ROBOTICS

Application manual

Arc for GoFa

Trace back information:
Workspace 23C version a6
Checked in 2023-09-20
Skribenta version 5.5.019

Application manual

Arc for GoFa

RobotWare 7.12

Document ID: 3HAC084371-001

Revision: B

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.

Table of contents

This page is intentionally left blank

Overview of this manual

About this manual

This manual contains instructions for installing and programming a RobotWare *Arc* system for the GoFa robot.

> **Note**
>
> It is the responsibility of the integrator to conduct a risk assessment of the final application.
>
> It is the responsibility of the integrator to provide safety and user guides for the robot system.

Prerequisites

Installation/maintenance/repair personnel working with an ABB Robot must be trained by ABB and have the knowledge required for mechanical and electrical installation/maintenance/repair work.

> **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.

References

References	Document ID
Operating manual - OmniCore	*3HAC065036-001*
Operating manual - RobotStudio	*3HAC032104-001*
Technical reference manual - RAPID Instructions, Functions and Data types	*3HAC065038-001*
Technical reference manual - RAPID Overview	*3HAC065040-001*
Technical reference manual - System parameters	*3HAC065041-001*
Application manual - Controller software OmniCore	*3HAC066554-001*

Revisions

Revision	Description
A	Published with RobotWare 7.8.
B	Published with RobotWare 7.12. • "Installation Manager" replaced by "Modify Installation".

This page is intentionally left blank

1 Introduction

Introduction

ArcWare for Collaborative Robots is a software package for arc welding with the CRB 15000 (GoFa) robot. It is distributed as a RobotWare add-in and can be downloaded in RobotStudio. The package has limited functionality compared to *Standard Arc*, and the installation is limited to the CRB 15000 robot only.

There are no brand-specific adaptions for power source in the Arc for GoFa package. The *Standard I/O welder interface* is included, which can handle most types of power sources.

The Arc for GoFa package provides a simplified version of Arc for cobot users. This affects the RAPID interface, which is simplified. The *Wizard Easy Programming* interface is used for programming.

The supported RAPID interface is described in section *RAPID reference on page 23*.

Prerequisites

The following software is required.

- RobotWare version 7.8 or higher
- *ArcWare for Collaborative Robots* version 1.0.0 or higher
- *Wizard Easy Programming* version 1.3 or higher
- *ASI HMI* web app 1.0.5 or higher

License 3416-1 *Arcwelding Standard* is required for this add-in.

Limitations

ArcWare for Collaborative Robots can only be used with the collaborative robot CRB 15000.

Overview of products and options

ArcWare for Collaborative Robots includes the following product selections.

- *3416-1 Arc welding Standard* is selected if a valid license is added.
- *Arc for Collaborative robot CRB15000*

  This will be added if any of the welder selections are made.
- *RW Add-in loaded welder*

  This selection makes it possible to install a standalone add-in for welder support.
- *Simulated welder*

  This selection installs *Standard I/O welder* with some pre-configured I/O so that welding can be activated in RobotStudio.
- *Standard I/O welder*

  This selection installs the *Standard I/O welder* support without any I/O signals or process configuration.

Continues on next page

Continued

Predefined welder configuration

The *Standard I/O welder* option does not have any pre-defined configuration, but configuration files for the Fronius TPSi (EtherNet I/P and DeviceNet) are included in the package and copied to \ADDINDATA\ABB.ROBOTICS.APPLICATIONS.ARC\HOME

The following files are loaded into the configuration:

- EIO.cfg
- PROC.cfg
- SYS.cfg

2 Installation

The add-in

The Add-in distribution is named *ArcWare for Collaborative Robots*, and the product in this distribution is named *GoFaArc*.

In the **Modify Installation** tool, the product is added along with the other products that are needed.

A valid license for *3416-1 ArcWelding Standard* is required. On virtual systems in RobotStudio, no licence is required.

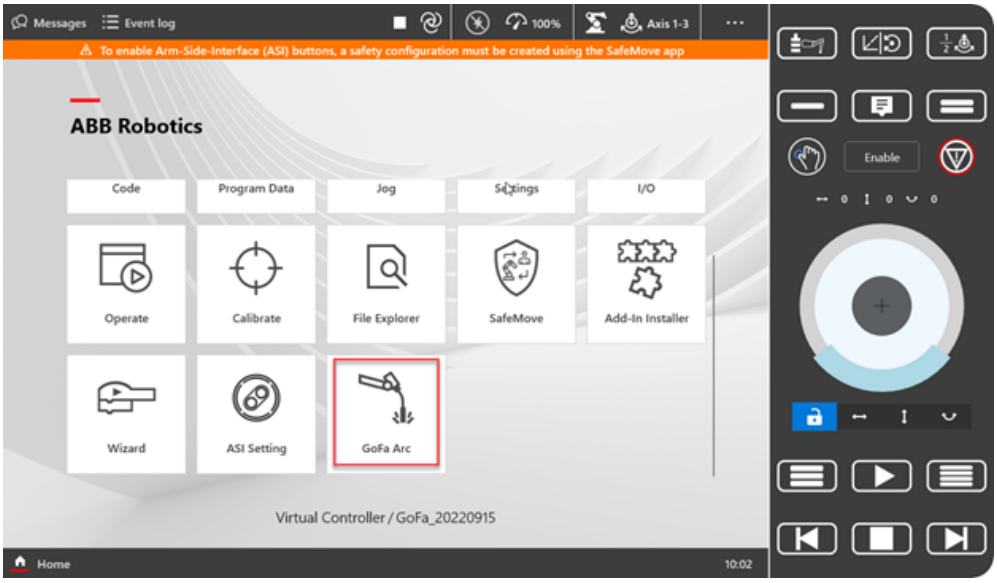
The options described earlier are visible in the tab **Features** in the **Modify Installation** tool.

This page is intentionally left blank

3 FlexPendant application for ArcWare for Collaborative Robots

FlexPendant application for ArcWare for Collaborative Robots

The ArcWare for Collaborative Robots add-in has its own application on the FlexPendant, **GoFa Arc**. It is available on the start page.



xx2200001361

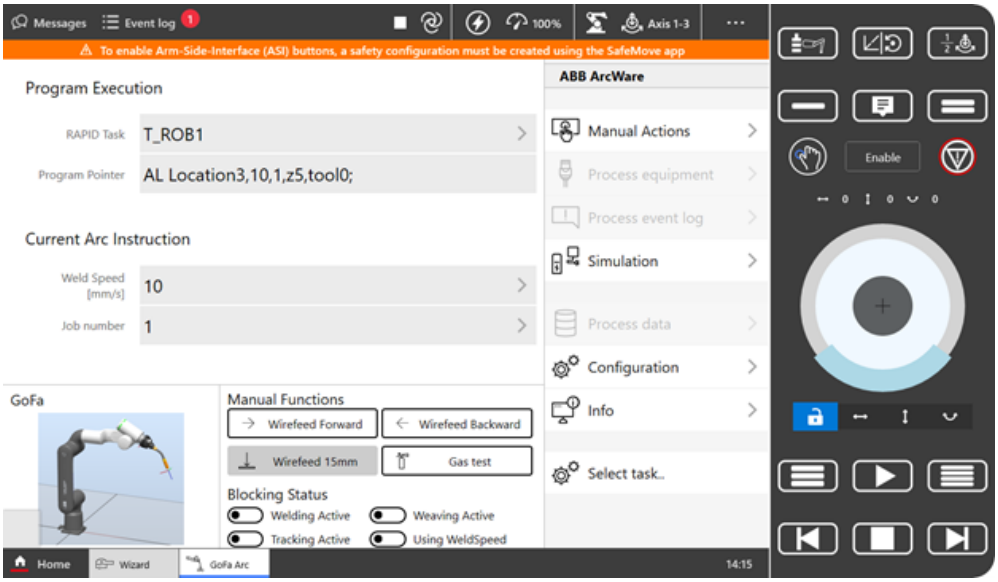
Main page

The main page shows relevant information about program and process related topics. The following information is available:

- Active RAPID task
- Location of program pointer
- Active weld speed in selected units
- Active job number active in the power source
- **Manual Functions**
- **Manual Actions**
- **Configuration**
- **Info**

Continues on next page

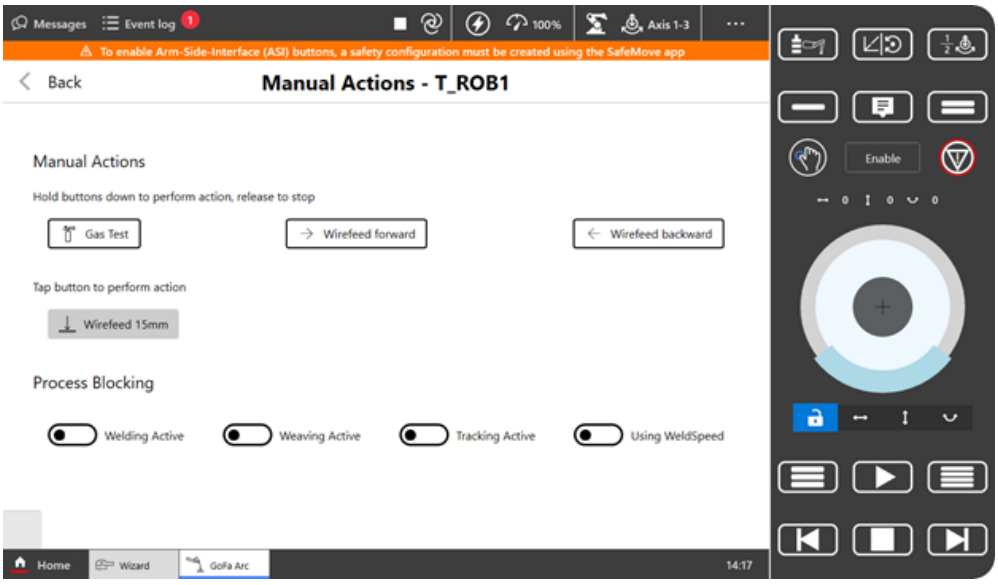
Continued



xx2200001362

Manual Actions

The **Manual Actions** view contains manual actions and process blocking. This functionality is also available on the main page.

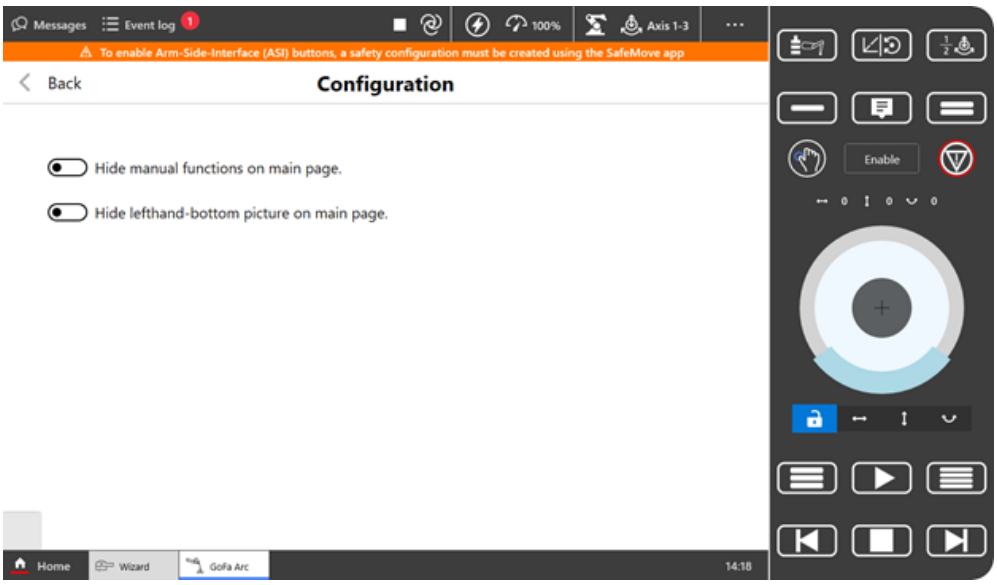


xx2200001363

Continues on next page

Configuration

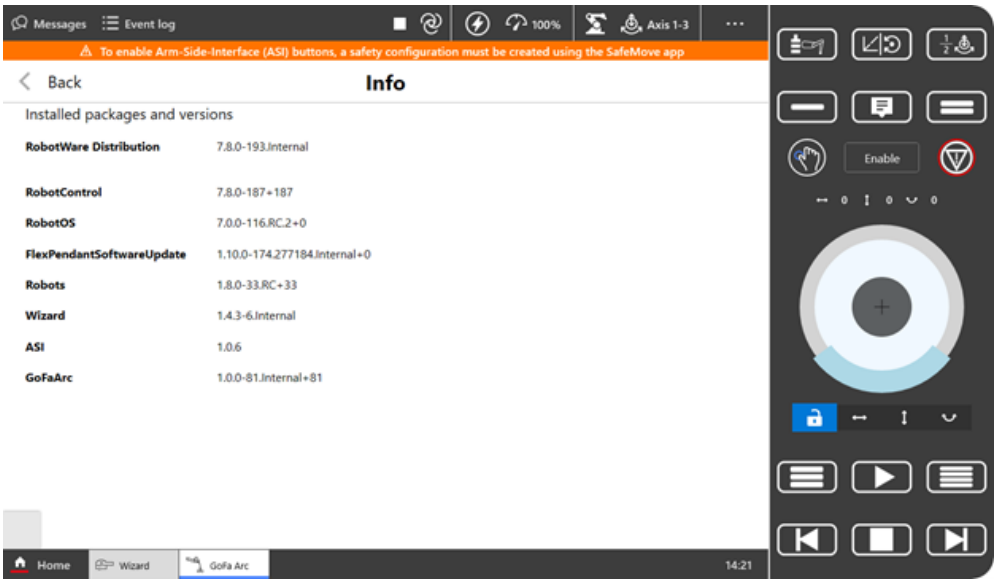
In the **Configuration** view, it is possible to hide manual functions on the main page, and also to hide the graphic shown bottom left on the main page.



xx2200001364

Info

The **Info** view shows the installed packages and versions.



xx2200001365

This page is intentionally left blank

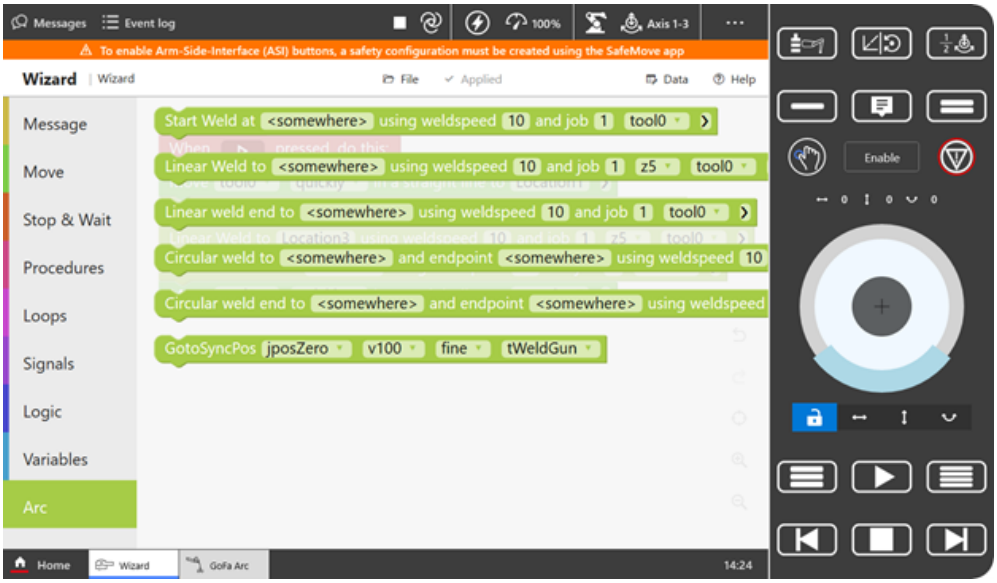
4 Wizard Easy Programming

Introduction

The *Wizard Easy Programming* add-in is a graphical programming tool designed to get users up and running quickly. Simply drag and drop instruction blocks on the FlexPendant to create and modify programs. The Wizard add-in contains an integrated user manual.

The add-in contains instruction blocks for arc welding with CRB 15000. New categories and blocks can easily be created with the *ABB Skill Creator*.

The following blocks related to arc welding is available in the category **Arc**.



xx2200001366

There are two methods available for programming wizard blocks. Drag & drop, or programming via I/O signals.

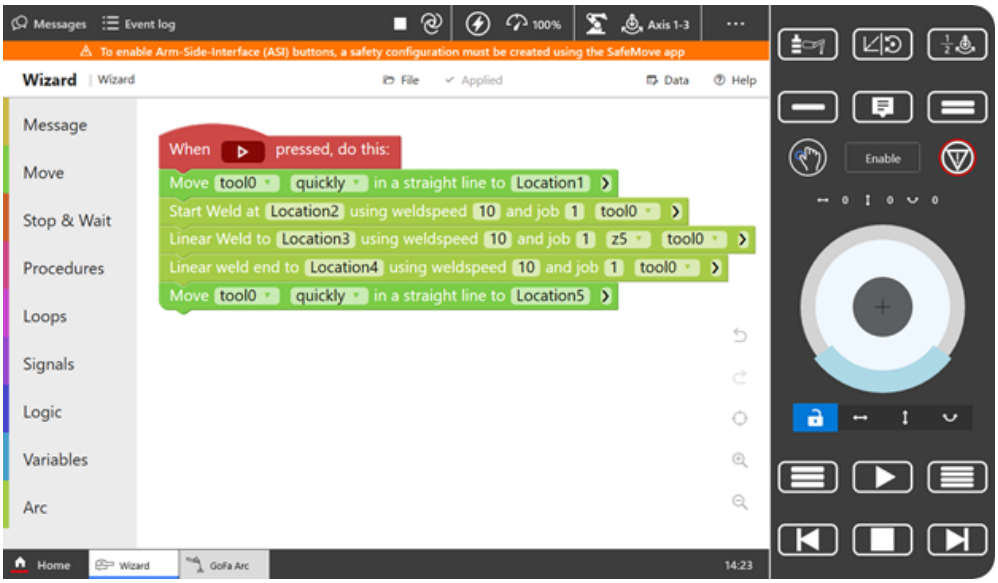
When using the drag & drop method, the position, in this case named *<somewhere>* must be updated manually by adding a new location. If the I/O bind block method is used, the location is automatically added and updated. Weld speed and job can be edited directly in the block.

For more information about the available RAPID objects, see *RAPID reference on page 23*.

Continues on next page

Continued

A finished program can look something like this.



xx2200001367

Optional parameters

There are optional parameters available in the Wizard blocks.

This is indicated with an arrow icon on the block ›

Weave

A weaving pattern can be added to the RAPID instruction by using a pre-defined weave template in the drop-down menu, **w2mm**, **w4mm**, or **w6mm**. The option **disable** will remove the weave pattern from the RAPID instruction. Collapsing the arrow icon will also remove the weave pattern.

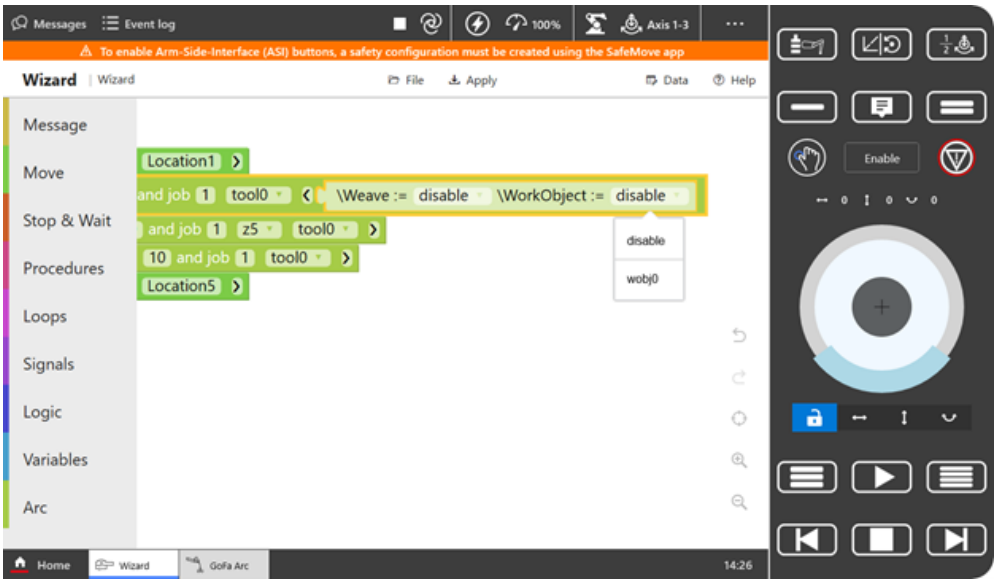
xx2200001428

Continues on next page

Continued

WorkObject

A workobject can be added using the drop-down menu, listing all workobjects that are available in the controller. The option **disable** will remove the workobject from the RAPID instruction. Collapsing the arrow icon will also remove the workobject.

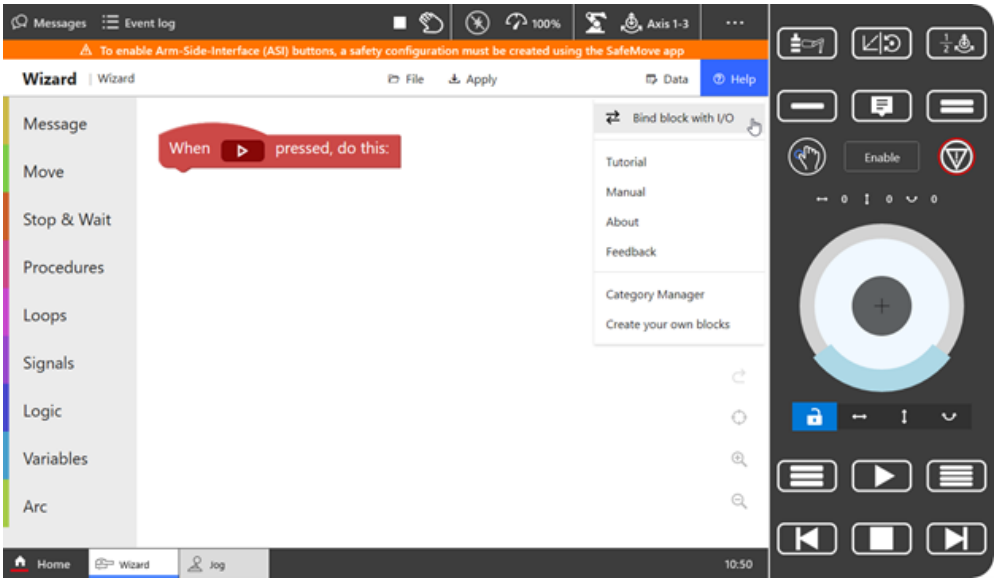


xx2200001429

Bind block with I/O

GoFa Arc comes pre-configured with block bind to I/O signals.

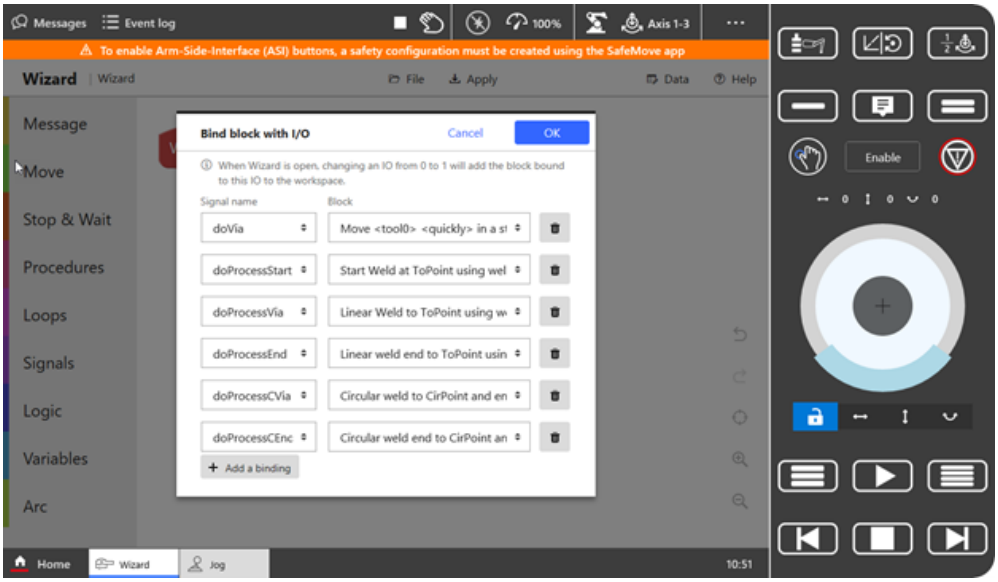
The configuration can be viewed in **Help** and then **Bind block with I/O**.



xx2200001368

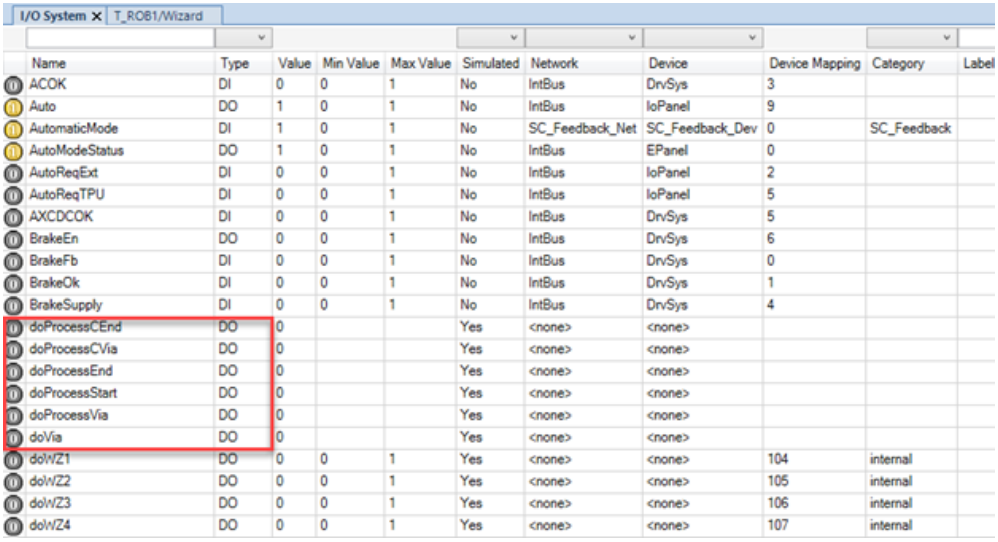
Continues on next page

Continued



xx2200001369

The pre-configured I/O can be viewed in RobotStudio.



I/O System | T_ROB1/Wizard

Name	Type	Value	Min Value	Max Value	Simulated	Network	Device	Device Mapping	Category	Label
ACOK	DI	0	0	1	No	IntBus	DrvSys	3		
Auto	DO	1	0	1	No	IntBus	IoPanel	9		
AutomaticMode	DI	1	0	1	No	SC_Feedback_Net	SC_Feedback_Dev	0	SC_Feedback	
AutoModeStatus	DO	1	0	1	No	IntBus	EPanel	0		
AutoReqExt	DI	0	0	1	No	IntBus	IoPanel	2		
AutoReqTPU	DI	0	0	1	No	IntBus	IoPanel	5		
AXCDCOK	DI	0	0	1	No	IntBus	DrvSys	5		
BrakeEn	DO	0	0	1	No	IntBus	DrvSys	6		
BrakeFb	DI	0	0	1	No	IntBus	DrvSys	0		
BrakeOk	DI	0	0	1	No	IntBus	DrvSys	1		
BrakeSupply	DI	0	0	1	No	IntBus	DrvSys	4		
doProcessCEnd	DO	0			Yes	<none>	<none>			
doProcessCVia	DO	0			Yes	<none>	<none>			
doProcessEnd	DO	0			Yes	<none>	<none>			
doProcessStart	DO	0			Yes	<none>	<none>			
doProcessVia	DO	0			Yes	<none>	<none>			
doVia	DO	0			Yes	<none>	<none>			
doWZ1	DO	0	0	1	Yes	<none>	<none>	104	internal	
doWZ2	DO	0	0	1	Yes	<none>	<none>	105	internal	
doWZ3	DO	0	0	1	Yes	<none>	<none>	106	internal	
doWZ4	DO	0	0	1	Yes	<none>	<none>	107	internal	

xx2200001370

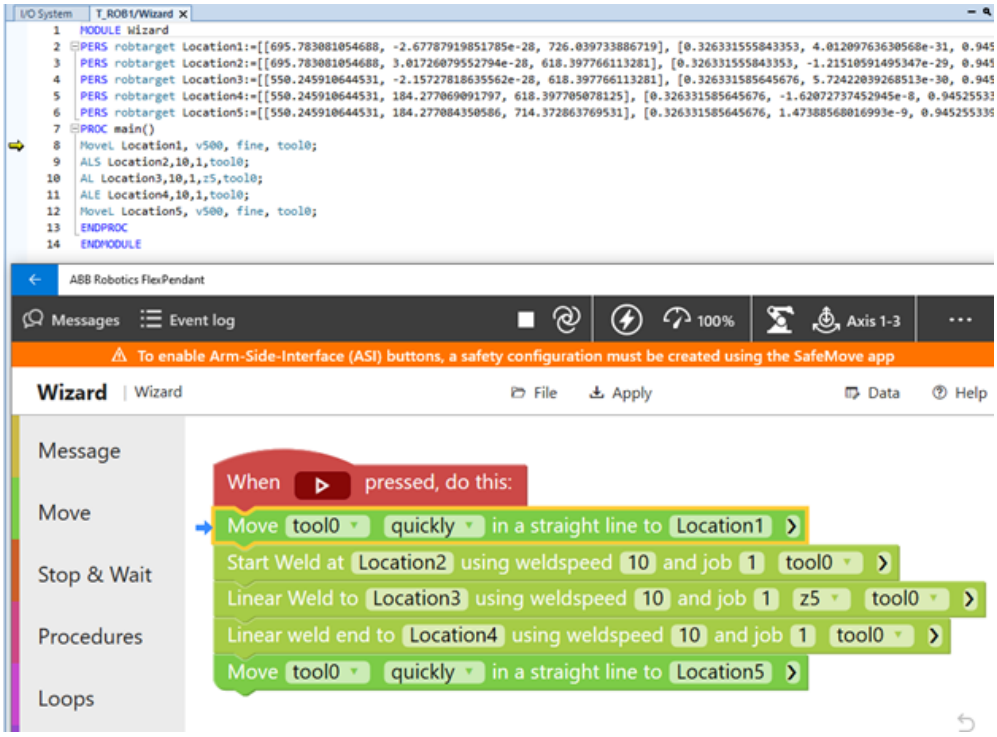
Toggling the I/O signals will create new entries in Wizard. Programming via I/O signals can be used with external devices, for example, setting the I/O signals via buttons for easy programming of welds.

Note

The circular instructions require two toggles of the I/O signal. The first toggle will add the instruction and update the first position. The second toggle will update the second position.

Continues on next page

Continued



xx2200001371

Advanced settings

If the predefined Wizard blocks are not suitable for the application, new Wizard blocks, and categories can be created with the ABB Skill Creator.

This page is intentionally left blank

5 RAPID reference

5.1 Instructions

5.1.1 ALS - Arcwelding Linear Start

Usage

ALS is an instruction used for arc welding with CRB 15000.

Basic examples

The following example illustrates the instruction ALS.

Example 1

```
ALS Location1, 10, 1, tWeldGun;
```

The robot moves linearly to position Location1 and prepares gas preflow and sets the job number in advance.

Default values for gas_preflow is 0.2 s and gas_purge is 0.05 s. The default value can be changed as follows:

- `sm_default.purge_time := 'new value';`
- `sm_default.preflow_time := 'new value';`

Arguments

```
ALS ToPoint, WeldSpeed, Job, Tool [\Weave] [\Wobj]
```

ToPoint

Data type: robtarget

Start position of the weld.

WeldSpeed

Data type: num

Weld speed in mm/s.

Job

Data type: num

Job number sent to the welder.

Tool

Data type: tooldata

The tool used during the movement to ToPoint.

[\Weave]

Data type: weavedata_gofa

Weave data used for the movement.

[\Wobj]

Data type: wobjdata

The work object used during the movement.

Continues on next page

5.1.1 ALS - Arcwelding Linear Start

RobotWare - OS

Continued

Syntax

```
ALS
  [ ToPoint ':=' ] < expression (IN) of robtarget > ','
  [ WeldSpeed ':=' ] < expression (IN) of num >
  [ Job ':=' ] < expression (IN) of num >
  [ Tool ':=' ] < persistent (PERS) of tooldata >
  [ '\' Weave ':=' < expression (IN) of weavedata_gofa > ]
  [ '\' WObj ':=' < persistent (PERS) of wobjdata > ] ';'
```

5.1.2 AL - Arcwelding Linear

Usage

AL is an instruction used for arc welding with CRB 15000.

Basic examples

The following example illustrates the instruction AL.

Example 1

```
AL Location1, 10, 1, z5, tWeldGun;
```

The robot moves linearly to position Location1 with welding process active.

Arguments

```
AL ToPoint, WeldSpeed, Job, Zone, Tool [\Weave] [\Wobj]
```

ToPoint

Data type: robtarget

Start position of the weld.

WeldSpeed

Data type: num

Weld speed in mm/s.

Job

Data type: num

Job number sent to the welder.

Zone

Data type: zonedata

Tool

Data type: tooldata

The tool used during the movement to ToPoint.

[\Weave]

Data type: weavedata_gofa

Weave data used for the movement.

[\Wobj]

Data type: wobjdata

The work object used during the movement.

Syntax

```
AL
  [ ToPoint ':=' ] < expression (IN) of robtarget > ','
  [ WeldSpeed ':=' ] < expression (IN) of num >
  [ Job ':=' ] < expression (IN) of num >
  [ Zone ':=' ] < expression (IN) of zonedata >
  [ Tool ':=' ] < persistent (PERS) of tooldata >
```

Continues on next page

5.1.2 AL - Arcwelding Linear

RobotWare - OS

Continued

```
                    [ '\' Weave ':=' < expression (IN) of weavedata_gofa > ]
                    [ '\' WObj ':=' < persistent (PERS) of wobjdata > ] ';'
```

5.1.3 ALE - Arcwelding Linear End

Usage

ALE is an instruction used for arc welding with CRB 15000.

Basic examples

The following example illustrates the instruction ALE.

Example 1

```
ALE Location1, 10, 1, tWeldGun;
```

The robot moves linearly to position Location1 with welding process active.

When the robot has reached Location1, the welding process is ended.

Arguments

```
ALE ToPoint, WeldSpeed, Job, Tool [\Weave] [\Wobj]
```

ToPoint

Data type: robtarget

Move linearly to ToPoint.

WeldSpeed

Data type: num

Weld speed in mm/s.

Job

Data type: num

Set the configured group output to this value.

Tool

Data type: tooldata

The tool used during the movement to ToPoint.

[\Weave]

Data type: weavedata_gofa

Weave data used for the movement.

[\Wobj]

Data type: wobjdata

The work object used during the search.

Syntax

```
ALE
  [ ToPoint ':=' ] < expression (IN) of robtarget > ','
  [ WeldSpeed ':=' ] < expression (IN) of num >
  [ Job ':=' ] < expression (IN) of num >
  [ Tool ':=' ] < persistent (PERS) of tooldata >
  [ '\' Weave ':=' < expression (IN) of weavedata_gofa > ]
  [ '\' WObj ':=' < persistent (PERS) of wobjdata > ] ';'
```

5.1.4 AC - Arcwelding Circular

Usage

AC is an instruction used for arc welding with CRB 15000.

Basic examples

The following example illustrates the instruction AC.

Example 1

```
AC Location1, Location2, 10, 1, z5, tWeldGun;
```

The robot moves circularly to position Location1 and Location2 with welding process active.

Arguments

```
AC CirPoint, ToPoint, WeldSpeed, Job, Zone, Tool [\Weave] [\Wobj]
```

CirPoint

Data type: robtarget

Circle position of the weld.

ToPoint

Data type: robtarget

ToPoint position of the weld.

WeldSpeed

Data type: num

Weld speed in mm/s.

Job

Data type: num

Job number sent to the welder.

Zone

Data type: zonedata

Tool

Data type: tooldata

The tool used during the movement to ToPoint.

[\Weave]

Data type: weavedata_gofa

Weave data used for the movement.

[\Wobj]

Data type: wobjdata

The work object used during the movement.

Continues on next page

Syntax

```
AC
  [ CirPoint ':=' ] < expression (IN) of robtarget > ','
  [ ToPoint ':=' ] < expression (IN) of robtarget > ','
  [ WeldSpeed ':=' ] < expression (IN) of num >
  [ Job ':=' ] < expression (IN) of num >
  [ Zone ':=' ] < expression (IN) of zonedata >
  [ Tool ':=' ] < persistent (PERS) of tooldata >
  [ '\' Weave ':=' < expression (IN) of weavedata_gofa > ]
  [ '\' WObj ':=' < persistent (PERS) of wobjdata > ] ';'
```

5.1.5 ACE - Arcwelding Circular End

Usage

`ACE` is an instruction used for arc welding with CRB 15000.

Basic examples

The following example illustrates the instruction `ACE`.

Example 1

```
ACE Location1, Location2, 10, 1, tWeldGun;
```

The robot moves circularly to position `Location1` and `Location2` with welding process active. When the robot has reached `Location2`, the welding process is ended.

Arguments

```
ACE CirPoint, ToPoint, WeldSpeed, Job, Tool [\Weave] [\Wobj]
```

`CirPoint`

Data type: `robtarget`

Circle position of the weld.

`ToPoint`

Data type: `robtarget`

`ToPoint` position of the weld.

`WeldSpeed`

Data type: `num`

Weld speed in mm/s.

`Job`

Data type: `num`

Job number sent to the welder.

`Tool`

Data type: `tooldata`

The tool used during the movement to `ToPoint`.

`[\Weave]`

Data type: `weavedata_gofa`

Weave data used for the movement.

`[\Wobj]`

Data type: `wobjdata`

The work object used during the movement.

Syntax

```
ACE
  [ CirPoint ':=' ] < expression (IN) of robtarget > ','
  [ ToPoint ':=' ] < expression (IN) of robtarget > ','
```

Continues on next page

```
[ WeldSpeed ':=' ] < expression (IN) of num >
[ Job ':=' ] < expression (IN) of num >
[ Tool ':=' ] < persistent (PERS) of tooldata >
[ '\' Weave ':=' < expression (IN) of weavedata_gofa > ]
[ '\' WObj ':=' < persistent (PERS) of wobjdata > ] ';'
```

5.2 Data types

5.2.1 weavedata_gofa

Usage

`weavedata_gofa` is used to define any weaving carried out during arc welding. Weaving can be used during the heat and weld phases of a seam.

Weaving is a movement, superimposed on the basic path of the process. That means, the weld speed is kept as defined in `WeldSpeed` data and the TCP speed is increased unless the physical robot limitations are reached. There are four types of weaving patterns, see *Types of weave shape on page 32*.

- zigzag
- V-shaped
- triangular weaving
- circular weaving

All weave data components apply to both the heat phase and the weld phase.

Components

`weave_shape` (weld weave shape)

Data type: `num`

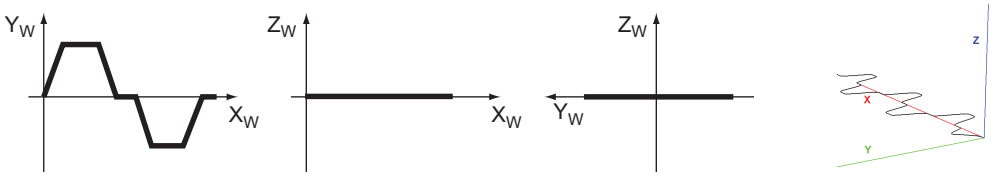
The shape of the weaving pattern in the weld phase as illustrated in the following figures.

> **Note**
>
> The path coordinate system is shown with x-axis in path direction.

Types of weave shape

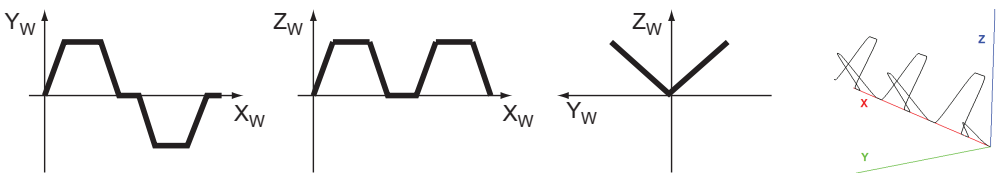
0 - No weaving.

1 - Zigzag weaving results in a weaving horizontal to the seam.



xx1200000714

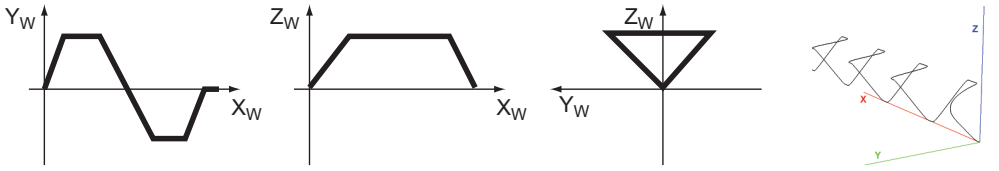
2 - V-shaped weaving results in weaving in the shape of a "V", vertical to the seam.



xx1200000715

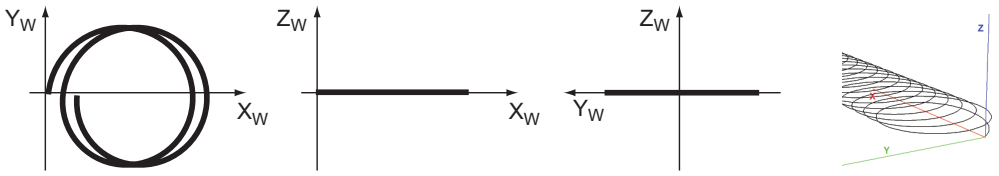
Continues on next page

3 - Triangular weaving results in a triangular shape, vertical to the seam.



xx1200000716

4 - Circular weaving results in a circular shape, vertical to the seam.



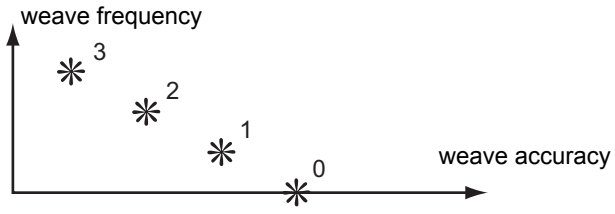
xx1200000717

The type of weaving in the weld phase

`weave_type` (weld weave interpolation type)

Data type: `num`

Specified value	Weaving type
0	Geometric weaving. All axes are used during weaving.
1	Wrist weaving.
2	Rapid weaving. Axes 1, 2, and 3 used.
3	Rapid weaving. Axes 4, 5, and 6 used.



xx1200000718

`weave_length`

Data type: `num`

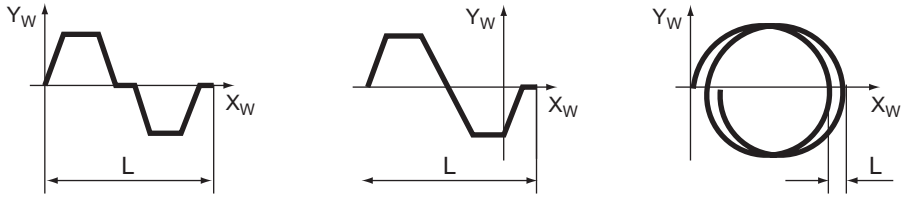
There are two meanings of the `weave_length` component: length and frequency. For length the component `weave_length` is defined as a length of the weaving cycle in the weld phase for weaving types 0 and 1, see the following figure. See the measurement L in the following figure.

For circular weaving the length attribute defines the distance between two successive circles if the `cycle_time` argument is set to 0. If the `cycle_time`

Continues on next page

5.2.1 weavedata_gofa
RobotWare - OS
Continued

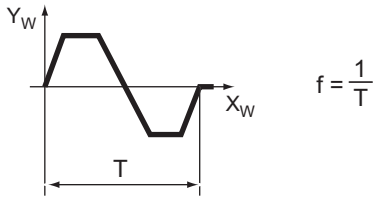
argument has a value then the length attribute can be displaced. The TCP rotates left with a positive length value, and right with a negative length value.



xx1200000719

For frequency the component `weave_length` is defined as the frequency of the weaving cycle in the weld phase for weaving types 2 and 3, see the following figure. For circular weaving the `weave_length` argument defines the number of circles per second.

The TCP rotates left with a positive `cycle_time` value, and right with a negative `cycle_time` value.

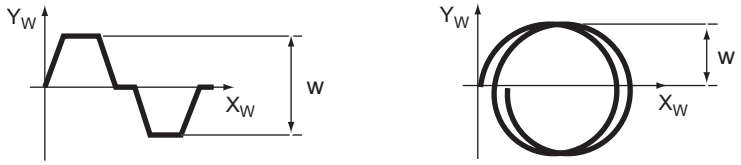


xx1200000720

`weave_width`

Data type: `num`

For circular weaving, width is the radius of the circle. For all other weaving shapes, width is the total amplitude of the weaving pattern. See the measurement W in the following figure.

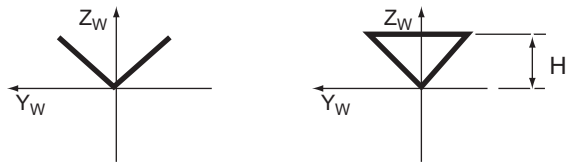


xx1200000721

`weave_height`

Data type: `num`

The height (H) of the weaving pattern during V-shaped and triangular weaving, see the following figure. Not available for circular weaving.



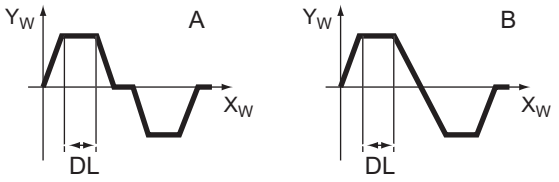
xx1200000722

Continues on next page

`dwell_left`

Data type: `num`

The length of the dwell (DL) used to force the TCP to move only in the direction of the seam at the left turning point of the weave. Not available for circular weaving.



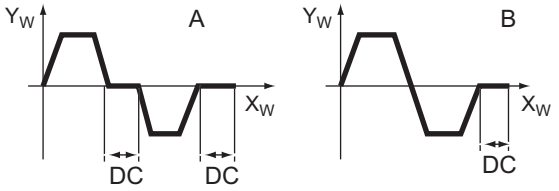
xx1200000723

A	Zigzag and V-shaped weaving
B	Triangular weaving

`dwell_center`

Data type: `num`

The length of the dwell (DC) used to force the TCP to move only in the direction of the seam at the center point of the weave. Not available for circular weaving.



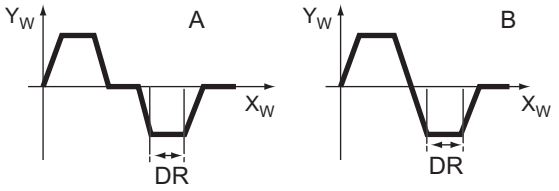
xx1200000724

A	Zigzag and V-shaped weaving
B	Triangular weaving

`dwell_right`

Data type: `num`

The length of the dwell (DR) used to force the TCP to move only in the direction of the seam at the right turning point of the weave. Not available for circular weaving.



xx1200000725

A	Zigzag and V-shaped weaving
B	Triangular weaving

Continues on next page

5.2.1 weavedata_gofa
RobotWare - OS
Continued

`weave_dir` (weave direction angle)

Data type: `num`

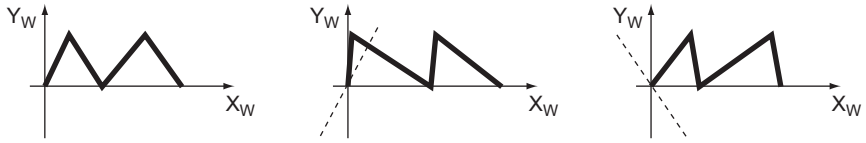
The weave direction angle horizontal to the seam. An angle of zero degrees results in a weave vertical to the seam.

xx1200000726

`weave_tilt` (weave tilt angle)

Data type: `num`

The weave tilt angle, vertical to the seam. An angle of zero degrees results in a weave which is vertical to the seam.

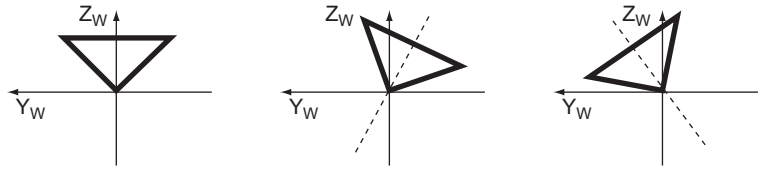


xx1200000727

`weave_ori` (weave orientation angle)

Data type: `num`

The weave orientation angle, horizontal-vertical to the seam. An angle of zero degrees results in symmetrical weaving.



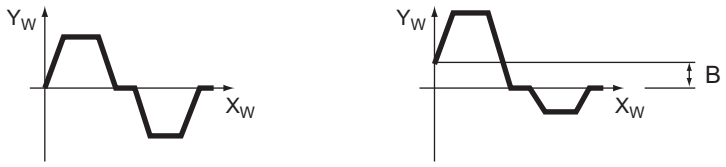
xx1200000728

`weave_bias` (weave center bias)

Data type: `num`

The bias horizontal to the weaving pattern. The bias can only be specified for zig-zag weaving and may not be greater than half the width of the weave. Not available for circular weaving.

The following figure shows zigzag weaving with and without bias (B).



xx1200000729

Continues on next page

`org_weave_width`

Data type: `num`

This component holds the last saved original value for the tuning function. It can be used for a quick restore of a changed value to the original value.

`org_weave_height`

Data type: `num`

This component holds the last saved original value for the tuning function. It can be used for a quick restore of a changed value to the original value.

`org_weave_bias`

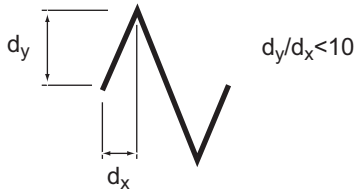
Data type: `num`

This component holds the last saved original value for the tuning function. It can be used for a quick restore of a changed value to the original value.

Limitations

The maximum weaving frequency is 2 Hz.

The inclination of the weaving pattern must not exceed the ratio 1:10 (84 degrees). See the following figure.



xx1200000730

Change of `weave_type` in `weavedata` is not possible in zone points, only in fine points. This is the behavior for both spline & decbuf interpolator. All robots, that use *TrueMove* or *QuickMove* second generation have the following changed behavior for the different weaving types available in RW Arc, compared to *TrueMove* or *QuickMove* first generation:

- Geometric weaving - There is no change.
- Wrist weaving - uses mainly the wrist axes (4, 5, and 6) but small corrections can also be added to the main axes to be able to keep the pattern in the desired plane.
- Rapid weaving - In *TrueMove* or *QuickMove* second generation both geometric weaving and wrist weaving have highly improved performance. Therefore Rapid weaving (both types) is not necessary as a special weaving type any more.

  Rapid weaving axis 1, 2, and 3 is the same as geometric weaving.

  Rapid weaving axis 4, 5, and 6 is the same as wrist weaving.

  The weaving types are still available for backward compatibility.

The system uses *TrueMove* or *QuickMove* second generation, if there is a switch `dyn_ipol_type 1` in MOC.cfg in the MOTION_PLANNER data (system parameters).

Continues on next page

5.2.1 weavedata_gofa

RobotWare - OS

Continued

Structure

```
<data object of weavedata_gofa>
  <weave_shape of num>
  <weave_type of num>
  <weave_length of num>
  <weave_width of num>
  <weave_height of num>
  <dwell_left of num>
  <dwell_center of num>
  <dwell_right of num>
  <weave_dir of num>
  <weave_tilt of num>
  <weave_ori of num>
  <weave_bias of num>
  <org_weave_width of num>
  <org_weave_height of num>
  <org_weave_bias of num>
```

Characteristics

`weavedata_gofa` is an alias data type for `weavedata` and consequently inherits its characteristics.

Index

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

3HAC084371-001, Rev B, en