

ROBOTICS

Operating manual

IRC5 with FlexPendant



Trace back information: Workspace 24B version a1 Checked in 2024-05-30 Skribenta version 5.5.019

Operating manual IRC5 with FlexPendant

RobotWare 6.15.07

Document ID: 3HAC050941-001 Revision: Q

© Copyright 2024 ABB. All rights reserved. Specifications subject to change without notice.

The information in this manual is subject to change without notice and should not be construed as a commitment by ABB. ABB assumes no responsibility for any errors that may appear in this manual.

Except as may be expressly stated anywhere in this manual, nothing herein shall be construed as any kind of guarantee or warranty by ABB for losses, damage to persons or property, fitness for a specific purpose or the like.

In no event shall ABB be liable for incidental or consequential damages arising from use of this manual and products described herein.

This manual and parts thereof must not be reproduced or copied without ABB's written permission.

Keep for future reference.

Additional copies of this manual may be obtained from ABB.

Original instructions.

© Copyright 2024 ABB. All rights reserved. Specifications subject to change without notice.

Table of contents

	Overv Produ	view of this manual uct documentation	9 12	
1	Welc	come to IRC5 15		
	1.1	About this section	15	
	1.2	The IRC5 controller	16	
	1.3	The FlexPendant	17	
	1.4	RobotStudio	24	
	1.5	When to use different jogging devices	25	
	1.6	Buttons and ports on the controller	28	
2	Navig	gating and handling FlexPendant	31	
	2.1	About this chapter	31	
	2.2	The Main menu	32	
		2.2.1 HotEdit menu	32	
		2.2.2 FlexPendant Explorer	34	
		2.2.3 Inputs and Outputs, I/O	35	
		2.2.4 Jogging	36	
		2.2.5 Production Window	38	
		2.2.6 Program Data	39	
		2.2.7 Program Editor	41	
		2.2.8 Backup and Bestore	43	
		229 Calibration	44	
		2210 Control Panel	46	
		2211 Event Log	47	
		2.2.17 Event Log	10	
		2.2.12 System into	51	
		2.2.13 hestalt	52	
	0.0	2.2.14 LOY OII	52	
	2.3	Operator window	23	
	2.4	Status par	54	
	2.5		55	
		2.5.1 The Quickset menu	55	
		2.5.2 Quickset menu, Mechanical unit	56	
		2.5.3 Quickset menu, Increment	61	
		2.5.4 Quickset menu, Run Mode	62	
		2.5.5 Quickset menu, Step Mode	63	
		2.5.6 Quickset menu, Speed	64	
	_	2.5.7 Quickset menu, Tasks	65	
	2.6	Basic procedures	66	
		2.6.1 Using the soft keyboard	66	
		2.6.2 Messages on the FlexPendant	68	
		2.6.3 Scrolling and zooming	69	
		2.6.4 Filtering data	70	
		2.6.5 Process applications	73	
		2.6.6 Granting access for RobotStudio	74	
		2.6.7 Logging on and off	75	
	2.7	Changing FlexPendant settings	77	
		2.7.1 System settings	77	
		2.7.1.1 Setting default paths	77	
		2.7.1.2 Defining a view to be shown during operating mode change or startup	78	
		2.7.1.3 Changing the background image	79	
		2714 Defining visibility level for LIAS protected functions	80	
		2715 Defining an additional test view	R1	
		2716 Defining on additional test view	82	
		2717 Defining which tasks should be selectable in the tasks papel	Q/I	
		2.7.1.7 Demining which lashs should be selectable in the lashs parter	25 25	
		Linito managing the display of controller and system hattle	00	

		2.7.2	Basic settings2.7.2.1Changing brightness and contrast2.7.2.2Adapting the FlexPendant for left-handed users2.7.2.3Controller settings2.7.2.4Configuring Most Common I/O2.7.2.5Changing language2.7.2.6Changing programmable keys2.7.2.7Calibrating the touchscreen	86 87 89 91 92 93 95
3	Joggi	ing		97
	3.1 3.2 3.3 3.4 3.5 3.6	Introdu Coordi Joystic Restric Coordi Basic s 3.6.1 3.6.2 3.6.3 3.6.4 3.6.5 3.6.6 3.6.7 3.6.8 3.6.9 3.6.10	uction to jogging nate systems for jogging nate systems for jogging 1 ctions to jogging 1 nated jogging 1 settings for jogging 1 settings for jogging 1 Selecting mechanical unit for jogging 1 Selecting motion mode 1 Selecting tool, work object, and payload 1 Setting the tool orientation 1 Jog axis by axis 1 Selecting coordinate system 1 Locking the joystick in specific directions 1 Incremental movement for precise positioning 1 Reading the exact position 1 Aligning tools 1	97 99 05 06 07 08 08 10 11 12 13 14 15 17 19 21
4	Prog	rammin	g and testing 1	23
	4.1	Before	you start programming	23
	4.2	Using	RAPID programs	24
	4.3	Progra	Imming concept	26
		4.3.1	Handling of programs	20
		4.3.2	Handling of modules	29
		4.3.3	Handling of routines	32
		4.3.4	Handling of instructions	37
		4.3.5	Example: Add movement instructions	41
		4.3.6	About the Program and Motion Pointers	42
	4.4	Data ty	/pes	43
		4.4.1	Creating new data in specific tasks, modules, or routines	43
		4.4.2	Editing data instance	44
	45	Toole	Luiting uata instances	50
	7.5	151	What is a tool?	50
		452	What is the tool center point?	52
		453	Creating a tool	54
		454	Defining the tool frame	57
		4.5.5	Editing the tool data	61
		4.5.6	Editing the tool declaration	64
		4.5.7	Deleting a tool	65
		4.5.8	Setup for stationary tools	66
	4.6	Work o	biects	68
		4.6.1	What is a work object? 1	68
		4.6.2	Creating a work object 1	69
		4.6.3	Defining the work object coordinate system 1	70
		4.6.4	Editing the work object data 1	74
		4.6.5	Editing the work object declaration 1	75
		4.6.6	Deleting a work object 1	76
	4.7	Payloa	lds 1	77
		4.7.1	Creating a payload 1	77

		4.7.2 Editing the payload data	179
		4.7.3 Editing the payload declaration	181
		4.7.4 Deleting a payload	182
	4.8	Testing	183
		4.8.1 About the automatic mode	183
		4.8.2 About the manual mode	185
		4.8.3 Using the hold-to-run function	188
		4.8.4 Running the program from a specific instruction	189
		4.8.5 Running a specific routine	190
	4.0	4.8.6 Stepping instruction by instruction	191
	4.9	Service routines	194
		4.9.1 Running a service routine	194
		4.9.2 Dattery Shutdown Service routine	190
		4.9.5 Axis Calibration Pondulum CalPondulum convice routine	200
		4.9.4 Calibration Fendulum, Calibration Service fourine contine	200
		4.9.5 Service mornation system, service routine	201
		4.9.0 Lodulueniny, load identification service routine	202
		198 Wrist ontimization service routine	201
			221
5	Runn	ing in production	223
	51	Basic procedures	223
	•	5.1.1 Starting programs	223
		5.1.2 Stopping programs	226
		5.1.3 Using multitasking programs	227
		5.1.4 Using motion supervision and non motion execution	229
		5.1.5 Connecting a FlexPendant	232
		5.1.6 Using the hot plug option	234
	5.2	Troubleshooting and error recovery	237
		5.2.1 General procedure when troubleshooting	237
		5.2.2 Returning the robot to the path	238
		5.2.3 Running RAPID program with uncalibrated mechanical unit	239
	5.3	Operating modes	240
		5.3.1 Present operating mode	240
		5.3.2 Switching from manual to automatic mode	242
		5.3.3 Switching from automatic to manual mode	244
		5.3.4 Switching to manual full speed mode	245
	5.4	Modifying positions	246
		5.4.1 Modifying and tuning positions	246
		5.4.2 Modifying positions in the Program Editor or Production Window	247
		5.4.3 Tuning positions with HotEdit	251
		5.4.4 Working with displacements and offsets	255
		5.4.5 Moving the robot to a programmed position	257
6	Hand	ling inputs and outputs, I/O	259
	61	Viewing signal lists	250
	6.2	Simulating and changing signal values	260
	6.3	Viewing signal group	263
	6.4	Safety signals	264
	0.1	6.4.1 Safety I/O signals	264
		,	
7	Hand	ling the event log	267
	7.1	Accessing the event log	267
	7.2	Deleting log entries	268
	7.3	Saving log entries	269
~	. .		0 - -
8	Back	up and restore	271
	8.1	Backup the system	271

	8.2 8.3	.2 Important when performing backups	
9	Calibrating		279
9.1 How to check if the robot needs calibration9.2 Updating revolution counters		How to check if the robot needs calibration Updating revolution counters	
Ind	dex		283

Overview of this manual

About this manual

This manual contains instructions for daily operation of IRC5 based robot systems using a FlexPendant.



Note

It is the responsibility of the integrator to provide safety and user guides for the robot system.

Usage

This manual should be used during operation.

Some actions that are more advanced, or not used in the daily operation, are described in *Operating manual - IRC5 Integrator's guide*.



Note

Before any work on or with the robot is performed, the safety information in the product manual for the controller and manipulator must be read.

Who should read this manual?

This manual is intended for:

- operators
- product technicians
- service technicians
- robot programmers

Prerequisites

The reader should:

- Be familiar with the concepts described in *Operating manual Getting started*, *IRC5 and RobotStudio*.
- Be trained in robot operation.

References

Product manual - IRC5 IRC5 with main computer DSQC1000 or later.	3HAC047136-001
Product manual - IRC5 Panel Mounted Controller IRC5 with main computer DSQC1000 or later.	3HAC047137-001
Product manual - IRC5 Compact IRC5 with main computer DSQC1000 or later.	3HAC047138-001
Operating manual - Getting started, IRC5 and RobotStudio	3HAC027097-001
Operating manual - RobotStudio	3HAC032104-001
Operating manual - Service Information System	3HAC050944-001

Continues on next page

Continued

Operating manual - Troubleshooting IRC5	3HAC020738-001
Operating manual - IRC5 Integrator's guide	3HAC050940-001
Operating manual - Calibration Pendulum	3HAC16578-1
Technical reference manual - System parameters	3HAC050948-001
Technical reference manual - RAPID Overview	3HAC050947-001
Technical reference manual - RAPID Instructions, Functions and Data types	3HAC050917-001
Technical reference manual - RAPID kernel	3HAC050946-001
Application manual - Additional axes and standalone controller	3HAC051016-001
Application manual - Controller software IRC5	3HAC050798-001
Application manual - MultiMove	3HAC050961-001



The document numbers that are listed for software documents are valid for RobotWare 6. Equivalent documents are available for RobotWare 5.

Revisions

Revision	Description
-	Released with RobotWare 6.0.
A	 Released with RobotWare 6.02. Added information about time limitation to the section <i>Switching to</i> manual full speed mode on page 245. Added information on <i>Load diagram check</i> in the section <i>LoadIdentify</i>, <i>load identification service routine on page 202</i>. Updated the section <i>Controller settings on page 89</i>.
В	 Released with RobotWare 6.03. Updated the section Step mode on page 63. Updated the section Illustration Control Panel on page 46. Updated the section System Info on page 49. Added a warning note regarding load data in the sections Before you start programming on page 123, Tools on page 150, Payloads on page 177, and Service routines on page 194.
С	 Released with RobotWare 6.04. Updated the section <i>Quickset menu</i>, <i>Increment on page 61</i>. Updated the section <i>Restore the system on page 275</i>. Updated the section <i>LoadIdentify, load identification service routine on page 202</i>.
D	 Released with RobotWare 6.05. Added the new section Managing the display of controller and system name on page 85. Added the new section Brake check service routine on page 214. Updated the section Defining a view to be shown during operating mode change or startup on page 78. Updated descriptions of stops. Removed the information about time limitation in the section Switching to manual full speed mode on page 245.

Continued

Revision	Description
E	 Released with RobotWare 6.06. Updated the section <i>Brake check service routine on page 214</i>. Updated the section <i>Backup and restore on page 271</i>. Updated the section <i>Example of backward execution on page 192</i>. Updated the section <i>Controller settings on page 89</i>.
F	 Released with RobotWare 6.07. Safety section restructured. FlexPendant Home Screen background image is updated. Updated the section <i>When is backup possible? on page 273</i> Updated the section <i>LoadIdentify for 4-axis palletizing and SCARA robots on page 209</i> Updated the section <i>Simulating and changing signal values on page 260</i>
G	 Released with RobotWare 6.08. Updated the section <i>Controller settings on page 89</i>. Updated the section <i>Editing motion supervision settings on page 230</i> Updated the section <i>Changing programmable keys on page 93</i>. Added information and code example to the section <i>Brake check for MultiMove systems on page 215</i>.
Н	 Released with RobotWare 6.10. Updated the information regarding tool data, see <i>Editing the tool data</i> on page 161.
J	 Released with RobotWare 6.10.01. Information regarding General SIS data and Mechanical units updated in section System Info on page 49.
к	 Released with RobotWare 6.11. The safety information is moved to the product manuals for the controller and the manipulator. Updated information about queueing backups. Added Axis Calibration service routine on page 199.
L	Released with RobotWare 6.12. Added Wrist optimization service routine on page 221.
М	 Released with RobotWare 6.13. Added information for delta robots, see LoadIdentify, load identification service routine on page 202.
N	 Released with RobotWare 6.14. Added information for delta robots, see LoadIdentify, load identification service routine on page 202. The name of the service routine Bat_Shutdown is changed to BatteryShutDown.
P	 Released with RobotWare 6.14.01. Added information about a new version of the FlexPendant. Removed information about T10.
Q	 Released with RobotWare 6.15.07. Added information that Advanced Restart is described in Integrators guide.

Product documentation

Categories for user documentation from ABB Robotics

The user documentation from ABB Robotics is divided into a number of categories. This listing is based on the type of information in the documents, regardless of whether the products are standard or optional.



All documents can be found via myABB Business Portal, <u>www.abb.com/myABB</u>.

Product manuals

Manipulators, controllers, DressPack, and most other hardware is delivered with a **Product manual** that generally contains:

- · Safety information.
- Installation and commissioning (descriptions of mechanical installation or electrical connections).
- Maintenance (descriptions of all required preventive maintenance procedures including intervals and expected life time of parts).
- Repair (descriptions of all recommended repair procedures including spare parts).
- Calibration.
- Troubleshooting.
- Decommissioning.
- Reference information (safety standards, unit conversions, screw joints, lists of tools).
- Spare parts list with corresponding figures (or references to separate spare parts lists).
- References to circuit diagrams.

Technical reference manuals

The technical reference manuals describe reference information for robotics products, for example lubrication, the RAPID language, and system parameters.

Application manuals

Specific applications (for example software or hardware options) are described in **Application manuals**. An application manual can describe one or several applications.

An application manual generally contains information about:

- The purpose of the application (what it does and when it is useful).
- What is included (for example cables, I/O boards, RAPID instructions, system parameters, software).
- How to install included or required hardware.
- How to use the application.

Continues on next page

Continued

• Examples of how to use the application.

Operating manuals

The operating manuals describe hands-on handling of the products. The manuals are aimed at those having first-hand operational contact with the product, that is production cell operators, programmers, and troubleshooters.

This page is intentionally left blank

1 Welcome to IRC5

1.1 About this section

Overview

This section presents an overview of the FlexPendant, the IRC5 controller, and RobotStudio.

A robot consists of a robot controller, the FlexPendant, RobotStudio, and one or several manipulators or other mechanical units.

This manual describes a robot without options, not a robot system. However, in a few places, the manual gives an overview of how options are used or applied. Most options are described in detail in their respective application manual.

1.2 The IRC5 controller

1.2 The IRC5 controller

The IRC5 controller	
	The IRC5 controller contains all functions needed to move and control the robot.
	The standard IRC5 controller consists of a single cabinet. The controller is also available in a compact version, <i>IRC5 Compact</i> , and it can also be integrated in an external cabinet, <i>Panel Mounted Controller</i> .
	When running more than one robot with one controller (MultiMove option), an extra drive module must be added for each additional robot. However, a single control module is used.
Related information	
	Product manual - IRC5, IRC5 of design M2004.
	Product manual - IRC5, IRC5 of design 14.
	Product manual - IRC5 Panel Mounted Controller, IRC5 of design M2004.
	Product manual - IRC5 Panel Mounted Controller, IRC5 of design 14.
	Product manual - IRC5 Compact, IRC5 of design M2004.
	Product manual - IRC5 Compact, IRC5 of design 14.
	Application manual - MultiMove.

1.3 The FlexPendant

1.3 The FlexPendant

Introduction to the FlexPendant

The FlexPendant is a hand held operator unit that is used for many of the tasks when operating a robot: running programs, jogging the manipulator, modifying programs, and so on.

The FlexPendant is designed for continuous operation in harsh industrial environment. Its touchscreen is easy to clean and resistant to water, oil, and accidental welding splashes.

The FlexPendant consists of both hardware and software and is a complete computer in itself. It is connected to the robot controller by an integrated cable and connector.

The hot plug button option makes it possible to disconnect the FlexPendant in automatic mode and continue running without it.

The FlexPendant is available in different versions, as the hardware has been updated over the years. The exact appearance on the graphics might therefore differ slightly from reality.



If protective gloves are used, these must be compatible with touchscreens when using the FlexPendant.

1 Welcome to IRC5

1.3 The FlexPendant *Continued*

These are the main parts of the FlexPendant. FlexPendant with emergency stop button FlexPendant with emergency stop button at the connector at the outer edge в С н В D D С G G È xx1400001636 F Ε xx2200002054 A Connector В Touchscreen С **Emergency stop button** D Joystick Е **Reset button** F USB port G Three-position enabling device н Thumb button (Not available on all versions of FlexPendant.) J Stylus pen (Not available on all versions of FlexPendant.)

Main parts

Joystick	
	Use the joystick to move the manipulator. This is called jogging the robot. There are several settings for how the joystick will move the manipulator.
Reset button	
	If the FlexPendant freezes during operation, press the reset button to restart the FlexPendant.
	The reset button resets the FlexPendant, not the system on the controller.
USB port	
	Connect a USB memory to the USB port to read or save files. The USB memory is displayed as drive /USB:Removable in dialogs and FlexPendant Explorer.
	Note
	Close the protective cap on the USB port when not used.
Stylus pen	
	The stylus pen included with the FlexPendant is located on the back. Pull the small handle to release the pen.

Use the stylus pen to tap on the touch screen when using the FlexPendant. Do not use screw drivers or other sharp objects.

(Not available on all versions of FlexPendant.)

Hard buttons

The following hard buttons are available on the FlexPendant.

Button	Description
	Programmable keys, 1 - 4.
	Select mechanical unit.
	Toggle motion mode, reorient or linear.
	Toggle motion mode, axis 1-3 or axis 4-6.
	Toggle increments.
	Step BACKWARD button. Executes one instruction backward as button is pressed.
	START button. Starts program execution.
	Step FORWARD button. Executes one instruction forward as button is pressed.
	STOP button. Stops program execution.

1 Welcome to IRC5

1.3 The FlexPendant *Continued*

Three-position enabling device

The person using the three-position enabling device is responsible to observe the safeguarded space for hazards due to robot motion and any other hazards related to the robot.

The three-position enabling device is located on the FlexPendant. When continuously held in center-enabled position, the three-position enabling device will permit robot motion and any hazards controlled by the robot. Release of or compression past the center-enabled position will stop the robot motion.



For safe use of the three-position enabling device, the following must be implemented:

- The three-position enabling device must never be rendered inoperational in any way.
- If there is a need to enter safeguarded space, always bring the FlexPendant. This is to enforce single point of control.

Thumb button

The thumb button is only available on the FlexPendant with emergency stop located at the connector.

The thumb button is used for hold-to-run.

Hold-to-run is described in section Using the hold-to-run function on page 188.

How to hold the FlexPendant

FlexPendant is typically operated while being held in the hand. The right-handed users use their left-hand to support the FlexPendant while their right-hand performs the operations on the touch screen. However, the left-handed users can easily adapt FlexPendant for their use.

For more details, see the section Adapting the FlexPendant for left-handed users on page 87.



Touchscreen elements

The illustration shows important elements of the FlexPendant touchscreen.



xx1400001446

Α	Main menu
В	Operator window
С	Status bar
D	Close button
E	Task bar
F	Quickset menu

Main menu

The following items can be selected from the Main menu:

- HotEdit
- Inputs and Outputs
- Jogging
- Production Window
- Program Editor
- Program Data
- Backup and Restore
- Calibration
- Control Panel
- Event Log

	FlexPendant Explorer
	System Info
	• etc.
	This is further described in section <i>The Main menu on page 32</i> .
Operator window	
	The operator window displays messages from robot programs. This usually happens when the program needs some kind of operator response in order to continue. This is described in section <i>Operator window on page 53</i> .
Status bar	
	The status bar displays important information about system status, such as operating mode, motors on/off, program state and so on. This is described in section <i>Status bar on page 54</i> .
Close button	
	Tapping the close button closes the presently active view or application.
Task bar	
	You can open several views from the Main menu, but only work with one at a time. The task bar displays all open views and is used to switch between these.
Quickset menu	
	The quickset menu provides settings for jogging and program execution. This is described in section <i>The Quickset menu on page 55</i> .

1.4 RobotStudio

1.4 RobotStudio

Overview of RobotStudio

RobotStudio is an engineering tool for the configuration and programming of ABB robots, both real robots on the shop floor and virtual robots in a PC. To achieve true offline programming, RobotStudio utilizes ABB VirtualRobot™ Technology. RobotStudio has adopted the Microsoft Office Fluent User Interface. The Office Fluent UI is also used in Microsoft Office. As in Office, the features of RobotStudio are designed in a workflow-oriented way.

With add-ins, RobotStudio can be extended and customized to suit the specific needs. Add-ins are developed using the RobotStudio SDK. With the SDK, it is also possible to develop custom SmartComponents which exceed the functionality provided by RobotStudio's base components.

For more information, see Operating manual - RobotStudio.

RobotStudio for real controllers

RobotStudio allows, for example, the following operations when connected to a real controller:

- Installing and modifying RobotWare systems on controllers, using the **Installation Manager 6** function.
- Text-based programing and editing, using the RAPID Editor.
- File manager for the controller.
- Administrating the User Authorization System.
- Configuring system parameters.

1.5 When to use different jogging devices

1.5 When to use different jogging devices

Overview

For operating and managing the robot, you can use any of the following:

- FlexPendant: Optimized for handling robot motions and ordinary operation
- RobotStudio: Optimized for configuration, programming and other tasks not related to the daily operation.
- RobotStudio Online Apps : Optimized for jogging, managing, working with the frames, calibration methods and RAPID programs available in the robot controller.

Start, restart and shut down the controller

То	Use
Start the controller	The power switch on the controller's front panel.
Restart the controller	The FlexPendant, RobotStudio, RobotStudio Online Apps or the power switch on the controller's front panel.
Shut down the controller	The power switch on the controller's front panel or the FlexPendant, tap Restart , then Advanced .
Shut down the main computer	The FlexPendant.

Run and control robot programs

То	Use
Jog a robot	The FlexPendant.
Start or stop a robot program	The FlexPendant, RobotStudio or RobotStudio Online Apps.
Start and stop background tasks	The FlexPendant, RobotStudio or RobotStudio Online Apps.

Communicate with the controller

То	Use
Acknowledge events	The FlexPendant or RobotStudio Online Apps.
View and save the controller's event logs	RobotStudio, FlexPendant or the RobotStu- dio Online Apps.
Back up the controller's software to files on the PC or a server	RobotStudio, FlexPendant or the RobotStu- dio Online Apps.
Back up the controller's software to files on the controller	The FlexPendant or RobotStudio Online Apps.
Transfer files between the controller and net- work drives	RobotStudio, FlexPendant or the RobotStu- dio Online Apps.

1 Welcome to IRC5

1.5 When to use different jogging devices *Continued*

Program a robot

То	Use
Create or edit robot programs in a flexible way. This is suit- able for complex programs with a lot of logic, I/O signals or action instructions	 RobotStudio to create the program's structure and most of the source code and the FlexPendant to store robot positions and make final adjustments to the program. When programming, RobotStudio provides the following advantages: A text editor optimized for RAPID code, with auto-text and tool-tip information about instructions and parameters. Program check with program error marking. Close access to configuration and I/O editing.
Create or edit a robot program in a supportive way. This is suitable for programs that mostly consist of move instruc- tions	 The FlexPendant. When programming, the FlexPendant provides the following advantages: Instruction pick lists Program check and debug while writing Possibility to create robot positions while programming
Add or edit robot positions	The FlexPendant with a combination of suitable RobotStudio Online Apps.
Modify robot positions	The FlexPendant with a combination of suitable RobotStudio Online Apps.

Configure the robot's system parameters

То	Use
Edit the system parameters of the running system	RobotStudio, FlexPendant or the RobotStu- dio Online Apps.
Save the robot's system parameters as config- uration files	RobotStudio, FlexPendant or the RobotStu- dio Online Apps.
Load system parameters from configuration files to the running system	RobotStudio, FlexPendant or the RobotStu- dio Online Apps.
Load calibration data	RobotStudio, FlexPendant or the RobotStu- dio Online Apps.

Create, modify and install systems

То	Use
Create or modify a system	RobotStudio together with RobotWare and a valid Ro- botWare Key for systems based on RobotWare 5.
	RobotStudio together with RobotWare and license file for systems based on RobotWare 6.
Install a system on a controller	RobotStudio
Install a system on a controller from a USB memory	The FlexPendant.

Calibration

То	Use
Calibrate base frame etc.	The FlexPendant or the RobotStudio Online Apps.

Continues on next page

1.5 When to use different jogging devices *Continued*

То	Use
Calibrate tools, work objects etc.	The FlexPendant or the RobotStudio Online Apps.

Related information

The table below specifies which manuals to read, when performing the various tasks referred to:

Recommended use	for details, see manual	Document number	
FlexPendant	Operating manual - IRC5 with Flex- Pendant	3HAC050941-001	
RobotStudio	Operating manual - RobotStudio	3HAC032104-001	

1 Welcome to IRC5

1.6 Buttons and ports on the controller

1.6 Buttons and ports on the controller

Buttons and ports on the controller

These are the buttons and ports on an IRC5 controller. Some buttons and ports are options and might not be available on your controller. The buttons and ports look the same but the placing can differ depending on the controller model (IRC5 Standard, IRC5 Compact, or IRC5 Panel Mounted Controller) and if there is an external operator's panel.



xx0600002782

А	Main switch
В	Emergency stop
С	Motors on
D	Mode switch
E	Safety chain LEDs (option)
G	Service port for PC (option)
н	Duty time counter (option)
J	Service outlet 115/230 V, 200 W (option)
К	Hot plug button (option)
L	Connector for FlexPendant

Related information

Product manual - IRC5, IRC5 of design 14.

Continues on next page

1.6 Buttons and ports on the controller *Continued*

Product manual - IRC5 Panel Mounted Controller, IRC5 of design 14. Product manual - IRC5 Compact, IRC5 of design 14. Operating manual - Troubleshooting IRC5. This page is intentionally left blank

2 Navigating and handling FlexPendant

2.1 About this chapter

Introduction to this chapter

This chapter will help you to work efficiently with the FlexPendant. The important elements for navigation illustrated in *Touchscreen elements on page 22* are described here.

All views of the Main menu, the main element for navigation, are described in overview with references to further details on how to use their functions.

In addition, this chapter provides information about basic procedures, such as how to use the soft keyboard for entering text or numbers, how to scroll and zoom the graphical touch screen, and how to use the filtering function. How to log on and log off is also described.

Handling and troubleshooting the FlexPendant

How to handle and clean the FlexPendant is described in the product manual for the robot controller.

Troubleshooting the FlexPendant is described in *Operating manual* - *Troubleshooting IRC5*.

Hardware and software options

Note that this manual covers only the views of a basic RobotWare system. Process applications such as arc welding, dispense, or plastics are started from the Main menu but not described in this manual. All options are detailed in their respective application manual.

2.2.1 HotEdit menu

2.2 The Main menu

2.2.1 HotEdit menu

HotEdit

HotEdit is used to tune programmed positions. This can be done in all operating modes and even while the program is running. Both coordinates and orientation can be tuned.

HotEdit can only be used for named positions of the type robtarget (see limitations below).

The functions available in HotEdit may be restricted by the user authorization system, UAS.

Illustration of the HotEdit view

Manual MySystem (RSTEST4)	1otors On Stopped (2 of 2) (Speed 100%)	X
Programmed targets	Selected targets Targets	Offset
anin main m20 (tool0)	p20 p30	0.0 0.0
● p20 (tool0) →		
 p40 (tool0) p10 (tool0) 		
	<< Tune Targ	ets
File Baseline		Apply
T_ROB1: HotEdit		

en0500001542

Functions available in HotEdit

Programmed targets	Lists all named positions in a tree view. Select one or several positions to be tuned by tapping the arrow. Notice that if a position is used at several places in your program, any change made to the offset will take effect everywhere it is used.
Selected targets	Lists all selected positions and their current offset. To remove a position from the selection you tap it and then tap the trash.
File	Saves and loads selections of positions to be tuned. If your system uses UAS, this may be the only way to select positions for HotEdit.

Continues on next page

2 Navigating and handling FlexPendant

2.2.1 HotEdit menu Continued

Baseline	Used to apply or reject new offset values to the baseline, which holds the position values currently seen as the original ones. When you are satisfied with your HotEdit session and want to save the new offset values as the original position values, you apply these to the baseline. The old baseline values for these positions are now gone, and cannot be restored.
Tune targets	Displays settings for tuning: Coordinate system, Tuning mode and Tuning increment. Make your choices and then use the plus and minus icons to specify tuning of selected targets.
Apply	Tap Apply to make the settings made in the Tune Targets view take effect. Note that this does not change the baseline values of the positions!

HotEdit offers advanced functionality, which has to be handled carefully. Be aware that new offset values will be used immediately by a running program once the **Apply** button has been tapped.

Before you start using the HotEdit functionality it is strongly recommended to read *Tuning positions with HotEdit on page 251*, where HotEdit limitations and procedures as well as the baseline concept is detailed.

Related information

See section *Modifying and tuning positions on page 246* for a general overview on how to modify programmed positions.

For modifying positions by jogging the robot to the new position, see section *Modifying positions in the Program Editor or Production Window on page 247.*

For detailed information about HotEdit, see *Tuning positions with HotEdit on page 251*.

Technical reference manual - RAPID Instructions, Functions and Data types. Technical reference manual - System parameters, section TopicController -TypeModPos Settings.

2.2.2 FlexPendant Explorer

2.2.2 FlexPendant Explorer

FlexPendant Explorer

The FlexPendant Explorer is a file manager, similar to Windows Explorer, with which you can view the file system on the controller. You can also rename, delete, or move files or folders.

Illustration FlexPendant Explorer

The illustration details the FlexPendant Explorer.



en0400001130

A	Simple view. Tap to hide type in the file window.
В	Detailed view. Tap to show type in the file window.
С	Path. Displays folder paths.
D	Menu. Tap to display functions for file handling.
E	New folder. Tap to create a new folder in current folder.
F	Up one level. Tap to change to parent folder.
G	Refresh. Tap to refresh files and folders.

2.2.3 Inputs and Outputs, I/O

2.2.3 Inputs and Outputs, I/O

Inputs and outputs

Inputs and outputs, I/O, are signals used in the robot system. Signals are configured with system parameters.

Illustration Inputs and Outputs view

This illustration details the Inputs and Outputs view.

	Motors On Stepped (2 of 2		
	Stopped (2 of 2	Tada	
			strial Networks
Most Common	Active filter: Se)evices
Select an I/O Signal from the list.		All Si	gnals
Name	Value	Digita	al Inputs
		Digita	al Outputs
		Analo	og Inputs
		Analo	og Outputs
		Grou	os Input
		Grou	os Output
		✓ Most	Common
		Safet	y Signals
		Simu	ated Signals
		1	View
≥ 1/0			
- en0400000770			

What is a signal

An I/O signal is the logical software representation of:

- Inputs or outputs located on an industrial network I/O device that is connected to an industrial network within the robot system (real I/O signal).
- An I/O signal without a representation on any industrial network I/O device (virtual I/O signal).

By specifying an I/O signal, a logical representation of the real or virtual I/O signal is created. The I/O signal configuration defines the specific system parameters for the I/O signal that will control the behavior of the I/O signal.

2 Navigating and handling FlexPendant

2.2.4 Jogging

2.2.4 Jogging

Overview

The Jogging functions are found in the Jogging window. The most commonly used are also available under the Quickset menu.

Jogging menu

The illustration shows the functions available under the **Jogging** menu:

Manual MySystem (RSTEST4)		Motors On Stopped (2 of 2) (Speed 100%)	X X	X
Tap a property to change it		Position —		
Mechanical unit:	ROB_1	1:	0.0	o
Absolute accuracy:	Off	2: 3:	0.0 0.0	o o
Motion mode:	Axis 1 - 3	4:	0.0	o
Coordinate system:	World	5: 6:	0.0 0.0	0 0
Tool:	tool0			
Work object:	wobj0	Position Fo	Position Format	
Payload:	load0	2		
Joystick lock:	None		Joystick directions	
Increment:	None	2 1	3	
Align Go To Activate				
L Jogging				OB_1

en0400000654

Property/button	Function
Mechanical unit	Select mechanical unit active for jogging, described in section Selecting mechanical unit for jogging on page 108.
Absolute accuracy	Absolute Accuracy: Off is default. If the robot has the <i>Absolute Accuracy</i> option, then Absolute Accuracy: On is displayed.
Motion mode	Select motion mode, described in section <i>Selecting motion mode on page 110</i> .
Coordinate system	Select coordinate system, described in section <i>Selecting coordinate</i> system on page 114.
ΤοοΙ	Select tool, described in section <i>Selecting tool, work object, and payload on page 111.</i>
Work object	Select work object, described in section <i>Selecting tool, work object, and payload on page 111</i> .
Payload	Select payload, described in section <i>Selecting tool, work object, and payload on page 111</i> .
2.2.4 Jogging Continued

Property/button	Function
Joystick lock	Select locking joystick directions, described in section <i>Locking the joystick in specific directions on page 115</i> .
Increment	Select movement increments, described in section <i>Incremental movement for precise positioning on page 117</i> .
Position	Displays each axis position in relation to the selected coordinate system, described in section <i>Reading the exact position on page 119</i> .
	If the position values are displayed in red, then the revolution counters must be updated. See section <i>Updating revolution counters on page 280</i> .
Position format	Select position format, described in section <i>Reading the exact position on page 119</i> .
Joystick directions	Displays current joystick directions, depending on setting in Motion mode. See section <i>Selecting motion mode on page 110</i> .
Align	Align the current tool to a coordinate system. See section <i>Aligning tools on page 121</i> .
Go To	Move the robot to a selected position/target. See section <i>Moving the robot to a programmed position on page 257</i> .
Activate	Activate a mechanical unit.

2.2.5 Production Window

2.2.5 Production Window

Overview

The **Production Window** is used to view the program code while the program is running.

Illustration Production Window

This section illustrates the Production Window.

	Ma	nual Motors On /System (RSTEST4) Stopped (2 of 2) (Speed 100%)	33	X
Production Win	dow:	MyProgram1 in T_ROB1/MainModule/main		
STOP	4	CONST robtarget p30:=[[515.00,0	.00,	712.
	5	CONST robtarget p40:=[[515.00,0	.00,	712.
	6	CONST robtarget p50:=[[515.00,0	.00,	712.
T_ROB1	7	TASK PERS tooldata tool1:=[TRUE	,[[0	,0,0
	8	TASK PERS wobjdata wobj1:=[FALS	E, TR	UE , "
	9	PROC main()		
	10	MoveL p10, v1000, z50, tool0;		
	11	MoveL p20, v100, z15, tool0;		
	12	WaitDI DI1, 1;		
	13	MoveL p30, v100, z15, tool0;		
T_ROB2	14	WaitDI DI2, 1;		
	15	MoveL p40, v100, z15, tool0;		
	16	MoveL p10, v100, z15, tool0;		
<u> </u>	17	ENDPROC		
Load Program	PI	P to Main	Debu	ıg 🗖
Production Window				^{B_1}

Load Program	Load a new program.	
PP to Main	Move the program pointer to the routine Main.	
Debug	The Debug menu is available only in manual mode. Modify Position , se Modifying positions in the Program Editor or Production Window on page 247. Show Motion Pointerand Show Program Pointer, see About the Program and Motion Pointers on page 142. Edit Program, see Program Editor on page 41.	
	Note The Visual Step mode icon indicates that the Step Into mode is not selec- ted. This means that if the step mode is Step Out, Step Over, or Next Move the Visual Step mode icon is displayed.	

2.2.6 Program Data

2.2.6 Program Data

Overview

The **Program Data** view contains functions for viewing and working with data types and instances. You can open more than one window of the **Program Data**, which can be useful when working with many instances or data types.

Illustration of Program Data

This section illustrates the Program Data view.

Program Data - Us	Manual MySystem (RSTEST4) sed Data Types	Motors On Stopped (2 of 2) (Speed 100	∑ ∑ %)	X
Select a data type f	rom the list.			
Scope: RAPID/T_RO	B1		Change S	Scope
clock	loaddata	num	1	to 6 of 6
robtarget	tooldata	wobjdata		
		Show Data	Viev	v 🔶
Program Data				0B_1
en040000659				
Channa Caana Cha		a tha list, and Missuing data in		

Change Scope	Changes scope of data types in the list, see <i>Viewing data in specific tasks, modules, or routines on page 143</i> .
Show Data	Shows all instances of the selected data type.
View	Shows all or only used data types.

2 Navigating and handling FlexPendant

2.2.6 Program Data

Continued

Illustration of a data type instances

This section illustrates a list of instances for a data type.



Filter	Filters the instances, see <i>Filtering data on page 70</i> .
New	Creates a new instance of the selected data type, see <i>Creating new data instance on page 144</i> .
Edit	Edits the selected instances, see Editing data instances on page 146.
Refresh	Refreshes the list of instances.
View Data Types	Returns to the Program Data menu.

2.2.7 Program Editor

2.2.7 Program Editor

Overview

The **Program Editor** is where you create or modify programs. You can open more than one window of the **Program Editor**, which can be useful with the *Multitasking* option installed.

The **Program Editor** button in the task bar displays the name of the task.

Illustration of Program Editor

This section illustrates the Program Editor view.

	Manual Motors On Image: Comparison of the state of th
	Tasks and Programs 🔻 Modules 👻 Routines 👻
9	PROC main()
14	MoveL p10, v1000, z50, tool0;
11	MoveL p20, v100, z15, tool0;
12	WaitDI DI1, 1;
13	MoveL p30, v100, z15, tool0;
14	WaitDI DI2, 1;
15	MoveL p40, v100, z15, tool0;
16	MoveL p10, v100, z15, tool0;
17	ENDPROC
18	ENDMODULE
A Ir	dd A Edit Debug Modify Hide Position Declarations
9	T_ROB1 MainMod

en0400001143

Tasks and Programs	Menu for program operations, see <i>Handling of programs on page 126</i> .	
Modules	Lists all modules, see Handling of modules on page 129.	
Routines	Lists all routines, see Handling of routines on page 132.	
Add Instruction	Opens instruction menu, see <i>Handling of instructions on page 137</i> .	
Edit	Opens edit menu, see <i>Handling of instructions on page 137</i> .	
Debug	Functions for moving the program pointer, service routines see <i>Running a service routine on page 194</i> , and <i>About the Prog and Motion Pointers on page 142</i> .	
	 Functions to search routines and view system data. Search Routine: Searches all routines across all the modules (except hidden routines). 	
	View System Data: Displays all the tasks.	

Continues on next page

2 Navigating and handling FlexPendant

2.2.7 **Program Editor**

unucu		

Modify Position	See Modifying positions in the Program Editor or Production Window on page 247.
Hide Declarations	Hides declarations to make the program code easier to read.

Automatically activate mechanical unit for jogging

If *Multitasking* is installed with more than one mechanical unit and more than one motion task, then when switching between **Program Editor** windows the selection of mechanical unit for jogging is not effected. This means when jogging then the last used mechanical unit will move, which not neccessarily is the one used in the active **Program Editor**.

This setting can be changed with system parameters of the type *Automatically Switch Jog Unit* in the topic *Man-machine Communication*. Turn this setting on to automatically activate the mechanical unit last used in a **Program Editor** when switching to that window. This means that when jogging, the mechanical unit last used in the active **Program Editor** moves. Note that when switching between **Program Editors** in the same task, there is no change.

Mechanical units are manually activated for jogging in the **Jogging** window or in the Quickset menu, see *Selecting mechanical unit for jogging on page 108*.

2.2.8 Backup and Restore

2.2.8 Backup and Restore

About backups

The **Backup and Restore** menu is used for performing backups and restoring the system. See section *Backup and restore on page 271*.

Illustration of Backup and Restore

This is the Backup and Restore menu.



See Restore the system on page 275.

Restore System

2.2.9 Calibration

2.2.9 Calibration

About calibration

The **Calibration** menu is used to calibrate mechanical units in the robot system. Calibration can be performed using the option *Calibration Pendulum*. See *Operating manual - Calibration Pendulum*.

Illustration of Calibration menu

This illustration shows the **Calibration** menu. All mechanical units are listed and their calibration status is displayed in the **Status** column.

Calibration	Manual MySystem (RSTEST4)	Motors On Stopped (2 of 2) (Speed 100%)	27 27 27 27 27 27 27 27 27 27 27 27 27 2	X
In order to use the	he <mark>sys</mark> tem all mecha	anical units must be calibrate	d.	
Select the mechanic	al unit you want to cali	brate.		
Mechanical Unit	Status		1	to 2 of 2
ROB_1	Calibrated			
ROB_2	Calibrated			
Calibration				0B_1

Calibration menu options

This illustration shows the **Calibration** menu options after selecting mechanical unit.



2.2.10 Control Panel

2.2.10 Control Panel

Control Panel

The **Control Panel** contains functions for customizing the robot and the FlexPendant.

Illustration Control Panel

Hanual Controlle	Guard Stop 🛐 🔀
🗡 Control Panel	
Name	Comment 1 to 10 of 11
E Appearance	Customizes the display
Supervision	Motion Supervision and Execution Settings
₽ 1/0	Configures Most Common I/O signals
🐑 Language	Sets current language
B ProgKeys	Configures programmable keys
🔑 Controller Settings	Sets Network, Date and Time and ID
Touch Screen	Calibrates the touch screen
FlexPendant	Configures the FlexPendant system
Configuration	Configures system parameters
🔑 Installed Systems	Manages installed systems
Control Panel	

Appearance	Settings to customize the brightness of the display. See <i>Changing brightness and contrast on page 86</i> .	
Supervision	Settings for motion supervision and execution settings. See Using motion supervision and non motion execution on page 229. When changing the collision detection level, it is necessary to go to motors off state to activate the changes.	
I/O	See Configuring Most Common I/O on page 91.	
Language	See Changing language on page 92.	
ProgKeys	See Changing programmable keys on page 93.	
Controller Settings	Settings for configuring the network, the date and time for the robot controller, and the ID of the controller. See <i>Controller settings on page 89</i> .	
Touch Screen	See Calibrating the touchscreen on page 95.	
FlexPendant	Configuration of views for operating mode switch and UAS, User Authorization System. See <i>Defining a view to be shown during</i> <i>operating mode change or startup on page 78</i> .	
Configuration	Configuration of the system parameters configuration.	

2.2.11 Event Log

2.2.11 Event Log

The Event Log

The event log stores information about past events for future reference to facilitate troubleshooting.

How to open the event log is described in Accessing the event log on page 267.

Illustration Event Log

	Manual MySystem (RSTEST4)	Motors On The second se	
Event Log - Con	nmon		
Tap a message	to open it.		
Code	Title	Date & Time 1 to 9 of 197	
10002	Program pointer has been re	set 2014-08-06 11:17:19	
10011	Motors ON state	2014-08-06 11:14:47	
10010	Motors OFF state	2014-08-06 11:14:46	
10002	Program pointer has been reset 2014-08-06 11:14:08		
10002	Program pointer has been re	set 2014-08-06 11:11:19	
10015	10015 Manual mode selected 2014-08-06 11:10:55		
10012	Safety guard stop state	2014-08-06 11:10:55	
10011	Motors ON state	2014-08-06- 11;10;50	
10017	Automatic mode confirmed	2014-08-06 11:10:49	
Save All Log As	s Delete Update	View	
Program Data	₿ I/O		

xx0300000447

Function	Description	
View a message	Tap the message. The message structure is described in <i>An</i> event log message on page 48.	
Scroll or zoom a message	See Scrolling and zooming on page 69.	
Delete the log	See Deleting log entries on page 268.	
Save the log	See Saving log entries on page 269.	
Close the log	See Accessing the event log on page 267.	

2.2.11 Event Log

Continued

An event log message Each event log entry consists of a message describing the event in detail, and it often contains advice on how to solve the problem. Event Log - Event Message (A) C 2014-08-06 11:17:19 Event Message 10002 i Program pointer has been reset B Description D The program pointer of task T_ROB1 has been reset. Consequences (E) When started, program execution will start on the first instruction of the task's entry routine. NOTE that the manipulator may move to unexpected position when restarted! Probable causes F The operator has probably requested this action manually. (н) ок G) Next Previous en0300000454

Α	Event number. All errors are listed by numbers.
В	Event title. Briefly states what has happened.
С	Event time marker. Specifies exactly when the event occurred.
D	Description. A brief description of the event. Intended to assist in understanding the causes and implications of the event.
E	Consequences. A brief description of any consequences inflicted on the system, transition to other operation mode, emergency stop, caused by the particular event. Intended to assist in understanding the causes and implications of the event.
F	Probable causes. A list of probable causes, listed in order of probability.
G	Recommended actions. A list of the recommended correcting actions, based on the Probable causes specified above. These may range from <i>Replace the xx</i> to <i>Run test program xx</i> , that i, may be actions to isolate the problem as well as fixing it.
Н	Acknowledge or OK button.

Related information about logs

Event log messages and more information about the event log are described in *Operating manual - Troubleshooting IRC5*.

2.2.12 System Info

2.2.12 System Info

About System Info

System Info displays information about the controller and the loaded system. Here you can find the RobotWare version and options currently in use, current keys for control and drive modules, network connections, and so on.

Illustration of System Info view



en0400000968

Controller properties

It contains controller and network information. When the **Controller properties** is expanded the following are visible:

Network connections	Service port and Local Area Network properties.
Installed systems	List of installed systems.

System properties

It contains information of the system that is currently in use. When the **System properties** is expanded the following are visible:

Control module	Name and key of the Control Module.	
Options	Installed RobotWare options and languages.	
Drive modules	Lists all Drive Modules.	
Drive module x	Name and key of Drive Module x.	
Options	Drive Module x options, with type of robot and so on.	
Additional options	Any RobotWare options and Process application options.	

Continues on next page

2 Navigating and handling FlexPendant

2.2.12 System Info

Continued

Hardware devices

It contains information of all the hardware attached. When the **Hardware devices** is expanded the following are visible:

Controller	Name and key of the Control Module.		
Computer system	Contains information of the main computer.		
Power supply system	Contains information of the Power supply unit.		
Panel board	Provides information about the panel board hardware and soft- ware.		
Drive module x	It contains information about the axis computer, drive unit, and contactor board.		
Mechanical units	Lists the robots or external axes that are connected to the con- troller.		
General SIS data	Lists the service information system data for the connected mechanical units.		
	For more information regarding the SIS data, see Operating manual - Service Information System.		

Software resources

It contains information about the RAPID. When the **Software resources** is expanded the following are visible:

System	Contains information about system uptime and memory.	
RAPID	Software used by the controller.	
RAPID memory	Memory allocated for RAPID programs.	
RAPID performance	Shows the execution load.	
Services	Contains information about additional services.	
Communication	Contains information about Remote Service Embedded.	

2.2.13 Restart

2.2.13 Restart

Restart

A running system normally does not need to be restarted.

Tap the ABB menu and then tap Restart to restart the system.

© Restart	Manual MySystem (RSTEST4)	Motors On Stopped (2 of 2) (Speed 100%)	žž	X
a	The controller will be nd any changed syst activated	restarted. The state is saved em parameter settings will be after the restart.		
	This operation	cannot be undone.		
Advanced		Restart		
Restart)B_1

en0500001557

Under certain scenarios, advanced restart options might be required. For details about advanced restart, see the section **Restart Procedures** in *Operating manual - IRC5 Integrator's guide*.

2.2.14 Log Off

2.2.14 Log Off

The Log Off menu		
-	This section details the Log Off menu. More about using this me in <i>Logging on and off on page 75</i> .	enu is described
	Log Off is available under the ABB menu.	
	🖋 User Authorization System	
	User Authorization System	
	To login as other than Default User, choose user and enter password.	
	User:	
	Default User 🗸	
	Password:	
	ABC	
	Default User	Login
	r	

2.3 Operator window

2.3 Operator window

Operator window

The operator window displays messages from the program. With *Multitasking* installed, all tasks' messages are displayed in the same operator window. If a message requires action then a separate window for that task will be displayed. The operator window is opened by tapping the icon to the right of the ABB logo in the status bar. The illustration shows an example of an operator window:

	Manual MySystem (RSTES	Ma T4) St	otors On opped (2 of 2) (Speed 100%)	₹¥	
All Tasks					
	Don't Show	Don't Show			
Clear	Logs	Ta sk N ame			
Program Data					ò

Clear	Clears all messages
Don't Show Logs	Clears all messages
Don't Show Task Name	Hides task names

2.4 Status bar

2.4 Status bar

Illustration of status bar

The Status bar displays information about the current status, such as operational mode, system, and active mechanical unit.

A B Manual	Motors On D F Z Z
MySystem (RSTEST4) C	(E)Stopped (2 of 2) (Speed 100%)

Α	Operator window
в	Operating mode
С	System name (and controller name)
D	Controller state
Е	Program state
F	Mechanical units. The selected unit (and any unit coordinated with the selected) is marked with a border. Active units are displayed in color, while deactivated units are grey.

2.5.1 The Quickset menu

2.5 Quickset

2.5.1 The Quickset menu

Quickset menu

The QuickSet menu provides a quicker way to change among other things jog properties rather than using the **Jogging**view.

Each button on the menu shows the currently selected property value or setting. In manual mode, the Quickset menu button shows the currently selected mechanical unit, motion mode, and increment size.

Illustration of the Quickset menu

This section describes the buttons in the Quickset menu.



Α	Mechanical unit, see Quickset menu, Mechanical unit on page 56.
в	Increment, see Quickset menu, Increment on page 61.
С	Run Mode, see Quickset menu, Run Mode on page 62.
D	Step Mode, see Quickset menu, Step Mode on page 63.
E	Speed, see Quickset menu, Speed on page 64.
F	Tasks, see <i>Quickset menu, Tasks on page 65</i> .

2.5.2 Quickset menu, Mechanical unit

2.5.2 Quickset menu, Mechanical unit

Illustration Mechanical unit button

On the Quickset menu, tap Mechanical unit, then tap to select a mechanical unit.





•	
А	
В	Mechanical unit, a selected unit is highlighted. See <i>Selecting mechanical unit for jogging on page 108</i> .
С	Motion mode settings (axes 1-3 motion mode currently selected), further settings are described in <i>Illustration Motion mode settings on page 57</i> .
D	Tool settings (tool 0 currently selected), further settings are described in <i>Illustra-</i> <i>tion Tool settings on page 58</i> .
E	Work object settings (work object 0 currently selected), further settings are de- scribed in <i>Illustration Work object settings on page 58</i> .
F	Coordinate system settings (world coordinate currently selected), further settings are described in <i>Illustration Coordinate system settings on page 59</i> .
G	Turn coordination off, further settings are described in <i>Turn coordination off on page 60</i> .
н	Show details, further settings are described in <i>Illustration Show Details on</i> page 60.



The Mechanical unit menu is only available in manual mode.

2.5.2 Quickset menu, Mechanical unit Continued

Illustration Motion mode settings

To view or change any motion mode functionality, tap the **Motion mode** settings button. These settings are also available in the Jogging window, see *Selecting motion mode on page 110*.



en0300000540

Select motion mode setting:

- Axis 1-3
- Axis 4-6
- Linear
- Reorient

2 Navigating and handling FlexPendant

2.5.2 Quickset menu, Mechanical unit *Continued*

Illustration Tool settings

To view or change the available tools, tap the **Tool** settings button. These settings are also available in the Jogging window, see *Selecting tool, work object, and payload on page 111*.



Illustration Work object settings

To view or change the available work objects, tap the **Work object** settings button. These settings are also available in the Jogging window, see *Selecting tool, work object, and payload on page 111*.



en0400000989



2 Navigating and handling FlexPendant

2.5.2 Quickset menu, Mechanical unit Continued

Select a work object to use.

Illustration Coordinate system settings

To view or change coordinate system functionality, tap the **Coordinate system** settings button. These settings are also available in the Jogging window, see *Selecting coordinate system on page 114*.



en0300000541

Select a coordinate system setting:

- · World coordinate system
- Base coordinate system
- Tool coordinate system
- · Work object coordinate system

2.5.2 Quickset menu, Mechanical unit *Continued*

Illustration Show Details

Tap Show Details to display the settings available for a mechanical unit.



en0500002354

Α	Override jog speed settings (100% currently selected)
в	Coordinate system settings
с	Motion mode settings
D	Turn on or off user increment
E	Turn on or off jog supervision

If any of the settings are not available, they will crossed over.

Motion mode and coordinate settings can be changed by tapping the buttons.

Tap Hide Details to return to the basic display.

Turn coordination off

To quickly change between coordinated and uncoordinated jogging, use the coordination off button.

The button is automatically hidden when you change anything that effects coordination, for example the work object or the coordinate system of the coordinated mechanical unit.

To re-enable the button you must setup coordination again manually.

For information about coordination between MultiMove robots, see *Application manual - MultiMove*.

2.5.3 Quickset menu, Increment

2.5.3 Quickset menu, Increment

Increment settings

The increment settings are also available in the Jogging window. For more details, see *Incremental movement for precise positioning on page 117*.

Illustration Increment

	Manual MarkerFunctio	ins ()	Guard Stop Stopped (2 of 2) (Spee	d 100%)	2
- Mechanical unit:	ROB_1		Increments		\mathbb{Z}
Increments	Value		\bigcirc		E <u>r</u>
1/3 Axis	0.00573	(deg)	0	None	
c 🛴 Linear	0.05000	(mm)	\bigcirc		
C 😡 Reorient	0.02865	(deg)		Small	1 D
G Select an item to m Angle unit: Degrees G	nodify the sta	ep size.	Hide Values	Medium Large User >>	
Program Editor Elitor	Program 2 1/0 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2				
None	1	No increment	S		
		.			

None	No increments
Small	Small movements
Medium	Medium movement
Large	Large movements
User	User defined movements
Show/Hide Values	Displays/hides increment values
Angle unit	Defines the unit for the angles.



The Increment menu is available only in manual mode.

٦

Continuous

2.5.4 Quickset menu, Run Mode

2.5.4 Quickset menu, Run Mode

Run mode

By setting run mode you define if the program execution should run once and then stop, or run continuously.

For information about run mode in:

- *Multitasking*, see *Application manual Controller software IRC5*, section *Multitasking*.
- *MultiMove*, see Application manual *MultiMove*, section User interface specific for Multimove.

Illustration Run mode

	Manual Test ()	Guard Stop Stopped (Speed 100%)	2
Robot	Ware	Run Mode	
en030000472			ROB_1 1/3
Single Cycle	Runs one cycle then sto	ops execution.	

Runs continuously.

2.5.5 Quickset menu, Step Mode

4

2.5.5 Quickset menu, Step Mode

Step mode

By setting the step mode you can define how the step-by-step program execution should function. For more details, see *Stepping instruction by instruction on page 191*

Illustration Step mode



en0300000543

Step Into	Steps into called routines and executes them step-by-step.
Step Out	Executes the remains of the current routine and then stops at the next in- struction in the routine from which the current routine was called. Not possible to use in the Main routine.
Step Over	Called routines are executed in one single step.
Next Move	Steps to the next move instruction. Stops before and after movement in- structions, for example, to modify positions.



If you select the Step Out/Step Over/Next Move step modes, an arrow icon



xx1500001580

) is also displayed in the **QuickSet** menu indicating that the selected mode is not a **Step Into** step mode.

2 Navigating and handling FlexPendant

2.5.6 Quickset menu, Speed

2.5.6 Quickset menu, Speed

Speed button

The speed settings apply to the current operating mode. However, if you decrease the speed in automatic mode, the setting also applies to manual mode if you change mode.

Illustration Speed

Tap the **Speed** button to view or change the speed settings. The current running speed, in relation to max, is displayed above the buttons.



-1%	Decrease running speed in steps of 1%
+1%	Increase running speed in steps of 1%
-5%	Decrease running speed in steps of 5%
+5%	Increase running speed in steps of 5%
0%	Set speed to 0%
25%	Run at quarter speed (25%)
50%	Run at half speed (50%)
100%	Run at full speed (100%)

2.5.7 Quickset menu, Tasks

2.5.7 Quickset menu, Tasks

Tasks button	
	If you have the option <i>Multitasking</i> installed there can be more than one task. Otherwise there is only one task.
	By default, you can activate/deactivate only Normal tasks from the Quickset menu. However, from the Control Panel , you can change the settings so that All tasks can be activated/deactivated.
	Activated tasks are started and stopped with the Start and Stop buttons on the FlexPendant.
	The tasks settings are only valid for manual operating mode.
Related information	
	Application manual - Controller software IRC5, section Multitasking.
	How to start and stop multitasking programs is described in section <i>Using multitasking programs on page 227</i> .
	<i>TrustLevel</i> for tasks are set with system parameters, see section <i>Task</i> in <i>Technical reference manual - System parameters</i> .
	You can define if all tasks or only normal tasks should be displayed. See section <i>Defining which tasks should be selectable in the tasks panel on page 84</i> .

2.6.1 Using the soft keyboard

2.6 Basic procedures

2.6.1 Using the soft keyboard

Soft keyboard

The soft keyboard is used frequently when operating the system, for example when entering file names or parameter values.

The soft keyboard works as an ordinary keyboard with which you can place the insertion point, type and correct typing errors. Tap letters, numbers and special characters to enter your text or values.

Illustration soft keyboard

This illustration shows the soft keyboard on the FlexPendant.

\equiv \vee		Manual MySystem (RSTEST4)	Motors On Stopped (2 of 2) (Speed 100%) 3 ¥	X
New Folder Na	ame				
1	2	3 [#] 4 ^{\$} 5 [%] 6 ^{&}	7 [{] 8 ⁽ 9 ⁾ 0 ^{} -}	=	\bigotimes
	q	w e r t	y u i o p	[]
САР	а	s d f g	h j k l ;	•	+
Shift	Z	x c v b	n m , · · / ·	Home	
Int'l	•	\~	→ ↓ ←	→ E	End
			ок	Canc	el
Explorer					DB_1
en0300000491					

Using international characters

All western characters can be used, also in usernames and passwords. To access international characters, tap the **Int'l** button on the soft keyboard.

2.6.1 Using the soft keyboard Continued

Changing the insertion point

Tap the arrow keys to change the insertion point, for instance when correcting typing errors.

If you need to move	then tap
backward	xx0300000492
forward	xx0300000493

Deleting

1 Tap the **Backspace** key (top right) to delete characters to the left of the insertion point.



xx0300000494

2.6.2 Messages on the FlexPendant

2.6.2 Messages on the FlexPendant

Overview of messad	les
	The FlexPendant displays messages from the system. These can be status messages, error messages, program messages, or requests for action from the user. Some require actions, and some are plain information.
Event log messages	
	The event log messages are messages from the RobotWare system about system status, events, or errors.
	How to work with the event log messages is described in section <i>Handling the event log on page 267</i> . All messages are also described in <i>Operating manual</i> - <i>Troubleshooting IRC5</i> .
System messages	
	Some messages sent out by the system are not from the event log. They can come from other applications, such as RobotStudio.
	To be able to change configurations and settings in the system from RobotStudio, the user must request write access. This generates a message on the FlexPendant where the operator can grant or deny access. The operator can at any time decide to withdraw the write access.
	How to request access and work with RobotStudio is described in <i>Operating manual - RobotStudio</i> .
Program messages	

RAPID programs can send out messages to the Operator window, see section *Operator window on page 53*.

How to generate program messages is described in *Technical reference* manual - *RAPID Instructions, Functions and Data types*.

2.6.3 Scrolling and zooming

2.6.3 Scrolling and zooming

Overview

The entire contents of a screen might not be visible at the same time. To see the entire contents, you can:

- Scroll up/down (and sometimes left/right)
- Zoom in or out (only available in the Program Editor)

		\checkmark	(j) Z	Mar MyS	ual ystem (RSTEST	4)	1	Motors C Stopped)n (2 of 2) (9	5peed 100)%)	X X	X
Ŀ	МуРі	rogram	1 in	T_ROB	1/Mair	nModul	e/main							
	Та	sks an	d Pro	grams	•		Mod	ules		•	F	louti	nes	•
7	ĸ	PEI	RS	too	olda	ata	too	L1:	=[Т	RUE	, [ˌ []	<u>7</u> 2	,01	, 11,
8	ĸ	PEE	RS	wok	ojda	ata	wob	j1:	=[F	ALSI	c, Th	ŪE		, [[)
9	С	mai	in ()							A)	B	Ô
14	Mo	veL	р	10,	v1	000	, z5	0,	to	510;				
11	501	<mark>የ</mark> ት	p2	20,	v1 ()0,	z15	, t	001	0;			Ē	
12	lai	t D	ĹΓ)11	, 1;	;							E	
13	[01	7eL	pЗ	80,	v1 (00,	z15	, t	001	0;	~			
14	lai	tD	C D	12	, 1;	;					F		G	9
15	ro]	7eL	p4	0,	v1 (00,	z15	, t	001	0;		2	\checkmark	
16	101	7eL	p 1	.0,	v1 (00,	z15	, t	001	0;			\checkmark	\checkmark
A	dd	uction			Edit		De	bug		Modify		Hi	ide	tions
	ISUL	uction						_		POSILIO	n	0	естага	uons
5	T_RO Main	B1 Mod												0B_1

Α	Zoom in (larger text)
в	Scroll up (the height of one <i>page</i>)
С	Scroll up (the height of one <i>line</i>)
D	Scroll left
E	Scroll right
F	Zoom out (smaller text)
G	Scroll down (the height of one <i>page</i>)
Н	Scroll down (the height of one <i>line</i>)

2.6.4 Filtering data

2.6.4 Filtering data

Filtering data

In several of the FlexPendant menus you can use filtering. This can be useful when you are looking at instances of a data type, for example, and there are more available than is possible to overlook. By filtering instances starting with a specific character for example, the number can be greatly reduced.

Depending on the type of data, you can filter data either alphabetically or numerically.

Illustration of filtering

The filter function is switched on until the active filter is removed (For example, by tapping **Reset**).

	anual ySystem (RSTES	5T4)	Motors On Stopped (2 of 2) (Speed 100%)					33	X
	el	Active	filter:						
Select the data you wa	ant to edit.	10000				-			
Scope: RAPID/T_ROB1			ABC						
Name	Value	Active							
p10	[[515,0		Α	В	С	D	Ε	F	G
p20	[[515,0		н	т	1	K		м	N
p30	[[515,0			-				8-8	
p40	[[515,0		0	Ρ	Q	R	S	Т	U
p50	[[515,0		V	W	X	Y	Ζ		\bigotimes
		Clear	F	leset		123 >	•	Filte	er
	New	Edi	t	•	Refr	esh	V T	iew D ypes	ata
Program Data								1	

en0500001539

Note

When filtering I/O signals there are more options than for many other types of data. For example,

- you can filter data by Name or Category.
- filtering function can be displayed automatically if the number of signals displayed exceeds a predefined number. See *Illustration of automatic filter display on page 72*.

2.6.4 Filtering data Continued

Inputs and Outputs Manual MySystem (RSTE	ST4)	Moi Sto	tors On pped (2	2 of 2) (Speed 1	00%)	₹¥	X	
Digital Outputs	Active	filter: [001	Selec	t Layo	out			
Select an I/O Signal from the list.	Name		(Category			Settings		
DO1	Active	D01							
		Α	В	С	D	Е	F	G	
		Н	I	J	K	L	Μ	Ν	
		0	Ρ	Q	R	S	Т	U	
		V	W	X	Y	Ζ		$\langle \times \rangle$	
	Clear	R	eset		123 >		Filte	er	
							Vie	ew 📩	
Program Data							1/2		

en1200000669

Figure 2.1:

Active filter	Displays the current filter. It is also displayed at the top of the list of items.
Clear	Clears the text within the Active filter text box.
Reset	Removes the filter string.
123 / ABC	Depending on type of data, there may be one or several ways to filter data, e.g. numeric, alphabetic.
Filter	Applies the filter.
en1100000506	Opens and closes the Active filter menu.

2.6.4 Filtering data *Continued*

Illustration of automatic filter display

The I/O signal filter can be set to be displayed automatically if the number of data exceeds a predefined number.

Manual MySystem (RSTE:	Motor ST4) Stopp	rs On ed (2 of 2) (5	peed 100%	/o)	×
Most Common	Active filter:	Select	Layout		
Select an I/O Signal from the list.	Name	Catego	ry	Sett	tings
Name					
		7	8	9	←
	100				
	Max value: 2048	4	5	6	\rightarrow
	Min value: 0	1	2	3	$<\!\!\times\!\!$
		0			
				Do	ne
	•			v	iew 🕈
Program Data ₽ 1/0				1	

	Action
1.	Tap Change to edit the setting controlling when the filter dialog should appear.
2.	Enter a new number defining the upper limit for not using the filter. Then tap Done .
3.	Tap Virtuals to select if all, or only virtual, or only non virtual signals should be listed.
2.6.5 Process applications

2.6.5 Process applications

Process applications

Custom process applications are started from the ABB menu. Each application is listed as a menu item together with the FlexPendant views.

Start a process application

Use this procedure to start a process application.

	Action
1	Tap the ABB button to display the ABB menu. Process applications are listed in the menu.
2	Tap the name of the process application to start.

Switch between started process applications

A started application has a quick-button in the taskbar, just like FlexPendant views. Tap the buttons to switch between the started applications and views.



en0400000768

The views and process application running in this case are:

Α	FlexPendant Explorer view
В	Program Editor view
С	RobotWare Arc, a process application
D	Control Panel view

2.6.6 Granting access for RobotStudio

2.6.6 Granting access for RobotStudio

About write access on the controller

The controller accepts *only* one user with write access at a time. Users in RobotStudio can request write access to the system. If the system is running in manual mode, the request is accepted or rejected on the FlexPendant.

Granting access for RobotStudio

This procedure describes how to grant access for RobotStudio.

	Action
1	When a user in RobotStudio requests access, a message is displayed on the FlexPend- ant. Decide whether to grant or reject access. If you want to grant access, then tap Grant . The user holds write access until he dis- connects or until you reject the access. If you want to deny access, then tap Deny .
2	If you have granted access and want to revoke the access, tap Deny.

2.6.7 Logging on and off

2.6.7 Logging on and off

Logout procedure	Use	this procedure to log off the system.	
		Action	
	1	On the ABB menu, tap Log Off.	
	2	Tap Yesto confirm.	
Login procedure	Use	this procedure to log on to the controller, using the User Autho	orization System,
	UA	S. UAS can limit the available functions for users.	
	Afte	r a log off, the Login window is displayed automatically.	
	~ (Jser Authorization System	
	U	ser Authorization System	
	То	login as other than Default User, choose user and enter password.	
		User:	
		Default User	
		Password:	
		ABC	
	D	efault User	Login

en0400000947

	Action	Information
1	Tap on the User menu and select a user. If there are more than seven users then the menu is replaced with a button.	If you select Default User , then no pass- word is required, and you are logged on automatically.
2	If the user you have chosen has a pass- word you must use the soft keyboard to enter password. Tap ABC to display the soft keyboard. Enter the password and tap OK .	
3	TapLogin.	

2.6.7 Logging on and off *Continued*

Handling users and authorization levels

Read more on how to add users or set the authorization in *Operating manual* - *RobotStudio*.

How to edit what views or functions are hidden for certain users is described in *Defining a view to be shown during operating mode change or startup on page 78.*

2.7.1.1 Setting default paths

2.7 Changing FlexPendant settings

2.7.1 System settings

2.7.1.1 Setting default paths

Introduction to default paths

You can set individual default paths for some actions using the FlexPendant. The following default paths can be set:

- Saving and loading RAPID programs.
- Saving and loading RAPID modules.
- Saving and storing configuration files.

This function is available if the user that is logged on is authorized. User authorization is handled via RobotStudio. See *Operating manual - RobotStudio*.

Setting default paths

Use this procedure to set a default path.

	Action
1	On the ABB menu, tap Control Panel and then FlexPendant.
2	On the ABB menu, tap Control Panel and then FlexPendant. Tap File System Default Path. Image: Manual MySystem (RSTEST4) MySystem (RSTEST4) Stopped (2 of 2) (Speed 100%) Control Panel - System - File System Default Path Here you define default paths to the controller file system. Select file type and browse to the folder to be used as default path. Then press OK. File type: RAPID programs Default path:
	Press clear to remove path. (No default path will be used.) Clear OK Cancel Control Panel en0500002361
3	 Tap the File type menu to choose type of default path: RAPID programs RAPID modules Configurations files
4	Type the default path or tap Browse, to choose the desired location.
5	If required, any previously entered path can be removed by tapping Clear.
6	Тар ОК.

2.7.1.2 Defining a view to be shown during operating mode change or startup

2.7.1.2 Defining a view to be shown during operating mode change or startup

View on operating mode change or startup

This function can be used, for example, if a view other than the **Production Window** is desired when switching to Auto mode.

Defining view on operating mode change

Use the following procedure to configure the FlexPendant to automatically show a specified view during a change in operating mode or during startup.

	Action
1	On the ABB menu, tap Control Panel and then tap FlexPendant.
2	Tap View on System Event. The View on System Event window is displayed. Image: Star of the View of System Event Image: Star of the View of System Event Image: Star of the View automatically shown at event changes. Select the event and browse for the view you prefer. Then press OK.
	System Event: Switching to Auto
	OK Cancel
3	 Tap the System Event list and select a type of system event. Following are the available types: Switching to Auto: Select this option to define the FlexPendant view when switching to auto mode. Switching to Manual: Select this option to define the FlexPendant view when switching to manual mode. Switching to Manual Full Speed: Select this option to define the FlexPendant view when switching to manual full speed mode. FlexPendant Startup: Select this option to define the view when the FlexPendant starts.
4	In the Selected view field, tap and select a desired application to be displayed during the selected system event.
5	Tap OK. The changes are saved.
ſ	Note

The **Clear View** button will remove the currently selected view if you do not want any view to be automatically shown.

2.7.1.3 Changing the background image

2.7.1.3 Changing the background image

Background images

The background image on the FlexPendant can be changed. Any image file on the controller hard disk can be used, a photo as well as an illustration.

For best result, use an image following these recommendations:

- 640 by 390 pixels (width, height)
- Format gif

Changing background image

Use this procedure to change background image on FlexPendant.

	Action
1	On the ABB menu, tap Control panel.
2	Tap FlexPendant and then Background Image. Image: Manual Motors On Stopped (2 of 2) (Speed 100%) Image: MySystem (RSTEST4) Image: Motors On Stopped (2 of 2) (Speed 100%) Image: Image:
	Default
	Browse Default OK Cancel
	Control Panel
	en0500001547
3	Tap Browse to locate another picture on the controller hard disk.
4	Tap Default to restore the original background image.
5	Тар ОК.

2.7.1.4 Defining visibility level for UAS protected functions

2.7.1.4 Defining visibility level for UAS protected functions

Introduction to visibility levels

This section describes how to define visibility level for functions protected by the user authorization system, UAS. The protected functions can be hidden or displayed but not accessible. All other administration of the user authorization system is done using RobotStudio. See *Operating manual - RobotStudio*.

Defining visibility level for UAS protected functions

	Action				
1	On the ABB menu, tap Control Panel and then tap FlexPendant.				
2	Tap User Authorization System. Image: Manual MySystem (RSTEST4) MySystem (RSTEST4) Stopped (2 of 2) (Speed 100%) Control Panel - System - User Authorization System Here you define system behaviour for functionality protected by the User Authorization System. Choose visibility level for protected functions. Then press OK. Hide non accessible functions Show message when trying to access protected functions				
	OK Cancel				
3	 Tap to select the level of visibility for UAS protected functions: Hide non accessible functions OR Show message when trying to access protected functions. 				
4	Тар ОК.				

2.7.1.5 Defining an additional test view

2.7.1.5 Defining an additional test view

Overview

If your system has a customized operator interface, that is one or several applications developed with FlexPendant SDK, it is possible to enable the user to start program execution in manual mode from such an application. If there is no such application, however, the screen for adding other test views will look as in the illustration below.

Defining an additional test view

Use this procedure to define an additional test view.

	Action
1	On the ABB menu, tap Control Panel and then FlexPendant.
2	Tap Additional Test View . The displayed screen might look like this:
	Manual Motors On MySystem (RSTEST4) Stopped (2 of 2) (Speed 100%)
	Control Panel - System - Additional Test View
	Define additional views to be allowed when starting a RAPID program in manual mode.
	Check the additional test views you want to use. Then press OK.
	Available Test Views
	OK Cancel
	Control Panel
	en0600003110
3	Usually only the Program Editor and the Production Window are allowed test views. In case there are additional views to choose from, these will appear in the list. Check one or several applications to be used as additional test views.
4	Тар ОК.

2.7.1.6 Defining position programming rule

2.7.1.6 Defining position programming rule

Introduction to position naming

Robot positions in a RAPID program are either named variables or not named (using the asterisk character, *). The programmer can choose which naming rule the FlexPendant should use when new move instructions are programmed.

Defining position programming rule

Use this procedure to define a naming rule for new robot positions.

	Action
1	On the ABB menu, tap Control Panel and then FlexPendant.
2	Tap Position Programming Rule. Image: Manual Mysight (RSTEST4) Mysight (RSTEST4) Stopped (2 of 2) (Speed 100%) Image: Control Panel - System - Position Programming Rule Here you define the programming rule for robot positions. Select RAPID rule when programming move instructions in the Program Editor, Press OK. Image: New position created; * or sequential naming rule Next sequential already existing position selected No new position created; * naming rule
3	OK Cancel
4	Тар ОК.

Position programming rules

This section gives a detailed description of the options available when programming robot positions, here referred to as *targets*. This signifies the position to which the mechanical unit is programmed to move.

New targets can be named according to any of these principles:

- New position created; * or sequential naming rule.
- Next sequential already existing position selected.
- No new position created; * naming rule.

New position created; * or sequential naming rule

This is the default setting. When a move instruction is programmed, a new target will automatically be created. If the last target was named, that is *not* using an "*", the new target will be named in sequence with the previous one.

2.7.1.6 Defining position programming rule *Continued*

For example: MoveJ p10 will be followed by MoveJ p20, unless this target already exists in the program. In such a case, MoveJ p30 (or the next free number) will be used instead.

Next sequential already existing position selected

When a move instruction is programmed, no new target will be created. Instead, the next target in a sequence created beforehand will be selected. The very first target, however, will be an "*", as no sequence yet exists. As soon as the first target has been defined this rule will be applied.

For example: A number of targets have been predefined; p10 to p50. In such a case, MoveJ p10 will be followed by MoveJ p20. The next instruction will use target p30, etc. until p50 is reached. Since no further targets have been defined, p50 will be used for the following targets as well.

No new position created; * naming rule

When a Move instruction is programmed, no new target will be created. Instead, an "*" will always be used. This can be replaced by an existing target at a later stage.

For example: MoveJ pl0 will be followed by MoveJ *.

2.7.1.7 Defining which tasks should be selectable in the tasks panel

2.7.1.7 Defining which tasks should be selectable in the tasks panel

Tasks panel

The tasks panel is found in the Quickset menu. See *Quickset menu, Tasks on page 65*.



To simplify debugging of background tasks you can make all tasks (including the background tasks) visible in the task panel on the FlexPendant. Then, in manual mode, all tasks that are selected in the task panel (including background tasks) will stop when pressing the stop button.

Defining which tasks to show

Use this procedure to define which tasks should be selectable in the tasks panel in the Quickset menu.

	Action
1	On the ABB menu, tap Control Panel and then tap FlexPendant.
2	Tap Task Panel Settings.
3	Select Only Normal tasks or All tasks . When All tasks is selected, then all tasks that are selected in the task panel (including background tasks) will stop when pressing the stop button. The selected background tasks will be treated as if the value of the system parameter <i>Trustlevel</i> is defined as <i>NoSafety</i> .
4	Тар ОК.

2.7.1.8 Managing the display of controller and system name

Overview

You can manage the display of controller and system name using this feature.

Controller and System name

Use the following procedure to manage the display of controller and system name in the status bar.

	Action
1	On the ABB menu, tap Control Panel and then tap FlexPendant . The FlexPendant Configuration Properties window is displayed.
2	Navigate and tap Controller and System Name. The Controller and System Name window is displayed. Manual Guard Stop To_MOD (6.05_Build80) Stopped (Speed 100%) Control Panel - FlexPendant - Controller and System Name Here you define Status bar behaviour for displaying Controller and System name. Select the Controller and System naming option. Then press OK. Controller name only. (Default) System name only. Both Controller and System name.
	OK Cancel
	Control Panel 1/3 3
3	 Select a controller and system naming option according to your requirement. Following are the available options and its description: Controller name only. (Default): Displays only the name of the controller in the status bar. This is the default option. System name only: Displays only the name of the system in the status bar. Both Controller and System name: Displays both the controller and system name in the status bar.
4	Tap OK. The changes are applied and displayed in the status bar
5	Tap Close.
L	

2.7.2.1 Changing brightness and contrast

2.7.2 Basic settings

2.7.2.1 Changing brightness and contrast

Appearance options

This section describes the **Appearance** menu, where you can adjust the screen's brightness and contrast. The contrast can only be adjusted on FlexPendant without USB port.

Changing brightness and contrast

Use this procedure to change brightness and contrast of the screen.

	Action
1	On the ABB menu, tap Control Panel.
2	Tap Appearance.
3	Tap the Plus or Minus button to adjust the levels. Tap Set Default to return to default. The brightness and contrast changes as you change the levels which gives you an instant view of how the new levels will affect the visibility.
4	Tap OK to use the new brightness and contrast levels.

Note

If you change brightness or contrast from the default levels, some screens can appear to be striped. This is however not a sign of a faulty screen. Change back to default settings to avoid the striped appearance.

2.7.2.2 Adapting the FlexPendant for left-handed users

Overview	Left-handed users, normally prefers thei FlexPendant can easily be adapted to sui can easily rotate the screen through 180 d the device.	r left-hand for the touch screen. The t the needs of the left-handed users. They egrees and use their right-hand to support
Illustration	The FlexPendant operated by a right-hand person at the bottom. On FlexPendant with the emergency sto unit, note the location of the emergency	ded person at the top and by a left-handed p button located on the outer edge of the button when the display is rotated for
	left-hand users. FlexPendant with emergency stop button at the connector	FlexPendant with emergency stop button at the outer edge
	xx1800000045	en0400000913

Rotating the FlexPendant screen

Use this procedure to adapt the FlexPendant to suit a left-handed user.

	Action
1	Tap the ABB menu, then tap Control Panel.
2	Tap Appearance.

2.7.2.2 Adapting the FlexPendant for left-handed users *Continued*

	Action
3	Tap Rotate right. Image: Control Panel - Appearance Motors On Stopped (2 of 2) (Speed 100%) Image: Control Panel - Appearance Image: Control Panel - Appearance Brightness Image: Control Panel - Appearance
	Set Default Rotate Right OK Cancel
4	Rotate the FlexPendant and move it to the other hand.

What is affected?

The following settings are affected when adapting the FlexPendant for a left-handed user.

Setting	Effect	Information
Jogging directions	The joystick directions are adjusted automatically.	The illustrations of jogging directions in the jogging menu are adjusted automatically.
Hardware buttons and pro- grammable keys	No change. Note Forward and Backward but- tons do not change place with each other.	See Hard buttons on page 19.
Emergency stop	No change.	
Three-position enabling device	No change.	

2.7.2.3 Controller settings

2.7.2.3 Controller settings

Date and time settings

Use the following procedure to configure date and time.

Step	Action
1	On the ABB menu, tap Control Panel.
2	Tap Controller Settings. The Date and Time settings window is displayed.
3	In the Settings section, select Network Time or Manual Time.
	Select Network Time for configuring the robot controller for automatic time synchronization using NTP protocol of a time server. The time server is identified by its IP address or DNS name. Select Manual Time if you do not have a time server that is reachable from the controller.
4	Select the required time zone from the Time Zone section.
5	In the Date and Time section, tap the + (Plus) or - (Minus) button and configure the date and time.
6	Tap OK. The selected settings are saved.



The date and time are displayed according to the ISO standard. That is, date is in the year-month-day format and time is in the hour:minute format (24-hour format).

Network settings

Use the following procedure to configure the network.

Step	Action
1	On the ABB menu, tap Control Panel.
2	Tap Controller Settings.
3	Navigate to the bottom menu, tap Settings, and select Network. The Control Panel - Controller Settings - Network window is displayed.
4	Configure the network settings according to your requirements.
5	Tap OK. The selected settings are saved.

Identity settings

Use the following procedure to configure the identity of the controller.

Step	Action
1	On the ABB menu, tap Control Panel.
2	Tap Controller Settings.
3	Navigate to the bottom menu, tap Settings, and select Identity. The Control Panel - Controller Settings - Identity window is displayed.
4	Edit the name of the controller in the Controller Name field according to your requirement.

Continues on next page

2.7.2.3 Controller settings *Continued*

Step	Action
5	Tap OK. The name of the controller is saved.

2.7.2.4 Configuring Most Common I/O

2.7.2.4 Configuring Most Common I/O

Most Common I/O

Most Common I/O is used in the **Program Editor** to display a list of the most commonly used I/O signals in the robot system. Since there can be many signals, it can be helpful to use this selection.

The sorting in the list can be rearranged manually. By default, the signals are sorted in the order that they are created.

Most Common I/O can also be configured using system parameters in the topic *Man-machine Communication*. However, sorting the list can only be done by using the function under the Control Panel.

Configuring Most Common I/O

Use this procedure to configure the list Most Common I/O.

	Action
1	On the ABB menu, tap Control Panel.
2	Tap I/O. A list of all I/O signals defined in the system is listed with check boxes.
3	Tap the names of the signals to select for the Most Common I/O list. Tap All or None to select all or no signals. Tap Name or Type to sort by name or signal type.
4	 Tap Preview to see the list of selected signals and adjust the sort order. Tap to select a signal and then tap the arrows to move the signal up or down in the list, rearranging the sort order. Tap APPLY to save the sort order. Tap Edit to return to the list of all signals.
5	Tap APPLY to save the settings.

2.7.2.5 Changing language

2.7.2.5 Changing language

Languages

The FlexPendant is installed with and supports twenty different languages. By default, the current language is English.

Switching from one installed languages to another is easy. For more information, see *Changing language on page 92*.



When you switch to another language, all buttons, menus and dialogs will use the new language. RAPID instructions, variables, system parameters, and I/O signals are not affected.

Changing language

Use this procedure to change language on the FlexPendant.

	Action
1	On the ABB menu, tap Control Panel.
2	Tap Language. A list of all installed languages is displayed.
3	Tap the language that you want to change to.
4	Tap OK . A dialog box is displayed. Tap Yes to proceed and restart the FlexPendant. The current language is replaced by the selected one.

2.7.2.6 Changing programmable keys

2.7.2.6 Changing programmable keys

Overview

Programmable keys are four hardware buttons on the FlexPendant that can be used for dedicated, specific functions set by the user. See *Hard buttons on page 19*.

The keys can be programmed to simplify programming or testing of programs. They can also be used to activate menus on the FlexPendant.

Change programmable keys

Use the following procedure to set the programmable keys:

	Action	
1	On the ABB menu, tap Control Panel.	
2	Tap ProgKeys. Manual Mysystem (IN-L-GISBT14178) Motors On Stopped (2 of 2) (Speed 100%) © Control Panel - ProgKeys © Key 3 © Key 4 None © Key 1 © Key 2 © Key 3 © Key 4 None © Key 4 None Type: Digital outputs: Output © Key pressed: D01 Toggle D03 soActLogs soActLogs soActLogs soAwCycleOn	
	OK Cancel	
3	Select key to set, Key 1-4 in the upper selection list.	
4	Tap the Type menu to select type of action: None Input Output System 	
5	 If Type Input is selected. Tap to select one of the digital inputs from the list. Tap the Allow in auto menu to select if the function is also allowed in automa operating mode. Note An input cannot be set by using the programmable keys, its value can only be puls The pulse will be the inverted value of the input. 	

2.7.2.6 Changing programmable keys *Continued*

	Action
6	 If Type Output is selected. Tap to select one of the digital outputs from the list Tap the Key pressed menu to define how the signal should behave when the key is pressed. Tap the Allow in auto menu to select if the function is also allowed in automatic operating mode Key pressed functions: Toggle - switches signal value from 0 to 1 or vice versa Set to 1 - sets the signal to 1 Set to 0 - sets the signal to 0 Press/Release - sets signal value to 1 while key is pressed (note that an inverted signal will be set to 0)
7	 If Type System is selected. Tap the Key pressed menu to select Move PP to main Tap the Allow in auto menu to select if the function is also allowed in automatic operating mode
8	Set the other keys as described in steps 3 to 7 above.
9	Tap OK to save the settings.

2.7.2.7 Calibrating the touchscreen

2.7.2.7 Calibrating the touchscreen

When to calibrate

The touchscreen is calibrated on delivery and normally never needs to be calibrated.

On FlexPendant with the emergency stop button located at the connector, it is not possible to calibrate the touchscreen.

On FlexPendant with the emergency stop button located on the outer edge of the unit, the touchscreen can be calibrated. Depending on the model of the FlexPendant, the appearance of the symbols will vary, the function is however the same.



en0400000974

Calibrating the touchscreen

Use this procedure to calibrate the touchscreen.

	Action	Information
1	On the ABB menu, tap Control Panel.	
2	Tap Touch Screen.	
3	Tap Recalibrate . The screen will go blank for a few seconds. A series of symbols will appear on the screen, one at a time.	
4	Tap the center of each symbol with a pointed object.	CAUTION Do not use a sharp object which can damage the surface of the screen.
5	The recalibration is complete.	

2.7.2.7 Calibrating the touchscreen *Continued*

About the touch calibration function

The touch calibration function waits on each calibration point for a couple of touch coordinates or that the touch will be released. Then the average of the collected coordinates will be calculated and the symbol moves to the next position.

The touch controller only sends new coordinates to the CPU when the coordinates are changing. If you touch the symbol very accurately with a stylus, the touch coordinates will not change. Then the touch controller sends only one coordinate and the touch calibration function is waiting endlessly for more coordinates.

The best way to avoid this problem is to tap the symbol for only one second and then release.

3 Jogging

3.1 Introduction to jogging

What is jogging?	To jog is to manually position or move robots or external axes.		
When can I jog?	Υοι	u can jog in manual mode but you cann	ot jog during program execution.
Jogging the robot This procedure details the main steps for jogging the robot.		ogging the robot.	
		Action	Information
	1	 It is possible to jog the robot under the following conditions: The system has been started as detailed in this manual. No programmed operation is running The system is in Manual mode. The three-position enabling device is pressed and the system is in Motors On state. 	The Manual mode is described in section <i>About the manual mode on page 185</i> . Starting in the Manual mode is detailed in section <i>Starting programs on page 223</i> . How to switch to manual mode is detailed in section <i>Switching from automatic to manual mode on page 244</i> .
	2	Determine in which way you want to jog.	The difference between different types of jogging is detailed in section <i>Introduction to jogging on page 97</i> . How to select coordinate system is detailed in section <i>Selecting coordinate system on page 114</i> .
	3	Select a mechanical unit. The axes can be jogged in different ways.	How to jog the robot axis by axis is detailed in section <i>Jog axis by axis on page 113</i> .
	4	Define the working range for the robot/ro- bots as well as for any other pieces of	The robot's working range is defined by system parameters. See <i>Technical refer</i> -

	equipment working in the robot cell.	ence manual - System parameters.	
5	Jog the manipulator using the joystick on the FlexPendant.	The FlexPendant and its various parts and sections are described in section <i>The FlexPendant on page 17</i> .	
		The joystick and how to map the directions of it, is detailed in section <i>Selecting motion mode on page 110</i> .	
		How to prevent causing manipulator movements in certain directions while jog- ging, is detailed in section <i>Locking the</i> <i>joystick in specific directions on page 115</i> .	
		There might be restrictions to how you can jog, see section <i>Restrictions to jogging on page 106</i> .	
6	In some cases, more than one manipulator can be jogged simultaneously. This re- quires the option <i>MultiMove</i> .	How to jog multiple manipulators is de- tailed in section <i>Coordinated jogging on</i> <i>page 107</i> .	

3 Jogging

3.1 Introduction to jogging *Continued*

About motion modes and robots

The selected motion mode and/or coordinate system determines the way the robot moves.

In linear motion mode the tool center point moves along straight lines in space, in a "move from point A to point B" fashion. The tool center point moves in the direction of the selected coordinate system's axes.

Axis-by-axis mode moves one robot axis at a time. It is then hard to predict how the tool center point moves.

About motion modes and additional axes

Additional axes can only be jogged axis-by-axis. An additional axis can either be designed for some kind of linear motion or for rotational (angular) motion. Linear motion is used in conveyers, rotational motion in many kinds of workpiece handlers.

Additional axes are not affected by the selected coordinate system.

About coordinate systems

Positioning a pin in a hole with a gripper tool can be performed very easily in the tool coordinate system, if one of the coordinates in that system is parallel to the hole. Performing the same task in the base coordinate system may require jogging in both x, y, and z coordinates, making precision much more difficult.

To select the proper coordinate systems to jog in will make jogging easier but there are no simple or single answers to which coordinate system to choose.

A certain coordinate system will make it possible to move the tool center point to the target position with fewer joystick moves than another.

Conditions such as space limitations, obstacles or the physical size of a work object or tool will also guide you to the proper judgement.

Read more about coordinate systems in section *Coordinate systems for jogging on page 99*.

3.2 Coordinate systems for jogging

Coordinate systems

A coordinate system defines a plane or space by axes from a fixed point called the origin. Robot targets and positions are located by measurements along the axes of coordinate systems.

A robot uses several coordinate systems, each suitable for specific types of jogging or programming.

- The *base coordinate system* is located at the base of the robot. It is the easiest one for just moving the robot from one position to another.
- The *work object coordinate system* is related to the work piece and is often the best one for programming the robot.
- The *tool coordinate system* defines the position of the tool the robot uses when reaching the programmed targets.
- The *world coordinate system* that defines the robot cell, all other coordinate systems are related to the world coordinate system, either directly or indirectly. It is useful for jogging, general movements and for handling stations and cells with several robots or robots moved by external axes.
- The user coordinate system is useful for representing equipment that holds other coordinate systems, like work objects.

Default settings	
-	If you change coordinate system in the jogging properties, this will automatically be reset to default settings after a restart.
Linear mode	
	For each mechanical unit the system will by default use the base coordinate system for the linear motion mode.
Reorient mode	
	For each mechanical unit the system will by default use the tool coordinate system for the reorientation motion mode.

3 Jogging

3.2 Coordinate systems for jogging *Continued*

Illustration of axes and joystick directions

The axes of a generic six axis manipulator can be jogged manually using the joystick. Please check your plant or cell documentation to determine the physical orientation of any additional axes.

The illustration shows the movement patterns for each manipulator axis.



xx0300000520

3.2 Coordinate systems for jogging *Continued*



xx0300000495

The base coordinate system has its zero point in the base of the robot, which makes movements predictable for fixed mounted robots. It is therefore useful for jogging a robot from one position to another. For programming a robot, other coordinate systems, like the work object coordinate system are often better choices. See *The work object coordinate system on page 102* for more information.

When you are standing in front of the robot and jog in the base coordinate system, in a normally configured robot system, pulling the joystick towards you will move the robot along the X axis, while moving the joystick to the sides will move the robot along the Y axis. Twisting the joystick will move the robot along the Z axis.

3 Jogging

3.2 Coordinate systems for jogging *Continued*



xx0600002738

Α	World coordinate system
В	Work Object coordinate system 1
С	Work Object coordinate system 2

The work object coordinate system corresponds to the work piece: It defines the placement of the work piece in relation to the world coordinate system (or any other coordinate system).

The work object coordinate system must be defined in two frames, the user frame (related to the world frame) and the object frame (related to the user frame).

A robot can have several work object coordinate systems, either for representing different work pieces or several copies of the same work piece at different locations.

It is in work object coordinate systems you create targets and paths when programming the robot. This gives a lot of advantages:

- When repositioning the work piece in the station you just change the position of the work object coordinate system and all paths are updated at once.
- Enables work on work pieces moved by external axes or conveyor tracks, since the entire work object with its paths can be moved.

Examples of use

For example, you are determining the positions of a number of holes to be drilled along the edge of the work object.

You are creating a weld between two walls in a box.

3.2 Coordinate systems for jogging *Continued*



en0300000497

The tool coordinate system has its zero position at the center point of the tool. It thereby defines the position and orientation of the tool. The tool coordinate system is often abbreviated TCPF (Tool Center Point Frame) and the center of the tool coordinate system is abbreviated TCP (Tool Center Point).

It is the TCP the robot moves to the programmed positions, when executing programs. This means that if you change the tool (and the tool coordinate system) the robot's movements will be changed so that the new TCP will reach the target.

All robots have a predefined tool coordinate system, called tool0, located at the wrist of the robot. One or many new tool coordinate systems can then defined as offsets from tool0.

When jogging a robot the tool coordinate system is useful when you don't want to change the orientation of the tool during the movement, for instance moving a saw blade without bending it.

Examples of use

Use the tool coordinate system when you need to program or adjust operations for threading, drilling, milling or sawing.

3 Jogging

3.2 Coordinate systems for jogging *Continued*

The world coordinate system



en0300000496

А	Base coordinate system for robot 1
в	World coordinate
С	Base coordinate system for robot 2

The world coordinate system has its zero point on a fixed position in the cell or station. This makes it useful for handling several robots or robots moved by external axes.

By default the world coordinate system coincides with the base coordinate system.

Examples of use

For example, you have two robots, one floor mounted and one inverted. The base coordinate system for the inverted robot would be upside down.

If you jog in the base coordinate system for the inverted robot, movements will be very difficult to predict. Choose the shared world coordinate system instead.

3.3 Joystick directions

3.3 Joystick directions

Introduction to joystick directions

The Joystick Directions area shows how joystick axes correspond to the selected coordinate system's axes.



CAUTION

The Directions properties is not intended to show the direction in which the mechanical unit will move. Always try jogging with small joystick movements so that you learn the true directions of the mechanical unit.

Joystick directions

The significance of the joystick directions depends on the selected motion mode.

Motion mode	Illustration joystick	Description
Linear	en0400001131	Linear mode is described in sec- tion Setting the tool orientation on page 112.
Axis 1, 2, and 3 (default for robots)	Joystick directions 2 1 3 en0300000536	Axis 1-3 mode is described in section <i>Jog axis by axis on page 113</i> .
Axis 4, 5, and 6	Joystick directions ♀ ↔ ↔ 5 4 6 en0300000537	Axis 4-6 mode is described in section <i>Jog axis by axis on page 113</i> .
Reorient	en0400001131	Reorient mode is described in section <i>Setting the tool orientation</i> on page 112.

3.4 Restrictions to jogging

3.4 Restrictions to jogging

Jog additional axes

Additional axes can only be jogged axis-by-axis. Please see *Application manual - Additional axes and standalone controller*.

Jog mechanical units that are not calibrated

If the mechanical unit is not calibrated the text **Unit not calibrated** will be displayed in the **Position** area of the **Jogging** window.

An uncalibrated mechanical unit can only be jogged axis-by-axis. Its working range will not be checked.

When the robot is not calibrated, incremental movement is restricted to one step per joystick deflection. A calibrated robot performs 10 steps/sec when deflecting the joystick.



Mechanical units whose working range is not controlled by the robot system can be moved to dangerous positions. Mechanical stops should be used and configured to avoid danger to equipment or personnel.

Jog robot axes in independent mode

It is not possible to jog axes in independent mode. You need to return the axes to normal mode in order to jog. Please see *Application manual - Controller software IRC5* for details.

Jog while using world zones

With the option *World Zones* installed, defined zones will restrict motion while you jog. Please see *Application manual - Controller software IRC5* for details.

Jog with axis loads not set

If equipment is mounted on any of the robot axes, then axes loads must be set. Otherwise overload errors might occur when jogging.

How to set axis loads are described in the Product Manuals delivered with your robot.

Jog with tool or payload weights not set

If the weight of tools and payloads is not set, then overload errors might occur when jogging. Loads for additional axes controlled by specific software (dynamic models) can only be set in programing.

3.5 Coordinated jogging

3.5 Coordinated jogging

Coordination

A robot that is coordinated to a work object will follow the movements of that work object.

Coordinated jogging

If the mechanical unit moving the work object is jogged, any robot that is currently coordinated with the work object will move so that it maintains its relative position to the work object.

Set up coordination

	Action	Information
1	Select the robot that is to be coordinated to another mechanical unit.	See Selecting mechanical unit for jogging on page 108.
2	Set Coordinate system to Work Object.	See Selecting coordinate system on page 114.
3	Set Work object to the work object moved by the other mechanical unit.	See Selecting tool, work object, and pay- load on page 111.
4	Select the mechanical unit that moves the work object.	Any jogging, while this mechanical unit is selected, will also affect the robot that is coordinated with it.

Coordinating robots

Coordinating robots, so that when jogging one robot another robot will follow, requires the option *MultiMove*. See *Application manual - MultiMove*.

3.6.1 Selecting mechanical unit for jogging

3.6 Basic settings for jogging

3.6.1 Selecting mechanical unit for jogging

Jogging properties

If your system has more than one robot, that is additional robots or additional axes, then you need to select which mechanical unit to jog when using the joystick.

There are three ways to select mechanical unit:

- Using the Select mechanical unit button.
- Using the **Jogging** window on the **ABB** menu.
- Using the Quickset menu Mechanical unit, see Quickset menu, Mechanical unit on page 56.

Any changes you make to jogging properties only affects the currently selected mechanical unit.

All jogging properties are saved and restored when you return to jog that mechanical unit.

Identifying mechanical units

Each mechanical unit that can be jogged is represented in the mechanical units list. The name of the unit is defined in the system configuration. Each unit also has a symbol that is used in the Status bar, see section *Status bar on page 54*.

In manual mode, the Quickset menu button displays which mechanical unit is selected.

Please consult your plant or cell documentation to see which mechanical units are available in your robot system.

Selecting mechanical unit using the hard button

Press the **Select mechanical unit** button to change unit. One press on the button changes to the next mechanical unit, as steps in a cycle.

Selecting mechanical unit in the Jogging window

Use this procedure to select a mechanical unit to jog in the **Jogging** window.

Action

1 On the **ABB** menu, tap **Jogging**.
3.6.1 Selecting mechanical unit for jogging *Continued*

	Action				
2	Tap Mechanical U	Jnit.			
		Manual MySystem (RSTEST4)	Motors On Stopped (2 of 2) (Speed 100%)	₹¥	X
	온 Jogging - Mechani	cal Unit			
	Current selectio	n: ROB_1	s.		
	Mechanical Unit		Status	1	L to 2 of 2
	S ROB_1		Activated		
	S ROB_2		Activated		
			ОК	Cano	el
	L Jogging				OB_1
	en040000653				
3	Tap the mechanic The selected mec the Jogging wind	cal unit to be jogged, hanical unit is active ow.	, and then tap OK. e until you select another ur	nit, ever	n if you clo

Examples of use

Your robot system may consist of more than a single robot. There can also be other mechanical units such as workpiece handlers or additional axes mounted on the robot that can also be jogged.

Related information

If the system uses *Multitasking*, and has more than one motion task, and uses more than one mechanical unit, then the selected mechanical unit can be switched automatically when switching between **Program Editor** windows. See section *Program Editor on page 41*.

Mechanical units can be activated or deactivated with the **Activate** function in the **Jogging** window.

3.6.2 Selecting motion mode

3.6.2 Selecting motion mode

Motion mode

There are three ways to select motion mode:

- 1 Using the **Toggle motion mode** button.
- 2 Using the **Jogging** window on the **ABB** menu.
- 3 Using the Quickset menu Mechanical unit, see Quickset menu, Mechanical unit on page 56.

Selecting motion mode using the toggle button Press the Toggle motion mode reorient/linear button to switch motion mode.

Selecting motion mode in the Jogging window

Use this procedure to select motion mode in the Jogging window.

	Action	Information
1	On the ABB menu, tapJogging.	
2	Tap Motion mode.	
3	Tap on the mode you want and then tap OK.	The significance of the joystick directions are shown in Joystick direction after making the selection.

Related information

Joystick directions on page 105.

3.6.3 Selecting tool, work object, and payload

3.6.3 Selecting tool, work object, and payload

Overview

It is always important to choose the proper tool, work object, or payload. It is absolutely vital when you create a program by jogging to the target positions. Failing to do so will most likely result in overload errors and/or incorrect positioning either when you jog or when you run the program in production.

Selecting tool, work object, and payload

	Action
1	On the ABB menu, choose Jogging to view jogging properties.
2	Tap Tool , Work object , or Payload to display the lists of available tools, work objects or payloads.
3	Tap the tool, work object, or payload of choice followed by OK.

3.6.4 Setting the tool orientation

3.6.4 Setting the tool orientation

Examples of use

Tools for arc welding, grinding and dispensing must be oriented in a particular angle to the work piece to obtain the best result. You also need to set up the angle for drilling, milling or sawing.

In most cases you set the tool orientation when you have jogged the tool center point to a specific position such as the starting point for a tool operation. After you have set the tool orientation you continue to jog in linear motion to complete the path and the supposed operation.

Definition of tool orientation

The tool orientation is relative to the currently selected coordinate system. From a user perspective however this is not noticeable.

Setting the tool orientation

ollowed by OK . er tool by following the procedure in <i>Selecting</i>
ollowed by OK. er tool by following the procedure in <i>Selecting</i>
er tool by following the procedure in <i>Selecting</i>
ling device to activate the mechanical unit's ation changes.



Use the QuickSet menu to select jogging mode faster.

3.6.5 Jog axis by axis

Jogging axis by axis

There are three ways to select axis for jogging.

- Using the Toggle motion mode axis group button.
- Using the **Jogging** window on the **ABB** menu.
- Using the Quickset menu Mechanical unit, see Quickset menu, Mechanical unit on page 56.

In manual mode, the Quickset menu button displays which axis group is selected. How to use the joystick when jogging axis by axis is displayed in the **Joystick directions** area. See *Illustration of axes and joystick directions on page 100*.

Examples of use

Use axis by axis jogging when you need to:

- Move the mechanical unit out of a hazardous position.
- Move robot axes out of singularities.
- Position axes for fine calibration.

Selecting axis group using the toggle button

Press the Toggle motion mode axis group button to switch motion mode.

Selecting axis group in the Jogging window

Use this procedure to select axis group in the Jogging window.

	Action
1	On the ABB menu, tap Jogging.
2	Tap Motion Mode.
3	Tap on the axis group 1-3 or 4-6 and then tap OK.

The orientation of any mounted tool will be affected by this procedure. If the resulting orientation is important, perform the procedure described in *Setting the tool orientation on page 112* when finished.

3.6.6 Selecting coordinate system

3.6.6 Selecting coordinate system

Coordinate systems for jogging

The coordinate system most suitable for your jogging depends on many things. See section *Coordinate systems for jogging on page 99*, for more information.

There are two ways to select coordinate system:

- Using the Jogging window on the ABB menu.
- Using the Quickset menu Mechanical unit, see Quickset menu, Mechanical unit on page 56.

Prerequisites

Select motion mode suitable for the intended jogging.

Stationary tools in the tool coordinate system

If your robot system uses stationary tools, you must select both the proper tool and the proper work object (held by the robot) to jog in tool coordinates.

The tool coordinate system is defined by the position and orientation of the stationary tool and is fixed in space. To perform the intended operations you move the work object. This way positions can be expressed in the tool coordinate system.

Selecting coordinate system

Use this procedure to select coordinate system in the Jogging window.

	Action
1	On the ABB menu, tap Jogging .
2	Tap Coordinate System.
3	Tap to select a coordinate system.
4	Тар ОК.

3.6.7 Locking the joystick in specific directions

3.6.7 Locking the joystick in specific directions

Overview

The joystick can be locked in specific directions to prevent movement for one or more axes.

This may be useful for instance while fine tuning positions or when programming operations that should only be performed in the direction of a specific coordinate system axis.

Note that the axes locked depends on the currently selected motion mode.

Which axes are locked?

This section describes how to see which joystick directions are locked

	Action
1	On the ABB menu, tap Jogging to view jogging properties.
2	Tap Joystick lock to check the joystick properties, or check the Joystick directions area properties in the right hand corner of the window. A padlock symbol is displayed for locked axes.

Locking the joystick in specific directions

This section describes how to lock the joystick in specific directions.

	Action
1	On the ABB menu, tap Jogging .
2	Tap Joystick lock.
	Manual Motors On
	∠ Jogging - Joystick Lock
	Current selection: None Select which joystick axis to lock.
	None Horizontal Vertical Rotation
	OK Cancel
	₽ Jogging en0300000488
3	Tap the joystick axis or axes that should be locked. The axis toggles between locked and unlocked each time you tap.
4	Tap OK to lock.

3 Jogging

3.6.7 Locking the joystick in specific directions *Continued*

Unlocking all axes

This section describes how to unlock all axes from the joystick directions lock.

	Action
1	On the ABB menu, tap Jogging .
2	Tap Joystick lock.
3	Tap None, then tap OK.

3.6.8 Incremental movement for precise positioning

Incremental movement

Use incremental movement to jog the robot in small steps, which enables very precise positioning.

This means that each time the joystick is deflected, the robot moves one step (increment). If the joystick is deflected for one or more seconds, a sequence of steps, (at a rate of 10 steps per second), will be performed as long as the joystick is deflected.

Default mode is no increment, then the robot moves continuously when the joystick is deflected.

There are three ways to select the increment size:

- Using the Toggle increments button.
- Using thee **Jogging** window on the **ABB** menu.
- Using the Quickset menu Increments, see Quickset menu, Increment on page 61.

To use the toggle button you must first select an increment size in the **Jogging** window or **Quickset** menu.

Selecting increments using the toggle button

Press the **Toggle increments** button to switch increment size, you toggle between no increments and the increment size you previously selected in the **Jogging** window.

Selecting increments in the Jogging window

Use this procedure to select the incremental movement size using the **Jogging** window.

Action

1 On the **ABB** menu, tap **Jogging**.

3 Jogging

3.6.8 Incremental movement for precise positioning *Continued*

	Action
2	Tap Increment.
	Image: Manual Motors On Manual MySystem (RSTEST4) Motors On MySystem (RSTEST4) Image: Manual MySystem (RSTEST4) Motors On MySystem (RSTEST4) Image: Manual
	윤 Jogging - Increment
	Current selection: None
	Select incremental mode.
	None Small Medium Large User
	OK Cancel
	2 Jogging
3	Tap the desired increment mode, see description in section <i>Incremental movement</i> sizes on page 118.
4	Тар ОК.

Incremental movement sizes

Choose between small, medium or large increments. You can also define your own increment movement sizes.

Increment	Distance	Angular
Small	0.05 mm	0.005°
Medium	1 mm	0.02°
Large	5 mm	0.2°
User		

3.6.9 Reading the exact position

About positions and revolution counters

The exact position of the robot is determined using the position of the resolvers and counters that count the number of resolver revolutions. These are called revolution counters.

If the robot is correctly calibrated then the current position is automatically calculated at start.



CAUTION

If the positions are displayed in red text then the values from the revolution counters are lost and instead the values stored on the SMB are displayed. Be careful when jogging the robot if the values are displayed in red text. Watch the robot closely and do not use the displayed values! If the mechanical unit is uncalibrated then the actual position can be very different from the position values stored by the SMB. You must update the revolution counters before a program can be started. See Updating revolution counters on page 280.



Note

If no positions are displayed then the mechanical unit is uncalibrated. Instead the text Selected mechanical unit is not calibrated is displayed.



When updating the revolution counters, the ongoing RAPID instruction or function is interrupted, and the path is cleared.

How robot positions are displayed

Positions are always displayed as:

- The point in space expressed in the x, y, and z tool center point coordinates.
- The angular rotation of the tool center point expressed in Euler angles or as a quaternion.

How additional axes' positions are displayed

When an additional axis is moved, only the axis position is displayed.

Linear axis positions are displayed in millimeters expressed as the distance to the calibration position.

Rotating axis positions are displayed in degrees expressed as the angle to the calibration position.

Reading the exact position

This procedure describes how to read the exact position.

Action

1 On the ABB menu tap .Jogging.

119

3 Jogging

3.6.9 Reading the exact position *Continued*

Action

The position is displayed in the **Position** area properties in the right hand side of the window.
 See illustration in *Jogging on page 36*.

Position format

The position can be displayed in different formats. Tap **Position Format** to change settings.

The **Position** can be displayed relative the following frames:

- World
- Base
- Work object

The Orientation format can be set to:

- Quaternion
- Euler angles

The Position angle format can be set to:

Angles

The Presentation angle unit can be set to:

- Degrees
- Radians

3.6.10 Aligning tools

3.6.10 Aligning tools

Overview

A tool can be aligned with another coordinate system.

When aligning a tool, the tool's z axis is aligned to the selected coordinate system's nearest axis. Therefore it is recommended to first jog the tool so it is close to the desired coordinates.

Note that the tool's data is not changed!

to start aligning the tool.

Tap Close when completed.

This procedure describes how to align tools.

Aligning mechanical units

Action 1 On the ABB menu, tap Jogging.

	Manual MySystem (RSTEST4)	Motors On Stopped (2 of 2) (Speed 100%)	X		
은 Jogging - Align					
Current tool:	tool0				
1. Select coord	inate system to alig	n the currently selected tool to	0:		
Coord:	World	World			
2. Press the enabling device then tap and hold 'Start Align'.					
		Start Align			
3. When ready	tap 'Close'.	Start Align			
3. When ready	tap 'Close'.	Start Align	Close		
3. When ready	tap 'Close'.	Start Align	Close		
3. When ready	tap 'Close'.	Start Align	Close		

5

This page is intentionally left blank

4.1 Before you start programming

Programming tools

You can use both the FlexPendant and RobotStudio for programming. The FlexPendant is best suited for modifying programs, such as positions and paths, while RobotStudio is preferred for more complex programming. How to program using RobotStudio is described in *Operating manual - RobotStudio*.

Define tools, payloads, and work objects

Define tools, payloads and work objects before you start programming. You can always go back and define more objects later, but you should define your basic objects in advance.



It is important to always define the actual tool load and, when used, the payload of the robot (for example, a gripped part). Incorrect definitions of load data can result in overloading of the robot mechanical structure. There is also a risk that the speed in manual reduced speed mode can be exceeded.

When incorrect load data is specified, it can often lead to the following consequences:

- The robot may not use its maximum capacity.
- Impaired path accuracy including a risk of overshooting.
- · Risk of overloading the mechanical structure.

The controller continuously monitors the load and writes an event log if the load is higher than expected. This event log is saved and logged in the controller memory.

Define coordinate systems

Make sure the base and world coordinate systems have been set up properly during the installation of your robot system. Also make sure that additional axes have been set up.

Define tool and work object coordinate systems before you start programming. As you add more objects later you also need to define the corresponding coordinate systems.



For more details about the RAPID language and structure, see *Technical reference* manual - RAPID Overview and *Technical reference manual* - RAPID Instructions, *Functions and Data types*.

4.2 Using RAPID programs

4.2 Using RAPID programs

Using the RAPID program

This procedure describes the main steps required in creating, saving, editing and debugging any RAPID program.

Note that there is more information available, than the one referred to in the procedure.

	Action	Information
1	Start by creating a RAPID program.	How to create a RAPID program is detailed in section <i>Handling of programs on page 126</i> .
2	Edit your program.	Proceed as detailed in section <i>Handling of instructions on page 137</i> .
3	To simplify programming and keep an overview of the program, you may want to divide the program into more than one module.	How to view, add, or delete a module is detailed in section <i>Handling of modules on page 129</i> .
4	To further simplify programming, you may want to divide the module into more than one routine.	How to add or delete a routine is detailed in section <i>Handling of routines on page 132</i> .
5	 When programming you may want to work with: Tools Work objects Payloads 	 Also read the following sections: Creating a tool on page 154. Creating a work object on page 169. Creating a payload on page 177.
6	In order to deal with potential errors that may occur during program execution, you may want to create an error handler.	Error handlers are described in the <i>RAPID</i> manuals.
7	After completing the actual RAPID pro- gram, it will require testing before being put into production.	Proceed as detailed in section <i>Testing on page 183</i> .
8	After test running your RAPID program, it may require altering. You may want to modify, or tune, programmed positions, the TCP positions, or paths.	How to modify positions while the program is running is described in section <i>HotEdit</i> <i>menu on page 32</i> . How to modify positions in manual mode is described in section <i>Modifying positions</i> <i>in the Program Editor or Production Win-</i> <i>dow on page 247</i> .
9	Programs that are no longer required may be removed.	

Running the program

This procedure specifies how to use an existing RAPID program.

	Action	Information
1	Load an existing program.	Described in section <i>Starting programs on page 223</i> .
2	When starting program execution, you may choose between running the program once, or running it continuously.	Described in section <i>Quickset menu, Run</i> <i>Mode on page 62</i> .

4.2 Using RAPID programs Continued

	Action	Information
3	Once the program has been loaded, you may start program execution.	Described in section <i>Starting programs on page 223</i> and in <i>Using multitasking pro-grams on page 227</i> .
4	After program execution is completed, the program may be stopped.	Proceed as detailed in section <i>Stopping</i> programs on page 226.

4.3.1 Handling of programs

4.3 Programming concept

4.3.1 Handling of programs

Overview

This section details how to perform normal handling of robot programs. It describes how to:

- create a new program
- load an existing program
- save a program
- rename a program
- delete a program

Each task must contain *one* program, no more, no less. Note that the following procedures describe a single task system, i.e. only one task is available.

How to create a new program *when no program is available* is detailed in section *Creating a new program on page 126*.

About program files

When saving a program to the controller hard disk, it is by default saved to the directory HOME in the system's folder unless otherwise stated. How to set another default path is detailed in section *Setting default paths on page* 77.

The program is saved as a folder, named as the program, containing the actual program file, of type pgf.

When loading a program you open the program folder and select the pgf file.

When renaming a program you rename the program folder and the program file.

When saving a loaded program which is already saved to the hard disk, you must not open the existing program folder. Instead, you should save the program folder again and overwrite the old version, or rename the program.

Creating a new program

This section describes how to create a new program.

	Action	
1	On the ABB menu, tap Program Editor.	
2	Tap Tasks and Programs.	
3	 3 Tap File, then New Program. 3 If there was already a program loaded, a warning dialog appears. • Tap Save to save the loaded program. • Tap Don't Save to close loaded program without saving it, i.e. delete from p gram memory. • Tap Cancel to leave the program loaded. 	
4	Continue by adding instructions, routines, or modules. A new program is created.	

4.3.1 Handling of programs Continued

Loading an existing program

This section describes how to load an existing program.

	Action		
1	On the ABB menu, tap Program Editor.		
2	Tap Tasks and Programs.		
3	 Tap File, then Load Program. If there was already a program loaded, a warning dialog appears. Tap Save to save the loaded program. Tap Don't save to close loaded program without saving it, i.e. delete from pr gram memory. Tap Cancel to leave the loaded program. 		
4	Use the file searching tool to locate the program file to be loaded (file type pgf). The tap OK. The program is loaded and the program code is displayed. Manual Motors On Stopped (2 of 2) (Speed 100%)		
	Tasks and Programs Modules Routines		
	PROC main()		
	MoveL p10, v1000, z50, tool0;		
	MoveL p20, v100, z15, tool0;		
	WaitDI DI1, 1;		
	¹³ MoveL p30, v100, z15, tool0;		
	WaitDT DT2. 1:		
	¹⁴ WaitDI DI2, 1; ¹⁵ MoveL p40, v100, z15, tool0;		
	14 WaitDI DI2, 1; 15 MoveL p40, v100, z15, tool0; 16 MoveL p10, v100, z15, tool0;		
	14 WaitDI DI2, 1; 15 MoveL p40, v100, z15, tool0; 16 MoveL p10, v100, z15, tool0; 17 ENDPROC		
	14 WaitDI DI2, 1; 15 MoveL p40, v100, z15, tool0; 16 MoveL p10, v100, z15, tool0; 17 ENDPROC 18 ENDMODULE		
	14 WaitDI DI2, 1; 15 MoveL p40, v100, z15, tool0; 16 MoveL p10, v100, z15, tool0; 17 ENDPROC 18 ENDMODULE Add Add Instruction Edit Debug Modify Position Debug Declarations		

Saving a program

This section describes how to save a loaded program to the controller's hard disk. A loaded program is automatically saved in the program memory, but saving to the controller hard disk is an extra precaution.

	Action
1	On the ABB menu, tap Program Editor.
2	Tap Tasks and Programs.
3	Tap File and select Save Program As
4	Use the suggested program name or tap to open the soft keyboard and enter a new name. Then tap OK.

4.3.1 Handling of programs *Continued*

Renaming a loaded program

This section describes how to rename a loaded program.

	Action
1	On the ABB menu, tap Program Editor.
2	Tap Tasks and Programs.
3	Tap File and select Rename Program . A soft keyboard is displayed.
4	Use the soft keyboard to enter the new name of the program. Then tap OK .

Deleting a program

This section describes how to delete a program.

	Action
1	On the ABB menu, tap Program Editor.
2	Tap Tasks and Programs.
3	Tap File and select Delete Program. A confirmation dialog is displayed.
4	Tap OK to delete, or Cancel to keep the program intact.

4.3.2 Handling of modules

4.3.2 Handling of modules

Overview

This section details how to handle program modules. i.e.:

- create a new module
- load an existing module
- save a module
- rename a module
- delete a module

Creating a new module

This section describes how to create a new module.

	Action			
1	On the ABB menu, tap Program Editor.			
2	Tap Modules.			
3	Tap File, then tap New Module.			
	Manual Motors On Image: Stopped (2 of 2) (Speed 100%) MySystem (RSTEST4) Stopped (2 of 2) (Speed 100%)			
	New Module - MyProgram1 in T_ROB1			
	New Module			
	Name: Module1 ABC			
	Type: Program			
	OK Cancel			
	en0400000688			
4	Tap ABC and use the soft keyboard to enter the new module's name. Then tap OP to close the soft keyboard.			
5	Select which type of module to be created: • Program			
	• System			
	Then tap OK.			
	now to later switch between these types is detailed in section <i>Changing type of modul</i> on page 131.			

4.3.2 Handling of modules *Continued*

Loading an existing module

This section describes how to load an existing module.

	Action	
1	On the ABB menu, tap Program Editor	r.
2	Tap Modules.	
3 Tap File, then Load Module.		
	Manual MySystem (RSTEST4)	Motors On Stopped (2 of 2) (Speed 100%)
		Module Files (*.mod)
	Name 🛆	Type 1 to 4 of 4
	🗅 Arc	Folder
	🗅 MyProgram1	Folder
	🗅 MyProgram2	Folder
		Folder
	File name:	
	📑 🖄 í	OK Cancel
	T_ROB1 MainMod	
	Locate the module to be loaded. See s default path may be defined as detailed	ection <i>FlexPendant Explorer on page 34</i> . A d in section <i>Setting default paths on page 7</i>

Saving a module

This section describes how to save a module.

	Action
1	On the ABB menu, tap Program Editor.
2	Tap Modules and tap to select the module you want to load.
3	Tap File, then Save Module As
4	Tap on the suggested file name and use the soft keyboard to enter the module's name. Then tap OK.
5	Use the file searching tool to locate where you want to save the module. See section <i>FlexPendant Explorer on page 34</i> . The default location is on the controller disk, but any other location may be set as default as detailed in section <i>Setting default paths on page 77</i> .
	Then tap OK. The module is saved.

4.3.2 Handling of modules Continued

Renaming a module

This section describes how to rename a module.

	Action
1	On the ABB menu, tap Program Editor.
2	Tap Modules.
3	Tap File, then Rename Module The soft keyboard is displayed.
4	Use the soft keyboard to enter the module's name. Then tap OK .

Changing type of module

This section describes how to change the type of module.

	Action
1	On the ABB menu, tap Program Editor.
2	Tap Modules and select the module to be changed.
3	Tap File, then Change declaration
4	Tap Type and select module type.
5	Тар ОК.

Deleting a module

This section describes how to delete a module from memory. If the module has been saved to disk, it will not be erased from the disk.

	Action
1	On the ABB menu, tap Program Editor.
2	Tap Modules and tap to select the module you want to delete.
3	Tap File, then Delete Module A dialog box is displayed.
4	Tap OK to delete the module without saving it. If you want to save the module first, tap Cancel and save the module first. How to save the module is detailed in section <i>Saving a module on page 130</i> .

4.3.3 Handling of routines

4.3.3 Handling of routines

Overview

This section details how to handle program routines. i.e.:

- create a new routine
- create a copy of a routine
- · change the declaration of a routine
- delete a routine

Creating a new routine

This section details how to create a new routine, set the declaration, and add it to a module.

	Action		
1	On the ABB menu, tap Program	n Editor.	
2	Tap Routines.		
3	Tap File, then New Routine. A new routine is created and dis	splayed with default declara	tion values.
	Manual MySystem (RSTEST	Motors On [4] Stopped (2 of 2) (Speed	i 100%)
	Boutine Declaration	OB1/MainModule	
	Name:	Routine1	ABC
	Туре:	Procedure	
	Parameters:	None	
	Data type:	num	
	Module:	MainModule	
	Local declaration:	Undo Handler	•
	Error Handler:	Backward Ha	ndler:
	Result	ОК	Cancel
	T_ROB1 MainMod		
	en040000692		, <u>, , , , , , , , , , , , , , , , , , </u>
4	Tap ABC and use the soft key	board to enter the new rout	ines' name. Then tap OK.
5	Select the type of routine: • Procedure: used for a no • Function: used for a norm • Trap: used for an interrup	rmal routine without return v nal routine with return value ot routine	value
6	Do you need to use any parame If YES; tap and proceed as do page 133. If NO; proceed to the next step.	eters? etailed in section <i>Defining p</i>	arameters in routine on
	· · · · ·		

Continues on next page

4.3.3 Handling of routines Continued

	Action
8	Tap the checkbox to select Local declaration if the routine should be local. A local routine can only be used in the selected module.
9	Тар ОК.

Defining parameters in routine

This section describes how to define parameters in a routine.

In the routine decl A list of defined pa	aration, ta arameters	p to def is displaye	ine parameters ed.	S.		
	Manual MySystem (R9	STEST4)	Motors On Stopped (2 of	2) (Speed 100%)	X X	X
New Routine - MyPro	gram1 in T_	_ROB1/MainM	Iodule Broporty:	Value		
Add)elete	† ↓		ОК	Canc	el

4.3.3 Handling of routines *Continued*

	Action			
2	If no parameters are shown, tap Add to • Add optional parameter adds a	add a new par parameter that	ameter. is optional	
	Add optional mutual parameter another parameter	adds a paramet	er that is mu	tually optional with
	Read more about routine parameters ir	n the RAPID ref	erence mani	uals.
	Manual MySystem (RSTEST4)	Motors On Stopped (2 of 2) (Speed 100%)	X
	New Routine - MyProgram1 in T_ROB1/MainM	odule	-	
	Parameters:	Property:	Value:	
	Add Darameter			
	Add Ontional Parameter			
	Add Optional Mutual Parameter			
	Add Delete † ↓		ок	Cancel
	E T_ROB1 MainMod			
	en040000695			
2	Use the soft keyboard to enter the nam	o of the new p	ramator and	than tan OK
3	The new parameter is displayed in the	list.	aranneter and	i ineli iap OK.
	Manual	Motors On		
	MySystem (RSTEST4)	Stopped (2 of 2) (Speed 100%)	
	New Routine - MyProgram1 in T_ROB1/MainM	odule	Valuer	
	parameters.	Name:	naram0	
		Data type:	num	
		ArgNo:	1	
		AltNo:	0	
		Required:	True	
		Mode:	In	
		Dimension:	0	
	Add Delete † 🚽		0 K	Cancel
	La T_ROB1 MainMod			
	en0400000696			
4	Tap to select a parameter. To edit value	es, tap the valu	e.	

4.3.3 Handling of routines Continued

	Action
5	Tap OK to return to the routine declaration.

Creating a copy of a routine

This section describes how to create a copy of a routine.

	Action
1	On the ABB menu, tap Program Editor.
2	Tap Routines.
3	Highlight the routine by tapping it.
4	Tap File, then Copy Routine. The new routine is displayed. The name of the new routine is set to the same as the original with the suffix <i>Copy</i> .
5	Make any changes in the declarations for the new routine copy. Then tap OK. How to make all declarations is detailed in section <i>Creating a new routine on page 132</i> .

Changing the declaration of a routine

This section describes how to change the declaration of a routine.

	Action
1	On the ABB menu, tap Program Editor.
2	Tap Routines.
3	Highlight the routine by tapping it.
4	Tap File, then Change Declaration
5	Change any declaration values for the routine. Then tap OK. Declaration settings are described in section <i>Creating a new routine on page 132</i> .

Moving a routine

This section describes how to move a routine to another module.

	Action
1	On the ABB menu, tap Program Editor.
2	Tap Routines.
3	Highlight the routine by tapping it.
4	Tap File, then Move Routine
5	Select task and module. Then tap OK.

Deleting a routine

This section describes how to delete a routine from memory.

	Action
1	On the ABB menu, tap Program Editor.
2	Tap Routines.
3	Highlight the routine by tapping it.
4	Tap File, then Delete Routine A dialog box is displayed.

4.3.3 Handling of routines *Continued*

	Actio	n
5	Tap:	OK to delete the routine without saving any changes made to it. Cancel to revert without deleting the routine.

4.3.4 Handling of instructions

4.3.4 Handling of instructions

Instructions A RAPID program consists of instructions. An instruction can, for example, move the robot, set an I/O signal, or write a message to the operator. A large number of instructions are available, and these are listed in *Technical reference manual - RAPID Instructions, Functions and Data types.* The basic procedure for adding instructions are, however, identical.

Undo and redo

When editing programs in the Program editor, you can undo and redo up to three steps. This function is available in the **Edit** menu.

Adding instructions

This section describes how to add instructions.

	Action					
l	On the ABB menu, tap Program Editor.					
2	Tap to highlight the instruction under which	you want to add a new instruction.				
3	Tap Add instruction. A category of instructions is displayed.	otors On 🛐 🕱 🔽				
WySystem (RSTEST4) Stopped (2 of 2) (Speed 100%)						
	Tasks and Programs Modules	▼ Routines ▼				
	9 PROC main()	Common				
	149 MoveL p10, v1000,	:= Compact IE				
	¹¹ MoveL p20, v100, z					
	¹² WaitDI DI1, 1;	MoveAbs1 MoveC				
	¹³ MoveL p30, v100, z	Movel Movel				
	14 WaitDI DI2, 1;					
	¹⁵ MoveL p40, v100, z	Proceall Reset				
	$\begin{bmatrix} 16 \\ MoveL p10, v100, z \\ moveL p10, v100, v100, z \\ moveL p10, v100, v100$	Set Set				
		< Previous Next>				
	Add Finstruction Edit Debug	▲ Modify Hide Position Declarations				
	T_ROB1 MainMod					
	en040000697					
	A large number of instructions, divided into a fault category is Common , where the most of	several categories, are available. The de common instructions are listed.				
	You can create three personalized lists using Common Instruction in the topic Man-machine are described in Technical reference manual	g the system parameters of the type <i>Mos</i> <i>Communication</i> . The system parameter <i>I - System parameters</i> .				
1	Tap Common to display a list of the availabl You can also tap Previous/Next at the botton next/previous category.	e categories. m of the list of instructions to move to th				

4.3.4 Handling of instructions *Continued*

	Action
5	Tap the instruction you want to add. The instruction is added to the code.

Editing instruction arguments

This section describes how to edit instruction arguments.

Continues on next page

4.3.4 Handling of instructions Continued

Tap Change Selected. Depending on the type of instruction, the arguments have different data types. the soft keyboard to change string values or proceed to the next steps for other types or multiple argument instructions.				
Manual MySystem (RSTEST	Motors On Stopped (2 of 2) (Speed 100%)			
🗄 Change Selected				
Current instruction: MoveL				
Select the argument to be changed.				
Argument	Value 1 to 4 of 4			
ToPoint	p10			
Speed	v1000			
Zone	z50			
Tool	tool0			
Optional Argument	OK Cancel			
T_ROB1	ROB_1			
MainMod	16 3			
n0400000702				
ap the argument to be changed A number of options are display	d. ved.			
Tap the argument to be changed A number of options are display	d. /ed. Motors On 14) Stopped (2 of 2) (Speed 100%)			
Tap the argument to be changed A number of options are display The manual MySystem (RSTEST Change Selected Current argument: Select argument value.	d. /ed. 14) Motors On Stopped (2 of 2) (Speed 100%) Stopped (2 of 2) (Speed 100%)			
Tap the argument to be changed A number of options are display The manual MySystem (RSTEST. Change Selected Current argument: Select argument value. Movel p10, v1000,	d. /ed. ¹⁴⁾ Motors On Stopped (2 of 2) (Speed 100%) it Active filter: , z50 , tool0;			
Tap the argument to be changed A number of options are display The selected Current argument: ToPoint Select argument value. Movel p10, v1000, Data	d. /ed. T4) Motors On Stopped (2 of 2) (Speed 100%) T t Active filter: , z50 , tool0; Functions			
Tap the argument to be changed A number of options are display The manual MySystem (RSTEST Change Selected Current argument: ToPoint Select argument value. Movel p10, v1000, Data New	d. /ed. (4) Motors On Stopped (2 of 2) (Speed 100%) (Second 100%) t Active filter: , z50 , tool0; Functions 100 9 of 1 *			
Tap the argument to be changed A number of options are display Tap the argument of options are display Manual MySystem (RSTEST Change Selected Current argument: ToPoint Select argument value. Movel pl0, v1000, Data New LastArcToPoint	d. /ed. (4) Motors On Stopped (2 of 2) (Speed 100%) (Stopped 10			
Tap the argument to be changed A number of options are display The manual MySystem (RSTEST) Change Selected Current argument: ToPoint Select argument value. Movel pl0, v1000, Data New LastArcToPoint p20	d. /ed. (4) Motors On Stopped (2 of 2) (Speed 100%) (Second 100%) (c) (Speed 100%) (Second 100%) (c) (Speed 100%) (C) (Second 100%) (C) (Second 100%) (c) (Speed 100%) (C) (Speed 100%) (C) (Second 100%) (C			
Tap the argument to be changed A number of options are display Tap the argument of options are display Manual MySystem (RSTEST Current argument: ToPoint Select argument value. Movel pl0, v1000, Data New LastArcToPoint p20 p40	d. /ed. (4) Motors On Stopped (2 of 2) (Speed 100%) (Second 100%) (Second 100%) tt Active filter: , z50 , tool0; Functions 1 to 9 of 1 * p10 p30 p50			
Tap the argument to be changed A number of options are display Manual MySystem (RSTEST: Change Selected Current argument: ToPoint Select argument value. MoveL pl0, v1000, Data New LastArcToPoint p20 p40 pAE ErrPoint	d. /ed. (4) Motors On Stopped (2 of 2) (Speed 100%) (Second 100%) (Second 100%) (Active filter: , z50 , tool0; Functions 1 to 9 of 1 * p10 p30 p50			
Tap the argument to be changed A number of options are display Tap the argument of options are display Manual MySystem (RSTEST: Change Selected Current argument: ToPoint Select argument value. Movel p10, v1000, Data New LastArcToPoint p20 p40 pAE_ErrPoint	d. /ed. (4) Motors On Stopped (2 of 2) (Speed 100%) (Second 100%) (Second 100%) (t) (Active filter: , z50 , tool0; Functions 1 to 9 of 1 * p10 p30 p50			
Tap the argument to be changed A number of options are display Manual MySystem (RSTEST Change Selected Current argument: ToPoint Select argument value. MoveL pl0, v1000, Data New LastArcToPoint p20 p40 pAE_ErrPoint 123 Expression	d. /ed. (4) Motors On stopped (2 of 2) (Speed 100%) () () () () () () () () () () () () ()			
Tap the argument to be changed A number of options are display The Analysian (RSTEST Change Selected Current argument: ToPoint Select argument value. Movel p10, v1000, Data New LastArcToPoint p20 p40 pAE_ErrPoint 123 Expression Manual Movel p10, v1000, Comparison (RSTEST) Manual Movel p10, v1000, Comparison (RSTEST) Manual Movel p10, v1000, Comparison (RSTEST) Manual Movel p10, v1000, Comparison (RSTEST) Manual Movel p10, v1000, Comparison (RSTEST) Manual Manual Manual Movel p10, v1000, Comparison (RSTEST) Manual Ma	d. /ed. T4) Motors On Stopped (2 of 2) (Speed 100%) T Active filter: f z50 , tool0; Functions t p10 p30 p50 on Edit OK Cancel $V_{E}^{ROB} 1$			

4.3.4 Handling of instructions *Continued*



Tapping twice on an instruction will automatically launch the Change selected option. Tapping twice on an instruction argument will automatically launch the argument editor.

Copying and pasting instructions or arguments

This section describes how to paste instructions or arguments.

	Action
1	Tap to select the argument or instruction you want to copy.
	To select more than one row: select the first row, tap Select Range in the Edit menu and then tap the last row.
2	Tap Edit and then tap Copy.
3	Place the cursor on the instruction above where you want to paste the instruction or argument, or tap on the argument or instruction you want to change and tap Paste .

Cutting an instruction

This section describes how to cut an instruction.

	Action
1	Tap to select the instruction you want to cut.
	To select more than one row: select the first row, tap Select Range in the Edit menu and then tap the last row.
2	Tap Edit and then tap Cut.

Changing motion mode for a move instruction

This section describes how to change the motion mode for a move instruction.

1 Ta	ap to select the move instruction you want to change and then tap Edit.
2 Ta Th	Tap Change to MoveJ or Change to MoveL. The change is performed.

Commenting instruction rows

Instruction rows can be commented, i.e. skipped in the program execution. The comment/uncomment command is found under the Edit menu in the **Program** Editor.

4.3.5 Example: Add movement instructions

4.3.5 Example: Add movement instructions

Overview

In this example you will create a simple program that makes the robot move in a square. You need four movement instructions to complete this program.



en0400000801

A	First point
в	Robot movement Speed data v50 = speed 50mm/s
С	Zone z50 = (50mm)

Add movement instructions

This section details how to add movement instructions.

	Action	Information
1	Jog the robot to the first point.	Tip: Use only left-right/up-down joystick movements to jog in a square.
2	In the program editor, tap Add Instruction.	
3	Tap MoveL to insert a MoveL instruction.	
4	Repeat for the next four positions of the square.	
5	For the first and last instruction. Tap z50 in the instruction, tap Edit and then Change selected to Fine . Tap OK	

Result

Your program code should look like this:

```
Proc main()
    MoveL *, v50, fine, tool0;
    MoveL *, v50, z50, tool0;
    MoveL *, v50, z50, tool0;
    MoveL *, v50, z50, tool0;
    MoveL *, v50, fine, tool0;
End Proc;
```

4.3.6 About the Program and Motion Pointers

4.3.6 About the Program and Motion Pointers

The Program Pointe	r
	The Program Pointer (PP) indicates the instruction with which the program will start when you press any of the Start, Forward, or Backward buttons on the FlexPendant.
	Program execution continues from the instruction where the Program Pointer is. However, if the cursor is moved to another instruction when the program is stopped, the Program Pointer can be moved to the position of the cursor (or the cursor can be moved to the Program Pointer), and execution can be restarted from there.
	The Program Pointer is shown as a yellow arrow to the left of the program code in the Program Editor and Production Window .
The Motion Pointer	
	The Motion Pointer (MP) indicates the instruction that the robot is currently executing. This is normally one or more instructions after the Program Pointer, as the system executes and calculates the robot path faster than the robot moves.
	The Motion Pointer is shown as a small robot to the left of the program code in the Program Editor and in the Production Window .
The cursor	
	The cursor can indicate a complete instruction or any of the arguments.
	The cursor is shown as blue highlighting of the program code in the Program Editor .
Program Editor	
	If you toggle between the Program Editor and another view and back again, the Program Editor will show the same part of the code as long as the program pointer has not been moved. If the program pointer is moved, the Program Editor shows the code at the position of the program pointer. The same behavior applies to the Production Window .
Deleted information	
	Production Window on page 38.
	Program Editor on page 41.
	Stepping instruction by instruction on page 191.
	Starting programs on page 223.

4.4.1 Viewing data in specific tasks, modules, or routines

4.4 Data types

4.4.1 Viewing data in specific tasks, modules, or routines

Overview

It is possible to view selections of data types by selecting a specific scope.

Viewing data in specific tasks, modules, or routines

This section details how to view data instances in specific modules or routines.

	Action			
1	On the ABB menu, tap Proc	gram Data.		
2	Tap Change Scope . The following screen is displayed:			
	Hanual MySystem	(RSTEST4) /pes	Motors On Stopped (2 of 2) (Speed 100	1%) X X
	Select a data type from the lis	t.		
	Scope: RAPID/T_ROB1	I		ОК
	clock	RAPID/T_ROB	51	
	robtarget	Built-In Da	ta Only:	
		© Current Ex	ecution:	
		Task:	T_ROB1	
		Module:	BASE	-
		Routine:		•
		Note that data dec routine scope.	lared in routines is listed only	when using the
	Program Data			
	en0400000661			
3	Select the required scope b • Built-In Data Only: S • Current execution: S • Task: Shows all data • Module: Shows all data • Routine: Shows all data	y selecting: hows all data typ Shows all data ty types used by a ata types used b ata types used b	pes used by the specif pes used in the curren specific task y a specific module by a specific routine	ic system t execution
4	Tap OK to confirm your cho	pice.		
5	Tap twice to select a data ty	/pe and view its	instances.	

4.4.2 Creating new data instance

4.4.2 Creating new data instance

Creating new data instance

This section details how to create new data instances of data types.

	Action		
1	On the ABB menu, tap Program Data . A list of all available data types is displayed.		
2	Tap the data instance type to be created, i.e. bool and then tap Show data . A list of all instances of the data type is displayed.		
3	Tap New.		
	Manual Motors On Stopped (2 of 2) (Speed 100%)		
	Data type: clock Current Task: T ROB1		
	Name: clock2		
	Scope: Global		
	Storage type: Variable		
	Task:		
	Module: MainModule		
	Routine:		
	Dimension:		
	OK Cancel		
	Program Data		
	en0400000663		
4	Tap the right of Name to define the data instance's name.Name		
5	Tap the Scope menu to set accessibility for the data instance. Select:		
	Local - reachable within the module		
	Task - reachable within the task		
6	Tap the Storage type menu to select type of memory used for the data instance. Select Persistent if the data instance is persistent 		
	Variable if the data instance is variable		
_	Constant in the data instance is constant		
17	lad the Module menu to select module.		
7 8	Tap the Routine menu to select module.		
7 8 9	Tap the Routine menu to select module. Tap the Routine menu to select routine. If you want to create an array of data instances, then tap the Dimensions menu and select the number of dimensions in the array, 1-3. • 1		
7 8 9	Tap the Routine menu to select module. Tap the Routine menu to select routine. If you want to create an array of data instances, then tap the Dimensions menu and select the number of dimensions in the array, 1-3. • 1 • 2 • 3		
7 8 9	Tap the Routine menu to select module. Tap the Routine menu to select routine. If you want to create an array of data instances, then tap the Dimensions menu and select the number of dimensions in the array, 1-3. • 1 • 2 • 3 • None		

Continues on next page
4.4.2 Creating new data instance *Continued*

	Action
10	Тар ОК.

4.4.3 Editing data instances

4.4.3 Editing data instances

Overview

This section describes how to view data instances in the **Program Data** window. It also details how to edit, delete, change declaration of, copy, and define a data instance.

For the data types tooldata, wobjdata and loaddata also see sections *Tools* on page 150, Work objects on page 168 or Payloads on page 177.

Viewing data instances

This section details how to view the available instances of a data type.

	Action					
1	On the ABB menu, tap Program Data.					
2	Tap the data type you want view and then tap Show Data.					
3	Tap the data instance you want to edit, and then tap Edit.					
	Manual Motors On MySystem (RSTEST4) Stopped (2 of 2) (Speed 100%)			00%)		
	Pata of type: robt	arget				
	Select the data you	ı want to edit.	Active filter:			
	Scope: RAPID/T_RO	B1				Change Scope
	Name	Value		Module		1 to 5 of 5
	p10	[[515,0,7	712],[0.70	MainMo	dule	Global
	p20	[[515,0,	Delete		dule	Global
	p30	[[515,0,	Change Dec	aration	lule	Global
	p40	[[515,0,	Change Valu	ie	lule	Global
	p50	[[515,0,	Сору		lule	Global
			Define			
			Modify Posit	ion		
		New	Edit	Re	fre s h	View Data Types
	Program Data			·		
	en0400000671					
4	 Depending on what you want to do, tap one of the following menu items: Tap Delete to remove the data instance. Tap Change Declaration to change the declaration of the data instance. Tap Change Value to edit the value of the data instance. Tap Copy to copy the data instance. Tap Define to define the instance (only available for tooldata, wobjdata and loaddata). Tap Modify Position to modify a position (only available for robtarget and instance) 					
	Proceed as descri	bed in the respe	ective section for	ollowing	below.	

4.4.3 Editing data instances Continued

Editing the value of a data instance

This section describes how to edit a data instance value.

	Action	Information
1	Tap Change Value to open the instance.	
2	Tap the value to open a keyboard or list of choices.	The way to edit a value depends on the data type and possible values, for instance text, numbers, predefined values etc.
3	Select or enter a new value.	
4	Тар ОК.	



If the value of a persistent variable is changed at any point in a running program, the **Program Editor** will still show the old value until the program stops. The **Program Data** view, however, always shows the current value of persistent variables. See *Persistent declaration* in the *Technical reference manual - RAPID Overview* for further information.

Deleting a data instance

This section details how to delete a data instance.

	Note
A	data instance can be of type tool, work object, payload or others.
	Action
1	Tap Delete in the menu for the data instance to be deleted, as detailed in section <i>Viewing data instances on page 146</i> . A dialog box is displayed.
2	Tap Yes if you are sure the data instance is to be deleted.

A deleted tool, work object or payload cannot be recovered, and all related data will be lost. If the tool, work object or payload is referenced by any program, those programs cannot run without changes.

If you delete a tool you cannot continue the program from the current position.

4.4.3 Editing data instances *Continued*

Changing the declaration of a data instance

This section details how to change the declaration of a data instance.

	Action				
1	Tap Change Declaration in in section <i>Viewing data insta</i>	the menu for the data instance to Inces on page 146.	be deleted, as detailed		
	Manual MySystem (Motors On RSTEST4) Stopped (2 of 2) (Sp	eed 100%)		
	A New Data Declaration				
	Data type: clock	Current Task: T_ROB	1		
	Name:	clock2			
	Scope:	Global	•		
	Storage type:	Variable	•		
	Task:	T_ROB1	•		
	Module:	MainModule	•		
	Routine:	<none></none>	•		
	Dimension:	<none></none>			
		ОК	Cancel		
	Program Data				
	en0400000672				
2	Select what data instance va	lues to be changed: out the soft keyboard and chan	ge the name.		
	Scope		g		
	Storage type				
	Module				
	Routine				

Copying a data instance

This section details how to copy a data instance.

	Action
1	Tap Copy in the menu for the data instance to be copied, as detailed in section <i>Viewing data instances on page 146</i> . A copy of the data instance is created.
	The copy has the same values as the original, but the name is unique.

Defining a data instance

How to define the tool frame or work object frame is described in the sections *Defining the tool frame on page 157* and *Defining the work object coordinate system on page 170*.

4.4.3 Editing data instances Continued

Modifying position of a data instance

Only instances of data types robtarget and jointtarget can use the function Modify Position. The currently active work object and tool will be used in the operation.

More information about modifying positions is detailed in *Modifying and tuning positions on page 246*.



Note

Make sure that the correct work object and tool are selected when modifying positions in the **Program Data** window. This is not verified automatically by the system.

4.5.1 What is a tool?

4.5 Tools

4.5.1 What is a tool?

Tool

A tool is an object that can be mounted directly or indirectly on the robot turning disk or fitted in a fixed position within the robot working range.



A fixture (jig) is not a tool.

All tools must be defined with a TCP (Tool Center Point).

Each tool that can be used by the robot must be measured and its data stored in order to achieve accurate positioning of the tool center point.



WARNING

It is important to always define the actual tool load and, when used, the payload of the robot (for example, a gripped part). Incorrect definitions of load data can result in overloading of the robot mechanical structure. There is also a risk that the speed in manual reduced speed mode can be exceeded.

When incorrect load data is specified, it can often lead to the following consequences:

- The robot may not use its maximum capacity. ٠
- Impaired path accuracy including a risk of overshooting. •
- Risk of overloading the mechanical structure.

The controller continuously monitors the load and writes an event log if the load is higher than expected. This event log is saved and logged in the controller memory.

4.5.1 What is a tool? *Continued*



Operating manual - IRC5 with FlexPendant
3HAC050941-001 Revision: Q

Illustration

4.5.2 What is the tool center point?

4.5.2 What is the tool center point?

Illustration

The illustration shows how the tool center point (TCP) is the point around which the orientation of the tool/manipulator wrist is being defined.



xx0300000604

Description

The tool center point (TCP) is the point in relation to which all robot positioning is defined. Usually the TCP is defined as relative to a position on the manipulator turning disk.



CAUTION

Incorrect settings for the TCP will result in incorrect speed. Always verify the speed after changing the settings.

The TCP will be jogged or moved to the programmed target position. The tool center point also constitutes the origin of the tool coordinate system.

The robot system can handle a number of TCP definitions, but only one can be active at any one time.

There are two basic types of TCPs: moveable or stationary.

Moving TCP

The vast majority of all applications deal with moving TCP, i.e. a TCP that moves in space along with the manipulator.

A typical moving TCP can be defined in relation to, for example the tip of a arc welding gun, the center of a spot welding gun, or the end of a grading tool.

4.5.2 What is the tool center point? *Continued*

Stationary TCP

In some applications a stationary TCP is used, for example when a stationary spot welding gun is used. In such cases the TCP can be defined in relation to the stationary equipment instead of the moving manipulator.

4.5.3 Creating a tool

4.5.3 Creating a tool

What happens when you create a tool?

When you create a new tool a variable of the data type tooldata is created. The variable name will be the name of the tool. For more information on data types, see *Technical reference manual - RAPID Instructions, Functions and Data types*.

The new tool has initial default values for mass, frame, orientation etc., which must be defined before the tool can be used.

How to create a tool

The tool center point of the default tool (tool0) is in the center of the robot's mounting flange and shares the orientation of the robot base.

By creating a new tool you define another tool center point. For more information about tools and the tool center points see *What is a tool? on page 150* and *What is the tool center point? on page 152*.



en0400000779

Α	Т	Tool center point, TCP, for tool0
	Action	1
1	On the	e ABB menu, tap Jogging.
2	Тар То	ool to display the list of available tools.

4.5.3 Creating a tool Continued

	Action					
3	3 Tap New to create a new tool.					
	Manual MySystem	(RSTEST4) Motors On Stopped (2 of 2) (Speed 100%)				
	A New Data Declaration					
	Data type: tooldata Current Task: T_ROB1					
	Name:	tool2				
	Scope:	Task				
	Storage type:	Persistent				
	Task:	T_ROB1				
	Module:	MainModule 🗾				
	Routine:	<none></none>				
	Dimension:	<none></none>				
	Initial Value	OK Cancel				
	£ Jogging					
	en0300000544					
	Enter values for each field, see table below.					
4	Тар ОК.					

Tool declaration settings

If you want to change	then	Recommendation
the name of the tool	tap button next to Name	Tools are automatically named tool followed by a running number, for example tool10 or tool21.
		You are recommended to change this to something more descriptive such as gun, gripper or welder.
		Note
		If you change the name of a tool after it is referenced in any program you must also change all occurrences of that tool.
the scope	select the preferred scope from the menu	Tools should always be glob- al, as to be available to all modules in the program.
the storage type	-	Tool variables must always be persistent.
the module	select the module in which this tool should be declared from the menu	

4.5.3 Creating a tool *Continued*

If you want to change	then	Recommendation
the size of the data array's axes	tap button next to Dimen- sion	

Note

The created tool is not useful until you have defined the tool data (TCP coordinates, orientation, weight etc.). See *Editing the tool data on page 161* and *LoadIdentify, load identification service routine on page 202* to learn more about how to do it.

4.5.4 Defining the tool frame

4.5.4 Defining the tool frame

Preparations

To define the tool frame, you first need a reference point in the world coordinate system. If you need to set the tool center point orientation, you also need to affix elongators to the tool.

You also need to decide which method to use for the tool frame definition.

Available methods

There are three different methods which can be used when defining the tool frame. All three require that you define the cartesian coordinates of the tool center point. What differs is how the orientation is defined.

If you want to	then select
set the orientation the same as the orientation of the robot's mounting plate	TCP (default orient.)
set the orientation in Z axis	TCP&Z
set the orientation in X and Z axes	TCP&Z,X

How to select a method

This procedure describes how to select the method to be used when defining the tool frame.

	Action
1	On the ABB menu, tap Jogging .
2	Tap Tool to display a list of available tools.
3	Select the tool you want to define.
4	In the Edit menu, tap Define

4.5.4 Defining the tool frame *Continued*

	Action			
5	In the dialog box v	vhich appears, select the m Manual MySystem (RSTEST4)	nethod to use. Motors On Stopped (2 of 2) (Speed 100%)	33 X
	🐣 Program Data - to	oldata - Define		
	Tool Frame Def	inition		
	Tool:	tool1		
	Select a method, n	nodify the positions and tap O	к.	
	Method:	ICP (default orient.)	▼ No. of points:	4 💌
	Point	Status		1 to 4 of 4
	Point 1	-		
	Point 2	-		
	Point 3	-		
	Point 4	-		
	Positions	Modify Position	ОК	Cancel
	L Jogging			
	en0600003147			
6	Select the number more points to get all of them.	of approach points to use. I a more accurate result, you	Jsually 4 points is enough. should be equally careful	lf you choose when defining
7	See <i>How to proce</i> gather positions a	ed with tool frame definition nd perform the tool frame o	n on page 159 for informat definition.	ion on how to

4.5.4 Defining the tool frame *Continued*



n04	100	000	906

	Action	Information
1	Jog the robot to an appropriate position, A, for the first approach point.	Use small increments to accurately posi- tion the tool tip as close to the reference point as possible.
2	Tap Modify Position to define the point.	
3	Repeat step 1 and 2 for each approach point to be defined, positions B, C, and D.	Jog away from the fixed world point to achieve the best result. Just changing the tool orientation will not give as good a result.
4	If the method you are using is TCP&Z or TCP&Z,X orientation must be defined as well.	Follow the instructions in <i>How to define elongator points on page 160</i> .
5	If, for some reason, you want to redo the calibration procedure described in step 1-4, tap Positions and then Reset All .	

4.5.4 Defining the tool frame *Continued*

	Action	Information
6	When all points are defined you can save them to file, which enables you to reuse them later. On the Positions menu, tap Save .	
7	Tap OK. The Calculation Result dialog box will now be displayed, asking you to cancel or to confirm the result before it is written to the controller.	For further information see <i>Is the calcu-</i> <i>lated result good enough? on page 160</i>

How to define elongator points

This procedure describes how to define the orientation of the tool frame by specifying the direction of the z and/or x axis. You need to do this only if you the tool orientation should differ from that of the robot base. The tool coordinate system by default resembles the coordinate system of tool0, as illustrated in *Measuring the tool center point on page 162*.

	Action
1	Without changing the orientation of the tool, jog the robot so that the reference world point becomes a point on the desired positive axis of the rotated tool coordinate system.
2	Tap Modify Position to define the point.
3	Repeat step 1 and 2 for the second axis if it should be defined.

Is the calculated result good enough?

The **Calculation Result** dialog box displays the calculated result of the tool frame definition. You have to confirm that you accept the result before it can take effect in the controller. The alternative is to redo the frame definition in order to achieve a better result. The result **Mean Error** is the average distance of the approach points from the calculated TCP (tool center point). **Max Error** is the maximum error among all approach points.

It is hard to tell exactly what result is acceptable. It depends on the tool, robot type etc. you are using. Usually a mean error of a few tenths of a millimeter is a good result. If the positioning has been undertaken with reasonable accuracy the result will be okay.

As the robot is used as a measuring machine, the result is also dependent on where in the robot's working area the positioning has been done. Variation of the actual TCP up to a couple of millimeters (for large robots) can be found between definitions in different parts of the working area. The repeatability of any following TCP calibrations will thus increase if these are done close to the preceding ones. Note that the result is the optimal TCP for the robot in that working area, taking into account any discrepancies of the robot in the configuration at hand.



A common way to check that the tool frame has been correctly defined is to perform a reorientation test when the definition is ready. Select the reorient motion mode and the tool coordinate system and jog the robot. Verify that the tool tip stays very close to the selected reference point as the robot moves.

4.5.5 Editing the tool data

4.5.5 Editing the tool data

Tool data

Use the value settings to set the tool center point position and physical properties of the tool such as weight and center of gravity.

This can also be done automatically with the service routine LoadIdentify. See sections *Running a service routine on page 194*, or *LoadIdentify, load identification service routine on page 202*.



If the tooldata is incorrectly defined there is a risk that the speed is higher than expected. This is particularly important in manual mode.

Displaying the tool data

This section details how to display the tool data.

	Action
1	On the ABB menu, tap Jogging .
2	Tap Tool to display the list of available tools.
3	 Tap the tool you want to edit, then tap Edit. A menu appears. Change Declaration Change Value Delete Define
4	In the menu, tap Change Value . The data that defines the tool appears. Green text indicates that the value can be changed.
5	Proceed with changing the data as described below.

4.5.5 Editing the tool data *Continued*

Measuring the tool center point

The easiest way to define the tool center point, TCP, is usually to use the predefined method described in *Defining the tool frame on page 157*. If you use this method, you do not have to write any values for the frame as these are supplied by the method.

If you already have the measurements of the tool, or for some reason want to measure them manually, the values can be entered in the tool data.



en0400000881

Х		X axis for tool0
Y		Y axis for tool0
Z		Z axis for tool0
Х		X axis for the tool you want to define
Y		Y axis for the tool you want to define
Z		Z axis for the tool you want to define
	Actio	n
1	Measure the distance from the center of the robot's mounting flange to the tool's center point along the X axis of tool0.	
2	Measure the distance from the center of the robot's mounting flange to the tool's center point along the Y axis of tool0.	

3 Measure the distance from the center of the robot's mounting flange to the tool's center point along the Z axis of tool0.

Editing the tool definition

	Action	Instance	Unit
1	Enter the cartesian coordinates of the tool	tframe.trans.x	[mm]
	center point's position.	tframe.trans.y	
		tframe.trans.z	

4.5.5 Editing the tool data *Continued*

	Action	Instance	Unit
2	If necessary, enter the tool frame orientation.	tframe.rot.q1	None
		tframe.rot.q2	
		tframe.rot.q3	
		tframe.rot.q4	
3	Enter the weight of the tool.	tload.mass	[kg]
4	If necessary, enter the tool's center of gravity.	tload.cog.x	[mm]
		tload.cog.y	
		tload.cog.z	
5	If necessary, enter the orientation of the axis of moment	tload.aom.q1	None
		tload.aom.q2	
		tload.aom.q3	
		tload.aom.q4	
6	If necessary, enter the tool's moment of inertia.	tload.ix	[kgm ²]
		tload.iy	
		tload.iz	
7	Tap OK to use the new values, Cancel to leave the definition unchanged.		

4.5.6 Editing the tool declaration

4.5.6 Editing the tool declaration

Tool declaration

Use the declaration to change how the tool variable can be used in the program's modules.

Displaying the tool declaration

	Action
1	On the ABB menu, tap Jogging.
2	Tap Tool to see the list of available tools.
3	 Tap the tool you want to edit, then tap Edit. A menu appears. Change Declaration Change Value Delete Define
4	In the menu, tap Change Declaration . The tool's declaration appears.
5	Edit the tool declaration as listed in section <i>Creating a tool on page 154</i> .
	Note

If you change the name of a tool after it is referenced in any program you must also change all occurrences of that tool.

4.5.7 Deleting a tool

4.5.7 Deleting a tool

Deleting a tool

For more information about deleting a tool, see *Deleting a data instance on page 147*.

4.5.8 Setup for stationary tools

4.5.8 Setup for stationary tools

Stationary tools

Stationary tools are used, for instance, in applications that involve large machines such as cutters, presses and punch cutters. You may use stationary tools to perform any operation that would be difficult or inconvenient to perform with the tool on the robot.

With stationary tools, the robot holds the work object.

Make a tool stationary

This section describes how to make a tool stationery.

	Action
1	On the ABB menu, tap Jogging .
2	Tap Tool to display the list of available tools.
3	Tap the tool you want to edit, then tap Edit . A menu appears.
4	In the menu, tap Change value . The data that defines the tool appears.
5	Tap the instance robhold.
6	Tap FALSE to make this tool stationary.
7	Tap OK to use the new setup, Cancel to leave the tool unchanged.

Make a work object robot held

This section describes how to make a work object robot held.

	Action
1	In the Jogging window, tap Work object to display the list of available work objects.
2	Tap the work object you want to edit, then tap Edit . A menu appears.
3	In the menu, tap Change value . The data that defines the work object appears.
4	Tap the instance robhold.
5	Tap TRUE to indicate that this work object is held by the robot.
6	Tap OK to use the new setup, Cancel to leave the work object unchanged.

Differences in coordinate system referencing

This section describes differences in coordinate system referencing.

The	normally references the	but now references the	
work object coordinate sys- tem	user coordinate system	user coordinate system (no change)	
user coordinate system	world coordinate system	robot's mounting plate	
tool coordinate system	robot's mounting plate	world coordinate system	

4.5.8 Setup for stationary tools Continued

Set up the tool coordinate system

You use the same measurement methods to set up a stationary tool coordinate system as with tools mounted on the robot.

The world reference tip must, in this case, be attached to the robot. Define and use a tool with the reference tip's measurements when you create approach points. You also need to attach elongators to the stationary tool if you need to set up the orientation.

You should enter the reference tip's tool definition manually to minimize errors when calculating the stationary tool's coordinate system.

You may enter the stationary tool's definition manually.



en0400000990

4.6.1 What is a work object?

4.6 Work objects

4.6.1 What is a work object?

Illustration



Description

A work object is a coordinate system with specific properties attached to it. It is mainly used to simplify programming when editing programs due to displacements of specific tasks, objects processes etc.

The work object coordinate system must be defined in two frames, the user frame (related to the world frame) and the object frame (related to the user frame).

Work objects are often created to simplify jogging along the object's surfaces. There might be several different work objects created so you must choose which one to use for jogging.

Payloads are important when working with grippers. In order to position and manipulate an object as accurate as possible its weight must be accounted for. You must choose which one to use for jogging.

4.6.2 Creating a work object

4.6.2 Creating a work object

What happens when I create a work object?

A variable of the type wobjdata is created. The variable's name will be the name of the work object. For more information on data types, see *Technical reference* manual - RAPID Instructions, Functions and Data types.

This is detailed in section What is a work object? on page 168.

Creating a work object

The work object's coordinate system is now identical with the world coordinate system. To define the position and orientation of the work object's coordinate system, see *Editing the work object declaration on page 175*.

	Action
1	On the ABB menu, tap Jogging.
2	Tap Work Object to display the list of available work objects.
3	Tap New to create a new work object.
4	Тар ОК.

Work object declaration settings

If you want to change	then	Recommendation
the work object's name	tap the button next to it	Work objects are automatic- ally named wobj followed by a running number, for ex- ample wobj10, wobj27.
		You should change this to something more descriptive.
		If you change the name of a work object after it is refer- enced in any program you must also change all occur- rences of that work object.
the scope	select the scope of choice from the menu	Work objects should always be global to be available to all modules in the program.
the storage type	-	Work object variables must always be persistent.
the module	select the module in which this work object should be declared from the menu	

4.6.3 Defining the work object coordinate system

4.6.3 Defining the work object coordinate system

Overview

Defining a work object means that the robot is used to point out the location of it. This is done by defining three positions, two on the x-axis and one on the y-axis.

When defining a work object you can use either the user frame or the object frame or both. The user select frame and the object frame usually coincides. If not, the object frame is displaced from the user frame.

How to select method

This procedure describes how to select method for defining either user frame or object frame or both. Note that this only works for a user created work object, not the default work object, wobj0. Defining work object can also be done from the **Program Data** window.

	Action			
1	On the ABB menu, tap Jogging			
2	Tap Work object to display the list of	Tap Work object to display the list of available work objects.		
3	Tap the work object you want to defin	ne, then tap Edit .		
4	In the menu, tap Define			
5	t method menu. See Hor e object frame on page Speed 100%) Speed 100%) Speed 100%) Speed 100%) Speed 100% Speed	w to 172		
	Point	Status		
	Positions Mod Posi	tion Ol	K Cancel	
	L Jogging			
	en0400000893			

4.6.3 Defining the work object coordinate system Continued

How to define the user frame



This section details how to define the user frame.

en0400000887

The x axis will go through points X1-X2, and the y axis through Y1.

	Action	Information
1	In the User method pop up menu, tap 3 points.	
2	Press the three-position enabling device and jog the robot to the first (X1, X2 or Y1) point that you want to define.	Large distance between X1 and X2 is preferable for a more precise definition.
3	Select the point in the list.	
4	Tap Modify Position to define the point.	
5	Repeat steps 2 to 4 for the remaining points.	

4.6.3 Defining the work object coordinate system Continued

How to define the object frame

This section describes how to define the object frame if you want to displace it from the user frame.



en040000899

The x axis will go through points X1-X2, and the y axis through Y1.

	Action
1	In the Object method pop up menu, tap 3 points .
2	See steps 2 to 4 in the description of <i>How to define the user frame on page</i> 171.

How to save the defined positions

Normally the defined positions are only used as temporary positions by the controller to calculate the position of the work object and are then discarded. However, the positions can also be saved to a program module for later use or analysis.

When saving the positions a new program module is created where the positions are stored with predefined names given by the controller. The names of the positions can be changed afterwards, but when loading the positions it is recommended to use the predefined names.



Note

Only the positions (robtargets) are saved. Make sure to note which tool was used when modifying the defined positions.

	Action
1	When the work object frame definition is completed and all positions have been modified, tap OK.
2	In the Save Modified Points dialog, tap Yes.
3	Tap ABC to change the name of the program module, tap OK to accept the name.

Continues on next page

4.6.3 Defining the work object coordinate system Continued

	Action
4	The names of the positions and the module is displayed in the Save dialog, tap OK.

How to load defined positions

In some cases it is not practical or possible to use the robot to define the positions. Then the positions can be defined or calculated elsewhere and loaded to the **Work Object Frame Definition** dialog.

Positions from any program module can be can be loaded, but is recommended to use the module from the **Save Modified Points** dialog with predefined position names given by the controller.



CAUTION

Make sure that the correct tool and work object is activated in the **Work Object Frame Definition** dialog before loading any positions.

	Action
1	In the Work Object Frame Definition dialog, tap Positions and Load.
2	Tap the module that holds the calibration points, tap OK.
3	If the controller finds all or any predefined positions in the module, the positions are automatically loaded to the correct user or object point. In the Load dialog, tap OK.
4	If some positions are missing or do not have the correct names, the controller cannot load the positions automatically so the user is asked to match the positions manually.
	Tap each point in the list to assign the positions manually from the drop down list. Tap OK.
5	If necessary, use Modify Position to define any remaining points that could not be loaded.

4.6.4 Editing the work object data

4.6.4 Editing the work object data

Overview

Use the work object data definition to set the position and rotation of the user and object frames.

How to display the work object data

	Action
1	On the ABB menu, tap Jogging .
2	Tap Work object to display the list of available work objects.
3	Tap the work object you want to edit, then tap Edit.
4	Tap Change Value . The data that defines the work object appears.

How to set user and object frame values manually

The easiest way to set the work object and user coordinate systems position is to use the method described in *Defining the work object coordinate system on page 170*. You can however edit the values manually using the guide below.

Values	Instance	Unit
The cartesian coordinates of the position of the object	oframe.trans.x	mm
frame	oframe.trans.y	
	oframe.trans.z	
The object frame orientation	oframe.rot.q1	-
	oframe.rot.q2	
	oframe.rot.q3	
	oframe.rot.q4	
The cartesian coordinates of the position of the user	uframe.trans.x	mm
frame	uframe.trans.y	
	uframe.trans.z	
The user frame orientation	uframe.rot.q1	-
	uframe.rot.q2	
	uframe.rot.q3	
	uframe.rot.q4	



Editing work object data can also be done from the Program Data window.

4.6.5 Editing the work object declaration

4.6.5 Editing the work object declaration

Overview

Use the declaration to change how the work object variable can be used in the program's modules.

Displaying the work object declaration

	Action			
1	On the ABB menu, tap Jogging .			
2	Tap Work object to see the list of available work objects.			
3	Tap the work object you want to edit, then tap Edit.			
4	In the menu, tap Change Declaration.			
5	The work object's declaration appears.			
6	Edit the tool declaration as listed in section <i>Creating a work object on page 169</i> .			
Note				

If you change the name of a work object after it is referenced in any program you must also change all occurrences of that work object.

4.6.6 Deleting a work object

4.6.6 Deleting a work object

Deleting a work object

For more information about deleting a **work object**, see *Deleting a data instance on page 147*.

4.7.1 Creating a payload

4.7 Payloads

4.7.1 Creating a payload

What happens when I create a payload?

A variable of the type loaddata is created. The variables name will be the name of the payload. For more information on data types, see *Technical reference manual - RAPID Instructions, Functions and Data types*.

Adding a new payload and setting data declaration

The payloads coordinate system will be set to the position, including orientation, of the world coordinate system.

	Action				
1	On the ABB menu tap Jogging.				
2	Tap Payload or Total Load to display the list of available payloads.				
	Note				
	Total Load is displayed only when the value of ModalPayLoadMode is set to 0 and the mechanical units are TCP robots. See <i>Setting the value for ModalPayLoadMode on page 178</i> .				
3	Tap New to create a new payload and enter the data. See <i>Payload declaration settings</i> on page 178.				
4	Тар ОК.				



It is important to always define the actual tool load and, when used, the payload of the robot (for example, a gripped part). Incorrect definitions of load data can result in overloading of the robot mechanical structure. There is also a risk that the speed in manual reduced speed mode can be exceeded.

When incorrect load data is specified, it can often lead to the following consequences:

- The robot may not use its maximum capacity.
- Impaired path accuracy including a risk of overshooting.
- · Risk of overloading the mechanical structure.

The controller continuously monitors the load and writes an event log if the load is higher than expected. This event log is saved and logged in the controller memory.

4.7.1 Creating a payload *Continued*

Payload declaration settings

If you want to change	then	Recommendation	
the payload's name	tap the button next to it	Payloads are automatically named load followed by a running number, for example load10, load31.	
		You should change this to something more descriptive.	
		If you change the name of a payload after it is referenced in any program you must also change all occurrences of that payload's name.	
the scope	select the scope of choice from the menu	Payloads should always be global to be available to all modules in the program.	
the storage type	-	Payload variables must al- ways be persistent.	
the module	select the module in which this payload should be de- clared from the menu	-	

Setting the value for ModalPayLoadMode

This procedure describes how to modify the value of ModalPayLoadMode:

- 1 On the ABB menu, tap Control Panel and then Configuration.
- 2 Select Controller.
- 3 Select the type System Misc and tap.
- 4 Select ModalPayLoadMode and then tap Edit.
- 5 Tap the parameter **Value** twice and set to **0**.
- 6 Click OK.
- 7 Tap Yes to the question The changes will not take effect until the controller is warm started. Do you want to restart now?.

4.7.2 Editing the payload data

4.7.2 Editing the payload data

Overview

Use the payload data to set physical properties of the payload such as weight and center of gravity.

This can also be done automatically with the service routine LoadIdentify. See sections *Running a service routine on page 194*, or *LoadIdentify, load identification service routine on page 202*.

Displaying the payload definition

	Action
1	On the ABB menu, tap Jogging .
2	Tap Payload to display the list of available payloads.
3	Tap the payload you want to edit, then tap Edit.
4	Tap Change Value . The data that defines the payload appears.

Changing the payload data

This procedure describes how to manually enter the payload data. This can also be done automatically by running the service routine LoadIdentify. How to run a service routine is described in section *Running a service routine on page 194*.

	Action	Instance	Unit
1	Enter the weight of the payload.	load.mass	[kg]
2	Enter the payload's center of gravity.	load.cog.x load.cog.y load.cog.z	[mm]
3	Enter the orientation of the axis of moment.	load.aom.q1 load.aom.q2 load.aom.q3 load.aom.q3	
4	Enter the payload's moment of inertia.	ix iy iz	[kgm ²]
5	Tap OK to use the new values, Cancel to leave the data unchanged.	-	-

Using the PayLoadsInWristCoords parameter

By using the PayLoadsInWristCoords parameter, the loaddata for payloads can be specified relative to the wrist instead of the active TCP or work object. This can be useful if several tool or TCP or work objects (when tool is stationary) are used for one payload. In this case only one load identification is needed instead of one for each tool or TCP or work object. Thus it is possible to use the same payload loaddata for any robhold or stationary tool being active. This saves the time (for example, during commissioning).

4.7.2 Editing the payload data *Continued*

For more information about PayLoadsInWristCoords, see Technical reference manual - System parameters and Technical reference manual - RAPID Instructions, Functions and Data types.
4.7.3 Editing the payload declaration

4.7.3 Editing the payload declaration

Overview

Use the declaration to change how the payload variable can be used in the program's modules.

Displaying the payload declaration

	Action
1	On the ABB menu, tap Jogging.
2	Tap Payload to see the list of available payloads.
3	Tap the payload you want to edit, then tap Edit.
4	In the menu, tap Change declaration.
5	The payload's declaration appears. See <i>Creating a payload on page 177</i> .



If you change the name of a payload after it is referenced in any program you must also change all occurrences of that payload's name.

4.7.4 Deleting a payload

4.7.4 Deleting a payload

Deleting a payload

For more information about deleting a **payload**, see *Deleting a data instance on page 147*.

4.8 Testing

4.8.1 About the automatic mode

What is the automatic mode?

In automatic mode the safety function of the three-position enabling device is bypassed so that the manipulator is allowed to move without human intervention.



Prior to selecting automatic mode, any suspended safeguards shall be returned to their full functionality.

Tasks normally performed in the automatic mode

The following tasks are normally performed in automatic mode:

- Starting and stopping processes.
- · Loading, starting, and stopping RAPID programs.
- Returning the manipulator to its path when returning to operation after an emergency stop.
- · Backing up the system.
- · Restoring backups.
- · Cleaning tools.
- Preparing or replacing work objects.
- Performing other process oriented tasks.

Limitations in automatic mode

Jogging is not possible in automatic mode. There may be other specific tasks that should not be performed in automatic mode.

Consult your plant or system documentation to find out which specific tasks should not be performed in automatic mode.

Active safeguard mechanisms

For IRC5, both the general mode stop (GS) mechanisms, the automatic mode stop (AS) mechanisms, and the superior stop (SS) are all active while operating in automatic mode.

Coping with process disturbances

Process disturbances may not only affect a specific manipulator cell but an entire chain of systems even if the problem originates in a specific cell.

Extra care must be taken during such a disturbance since that chain of events may create hazardous operations not seen when operating the single manipulator cell. All remedial actions must be performed by personnel with good knowledge of the entire production line, not only the malfunctioning manipulator.

183

4.8.1 About the automatic mode *Continued*

Process disturbance examples

A manipulator picking components from a conveyer might be taken out of production due to a mechanical malfunction, while the conveyer must remain running in order to continue production in the rest of the production line. This means, of course, that extra care must be taken by the personnel preparing the manipulator in close proximity to the running conveyor.

A welding manipulator needs maintenance. Taking the welding manipulator out of production also means that a work bench as well as a material handling manipulator must be taken out of production to avoid personnel hazards.

4.8.2 About the manual mode

4.8.2 About the manual mode

What is the manual mode?

The manual mode is used when programming and for program verification. In manual mode the manipulator movement is under manual control. The three-position enabling device must be pressed to activate the motors of the manipulator, that is, enabling movement.

There are two manual modes:

- Manual reduced speed mode, usually called manual mode.
- Manual full speed mode (not available in USA or Canada).

What is the manual reduced speed mode?

In manual reduced speed mode the movement of the TCP is limited to 250 mm/s. In addition, there is a limitation on the maximum allowed speed for each axis. These axis limitations are robot dependent and cannot be changed.

The three-position enabling device must be pressed to center-position to activate the motors of the manipulator.



Wherever possible, the manual mode of operation shall be performed with all persons outside the safeguarded space.

What is the manual full speed mode?

The manual full speed mode is used for program verification only.

In manual full speed mode, the manipulator can move in programmed speed but only under manual control.

In manual full speed mode the initial speed limit is up to, but not exceeding, 250 mm/s. This is achieved by limiting the speed to 3% of the programmed speed. Through manual control the speed can be increased up to 100%.

The three-position enabling device must be pressed to center-position and the hold-to-run button pressed to enable program execution.



Wherever possible, the manual mode of operation shall be performed with all persons outside the safeguarded space.

4.8.2 About the manual mode *Continued*

Note that the manual full speed mode is optional and therefore not available in all robots.

Note

As per the updated standard, ISO 10218-1:2011 *Robots and robotic devices* – *Safety requirements for industrial robots* – *Part 1 Robots*, the following adaptations are made to the manual full speed mode.

- Resetting the speed to 250 mm/s every time the three-position enabling device is re-initiated by placing the switch in the center-enabled position after either having been released or fully compressed.
- Editing RAPID programs and jogging the manipulator are disabled.

Bypassed safeguard mechanisms

Automatic mode safeguarded stop (AS) mechanisms are bypassed while operating in manual mode.

The three-position enabling device

In **manual mode**, the motors of the manipulator are activated by the three-position enabling device on the FlexPendant. This way, the manipulator can only move as long as the device is pressed.

The three-position enabling device is designed so that its push-button must be pressed just half-way to activate the motors of the manipulator. The manipulator cannot be operated in the fully-in and fully-out positions.

The hold-to-run function

The hold-to-run function allows stepping or running a program in manual full speed mode. For safety reasons, it is necessary to keep pressing both the three-position enabling device and the **Start** button.

Note that jogging does not require the hold-to-run function, regardless of operating mode. The hold-to-run function can also be activated for manual reduced speed mode.

Tasks normally performed in manual reduced speed mode

The following tasks are normally performed in manual reduced speed mode.

- Jogging the manipulator back on its path when returning to operation after an emergency stop
- Correcting the value of I/O signals after error conditions
- Creating and editing RAPID programs
- Starting, stepping, and stopping program execution, for example while testing a program
- Tuning programmed positions

4.8.2 About the manual mode *Continued*

Tasks normally performed in manual full speed mode

As per the standard, ISO 10218-1:2011, the following tasks can be performed in the manual full speed mode.

- Starting and stopping program execution for final program verification
- Stepping program execution
- Setting speed (0–100%)
- Setting program pointer (to Main, to routine, to cursor, to service routine, etc.)

The following tasks cannot be performed in the manual full speed mode:

- Changing system parameter values
- Editing system data

4.8.3 Using the hold-to-run function

4.8.3 Using the hold-to-run function

When to use the hold-to-run function

The hold-to-run function is used to run or step programs in manual full speed mode, in combination with the three-position enabling device.

In order to run a program in manual full speed mode it is necessary, for safety reasons, to keep pressing both the three-position enabling device and the hold-to-run button. This hold-to-run function also applies when stepping through a program in manual full speed mode.

Some versions of the FlexPendant have a separate hold-to-run button (thumb button). If there is no separate hold-to-run button, the program execution buttons are used as hold-to-run buttons. When **Start**, **Forward**, and **Backward** buttons are used like this (press and hold) they are referred to as hold-to-run buttons.

Operating mode	Function
Manual reduced speed mode	Normally, hold-to-run has no effect in the manual reduced speed mode.
	However, it is possible to activate for manual reduced speed mode by changing a system parameter.
Manual full speed mode	Pressing hold-to-run AND pressing the three-position en- abling device enables running a program. It may be run continuously or step-by-step.
	Releasing hold-to-run in this mode immediately stops ma- nipulator movement as well as program execution. When pressing it again, execution is resumed from that position.
Automatic mode	Hold-to-run is not used in automatic mode.



If the FlexPendant has a separate hold-to-run button, pressing both that button and the **Start** button will stop the program execution.

Using the hold-to-run function

This instruction details how use the hold-to-run function in manual full speed mode.

	Action
1	Press the three-position enabling device on the FlexPendant.
2	 Choose execution mode by pressing and holding either: Start (continuous program execution) or thumb button Forward (step-by step program execution forwards) Backward (step-by step program execution backwards)
3	If Start or the thumb button was pressed, then the program execution continues as long as the button is pressed. If Forward or Backward was pressed, the program is executed step-by-step by alternately releasing and pressing the Forward/Backward button. Note that the button must be pressed and held until the instruction is executed. If the button is released, program execution will stop immediately!
4	If the three-position enabling device is released, intentionally or by accident, the complete procedure must be repeated to enable running.

4.8.4 Running the program from a specific instruction

Overview

When starting a program the execution starts from the program pointer. To start from another instruction, move the program pointer to the cursor.



When execution is started the robot will move to the first programmed position in the program. Make sure that the robot does not risk running into any obstacles.

Running the program from a specific instruction

	Action
1	On the ABB menu, tap Program Editor.
2	Tap on the program step where you want to start, then tap Debug and then PP to Cursor .
3	DANGER Make sure that no personnel are in the safeguarded space.
4	Press the Start button on the FlexPendant.

4.8.5 Running a specific routine

4.8.5 Running a specific routine

Overview		
	Wh fro	nen starting a program the execution starts from the program pointer. To start m another routine, move the program pointer to the routine.
Prerequisites		
	In o the	order to run a specific routine the module with the routine must be loaded and controller must be in manual stopped mode.
Running a specifi	c routi	ine
	Thi poi	is procedure describes how to run a specific routine by moving the program inter.
		Action
	1	On the ABB menu, tap Program Editor.
	2	Tap Debug and then PP to Routine to place the program pointer at the start of the routine.
	3	Press the Start button on the FlexPendant.
Related information	on	
	Но	w to run a service routine is described in <i>Running a service routine on page 194</i> .
	The	e same method can be used to run a specific routine in the task scope. See
	Ru	nning a service routine on page 194 for detailed information.

4.8.6 Stepping instruction by instruction

4.8.6 Stepping instruction by instruction

Overview

In all operating modes the program may be executed step by step forwards or backwards.

Stepping backwards is limited, see *Technical reference manual* - *RAPID Overview* for more details.

Select step mode

This section details how to select step mode. Stepping can be done in three ways; step in, step over, and motion step.

	Action	Information
1	Select step mode using the Quickset menu.	Described in <i>Quickset menu, Step Mode on page 63</i> .

Stepping

This section details how to step forwards and backwards.

If you want to step	then press
forward	Forward button on FlexPendant
backward	Backward button on FlexPendant

Limitations of backward execution

There are some restrictions for the backward execution:

- When stepping backwards through a MoveC instruction, the execution does not stop in the circular point.
- It is not possible to step backwards out of a IF, FOR, WHILE and TEST statement.
- It is not possible to step backwards out of a routine when reaching the beginning of the routine.
- There are instructions affecting the motion that cannot be executed backwards (e.g. ActUnit, ConfL and PDispOn). If attempting to execute these backwards, an alert box will inform you that this is not possible.

Backward execution behavior

When stepping forward though the program code, a program pointer indicates the next instruction to execute and a motion pointer indicates the move instruction that the robot is performing.

When stepping backward though the program code, the program pointer indicates the instruction above the motion pointer. When the program pointer indicates one move instruction and the motion pointer indicates another, the next backward movement will move to the target indicated by the program pointer, using the type of movement and speed indicated by the motion pointer.

4.8.6 Stepping instruction by instruction Continued

Example of backward execution

The following example illustrates the behavior when stepping backwards through move instructions. The program pointer and motion pointer helps you keep track of where the RAPID execution is and where the robot is.

MoveL, MoveJ, and MoveC are move instructions in RAPID, see Technical reference manual - RAPID Instructions, Functions and Data types.

		■ ∨ 🖗	Manual RW6.05 (10.1	40.60.195)	Gua Stoj	rd Stop oped (Speed 100%)	Z	X
	٩N	lewProgramName	in T_ROB1/₩	lainModule/	main			
		Tasks and Prog	jrams 🔻	M	odules	-	Routines	•
	5	CONST r	obtarget	p40:=	[[119.	73,-680.4	4,575.86]	<u>, N</u> .
	6	CONST r	obtarget	p50:=	[[119.	73,-637.9	5,515.87)	,[0,-
	7	CONST r	obtarget	p60:=	[[97.4	3,-637.95	,575.87],	[0.5
	8	CONST r	obtarget	p80:=	[[146.	54,-582.1	4,628.12]	,[0.
	9	CONST r	obtarget	p70:=	[[146.	54,-582.1	4,628.12]	,[0.
	10	PROC ma	in()					
	11	MoveJ	p10, v1	000, z	50, to	010;		
C	12	🔨 MoveL	p20, v1	000, z	50, to	010;		
\sim	13	MoveC	p30, p4	0, v30	0, z10	, tool0;		
B	NN.	Mover	p50 , v1	000, z	50, to	010;		
	15	MoveL	p60, v1	000, z	50, to	010;		
9	16	ENDPROC						
	17	ENDMODULE						
	Ac	ld 🔺		<u>م</u>		Modify	Hide	
	In	struction	Ealt	D	ebug	Position	Declarati	ons
	2 :	Jogging Prod	uction low	DB1 Module	ogram Ita		RO L	B_1

en0400001204

A	Program pointer
В	Motion pointer
С	Highlighting of the robtarget that the robot is moving towards, or already has reached.

When	then
stepping forward until the robot is in p50	the motion pointer will indicate p50 and the program pointer will indicate the next move instruction (MoveL $p60).$
pressing the Back- ward button once	the robot will not move but the program pointer will move to the previous instruction ($MoveC p30$, $p40$). This indicates that this is the instruction that will be executed the next time Backward is pressed.
pressing the Back- ward button again	the robot will move to p40 linearly with the speed v300. The target for this movement (p40) is taken from the MoveC instruc- tion. The type of movement (linear) and the speed are taken from the instruction below (MoveL p50). The motion pointer will indicate p40 and the program pointer will move up to MoveL p20.

4.8.6 Stepping instruction by instruction *Continued*

When	then
pressing the Back- ward button again	the robot will move circularly, via p30, to p20 with the speed v1000. The target p20 is taken from the instruction $Movel p20$. The type of movement (circular), the circular point (p30) and the speed are taken from the $MoveC$ instruction.
	The motion pointer will indicate p40 and the program pointer will move up to Movel p10.
pressing the Back- ward button again	the robot will move linearly to p10 with the speed v1000. The motion pointer will indicate p10 and the program pointer will move up to MoveJ p10.
pressing the Forward button once	the robot will not move but the program pointer will move to the next instruction ($Movel p20$).
pressing the Forward button again	the robot will move to p20 with the speed v1000.

4.9.1 Running a service routine

4.9 Service routines

4.9.1 Running a service routine

	Service routines perform a number of common services. The service routines available to you depends on your system setup and available options.
	The service routines described in this manual are some of the commonly available.
Prerequisites	
	Service routines can only be started in manual reduced speed mode or in manual full speed mode.
	The program must be stopped and there has to be a program pointer.
	It is not possible to call a routine when in synchronized mode.
	If the service routine contains parts that must be carried out in automatic mode, then the program pointer must not be moved manually before starting the service routine. The program pointer should be where the program flow was stopped.
	If a service routine is started in the middle of a stopped movement instruction (that is, before the end position is reached), then the movement will be resumed when the execution of the service routine starts.
	! CAUTION

This section describes how to execute a service routine or another routine in the task scope using **Call Routine**.

	Action
--	--------

1 On the ABB menu tap Program Editor.

4.9.1 Running a service routine *Continued*

	Action				
	On the Debug menu tap Call Routine .				
	Manual Motors On ST MySystem (RSTEST4) Stopped (2 of 2) (Speed 100%)				
Hyprogram1 in T_ROB1/MainModule/main			_		
	Tasks and Programs 🔻 Modu	ıles	▼	Routines	~
	<pre>9 PROC main()</pre>	\land	PP to Main	PP to Cursor	
	149 MoveL p10, v100	0,	PP to Routine	Cursor to PP	70
	¹¹ MoveL p20, v100,	z:	Cursor to MP	Go to position	1
	¹² WaitDI DI1, 1;		Call Routine	Cancel Call Rout.	11
	¹³ MoveL p30, v100,	Z	View Value	Check Program	11
	14 WaitDI DI2, 1;		View System Data	Search Routine	
	¹⁵ MoveL p40, v100,	z	view System Data	Search Routine	-11
	¹⁶ MoveL p10, v100,	z:			
	17 ENDPROC				
	18 ENDMODULE		-		
	Add Edit Del	bug	Position	Hide Declarations	
			1 control	ROB_1	
	MainMod			16 12	à
	en040000884				
	Call Service Routine		Motors On Stopped (2 of 2) (Spee	물 포 d 100%)	×
	Selected routine: Select the routine to call and tap 'Go to'.				
	Bat_Shutdown		CalPendelum	1	to 8 of 8
	Commutation	<u>ا</u>	Linked_m		
	E LoadIdentify		 ManLoadIdenti	ifv	
	ServiceInfo		SkipTaskExec	,	
	View T_ROB1 MainMod		Go to	Cance V	
	en0400000885				
L	Tap a service routine and then tap Go to	n			

4.9.1 Running a service routine *Continued*

Action

5 Press the **Start** button on the FlexPendant and follow the instructions displayed on the FlexPendant.

After execution of the routine the task is stopped and the program pointer is returned to where it was before the service routine started.

Press **Cancel Call Rout** if you need to interrupt the routine before it has finished executing. Before resuming normal program flow, however, you must see to it that the robot is correctly positioned. If the interrupted routine has moved it, you will need to take actions to return the robot to its position. See *Returning the robot to the path on page 238* for further information.



Do not execute a service routine in the middle of a move or a weld.

If you execute a service routine in the middle of a movement, the unfinished movements will be completed before the called routine is executed. This can result in an unwanted movement.

If possible, step and complete the interrupted movement before the service routine is called. Otherwise save the current movement by adding StorePath and RestoPath in the service routine. The movement will then be completed after the service routine has ended and the program starts again.

However, it is not possible to save more than one interrupted movement each time as wanted, if the service routine would be called from an error handler with StorePath and RestoPath.

Limitations

Besides service routines, **Call Routine** applies to all routines with the following criteria:

- Must be a procedure with empty parameter list. This means not a function and not a trap routine.
- Must be in the task scope, not local. If the procedure is local in a module the scope is restricted to that module, and the procedure is not visible from the task level.
- Must be in a loaded module, not installed. (Check the system parameter **Installed** in the type *Automatic Loading of Modules* in the *Controller* topic.)

Related information

Battery shutdown service routine on page 198. LoadIdentify, load identification service routine on page 202. Service Information System, ServiceInfo service routine on page 201. Calibration Pendulum, CalPendulum service routine on page 200.

4.9.1 Running a service routine *Continued*

For more information about StorePath and RestoPath, see Technical reference manual - RAPID Instructions, Functions, and Data types.

4.9.2 Battery shutdown service routine

4.9.2 Battery shutdown service routine

When to use this service routine

For SMB units with 2-pole battery contact, it is possible to shut down the battery backup of the serial measurement board to save battery power during transportation or storage. This is the *BatteryShutDown* service routine.

For SMB units with 3-pole contact, this function shall not be used since the power consumption is so low that it is not needed.



The service routine was earlier called Bat_Shutdown.

BatteryShutDown

When the system is powered on again, the function is reset. The revolution counters will be lost and need an update but the calibration values will remain.

The consumption in ordinary shutdowns is then approximately 1 mA. When using sleep mode the consumption is reduced to 0.3 mA. When the battery is nearly discharged, with less then 3 Ah left, an alert is given on the FlexPendant and the battery should be replaced.



Before starting the service routine *BatteryShutDown*, run the robot to its calibration position. This will make it easier to recover after the sleep mode.

Related information

How to start a service routine is described in Running a service routine on page 194.

How to update the revolution counters is described in *Updating revolution counters* on page 280.

4.9.3 Axis Calibration service routine

4.9.3 Axis Calibration service routine

Description of the service routine

Axis Calibration is a standard calibration method for calibration of ABB robots and is the most accurate method for the standard calibration.

The following routines are available for the Axis Calibration method:

- Fine calibration
- Update revolution counters
- Reference calibration

The calibration equipment for Axis Calibration is delivered as a toolkit.

The actual instructions of how to perform the calibration procedure and what to do at each step is given on the FlexPendant. You will be guided through the calibration procedure, step by step.

Related information

Running a service routine on page 194.

A more detailed description of the calibration method is given in the product manual for the respective manipulator.

4.9.4 Calibration Pendulum, CalPendulum service routine

4.9.4 Calibration Pendulum, CalPendulum service routine

Description of the service routine

For IRC5, Calibration Pendulum is one of the standard calibration methods for calibration of ABB robots (option).

Two different routines are available for the Calibration Pendulum method:

- Calibration Pendulum II
- Reference calibration

The calibration equipment for Calibration Pendulum is delivered as a complete toolkit, including the *Operating manual - Calibration Pendulum*, which describes the method and the different routines further.

Related information

Running a service routine on page 194.

Calibration Pendulum is described in full in the manual *Operating manual - Calibration Pendulum*. Specific information for each robot is described in the robot's product manual. 4.9.5 Service Information System, ServiceInfo service routine

4.9.5 Service Information System, ServiceInfo service routine

When to use this service routine

ServiceInfo is a service routine based on Service Information System, SIS, a software function which simplifies maintenance of the robot system. It supervises the operating time and mode of the robot, and alerts the operator when a maintenance activity is scheduled.

ServiceInfo

Maintenance is scheduled by setting the system parameters of the type *SIS Parameters*. All system parameters are described in *Technical reference manual* - *System parameters*.

More details about SIS is described in *Operating manual - Service Information System*.

Supervised functions

The following counters are available:

- Calender time counter
- Operation time counter
- Gearbox operation time counters
- Moved distance counter¹

I The moved distance value cannot be reset through the service routines.

Counters are reset when maintenance has been performed.



Resetting counters cannot be undone.

The counter status is displayed after running the ServiceInfo routine for maintenance. Status **OK** indicates that no service interval limit has been exceeded by that counter. Status **NOK** indicates that service interval limit has been exceeded by that counter.

Related information

Running a service routine on page 194.

4.9.6 LoadIdentify, load identification service routine

4.9.6 LoadIdentify, load identification service routine

When to use this service routine

The service routine LoadIdentify is used to automatically identify the data of loads mounted on the robot. The data can also be entered manually, but this requires information that may be difficult to calculate.

To run LoadIdentify, there are a number of things to consider. These are described on the following pages. There is also information on error handling and limitations described in this chapter.



D Important to always define the act

It is important to always define the actual tool load and, when used, the payload of the robot (for example, a gripped part). Incorrect definitions of load data can result in overloading of the robot mechanical structure. There is also a risk that the speed in manual reduced speed mode can be exceeded.

When incorrect load data is specified, it can often lead to the following consequences:

- The robot may not use its maximum capacity.
- Impaired path accuracy including a risk of overshooting.
- Risk of overloading the mechanical structure.

The controller continuously monitors the load and writes an event log if the load is higher than expected. This event log is saved and logged in the controller memory.



When using LoadId or LoadIdentification to identify the load of a tool or payload with unknown mass, the mass is estimated using the manipulator and the result can deviate from the actual mass. This is due to tolerances and variations between mechanical units. This does not necessarily mean that the identified payload or tool will cause issues with motion performance. If a very accurate value for the mass is required, it is recommended to weigh the tool or payload and use known mass in the identification.

4.9.6 LoadIdentify, load identification service routine *Continued*

LoadIdentify

LoadIdentify can identify the tool load and the payload. The data that can be identified are mass, center of gravity, and moments of inertia.



en0500001535

Α	Upper arm load
В	Tool load
С	Payload

Before running the load identification for the payload, make sure that the tool load data is correctly defined first, for example by running LoadIdentify for the tool.

To identify the mass of B and C, axis 3 has to perform some movement. This means that to identify the mass, the upper arm load A must be known and correctly defined first.

To improve accuracy if the upper arm load A is mounted, input the known mass of B and C and choose the *known mass* method when identifying.

Configuration angles

To perform the identification the robot moves the load after a specific pattern and calculates the data. The axes that move are 3, 5 and 6. At the identification position, the motion for axis 3 is approximately ± 3 degrees and for axis 5 it is approximately ± 30 degrees. For axis 6 the motion is performed around two configuration points. The optimum value for the configuration angle is either ± 90 degrees or -90 degrees.



en0500001537

А	Configuration 1 (start position)
В	Configuration angle

4.9.6 LoadIdentify, load identification service routine *Continued*

C Configuration 2

LoadIdentify with arm loads mounted

The best way to perform load identification is to use a robot with no arm loads mounted. If this is not possible, good accuracy can still be achieved. Consider, for example, the robot in the figure below, which has arc welding equipment mounted on it.



en0500001536

Α	Cable 1
В	Load 1
С	Cable 2
D	Load 2

If we want to use load identification to find the data of load 2, the most important thing to remember is to make sure that the upper arm load is correctly defined, in particular its mass and center of gravity along the robot arm. The arm load includes everything that is mounted on the robot, except tool load and payload. In the figure above, cable 1, cable 2, and load 1 are included in the arm load, the total weight and center of gravity have to be calculated.

When performing the load identification, cable 2 should be disconnected since it will otherwise put an extra force on load 2. When identifying load 2 with such a force present, the result may differ considerably from the correct load. Ideally, cable 2 should be disconnected from load 2 and fastened on the upper arm. If this is not possible, the cable can also be disconnected at load 1 and fastened to the upper arm in such a way that the resulting force on load 2 is minimized.

Prerequisites for tool loads

Before running the LoadIdentify service routine for a tool load, make sure that:

- · The tool is selected in the jogging menu.
- The tool is correctly mounted.
- Axis 6 is close to horizontal.
- The upper arm load is defined, if the tool mass is to be identified.
- The axes 3, 5, and 6 are not close to their corresponding working range limits.

4.9.6 LoadIdentify, load identification service routine Continued

- The speed is set to 100%.
- The system is in manual mode.

Note that LoadIdentify cannot be used for tool0.

Prerequisites for payloads

Before running the LoadIdentify service routine for a payload, make sure that:

- The tool and payload are correctly mounted.
- Axis 6 is close to horizontal.
- The tool load is known (run LoadIdentify for the tool first).
- The upper arm load is defined, if the payload mass is to be identified.
- When using a moving TCP, the tool must be calibrated (TCP).
- When using a stationary TCP, the corresponding work object must be calibrated (user frame and object frame).
- The axes 3, 5, and 6 are not close to their corresponding working range limits.
- The speed is set to 100%.
- The system is in manual mode.

Note that LoadIdentify cannot be used for load0.

Running LoadIdentify

To start the load identification service routine you must have an active program in manual mode and the tool and payload that you want to identify must be defined and active in the **Jogging** window.



Always run load identification with cold motors (no warm-up).

	Action	Information
1	Start LoadIdentify from the Program Editor . Press the three-position enabling device and then press the Start button on the FlexPendant.	How to start service routines is described in <i>Running a</i> <i>service routine on page 194</i> .
2	Tap OK to confirm that current path will be cleared and that the program pointer will be lost.	Tap Cancel and then Cancel Call Rout to quit the service routine without loosing the program pointer.
3	Tap Tool or Payload.	
4	Tap OK to confirm that the correct tool and/or payload is active in the jogging menu and that the tool load/pay- load is correctly mounted.	If it is not correct, release the three-position enabling device and select the correct tool/payload in the Jogging menu. Then return to Load- ldentify, press the three-pos- ition enabling device, and press Start. Tap Retry and confirm that the new tool/payload is correct.

4.9.6 LoadIdentify, load identification service routine *Continued*

	Action	Information
5	When identifying tool loads, confirm that the tool is active. When identifying payloads, confirm that the payload's tool is active and calibrated.	See <i>step 4.</i>
6	When identifying payloads with stationary TCP, confirm that the correct work object is active and (preferably) calibrated. If it is correct, tap OK to confirm.	See step 4.
7	Select identification method. If you select the method where the mass is assumed to be known, remember that the tool/payload that you use must have the correct mass defined. Tap OK to confirm.	
8	Select configuration angle. The optimum is +90 or -90 degrees. If this is impossible, tap Other and set the angle. The minimum is +30 or -30 degrees.	
9	If the robot is not in a correct position for load identifica- tion, you will be asked to jog one or more axes roughly to a specified position. When you have done this tap OK to confirm. If the robot is still not in a correct position for load identi- fication, the robot will slowly move to the correct position. Press Move to start the movement.	Axis 1 to 3 must not be more than 10 degrees from pro- posed position.
10	The robot can go through the load identification move- ments slowly before performing the load identification (slow test). Tap Yes if you want a slow test and No to proceed to the identification.	This is useful for ensuring that the robot will not hit anything during the identific- ation. However, this will take a lot longer time. Note If the load identification is planned to be run in manual full speed, then the slow test is required before the actual measurement can start.
11	The setup for load identification is now complete. To start the motion, switch to Automatic mode and Motors On. Then tap Move to start the load identification movements.	
12	When the identification is finished, switch back to manual mode, press the three-position enabling device and the Start button. Tap OK to confirm.	
13	The result of the load identification is now presented on the FlexPendant. For robots that support the <i>Load diagram check</i> function- ality, there is a message if the load is approved or not, and an Analyze button to view more information.	See Load diagram check on page 207.
14	Tap Yes to update the selected tool or payload with the identified parameters. Tap No to exit LoadIdentify without saving the parameters.	

4.9.6 LoadIdentify, load identification service routine Continued

Load diagram check

For robots that support the *Load diagram check* functionality, the combination of the arm load, tool and payload, will be assessed against the load diagram. The ratings of total handling weight as well as of the center of gravity distance to the load diagram in Z and L directions will be provided both for wrist-up and wrist-down configurations.

There is a message if the load is approved or not, and an **Analyze** button to view more information.

- Load approved
- Load not approved
- Load approved only with wrist down

Running LoadIdentify with ModalPayLoadMode deactivated

When the system parameter ModalPayLoadMode is deactivated, set to 0, LoadIdentify will identify the tool load and the total load. It is no longer possible to define the payload.

With ModalPayLoadMode deactivated it is possible to use the \TLoad argument in movement instructions. The \TLoad argument describes the total load used in the movement. The total load is the tool load together with the payload that the tool is carrying. If the \TLoad argument is used, then the loaddata in the current tooldata is not considered. For more information about ModalPayLoadMode in movement instructions, see the section MoveL in *Technical reference manual - RAPID Instructions, Functions and Data types*.

To start the load identification service routine you must have an active program in manual mode and the tool and payload that you want to identify must be defined and active in the **Jogging** window.



Always run load identification with cold motors (no warm-up).

	Action	Information
1	Start LoadIdentify from the Program Editor . Press the three-position enabling device and then press the Start button on the FlexPendant.	How to start service routines is described in <i>Running a</i> <i>service routine on page 194</i> .
2	Tap OK to confirm that current path will be cleared and that the program pointer will be lost.	Tap Cancel and then Cancel Call Rout to quit the service routine without loosing the program pointer.
3	Tap OK to continue with the LoadIdentify process.	The selection to update the tool load or the total load is done at a later stage.

4.9.6 LoadIdentify, load identification service routine *Continued*

		1
	Action	Information
4	Tap OK to confirm that the correct tool and/or total load is active in the jogging menu and that the tool load/total load is correctly mounted.	If it is not correct, release the three-position enabling device and select the correct tool/payload in the Jogging menu. Then return to Load- ldentify, press the three-pos- ition enabling device, and press Start. Tap Retry and confirm that the new tool/payload is correct.
5	When identifying tool loads, confirm that the tool is active.	See step 4.
6	Select identification method. If you select the method where the mass is assumed to be known, remember that the tool/total load that you use must have the correct mass defined. Tap OK to confirm.	
7	Select configuration angle. The optimum is +90 or -90 degrees. If this is impossible, tap Other and set the angle. The minimum is +30 or -30 degrees.	
8	If the robot is not in correct position for load identification, you are asked to jog one or more axes roughly to a spe- cified position. When you have done this tap OK to con- firm. If the robot is still not in a correct position for load identi- fication, the robot will slowly move to the correct position. Press Move to start the movement.	Axis 1 to 3 must not be more than 10 degrees from pro- posed position.
9	The robot can go through the load identification move- ments slowly before performing the load identification (slow test). Tap Yes if you want a slow test and No to proceed to the identification.	This is useful for ensuring that the robot will not hit anything during the identific- ation. However, this will take a lot longer time. Note If the load identification is planned to be run in manual full speed, then the slow test is required before the actual measurement can start.
10	The setup for load identification is now complete. To start the motion, switch to Automatic mode and Motors On. Then tap Move to start the load identification movements.	
11	When the identification is finished, switch back to manual mode, press the three-position enabling device and the Start button. Tap OK to confirm.	
12	The result of the load identification is now presented on the FlexPendant. Tap Tool if you want to update the se- lected tool, tap Loaddata if you want to update the total load, or tap No if you want to quit without saving.	
13	If Loaddata is selected it is possible to update the total load to either an existing or a new loaddata persistent variable.	

4.9.6 LoadIdentify, load identification service routine *Continued*

Error handling

If the three-position enabling device is released during the load identification (before the movements start), the routine can always be restarted by pressing the three-position enabling device again and then pressing the Start button.

If an error should occur during the load identification movements, the routine must be restarted from the beginning. This is done automatically by pressing Start after confirming the error. To interrupt and leave the load identification procedure, tap **Cancel Call Routine** in **Program Editor** debug menu.

Limitations for LoadIdentify

Only tool loads and payloads can be identified with LoadIdentify. Thus arm loads cannot be identified.

If the load identification movements are interrupted by any kind of stop (program stop, emergency stop, etc.), the load identification must be restarted from the beginning. Confirm the error and press **Start** to automatically restart.

If the robot is stopped on a path with program stop and load identification is performed at the stop point, the path will be cleared. This means that no regain movement will be performed to return the robot back to the path.

The load identification ends with an EXIT instruction. That means that the program pointer is lost and must be set to main before starting any program execution.



The tool and/or payload data can be set manually if the load is small (10% or less of the maximum load) or symmetrical, for example if the tool load is symmetrical around axis 6.



If the mass of the tool or payload is unknown, the service routine LoadIdentify can in some cases identify a 0 kg mass. If the load is very small compared to the maximum load for the robot, then a 0 kg mass can be ok. Otherwise, try the following to identify the mass.

- Check that the arm loads are correctly defined and redo the identification.
- Find the weight of the load in some other way and perform a load identification with known mass to remove the dependency on arm loads.

LoadIdentify for 4-axis palletizing and SCARA robots

When running LoadIdentify on a robot with 4 instead of 6 axes, there are some differences. In this description of the differences the robot type is assumed to be similar to IRB 260, IRB 460, IRB 660, IRB 760, or IRB 910SC.

The main differences are:

- The used axes are:
 - 1 (or 2 for some robot models)
 - 3 (for all robot models)

4.9.6 LoadIdentify, load identification service routine *Continued*

- 6 (or 4 for some robot models)
- Because the first axis (axis 1 or 2) is used, the resulting movements can be large.
- Not all load parameters can be identified.

The first axis (axis 1 or 2) will move approximately ± 23 degrees from its current position. Therefore, the load can move a large distance during the identification. Axes 3 and 6 (or 4) will move as for 6-axis robots. The configuration angle for axis 6 (or 4) works exactly as for 6-axis robots.

Because there is not 6 axes, a 4-axis robot cannot identify all parameters of the load. The following parameters cannot be identified:

- I_x The inertia around the x-axis.
- I_v The inertia around the y-axis.
- m_z The z-coordinate for the center of mass.

However, for this type of robot the above parameters have negligible effect on the motion performance. See the definition of the load coordinate system in the following figure.



xx0900000021



It is possible that the identification procedure fails to estimate the center of gravity if the measured torque data has too high variance. If this happens, it should still be possible to get good results by running the LoadIdentify routine again, preferably with another position of the last axis.

LoadIdentify for 4-axis delta robots

When running LoadIdentify on a delta (picker/packer) robot with 4 axes, there are some differences. In this description of the differences the robot type is assumed to be similar to IRB 360, IRB 365, or IRB 390.

The main differences are:

- The used axes are:
 - 3 (for all robot models)
 - 4 (for all robot models)

4.9.6 LoadIdentify, load identification service routine Continued

• Not all load parameters can be identified.

Axis 4 will move approximately +60 degrees from its current position. Therefore, the load will move a large rotational distance during the identification.

Because there is not 6 axes, a 4-axis delta robot cannot identify all parameters of the load. In fact, only the following parameters are possible to identify:

- *Iz*, the inertia around the z-axis.
- *m*, the total mass.

However, for this type of robot the total mass and the inertia around the z-axis is enough to give good motion performance. All other loaddata parameters will be set to zero.



An x,y-offset in the mass center (mx,my) should not be manually entered since this information will be included in the identified *Iz*.

(However, if known, an offset mz (z-coordinate for the center of mass) can be manually entered to slightly enhance motion performance.)

See the resulting loaddata from Loadldentify for 4-axis delta robots in the following figure.



xx2100002581

4.9.6 LoadIdentify, load identification service routine *Continued*

LoadIdentify for 5-axis delta robots

When running LoadIdentify on a delta (picker/packer) robot with 5 axes, there are some differences. In this description of the differences the robot type is assumed to be similar to IRB 365 or IRB 390.

The main differences are:

- The used axes are:
- 3 (for all robot models)
 - 4 (for all robot models)
 - 5 (for all robot models)
- Not all load parameters can be identified.

Axis 4 will move approximately +45 degrees from its current position. Therefore, the load will move a large rotational distance during the identification.

Axis 5 will move approximately +35 degrees from its current position. Therefore, the load will move a large rotational distance during the identification.

Instead of the usual 0-90 degree configuration movement, the axis 5 will move ± 45 degrees.

Because there is not 6 axes, a 5-axis delta robot cannot identify all parameters of the load. The following parameters are possible to identify:

- *ly*, the inertia around the y-axis.
- *Iz*, the inertia around the z-axis.
- *mx*, the x-coordinate for the center of mass.
- *mz*, the z-coordinate for the center of mass.
- *m*, the total mass.

4.9.6 LoadIdentify, load identification service routine *Continued*

See the resulting <code>loaddata</code> from <code>LoadIdentify</code> for a 5-axis delta robot in the following figure.



xx2100002620

Related information

It is also possible to include LoadIdentify in a program by using RAPID instructions. See LoadID in *Technical reference manual - RAPID Instructions, Functions and Data types*.

How to enter the data manually is described in *Editing the tool data on page 161*, and *Editing the payload data on page 179*.

The product manual for the robot contain information on how and where to mount the loads.

Load identification for positioners is done with the service routine ManLoadIdentify, see *Product manual - Product.ProductName /D2009*.

How to define the system parameters for arm loads is described in *Technical reference manual - System parameters*.

4.9.7 Brake check service routine

4.9.7 Brake check service routine

Overview

The BrakeCheck service routine is used to verify whether the mechanical brakes work correctly.

The BrakeCheck service routine is included in the RobotWare installation if the controller does not have SafeMove or EPS option.



If the controller has SafeMove or EPS option, the RobotWare installation includes Cyclic Brake Check service routine. For more details, see Application manual - Functional safety and SafeMove2.

While running the BrakeCheck service routine the brakes are tested in consecutive order and each test takes 10-15 seconds.

Prerequisites for running the BrakeCheck service routine

Following are the prerequisites for running the BrakeCheck service routine:

- The robot and all additional axes must be moved to a safe and relaxed position (away from people, equipment and not too much stretched) before performing a brake check. Normally the robot moves only a few centimeters during the brake check.
- Move the robot to a stop point before performing a brake check.
- The brake check can be performed only at normal execution level (not from a trap routine, error handler, event routine, or store path level).

Exclude individual axes from the brake check

It is possible to exclude individual axes from the brake check. For this, set the value of system parameter Deactivate Cyclic Brake Check for axis to On. For more details, see Configure system parameters on page 220.

Running the brake check

Following are the two ways to initiate a BrakeCheck service routine:

- Call the BrakeCheck service routine from FlexPendant. The controller must be in manual mode.
- Call the procedure BrakeCheck from the RAPID program.



While the brake check routine is active, do not change the speed from the FlexPendant and do not use the instructions VelSet, AccSet, SpeedRefresh, or any other instruction that affects the motion performance in trap routines or event routines.

4.9.7 Brake check service routine Continued



The RAPID function IsBrakeCheckActive can be used to check if BrakeCheck is active.

Interrupt the brake check

It is not recommended, but it is possible to stop the execution while running a brake check.

If the brake check is interrupted, it will be resumed when the program execution starts again. The brake check can be resumed up to 3 times.

Brake maintenance

Brake maintenance is a feature in the brake check functionality.

The BrakeCheck routine automatically detects if maintenance of the mechanical brakes is needed and activates the *Brake maintenance* functionality during execution. *Brake maintenance* applies the brake and turns the motor shaft 1 radian five times, which gives a movement of the robot arm of less than 1 degree.

There are event logs that tell if *Brake maintenance* is needed, and if it has been run.

For more information see parameter *Brake Maintenance*, type *General Rapid*, topic *Controller*, in *Technical reference manual - System parameters*.

Event logs

When BrakeCheck is executed, the following event logs will be shown:

Event log	Title
10272	Brake Check Done
10273	Brake Check Started

If there is a problem with one or several mechanical brakes, an event log that is describing which mechanical unit and which axis that has bad brakes will be shown:

Event log	Title
37234	Brake Performance Warning
37235	Brake Performance Error

Brake check for MultiMove systems



Make sure that all mechanical units are standing still before starting a BrakeCheck routine.

One of the motion tasks call the routine BrakeCheck to perform brake check for all mechanical units in all tasks.

The brake check must not be performed while any tasks are in synchronized mode (coordinated movement). It is necessary to synchronize all motion tasks with WaitSyncTask instructions before and after the actual brake check. If running a

4.9.7 Brake check service routine *Continued*

movement instruction when one motion task is execution a BrakeCheck, you will have an error (41888) and all execution will stop. Instruction ExitCycle is also forbidden to use during an active BrakeCheck.

The RAPID function IsBrakeCheckActive can be used to check if there is an ongoing BrakeCheck.

Only one call to BrakeCheck can be done at a time. This is checked by the service routine and if more than one RAPID task or client try to execute the routine, you will have an error (41886).

Program example

```
T_ROB1
PERS tasks task_list{2} := [ ["T_ROB1"], ["T_ROB2"] ];
VAR syncident sync1;
VAR syncident sync2;
. . .
IF PLC_di1_DO_CBC = 1 THEN
 WaitSyncTask sync1, task_list;
 BrakeCheck;
 WaitSyncTask sync2, task_list;
ENDIF
T_ROB2
PERS tasks task_list{2} := [ ["T_ROB1"], ["T_ROB2"] ];
VAR syncident sync1;
VAR syncident sync2;
. . .
IF PLC_di1_DO_CBC = 1 THEN
 WaitSyncTask sync1, task_list;
  ! Wait for T_ROB1 to be ready with BrakeCheck
 WaitSyncTask sync2, task_list;
ENDIF
```

Brake check on additional axes

To be able to run brake check on additional axes, the parameter *Max Static Arm Torque* (in topic *Motion* and type *Brake*) needs to be calculated for the additional axis and entered into the configuration. Brake check uses this value when testing the brake at error-level.

The parameter should be the maximum static torque that the brake needs to withstand when the additional axis is positioned in maximum gravity. The following formula should be used:

Max Static Arm Torque = (M*L*g)/n
4.9.7 Brake check service routine Continued



To calculate the parameter for an axis that has no gravity, for example a track, the below formula may be used:

Max Static Arm Torque = Tbrake min/1.35

Tbrake min for ABB motor units can be found in the product specification for the specific motor unit, see Product specification - Motor Units and Gear Units.

For more information about parameter Max Static Arm Torque, see topic Motion, type Brake in Technical reference manual - System parameters.



Note that the calculated value should be entered in [Nm] and calculated to the motor side.

Description of the I/O setup

Signal configuration

It is possible to configure digital output signals that reflect the status of the mechanical brakes in an open RAPID module. The digital output signals that can be configured are OK, WARNING, ERROR, and ACT (brake check active) for each drive module.

The signal configuration should be done in the RAPID module BC config IO.sys, see Description of the I/O setup on page 217.

The file BC_config_IO.sys can be found in directory hd0a\<active system>\PRODUCTS\RobotWare 6.0x.xxxx\utility/BrakeCheck, and must then be copied to the HOME directory of the active system.



Note

Remember to update the I/O configuration with the digital output signals.

In a MultiMove system you need to define an OK, WARNING, ERROR, and ACT digital output signals for each drive module.

Note

If the signals should keep their values after a power fail, the power fail settings in the system parameters must also be updated, see Description of the I/O setup on page 217.

Continues on next page

4 Programming and testing

4.9.7 Brake check service routine *Continued*

```
Description of the BC_config_IO module
                      MODULE BC_config_IO(SYSMODULE,NOVIEW)
                         PROC BC_config_IO_proc(VAR string user_io_names{*,*})
                           !TPWrite "BC_config_IO_proc";
                           ! Define your own signals. The signal
                           ! names must be signals defined in EIO.cfg
                           ! If 1 drive module
                           user_io_names{1, 1}:="BCACT1";
                           user_io_names{1, 2}:="BCOK1";
                           user_io_names{1, 3}:="BCWAR1";
                           user_io_names{1, 4}:="BCERR1";
                           ! If 2 drive modules
                           !user_io_names{2, 1}:="BCACT2";
                           !user_io_names{2, 2}:="BCOK2";
                           !user_io_names{2, 3}:="BCWAR2";
                           !user_io_names{2, 4}:="BCERR2";
                           ! If 3 drive modules
                           !user_io_names{3, 1}:="BCACT3";
                           !user_io_names{3, 2}:="BCOK3";
                           !user_io_names{3, 3}:="BCWAR3";
                           !user_io_names{3, 4}:="BCERR3";
                           ! If 4 drive modules
                           !user_io_names{4, 1}:="BCACT4";
                           !user_io_names{4, 2}:="BCOK4";
                           !user_io_names{4, 3}:="BCWAR4";
                           !user_io_names{4, 4}:="BCERR4";
                         ENDPROC
                      ENDMODULE
Description of the EIO.cfg file
                      EIO:CFG_1.0:6:1::
                       . . .
                       #
                      EIO_SIGNAL:
                         -Name "BCACT1" -SignalType "DO"
                         -Name "BCOK1" -SignalType "DO"
                         -Name "BCWAR1" -SignalType "DO"
                         -Name "BCERR1" -SignalType "DO"
```

4.9.7 Brake check service routine Continued

Brake check signal description

Introduction

Description of different signal states for brake check in the BrakeCheck routine. The signal names are according to *Description of the I/O setup on page 217*.

Timing sequence for brake check signals

Description of which signals are set at different times during the BrakeCheck execution.

Beginning of brake check

The following signals are set in the beginning of the BrakeCheck execution.

Signal	Set to
всок	0
BCACT	1
BCERR	0
BCWAR	0

End of brake check

The following signals are set in the end of the BrakeCheck execution.

Signal	BrakeCheck test OK Set to	BrakeCheck test WARNING Set to	BrakeCheck test ER- ROR Set to
		56110	Jerio
всок	1	0	0
BCERR	0	0	1
BCWAR	0	1	0
BCACT	0	0	0

Program Pointer moved to Main after interrupted brake check

When the program pointer is moved to Main after an interrupted BrakeCheck execution, the following signals are set.

Signal	Set to
ВСОК	0
BCACT	0

During the first brake check test

Signal	Signal state
всок	0
BCERR	0
BCWAR	0
BCACT	1

Interrupted brake check test, program pointer still in BrakeCheck routine

Signal	Signal state
всок	0

4 Programming and testing

4.9.7 Brake check service routine *Continued*

Signal	Signal state
BCERR	0
BCWAR	0
BCACT	1

Interrupted brake check test, program pointer moved from BrakeCheck routine

Signal	Signal state
BCOK	0
BCERR	0
BCWAR	0
BCACT	0

Configure system parameters

About the system parameters

The configuration of system parameters required for a robot system should be made before running the brake check.



.....

The controller must be restarted after changing the system parameters.

Type Mechanical Unit

All mechanical units for additional axes that shall be supervised must have the parameters *Activate at Start Up* and *Deactivation Forbidden* set to On. (Supervised mechanical units must always be active.)

If an axis should be excluded from brake check, set the parameter Deactivate
Cyclic Brake Check for axis to On.

Type Brake

Type Arm

If brake check is executed on an additional axis, a lowest safe brake torque must be defined. A 5% margin is added during the test for setting the fail limit. The parameter used is *Max Static Arm Torque* defined in on motor side. A warning limit is set with a higher torque value (depending on the brake).

4.9.8 Wrist optimization service routine

4.9.8 Wrist optimization service routine

Overview

The wrist optimization service routine is used to improve the TCP reorientation performance. More information is available in the product manual for the manipulator.

This page is intentionally left blank

5.1.1 Starting programs

5 Running in production

5.1 Basic procedures

5.1.1 Starting programs

Starting programs

Use this procedure to start a program for the first time or to continue running a program that has been stopped.

If your robot system has the option *Multitasking* installed, also see *Using multitasking* programs on page 227.

	Action	Information
1	Check that all necessary preparations are done to the robot and in the robot cell and that no obstacles exist within the robot work area.	
2	Make sure no personnel are inside the ro- bot cell.	
3	Select operating mode on the controller with the mode switch.	A B C D E C C D C C D C C C C C D C C C C C
4	Press the Motors on button on the control- ler to activate the robot.	
5	Is a program loaded? If yes, proceed to the next step. If no, load a program.	How to load programs is described in sec- tion <i>Handling of programs on page 126</i> .
6	If needed, select run mode and speed us- ing the Quickset menu.	See Quickset menu, Run Mode on page 62 and Quickset menu, Speed on page 64.

5 Running in production

5.1.1 Starting programs *Continued*

	Action	Information
7	 In Auto mode: Press the Start button on the Flex- Pendant to start the program. In manual mode: Select start mode. Press and hold the three-position enabling device. Press the Start button on the Flex- Pendant to start the program. 	The button is shown in the section <i>Hard buttons on page 19.</i> How to select start mode is detailed in section <i>Using the hold-to-run function on page 188.</i>
8	Is the Regain Request dialog box dis- played? If yes, return the robot to the path using a suitable method. If no, proceed.	Returning the robot to the path is described in section <i>Returning the robot to the path</i> <i>on page 238</i> .
9	If the Cursor does not coincide with PP dialog box is displayed then tap PP or Cursor to select from where the program should start. Then press the Start button again.	This dialog box is only displayed if the system parameters of type <i>Warning at start</i> are defined. See <i>Technical reference manual - System parameters</i> .

Continue running after the program is changed

You can always continue a program even if it has been changed.

In automatic mode, a warning dialog may appear to avoid restarting the program if the consequences are unknown.

If you	then tap
Are sure the changes you have made are not in conflict with the current robot position and that the program can continue without danger to equipment or personnel	Yes
Are unsure of the consequences your changes might have and want to investigate further	Νο

Restart from the beginning

A program can be restarted from the Production Window or the Program Editor.

PP to Main from the **Production Window** will reset the program pointer to the production entry in all normal tasks, including tasks deactivated in the task selection panel.

PP to Main from the **Program Editor** will reset the program pointer to the production entry in the specified task only, even if the task is deactivated in the task selection panel.

Use this procedure to restart a program from the Production Window.

	Action
1	On the ABB menu, tap Production Window.
2	Tap PP to Main.
3	Start the program by pressing the Start button on the FlexPendant.

Use this procedure to restart a program from the Program Editor.

Action

1

On the ABB menu, tap Program Editor.

Continues on next page

5 Running in production

5.1.1 Starting programs Continued

	Action
2	Tap Debug.
3	Tap PP to Main.
4	Start the program by pressing the Start button on the FlexPendant.

Limitations

Only one program at a time can be executed, unless your system has the *Multitasking* option. If so several programs can be executed simultaneously. See how to select tasks in *Quickset menu, Tasks on page 65*.

If the robot system encounters program code errors while the program is running, it will stop the program and the error is logged in the event log.

5.1.2 Stopping programs

5.1.2 Stopping programs

Stopping programs

If your robot system has the *Multitasking* option installed, see *Using multitasking programs on page 227*.

ACI	ION		

A

- 1 Check that the ongoing operation is in such a state that it can be interrupted.
- 2 Make sure it is safe to stop the program.
- 3 Press the Stop button on the hardware button set of the jogging device.



Do not use the **Stop** button in an emergency. Use the emergency stop button. Stopping a program with the **Stop** button does not mean that the robot will stop moving immediately.

Stopping execution when using hold-to-run or step-by-step execution

When using hold-to-run or step-by-step execution, execution can be stopped according to the following.

Mode	Action	Information
Operation <i>with</i> hold-to-run	Release the Start button	The hold-to-run function is described in section <i>The FlexPendant on page 17</i> .
Step-by-step mode	The robot will stop after ex- ecuting each instruction. Execute the next instruction by pressing the Forward but- ton again.	The STOP and Forward but- ton are described in section <i>The FlexPendant on page 17</i> . If you press the STOP button while executing a move in- struction, the robot will stop without completing the move.

5.1.3 Using multitasking programs

5.1.3 Using multitasking programs

Overview					
	ln a pro rob	system with the option <i>Multitasking</i> installed, you may have one or several grams running in parallel, for instance in a <i>MultiMove</i> cell with more than one ot where each robot has its own task and program (multitasking).			
	For Mu	general information on program handling, see <i>Handling of programs on page 126</i> . Ititasking is described in <i>Application manual - Controller software IRC5</i> .			
Manually set up tas	ks				
	Tas on <i>Col</i>	ks need to be set up in order to run as planned. Normally, all tasks are set up delivery. Setting up tasks is done by defining system parameters of the type <i>ntroller</i> , see <i>Technical reference manual - System parameters</i> .			
	You need detailed information to set up tasks manually. Please read your plant or cell documentation for details.				
How tasks are run					
	Tasks may be defined as Normal, Static, or Semistatic. Static and Semistatic tasks are automatically started as soon as a program is loaded into that task.				
	Normal tasks are started when you press the Start button of the FlexPendant, and stopped when you press the Stop button.				
	To be able to step, start and stop a Static or Semistatic task: set Task Panel Settings to All tasks and activate the task using the Quickset menu. See <i>Application</i> <i>manual - Controller software IRC5</i> , section <i>Multitasking</i> .				
	The concepts of Static, Semistatic, and Normal are described in <i>Technical reference manual - System parameters</i> , type <i>Tasks</i> .				
Load, run, and stop	mul	litasking programs			
	Thi	s section describes how to load, run, and stop multitasking programs.			
	Action				
	1	Make sure there is more than one task set up. This is done using system parameters, see <i>Technical reference manual - System parameters</i> .			
	2	Load programs to respective task using the Program Editor or the Production Window , this is described in section <i>Loading an existing program on page 127</i> .			
	3	If one or more task should be disabled, go to the Quickset menu to do this. See section			

If one or more task should be disabled, go to the Quickset menu to do this. See section *Quickset menu, Tasks on page 65*. Deselecting tasks can only be done in manual mode. When switching to automatic mode, an alert box will appear warning that not all tasks are selected to run.
 Start program execution by pressing the start button. All active tasks are started.

5 Stop program execution by pressing the stop button. All active tasks are stopped.

5 Running in production

5.1.3 Using multitasking programs

Continued

How to load a program to a task

This section describes how to load a program to a task in a multitasking system. It is assumed that the tasks have been configured.

Load a program from the **Production Window**

	Action
1.	On the ABB menu, tap Production Window.
2.	Tap the task into which you want to load a program.
3.	Tap Load Program
	If you want to open a program in another folder, locate and open that folder. See de- scription in <i>FlexPendant Explorer on page 34</i> .
	The file dialog box appears.
4.	Tap the program you want to load followed by OK.

Load a program from the Program Editor

	Action
1.	On the ABB menu, tap Program Editor.
2.	Tap Tasks and Programs.
3.	Tap the task into which you want to load a program.
4.	On the File menu, tap Load Program
	If you want to open a program in another folder, locate and open that folder. See de- scription in <i>FlexPendant Explorer on page 34</i> .
	The file dialog box appears.
5.	Tap the program you want to load followed by OK.
6.	Tap Close to close the Program Editor.

Viewing multitasking programs

In the **Production Window**, there is one tab for each task. To switch between viewing the different tasks, tap on the tabs.

To edit several tasks in parallel, open one **Program Editor** for each task. To edit static and semistatic tasks, see *Application manual - Controller software IRC5*, section *Multitasking*.

5.1.4 Using motion supervision and non motion execution

Motion supervision

The controller software has functionality aiming at reducing collision impact forces on the robot. This helps protecting the robot and external equipment from severe damage if a collision occurs.

Motion supervision during program execution is by default always active, regardless which options are installed in the controller. When a collision is detected, the robot will immediately stop and relieve the residual forces by moving in reversed direction a short distance along its path. The program execution will stop with an error message. The robot remains in the state Motors on so that program execution can be resumed after the collision error message has been acknowledged.

Moreover, there is a software option called *Collision Detection*, which has extra features such as supervision during jogging. To find out if your system has this option installed, tap System Info on the ABB menu. Expand the node System Properties and tap Options under Control Module.

Functions in RobotWare base

Description of functions in RobotWare base:

- · Path Supervision in automatic and manual full speed mode used to prevent mechanical damage due to the robot running into an obstacle during program execution.
- Non motion execution used to run a program without robot motion.

Functions in Collision Detection

A RobotWare system with Collision Detection has additional functionality:

- Path Supervision in manual mode and the possibility to tune supervision in all modes.
- Jog Supervision used to prevent mechanical damage to the robot during jogging.
- RAPID instruction MotionSup used to activate/deactivate collision detection and to tune sensitivity during program execution.



Note

All motion supervision must be set for each task separately.

5 Running in production

5.1.4 Using motion supervision and non motion execution *Continued*

Editing motion supervision settings

This section describes how to modify settings for motion supervision.

	Action	Information
1	On the ABB menu tap Control Panel and then Supervision.	
2	Tap the Task list and select a task.	If you have more than one task, you need to set the desired val- ues for each task separately.
3	 Tap OFF/ON to remove or activate path supervision. Tap the 123 button to open the soft number pad and type a value to adjust the sensitivity of path supervision. Note If the option <i>Collision Detection</i> is not installed, sensitivity setting will have no effect. path supervision affects only the robot in auto and manual full speed mode. 	TipSensitivity can be set between 0 and 300. If it is set lower than 80, however, the robot may stop due to internal drag.NoteNoteYou can modify the sensitivity of Path supervision. For more in- formation, see Setting sensitivity of Motion Supervision on
4	Tap OFF/ON to remove or activate jog supervision. Tap the 123 button to open the soft number pad and type a value to adjust the sensitivity of jog super- vision. Note If the option <i>Collision Detection</i> is not installed, these settings will have no effect.	TipSensitivity can be set between 0 and 300. If it is set lower than 80, however, the robot may stop due to internal drag.NoteNoteYou can modify the sensitivity of Path supervision. For more in- formation, see Setting sensitivity of Motion Supervision on page 231.
5	Under Execution Settings , tap OFF/ON to deactivate or activate non motion execution. This is a separate function, not a part of motion supervision.	See <i>Non motion execution on</i> <i>page 231</i> for information about this function.

Setting sensitivity of Motion Supervision

Use the following procedure to set the sensitivity of Path Supervision and Jog Supervision.

	Action	Information
1	On the ABB menu, tap Control Panel and then Configuration .	
2	Tap Topics and select Motion.	
3	Select the type Motion Supervision and tap.	
4	Select one from the list and tap Edit.	For example: rob1
5	Select <i>Path Collision Detection Level</i> , tap twice and set a value.	The maximum value that can be set is 500.
6	Click OK.	
7	Select <i>Jog Collision Detection Level</i> , tap twice and set a value.	The maximum value that can be set is 500
8	Click OK.	

Non motion execution

Non motion execution enables you to run a RAPID program without robot motion. All other functions work normally; current cycle times, I/O, TCP speed calculation and so on.

Non motion execution can be used for program debugging or cycle time evaluation. It also represents a solution if you need to measure for example glue or paint consumption during a cycle.

When non motion execution is activated it can be executed in:

- manual mode
- manual full speed mode •
- auto mode •

Cycle times will be simulated according to the selected mode.



Note

Non motion execution can only be activated when the system is in Motors Off state.



CAUTION

Non motion execution is reset after a reboot. If you intend to run the program in non motion mode, do not restart without checking the status of Non motion execution. Starting the program incorrectly may cause serious injury, or damage the robot or other equipment.

Related information

For more information on Collision Detection, see Application manual - Controller software IRC5.

5.1.5 Connecting a FlexPendant

5.1.5 Connecting a FlexPendant

Location of FlexPendant connector

The FlexPendant connector is located on the operator's panel on the controller, or on an external operator's panel. The Panel Mounted Controller has a connector on the front.

Connecting a FlexPendant

CAUTION

Always inspect the connector for dirt or damage before connecting it to the controller. Clean or replace any damaged parts.

	Action	Information
1	Locate the FlexPendant socket connector on the controller or operator's panel.	The controller must be in manual mode. If your system has the option Hot plug, then you can also disconnect in auto mode. See section Using the hot plug option on page 234.
		The controller must be in manual mode.
2	Plug in the FlexPendant cable connector.	
3	Screw the connector lock ring firmly by turning it clockwise.	
4	The FlexPendant starts automatically when connected and verifies that it has the cor- rect software installed. If an update is needed, this is shown.	<i>Updating the add-in FlexPendant SxTPU4 Software on page 232</i>

Updating the add-in FlexPendant SxTPU4 Software

Note

The add-in is only available for the FlexPendant with the emergency stop located at the connector. All other FlexPendant versions will automatically update their software via the controller (if needed).

The FlexPendant with the emergency stop located at the connector has an add-in that enables support for different RobotWare versions. This is the **FlexPendant SxTPU4 Software** add-in. The version of the add-in is shown during start-up.

When connecting the FlexPendant, the add-in verifies that it has support for the RobotWare version on the controller. If the RobotWare version is not supported by default, then the add-in requires an update. There are two method to update the FlexPendant add-in. The update is distributed as a software package.

- The update can be installed using a USB drive.
- If the update is available on the controller, then the FlexPendant will update itself when connecting it to the controller.

5.1.5 Connecting a FlexPendant *Continued*

Once the add-in is updated, the FlexPendant can be connected to other IRC5 controllers with the same RobotWare version without requiring additional updates.

Update using a USB drive

Use the following procedure to update the add-in using a USB drive.

- 1 Download the update from RobotStudio, in the tab Add-Ins.
- 2 Save the software package (.rspak) on a USB drive in the folder SxTPU4, located in the root folder.
- 3 With the FlexPendant connected to the controller, reset the FlexPendant with the USB drive connected.
- 4 The update starts automatically and takes approximately 3-4 minutes.

Update from the controller

Use the following procedure to update the add-in from the controller.

- 1 In RobotStudio, use **Installation Manager 6** to create or update a system on your controller. Add the product *FlexPendantSxTPU4Software*.
- 2 Connect the FlexPendant to the controller.
- 3 The update starts automatically and takes approximately 3-4 minutes.

Handling the FlexPendant cables

FlexPendant cables are allowed to be rolled up by hand with a minimum bending radius of 10 times the cable diameter. This also applies to the extension cable. For example, if the cable is 9.5 mm then it is allowed to roll it with a radius of 95 mm. Extension cables are not allowed to be used in chains.

5.1.6 Using the hot plug option

5.1.6 Using the hot plug option

Hot plug option

The hot plug option makes it possible to:

- Disconnect the FlexPendant from a system in automatic mode and thereby run the system without a FlexPendant connected.
- Temporarily connect and operate a FlexPendant without interrupting the application running on the system.



Pressing the hot plug button disables the emergency stop button on the FlexPendant. Only press the hot plug button while connecting or disconnecting the FlexPendant.



A disconnected FlexPendant cannot initiate a protective or emergency stop. It must be stored out of sight so that it cannot be mistaken for being in use.

Connect and disconnect the FlexPendant using the hot plug button

The following procedure describes how to connect or disconnect the FlexPendant on a system in automatic mode using the hot plug button option.



Do not switch to manual mode (or manual full speed mode) while the system is running without the FlexPendant. The FlexPendant must be connected when you switch to automatic mode otherwise you cannot confirm the mode change.

	Action	Information
1	Make sure that the system is in automatic mode.	
2	Press and hold the hot plug button.	A red lamp inside the button indicates when pressed.

5.1.6 Using the hot plug option Continued

	Action	Information
3	Keep pressing the hot plug button and at the same time, switch the jumper plug with the FlexPendant plug.	A
		B
		xx0600002784
		A: Hot plug button
		xx0600002796
		Jumper plug
4	Release the hot plug button.	Make sure that the button is not stuck in the actuated position since this disables the FlexPendant emergency stop button.
5	If the connected FlexPendant does not have support for the RobotWare version running on the controller, then a dialog is shown that the add-in must be updated. See Updating the add-in FlexPendant Sx- TPU4 Software on page 232.	The three-position enabling device and emergency stop button are active even if the add-in dialog is shown.



Note

When the FlexPendant is disconnected, the jumper plug must be connected in its place.



If the hot plug button is released while neither the jumper plug, nor the FlexPendant is connected, the robot movements will be stopped since the emergency stop chains are opened.

5 Running in production

5.1.6 Using the hot plug option *Continued*

Limitations for messages on the FlexPendant

When using the hot plug option, the following limitations apply to messages on the FlexPendant:

Operator messages

Some applications may require input from the operator by using the FlexPendant (e.g. applications using RAPID instructions TPReadNum, UIMsgBox, etc.). If the application encounters such an operator message, program execution will wait. After connecting the FlexPendant you must then stop and start the program execution to be able to see and respond to these messages. They are not displayed automatically by just connecting the FlexPendant.

If possible, avoid using these types of instructions when programming systems that are using the hot plug button option.

Event log messages

When connecting the FlexPendant, event log messages can be viewed also for the period when the FlexPendant was disconnected, since these are stored on the controller.

5.2 Troubleshooting and error recovery

5.2.1 General procedure when troubleshooting

Types of faults	
	Faults occurring in the robot system may be of two categories:
	 Faults detected by the built-in diagnostics system. These faults are described in section <i>Event log messages</i> in <i>Operating manual - Troubleshooting IRC5</i>. Faults NOT detected by the built-in diagnostics system. These faults are described in section <i>Other types of faults</i> in <i>Operating manual - Troubleshooting IRC5</i>.
Faults causing er	ror message on the FlexPendant
	The control system is supplied with diagnostic software to facilitate troubleshooting and to reduce downtime. Any errors detected by the diagnostics are displayed in plain language with a code number on the FlexPendant.
	All system and error messages are logged in a common log in which the last 150 messages are saved. The log can be accessed from the Status bar on the FlexPendant.
	To facilitate troubleshooting, it is important that some basic principles are followed. These are specified in <i>troubleshooting principles</i> in <i>Operating manual - Troubleshooting IRC5</i> .
Faults NOT causi	ng error messages on the FlexPendant
	These faults are not detected by the diagnostic system and are handled in other ways. The way the symptom of the fault is observed greatly influences the type of fault. Instructions are given in section <i>Other types of faults</i> in <i>Operating manual - Troubleshooting IRC5</i> .
	To troubleshoot faults NOT causing error messages on the FlexPendant, follow steps 3 and 4 in the procedure above.
Other possible ac	tions
	Some errors may require running a service routine. See section Service routines

on page 194.

5.2.2 Returning the robot to the path

5.2.2 Returning the robot to the path

About paths and return regions

While a program is running, the robot or additional axis is considered to be *on path*, which means that it follows the desired sequence of positions.

If you stop the program the robot is still on path, unless you change its position. It is then considered to be *off path*. If the robot is stopped by an emergency or safety stop it may also be off path.

If the stopped robot is within the *path return region* you can start the program again, and the robot will return to the path and continue program execution.

Note that there is no way to predict the exact return movement for the robot.



The path return region is set with system parameters, see *Technical reference* manual - System parameters, *Type Path Return Region*.

Returning to path

Turning off the power to the robot motors often results in the robot slipping from its programmed path. This may occur after an uncontrolled emergency or safety stop. The allowed slip distance is configured with system parameters. The distance can be different depending on operating mode.

If the robot is not within the configured allowed distance, you may choose to let the robot return to the programmed path or continue to the next programmed point in the path. Then the program execution continues automatically in programmed speed.

- 1 Make sure there are no obstacles blocking the way and that payload and work objects are properly placed.
- 2 If necessary, put the system in automatic mode and press the Motors on button on the controller to activate the robot motors.
- 3 Press the Start button on the FlexPendant to continue execution from where it stopped. One of these things will happen:
 - The robot or axis slowly returns to the path and the execution continues.
 - The Regain Request dialog will be displayed.
- 4 If the Regain Request dialog is displayed, select the proper action.
 - Tap Yes to return to the path and continue the program.
 - Tap No to return to the next target position and continue the program.
 - Tap Cancel to cancel the program.

5.2.3 Running RAPID program with uncalibrated mechanical unit

5.2.3 Running RAPID program with uncalibrated mechanical unit

When is this useful?

If a servo gun is damaged or uncalibrated, you may want to run a service routine. In order to run the service routine (or any RAPID code), even though an additional axis is uncalibrated, the steps in this description must be followed.

How to get the program started

	Action
1	Set the system parameter <i>Active at Start Up</i> (in type <i>Mechanical Unit</i> , topic <i>Motion</i>) to No.
	Set the system parameter <i>Disconnect at Deactivate</i> (in type <i>Measurement Channel</i> , topic <i>Motion</i>) to Yes.
2	If any of the system parameter values are changed, restart the controller.
3	Deactivate the uncalibrated mechanical unit.
4	Move the program pointer to Main (otherwise the mechanical unit will be automatically activated).
5	Run the service routine or other RAPID code.

5.3.1 Present operating mode

5.3 Operating modes

5.3.1 Present operating mode

Check the position of the controller's mode switch or the status bar of the FlexPendant.

Operational mode changes are also logged in the event log.

The mode switch

Overview

The mode switch should be in the position as illustrated:



xx0300000466

Α	Two position mode switch			
в		Three position mode switch		
С	Automatic mode			
D	Manual reduced speed mode			
Е	Manual full speed mode			
	Actio	n	Information	
1	To sw	vitch from manual to automatic mode	detailed in Switching from manual to auto- matic mode on page 242.	
2	To sv	vitch from automatic to manual mode	detailed in Switching from automatic to manual mode on page 244.	

Viewing present mode on the FlexPendant

On the FlexPendant, you can view the present operating mode in the status bar. An example of the status bar is shown below:



Continues on next page

5.3.1 Present operating mode Continued

В	Operating mode
с	Active system
D	Controller state
E	Program state
F	Mechanical units, active is highlighted

Related information

About the automatic mode on page 183 About the manual mode on page 185 5.3.2 Switching from manual to automatic mode

5.3.2 Switching from manual to automatic mode

When should I put the system in automatic mode?

Put the system in automatic mode when you have a process application or a RAPID program that is ready to be run in production.



When put in automatic mode the robot may move without warning.

Make sure no personnel are in safeguarded space before you change operating mode.

Switching from manual to automatic mode

	Action	Information
1	Set the mode switch in the automatic position. A mode change dialog is displayed.	xx030000467
2	If any debug settings have been changed, a dialog informs about the changes and if these values will be reset. Tap Acknowledge .	If these values are reset or not is defined by system paramet- ers in the type <i>Auto Condition</i> <i>Reset</i> in the topic <i>Controller</i> .
3	Tap OK to close the dialog.	
	If you change the switch back to manual mode the dialog will be closed automatically.	
4	Did the system change mode without errors? If yes, then resume or start the process application or RAPID program. If no, stop and troubleshoot the problem.	How to start programs is de- scribed in <i>Starting programs on</i> <i>page 223</i> .



If your specific system uses a distributed operator's panel, controls and indicators may not be placed exactly as described in this manual. Consult your plant or cell documentation for details.

Controls and indicators do however look and function the same way.

When can I start using the robot system?

As long as the mode change dialog is displayed programs cannot be started and the robot's motors cannot be activated either manually or remotely.

5.3.2 Switching from manual to automatic mode *Continued*

Exceptions

In automatic mode it is possible to start a RAPID program and turn motors on remotely. This means that the system will never enter a safe standby state and the robot may move at any time.

Please consult your plant or cell documentation for details on how your system is configured.

Related information

A number of conditions can be set or reset when switching to automatic mode, see *Technical reference manual - System parameters*, sections *Auto Condition Reset* and *Run Mode Settings*.

5.3.3 Switching from automatic to manual mode

5.3.3 Switching from automatic to manual mode

Switching from automatic to manual mode

	Action	Information
1	Set the mode switch in the manual position.	xx030000468
2	Did the system change mode without er- rors? If yes, then this procedure is completed. If no, try to locate the error.	Error handling is detailed in <i>Operating manual - Troubleshooting IRC5</i> .



If your specific system uses a distributed operator's panel, controls and indicators may not be placed exactly as described in this manual. Please consult your plant or cell documentation for details.

Controls and indicators do however look and function the same way.

5.3.4 Switching to manual full speed mode

5.3.4 Switching to manual full speed mode

When should I use the manual full speed mode?

Use full speed manual mode when the program is to be tested at full speed.

The manual full speed mode allows you to run the program at full speed while still having access to all the available debugging functions of the program editor.



Testing at full speed is dangerous.

Make sure no personnel are in safeguarded space when starting the program.

Switching to manual full speed mode

	Action	Information
1	Set the mode switch to the manual full speed position.	
2	Did the system change mode without er- rors? If yes, then this procedure is completed. If no, try to locate the error.	Error handling is detailed in <i>Operating manual - Troubleshooting IRC5</i> .

Note

When you switch to manual full speed mode, all the functionality except for Start, Stop, and Step are disabled.

FlexPendant alert

When changing mode, a dialog is displayed on the FlexPendant to alert you about the change of mode. Tap **OK** to close the dialog.

If you switch back to the previous mode the dialog is closed automatically and there is no change in the mode.

5.4.1 Modifying and tuning positions

5.4 Modifying positions

5.4.1 Modifying and tuning positions

Overview

Positions are instances of the data type robtarget or jointtarget. See *Technical reference manual* - *RAPID Instructions, Functions and Data types*.

The positions can be tuned using the function HotEdit, where you enter offset values using a soft keyboard. The offset value is used together with the original position value. See *Tuning positions with HotEdit on page 251*. The HotEdit menu is described in section *HotEdit menu on page 32*.

The positions can also be modified using the **Modify positions** function in the **Program Editor** or the **Production Window** where you step and jog the robot to the new position. A modified position value overwrites the original value. See *Modifying positions in the Program Editor or Production Window on page 247.*



Changing programmed positions may significantly alter the robot's movement pattern.

Always make sure any changes are safe for both equipment and personnel.

Positions in arrays

If a position is declared as an array, then the procedure for modifying or tuning may differ slightly depending on how the array is indexed in the move instruction.

Limitations

Note that jointtargets can only be modified using the Modify positions method in the **Program Editor** and the **Production Window**, i.e. not with HotEdit.



Your system can have restrictions on how positions can be modified. Restrictions can apply to distance using system parameters (topic *Controller*, type *ModPos Settings*) and which positions can be modified using UAS.

5.4.2 Modifying positions in the Program Editor or Production Window

Overview

When modifying positions by jogging the robot to the new position you can either single-step through the program to the position(s) you want to modify, or jog directly to the new position and change the corresponding position argument of the instruction.

The recommendation is to step through the program to the position, but if you know your robot program well and the new position is known, it is faster to use the jogging method.



Do not use this method to change orientation values.

Prerequisites

To modify positions using the Program Editor or Production Window.

- · the system must be in manual mode
- the target position must have a initial value. For example: CONST robtarget
 p10:=[[515.00,0.00,712.00],[0.707107,0,0.707107,0],[0,0,0,0],
 [9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]]; CONST jointtarget
 jpos10:=[[-0,-0,0,-0,-0,-0],
 [9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];



To modify positions in the Production Window, you must have started the program so that the motion pointer is set.

Applying modified positions

The modified position values will normally be used when you restart the program. If the robot cannot use the values directly at start, a warning is displayed. Then the modified position will be used the next time that the position is used in the program.

Modifying positions

This procedure describes how to modify positions, either by single-stepping to the positions or jogging. You can use the **Program Editor** or the **Production Window**, the functionality is the same.

	Action	Information
1	On the ABB menu, tap Program Editor.	
2	Stop the program, if running.	

5.4.2 Modifying positions in the **Program Editor** or **Production Window** *Continued*

	Action	Information
3	Do you want to single-step to the position or jog? If <i>single-stepping</i> , step through the program to the position you want to change. Make sure the correct argument is selected. If <i>jogging</i> , use the Jogging view to make sure that the same work object and tool that are used in the instruction are selected.	When single-stepping, if the instruction or procedure call has more than one position argument, continue to step to reach each argument.
4	Jog to the new position.	
5	When using the jogging method, tap to select the position argument you want to change.	
6	In the Program Editor, tap Modify Position . In the Production Window, tap Debug and then Modify Position . A confirmation dialog appears.	When modifying a position in an array that is indexed with a variable you will have to select which element in the ar- ray to modify before the modification is executed.
7	Tap Modify to use the new position, Cancel to keep the original.	If you select the check box Don't show this dialog again in the confirmation dialog, then you will not get any more confirmation dialogs when modifying positions. Note This is only valid for the current Pro- gram Editor.
8	Repeat step 3 through 7 for each position ar- gument you want to change.	

Limitations

The **Modify position** button in the **Program Editor** is disabled until you select a position argument (that is possible to modify).

The **Modify position** button in the **Production Window** is disabled until the motion pointer is set and a position is selected. To set the motion pointer, the program must be started and then stopped.

The maximum movement or change in orientation, may be restricted by the system parameters (topic *Controller*, type *ModPos Settings*) in the system design. Please read your cell or plant documentation for details.

If the system parameters are setup to use absolute limits for position changes, then the original positions can only be restored or changed using the baseline menu in HotEdit. The baseline concept is described in section *Tuning positions with HotEdit on page 251*.

If a named position is modified, all other instructions using that position will be affected.

In the **Production Window**, circle points cannot be modified in synchronized mode. See *Application manual - MultiMove*.

Differences between Program Editor and Production Window

The procedure for modifying positions is the same in the **Program Editor** and the **Production Window**. However, there are differences in how positions are selected.

Continues on next page

```
5.4.2 Modifying positions in the Program Editor or Production Window
Continued
```

Also, if your system uses MultiMove, then the result from the **Program Editor** and the **Production Window** will differ. See *Application manual - MultiMove*.

Program Editor selections

To select a position for modification in the **Program Editor**, tap the desired position.

Production Window selections

To select a position for modification in the **Production Window** you must step the program to the desired position.



If you have executed the program from another window and then switch back to the **Production Window**, the selected position will be changed to the position where the motion pointer now is. Make sure the correct position is selected before making the modification!

Related information

For an overview on how to modify positions, see *Modifying and tuning positions* on page 246.

HotEdit and baseline are described in Tuning positions with HotEdit on page 251.

The HotEdit menu is also described in HotEdit menu on page 32.

Modifying positions in the **Program Data** window is described in *Editing data instances on page 146*.

Technical reference manual - RAPID Instructions, Functions and Data types.

Technical reference manual - System parameters.

Application manual - MultiMove

Examples planned path

The following examples show how the planned path will be effected when modifying positions.

Linear movement

In example A the robot is stopped on path before reaching the position P10. The robot is jogged off path to the new position (P10x) and the position P10 is modified. In example B the robot is stopped on path in position P10. The robot is jogged off path to the new position (P10x) and the position P10 is modified.



5 Running in production

5.4.2 Modifying positions in the **Program Editor** or **Production Window** *Continued*

In both examples, when restarting the program the robot continues from the new P10 (that is now the same as P10x) directly to P20 without returning to the previous planned path (via the old P10).

Circular movement

In this example the robot is stopped on path in position P20 (circle point) and then jogged to the new position P20x. The position P20 is modified.



xx0800000176

In single robot systems or *MultiMove* systems in unsynchronized mode: When restarting the program the robot continues directly from the new P20 (that is now the same as P20x) to P30 without returning to the previous planned path (via the old P20). The new planned path from P20 (P20x) to P30 is calculated using these two positions and position P10.

In *MultiMove* synchronized mode: When restarting the program the robot returns to the old P20 and uses the previously planned path to P30. In the next cycle only the new P20 (P20x) is used.

5.4.3 Tuning positions with HotEdit

5.4.3 Tuning positions with HotEdit

Overview

HotEdit is used to tune programmed positions. This can be done in all operating modes and even while the program is running. Both coordinates and orientation can be tuned.

HotEdit can only be used for named positions of the type robtarget (see limitations below).

The functions available in HotEdit may be restricted by the user authorization system, UAS.

The HotEdit menu is described in the section HotEdit menu on page 32.

Applying tuned positions

Tuning values are used directly by an executing program when you tap **Apply**. If tuning is done close to the program or motion pointer it may be hard to predict exactly when it will take effect. It is therefore important that you know where in the program the robot is if applying offset values while the program is running.

However, the new values are not stored in the baseline until you use a **Commit** command.

How to tune positions

This is how you tune programmed positions using HotEdit:

	Action
1	In the Programmed targets window, select the positions to be tuned and add them to Selected targets by tapping the arrow.
2	Tap Tune Targets and select tuning mode (linear, reorient, or external axes) and then coordinate system (tool or work object).
3	Tap + and - to specify the exact tuning of the position(s) in x, y and z direction. Select Increment to define the step size of these buttons.
4	To activate the new values, tap APPLY . The offset will be used directly if the program is running.
5	If you are satisfied with the result and want the tuned positions to become part of the baseline, tap Baseline and then Commit Selection .
6	If, however, the selected targets need further tuning, you can tap Baseline and then Restore Selection and start all over again, or you can simply continue tuning until you are satisfied.

Working with selections

A selection of positions to be tuned later can be saved on the controller mass memory unit. If your system uses UAS, this may be the only way to select positions for tuning.

The commands for working with selections are located in the File menu:

Save Selection As	Make sure that the window Selected targets shows nothing but the
	positions to be saved. Tap File and Save Selection As. Enter the
	name and optionally a description of the file, then tap OK .

5 Running in production

5.4.3 Tuning positions with HotEdit *Continued*

Open Selection	Tap File and Open Selection. Then tap the selection you want to use and tap OK.
Clear Selection	Clear the Selected targets area by tapping File and Clear Selection.

Baseline concept

A baseline can be defined as a reference against which future changes are measured. The baseline concept makes it possible to undo tuning and revert to the position values stored in the latest baseline. To do this you use a **Restore** command.

When a **Commit** command is performed the baseline is updated with new offset values, and the old values no longer exist in program memory.

Use the baseline menu to apply or reject tuning.

- **Restore Selection** will discard all tuning of the currently selected positions and revert them to the values of the latest baseline, meaning that their offset values will be 0,0.
- **Restore Entire Program** will discard ALL tuning to programmed positions since the latest **Commit** command. This may include several HotEdit sessions for the same task. If the system uses *Absolute Limit ModPos* any **Modify Position** command from the Program Editor will also be undone.
- **Commit Selection** will apply the offset of the currently selected positions to the baseline.
- **Commit Entire Program** will apply ALL tuning to programmed positions. This may include several HotEdit sessions for the same task. If the system uses *Absolute Limit ModPos* it also includes **Modify Position** performed in the Program Editor.

Baseline target criteria

Targets that fulfil all of the following criteria are part of the baseline:

- The data type must be robtarget or jointtarget
- It must not be declared locally in a routine
- It must **not** be declared as part of an array of targets
5.4.3 Tuning positions with HotEdit *Continued*

Illustration of baseline concept

The baseline concept is illustrated below, where a point is moved, restored and committed. Starting out from the original baseline (A), let us assume that you move the point (B) twice. If you regret the changes you perform a restore command (C). But if you instead continue moving the point and perform a commit command (B +D), you will have created a new baseline (E) and there is no way to revert to the original baseline. If you move the point one more time and then restore, the point is moved back to the latest baseline (E).



xx0600002620

A	Original baseline
В	Move selected point
С	Restore
D	Commit
E	New baseline

Restore Selection or Restore Entire Program

The following example shows the difference between **Restore Selection** and **Restore Entire Program** to original. The same idea applies for **Commit Selection** and **Commit Entire Program**.

	Action
1.	The robtargets p10 and p30 are added to Selected Targets and tuned once.
2.	p10 is removed from Selected Targets
3.	p30 is tuned again.
4.	• Restore Selection sets the currently selected position, p30, to its value in the latest baseline. p10 is not affected, thus still tuned.
	 Restore Entire Program sets all tuned positions, that is both p10 and p30 to their baseline values.

HotEdit for external axes

External axes can be tuned with HotEdit if they are activated in at least one of the selected robtargets. Only axes with active values are tuned.

Limitations

HotEdit tuning is only possible for named (e.g. p10, p20) robtargets. (* robtargets are not visible in the treeview.)

If a robtarget is declared as an array, it must be indexed with a number to be modified in HotEdit.

5.4.3 Tuning positions with HotEdit Continued

> It is only possible to perform HotEdit tuning on targets that are part of the baseline. Targets that are NOT part of the baseline will not be shown in the HotEdit treeview, as they cannot be selected for tuning. This means that a target declared locally in a routine, for example, will not be displayed.

HotEdit tuning is possible for robtargets only. (Jointtargets can only be tuned by using Modify Position in the Program Editor.) If the system uses Absolute limit ModPos these jointtargets are however part of the baseline and will be affected when Restore Entire Program and Commit Entire Program are used.



Note

For more information about Absolute Limit ModPos, see the Technical reference manual - System parameters, section Topic Controller - Type ModPos Settings.

Using UAS in HotEdit

The user authorization system can be used to restrict the Hot Edit functionality and only allow a user to edit pre-selected positions. These are loaded by tapping File and then Open Selection. The selected positions can then be tuned in the usual way.

Related information

Technical reference manual - System parameters.

5.4.4 Working with displacements and offsets

5.4.4 Working with displacements and offsets

About displacements

Sometimes, the same path is to be performed at several places on the same object, or on several work pieces located next to each other. To avoid having to reprogram all positions each time a displacement coordinate system can be defined.

This coordinate system can also be used in conjunction with searches, to compensate for differences in the positions of the individual parts.

The displacement coordinate system is defined based on the object coordinate system.

The displacement coordinate system is described in section *Coordinate systems for jogging on page 99*.

Select displacement method

Depending on how, when, and how often you want to use displacements, the best method may vary.

Moving a work object

Moving a work object is suitable when you do not need to move or displace the work object very often.

This is detailed in section Defining the work object coordinate system on page 170.

Displace a work object

A work object consists of a user frame and a object frame. You can move one or both of these frames. If you move both frames, then the whole work object is moved. It can be useful to displace the object frame from the user frame for instance when using one fixture for several work objects. Then you can keep the user frame and displace the object frame for the work objects.

See procedure *How to define object frame* in section *Defining the work object coordinate system on page 170*.

Displace and rotate a work object

You may want to displace and rotate the object frame from the user frame if the displacement is not in just x, y, and z.

To displace in x, y, and z, you can use the same method as above. To rotate the work object, follow the procedure in section *Editing the work object data on page 174*.

About offsets

Sometimes it is easier to define a position as an offset from a given position. If, for example, you know the exact dimensions of a work object, it will only be necessary to jog to one position.

The offset is programmed with the displacement distance in x, y, and z direction, in relation to the work object. For instance:

MoveL Offs(p10, 100, 50, 0), v50...

Define the offset for the position with the following expressions:

1 Original position / starting point

5 Running in production

5.4.4 Working with displacements and offsets *Continued*

- 2 Displacement in x direction
- 3 Displacement in y direction
- 4 Displacement in z direction

Examples

This example shows the move instructions with offsets to move the robot in a square (clockwise), starting at p10, with a 100 mm displacement in x and y.

```
MoveL pl0, v50...
MoveL Offs(pl0, 100, 0, 0), v50...
MoveL Offs(pl0, 100, 100, 0), v50...
MoveL Offs(pl0, 0, 100, 0), v50...
MoveL pl0, v50...
```

How to create position offsets

This procedure details how to change a position to become an offset position.

	Action	Information
1	In the Program Editor, tap to select the position argument to edit.	
2	Tap Edit and then Change Selected.	
3	Tap Functions and then Offs.	
4	Tap to select each expression, < EXP >, and then tap any of the desired available data or functions.	You can use the filter to narrow down the available data. You can also change data type of the available data.
	You can also tap Edit to access more functions. Tap All to open the soft key- board and edit all expressions at the same time, or tap Only Selected to edit one at a time with the soft keyboard.	
5	Tap OK to save changes.	

Related information

There are a number of functions in RAPID that may be useful. See *Technical reference manual - RAPID Instructions, Functions and Data types, and Technical reference manual - RAPID Overview.*

5.4.5 Moving the robot to a programmed position

5.4.5 Moving the robot to a programmed position

Positions

A robot program usually contain programmed positions. The robot can move automatically to a programmed position using a function in the Jogging menu. The robot will move at 250 mm/s.



DANGER

When moving the robot automatically, the robot arm may move without warning. Make sure no personnel are in safeguarded space and that no objects are in the way between the current position and the programmed position.

Moving the robot to a programmed position

This procedure describes how to move a robot automatically to a programmed position.

	Action	Information
1	On the ABB menu, tap Jogging.	
2	Make sure the correct mechanical unit is selected and then tap Go To .	
3	Tap to select a programmed position.	If you have many pro- grammed positions you can use a filter to narrow down the visible positions. See section <i>Filtering data on</i> <i>page 70</i> .
4	Press and hold the three-position enabling device and then tap and hold the Go To button. The robot now moves directly from the current position to the programmed position. Make sure no objects are in the way.	

This page is intentionally left blank

6.1 Viewing signal lists

6 Handling inputs and outputs, I/O

6.1 Viewing signal lists

Overview

I/O signal properties is used to view the input and output signals and their values. Signals are configured with system parameters .

How to view signal lists

	Action	
1	On the ABB menu tap Inputs and Outputs. The list of Most Common I/O signals is displayed. Manual Motors On MySystem (RSTEST4) Stopped (2 of 2) (Inputs and Outputs Most Common Active filter: Sele Select an I/O Signal from the list. Def Name Value Ty	Alias I/O Industrial Networks I/O Devices All Signals Digital Inputs
		Digital Outputs Analog Inputs Analog Outputs Groups Input Groups Output Most Common Safety Signals Simulated Signals
		View
	en0400000770	¥ 3
2	Tap View to change the selection of signals in the	list.
~(Тір	

This section details how to view a list of signals.

Tap the Select Layout menu if you want to view signal labels in the list.

Related information

Simulating and changing signal values on page 260. Filtering data on page 70. Configuring Most Common I/O on page 91.

6 Handling inputs and outputs, I/O

6.2 Simulating and changing signal values

6.2 Simulating and changing signal values

Simulating and changing signal values

A signal can be changed into a simulated signal. The value of the signal can also be changed.

Use the following procedure to simulate a signal or to change the value of a signal:

	Action			
1	On the ABB menu, tap Inputs and Ou A list of most common signals is displa <i>I/O on page 91</i> .	t puts . ayed. See sect	ion <i>Configur</i> i	ing Most Common
2	Tap View and select a signal filter. The signal list is updated.			
	Manual VC_Merker (IN-L-BTGI51	Guard St 4975) Stopped	op (2 of 2) (Speed 10	10%)
	🛱 Inputs and Outputs			
	All Signals	Select Layou	t	
	Select an I/O Signal from the list.		Default	
	Name 🛆	Value	Туре	Device
	di3	0	DI	\bigtriangleup
	di4	0	DI	
	di5	0	DI	
	diSignal1	0 (Sim)	DI	EN_Internal_A
	diSignal2	0	DI	EN_Internal_A
	diSignal3	0	DI	
	diSignal4	0	DI	
	diSignal5	0	DI	
	diSignal6	0	DI	
	diSignal7	0	DI	\checkmark
	0 1 S	Simulate 🗖		View 🔷
	≓ 1/0			
	xx1700002252			
	Note			
	Tap All Signals to view all the signals.	-		
3	Tap on a signal.			
4	Tap on Simulate to change the signal Tap on Remove Simulation to remove	into a simulate the simulation	ed signal. n from the sig	ınal.
	Note			
	When you remove the simulation from value it had before it was simulated. A ingly.	a signal, the s lso the cross c	ignal value re onnections a	eturns to the same re updated accord-

6.2 Simulating and changing signal values *Continued*

	Action
5	For a digital signal, tap 0 or 1 to change the value of a signal. For analog signals and groups, tap on 123 to change the signal value. The soft nu- meric keyboard is displayed. Enter the new value and tap OK.
	Note
	If there is a cross connection created between two signals, changing the value of one signal automatically changes the value of the corresponding cross connected signal.

For more information about changing the signal properties see the section Control Panel, *Configuring Most Common I/O on page 91*.

Managing I/O devices

You can manage I/O devices from the I/O Devices page.

Inputs and Outputs	lanual :C_R₩6.07.00xx (10.140.¢	Guard Stop 5) Stopped (Speed 1004	%)
I/O Devices			
Select an I/O Device fi	rom the list.		
Name 🛆	Network	Address	State
DN_Internal_Device	DeviceNet	2	Running
DRV_1	Local	-	Running
EN_Internal_Anyb	EtherNetIP_Anybus	0.0.0	Running
EN_Internal_Device	EtherNetIP	192.168.125.1	Running
Level TO	Ed. N. JER	100 100 100 101	
Local_IU	EtherNetIP	192.168.125.101	Unknown
Local_IO1	EtherNetIP EtherNetIP	192.168.125.101 192.168. Bit Value	Unknown hnown
Local_IO1 PANEL	EtherNetIP EtherNetIP Local	192.168.125.101 192.168. Bit Value	Unknown Thown Sining
Local_IO1 PANEL	EtherNetIP EtherNetIP Local	192.168.125.101 192.168.Bit Value Identify	Unknown Inown I ^S Ining
Local_IO1 PANEL	EtherNetIP EtherNetIP Local	192.168.125.101 192.168.Bit Value Identify Configur	Unknown PS Ining e
Local_IO1 PANEL	EtherNetIP EtherNetIP Local	192.168.125.101 192.168.Bit Value Identify Configur	Unknown ¹⁵ ining e
Local_IO1 PANEL Activate	EtherNetIP EtherNetIP Local Signals	192.168.125.101 192.168.Bit Value Identify Configur Firmware Action	Unknown Inown Inown Ining e s View

xx1700002253

Following are the actions that you can perform from I/O Devices page:

Action	Output
Activate/Deactivate	Activates or deactivates a selected I/O device.
Signals	Displays the signals associated with the selected I/O device.
Actions - Bit Values	Displays the input and output bit values of the selected device.
Actions - Identify	Displays the MAC ID of the selected device.
	Note Note
	Actions - Identify option is available only for the network type EtherNet IP.

Continues on next page

6 Handling inputs and outputs, I/O

6.2 Simulating and changing signal values *Continued*

Action	Output
Actions - Configure	Allows you to configure the selected local I/O device.
	1 Note
	Actions - Configure option is available only for the network type EtherNet IP.
Actions - Firmware	Allows you to upgrade the firmware.
	Note
	Actions - Firmware option is available only for the network type EtherNet IP.

Upgrading firmware

You can upgrade the firmware of I/O device of type EtherNet IP. Use the following procedure to upgrade the firmware.

	Action	Description
1	Tap Main > Inputs and Outputs.	The Input & Output window is displayed.
2	Tap View > I/O Devices.	The I/O Devices window is displayed. It displays the available I/O devices.
3	Select the EtherNet IP I/O device.	Note The Firmware upgrade option is available only for the net- work type EtherNet IP.
4	Tap Actions > Firmware.	The Device Firmware window is displayed with the details of the firmware. Image: C_RW6.07.00xx. (10.140.6.) Stopped (Speed 100%) Device Firmware Local_IO : Finished : Finished [0] Base ProgramName: A_HYPIOM_B_3_8 SerialNumber: 6923766 HardwareRevision: C.1 LatestProgramNameAvailable: A_HYPIOM_B_3_8 Firmware path: /hd0a/RC_RW6.07.00xxECI/PRODUCTS/RobotWare_6.07.0087/ Refresh Upgrade Close Production Image: Non-System Info xx1800000120
5	Tap Upgrade .	The firmware for the selected EtherNet IP I/O device is upgraded.

6.3 Viewing signal group

6.3 Viewing signal group

Viewing signal group

This section details how to view signal groups.

	Action
1	On the ABB menu, tap I/O . A list of most common signals is displayed. See section <i>Configuring Most Common</i> <i>I/O on page 91</i> .
2	In the View menu, tap Groups.
3	Tap on the signal group's name in the list and then tap Properties . Or tap twice on the signal group's name. The signal group's properties is displayed.

6.4.1 Safety I/O signals

6.4 Safety signals

6.4.1 Safety I/O signals

General

In the IRC5 controller's basic and standard form, certain I/O signals are dedicated to specific safety functions. These are listed below with a brief description of each. All signals can be viewed in the I/O menu on the FlexPendant.

Safety I/O signals

The list below contains the safety I/O signals as used by the standard system.

	,	3 ,	,
Signal name	Description	Bit value condition	From - To
ES1	Emergency stop, chain 1	1 = Chain closed	From panel board to main computer
ES2	Emergency stop, chain 2	1 = Chain closed	From panel board to main computer
SOFTESI	Soft Emergency stop	1 = Soft stop enabled	From panel board to main computer
EN1	Enabling device1&2, chain 1	1 = Enabled	From panel board to main computer
EN2	Enabling device1&2, chain 2	1 = Enabled	From panel board to main computer
AUTO1	Op mode selector, chain 1	1 = Auto selected	From panel board to main computer
AUTO2	Op mode selector, chain 2	1 = Auto selected	From panel board to main computer
MAN1	Op mode selector, chain 1	1 = MAN selected	From panel board to main computer
MANFS1	Op mode selector, chain 1	1 = Man. full speed selected	From panel board to main computer
MAN2	Op mode selector, chain 2	1 = MAN selected	From panel board to main computer
MANFS2	Op mode selector, chain 2	1 = Man. full speed selected	From panel board to main computer
USERDOOVLD	Over load, user DO	1 = Error, 0 = OK	From panel board to main computer
MONPB	Motors-on pushbutton	1 = Pushbutton pressed	From panel board to main computer
AS1	Auto stop, chain 1	1 = Chain closed	From panel board to main computer
AS2	Auto stop, chain 2	1 = Chain closed	From panel board to main computer
SOFTASI	Soft Auto stop	1 = Soft stop enabled	From panel board to main computer
GS1	General stop, chain 1	1 = Chain closed	From panel board to main computer

6.4.1 Safety I/O signals Continued

Signal name	Description	Bit value condition	From - To
GS2	General stop, chain 2	1 = Chain closed	From panel board to main computer
SOFTGSI	Soft General stop	1 = Soft stop enabled	From panel board to main computer
SS1	Superior stop, chain1	1 = Chain closed	From panel board to main computer
SS2	Superior stop, chain2	1 = Chain closed	From panel board to main computer
SOFTSSI	Soft Superior stop	1 = Soft stop enabled	From panel board to main computer
CH1	All switches in run chain 1 closed	1 = Chain closed	From panel board to main computer
CH2	All switches in run chain 2 closed	1 = Chain closed	From panel board to main computer
ENABLE1	Enable from MC (read back)	1 = Enable, 0 = break chain 1	From panel board to main computer
ENABLE2_1	Enable from AXC1	1 = Enable, 0 = break chain 2	From panel board to main computer
ENABLE2_2	Enable from AXC2	1 = Enable, 0 = break chain 2	From panel board to main computer
ENABLE2_3	Enable from AXC3	1 = Enable, 0 = break chain 2	From panel board to main computer
ENABLE2_4	Enable from AXC4	1 = Enable, 0 = break chain 2	From panel board to main computer
PANEL24OVLD	Overload, panel 24V	1 = Error, 0 = OK	From panel board to main computer
DRVOVLD	Overload, drive mod- ules	1 = Error, 0 = OK	From panel board to main computer
DRV1LIM1	Read back of chain 1 after limit switches	1 = Chain 1 closed	From axis computer to main computer
DRV1LIM2	Read back of chain 2 after limit switches	1 = Chain 2 closed	From axis computer to main computer
DRV1K1	Read back of contactor K1, chain 1	1 = K1 closed	From axis computer to main computer
DRV1K2	Read back of contactor K2, chain 2	1 = K2 closed	From axis computer to main computer
DRV1EXTCONT	External contactors closed	1 = Contactors closed	From axis computer to main computer
DRV1TEST1	A dip in run chain 1 has been detected	Toggled	From axis computer to main computer
DRV1TEST2	A dip in run chain 2 has been detected	Toggled	From axis computer to main computer
SOFTESO	Soft Emergency stop	1 = Set soft E-stop	From main computer to panel board
SOFTASO	Soft Auto stop	1 = Set soft Auto stop	From main computer to panel board
SOFTGSO	Soft General stop	1 = Set soft General stop	From main computer to panel board

6 Handling inputs and outputs, I/O

6.4.1 Safety I/O signals *Continued*

Signal name	Description	Bit value condition	From - To
SOFTSSO	Soft Superior stop	1 = Set soft Sup. E- stop	From main computer to panel board
MOTLMP	Motors-on lamp	1 = Lamp on	From main computer to panel board
TESTEN1	Test of Enable1	1 = Start test	From main computer to panel board
DRV1CHAIN1	Signal to interlocking circuit	1 = Close chain 1	From main computer to axis computer 1
DRV1CHAIN2	Signal to interlocking circuit	1 = Close chain 2	From main computer to axis computer 1
DRV1BRAKE	Signal to brake-release coil	1 = Release brake	From main computer to axis computer 1

7.1 Accessing the event log

7 Handling the event log

7.1 Accessing the event log

Event log

Open the event log to:

- view all present entries.
- study specific entries in detail.
- handle the log entries, such as saving or deleting.

The log can be printed by using RobotStudio.

Open and close the event log

This section details how to open the event log.

	Action
1	Tap the status bar. The status window is displayed.
2	Tap Event Log . The event log list is displayed.
3	If the log contents do not fit into a single screen, it can be scrolled.
4	Tap a log entry to view the event message.
5	Tap the status bar again to close the log.

Related information

Operating manual - RobotStudio.

7 Handling the event log

7.2 Deleting log entries

7.2 Deleting log entries

Why should I delete log entries?

Logs can be deleted to increase available disk space. Deleting log entries is often a good way to trace faults since you remove old and insignificant log entries not related to the problem you are trying to solve.

Delete all log entries

	Action
1	Tap the status bar, then the Event Log tab to open the event log.
2	On the View menu, tap Common.
3	Tap Delete and then Delete all logs. A confirmation dialog is displayed.
4	Tap Yes to delete, or No to keep the log intact.

Delete log entries of a specific category

	Action
1	Tap the status bar, then the Event Log tab to open the event log.
2	On the View menu, tap the category of choice.
3	Tap Delete and then Delete log . A confirmation dialog is displayed.
4	Tap Yes to delete, or No to keep the log intact.

7.3 Saving log entries

7.3 Saving log entries

Why should I save log entries?

You should save log entries when:

- you need to clear the log but want to keep the current entries to be viewed later.
- you want to send log entries to support to solve a problem.
- you want to keep log entries for future reference.



The log can keep up to 20 entries per category and up to 1000 entries in the all events list. When the buffer is full the oldest entries will be overwritten and lost. There is no way to retrieve these lost log entries.

Save all log entries

This section details how to save all log entries.

	Action
1	Tap the status bar to open the event log.
2	Tap Save all logs as . The file dialog is displayed.
3	If you want to save the log in a different folder, locate and open the folder.
4	In the File name box, type a name for the file.
5	Tap Save.

This page is intentionally left blank

8 Backup and restore

8.1 Backup the system

When do I need this?

We recommend performing a backup:

- Before installing new RobotWare.
- *Before* making any major changes to instructions and/or parameters to make it possible to return to the previous setting.
- *After* making any major changes to instructions and/or parameters and testing the new settings to retain the new successful setting.

Backup the system

This section describes how to backup the system.

ap Backup Curr	ent System.				
he Backup Curr	ent System window	is displayed.			
a default path h <i>age 77</i> , this is di	as been defined as splayed.	detailed in the section S	etting defau	ılt p	
	Manual IN-L-GISBT14140	Guard Stop Stopped (Speed 100%)	X	X	
Backup Current Sy	stem			·	
Backup folder:				_	
RW6.06_Bac	:kup_20170713		ABC		
Backup path:					
C:/Users/inprkm2/Documents/RobotStudio/Systems/BA					
Backup will be created at:					
C:/Users/inprkm2/Documents/RobotStudio/Systems/BACKUP/RW6.06_B					
		Backup	Cance	el	
Advanced	•	Duckup			
Advanced	up ore	Duckup		B_1	
Advanced	up ore	buddep		B_1	
Advanced	up ore			B_1	
Advanced Production Window Back CO300000441 Note By default, the user lat	aname for the Back	cup folder is created whi	ch can be re	ena	

8 Backup and restore

8.1 Backup the system *Continued*

	Action		
3	Tap Advanced The Controller Settings window is displayed.		
	Note		
	Use this step to create the backup as a zip file. If this is not required navig to the last step.	gate directly	'
	Manual Guard Stop IN-L-GISBT14140 Stopped (Speed 100%)	X	
	🖹 Controller Settings		
	Please select the controller settings and press OK to continue.		
	ОК С	Cancel	
	Production Window xx1700001302		
4	Select the Backup to archive file checkbox.		_
5	Тар ОК.		
6	Is the displayed backup path the correct one? If YES: Tap Backup to perform the backup to the selected directory. A banamed according to the current date is created. If NO: Tap to the right of the backup path and select directory. Then tap backup folder named according to the current date is created.	ackup file p Backup . A	·
	·		-

Disable or queue backup

Backing up the system during production can interfere with the RAPID execution. To avoid that a backup is taken during critical process steps or sensitive robot movements, a system input (*Disable Backup*, type *System Input*) can be set during these critical steps. When the critical steps are done, the input should be reset to allow backups again.

If needed, the backup can be queued while *Disable Backup* is set, using the system parameter *General RAPID*, with action value *QueueBackup* set to *TRUE*. Then the backup will be queued until the signal is reset.

Disable Backup and *QueueBackup* are described in *Technical reference manual - System parameters*.

8.2 Important when performing backups

8.2 Important when performing backups

BACKUP directory

A local default backup directory, BACKUP, is automatically created by the system. We recommend using this directory for saving backups.

Such backups are not copied to the directory HOME in following backups.

Never change the name of the BACKUP directory.

Never change the name of the actual backup to BACKUP, since this will cause interference with this directory.

A default path can be created to any location on the network where the backup should be stored, see *Setting default paths on page* 77.

When is backup possible?

A backup of a system can be performed during program execution, with a few limitations:

- Start program, load program, load module, close program, and erase module cannot be done during backup in executing state. The RAPID instructions Load and StartLoad can, however, be used.
- Do not create backups while performing critical process steps or sensitive robot movements. This may affect the accuracy and performance of the movement. To make sure that no backup is requested, use a system input with the action value Disable Backup (type *System Input*). When the critical steps are done, the input should be reset to allow backups again.

If needed, the backup can be queued while Disable Backup is set, using the system parameter *General RAPID*, with action value QueueBackup set to *TRUE*. Then the backup will be queued until the signal is reset.

(Queueing functionality available from RobotWare 6.11.)

Disable Backup and QueueBackup are described in *Technical reference manual* - *System parameters*.

The system input signal can be set from RAPID for the parts of the code that are critical for disturbances.

What happens during backup?

During the backup process, background tasks continue to execute.

Duplicated modules?

No save operation is performed in the backup command. This implies that two revisions of the same module can exist in the backup, one from the program memory saved in Rapid\Task\Progmod\ directory and one from the HOME directory copied to the backup's home directory. Restoring such a backup will restore both revisions of the module, so the status remains unchanged.

8 Backup and restore

8.2 Important when performing backups *Continued*

Large data amount

Since the HOME directory is included in the backup, large files contained in this folder will make the backup larger. To avoid this situation, you should either clean the HOME directory on regular basis removing the unnecessary files, or keep large files in some other location.

Faults during backup

If a fault occurs during the backup, for example full disk or power failure, the whole current backup is deleted to make sure that only valid fully saved backups are present on the disk.

8.3 Restore the system

When do I need this?

We recommend performing a restore:

- If you suspect that the program file is corrupt.
- If any changes made to the instructions and/or parameters settings did not prove successful, and you want to return to the previous settings.

During the restore, all system parameters are replaced and all the modules from the backup directory are loaded.

The Home directory is copied back to the new system's HOME directory during the restart.

Restore the system

This section describes how to restore the system.

	Action
1	On the ABB menu, tap Backup and Restore.
2	Tap Restore System. The Restore System page is displayed. Auto Motors Off Restore System Restore System A restart occurs when the system is restored, and any unsaved changes to system parameters and modules will be lost. Browse to the backup folder to be used. Then press Restore. Backup folder: C:/Users/sedahal1/Documents/RobotStudio/Systems/BAC KUP/6.03.01_Backup_20160510_CS/
	Advanced Restore Cancel
	xx0300000442
	Note If a default path has been defined as explained in the section <i>Setting default paths on page 77</i> , that path is displayed in the Backup folder path.

8 Backup and restore

8.3 Restore the system *Continued*

	Action		
3	To select the backup folder path or to select The Select Folder window is displayed	ect the TAR backup file form	at, tap
		Motors Off	3 🔽
	IN-L-GISBT14140	Stopped (Speed 100%)	
	Select Folder		
	Name 🛆	Туре	1 to 3 of 3
	RW6.06_Backup_20170622.tar	Tar file	
	RW6.06_Backup_20170622_1	Tar file	
	□ RW6.06_Backup_20170622_2	Folder	
	Selected folder: :/Users/inprkm2/Docum	ents/RobotStudio/Systen	ns/BACKUP
	📑 🖄 🔂	ок	Cancel
	Production Window Backup Restore		[©] ©
	xx1700001303		
4	Select the backup folder or select the bac	kup file which is in TAR forn	nat.
5	Tap OK.		
-	The Restore System page is displayed age path details.	jain with the selected backu	ıp folder or file
	If you want to replace the current controlle backup controller and safety settings, click and Safety Settings check box, and click	er and safety settings with the Advanced , select the Con OK.	he selected t roller Settings
	Guard Guard	Stop 🛐 🔽	
	UPInstaller (ABB-Robotics-123) Stoppe	ed (Speed 100%)	
	Diago coloct the cottings and proce OK to con	tipuo	
	It will overwrite parameters from the backup to	o the current system.	
	Controller Settings (Network, Date and Time, Controller Info	, and UAS)	
	Safety Settings		
		OK Cancel	
	Backup	ROB_1	
	Hestore	<u>v</u> t	
	xx1600001082		

8.3 Restore the system *Continued*

	Action		
6	Tap Restor The followin	e. g screen is displayed.	
		Manual Motors On My_System (IN-L-GISBT14178) Stopped (2 of 2) (Speed 100%)	∑ ₹
		lem estere	
	A restart to syster Browse t	All unsaved changes made to system parameters and modules will be lost.	nges
		Are you sure you want to proceed?	
	Backu C:/U		
		Yes No	
		Restore	Cancel
	Backup		
	xx1400002325		δ ² ε ^λ
7	Tap Yes.		
	The restore	is performed, and the system is restarted.	
	Not	e	
	If there is a warning me	mismatch between the backup and the current syste ssage is displayed.	m, the following
		i) Manual Motors On	33
		MySystem (IN-L-GISBT14178) Stopped (2 of 2) (Speed 100%)	
	🖽 Restore Sys	tem	
	A restart to system		nges
	Browse t Backu	Template IDs mismatch. The system to restore and the current system are different in type, probably generated using different keys; they might differ in robot type, language and options.	
	C:/U	It is not recommended to restore controller data from one system type to another. Do you want to restore anyway?	,
		Yes No	
		Restore	Cancel
	Backup		ROB_1
	Restore		1/3 3
1	xx1400002326		

This page is intentionally left blank

9 Calibrating

9.1 How to check if the robot needs calibration

Check robot calibration status

This section describes how to check the robot's calibration status.

	Action
1	On the ABB menu, tap Calibration.
2	In the list of mechanical units, check the calibration status.

What kind of calibration is needed?

If the calibration status is	then		
Not calibrated	the robot must be calibrated by a qualified service technician.		
Rev. counter update needed	You must update the revolution counters. How to update the revolution counters is de- scribed in the product manual for the robot.		
Calibrated	No calibration is needed.		



Do not attempt to perform the fine calibration procedure without the proper training and tools. Doing so may result in incorrect positioning that may cause injuries and property damage. 9.2 Updating revolution counters

9.2 Updating revolution counters

Overview

This section details how to perform a rough calibration of each robot axis, that is updating the revolution counter value for each axis, using the FlexPendant. Detailed information about revolution counters and how to update them, with calibration positions and scales, can be found in the respective robot product manual. Also, see the manual *Operating manual - Calibration Pendulum* for information on calibration.

For robots using the *Absolute Accuracy* option, the calibration data file absacc.cfg must be loaded first.

Storing the revolution counter setting

This procedure details the second step when updating the revolution counter; storing the revolution counter setting.

	Action					
1	On the ABB menu, tap Calibration . All mechanical units connected to the system are shown along with their calibration status.					
2	2 Tap the mechanical unit in question.					
A screen is displayed: tap Rev. Counters .						
		Manual MySystem (RSTEST4)	Motors On Stopped (2 of 2) (Speed 100%)	ž ž	×	
	Calibration - ROB	1				
	Rev. Counters	Update F	Revolution Counters			
	🔗 Calib. Paramete	rs				
	SMB Memory					
	Base Frame					
				Close	e	
	Calibration				^{(B_1}	
	en0400000771					

9.2 Updating revolution counters *Continued*

	Action			
3	Tap Update Revolution Counters. A dialog box is displayed, warning that updating the revolution counters may change programmed robot positions:			
	 Tap Yes to update the revolution counters. Tap No to cancel updating the revolution counters. 			
	Tapping Yes displays the axis selection window.			
	Note			
	When updating the revolution counters, the ongoing RAPID instruction or function is interrupted, and the path is cleared.			
4	 Select the axis to have its revolution counter updated by: Ticking in the box to the left Tapping Select all to update all axes. Then tap Update. 			
5	 A dialog box is displayed, warning that the updating operation cannot be undone: Tap Update to proceed with updating the revolution counters. Tap Cancel to cancel updating the revolution counters. Tapping Update updates the selected revolution counters and removes the tick from the list of axes. 			
6	! CAUTION			
	If a revolution counter is incorrectly updated, it will cause incorrect manipulator posi- tioning, which in turn may cause damage or injury!			
	See section <i>Checking the calibration position</i> in either of the calibration manuals, depending on which calibration method to be used. The Product manual for the robot also contains more information about calibration.			

Related information

Operating manual - Calibration Pendulum

This page is intentionally left blank

Index

A

ABB menu, 22 approach points, 158 array modify positions, 149 automatic mode limitations, 183 switching to, 242 axes illustration, 100 **Axis Calibration** service routine, 199 В background changing, 79 backup default file path, 77 directory, 273 important, 273 menu, 43 system, 271 backward button, 19 backward execution about, 191 limitations, 191 base coordinates default settings, 99 selecting, 114 baseline concept, 252 target criteria, 252 battery shutdown service routine, 198 buttons controller, 28

С

cabinets, 16 calculation result, 160 calender time counter, 201 calibrating Axis Calibration, 199 CalPendulum, 200 LoadIdentify, 202 calibration, 26 status, 279 touchscreen, 95 calibration menu, 44 CalPendulum service routine, 200 characters entering, 66 international, 66 close button, 22 connection FlexPendant, 232 connector, 18 controller buttons, 28 ports, 28 single cabinet, 16 control tools, overview, 25 coordinate systems

default settings, 99 overview, 99 Quickset, 59 selecting, 59, 114 cursor about, 142 D data instance, 39, 144 data types creating new, 144 editing, 146 menu, 39 viewing, 143 date and time, 89 default paths setting, 77 disable backup, 272 displacements about, 255 work object, 170 Е elongator points define, 160 emergency stop button FlexPendant, 18 enabling device, 18, 20 using, 186 error messages, 68 event log message, 48 expressions offs, 255 positions, 255 F fieldbuses, 35 files programs, 126 filtering about, 70 data types, 70 files, 70 programs, 70 FlexPendant connecting, 232 connecting in operation, 234 connector. 28 disconnecting, 234 hardware buttons, 19 hot plug, 234 how to hold, 21, 87 jumper plug, 234 left-hander, 21 main parts, 18 overview, 17 rotating, 88 screen, 22 FlexPendant Explorer, 34 forward button, 19

Н

hard buttons, 19 hold-to-run, 20, 226 using, 186, 188 HotEdit, 32, 247

using, 251 hot plug, 234 I/O about, 35 changing values, 260 menu, 35 most common, 91 safety signals, 264 simulating, 260 viewing groups, 263 I/O, inputs and outputs, 259 incremental movement definition. 117 quickset, 61 setting size, 117 size settings, 61, 118 insertion point, change, 67 instances data types, 144 instructions backward execution, 191 changing motion mode, 140 commenting, 140 copying and pasting, 140 copying arguments, 140 cutting, 140 editing arguments, 138 handling of, 137 running from a specific, 189 undo, redo, 137 international characters, 66 IRC5 controller, 16 overview, 16 IsBrakeCheckActive, 215

J

jogging about, 97 additional axes, 106 axes in independent mode, 106 coordinated, 107 coordinate systems, 114 non calibrated mechanical units, 106 restrictions, 106 world zones, 106 jogging menu, 36 joystick, 18 locking directions, 115 using, 19 joystick directions about, 105 illustration, 100 locking, 115 jumper plug, 234

Ļ

left-handed, 87 linear mode default settings, 99 linear motion mode, 105 LoadIdentify service routine, 202 loads identifying, 202 log off, 75 log on, 75

Μ

main switch controller, 28 ManLoadIdentify service routine, 213 manual full speed mode about, 185 switching to, 245 manual mode about, 185 limitations, 185 switching to, 244 mechanical unit quickset, 56 selecting, 56, 108 mechanical units activate automatically, 42 modifying positions, 247 arrays, 149 data instances, 146 overview, 246 ModPos. 247 Absolute Limit ModPos, 246 modules creating, 129 deleting, 131 handling of, 129 loading, 130 renaming, 131 saving, 130 most common I/O configuring, 91 motion mode Quickset, 57 selecting, 57, 110 motion pointer, MP, 192 Motion Pointer, MP about, 142 multitasking programs about, 227 loading, running and stopping, 227 viewing, 228

Ν

network settings, 89

0

offsets about, 255 creating, 256 description, 255 operation time counter, 201 operator window, 22, 53 options Calibration Pendulum, 44 duty time counter, 28 hot plug button, 28, 234 IRC5 Compact, 16 **IRC5** Panel Mounted Controller, 16 Levelmeter Calibration, 44 MultiMove, 107 safety chain LEDs, 28 service outlet, 28 service port, 28

Ρ

path returning to, 238 path return region, 238 payloads creating, 177 declarations, 178 deleting, 182 display definitions, 179 editing, 179 editing declarations, 181 identifying, 202 selecting, 111 ports controller, 28 positions about, 119 exact, 119 HotEdit, 247 modifying, 246-247 moving to, 257 naming rules, 82 offset, 255 reading, 119 tuning, 32, 246-247, 251 program a robot, 26 program data editing, 146 menu, 39 program directory, 126 program execution start button, 19 programmable buttons editing, 93 programmable keys editing, 93 program pointer, PP, 192 Program Pointer, PP about, 142 programs about files, 126 creating, 126 default file path, 77 handling of, 126 loading, 127 multitasking, 227 renaming, 128 saving, 127 starting, 223 step by step execution, 191 stopping, 226 Q

queue backup, 272 quickset increments, 61 mechanical unit, 56 Quickset coordinate systems, 59 motion mode, 57 run mode, 62 speed mode, 64 step mode, 63 tasks, 65 tools, 58 work objects, 58 quickset menu, 22

R

RAPID, 124 redo instructions, 137 reorient mode default settings, 99 reorient motion mode, 105 reset button location, 18 using, 19 resolvers about, 119 restart menu, 51 restore default file path, 77 menu, 43 system, 275 revolution counters about, 119 battery shutdown, 198 setting, 280 updating, 280 RobotStudio overview, 24 routines changing declarations, 135 copying, 135 creating, 132 defining parameters, 133 deleting, 135 handling of, 132 running a specific, 190 running service routines, 194 run button, 19 run mode quickset, 62 setting, 62

S

safety I/O signals, 264 scrolling, 69 service port, 28 service routine, 239 service routines Axis Calibration, 199 BatteryShutDown, 198 CalPendulum, 200 LoadIdentify, 202 ManLoadIdentify, 213 running, 194 ServiceInfo. 201 signals changing values, 260 simulating, 260 viewing, 259 SIS, Service Information System counters, 201 service routine, 201 SMB battery shutdown, 198 soft keyboard, 66 speed mode Quickset, 64 setting, 64 start button, 19 status bar, 22, 54

step backward button, 19 step by step execution, 191 step forward button, 19 step mode Quickset, 63 setting, 63 stop button, 19 stylus pen location, 18 using, 19 system backup, 271 restore, 275 system parameters, 26 т targets modifying, 246-247 moving to, 257 naming rules, 82 tuning, 32, 246-247, 251 task bar, 22 tasks debugging, 84 loading program to, 228 normal, static, semistatic, 227 setting up, 227 starting and stopping, 227 tasks panel, 65, 84 three-position enabling device, 18, 20 using, 186 thumb button using, 20 tool, overview control tools, 25 tool center point about, 154 calculation result, 160 define, 159 defining, 160 measuring, 162 TCP, 154 working area variations, 160 tool coordinates default settings, 99 selecting, 114 tool frame defining, 157 methods, 157 reorientation test, 160 tool orientation, 160 setting, 112 tool orientation, definition, 112 tools aligning, 121 creating, 154 deleting, 165 editing declarations, 164 editing definitions, 162

editing tool data, 161 identifying loads, 202 make stationary, 166 Quickset, 58 selecting, 58, 111 setting up tool coordinate system, 167 stationary, 166 Total Load, 177 touchscreen, 22 background image, 79 brightness, 86 calibrating, 95 touch screen rotating, 88 troubleshooting, 237 tuning HotEdit, 247 positions, 246-247, 251 targets, 246-247

U UAS

configuring views, 80 uncalibrated mechanical unit, 239 undo instructions, 137 USB port FlexPendant, 18

۷

viewing messages in programs, 53 view settings additional test views, 81 configuring, 78

W

work object coordinates selecting, 114 work objects creating, 169 declarations, 169 defining coordinate system, 170 deleting, 176 displacements, 170 editing declarations, 175 editing work objects data, 174 Quickset, 58 selecting, 58, 111 world coordinates selecting, 114 wrist optimization, 221 write access granting, 74 message, 68 rejecting, 74

Z

zooming, 69



ABB AB Robotics & Discrete Automation S-721 68 VÄSTERÅS, Sweden Telephone +46 10-732 50 00

ABB AS

Robotics & Discrete Automation Nordlysvegen 7, N-4340 BRYNE, Norway Box 265, N-4349 BRYNE, Norway Telephone: +47 22 87 2000

ABB Engineering (Shanghai) Ltd.

Robotics & Discrete Automation No. 4528 Kangxin Highway PuDong New District SHANGHAI 201319, China Telephone: +86 21 6105 6666

ABB Inc.

Robotics & Discrete Automation 1250 Brown Road Auburn Hills, MI 48326 USA Telephone: +1 248 391 9000

abb.com/robotics