



ROBOTICS

Application manual

Wizard



Trace back information:
Workspace Main version a723
Checked in 2025-12-01
Skribenta version 5.6.019

Application manual

Wizard

1.7.1

Document ID: 3HAC073766-001

Revision: L

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 2020-2025 ABB. All rights reserved.
Specifications subject to change without notice.

Table of contents

Overview of this manual	7
Network security	10
1 Introduction	11
2 Getting started	17
3 Programming with Wizard	21
3.1 Communicating with the FlexPendant	21
3.2 Controlling robot motion	22
3.2.1 Moving the tools	22
3.2.2 Stopping and waiting	23
3.3 Controlling the fingers and suction cups	24
3.4 Operating the gripper	25
3.5 Picking up and placing with fingers and suction cups	26
3.6 Picking up and placing with the gripper	28
3.7 Operation repetitions	29
3.8 Working with I/O signals	30
3.9 Logical executions	31
3.10 Programming procedures in subworkspace	35
3.11 Searching and operating with force control	37
3.12 Customizing comments or RAPID instructions	38
3.13 Working with ASI	39
3.14 Working with lead-through device	40
3.15 Using Favorites for your blocks	41
4 Defining parameters and data	43
4.1 Parameters	43
4.2 Data definition	47
4.2.1 Defining a location	47
4.2.2 Defining an object	50
4.2.3 Defining a hole	51
4.2.4 Defining a tray	53
4.2.5 Defining a variable	54
4.2.6 Defining a custom data	56
4.3 Operations	58
4.4 Shortcut menu	62
4.5 Skill creator	65
Index	67

This page is intentionally left blank

Overview of this manual

About this manual

This manual introduces the user interface and instructions contained in the blocky programming application Wizard on the FlexPendent.

Usage

This manual should be read during robot programming.

Who should read this manual?

This manual is intended for robot programmers, especially the beginners.

Prerequisites

The reader should:

- Be trained in robot operation.

References

Documentation referred to in the manual, is listed in the table below.

Document name	Document ID
<i>Operating manual - Emergency safety information</i>	3HAC027098-001
<i>Safety manual for robot - Manipulator and IRC5 or OmniCore controller</i>	3HAC031045-001
<i>Operating manual - OmniCore</i>	3HAC065036-001
<i>Product manual - OmniCore C30</i>	3HAC060860-001
<i>Product manual - IRB 14050</i>	3HAC064625-001
<i>Product manual - Grippers for IRB 14050</i>	3HAC064626-001
<i>Product manual - IRB 1010</i>	3HAC081964-001
<i>Product manual - IRB 1090</i>	3HAC088056-001
<i>Product manual - IRB 1100</i>	3HAC064992-001
<i>Product manual - IRB 1200</i>	3HAC046983-001
<i>Product manual - IRB 1300</i>	3HAC070390-001
<i>Product manual - IRB 2600</i>	3HAC035504-001
<i>Product manual - IRB 5710</i>	3HAC075184-001
<i>Product manual - IRB 5720</i>	3HAC079195-001
<i>Product manual - IRB 6650S</i>	3HAC020993-001
<i>Product manual - IRB 6700</i>	3HAC044266-001
<i>Product manual - IRB 6710</i>	3HAC085696-001
<i>Product manual - IRB 6720</i>	3HAC085697-001
<i>Product manual - IRB 6730</i>	3HAC085699-001
<i>Product manual - IRB 6740</i>	3HAC085701-001
<i>Product manual - CRB 1100</i>	3HAC078007-001
<i>Product manual - CRB 1300</i>	3HAC083111-001

Continues on next page

Document name	Document ID
Product manual - CRB 1810	3HAC093218-001
Product manual - CRB 15000	3HAC077389-001

Revisions

Revision	Description
A	First edition.
B	Released with RobotWare 7.1. The following updates are made in this revision: • Supported programming on IRB 1100. • Updated description about trigger conditions for the if block and while loop block. • Added description about the Variables blocks. • Added description about Skill Creator.
C	Released with RobotWare 7.4. The following updates are made in this revision: • Supported programming on CRB 1100, CRB 15000 and IRB 1300. • Added network security disclaimer. • Added Message blocks and Stop & Wait blocks • Updated descriptions of buttons in the user interface. • Updated values of predefined motion speed parameter in Move blocks. • Updated procedures about location modification. • Added Play from this block and Update Location to shortcut commands. • Updated description about Skill Creator.
D	Released with RobotWare 7.6.1. The following updates are made in this revision: • Added Gripper blocks and Force blocks. • Added parameters used in Gripper blocks and Force blocks. • Added Gripper operations. • Added procedure about defining a hole.
E	Released with RobotWare 7.7. The following updates are made in this revision: • Supported to work with lead-through device for SWIFTI™ robots. • Added figures to show the user interface.
F	Released with RobotWare 7.8. The following updates are made in this revision: • Added function of binding blocks with I/O. • Added shortcut command Open Application. • Supported programming on CRB 1300. • Removed the <i>Procedure_name</i> block from Procedure block group and added the Add Procedure button to support working with procedures in subworkspace. • Updated description of the call <procedure> block. • Added notes to the added data. • Added shortcut commands Edit Procedure and Delete Procedure. • Updated shortcut command Update Location.

Continues on next page

Revision	Description
G	Released in Wizard Easy Programming add-in version 1.5.1. The following updates are made in this revision: Supported programming on IRB 1010, IRB 1200, IRB 2600, IRB 4600, IRB 5710, IRB 5720, IRB 6650S, IRB 6700, IRB 6710, IRB 6720, IRB 6730 and IRB 6740.
H	Released in Wizard Easy Programming add-in version 1.5.2. The following updates are made in this revision: - Supported programming on IRB 1090. - Added parameters used in Move blocks. - Added Script blocks. - Updated function description of binding blocks with I/O. - Updated shortcut command list.
J	Released in Wizard Easy Programming add-in version 1.6.0. The following updates are made in this revision: - Updated the figures to show the latest UI design. - Updated description of the Move block. - Restructured the shortcut command list. - Added user cases for Logic blocks and Procedures blocks.
K	Released in Wizard Easy Programming add-in version 1.7.0. The following updates are made in this revision: - Updated the list of robots supporting programming using Wizard. - Combined Move block and Stop & Wait blocks into Motion category. - Updated figures to show the new user interface of Wizard and the procedure to get start. - Added a toggle to enable/disable program editing and removed the Apply button for automatically applying the changes. - Added function description of adding favorite blocks. - Updated the shortcut command list.
L	Released in Wizard Easy Programming add-in version 1.7.1. The following updates are made in this revision: - Supported programming on CRB 1810. - Added information about defining custom data.

Network security

Network security

This product is designed to be connected to and to communicate information and data via a network interface. It is your sole responsibility to provide, and continuously ensure, a secure connection between the product and to your network or any other network (as the case may be).

You shall establish and maintain any appropriate measures (such as, but not limited to, the installation of firewalls, application of authentication measures, encryption of data, installation of anti-virus programs, etc) to protect the product, the network, its system and the interface against any kind of security breaches, unauthorized access, interference, intrusion, leakage and/or theft of data or information. ABB Ltd and its entities are not liable for damage and/or loss related to such security breaches, any unauthorized access, interference, intrusion, leakage and/or theft of data or information.

1 Introduction

Overview

Wizard is a programming application available on the FlexPendant. It provides an easy and interactive programming means, enabling users to program by simply combining visible instruction blocks and setting pre-defined parameters.

Supported robot type



Note

Robots supported by the latest Wizard version are listed. Contact ABB Service for any updated information about the support on other robots not listed.

Articulated robots

- IRB 1010 (supported from RobotWare 7.10)
- IRB 1090 (supported from RobotWare 7.12)
- IRB 1100 (supported from RobotWare 7.1)
- IRB 1200 (supported from RobotWare 7.10)
- IRB 1300 (supported from RobotWare 7.4)
- IRB 1510 (supported from RobotWare 7.15)
- IRB 1520 (supported from RobotWare 7.15)
- IRB 1600 (supported from RobotWare 7.15)
- IRB 2400 (supported from RobotWare 7.15)
- IRB 2600 (supported from RobotWare 7.10)
- IRB 4400 (supported from RobotWare 7.15)
- IRB 4600 (supported from RobotWare 7.10)
- IRB 5710/5720 (supported from RobotWare 7.10)
- IRB 6650S (supported from RobotWare 7.10)
- IRB 6700 (supported from RobotWare 7.10)
- IRB 6710/6720/6730/6740 (supported from RobotWare 7.10)
- IRB 6790 (supported from RobotWare 7.15)
- IRB 7600 (supported from RobotWare 7.15)
- IRB 7710/7720 (supported from RobotWare 7.15)
- IRB 8700 (supported from RobotWare 7.15)

Palletizer robots

- IRB 460 (supported from RobotWare 7.15)
- IRB 660 (supported from RobotWare 7.15)
- IRB 760 (supported from RobotWare 7.15)

SCARA robots

- IRB 910INV (supported from RobotWare 7.15)
- IRB 920 (supported from RobotWare 7.15)

Continues on next page

1 Introduction

Continued

- IRB 930 (supported from RobotWare 7.15)

Delta robots

- IRB 360 (supported from RobotWare 7.15)
- IRB 365 (supported from RobotWare 7.15)
- IRB 390 (supported from RobotWare 7.15)

Collaborative robots

- IRB 14050 (supported from RobotWare 7.0)
- CRB 1100 (supported from RobotWare 7.2)
- CRB 1300 (supported from RobotWare 7.8)
- CRB 1810 (supported from RobotWare 7.20)
- CRB 15000 (supported from RobotWare 7.2)

Safety information

Before using the Wizard to program, make sure that all safety information included in but not restricted to the following manuals are acknowledged and necessary safety measures are conducted.

All documents can be found via ABB Robotics One, <https://one.robotics.abb.com/en/knowledge-hub>.

Controllers
• <i>Operating manual - Emergency safety information</i> (3HAC027098) • <i>Operating manual - General safety information</i> (3HAC031045) • <i>Operating manual - OmniCore</i> (3HAC065036) • <i>Product manual - OmniCore C30</i> (3HAC060860) • <i>Product manual - OmniCore C30 Type A</i> (3HAC089064) • <i>Product manual - OmniCore C90XT</i> (3HAC073706) • <i>Product manual - OmniCore C90XT Type A</i> (3HAC089065) • <i>Product manual - OmniCore V250XT Type B</i> (3HAC087112) • <i>Product manual - OmniCore V400XT</i> (3HAC081697)
Articulated robots

Continues on next page

• <i>Product manual - IRB 1010</i> (3HAC081964) • <i>Product manual - IRB 1090</i> (3HAC088056) • <i>Product manual - IRB 1100</i> (3HAC064992) • <i>Product manual - IRB 1200</i> (3HAC046983) • <i>Product manual - IRB 1300</i> (3HAC070390) • <i>Product manual - IRB 1510</i> (3HAC087870) • <i>Product manual - IRB 1520</i> (3HAC043435) • <i>Product manual - IRB 1600/1660</i> (3HAC026660) • <i>Product manual - IRB 2400</i> (3HAC022031) • <i>Product manual - IRB 2600</i> (3HAC035504) • <i>Product manual - IRB 4400</i> (3HAC022032) • <i>Product manual - IRB 4600</i> (3HAC033453) • <i>Product manual - IRB 5710</i> (3HAC075184) • <i>Product manual - IRB 5720</i> (3HAC079195) • <i>Product manual - IRB 6650S</i> (3HAC020993) • <i>Product manual - IRB 6660</i> (3HAC028197) • <i>Product manual - IRB 6700</i> (3HAC044266) • <i>Product manual - IRB 6710</i> (3HAC085695) • <i>Product manual - IRB 6720</i> (3HAC085697) • <i>Product manual - IRB 6730</i> (3HAC085699) • <i>Product manual - IRB 6740</i> (3HAC085701) • <i>Product manual - IRB 6790</i> (3HAC063331) • <i>Product manual - IRB 7600</i> (3HAC022033) • <i>Product manual - IRB 7710</i> (3HAC089600) • <i>Product manual - IRB 7720</i> (3HAC089605) • <i>Product manual - IRB 8700</i> (3HAC052853)
Palletizer robots
• <i>Product manual - IRB 460</i> (3HAC039842) • <i>Product manual - IRB 660</i> (3HAC025755) • <i>Product manual - IRB 760</i> (3HAC039838)
SCARA robots
• <i>Product manual - IRB 910INV</i> (3HAC068055) • <i>Product manual - IRB 920</i> (3HAC075721) • <i>Product manual - IRB 930</i> (3HAC086009)
Delta robots
• <i>Product manual - IRB 360</i> (3HAC030005) • <i>Product manual - IRB 365</i> (3HAC079185) • <i>Product manual - IRB 390</i> (3HAC066566)
Cobots
• <i>Product manual - IRB 14050</i> (3HAC064625) • <i>Product manual - Grippers for IRB 14050</i> (3HAC064626) • <i>Product manual - CRB 1100</i> (3HAC078007) • <i>Product manual - CRB 1300</i> (3HAC083111) • <i>Product manual - CRB 1810</i> (3HAC093218) • <i>Product manual - CRB 15000</i> (3HAC077389)

Prerequisites

To work with Wizard, the followings are required:

- OmniCore controller operating in RobotWare 7.0 or later
- Installed Wizard add-in

Continues on next page



Note

The Wizard add-in version is backward compatible with RobotWare versions. However, an older version of the Wizard add-in cannot work with a newer RobotWare version.

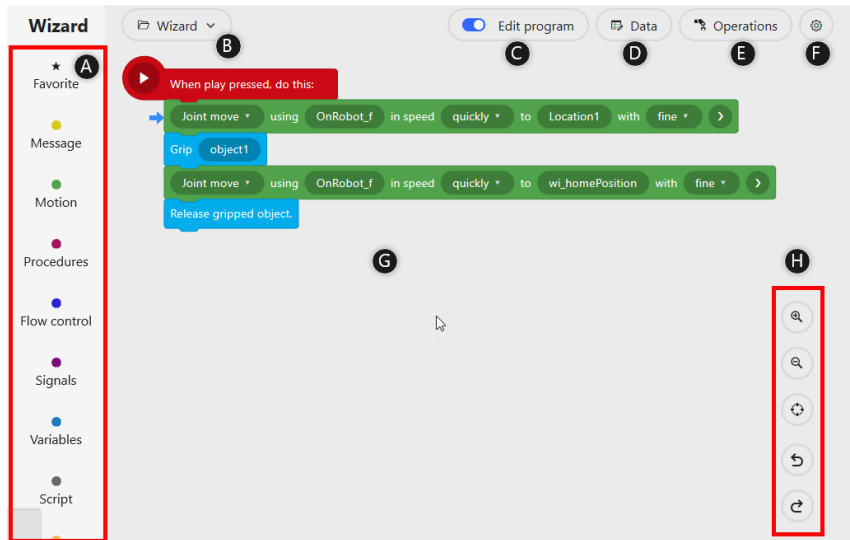


Note

For IRB 14050 working with controller in RobotWare 7.0, Smart Gripper, Collision Detection [3107-1] and Multitasking [3114-1] must be selected during setup of the robot system.

For IRB 14050 working with controller in other RobotWare versions, the Smart Gripper add-in is optional. However, if Smart Gripper is selected, Collision Detection [3107-1] and Multitasking [3114-1] must be selected together.

The user interface



xx1900002111

Label	Buttons/Section	Description
A	Block category	Tapping each block category button displays a list of available blocks in this category. During programming, tap the required block and drag it to the editing area. You can sort and set visibility of the categories in the category manager by tapping the Settings button in the right corner of the window and then tap Category Manager.

Label	Buttons/Section	Description	
B	File menu ^{note}	New	Clears the blocks in the editing area and creates a new program. If there is a program under editing, a dialog displays, prompting to discard or save the program.
		Save	Saves the current program to the default directory \$HOME/Programs/.
		Save as	Saves the current program under a new name in the default directory \$HOME/Programs/.
		Load	Loads a program from the default program saving directory \$HOME/Programs/.
		Rename	Renames the current program.
C	Edit program	The Wizard is in read-only mode by default. Tapping Edit program to turn on the editing mode. In read-only mode, users can only, - view the program, blocks and variables, but cannot use or edit them. - play the program. - use the Start from this block and Edit Procedure commands from the shortcut menu.	
D	Data	Tapping the Data button displays the quickset window for setting data and variables.	
E	Operations	Tapping the Operation button allows a quick access to basic gripper-related operations.	
F	Settings	Tapping the Settings button to access the help contents, such as tutorial, user manual and Wizard version information. You can click Feedback to give comments or suggestions.	
G	Editing area	The main programming window of the Wizard. Users can easily sequence and edit the blocks in this area.	
H	View settings		Removes more than one action until the program returns to what you previously had.
			Redoes more than one action.
			Aligns the blocks/program to the left corner.
			Zooms in the window view.
			Zooms out the window view. Double-tapping the icon will directly zoom out the display to fit to window.

Continues on next page



Note

In the main workspace, the file menu is titled with the program name. Tapping on it displays the file functions.

In the subworkspace, the file menu is titled with the procedure name together with its belonged module name. Tapping on it will go back to the main workspace.

To delete unnecessary block(s), tap and press the block(s) and drag them to the left pane of the window. Release the tapping when a trash can icon displays. Note that deletion cannot be undone.

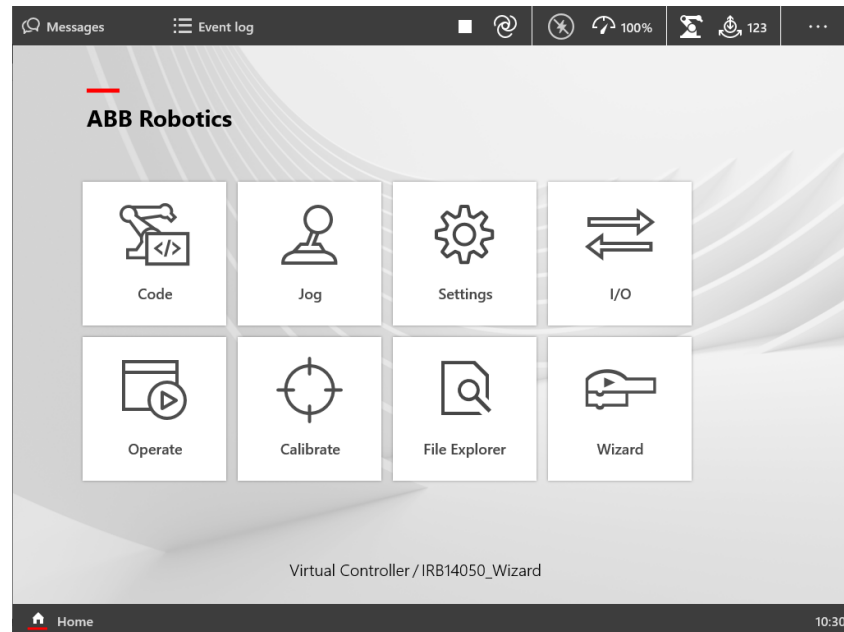


xx2000000202

2 Getting started

General

Wizard is a default application on the FlexPendant for supported robots (see [Supported robot type on page 11](#)). You can access the main window of Wizard directly by tapping the following application icon on the FlexPendant touch screen.



xx2000000258

Procedure

- 1 Tap **Edit program** to turn on editing mode.



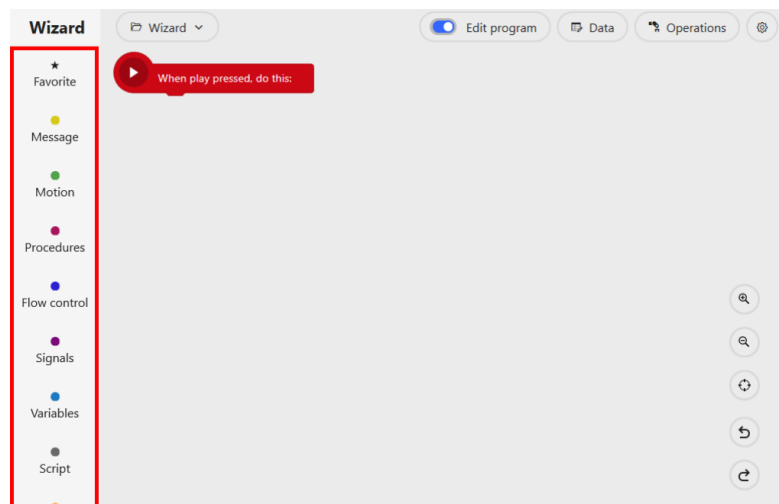
xx2400001971

- 2 Tap the required block category icon to display the list.

Continues on next page

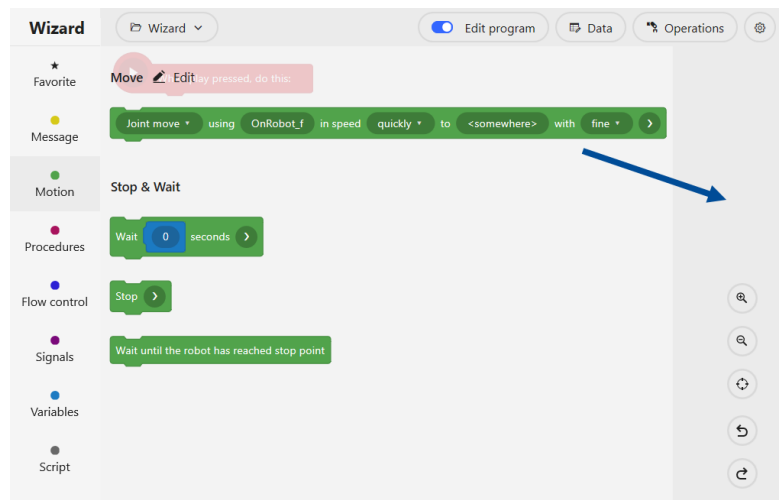
2 Getting started

Continued



xx1900002112

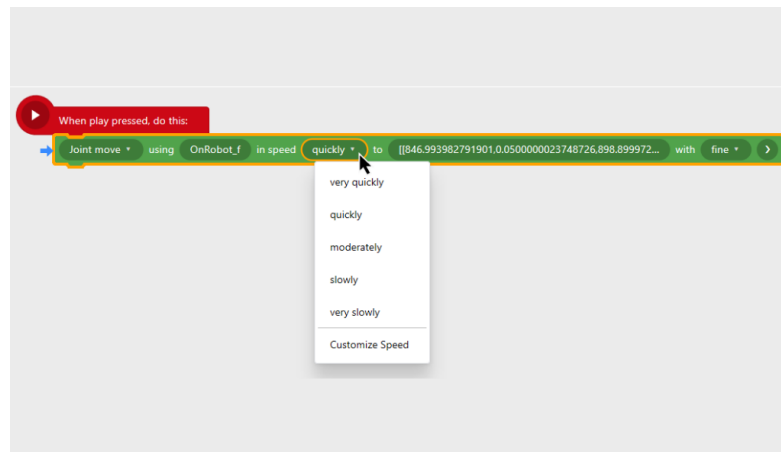
3 Tap or drag the required block to the editing area.



xx1900002113

4 Set necessary parameters by choosing from the predefined list or a customized list.

Continues on next page



xx1900002114

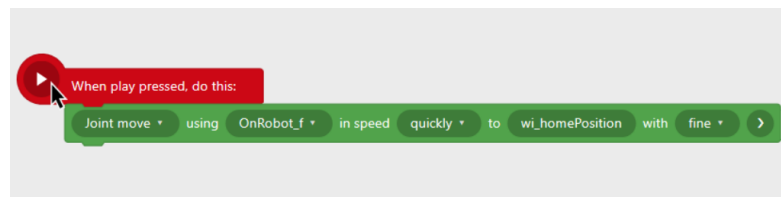
- 5 Repeat steps 2 and 3 until the program is created or modified as required.



Tip

You can drag the blocks to adjust their sequences.
The modifications are applied to the controller automatically.

- 6 Tap the **Play** button to run the program.



xx1900002116



Note

To view the corresponding RAPID data of the program edited in the Wizard,

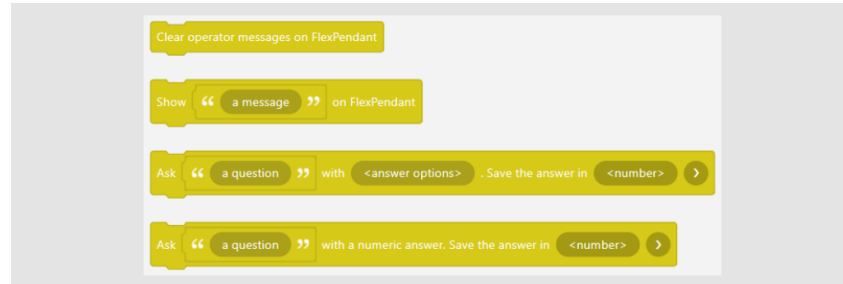
- 1 Tap the **Home** button on the FlexPendant.
- 2 Tap **Code** on the start screen.
- 3 Tap **Wizard** in the displayed **Modules** page.

This page is intentionally left blank

3 Programming with Wizard

3.1 Communicating with the FlexPendant

Block illustration



xx2200001333

Descriptions

Users can define the texts written on the FlexPendant display, which could be displayed as messages to show information, as questions to ask users to operate or as answers read from the FlexPendant. Available message blocks are listed as follows:

- Clear operator messages on FlexPendant
- Show *a message* on FlexPendant
- Ask *a question* with *<answer options>*. Save the answer in *<number>*.: label the function keys and to read which key is pressed. This block allows to set optional parameters.
- Ask *a question* with a numeric answer. Save the answer in *<number>*.: read a numeric value from the FlexPendant. This block allows to set optional parameters.

For details about the numeric variable in the blocks and optional parameters, see [Parameters on page 43](#).

3 Programming with Wizard

3.2.1 Moving the tools

3.2 Controlling robot motion

3.2.1 Moving the tools

Block illustration



xx2200001334

Descriptions

Use the **Move** block to move the tool to the destination location along a non-linear path or linearly.

- *Joint move tool0 quickly to <somewhere> with fine*

The motion type can be selected between **Joint move** and *Linear move*.

If a gripper has been configured and selected in Wizard, the gripper is set as the tool by default. You can change to another required tool from the list.

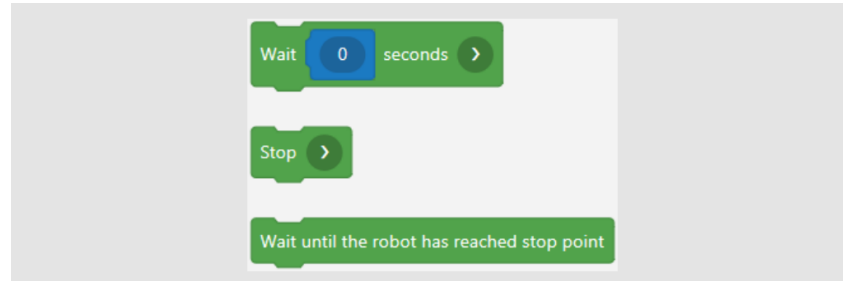
For IRB 14050 with an ABB Smart Gripper, the tool is set to **Finger** by default. See [Fingers on page 58](#) for finger settings. It can also be set to **VacuumX** if one or two vacuum modules are configured for the smart gripper. See [Vacuum modules on page 59](#) for definition of **Vacuum1** and **Vacuum2**.

The movement speed and zone data are set to **quickly** and **fine** by default, respectively. It can be reset to another value by users.

The **Move** blocks also allow to set optional parameters. For details about the parameters, including optional parameters, in the blocks, see [Parameters on page 43](#).

3.2.2 Stopping and waiting

Block illustration



Descriptions

Use the **Stop & Wait** blocks to control the robot motion stopping or waiting until a defined condition is met.

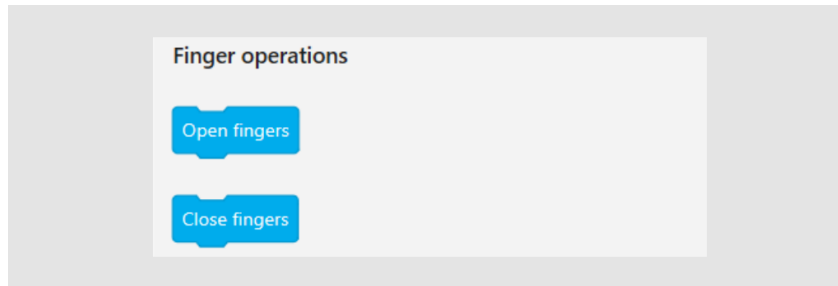
- Wait X seconds
- Stop
- Wait until the robot has reached stop point

3 Programming with Wizard

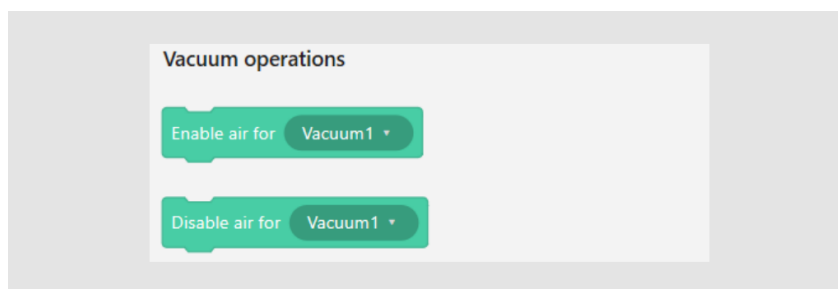
3.3 Controlling the fingers and suction cups

3.3 Controlling the fingers and suction cups

Block illustration



xx2200001335



xx2200001336

Descriptions



Note

The Finger and Air blocks are available only for IRB 14050 configured with ABB Smart Gripper.

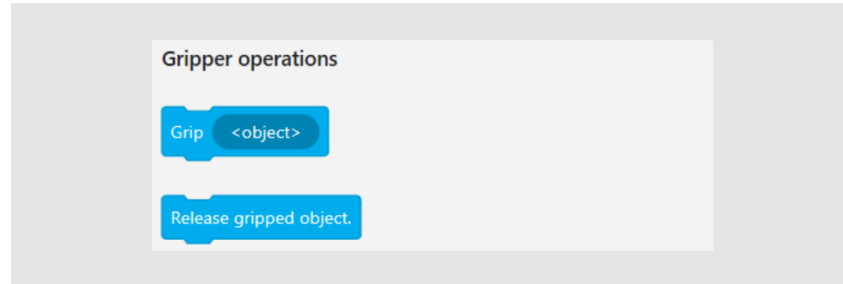
To open and close the gripper fingers, use the Open finger and Close finger blocks in the **Finger** block group.

The gripper can also be configured with one or two vacuum modules. To enable and disable air supply to a suction cup, use the Enable air for VacuumX and Disable air for VacuumX blocks in the **Air** block group.

For details about the parameters in the blocks, see [Parameters on page 43](#).

3.4 Operating the gripper

Block illustration



xx2200001337

Descriptions



Note

The Gripper blocks are available for robots (except IRB 14050) configured with a gripper (other than ABB Smart Gripper) and its corresponding application add-in.

To grip an object using a configured gripper or release the object gripped by the gripper, use the Grip *<object>* and Release gripped object blocks in the **Gripper** block group.

If the gripper is configured with fingers and suction cups, and both of them have objects gripped, the Release gripped object block will release the object gripped by fingers first.

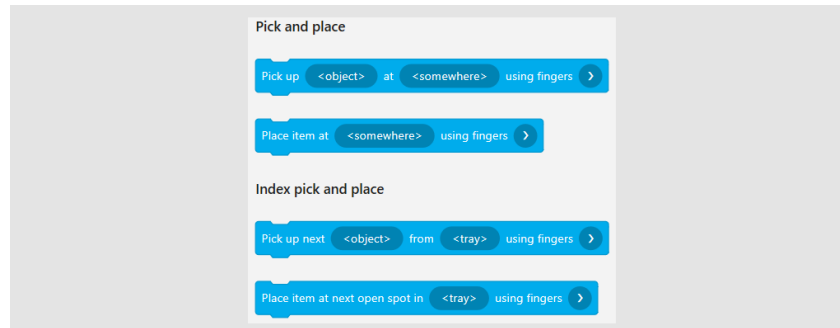
For details about the parameters in the blocks, see [Parameters on page 43](#).

3 Programming with Wizard

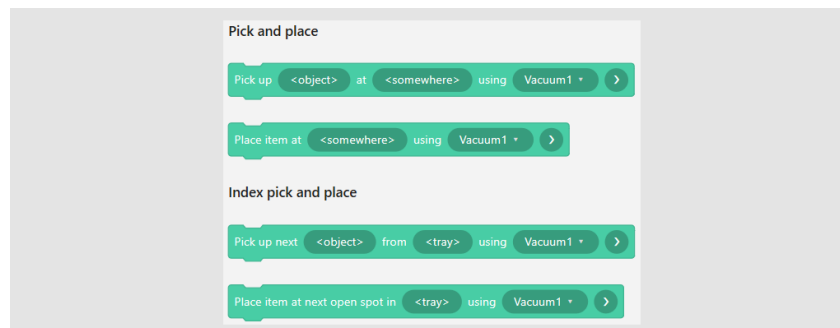
3.5 Picking up and placing with fingers and suction cups

3.5 Picking up and placing with fingers and suction cups

Block illustration



xx2200001339



xx2200001338

Descriptions



Note

Finger- and air-related blocks are available only for IRB 14050 with gripper(s).

Grippers can pick up and place objects using fingers or suction cups if configured. Available blocks are listed as follows.

Use...	To...	Single object	Objects in a tray
Fingers	Pick	Pick up <i><object></i> at <i><somewhere></i> using finger	Pick up next <i><object></i> from <i><tray></i> using finger
	Place	Place item at <i><somewhere></i> using finger	Place item at next open spot in <i><tray></i> using finger
Suction cups	Pick	Pick up <i><object></i> at <i><somewhere></i> using VacuumX	Pick up next <i><object></i> from <i><tray></i> using VacuumX
	Place	Place item at <i><somewhere></i> using VacuumX	Place item at next open spot in <i><tray></i> using VacuumX

The picking and placing blocks must be used in pair. If the gripper uses suction cups to pick up and place, the suction cup selected in a pair of picking and placing blocks must be the same one.

Continues on next page

In scenarios where the gripper needs to pick up same objects that are orderly arranged in a tray and then place them to another tray for several times, use the **Index pick and place** blocks in which a tray is defined.

For details about the parameters in the blocks, see [Parameters on page 43](#).

Failure handling

For picking and placing blocks, it is possible to define the gripper behavior if the operation experiences a failure. Tapping the > button after a picking/placing block displays the **on failure** parameter.

When a failure occurs,

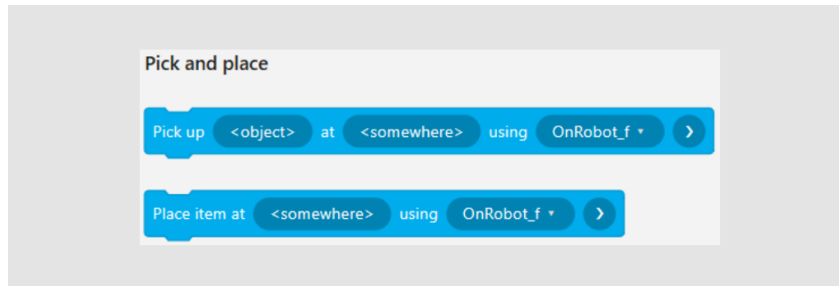
- if **pause** is set, the gripper stops until users restore it manually;
- if **wait and retry** is set, the gripper will stop for a short period and then retry the operation until it succeeds.

3 Programming with Wizard

3.6 Picking up and placing with the gripper

3.6 Picking up and placing with the gripper

Block illustration



xx2200001340

Descriptions



Note

Gripper-related blocks are available for robots (except IRB 14050) configured with a gripper (other than ABB Smart Gripper) and its corresponding application add-in.

Grippers can pick up and place objects using any available configured tools.

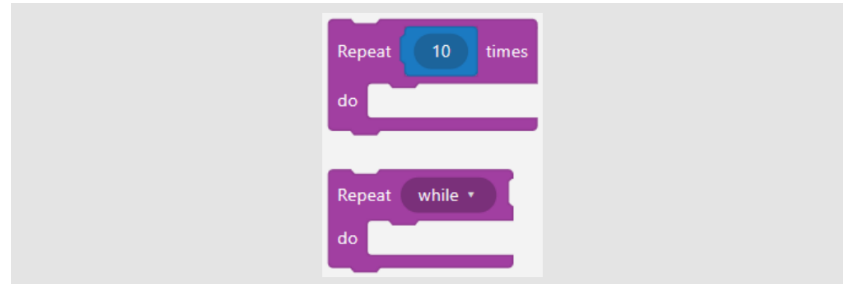
- Pick up *<object>* at *<somewhere>* using *<tool>*
- Place item at *<somewhere>* using *<tool>*

The picking and placing blocks must be used in pair, and the tool selected in a pair of picking and placing blocks must be the same one.

The blocks also allow to set optional parameters. For details about the parameters, including optional parameters, in the blocks, see [Parameters on page 43](#).

3.7 Operation repetitions

Block illustration



xx2200001342

Descriptions

Operations can repeat for several times with a loop block is set. It can be a loop with defined repeated time or a while/until loop.

- `repeat X times`: defines the number that the blocks are to be repeated. The *X* parameter in the block must be set to a value equal to or larger than 1. The default repeated time is 10.
- `repeat while <condition> do <something>`: used when the blocks need to repeat under a condition. The judgement parameter can be set to *while* or *until*.

For details about the conditions, see [Trigger conditions on page 32](#).

3 Programming with Wizard

3.8 Working with I/O signals

3.8 Working with I/O signals

Block illustration



xx2200001343

Descriptions

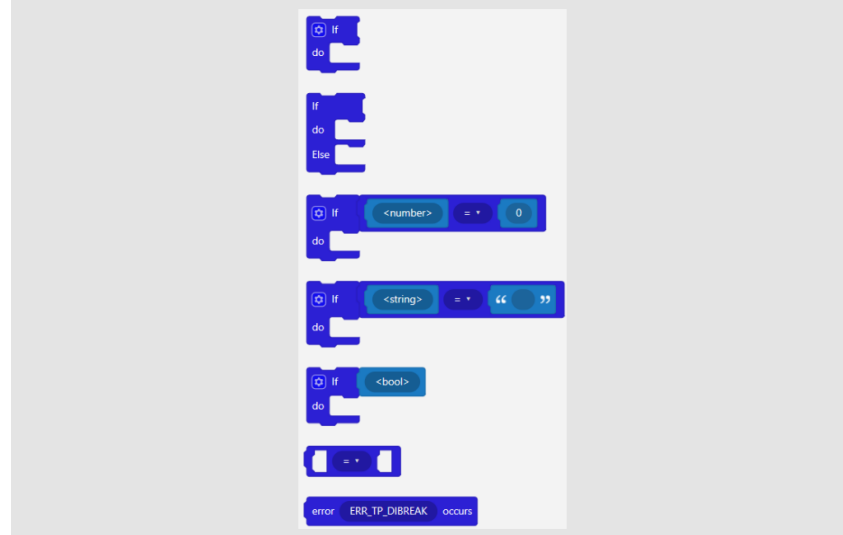
The signal blocks control the responses of the gripper with the I/O signal values **1-True** and **0-False**. Available signal blocks are listed as follows:

- set signal *<signal name>* to *1-True*
- send pulse on signal *<signal name>*
- wait until DO signal *<signal name>* become *1-True*
- wait until DI signal *<signal name>* become *1-True*
- if signal *<signal name>* is *1-True* do
- signal *<signal name>* is *1-True*

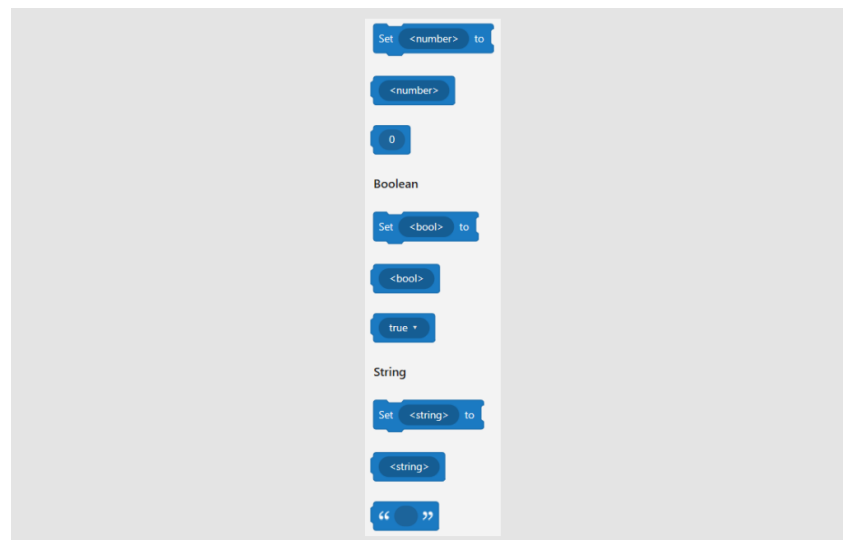
For details about the parameters in the blocks, see [Parameters on page 43](#).

3.9 Logical executions

Block illustration



xx2200001344



xx2200001345

Descriptions

To execute an operation under a specified condition, the **if** and **if else** blocks in the **Logic** block group are used. It is also possible to expand the **if** block to the **else** and **else if** blocks with more conditions by tapping the button at the left corner. Three predefined **if** block with variable conditions are also available for use directly.

Continues on next page

3 Programming with Wizard

3.9 Logical executions

Continued

Trigger conditions

Both the if block and while loop block require a condition to take effect. The condition can be a signal, a variable, a logical expression, an error or a customized RAPID instruction.

- The signal *<signal name>* is 1-True block is used as the signal condition.
- A variable can be in number type, boolean type or string type. Both the variable name and variable value can be used as a condition.
- A logical expression can be used as a condition, in which the relationship between two variables are defined. The variables on the right and left sides of the relations operator must be the same type.
- The error *<error name>* occurs block is used as the error condition.
- The Custom code *<Insert the RAPID>* block is used as RAPID condition.

For details about the parameters in the blocks, see [Parameters on page 43](#).

Variable blocks

In the Variable block group, three types of variables are available, number, boolean and string.

Both variable name and variable value can be user-defined. The Set *<variable name>* to block is used to assign a value to a specific variable. The default values of the numeric, boolean and string variables are 0, true and default, respectively.

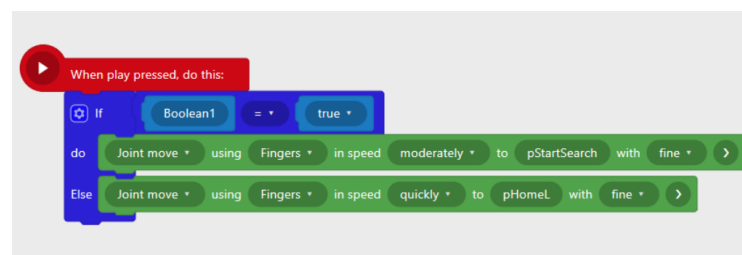
All the defined variables are listed in corresponding tab in Data window.

For details about the parameters in the blocks, see [Parameters on page 43](#).

User cases

More examples of how to use the logic block with conditions are illustrated below.

Case 1 - Logic block with a boolean-type condition



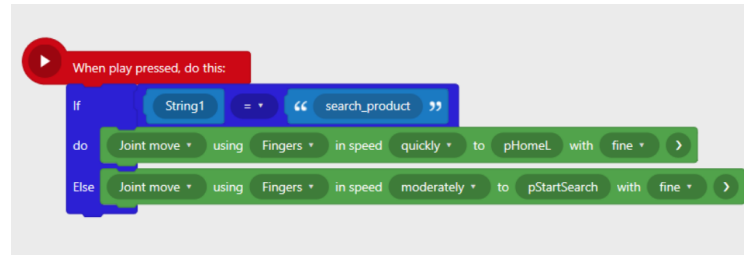
xx2100001715

The case shows the robot behaviors when a boolean-type condition is met.

When the value of a boolean-type variable Boolean1 is true, the robot moves its gripper fingers linearly using MoveL in a moderate speed (default value v200 or a user-defined value) to position pStartSearch. Otherwise, the robot moves its gripper fingers linearly using MoveL in a quick speed (default value v500 or a user-defined value) to position pHomeL.

Continues on next page

Case 2 - Logic block with a string-type condition

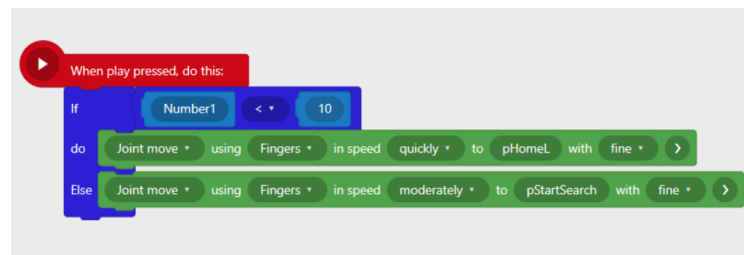


xx2100001716

The case shows the robot behaviors when a string-type condition is met.

When the value of a string-type variable String1 is not search_product, the robot moves its gripper fingers linearly using MoveL in a quick speed (default value v500 or a user-defined value) to position pHomeL. Otherwise, the robot moves its gripper fingers linearly using MoveL in a moderate speed (default value v200 or a user-defined value) to position pStartSearch.

Case 3 - Logic block with a number-type condition

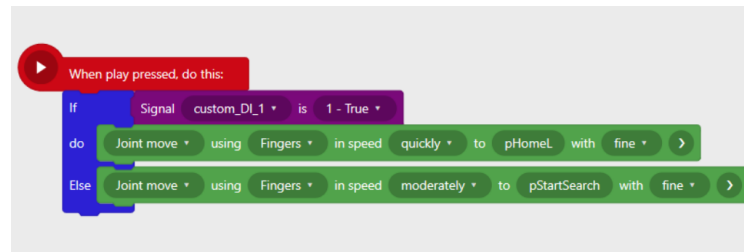


xx2100001717

The case shows the robot behaviors when a number-type condition is met.

When the value of a number-type variable Number1 is smaller than 10, the robot moves its gripper fingers linearly using MoveL in a quick speed (default value v500 or a user-defined value) to position pHomeL. Otherwise, the robot moves its gripper fingers linearly using MoveL in a moderate speed (default value v200 or a user-defined value) to position pStartSearch.

Case 4 - Logic block with a signal condition



xx2100001718

The case shows the robot behaviors when a signal condition is met.

When the signal custom_DI_1 is 1 - True, the robot moves its gripper fingers linearly using MoveL in a quick speed (default value v500 or a user-defined value) to position pHomeL. Otherwise, the robot moves its gripper fingers linearly using

Continues on next page

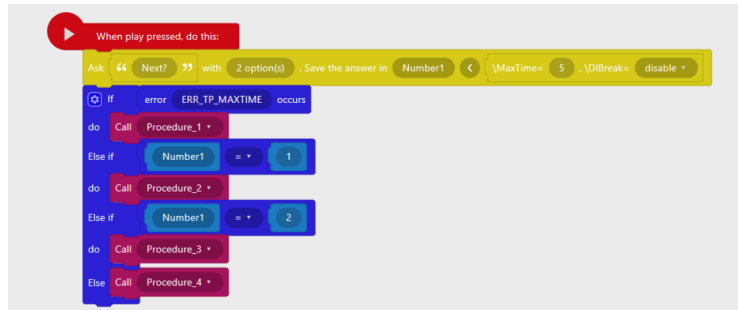
3 Programming with Wizard

3.9 Logical executions

Continued

MoveL in a moderate speed (default value v200 or a user-defined value) to position pStartSearch.

Case 5 - Logic block with function keys



xx2100001719

The case shows the robot behaviors when function keys are set.

The text Next? is written on the FlexPendant display and the function keys 1 and 2 are activated by means of the text strings Yes and No respectively. The number-type variable Number1 will be assigned 1 or 2 depending on which of the key is pressed. The program execution will wait for a maximum of 5 seconds until one of the function keys is pressed. If no function key is pressed within 5 seconds, the program continues to execute in the error handler. ERR_TP_MAXTIME can be used to test whether or not this maximum 5 seconds has elapsed. Inputting digital signal to interrupt the operator dialog is disabled.

If the maximum waiting time 5 seconds has elapsed with no function key is pressed, that is the error ERR_TP_MAXTIME occurs, the program defined in procedure Program_1 executes.

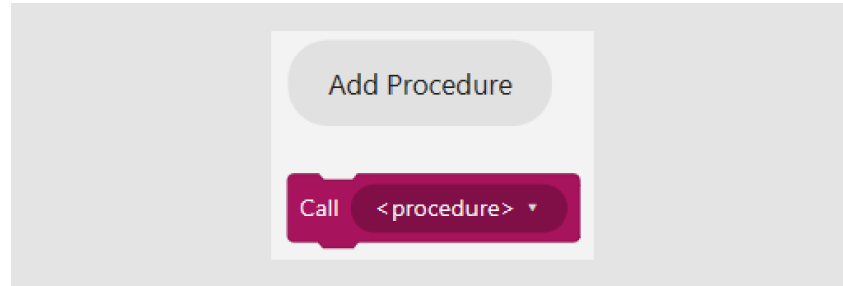
If the user chooses the answer Yes, that is, function key 1 is pressed within 5 seconds, the value of number-type variable Number1 is 1. In this case, the program defined in procedure Program_2 executes.

If the user chooses the answer No, that is, function key 2 is pressed within 5 seconds, the value of number-type variable Number1 is 2. In this case, the program defined in procedure Program_3 executes.

If none of previous situation happens, the program defined in procedure Program_4 executes.

3.10 Programming procedures in subworkspace

Block illustration



xx2200001346

Descriptions

Working with procedures in subworkspace

The **Add Procedure** button in the **Procedures** block group is used to open a subworkspace and add a new procedure. You can edit the procedures in the subworkspace as you do in the main workspace. This facilitates the programming of multiple blocks that need to be reused for several times.

After tapping **Add Procedure**, a dialog box is displayed for you to define the procedure name and the module to which the procedure belongs. You can also add a new module to the program by clicking **New module**.

The default procedure name is *ModuleName_ProcX*. You can rename it either during adding or from the block header in the subworkspace.



Note

The renaming can be automatically updated for the procedures invoked in the main procedure. If the procedures are invoked by procedures other than main, you need to modify the invocations with the new naming; otherwise, errors in RAPID program will raise.

When the programming is done in the subworkspace, tap the file menu button (titled with the procedure name with its belonged module name) at the upper left to go back to the main workspace. If the added procedure belongs to the same module, a new **Call <ModuleName_ProcX>** is displayed in the Procedure block category. If the added procedure belongs to other modules, you need to use the **Call <Procedure>** block and choose the procedure from the procedure list.

Calling procedure

The **call <procedure>** block in the **Procedures** block group is used to invoke a defined procedure from current module or other modules in the active program in the controller.

Continues on next page

3 Programming with Wizard

3.10 Programming procedures in subworkspace

Continued

User cases

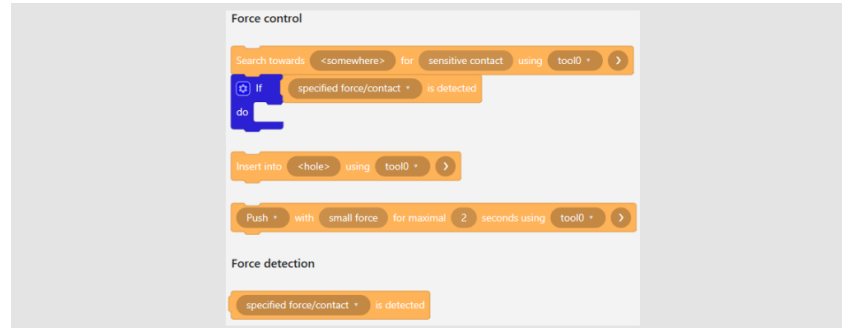
Moving exceeding blocks to procedures

A maximum of 180 blocks can be added to a procedure. If exceeding, it is recommended to use **Add Procedure** and **Call <procedure>** for optimization.

- 1 Tap and hold the block which will be the first block of the new procedure to be added.
- 2 Tap **Cut following blocks** from the shortcut menu.
- 3 Tap **Procedures** category and then tap **Add Procedure**.
- 4 In the displayed dialog box, define the procedure name (e.g, Wizard_Proc1 in this case example) and the module to which the procedure belongs.
- 5 Tap **Add** to display the subworkspace for the new procedure.
- 6 Tap and hold the blank area in the subworkspace and choose **Paste blocks** to paste all cut blocks.
- 7 Drag and attach the pasted blocks to the main program tree.
The procedure will be applied to the controller automatically.
- 8 Tap the file menu button at the upper left to close the subworkspace and go back to the main workspace.
- 9 Tap **Procedures** category and then tap **Call <Wizard_Proc1>**.
The procedure block is added in the main workspace.

3.11 Searching and operating with force control

Block illustration



xx2200001347

Descriptions



Note

The Force blocks are available only for robots configured with ABB force sensors. Please note that if force control is enabled, any defined force will have an impact on the robot movement speed.

Robots with Force Control are sensitive to contact forces. Use the Force blocks to control the robot to "feel" the surroundings and perform specified actions with force. Available force blocks are listed as follows:

- **Search towards** <somewhere> for <sensitive contact> using <tool0>: find whether a contact can be detected in the searching direction. Used together with a **Logic** block to define the following operations if a specified detection result is met. This block allows to set optional parameters.
- **Insert into** <hole> using <tool0>: insert an object. This block allows to set optional parameters.
- **Push** with *small force* for maximal 2 seconds using <tool0>: push or pull with force. This block allows to set optional parameters.
- **Specified force/contact** is detected: trigger condition used together with the searching for contact block.

For details about the parameters, including optional parameters, in the blocks, see [Parameters on page 43](#).

3 Programming with Wizard

3.12 Customizing comments or RAPID instructions

3.12 Customizing comments or RAPID instructions

Block illustration



xx2300001222

Descriptions

Adding comments

Users can use the **Comment** *Insert the comment* block to add a string comment to the program, which can make the program easier to understand and has no effect on the execution of the program.

Clicking **Insert the comment** in the block will display the **Comment** dialog box, in which it allows to type any text string in the editing area.

Customizing RAPID instructions

With the **Custom code** *Insert the RAPID* block, users can customize RAPID instructions to the program as in the RAPID editor.

It is also possible to customize RAPID instructions as a condition for an if block or a condition loop block. The condition block **Custom code** *Insert the RAPID* cannot be used alone.

In both blocks, clicking **Insert the RAPID** will display the **Script** dialog box, in which it allows to type RAPID codes in the editing area.



Note

Syntax error check is not available in Wizard. Always check your codes carefully before applying; otherwise, a warning is displayed, prompting you to modify the codes using RAPID editor in RobotStudio. Your works in Wizard can only continue after the syntax errors are corrected and applied from RobotStudio.

3.13 Working with ASI

ASI for GoFa™ CRB 15000



Note

Arm-side interface (ASI) is located on axis 5 of GoFa™ CRB 15000, opposite to the tool flange. Blocks added using ASI or in ASI Settings are available only on real controller.

The down button on the ASI is, by default, configured to add a Move block. It allows to change the block type by tapping **Settings > ASI Setting**.

Pressing the button will add a Move block or a block containing only one location parameter, depending on the setting, with the current location of the cobot.

Buttons on the ASI could also be configured to other functions, such as gripper to control the opening and closing of the fingers on the gripper. See the product manual of the manipulator for ASI button configuration details.

ASI for CRB 1810



Note

Arm-side interface (ASI) is located around the tool flange of CRB 1810. Blocks added using ASI or in ASI Settings are available only on real controller.

Insert an instruction

The POINT button on the ASI is, by default, configured with the *Insert an instruction* function. Pressing the button will add a default Move block. It allows to change the block type by tapping **Settings > ASI Setting**.

Switch the inserted instruction type

The CUSTOM button on the ASI is, by default, configured with the *Switch the inserted instruction type* function. This function shall be used together with the *Insert an instruction* function. In the **ASI Setting** window, you can define up to 5 block types by tapping **Add option**. With multiple block types defined, pressing the CUSTOM button will switch the block type from one to another based on the sequence you specified in the **ASI Setting** window. Then, pressing the POINT button will add a block in the selected type.

You can also switch block types from Wizard. The switch button at the top of the Wizard window presents a quick view of the active block type. Tapping the button shows the list of blocks that defined in **ASI Setting**. You can then tap the required one to switch the block type.

Add a robtarget

Buttons on the ASI could also be configured to other functions. If a button is configured with the *Add a robtarget* function, pressing the button will add the current robot position as robot target point data to the robot task and module that are specified in **ASI Setting > Add a position > Save path**.

See the product manual of the manipulator for ASI button configuration details.

3.14 Working with lead-through device

Description



Note

The lead-through device is installed on SWIFTI™ robots. Blocks added using the buttons on the device and lead-through device settings are available only on real controller.

The flat button on the lead-through device is, by default, configured to add a Move block. It allows changing the block type by tapping **Settings > Lead-through Device Setting**.

Pressing the button will add a Move block or a block containing only one location parameter, depending on the setting, with the current location of the cobot.

Buttons on the lead-through device could also be configured to other function, such as locking the movement along a specified direction. See the product manual of the manipulator for lead-through device button configuration details.

3.15 Using Favorites for your blocks

How to?

Adding blocks to **Favorites** will help a lot for quick access to frequently used blocks. Follow the steps below to add favorite blocks:

- Tap the category to which the required block belongs.
- Tap **Edit** at the upper left of the category pane.
- Tap the star next to the block to add it to **Favorites**.
You can tap the star again to remove the block from **Favorites**.
- Tap **Save** at the top right.
A new **Favorites** block category will appear on the left pane.



Tip

Adding blocks to **Favorites** only creates a shortcut and will not remove the blocks from their original category. Removing a block from **Favorites** doesn't delete it, it only deletes the shortcut.

This page is intentionally left blank

4 Defining parameters and data

4.1 Parameters

Predefined parameters

The following table lists the parameters whose values are predefined. Users can only choose a value from the parameter list during programming.

Parameter	In blocks...	Value
Motion speed	Move	The robot movement speed can be set to very quickly, quickly, moderately, slowly and very slowly , which corresponding to the movement speed v1000, v500, v200, v50 and v20 respectively.
Zone data	Move	It can be set to Fine indicating that the position is to be terminated in the form of a stop point, or to a predefined value indicating that the position is to be terminated in the form of a fly-by point.
Suction	Air	The gripper can be configured with one or two vacuum modules. Vacuum2 is available only for the gripper with two vacuum modules. Vacuum1 and Vacuum2 are defined as follows.  xx1900002120
Tool name	Gripper	All the configured tools are listed by tooldata name.
Signal name	Signals	All the configured I/O signals in the controller are listed.
Signal value	Signals	The values can be set to 1-True or 0-False for signal blocks.
Error name	Logic	Common errors are listed.

Continues on next page

4 Defining parameters and data

4.1 Parameters

Continued

Parameter	In blocks...	Value
Contact type	Force	Type of the contact to be searched for can be set. • Sensitive(5N) is recommended for soft contact, such as foam board and sponge. The contact force is 5N. • Hard(10N) is recommended for hard contact, such as metal. The contact force is 10N. • For other contact types, user can customize a contact force that is suitable to be applied to the contact.
Push/pull	Force	It allows to set the robot behavior to push or pull an object.
Pushing/pulling force	Force	The force that the robot uses to push or pull can be set to Small(10N) , Medium(14N) or a customized force.
Pushing/pulling duration	Force	It allows to set the duration that the robots pushes or pulls the object. The unit is seconds. The default value is 2s.

Customized parameters

The following table lists the parameters whose values can be customized.

Parameter	In blocks...	Value
somewhere	Move Finger Air Gripper Force	Defines a destination location. By default, the current robtargt value is filled in. Users can also choose an existing location or define a new location. • If specified in Move blocks, it refers to the location where the gripper fingers or suction cups moves to. • If specified in Finger blocks, it refers to the location where the gripper fingers moves to. • If specified in Air blocks, it refers to the location where the gripper suction cups moves to. • If specified in Gripper blocks, it refers to the location where the gripper moves to. • If specified in Force blocks, it refers to the location towards which the robot will search along a defined direction (Tool Z direction by default).  Note If the block is added using the IO binding function, a new location is defined and filled in. For details about how to define a location, see Defining a location on page 47.

Continues on next page

Parameter	In blocks...	Value
object	Finger Air Gripper	Defines the object that requires the gripper to pick up or place using fingers, suction cups or specified tools. The object defined in a pair of picking and placing blocks must be the same. It also defines the object to be released in Release <object> block in Gripper category. For details about how to define an object, see Defining an object on page 50 .
tray	Finger Air	Defines the tray where the object to be picked up from or to be placed to. Users can specify the tray orientation and dimensionality. For details about how to define a tray, see Defining a tray on page 53 .
number/bool/string	Variables	Defines a variable name or a variable value. It is also possible to assign a variable value to a specific variable using the Set <variable name> block. For details about how to define a variable, see Defining a variable on page 54 .
hole	Force	Defines the hole into which the object is insert. Users can choose an existing hole defined before or define a new hole. For details about how to define a hole, see Defining a hole on page 51 .

Optional parameters

Optional parameters are available in some blocks for user to extend settings. Tapping the > button after the block displays the available optional parameters. To modify the parameter value, you can tap the parameter value box, turn on the switch to enable the parameter (if it is disabled), and select the required data from the available list.

Parameter	Definition
\MaxTime	Defines the maximum amount of time in seconds that program execution waits. If no function key is pressed within this time then the program continues to execute in the error handler.
\DIBreak	Defines the digital signal that may interrupt the operator dialog. If no function key is pressed when the defined signal is set to 1 (or is already 1) then the program continues to execute in the error handler.
\WorkObject	Defines the work object (coordinate system) to which the specified location in the block is related.
\Load	Defines the load attached to the robot's mounting flange.

Continues on next page

4 Defining parameters and data

4.1 Parameters

Continued

Parameter	Definition
\Speed	Defines the movement speed. Can be set to very quickly, quickly, moderately, slowly and very slowly, which corresponding to the movement speed v1000, v500, v200, v50 and v20 respectively by default.  Note The movement speed to which each value corresponds can be modified by clicking Customize Speed. The modified speeds will be applied to all the blocks with a speed parameter of all the Wizard programs. The blocks customized in Skill Creator are not affected.
Direction	Defines the direction along which the tool moves. If no direction is defined, the tools moves along the Z- direction of TCP coordination system.
InsertDuration	Define the inserting duration from the insertion starting position to the desired inserted position when the robot inserts an object into a hole. The default duration is 10s. When force control is enabled, the actual robot movement speed depends on all of the defined force, moving distance and specified moving duration. If the duration is set to an extremely small value, the robot stops when the duration elapses, no matter it reaches the desired final position or not. If the duration is set to an extremely large value, the robot may reach the desired final position earlier than the defined duration to ensure the defined force could be achieved. In this case, always take the force and distance into consideration when setting the duration.
SearchDuration	Defines the searching duration from the starting position to the position of the object to be pushed/pulled when the robot pushes or pulls an object. When force control is enabled, the actual robot movement speed depends on all of the defined force, moving distance and specified moving duration. If the duration is set to an extremely small value, the robot stops when the duration elapses, no matter it reaches the desired final position or not. If the duration is set to an extremely large value, the robot may reach the desired final position earlier than the defined duration to ensure the defined force could be achieved. In this case, always take the force and distance into consideration when setting the duration.

4.2 Data definition

4.2.1 Defining a location

Adding a location

- 1 Tap the location parameter **<somewhere>** in a block and then tap **New Location** in the displayed **Location** tab page, or tap **Data > Location > New Location**.
- 2 In the displayed **Add new location** window, type the location name and update the location data.



Tip

If the location is added from a block, the selected tool in the block will be used to read the robtarget value for the location.

If the location is added from the **Data** panel, the active tool will be used to read the robtarget value for the location.

- 3 Tap **Save**.
The added locations are listed in the **Location** tab page.



Note

Added data will be saved only in the current module, and can only be edited or deleted from the module where it is added. But the data could be used in other modules.

Deleting a location

To delete a specific location,

- 1 Tap **Data > Location**.
- 2 In the displayed **Location** tab page, tap **More options** button next to the location you want to delete.
- 3 Tap **Delete**.
- 4 Tap **Delete** in the confirmation dialog box.
The location is now removed from the list.

To delete multiple locations,

- 1 Tap **Data > Location**.
- 2 In the displayed **Location** tab page, click **Select**.
- 3 In the displayed list, select the check boxes of all locations you want to delete.
- 4 Tap **Delete**.
- 5 Tap **Delete** in the confirmation dialog box.
The locations are now removed from the list.

Continues on next page

4 Defining parameters and data

4.2.1 Defining a location

Continued

Modifying a location

- 1 Tap the location parameter in a block, or tap **Data > Location**.
- 2 In the displayed **Location** tab page, tap **More options** button next to the location you want to modify.
- 3 In the displayed list,
 - to modify name,
 - a tap **Rename**.
 - b edit the new name in the **Location name** box in the displayed **Edit location** window.
 - c tap **Save**.
 - to modify location coordinates,
 - a tap **Edit location**.
 - b move the robot and tap **Update location** in the displayed **Edit location** window.



Tip

If the location is modified from a block, the selected tool in the block will be used to read the robtarget value for the location.

If the location is modified from the **Data** panel, the active tool will be used to read the robtarget value for the location.

- c tap **Save**.

Checking the location



Note

This function is valid only in manual mode and when the motor is on.

- 1 Tap **Data > Location**.
- 2 In the displayed **Location** tab page, tap **More options** button next to the location you want to check.
- 3 Tap **Go to**.
- 4 Choose the tool in use from the list.
- 5 Tap and hold **Press to Go to**.



Tip

You can also check the location by tapping **Edit location** from the **More options** list for the required location and then tapping **Go to** in the displayed **Edit location** window.

Continues on next page

About home location

A home position **wi_homePosition** is predefined for use. This position cannot be deleted but is allowed to be modified.

4 Defining parameters and data

4.2.2 Defining an object

4.2.2 Defining an object

Adding an object

- 1 Click **<object>** in a block and then click **New Object** in the displayed **Object** tab page, or tap **Data > Object > New Object**.
- 2 In the displayed **Add new object** wizard, follow the instructions to add a new object.

The added objects are listed in the **Object** tab page.



Note

Added data will be saved only in the current module, and can only be edited or deleted from the module where it is added. But the data could be used in other modules.

Deleting an object

- 1 Tap **Data > Object**.
- 2 In the displayed **Object** tab page, tap **More options** button next to the object you want to delete.
- 3 Tap **Delete**.
- 4 Tap **Delete** in the confirmation dialog box.

The object is now removed from the list.

Modifying an object

- 1 Tap **Data > Object**.
- 2 In the displayed **Objects** tab page, tap **More options** button next to the object you want to modify.
- 3 Tap **Edit**.
- 4 In the displayed **Edit object** window, modify the required data:
 - object name
 - tool used for picking
 - grasping way of the fingers
 - finger position and holding force
 - air pressure of the suction cup

4.2.3 Defining a hole

Adding a hole

- 1 Click **<hole>** in a block and then click **New Hole** in the displayed **Hole** tab page, or tap **Data > Hole > New Hole**.
- 2 In the displayed **Add new hole** wizard,
 - a Jog the robot to the desired inserted position where the sensor pin is properly inserted into the hole, and click **Update position**.
 - b Click **Search starting position** to activate it.
 - c Jog the robot a position from where the search starts, and click **Update position**.
 - d If required, click **Lock tool direction** to open the **Lead-through** window (for collaborative robots) or **Joystick Jog** window (for industrial robots) in the **Jog** app and lock required direction.
 - e Enter a name for the hole in the **Hole name** text box, or keep the default provided name.
 - f Set the force used for searching the hole.
 - g Click **Test inserting** to verify the settings.
 - h If required, click **Advanced** to set the searching diameter.
The tool searches towards the hole in a spiral pattern. The searching diameter defines the circle that forms the spiral.
 - i Click **Save**.

The added holes are listed in the **Hole** tab page.



Note

Added data will be saved only in the current module, and can only be edited or deleted from the module where it is added. But the data could be used in other modules.

Deleting a hole

To delete a specific hole,

- 1 Tap **Data > Hole**.
- 2 In the displayed **Hole** tab page, tap **More options** button next to the hole you want to delete.
- 3 Tap **Delete**.
- 4 Tap **Delete** in the confirmation dialog box.
The hole is now removed from the list.

To delete multiple holes,

- 1 Tap **Data > Hole**.
- 2 In the displayed **Hole** tab page, click **Select**.
- 3 In the displayed list, select the check boxes of all holes you want to delete.
- 4 Tap **Delete**.

Continues on next page

4 Defining parameters and data

4.2.3 Defining a hole

Continued

- 5 Tap **Delete** in the confirmation dialog box.

The holes are now removed from the list.

Modifying a hole

- 1 Tap **Data > Hole**.
- 2 In the displayed **Hole** tab page, tap **More options** button next to the hole you want to modify.
- 3 Tap **Edit**.
- 4 In the displayed **Edit hole** window, modify the required data:
 - inserted position and search starting position
 - hole name
 - searching force
 - searching diameter

4.2.4 Defining a tray

Adding a tray

- 1 Choose **New Tray** from the <tray> drop-down list in a block, or tap **Data > Tray > New Tray**.
- 2 In the displayed **Add new tray** wizard, follow the instructions to add a new tray.

The added trays are listed in the **Tray** tab page.



Note

Added data will be saved only in the current module, and can only be edited or deleted from the module where it is added. But the data could be used in other modules.

Deleting a tray

To delete a specific tray,

- 1 Tap **Data > Tray**.
- 2 In the displayed **Tray** tab page, tap **More options** button next to the tray you want to delete.
- 3 Tap **Delete**.
- 4 Tap **Delete** in the confirmation dialog box.

The tray is now removed from the list.

To delete multiple trays,

- 1 Tap **Data > Tray**.
- 2 In the displayed **Tray** tab page, click **Select**.
- 3 In the displayed list, select the check boxes of all trays you want to delete.
- 4 Tap **Delete**.
- 5 Tap **Delete** in the confirmation dialog box.

The trays are now removed from the list.

Modifying a tray

- 1 Tap **Data > Tray**.
- 2 In the displayed **Tray** tab page, tap **More options** button next to the tray you want to modify.
- 3 Tap **Edit**.
- 4 In the displayed **Edit tray** window,
 - to modify the tray name, edit the new name in the text box and click **Save**.



Note

The information of the object arrangement in each tray dimension is shown for reference.

4 Defining parameters and data

4.2.5 Defining a variable

4.2.5 Defining a variable



Note

Following procedures are valid for all number, boolean and string variables. The number variable is used as an example.

Adding a variable

- 1 Click **<number>** in a variable name block or the Set **<number>** to block, or tap **Data > Number > New Number Variable**.
- 2 In the displayed **Add new number** window, edit the name in the **Variable name** text box.
- 3 Click **Save**.



Note

Added data will be saved only in the current module, and can only be edited or deleted from the module where it is added. But the data could be used in other modules.

Deleting a variable

To delete a specific variable,

- 1 Tap **Data > Number**.
- 2 In the displayed **Number Variable** tab page, tap **More options** button next to the variable you want to delete.
- 3 Tap **Delete**.
- 4 Tap **Delete** in the confirmation dialog box.
The variable is now removed from the list.

To delete multiple variables,

- 1 Tap **Data > Number**.
- 2 In the displayed **Number Variable** tab page, click **Select**.
- 3 In the displayed list, select the check boxes of all variables you want to delete.
- 4 Tap **Delete**.
- 5 Tap **Delete** in the confirmation dialog box.
The variables are now removed from the list.

Renaming a variable

- 1 Tap **Data > Number**.
- 2 In the displayed **Number Variable** tab page, tap **More options** button next to the variable whose name you want to modify.
- 3 Tap **Rename**.
- 4 In the displayed window, edit the new name in the text box and click **Save**.

Continues on next page

Data scope

The scope of data denotes the area in which the data is visible. The optional local directive of a data declaration classifies data as local (within the module), otherwise it is global. Note that the local directive may only be used at module level, not inside a routine.

For more details about data scope, see *Technical reference manual - RAPID overview (3HAC050947-001)*.

4 Defining parameters and data

4.2.6 Defining a custom data

4.2.6 Defining a custom data

About custom data

A custom data tab page will display in the **Data** panel when,

- A RECORD data type has been defined in RAPID.
- A custom block with a custom parameter in defined RECORD data type has been created in Skill Creator for Wizard and exported to the active controller.

Adding custom data

- 1 Tap the custom parameter in the custom block and then tap **New Custom Data Variable** in the displayed *Custom Data* tab page, or tap **Data > Custom Data > New Custom Data Variable**.
- 2 In the displayed **Create New Custom Data** window, type the custom data name and define the data values according to the actual data type structure.
- 3 Tap **Save**.

The added custom data is listed in the *Custom Data* tab page.



Note

Added data will be saved only in the current module, and can only be edited or deleted from the module where it is added. But the data could be used in other modules.

Deleting custom data

To delete a specific custom data,

- 1 Tap **Data > Custom Data**.
- 2 In the displayed *Custom Data* tab page, tap **More options** button next to the custom data you want to delete.
- 3 Tap **Delete**.
- 4 Tap **Delete** in the confirmation dialog box.

The custom data is now removed from the list.

To delete multiple custom data,

- 1 Tap **Data > Custom Data**.
- 2 In the displayed *Custom Data* tab page, click **Select**.
- 3 In the displayed list, select the check boxes of all the data you want to delete.
- 4 Tap **Delete**.
- 5 Tap **Delete** in the confirmation dialog box.

The data is now removed from the list.

Modifying custom data

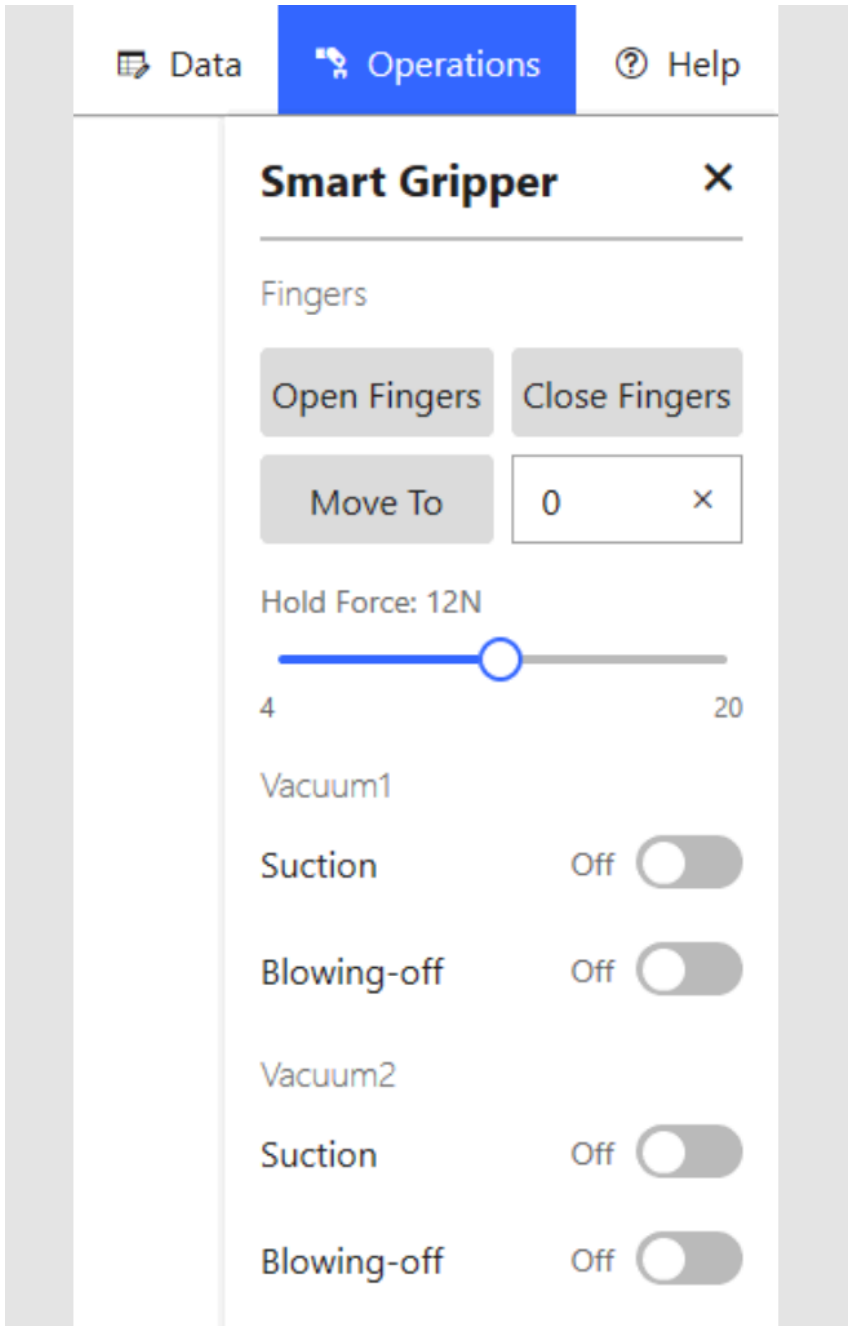
- 1 Tap the custom parameter in the custom block, or tap **Data > Custom Data**.
- 2 In the displayed *Custom Data* tab page, tap **More options** button next to the custom data you want to modify.
- 3 Tap **Edit**.

Continues on next page

- 4 In the displayed **Edit Custom Data** window, modify the data name or other required data values.

4.3 Operations

Smart Gripper



xx2200001354

Fingers



Note

The **Finger** tab page is available only for IRB 14050 having ABB Smart Gripper configured with fingers.

Continues on next page

Finger status is available for set. It can be opened, closed or moved to a specific distance. Specify the finger distance (unit: mm) in the text box and tap **Move to**. When the specified distance is reached, the finger movement stops. You can also tap **Stop** to manually stop its movement.

Hold force can also be set.

Vacuum modules



Note

The **Vacuum** tab page is available only for IRB 14050 having ABB Smart Gripper configured with one or two vacuum modules.

Air supply can be enabled for gripping objects using vacuum modules and disabled to remove suction and release object.

Two vacuum modules **Vacuum1** and **Vacuum2** are defined as follows.



xx1900002120

Gripper



Note

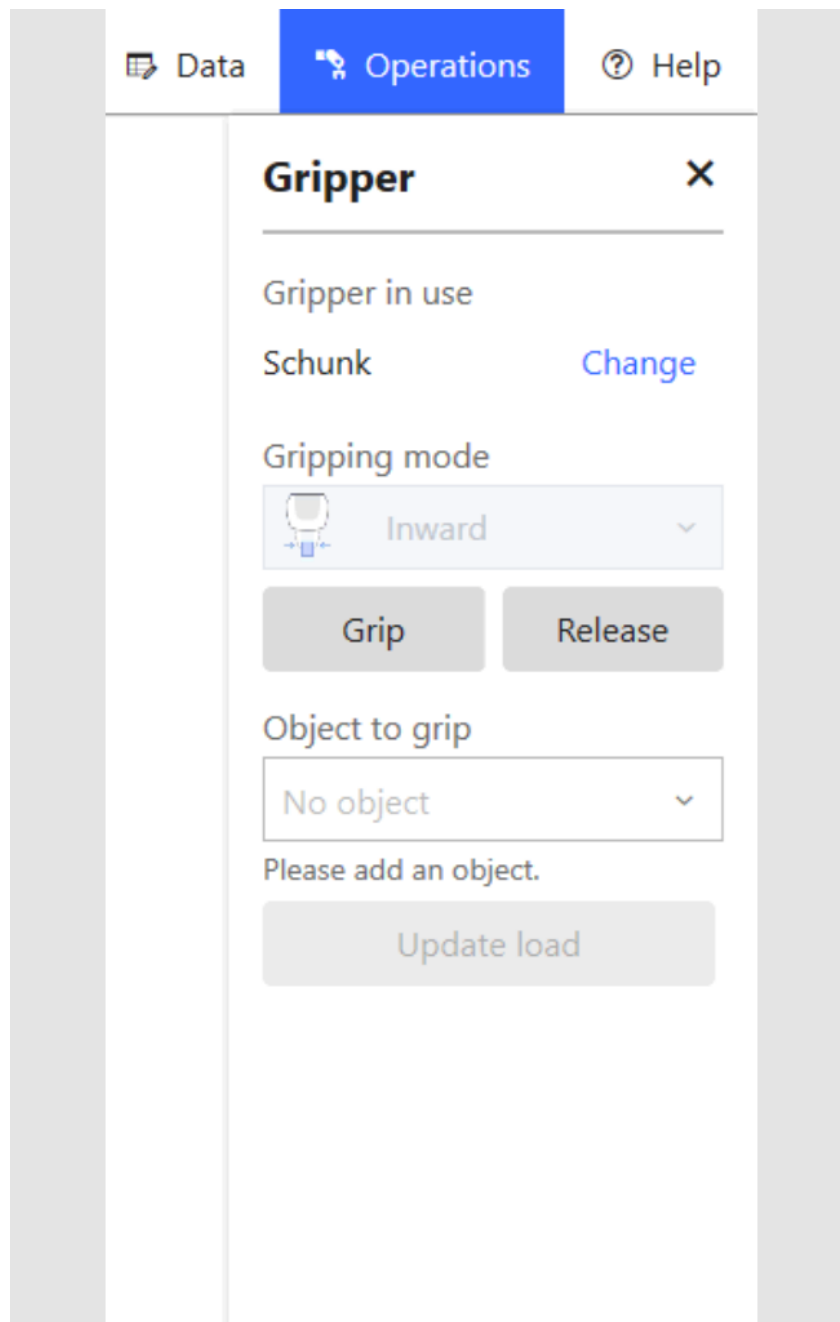
The **Gripper** tab page is available only robots (except IRB 14050) configured with a gripper and its corresponding application add-in.

Continues on next page

4 Defining parameters and data

4.3 Operations

Continued



xx2200001356

If it is detected that the robot is configured with a gripper, a dialog box is displayed when you enter the Wizard program for the first time, asking selecting the gripper in use. You can also change the gripper in the **Gripper** tab page.

The **Grip** or **Release** buttons are available to have quick gripping operations. For grippers with tools such as fingers, gripper modes **Inward** and **Outward** are available to choose; for grippers with tools such as suction cups, gripping mode **Vacuum** is available.

Existing objects are listed for a quick and easy selection for the gripper to grip or release. You could also click **New** and following the instructions in the displayed dialog box to create a new object.

Continues on next page

The object mass is displayed under the object list for a reference. Click **Update load** if it needs an update. For CRB 15000, the mass could be calculated automatically but is also allowed to enter manually. For other robots, you should always enter a mass value.

Other configurable parameters, such as hold force, can also be set if the gripper supports such functions.

Binding blocks with I/O

If a block is defined with a binding to an I/O signal, it can also be added to the workspace when changing the value of the corresponding I/O signal from 0 to 1.

To define the bindings, tab **Help** and then tap **Bind block with I/O**. On the **Add block** tab page of the displayed dialog box, you can add a new binding or remove an existing binding. Note that an I/O signal is allowed to be bound with one block at one time. When adding a new binding, I/O signals that are already bound to blocks will not be available for choosing.

It is also allowed to add a feedback signal for the blocks added from another FlexPendant application or an external device. With the feedback signal defined, after successfully adding blocks from an application or device, you will be noticed by a light, sound or any other signs depending on the signal selected on the **Feedback** tab page of the dialog box and the physical device that the signal assigned to. Note that the **Access Level** property of the feedback signal must be set to **All**; otherwise, modify in the RobotStudio.

4 Defining parameters and data

4.4 Shortcut menu

4.4 Shortcut menu

Shortcut commands

Tapping and holding a block or the blank area of the editing area displays the shortcut command list.

[For common functions on page 62](#)

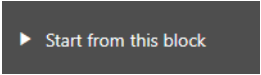
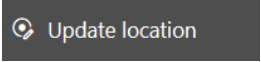
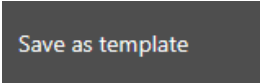
[For single block on page 63](#)

[For single block on page 63](#)

[For Procedure block on page 64](#)

[For workspace on page 64](#)

For common functions

Command	Description
Start from this block 	Sets program pointer to the selected block. You can either tap the Play button or press Play from the FlexPendant to execute the program from this block, or press Prev or Next from the FlexPendant to execute the program step by step from this block.
Update location 	Modifies the location coordinates in the Update location wizard. Valid only for the blocks that have a location parameter defined in the same module.
Save as template 	Creates a block template by grouping 2-15 blocks in a block container. The block you click and all the following blocks attached to it will be grouped in the container. Using block templates eases the future reuse of these block groups. In the dialog box displayed after clicking Save as template , you can define the block category where the block template will be listed and name the block template. It allows to redefine the block category and rename the block template by tapping Edit at the upper left of the corresponding category pane and then tapping the Edit icon next to the template. You can also remove the template by tapping the Delete icon.
Open Application 	Opens a FlexPendant application from Wizard. This command is only available to blocks created using Skill Creator for Wizard, and also displayed as the  icon in the front of the block. <i>Application</i> refers to the application name displayed on the FlexPendant. To define the command for a block, choose the Open a webapp checkbox in the Shortcut settings tab for the block in Skill Creator for Wizard. The application to be opened and related information can be specified in the Webapp name and Additional information fields, respectively. After exporting the block to the controller, the block is available in Wizard with the defined command.
Help 	Displays the help contents for the selected block.

Continues on next page

For single block

Command	Description
Copy block 	Copies the single selected block. Tapping the blank area and choosing Paste pastes the copied block.
Cut block 	Cuts the single selected block. Tapping the blank area and choosing Paste pastes the cut block.
Unplug block 	Separate the single selected block from the program.
Collapse block 	Folds block(s) to display only a concise information of the block(s).
Expand block 	Expands block(s) to show all information.
Disable block 	Disables the block(s) so that they will not be functional in the program. The disabled blocks will be shown as comments in RAPID and no effect on the execution of the program.
Enable block 	Enable the block(s) that are disabled so that they will be functional in the program. You can also edit comments to change them to instructions in the RAPID editor. After applying and loading from the controller, the instructions will be available as blocks in the Wizard.
Delete block 	Tapping a specific block and choosing Delete deletes only the selected block.
Duplicate 	Creates the same block under the selected one.

For multiple blocks

Command	Description
Copy following blocks 	Copies the selected block and all its following blocks. Tapping the blank area and choosing Paste pastes all the copied blocks.

Continues on next page

4 Defining parameters and data

4.4 Shortcut menu

Continued

Command	Description
Cut following blocks 	Cuts the selected block and all its following blocks. Tapping the blank area and choosing Paste pastes all the cut blocks.
Delete following blocks 	Deletes the selected block and all its following blocks.

For Procedure block

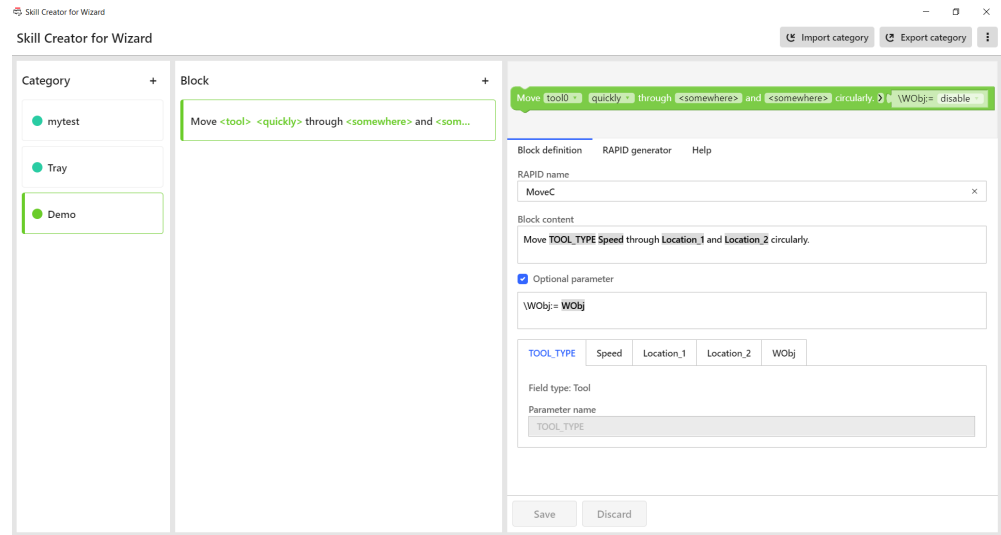
Command	Description
Edit procedure  Edit procedure	Opens the subworkspace to edit the selected procedure. Valid only for the Call <Procedure> blocks.
Delete procedure  Delete procedure	Deletes the selected procedure. The deletion takes effect immediately after you confirm the operation, and is automatically updated for the procedures invoked in the main procedure. If the procedures are invoked by procedures other than main, you need to delete the invocations manually; otherwise, errors in RAPID program will raise.

For workspace

Command	Description
Paste block(s) 	Tapping the blank area and choosing Paste to paste copied or cut block(s).
Delete X Blocks 	Tapping the blank area and choosing Delete deletes all the blocks under programming.

4.5 Skill creator

Create your own blocks



xx2200001358

Skill Creator for Wizard provides an open interface with which you can easily define your own categories, blocks and related parameters. By exporting or importing categories, a group of blocks, to or from the real/virtual controller, you can expand the applications in Wizard.

By default, the exported categories are saved as *CategoryName.coblox* files in the controller directory `$HOME/BlockLibrary/`. If no controller is connected, it is possible to export the categories to a local folder and then load them to Wizard by placing the *.coblox* file from their store folder to directory `$HOME/BlockLibrary/`.

The customized categories will display in Wizard after restarting the FlexPendant. If a category created in Skill Creator already exists as a default category in Wizard, the blocks created in this category will be merged in the Wizard-default category.

How to get?

You can download the Skill Creator for Wizard in [ABB-Developer Center](#).

Explore more

If you are an add-in developer, explore more advanced features in *Release Notes of Wizard*.

This page is intentionally left blank

Index

A

Air block, 24

B

Block

Air, 24, 26

Finger, 24, 26

Force, 37

Gripper, 25, 28

Logic, 31

Loop, 29

Message, 21

Move, 22

Procedure

subworkspace, 35

Signal, 30

F

Finger block, 24

Force block, 37

G

Gripper block, 25

L

Logic block, 31

Loop block, 29

M

Message block, 21

Move block, 22

N

network security, 10

P

Pick and place, 26, 28

Procedure block, 35

S

Signal block, 30



ABB Robotics Sweden AB
S-721 71 VÄSTERÅS, Sweden
Telephone +46 10-732 50 00

ABB Robotics Norway AS
Nordlysvegen 7, N-4340 BRYNE, Norway
Box 265, N-4349 BRYNE, Norway
Telephone: +47 22 87 2000

ABB Robotics (Shanghai) Limited
No. 39, 99 Miaoqiao Road,
Pudong New Area,
Shanghai, 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