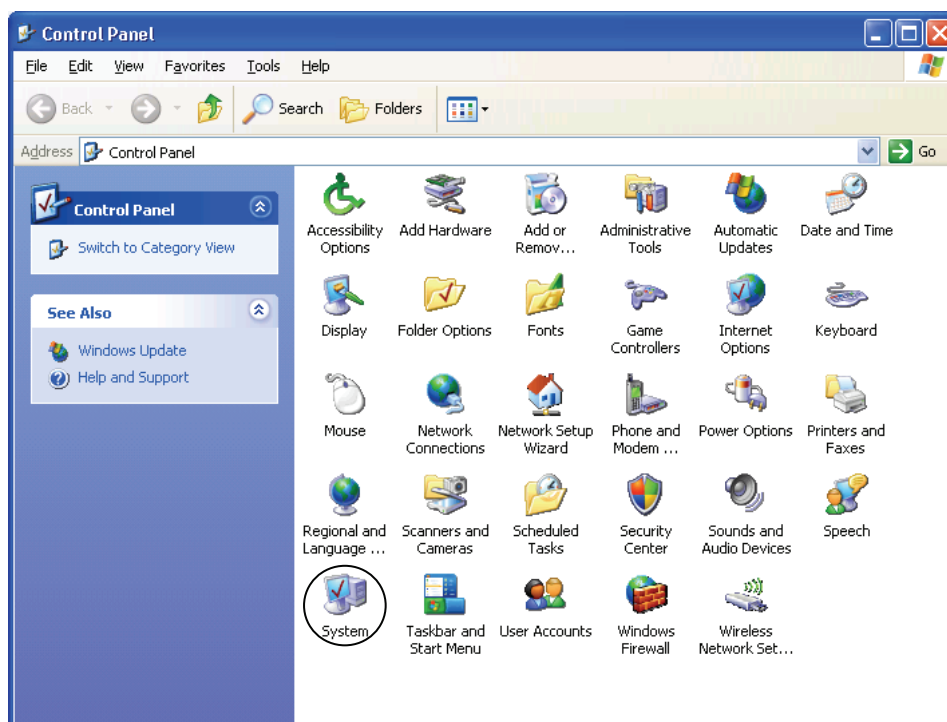
NOTE



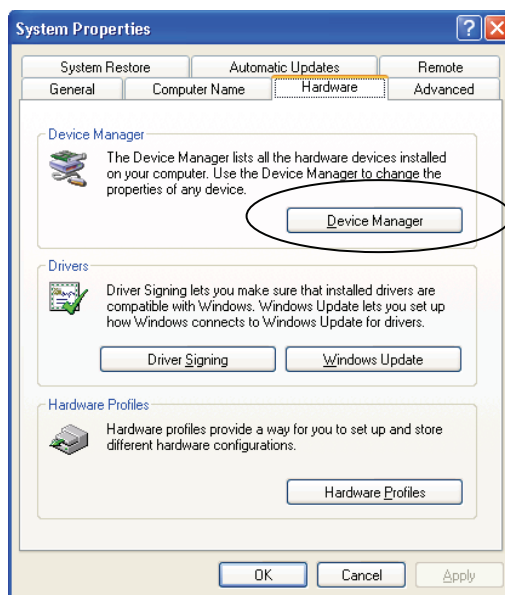
Make sure to update the graphic driver and setup the graphic accelerator. If the setup is not proper, real time command over run error may occur.

Follow these instructions to update the graphic driver.

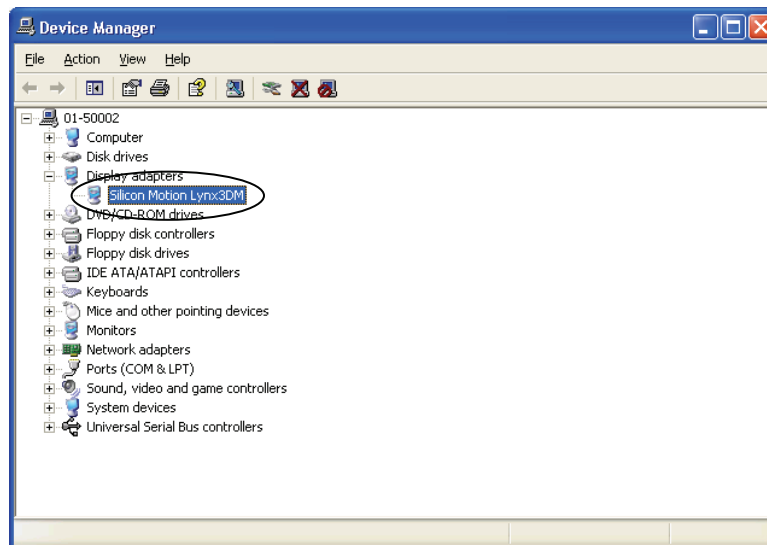
1. Insert the driver disk (DRIVERS CD-ROM Ver.1.10) to the CD-ROM drive.
2. Open the Control Panel and select **Switch to Classic View**.
3. Double-click the **System** icon.



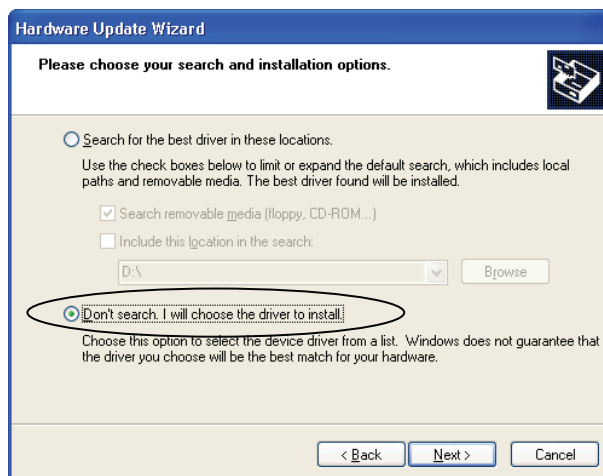
4. Select the **Hardware** tab and click **Device Manager** in the Device Manager group.



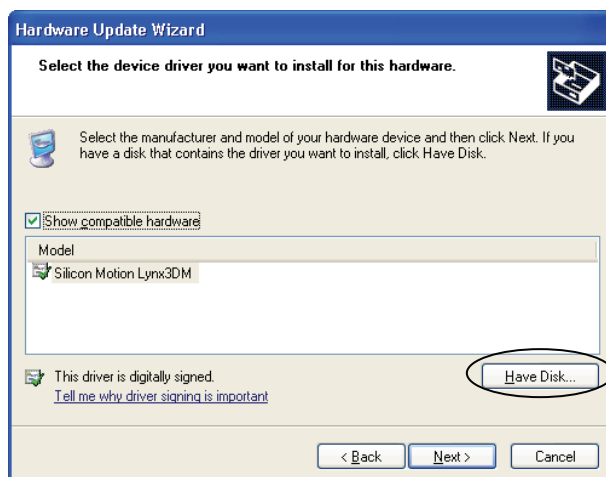
5. Right-click on the **Silicon Motion Lynx3DM** icon in the **Display adapters**.



6. Click **Update Driver ...**.
7. Select **No, not this time** in the Hardware Update Wizard, then click **Next**.
8. Select **Install from a list or specific location**, then click **Next**.
9. Select **Don't search. I will choose the driver to install.**, then click **Next**.



10. Click **Have disk...** button.



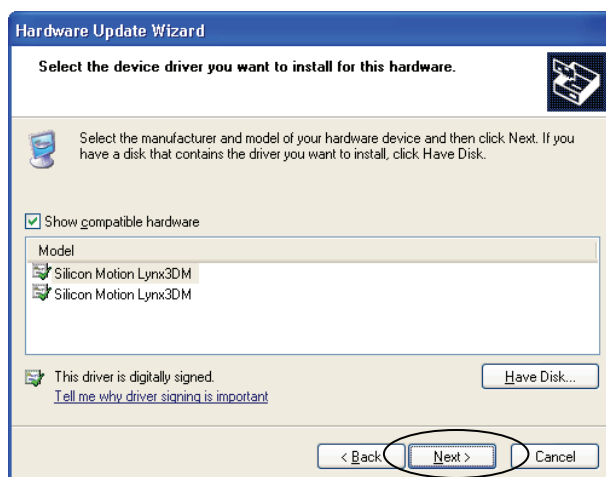
11. Enter or browse for the target path for the smisetup.inf file located in D:\SPI_6941\Video\Lynx3DM\WinXP. Then click **OK**.

NOTE



Make sure to specify the file for Windows XP.

12. Click **Next** on Hardware Update Wizard dialog.

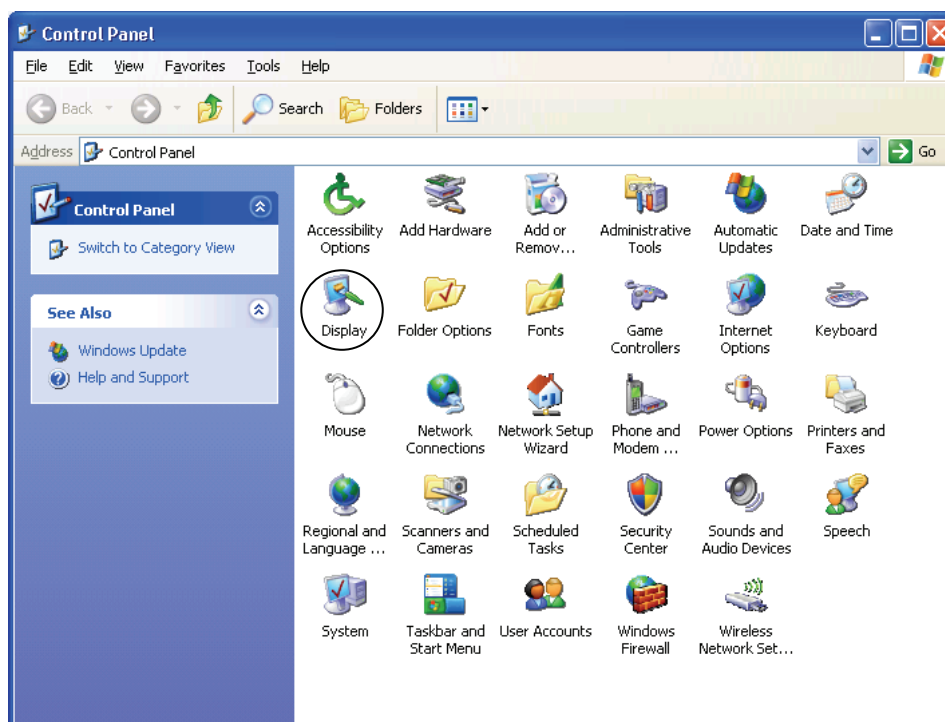


13. Click **Finish** in the wizard and finish the installation.
 14. Remove the driver disk (DRIVERS CD-ROM Ver.1.10) from the CD-ROM driver.
 15. Reboot the system when prompted.

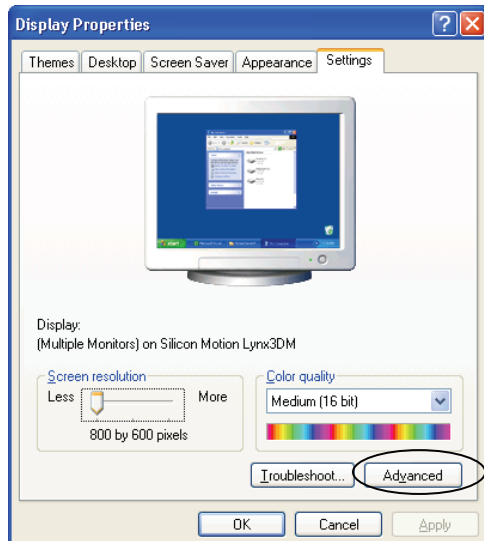
The graphic driver has been updated.

Execute the following procedure to setup the graphic accelerator.

16. Open the Control Panel.
17. Double-click the **Display** icon.

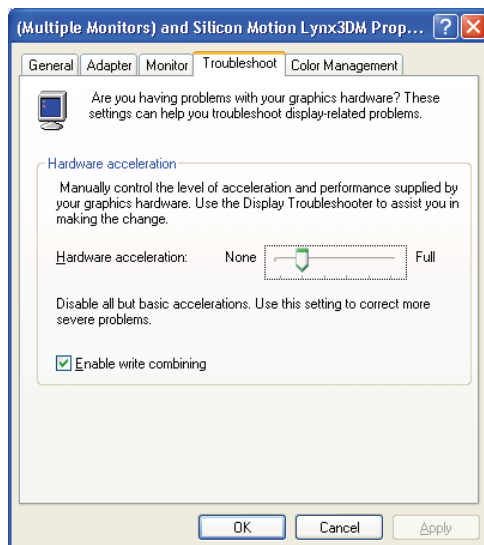


18. The **Display Properties** dialog box will appear. Select the **Settings** tab.



19. Change the resolution to [800 by 600 pixels] or higher by dragging the slider in [Screen area].
20. Change the Color to [High Color (16 bit)] in [Colors].
21. Click the **Advanced** button.
22. The **(Multiple Monitors) and Silicon Motion Lynx3DM Properties** dialog box will appear. Select the **Troubleshooting** tab.

23. Change the hardware acceleration to “Disable all but basic accelerations” by dragging the slider in [Hardware acceleration].

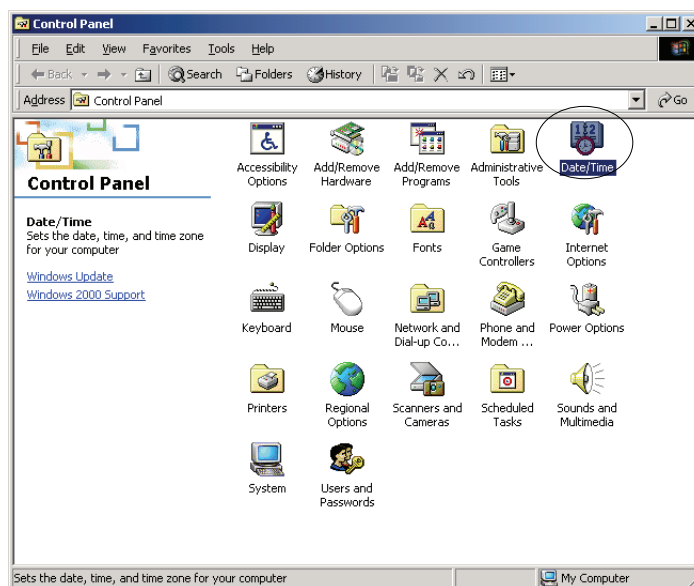


24. Click the **OK** button.
25. Click the **OK** button in the **Display Properties** dialog box to save changes. If you are asked to restart Windows, do so now.

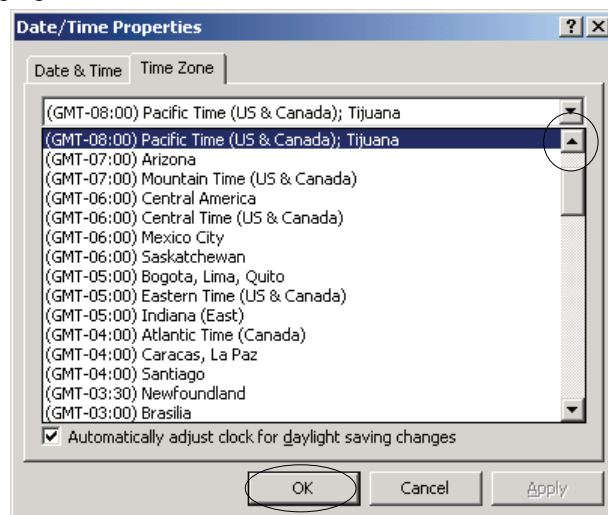
Setting Time and Date

Follow these instructions to set time and date for your system.

1. Open the Control Panel and select **Switch to Classic View**.
2. Double-click the **Date/Time** icon.

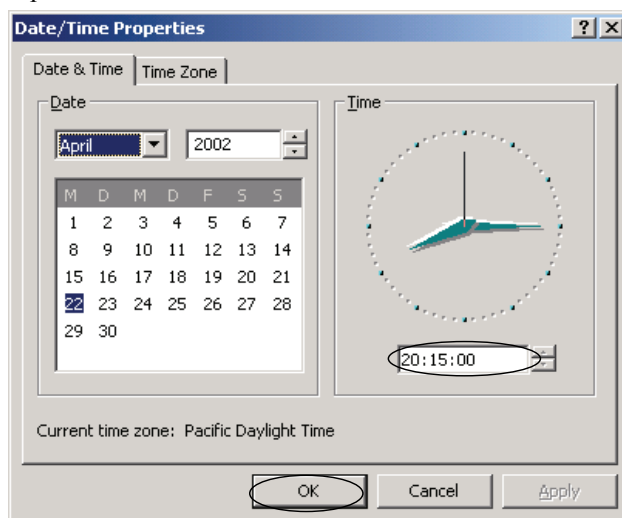


3. Click the arrow at the right of the combo box in the **Time Zone** panel. Select the appropriate time zone from the combo box.



4. Click the **Date & Time** panel.

5. Enter the month, year, and date in the **Date** group box. Enter the correct time in the **Time** group box.

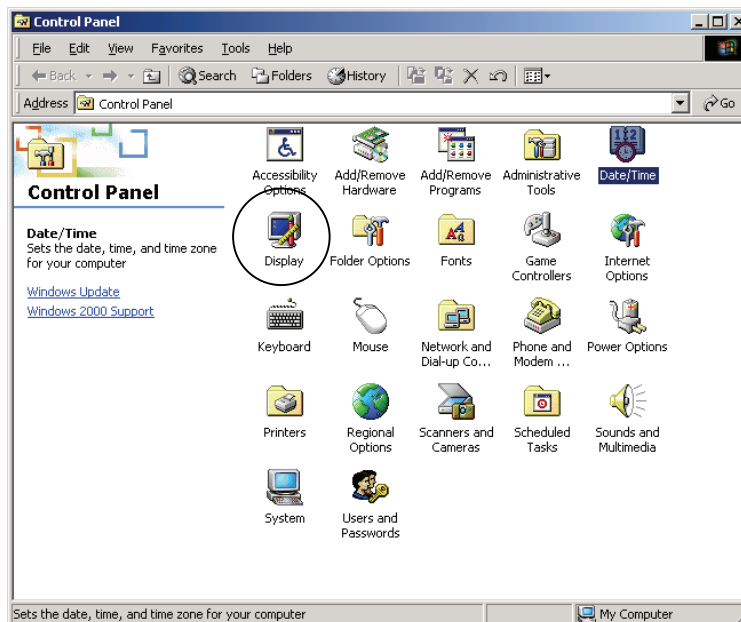


6. Click **OK** to save changes.

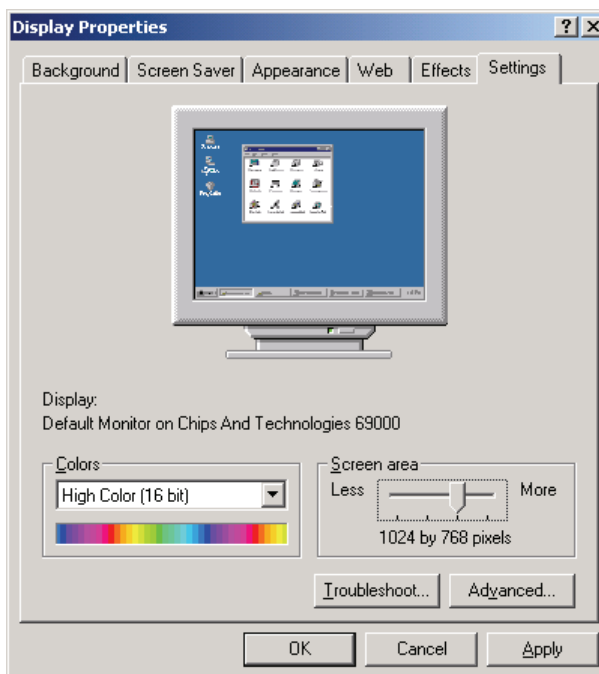
Setting Screen Resolution

Follow these instructions to set the screen resolution for your system.

1. Open the Control Panel and select Switch to Classic View.
2. Double-click the **Display** icon.

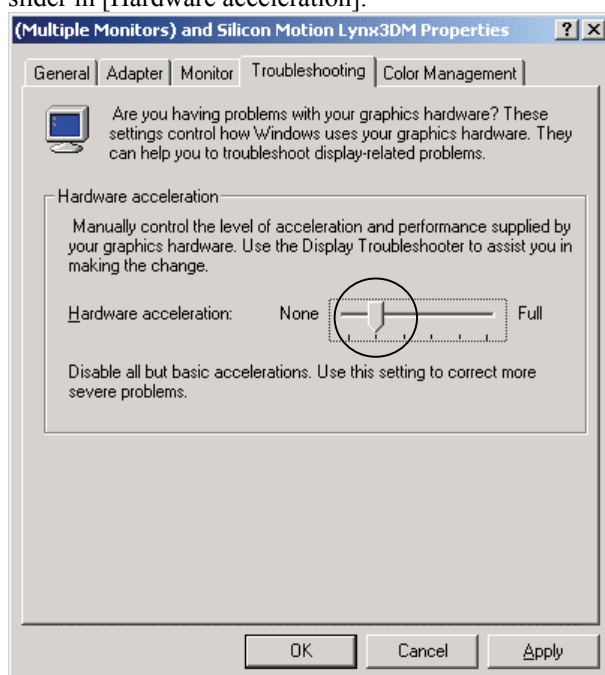


3. The **Display Properties** dialog box will appear. Select the **Settings** tab.



4. Change the resolution to [1024 by 768 pixels] by dragging the slider in [Screen area].
5. Change the Color to [High Color (16 bit)] in [Colors].
6. Click the **Advanced** button.
7. The **(Multiple Monitors) and Silicon Motion Lynx3DM Properties** dialog box will appear. Select the **Troubleshooting** tab.

8. Change the hardware acceleration to “Disable all but basic accelerations” by dragging the slider in [Hardware acceleration].



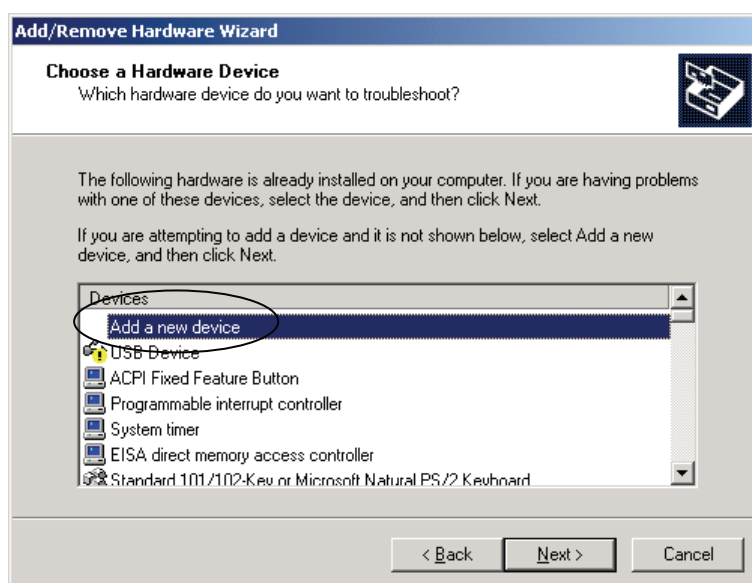
9. Click the **OK** button.
10. Click the **OK** button in the **Display Properties** dialog box to save changes. If you are asked to restart Windows, do so now.

Installing Optional Device Port (COM3)

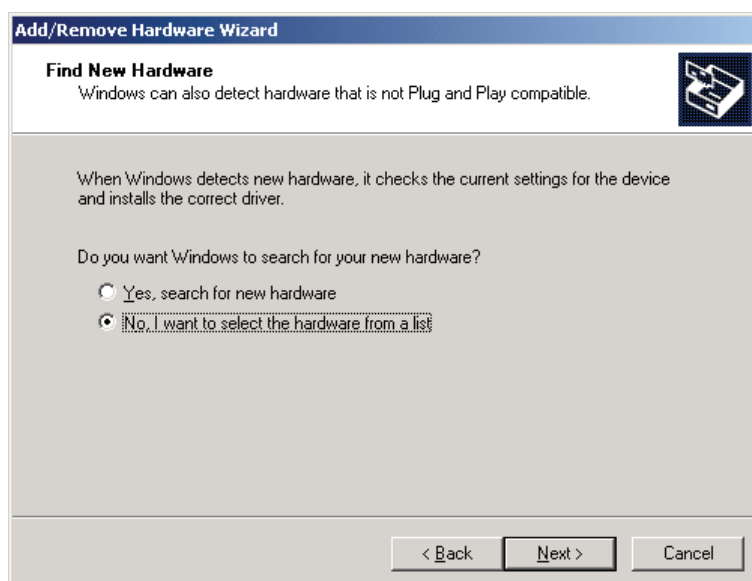
When an optional device is being used, such as the OP500RC, you must configure COM3 to allow communication with the device.

Follow these steps:

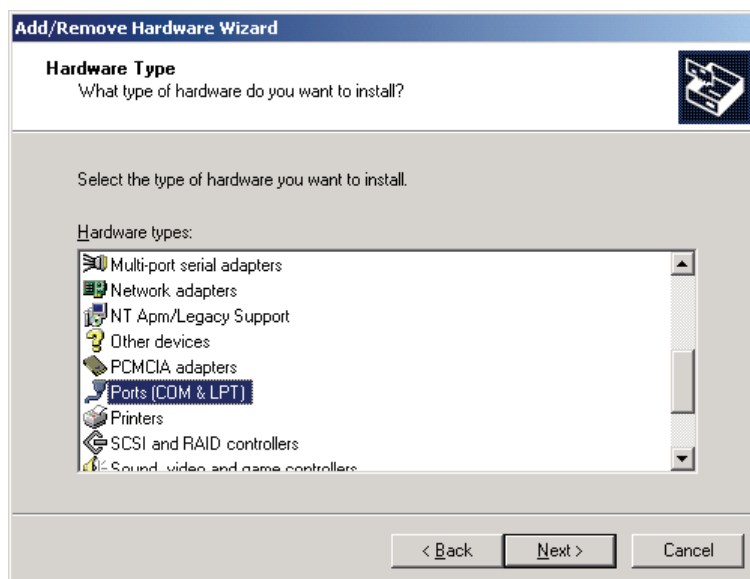
1. Ensure that IRQ5 is set to Legacy in the BIOS PCI configuration.
2. Start Windows.
3. Open the Control Panel and select Switch to Classic View.
4. Double-click the **Add Hardware** icon.
5. Click **Next** on Add Hardware Wizard dialog.
6. Select **Yes, I have already connected the hardware**, then click **Next**.
7. Select **Add a new hardware device**, then click **Next**.



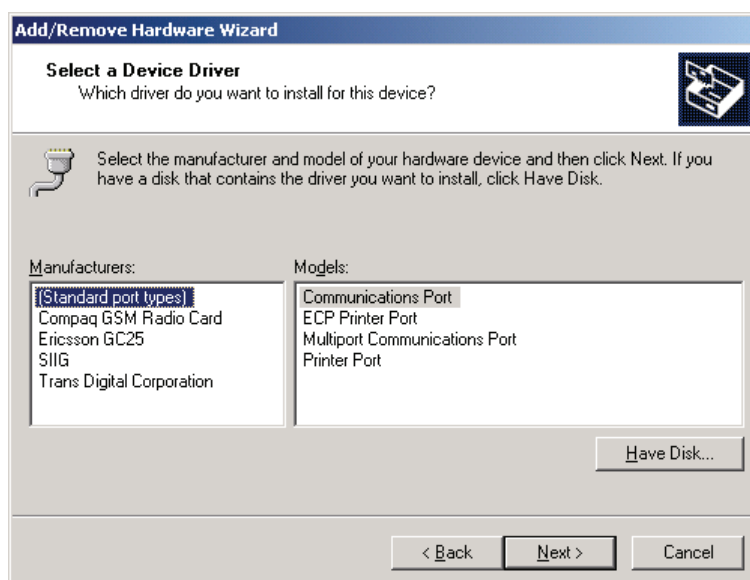
8. Select **Install the hardware that I manually select from a list**, then click **Next**.



9. Select **Ports (COM & LPT)** then click **Next**.

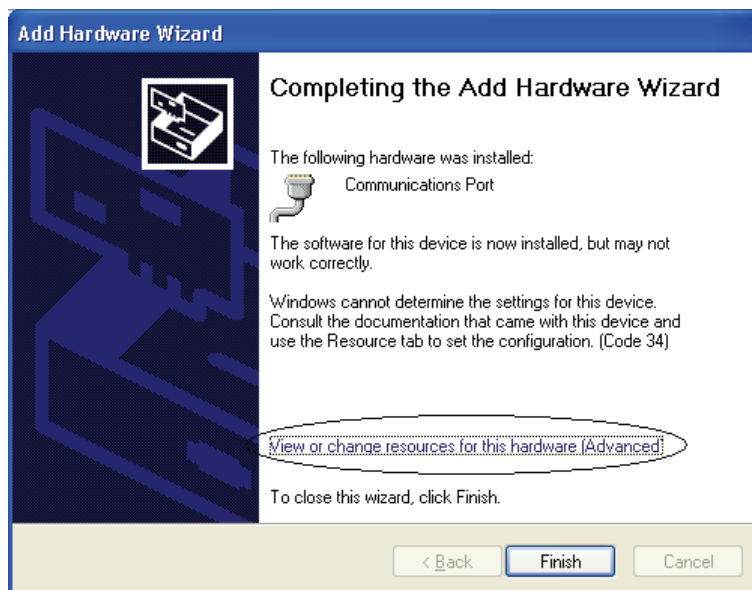


10. Select **Communications Port**, then click **Next**.

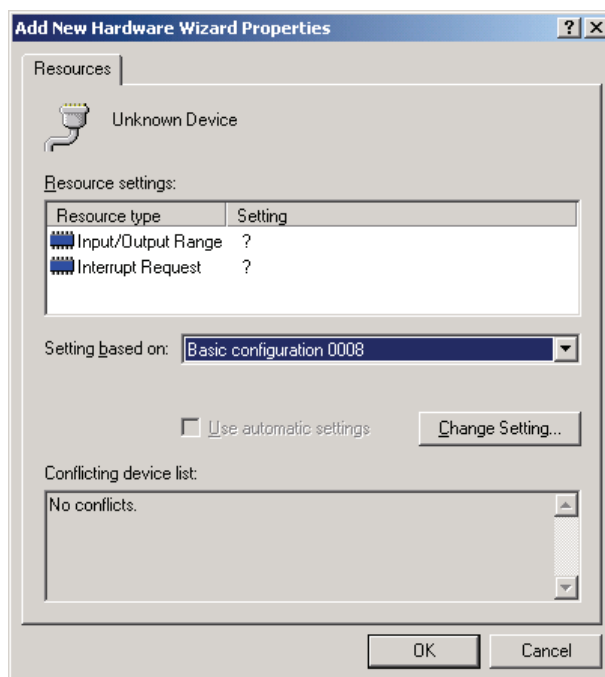


11. Click **Next** in the window.

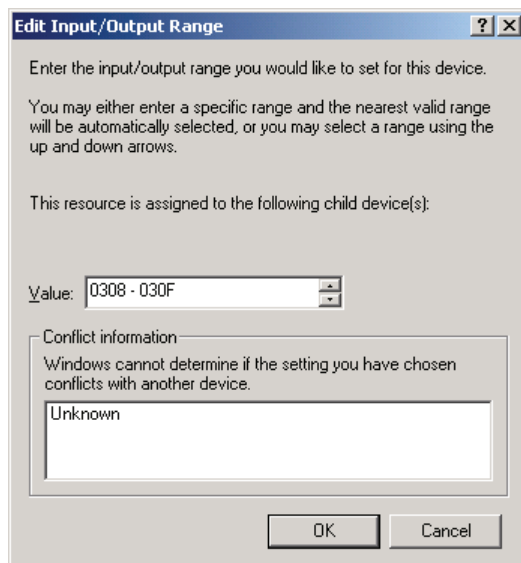
12. Click **View or change resources for this hardware** in the Add Hardware Wizard.



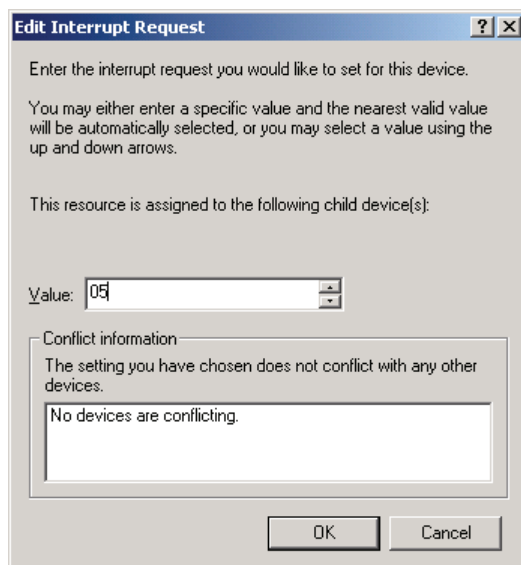
13. Click the **Set Configuration Manually** button.
14. Select **Basic configuration 0008** from **Setting based on**.



15. Click on **Input/Output Range** and then click **Change Setting**.
16. Enter **0308-030F** for the value and click **OK**



17. Select **Interrupt Request** and then click **Change Setting**.
18. Enter **05** for the value and click **OK**.



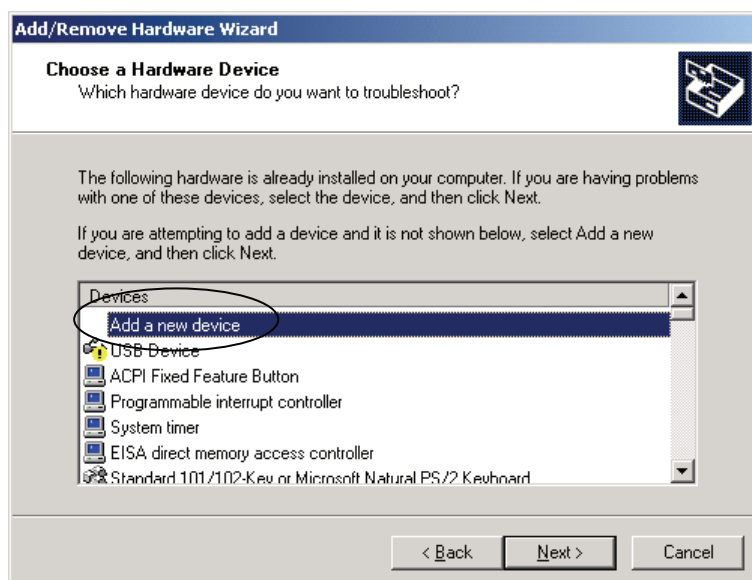
19. Click the **OK** button.
20. Click **Finish** in the wizard and finish the installation.
21. Reboot the system when prompted to do so.

Installing RC420 UPS Drivers

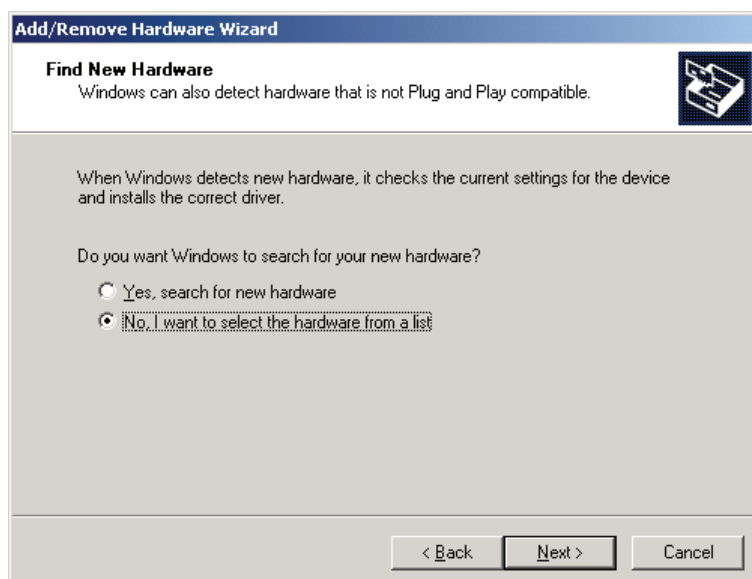
When the optional UPS is installed in the RC420 controller, you must configure COM4 to allow communication with the device and also install the UPS driver.

Follow these steps:

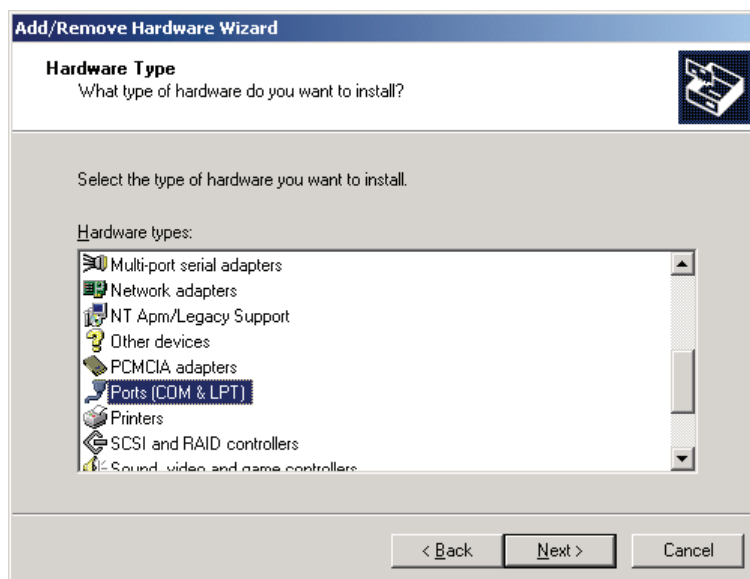
1. Restart the PC and enter the BIOS.
2. In the PCI configuration, set IRQ11 to Legacy. This will reserve IRQ11 for COM4.
3. Reboot the PC and start Windows.
4. If you are using Windows XP, open the Control Panel and select Switch to Classic View.
5. Double-click the **Add Hardware** icon.
6. Click **Next** on Add Hardware Wizard dialog.
7. Select **Yes, I have already connected the hardware**, then click **Next**.
8. Select **Add a new hardware device**, then click **Next**.



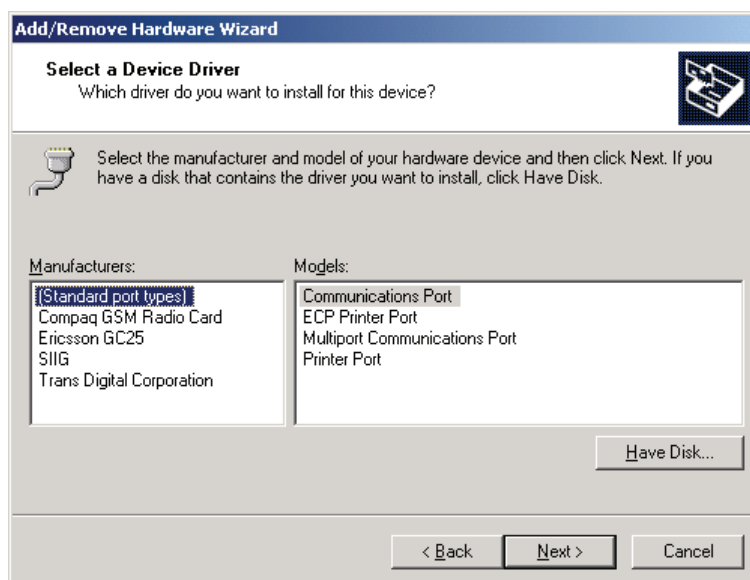
9. Select **Install the hardware that I manually select from a list**, then click **Next**.



10. Select **Ports (COM & LPT)** then click **Next**.

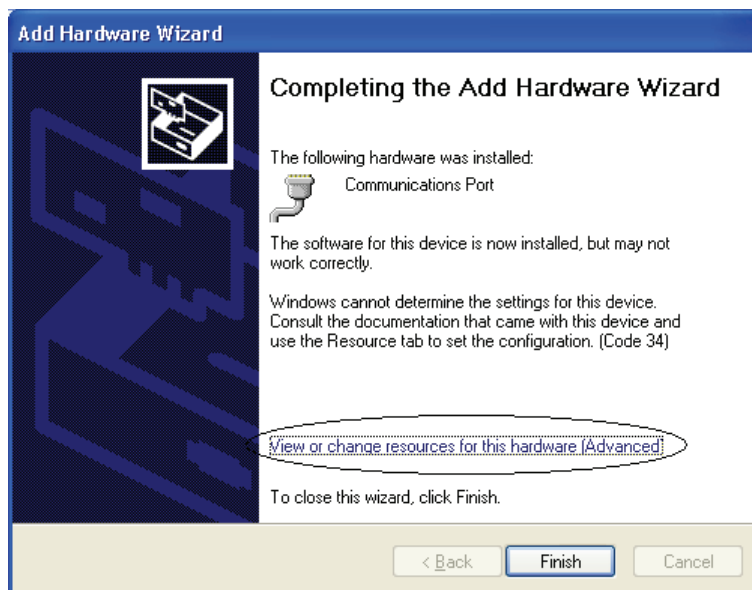


11. Select **Communications Port**, then click **Next**.

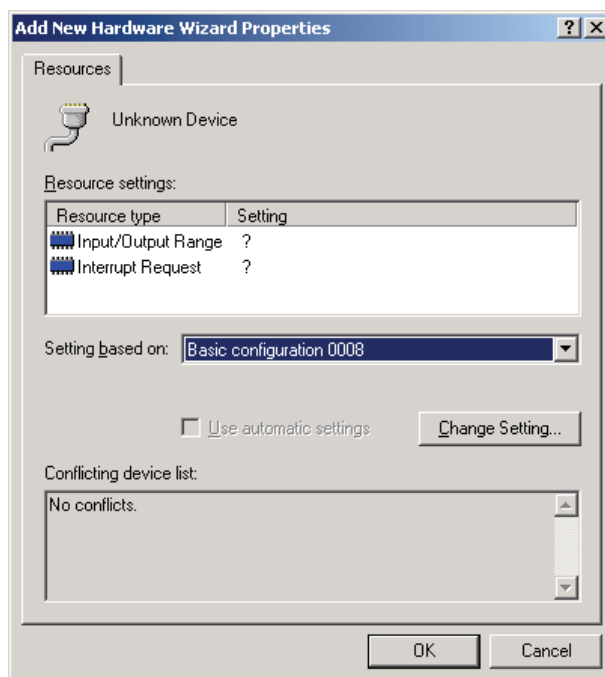


12. Click **Next** in the window.

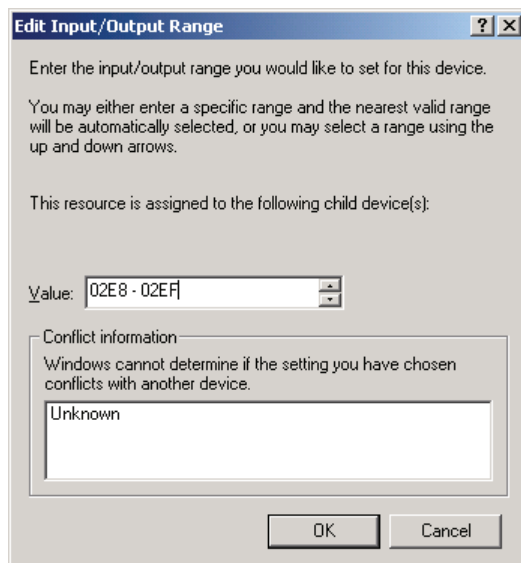
13. Click **View or change resources for this hardware** in the Add Hardware Wizard.



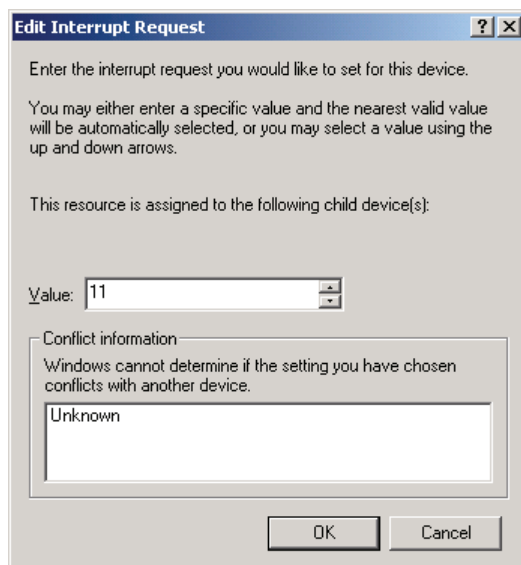
14. Click the **Set Configuration Manually** button.
15. Select **Basic configuration 0008** from **Setting based on**.



16. Click on **Input/Output Range** and then click **Change Setting**.
17. Enter **02E8-02EF** for the value and click **OK**

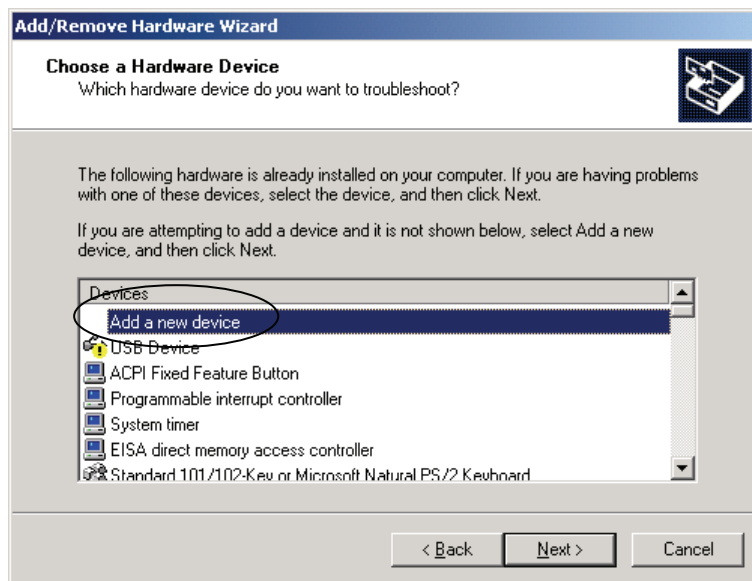


18. Select **Interrupt Request** and then click **Change Setting**.
19. Enter **11** for the value and click **OK**.

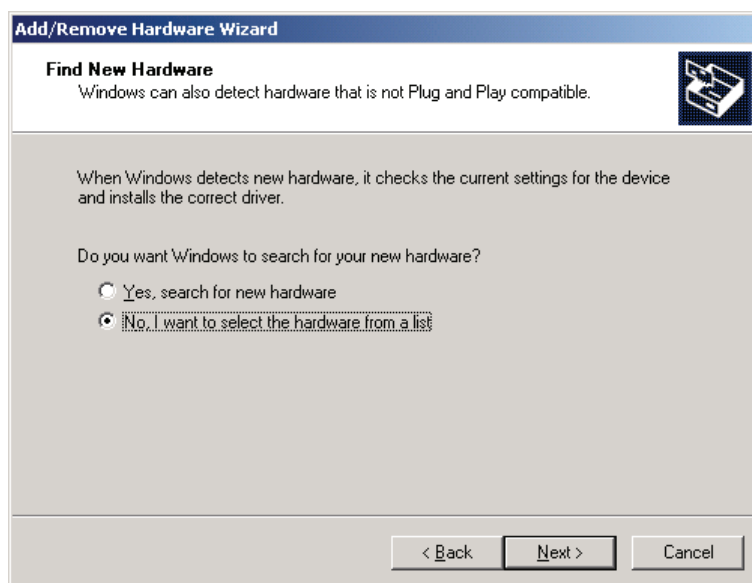


20. Click the **OK** button.
21. Click the **Finish** button in the wizard and finish the installation.
22. Reboot the system when prompted to do so.
23. Open Control Panel.
24. Double-click **Add Hardware** icon.
25. Click **Next** on Add Hardware Wizard dialog.

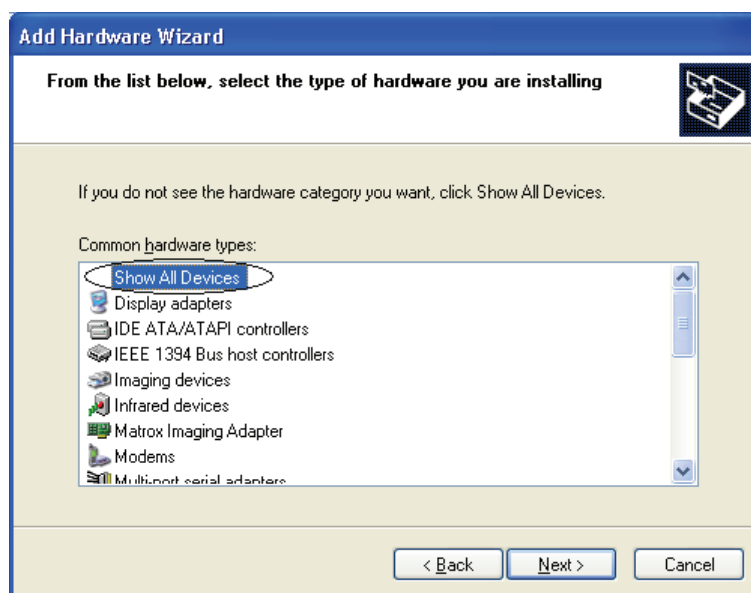
26. Select **Yes, I have already connected the hardware**, then click **Next**.
27. Select **Add a new hardware device**, then click **Next**.



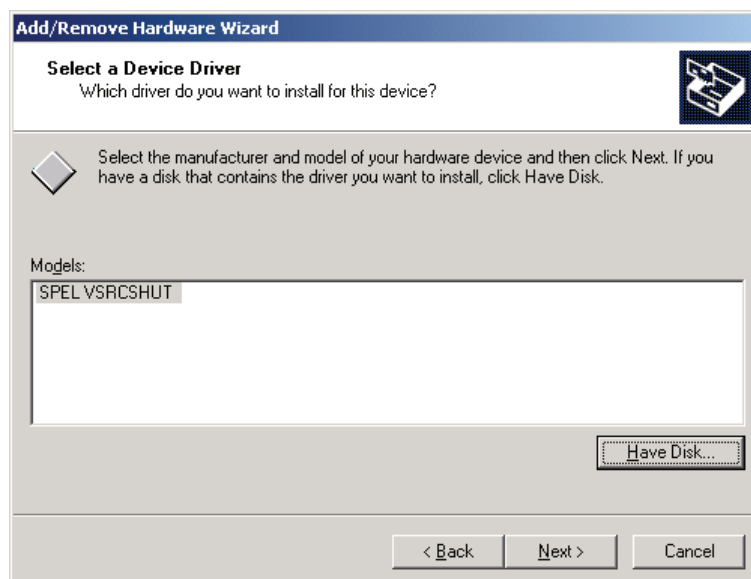
28. Select **Install the hardware that I manually select from a list**, then click **Next**.



29. Select **Show All Devices**, then click **Next**.



30. Click **Have disk...** button.
31. Enter or browse for the target path for the VSRCShut.inf file located in c:\EpsonRc\RC420\Drivers. Then click **Next**.



32. Click **Next** in the wizard and finish the installation.
33. Reboot the system when prompted.

Adding and Removing Memory



NOTE If you do not use memory supplied by EPSON, a controller malfunction may occur. When you add memory, use memory offered by EPSON for your controller model.

This section contains information on how to add or remove memory for RC520 and RC420 controllers using EPSON RC+.

Memory type: PC-100 SDRAM

Maximum memory: 512 Megs

DIMM size: 128 Meg or 256 Meg

Follow these steps to add or remove memory:

1. If Vision Guide is installed, you must first uninstall it. This is because the vision drivers must reserve non-paged memory for the frame grabber during installation with the new memory size. From the Windows Control Panel, open Add / Remove Programs and launch the EPSON RC+ Installer. From the Modify / Repair / Remove dialog, select Modify, then deselect the Vision Guide check box. Complete the installation wizard.
2. Shutdown the controller and disconnect from power.
3. Remove the top cover of the controller and add or remove the memory DIMMs located on the CPU board. Ground yourself before handling the DIMMs.
4. Install the top cover of the controller and connect to power.
5. Restart the controller.
6. If you are using Vision Guide, you must reinstall the Vision Guide option. From the Windows Control Panel, open Add / Remove Programs and launch the EPSON RC+ Installer. From the Modify / Repair / Remove dialog, select Modify, then select Vision Guide. Complete the installation wizard.

Precaution for Epson ProSix PS Series robots

After EPSON RC+ is installed in the robot system with EPSON ProSix PS series robot, the error message “error starting SPEL drivers” appears during EPSON RC+ startup because the Drive Unit Type of RC520 motion system is set to “RC520DU” by default

Follow the steps below and change the Drive Unit Type from “RC520DU” to “RC520DU6” to clear the error.

1. Click the **No** button in the error message.
2. Select the **System Configuration** from the **Setup** menu.
3. Click the **Motion Systems** tab.
4. Click on the RC520 Motion System item in the list box.
5. Click the **Configure** button. The RC520 Motion System dialog will be displayed.
6. Click the Drive Unit Type field, then click the arrow to display the selections. Select RC520DU6.
7. Click the **OK** button.
8. Restore the backed-up system configuration using the SETVER command.

Settings at Shipment

Some of the settings of RC520 and RC420 at shipment from the factory differ from Windows default at reinstallation.

Item	Factory Shipment	Windows default
User Name	EPSON RC User	Optional
Password	epson	N/A
Windows Error Report	Invalid	Valid
Windows automatic updating	Invalid	Auto
Remote Assistance	Invalid	Valid
Screensaver	Invalid	Valid
Synchronization with Internet time server	Invalid	Valid
Windows Halt	Invalid	Valid
Firewall	Invalid	Valid
Security Warning	Do not display	Display



User Name and Password at shipment

The RC520 and RC420 robot controllers are configured at factory as shown below:

User Name: EPSON RC User (with Administrator authority)

Password: epson

When the computer name, user name, or password is changed in RC420, the UPS service will not automatically shut down the robot controller in the case of a power failure despite its setting. After you change the computer name, user name, or password, configure the UPS service again so that it will automatically shut down the robot controller in the case of a power failure. For details, refer to the section *Setup & Operation 10.6.3 Setting up the Automatic Shutdown at the Time of a Power Failure* in the RC420 robot controller manual.



When the Windows firewall is activated, you cannot use TCP/IP communications. Make sure to disable the firewall. Firewall is disabled at shipment.

Appendix B: Changes from previous SPEL versions

This appendix describes the changes from previous versions of SPEL.

Changes from SPEL to SPEL⁺

Point files are no longer associated with program groups

Point files are separate from groups. In SPEL, there was a point file for each group if separate point files was enabled. When a group was started, its point file was automatically loaded.

Now point files are not associated with groups. If you convert an old project using SPEL that also used separate point files, you will get a warning message and all point files will be removed from the project (but remain in the project directory). You must use Project | Edit and add the point files. Then you must add the LoadPoints command to your code to load the correct points.

Program Groups are now optional

Program groups are no longer required, but are still supported. They are also useful for remote control. A remote device can select which group to run.

Longer variable, label and function names

Function and variable identifiers can now be 32 characters long instead of 8. I/O Labels and point labels can be 16 characters long.

Functions can now have parameters

You can now pass parameters to functions by value or by reference. You can also pass arrays by reference.

Functions can now return values

User functions can return values.

Call keyword can be omitted

You can implicitly call a user function or DLL function by using the name of the function and omitting the argument parentheses. For example:

```
Call MyFunc(1, 2)  
MyFunc 1, 2
```

Identifier case is formatted according to declaration

When you type a statement containing a variable name, function name, or label, the case is automatically changed to the same case as the name was declared. If you change the declaration case, then the new case is updated throughout the project.

Faster execution

Programs execute significantly faster.

Variable scope improvements

Now all variables declared in a function are local to that function.

Module level variables are supported.

The Variable Library is no longer needed.

Entry variables are now called Globals, which can be accessed by any program in any group.

Project Subdirectories are supported

You can now create project subdirectories to help organize projects.

Extern statements no longer required

Now all Global variables and all functions are available in all programs without the need for Extern statements.

All variables are initialized automatically

Numeric variables are initialized to 0 and string variables are set to empty when programs and functions start.

More memory available for programs and variables

Strings can now be 255 bytes long

String size has been increased from 80 to 255 characters.

Robot point labels are now unique per file

Now each point file has its own set of labels for each point.

Any number of point files can be added to the project for each robot

Each robot in the project can have one or more point files.

Point files are no longer associated with program groups

Point files are now separate from program groups.

New Boolean data type

A new data type has been added to support Boolean (True / False) variables.

Multiple robots can be controlled in the same project

Several robots can be controlled in one project.

More tasks

The number of tasks has been increased from 16 to 32.

Auto synchronization for Vision Guide commands

Vision Guide commands can now be executed from any task without using ZeroFlg or other synchronization. When two or more tasks execute a vision command at the same time, the commands are executed on a first come, first serve basis. One task's command will execute while the other tasks will block (wait for access to vision system) until they can execute their commands.

TCP/IP Communications are supported

Controllers can communicate with one another using TCP/IP.

Improved Error Handling

Error handlers are now local for the function, not the task. Also, you can resume the same statement or the next statement.

Arrays can be redimensioned at runtime

Arrays can be dynamically redimensioned at runtime.

More RS-232 ports are supported

Additional RS-232 ports can be added up to 16 maximum.

Spaces are no longer added to numbers in Print statement

When numbers are displayed using a Print statement, spaces are no longer added before each number.

Syntax Help and Variable Help Display

Syntax help is now displayed as you type. Variable values are displayed when the mouse cursor is positioned over a variable name at run time when a task is halted.

User DLL functions can be called from SPEL

You can declare and call DLL functions from your SPEL programs. This allows you to extend the language by adding your own external functions to support custom interfaces. You use the Declare statement to declare DLL functions.

Multiple files can be opened

Now you can select more than one file when opening files for editing.

New Commands in SPEL⁺

Command	Description
Accel Function	Returns acceleration and deceleration settings for point to point motion.
Accels Function	Returns acceleration and deceleration settings for linear motion.
Arch Function	Returns arch settings.
BClr Function	Clear one bit and return modified number.
Boolean Statement	Declares Boolean variables.
BSet Function	Set one bit and return modified number.
BTst Function	Tests one bit of a number and returns True or False.
ChDrive Statement	Changes the current drive.
ChkCom Function	Returns status of RS-232 port.
ChkNet Function	Returns status of TCP/IP port.
CurDir\$ Function	Returns current directory.
CurDrive\$ Function	Returns current drive.
CurPos Function	Returns current position while moving.
CU Statement	Sets U coordinate of a point.
CX Statement	Sets X coordinate of a point.
CY Statement	Sets Y coordinate of a point.
CZ Statement	Sets Z coordinate of a point.
Declare Statement	Declare a DLL function.
DegToRad Function	Converts degrees to radians.
Dist	Returns the distance between two points.
Do...Loop	Do...Loop construct.
EResume Statement	Resume from error handler.
Error Statement	Generates an error.
EStopOn Function	Returns EStop status.
Eval Function	Executes a monitor statement and returns error status.
Exit Statement	Exit a loop construct.
FileDateTime\$ Function	Returns the date and time of a file.
FileExists Function	Checks if a file exists.
FileLen Function	Returns the length of a file.
Find Statement	Specifies condition for storing a position during motion.
FindPos Function	A point returned from a successful Find.
FolderExists Function	Checks if a folder exists.
Here Function	Returns the current robot position point (same as P*).

Here Statement	Teaches a point at the current position.
InStr Function	Search for a substring within a string.
InW Function	Returns 16 bits of inputs.
JA Function	Return a point defined with joint angles.
JTran Statement	Relative move for one joint using joint angle.
Kill Statement	Delete a file at runtime.
LCase\$ Function	Returns a string converted to lower case.
LoadPoints Statement	Load a point file and optionally merge it with existing points.
LTrim\$ Function	Trims spaces from left side of string.
Motor Function	Returns motor status.
OLRate Function	Returns overload rating for specified joint.
OpenCom Statement	Opens an RS-232 port.
OpenNet Statement	Opens a TCP/IP port.
Out Function	Returns status of 8 output bits.
OutW Statement	Sets 16 output bits.
OutW Function	Returns status of 16 output bits.
PAGl Function	Retrieve a joint angle from a point.
ParseStr Statement	Parse a string and return array of tokens.
PauseOn Function	Returns the status of Pause.
PDef Function	Check if a point exists.
PLocal Function	Returns the local number of a point.
PLocal Statement	Sets the local number of a point.
POrient Function	Returns the orientation of a point.
POrient Statement	Sets the orientation of a point.
PosFound Function	Returns status of a Find.
PPIs Function	Returns the joint pulse value of a point.
PTran Statement	Relative move for one joint using pulses.
RadToDeg Function	Converts radians to degrees.
Randomize Statement	Creates a new seed for Rnd.
Read Statement	Read one or more characters from a file or port.
Redim Statement	Redimension an array at runtime.
RTrim\$ Function	Trims spaces from right side of string.
SafetyOn Function	Returns status of safeguard.
SavePoints Statement	Save point file.
Seek Statement	Move a file pointer for random access.
SetCom Statement	Sets the parameters of an RS-232 port.
SetNet Statement	Sets the parameters of a TCP/IP port.
Signal Statement	Signals one or more tasks executing WaitSig.
TaskDone Function	Returns True if task is done.

TaskState Function	Returns task state.
UBound Function	Returns upper bound of array dimension.
UCase\$ Function	Converts a string to upper case.
WaitNet Statement	Waits for TCP/IP connection to be established.
WaitPos Statement	Waits for robot to finish move.
WaitSig Statement	Waits for a signal from another task.
Write Statement	Writes one or more bytes to a file or port.
XY Function	Returns a point defined with Cartesian coordinates.

Obsolete commands

The following commands used in previous versions of SPEL (SPEL for Windows 1.x and 2.x) are obsolete. Several commands have been replaced. Others are no longer required.

Command	Description
Carc	Replaced by Arc with CP parameter. Converted automatically when projects created in v1.x or 2.x are opened.
CharSize	Previously used for OPU-300. Will be supported in a future version.
ClrLib	No longer required. Variable library is no longer used and is replaced by Global Preserve variables. Use Erase command to clear Global Preserve variables.
Cls	Previously used for OPU-300. Will be supported in a future version.
CMove	Replaced by Move with CP parameter. Converted automatically when projects created in v1.x or 2.x are opened.
Config	Replaced by SetCom. Converted to SetCom statement when projects created in v1.x or 2.x are opened.
Cooked	No longer required. Removed when projects created in v1.x or 2.x are opened.
CopyFromPc	No longer required. Files are now stored on the PC.
CopyToPc	No longer required. Files are now stored on the PC.
Cursor	Previously used for OPU-300. Will be supported in a future version.
Delete	No longer required. This command was used to delete lines from the monitor window for the current program in the SRC3xx controller main memory.
Dim	No longer supported.
DLoad	Replaced with LoadPoints. Converted to LoadPoints statement when projects created in v1.x or 2.x are opened.
DSave	Replaced with SavePoints. Converted to SavePoints statement when projects created in v1.x or 2.x are opened.
Dsw	No longer supported. This command was mainly used to read switches on the OPU-300.
End	Replaced by Exit Function. Converted to Exit Function statement when projects created in v1.x or 2.x are opened.
Entry	Replaced by Global. Converted to Global statement when projects created in v1.x or 2.x are opened.
Ex_Call	No longer required.
Extern	No longer required.
Feed	No longer supported.
Files	No longer supported.
Format, Frmt	No longer required. This command was previously used to format the RAM disk on the robot controller.
GetDate	No longer required. This command was used to get the PC date from the controller. Now the Date\$ function can be used.
GetTime	No longer required. This command was used to get the PC time from the controller. Now the Time\$ function can be used.
HTask	No longer required. Xqt now has an optional NoPause parameter.
InKey	Replaced by Read statement.
Library	No longer required. The Variable library is no longer used.
LineHist	No longer supported.
Link	No longer required.
List	No longer required.

LLocal	Replaced by Local with L parameter. Converted automatically when projects created in v1.x or 2.x are opened.
Lp	Replace by Power statement.
LpOff	Replace by Power statement.
LpOn	Replace by Power statement.
Merge	This is now an option for the LoadPoints statement.
MkVer	
New	No longer supported. Programs cannot be created in the monitor window.
Normal	Previously used for OPU-300. Will be supported in a future version.
OpUnit	Previously used for OPU-300. Will be supported in a future version.
Opu Print	Previously used for OPU-300. Will be supported in a future version.
Path	No longer required.
PrgRst	Translated to Quit All.
PrgSize	No longer required.
PrgNo	No longer required.
Renum	No longer supported. Use the Edit Renumber command.
Reverse	Previously used for OPU-300. Will be supported in a future version.
RLocal	Replaced by Local with R parameter. Converted automatically when projects created in v1.x or 2.x are opened.
SelDisk	No longer required. This was for disk drives on the SRC3xx controllers. Use ChDrive to change the current drive on the PC at runtime.
SetEnv	No longer required.
SetRaw	No longer required. Removed when projects create in v1.x or 2.x are opened. Use Read to read one or more characters at a time.
SelRB	Replaced by Robot statement. No longer required on a single robot system. All functions can access the robot.
SLocal	Not currently supported.
SysInit	No longer required.
TOn, TOff	No longer supported.
TSpeed	No longer supported.
TSpeeds	No longer supported.
TStat	No longer supported.

SRC-3xx Error Codes

Several error codes have been preserved from the SRC-3xx series controllers.

The following errors have not changed. For a complete error listing, see the SPEL⁺ Language Reference Manual.

Error #	Error Message
0002	Syntax error.
0003	A corresponding GOSUB statement was not found for a RETURN statement.
0004	Too many [GOSUB] statements. (Maximum 512)
0007	There are more nested GOSUB statements than allowed [16].
0009	Subscript out of range.
0016	Program is outside of Function...Fend block.
0027	The (Input or Output) Bit No. entered is out of available range.
0030	There was not enough data to satisfy INPUT parameters.
0053	The specified file cannot be opened.
0063	The specified disk drive is not available.
0065	File read error.
0066	File write error.
0077	Point number is invalid.
0078	The specified point data has not been defined.
0088	The bit has been assigned for remote use.
0092	Safety door input signal failure.
0094	The specified event condition did not occur for WAIT during the timeout period set by TMOUT.
0121	The command could not be executed under emergency stop condition.
0124	The specified value is out of allowable range.
0125	The arm reached the limit of motion range.
0126	The Arm reached the limit of XY motion range (specified by XY LIM).
0129	The current point position is above the specified LIMZ value.
0143	HOME was executed before setting HOMES.
0146	Command not allowed with one or more joints free.
0150	Command not allowed with motors off.
0151	Arm positioning cannot be completed as specified by FINE.
0173	Torque error in the low-power state.
0174	Torque error in the high-power state.
0185	Encoder signal cable disconnection.
0190	Absolute Encoder overheat.
0191	Absolute Encoder overspeed.
0193	Absolute Encoder battery alarm.

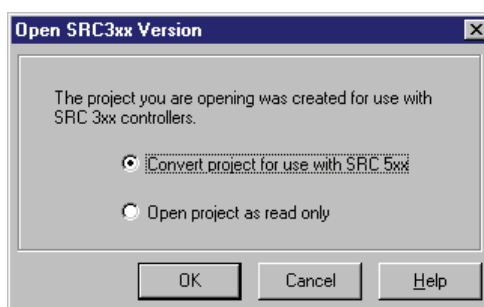
0194	Absolute Encoder check-sum error.
0195	Absolute encoder backup alarm.
0230	MCORG has not been completed.
0231	MCAL has not been completed.
0233	The distance from the Home Sensor to the Encoder Z Phase varies significantly from the previous value.
0360	Undefined vision sequence
0361	Undefined vision object
0362	Undefined vision property
0363	Vision object not found
0364	Improper value for calibration property
0365	No previous sequence with acquire
0366	Illegal property value
0367	Illegal result for specified object
0368	Invalid property or result
0369	Wrong number of arguments
0370	Model not trained
0371	Incomplete calibration
0372	Undefined calibration
0373	Calibration points have not been taught
0374	Printer error
0375	No controller
0376	Bad object type
0377	Vision System error
0378	Sequence already exists
0379	Path not found
0380	Invalid camera
0381	Format string error
0382	Invalid image file
0383	ModelObject does not have angle enabled

Appendix C: Converting SPEL for Windows projects

EPSON RC+ projects are backwards compatible with SPEL for Windows 1.xx and 2.xx projects.

Since there were significant changes between the SPEL and SPEL+ languages, some keywords are obsolete and variable scope has changed.

When a project from a previous version is opened, a dialog box will be displayed asking if you want to convert the project or open as read-only.



The following conversions are performed:

- A new EPSON RC+ project will be created and all program and point files will be copied into the new project. The SPEL for Windows project will remain unchanged.
- Keywords that have been renamed are automatically converted. See Appendix B for details.
- Some statements that are no longer required are removed. For example, all Extern and SelRB statements are removed. See Appendix B for details.
- All Variable Library variables are converted to Global Preserve variables and are declared in the main program of each group.
- All Entry variables are converted to Global variables and the declarations are moved to the top of the file where they are declared.
- Variables declared in a function are moved outside of the function if used in more than one function (recall that in previous versions using SRC 3xx controllers, variables declared in functions can be used in the entire file). If a variable is declared in a function and only used by that function, the declaration is not changed.
- Program Groups are preserved. However, you may want to disable groups after the conversion.

Appendix D: Ethernet I/O Hardware

The table below shows the hardware that is supported for the Ethernet I/O option.

Function	Opto22 Part #	Description
Brain	SNAP-B3000-ENET	Analog / Digital rack controller
	SNAP-ENET-D64	Digital rack controller
	SNAP-WLAN-FH-ADS	Wireless Analog / Digital rack controller
Rack	SNAP-B4Mxx	4-Slot Digital/Analog
	SNAP-B8Mxx	8-Slot Digital/Analog
	SNAP-B12Mxx	12-Slot Digital/Analog
	SNAP-B16Mxx	16-Slot Digital/Analog
	SNAP-D64RS	16-Slot Digital
Digital Input Module	SNAP-IDC5	4-Channel DC Input 10-32 VDC
	SNAP-IDC5FAST	4-Channel Hi-Speed DC Input 2.5 - 16 VDC
	SNAP-IAC5	4-Channel AC Input 90-140 VAC
	SNAP-IAC5A	4-Channel AC Input 90-140 VAC
	SNAP-IDC5Q	2-Axis Quadrature Encoder Input
Digital Output Module	SNAP-ODC5R	DC Output Relay N.O.
	SNAP-ODC5R5	DC Output Relay N.C.
	SNAP-ODC5SNK	4-Channel DC Output 5-60 VDC, Sink
	SNAP-ODC5SRC	4-Channel DC Output 5-60 VDC, Source
	SNAP-OAC5	4-Channel AC Output 12-280 VAC
	SNAP-OAC5-I	4-Channel AC Output Isolated
	SNAP-OAC5MA	4-Channel AC Output Manual / Auto
Analog Input Module	SNAP-AIARMS	2-Channel Analog Input 0 to 10 amp AC/DC
	SNAP-AICTD	2-Channel Analog Temperature Input - ICTD
	SNAP-AICTD-4	4-Channel Analog Temperature Input - ICTD
	SNAP-AIMA	2-Channel Analog Input -20 to +20 ma
	SNAP-AIMA-4	4-Channel Analog Input -20 to +20 ma

Function	Opto22 Part #	Description
	SNAP-AIR40K-4	4-Channel Analog Input Thermistor
	SNAP-AIRATE	2-Channel Analog Rate Input 0 - 25000 Hz
	SNAP-AIRTD	2-Channel Analog Input RTD
	SNAP-AITM	2-Channel Analog Input E, J, K Thermocouple
	SNAP-AITM-2	2-Channel Analog Input B, C, D, G, N, T, R, S, Thermocouple
	SNAP-AIV	2-Channel Analog Input -10 to +10 VDC
	SNAP-AIV4	4-Channel Analog Input -10 to +10 VDC
	SNAP-AIVRMS	2-Channel Analog Input 0 to 250 VRMS
Analog Output Module	SNAP-AOA3	1-Channel Analog Output 4 to 20 ma
	SNAP-AOA23	2-Channel Analog Output 4 to 20 ma
	SNAP-AOA28	2-Channel Analog Output 0 to 20 ma
	SNAP-AOD29	2-Channel Analog Output 5 to 60 VDC
	SNAP-AOV5	1-Channel Analog Output 0 to +10 VDC
	SNAP-AOV7	1-Channel Analog Output -10 to +10 VDC
	SNAP-AOV25	2-Channel Analog Output 0 to +10 VDC
	SNAP-AOV27	2-Channel Analog Output -10 to +10 VDC

Index

A

abort all · 71
accuracy · 133
application layout · 151
applications · 151
arc · 133
arch · 52
armset · 57
arrange icons · 103
arrays · 113
audit · 225
auto start · 152

B

breakpoints · 71, 164

C

Call Stack · 73
chaining · 119
clear breakpoints · 72
close all · 103
command line options · 29
commands · 109
comments · 116
compile · 67
constants · 142
control device · 70
Conveyor Tracking · 227
copy selection · 42
curves · 133
cut selection · 42

D

data types · 110
debugging · 164
designing applications · 151
digital I/O · 179
DLL

calling · 144
creating · 145
debugging · 149
documentation · 4

E

edit menu · 42
editor
 preferences · 96
emergency stop · 141
EPSON RC+
 about box · 106
 GUI · 34
 shutdown · 32
 starting · 25
error editor · 91
error handling · 117
Ethernet I/O · 183
event · 264
exit application · 42
extract coordinates · 138

F

file menu · 37
files
 creating · 37
 deleting · 39
 importing · 39
 printing · 40
 renaming · 38
 restoring · 38
 saving · 38
 working with · 115
find next · 44
find text · 43
force monitor · 90
force sensing · 211
Force Sensor Specifications · 313
functions
 calling · 108
 naming · 109

G

getting started · 19
global variables · 112
group rights · 224
groups · 49

H

hardware layout · 3
help · 35, 104
help menu · 104
homeset · 61
hordr · 62

I

I.O
 memory · 139
I/O
 commands · 139
 hardware · 139
I/O monitor · 85
I/O systems · 179
import · 39
indent · 45
installation
 hardware · 19
 offline development · 20
 software · 19
instant watch · 72
introduction · 1

J

jog and teach · 76
jog controls · 77
jogging preferences · 97
joint motion · 133

K

key · 20

L

labels · 116
language preferences · 102
license agreement · 263
line numbers · 109
linear motion · 132
local variables · 111

M

macros · 88

maintenance · 89, 141
mccodr · 63
memory · 298, 299
memory I/O · 139
MKVER · 89
module level variables · 111
monitor window · 86
motion commands · 132
motion systems · 169
multi-statements · 116
multi-tasking · 118

N

new commands · 304

O

obsolete commands · 307
online help · 35
operation · 25
operator window · 69, 167
operator window preferences · 98
options · 4, 102, 207
 force sensing · 211
 installing · 261
 security · 221
outdent · 45

P

paste · 42
point to point motion · 132
points · 51
 attributes · 136
 defining · 135
 editing · 160
 saving and loading · 136
 teaching · 80
predefined constants · 142
preferences
 editor · 96
 jogging · 97
 operator window · 98
 project · 95
 run window · 99
 vision · 100, 102
printing · 40
program
 stepping · 164
program groups · 49
programs
 comments · 116
 compile · 67
 editing · 157
 structure · 108
 writing your first · 22
project menu · 47

projects

- backing up · 157
- building · 68, 156
- copying · 66, 157
- creating · 47
- default · 153
- deleting · 67
- editing · 49
- managing · 152
- opening · 48
- preferences · 95
- rebuilding · 68
- renaming · 65
- save as · 64
- saving · 64

R

- range · 60
- remote control · 191
- renumber · 46, 159
- replace text · 44
- resume · 71
- rights · 224
- robot
 - adding · 175
 - configuration · 175
 - control panel · 75
 - deleting · 176
 - homing · 132
 - motion commands · 132
 - parameters · 52, 177
 - points · 133, 152
 - using multiple · 120
- rs232 · 199
- run menu · 69
- run window · 162
- run window preferences · 99

S

- safety · 151
- search for help · 105
- security · 221
- security audit · 225
- security commands · 225
- security configuration · 221
- SeikoRCS users · 4
- select all · 45
- setup menu · 93
- SETVER · 90
- software key · 20
- SPEL driver manager · 32
- SPEL for Windows users · 4
- SPEL+ Language · 107
- start command · 69
- statements · 109
- status window · 156
- step into · 70

- step over · 70, 71
- strings
 - working with · 114
- strip line numbers · 46
- syntax errors · 159
- syntax help · 158
- system
 - block diagram · 3
 - overview · 2
 - power up · 25
 - shutdown · 32

T

- task manager · 81
- tcp/ip · 201
- teach mode · 80
- tile windows · 103
- tlset · 55, 56, 58
- toolbars · 36
- tools menu · 74
- traps · 140

U

- undo · 42
- user configuration · 223
- user errors · 91
- user groups configuration · 224

V

- variables
 - clearing · 113
 - displaying · 72, 166
 - global · 112
 - local · 111
 - module level · 111
 - naming · 109
 - points · 135
 - working with · 110, 111
- vision preferences · 100

W

- weight · 53, 54
- welcome · 1
- window menu · 103
- windows security administration · 20

X

- xylim · 59

