Deep | Link

V 1.0.6

Weld weave URCap manual

Deep | Link
Published 2023-Janurary, HangZhou

Copyright

The information contained herein is property of Deep-Link Co. Ltd., and shall not be reproduced in its entirety or in part without prior written approval of Deep-Link Co. Ltd. The information herein is subject to change without notice and should not be interpreted as a commitment by Deep-Link Co. Ltd. This manual is periodically reviewed and revised.
Deep-Link Co. Ltd. may take no responsibility for any errors or omissions in this document.

Version:	V 1.0.5 and newer
Date:	2022-June-15
Language:	English and Chinese

*****English Documentation *****

1. Brief Introduction

This URCap provides an add-in software to enable weave function in UR robot, thanks to the **precisely cycle controlled,** it can maintain correct weaving pattern with **constant process TCP velocity**, to further enhance welding quality. It provides currently **circle weaving**, **triangle weaving**, **zigzag weaving**, and **pendulum weaving**. We are working on more types to enrich the proposal capability. The objective of this add-in is to facilitate programmer easily customizes a welding project **re-using maximumly native Polyscope functions** and quickly build weaving features.

The compatibility minimum requirement is:

Universal Robots
e-series: 5.19 or newer.

2. Installation

1st step: click hamburger menu on top right of Polyscope

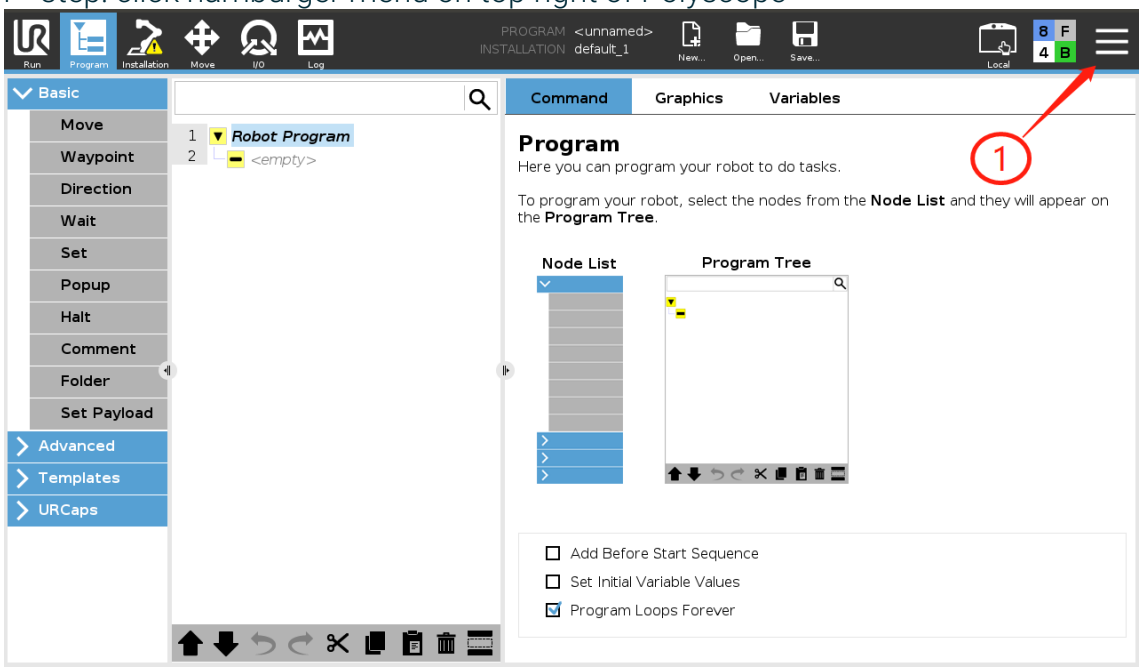


Figure 1 Installation 1st step

2nd step: **Select Settings / System / URCaps**, then click "**+**" button on button area.

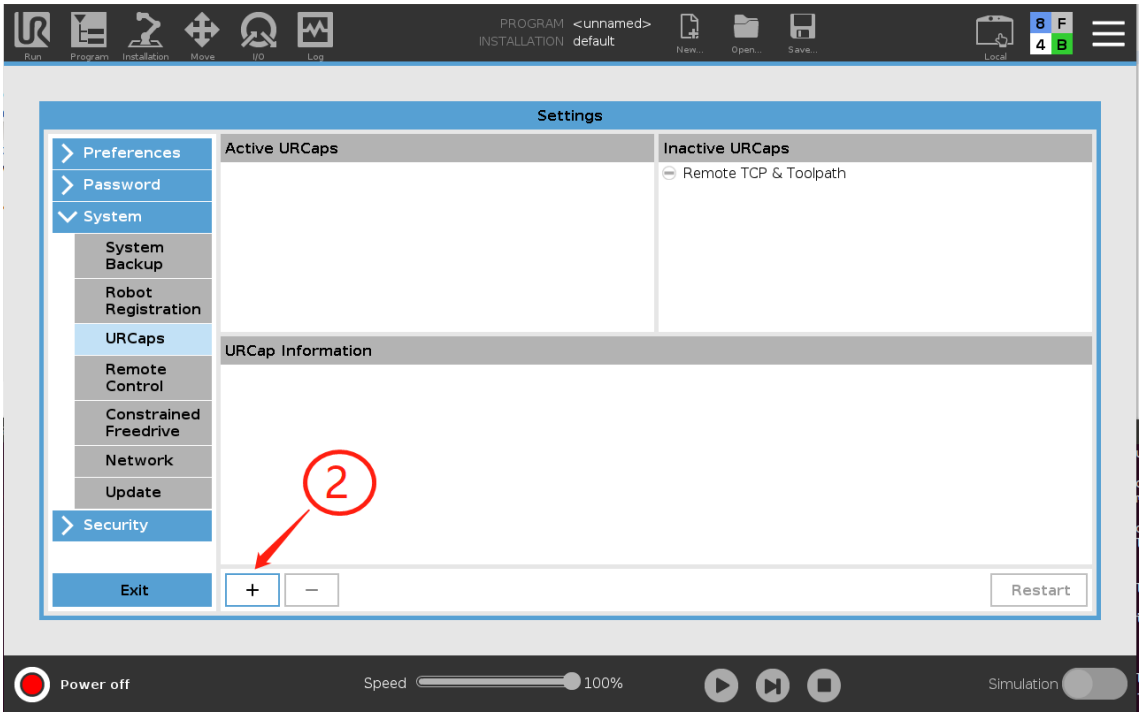


Figure 2 Installation 2nd step

3rd step: select the <welding.Weave-1.0.x .urcap> file in the folder browser and click **Open** button, then click **Restart**. After restart Polyscope, it will display as below figure.

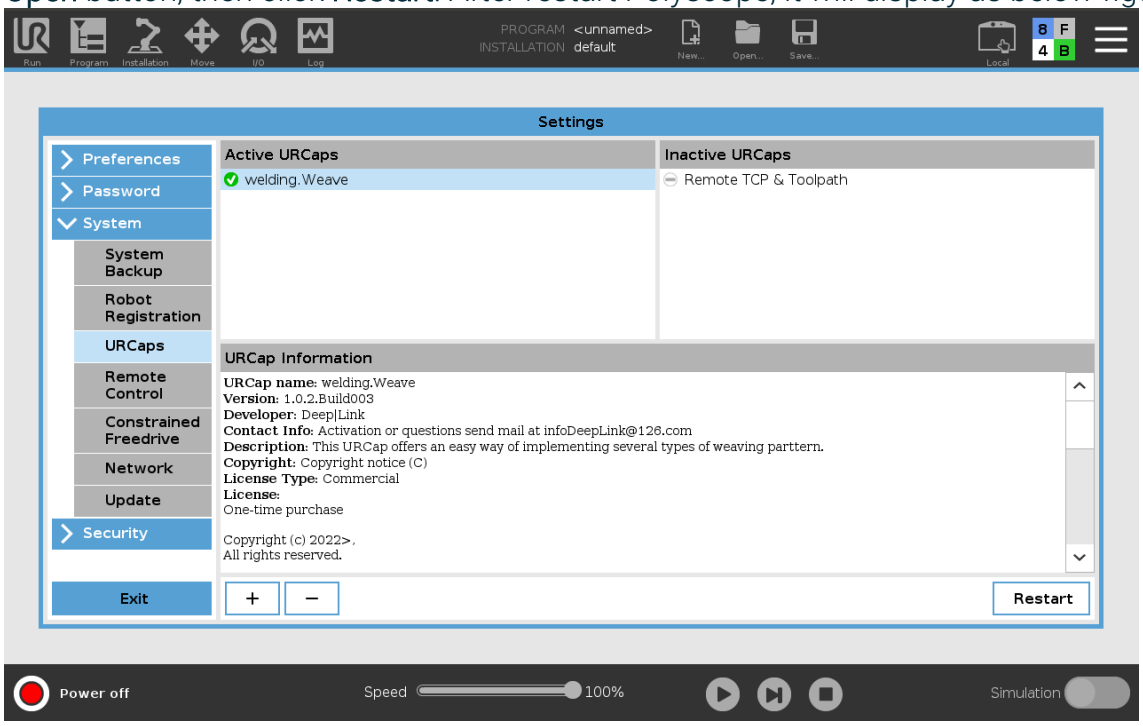


Figure 3 Installation 3rd step

4th step: Click **Installation** and click License Import button to select your activation file in the USB sticker, subsequently click Activate button until you get Figure 5 message box means license activation is completed.

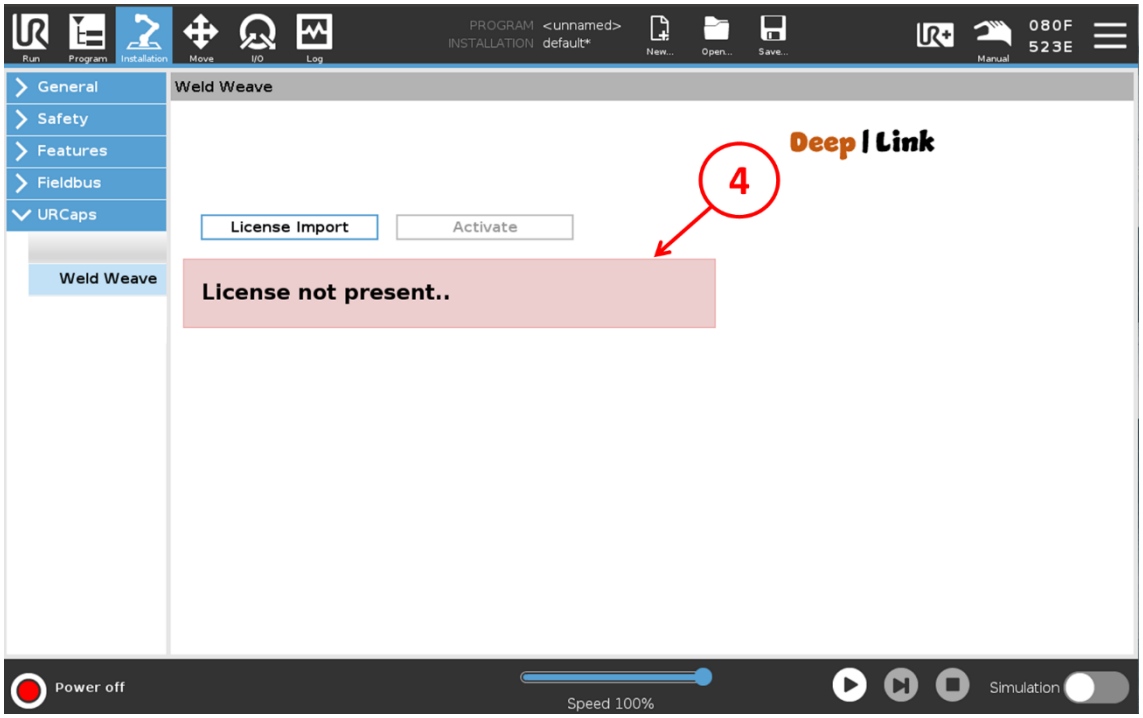


Figure 4 Installation 4th step

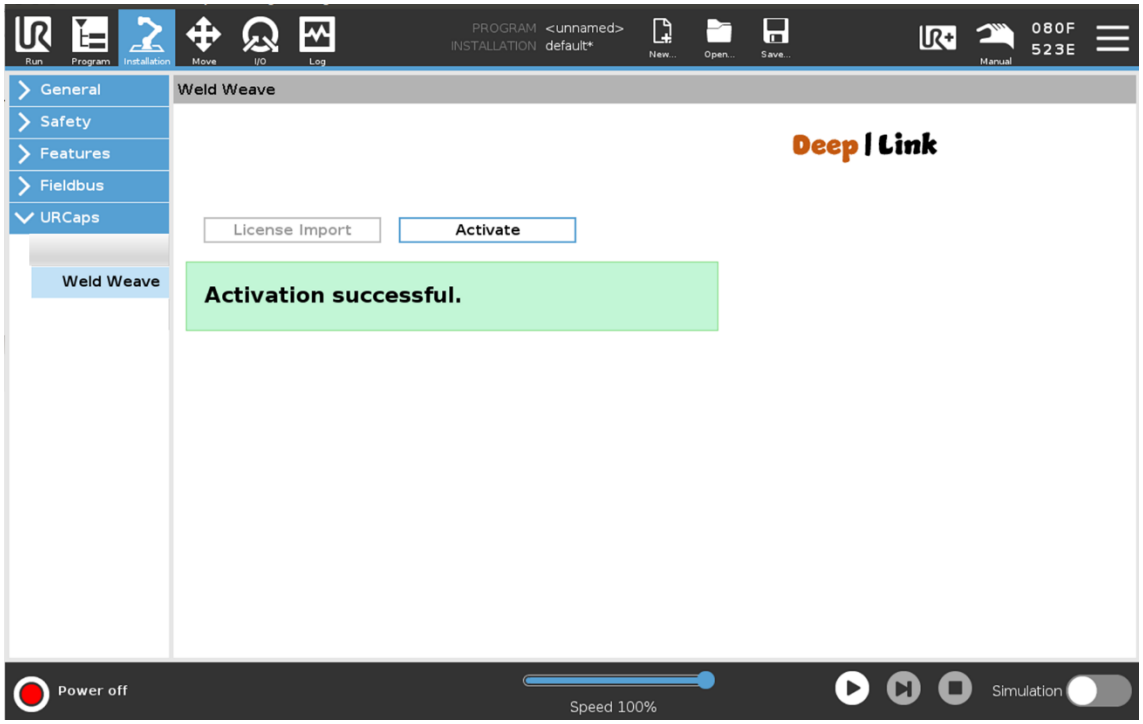


Figure 5 Installation activation view

5th step: Subsequently, the final step is taking care of tool coordinate configuration, please follow UR robot User Manual through online technical resource, and make sure TCP x-axis is pointing same direction with trajectory vector, like below Figure 6.

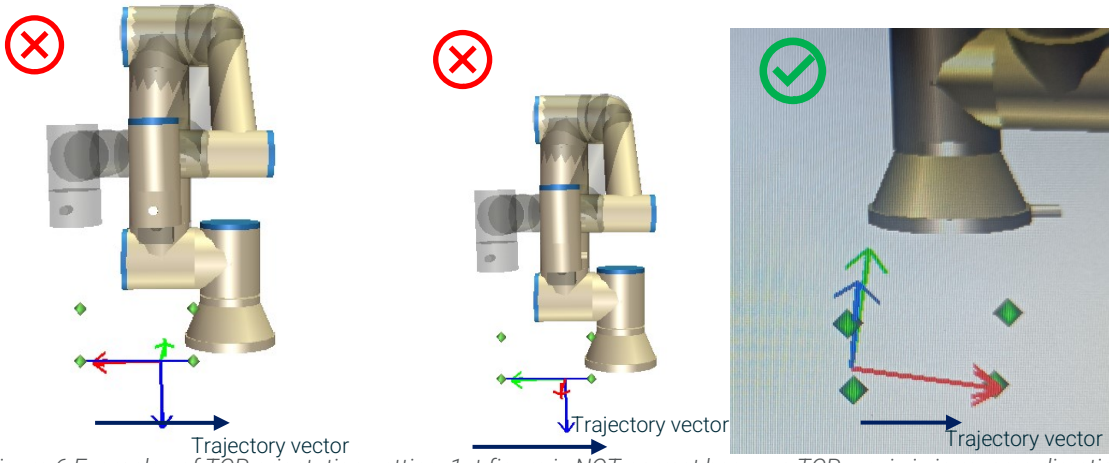


Figure 6 Examples of TCP orientation setting, 1st figure is NOT correct because TCP x-axis is in reverse direction of trajectory vector; 2nd figure is NOT correct because TCP x-axis is not parallel with trajectory vector; 3rd TCP direction is correct.

6th step: click tool bar button on Polyscope top **Save / Save Installation As..** to save current license so user won't have to configure Installation every time. And now all installation is complete.

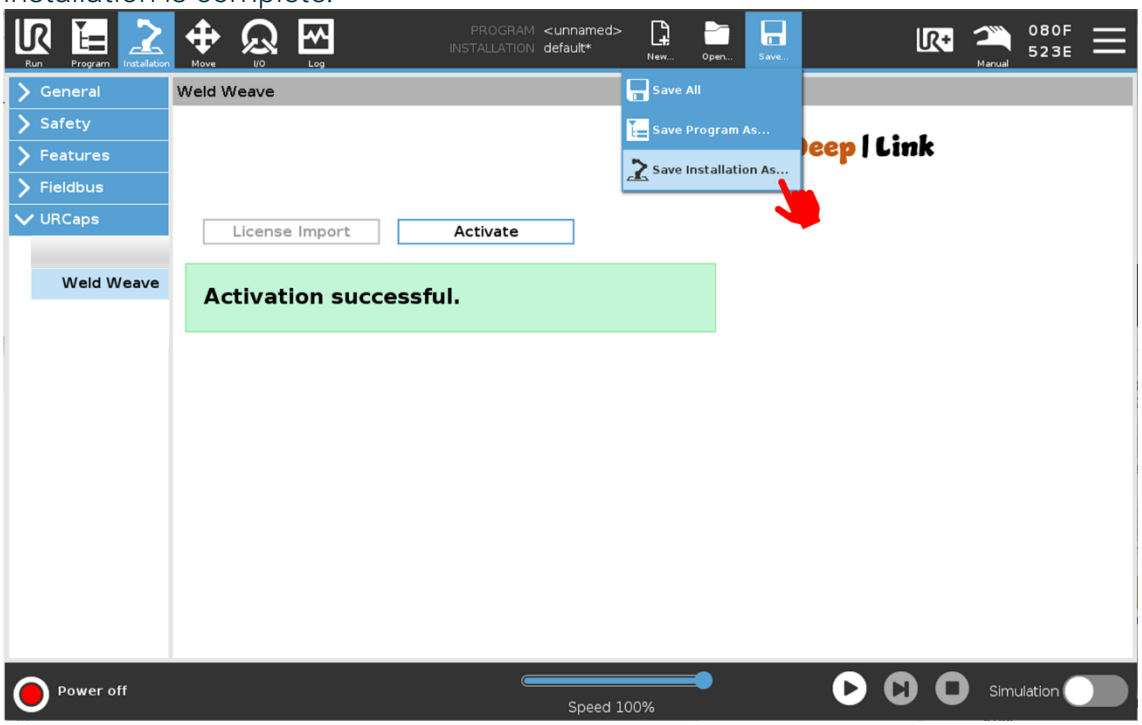


Figure 7 Save installation configuration.

3. Activation

Send your question and request at mailbox: info@deep-link.cn

4. Context configuration

This section demonstrates a most simple setup configuration with the URCap program nodes.

1st step: to create a trajectory with below configuration table:

WAYPOINT	MOVE TYPE	VELOCITY	ACCELERATION	BLEND RADIUS
Waypoint_1	MoveJ/L	Unlimited	Unlimited	Unlimited
Waypoint_2	MoveP	40 mm/s	100 mm/s^2	0
Waypoint_3	MoveP	8 mm/s	30 mm/s^2	0
Waypoint_4	MoveJ/L	Unlimited	Unlimited	Unlimited

Attention: the 8 mm/s in the above example parameter table is the linear velocity during weaving movement.
Recommending use MoveP for weaving segment.
The MoveL is not smooth enough and may more easily lead to Acceleration limit exceed error.

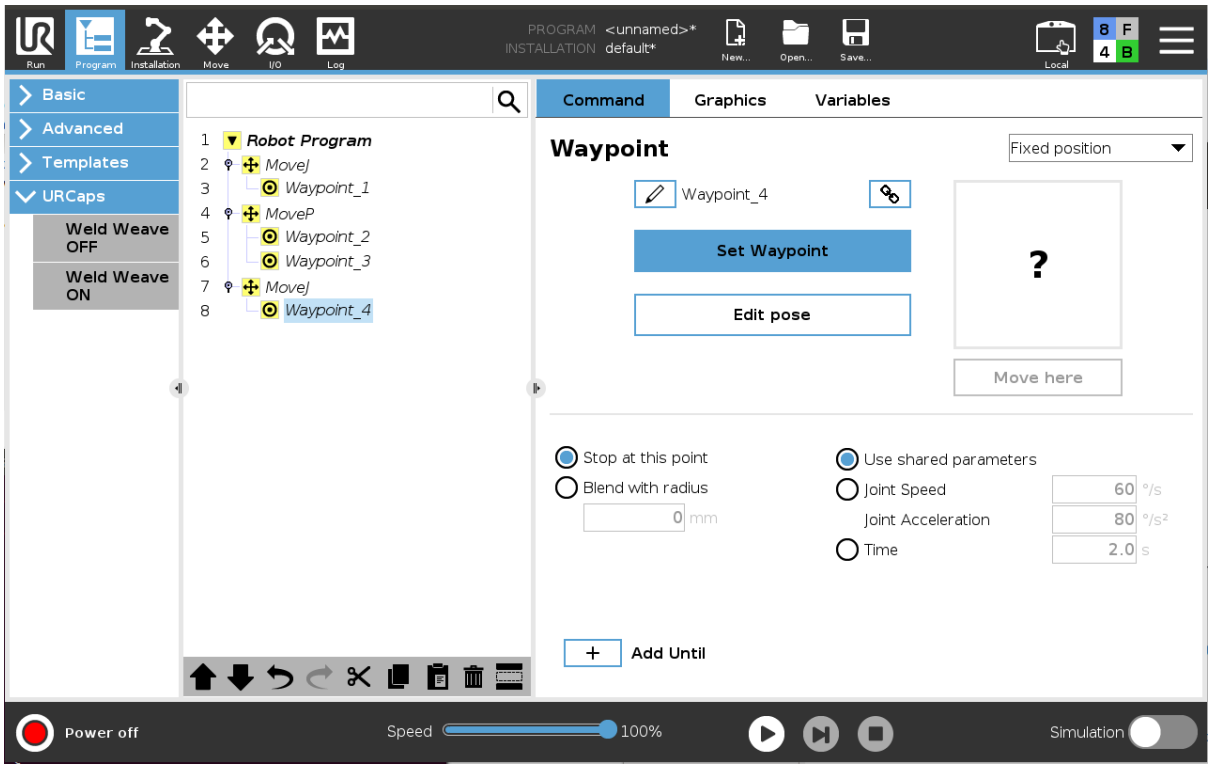


Figure 8 Quick-start 1st step-Building a trajectory

2nd step: Insert a WeldWeaveON program node after the MoveP(Waypoint_2), and insert a WeldWeaveOFF program node after the MoveP(Waypoint_3), as below Figure 9 demonstrated.

Attention: the ON and OFF nodes should always be 1:1 coupled. And adding Weave On/Off node in a nesting way is NOT recommended.

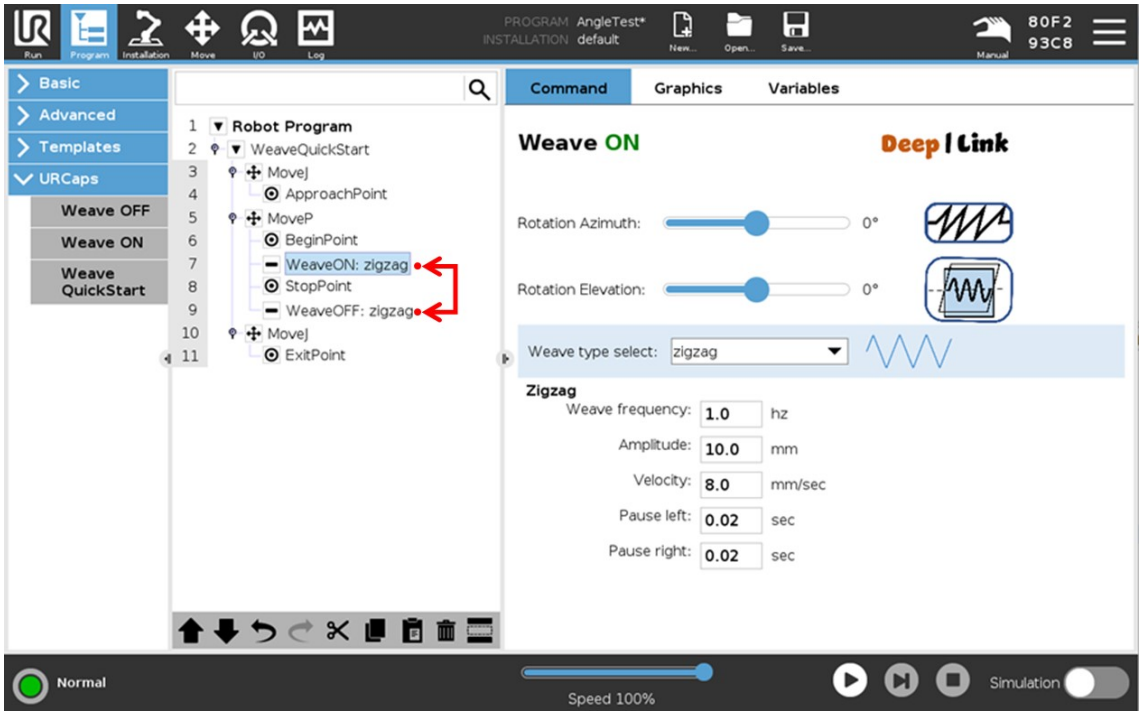


Figure 9 Placing URCap program node in Robot Program, the ON and OFF nodes should always be 1:1 coupled.

3rd step: start configuration <WeaveON> <WeaveOFF> URCap program nodes.

Attention: make sure the WeaveOFF type selection should be same as WeaveON type selection, as Figure 11 explained.

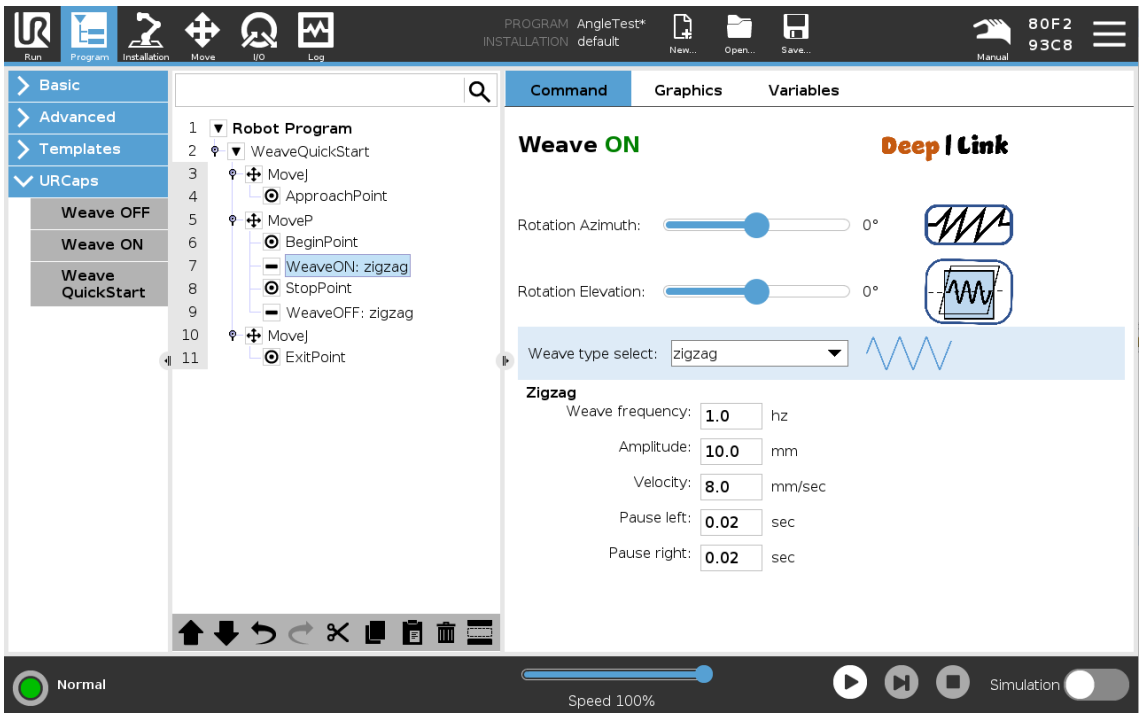


Figure 10 WeaveON node configuration example

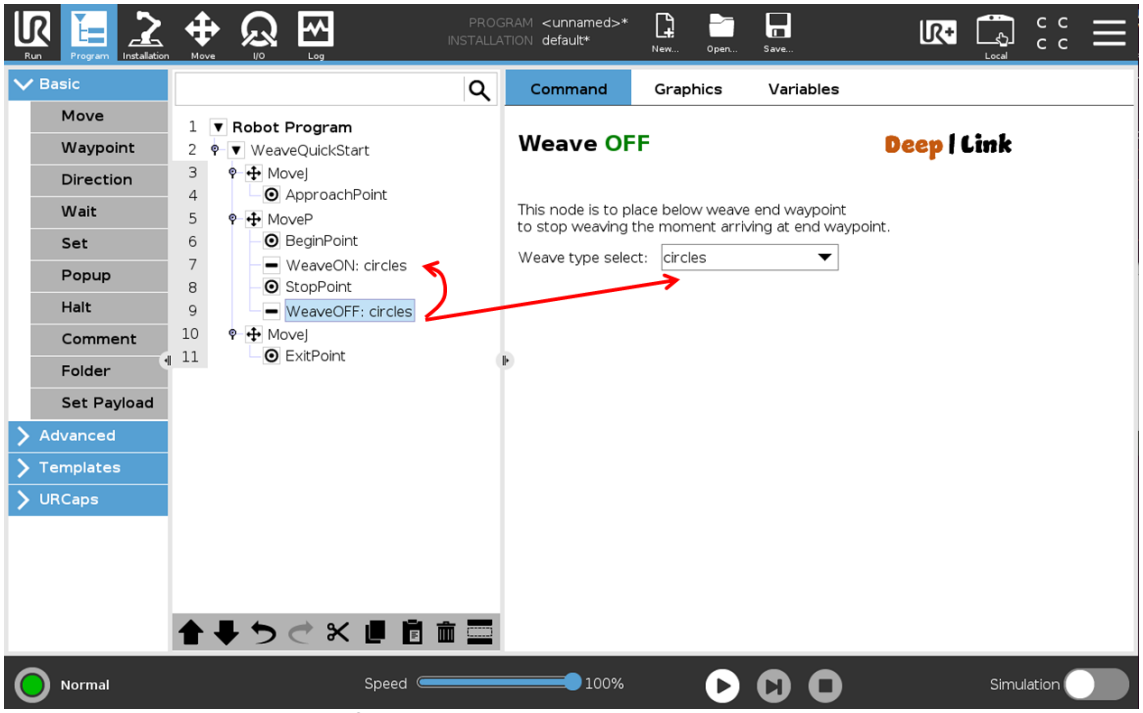


Figure 11 WeaveOFF node configuration example, make sure that the WeaveOFF type is same as WeaveON type.

5th step: double check in case WeaveON node is unfinished [displaying with yellow background] and follow the warning message instruction to modify.

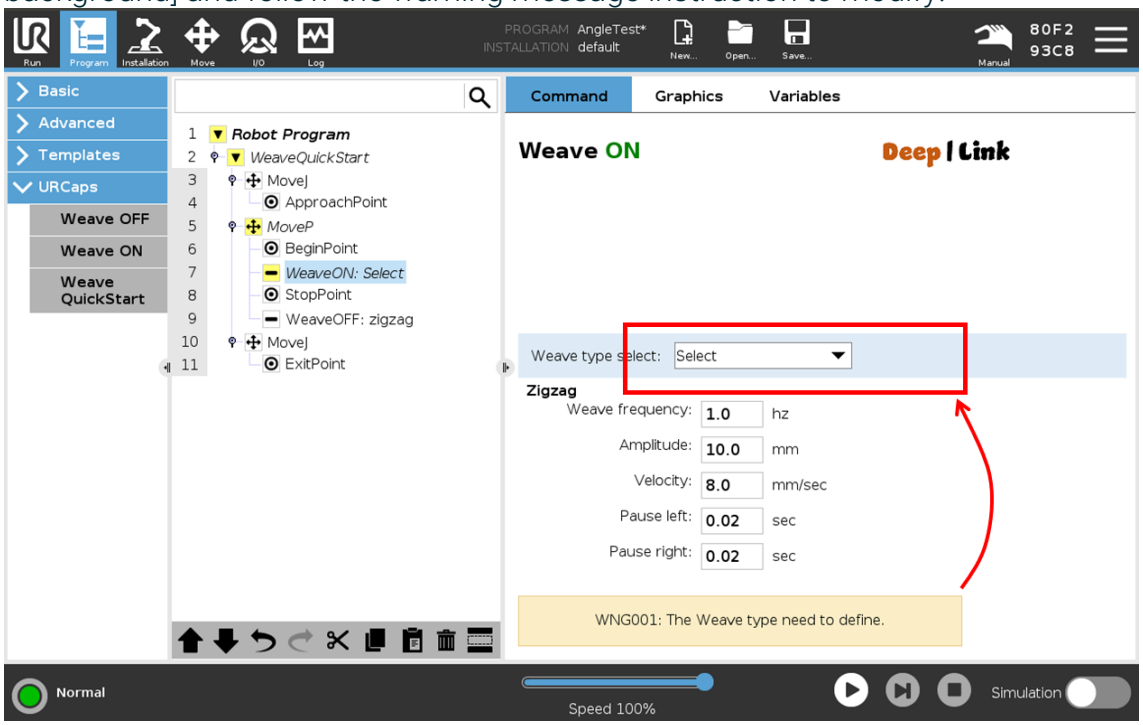


Figure 12 Warning message box

6th step: double check the Velocity parameter is same as MoveP motion parameter. before launching a dry run. Click on the ▶ button on button area of Polyscope.

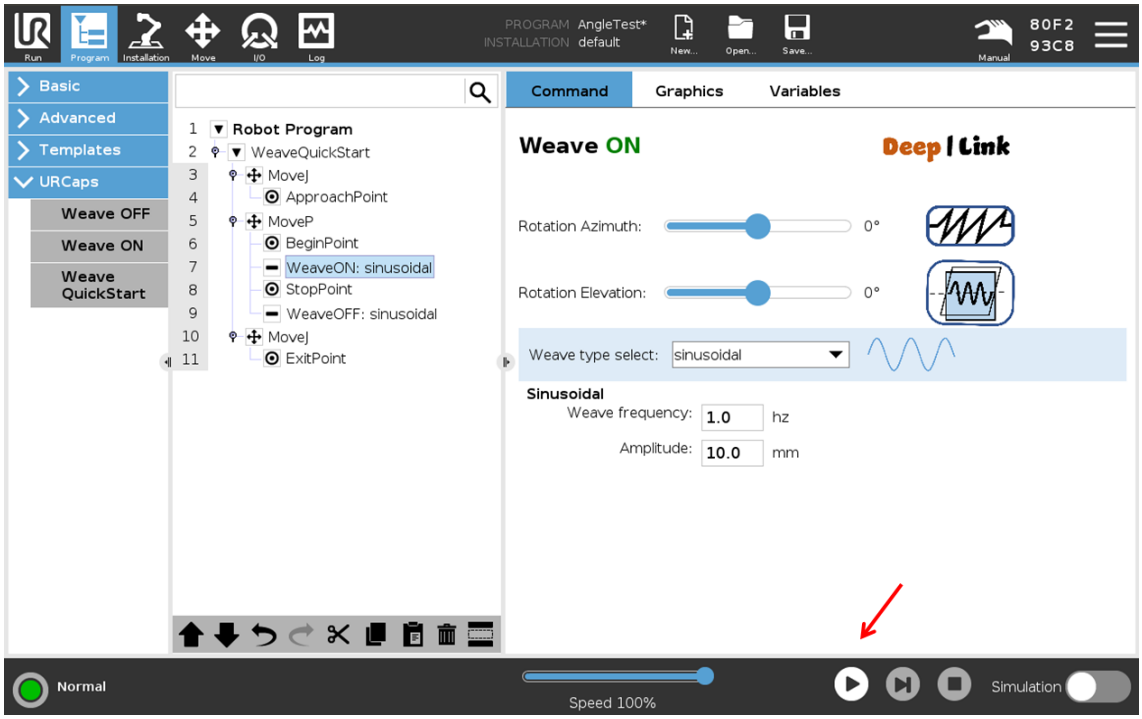
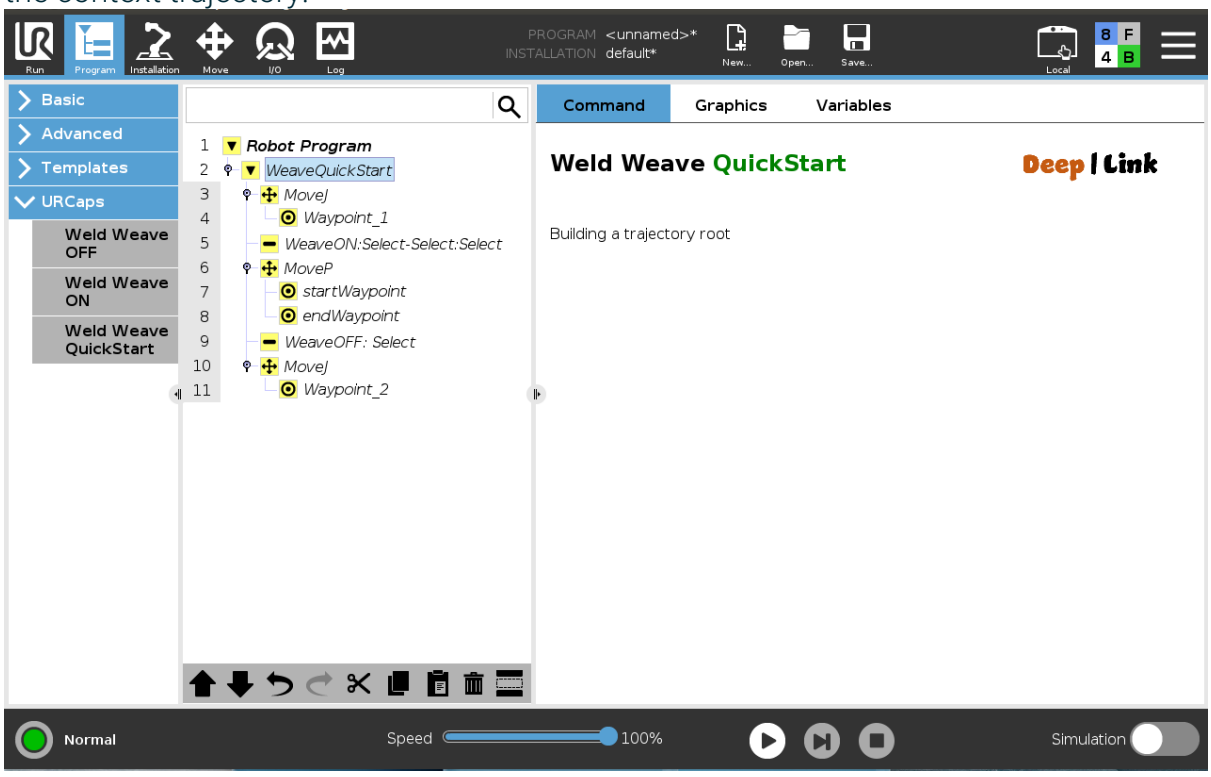


Figure 13 The complete Robot Program ready for weave starts.

Now, you should be ready to start with a weaving dry-run.

5. Quick start

It is possible to quick generate trajectory template/root by adding a Weld Weave QuickStart ProgramNode, which will facilitate user from setting detail parameters in the context trajectory.



6. Weave Configuration

6.1 Weave shape configuration

6.1.1 Circles

For circles shape weaving pattern, the configuration parameters are with Figure 14 circles weaving configuration.

Name	Function	Range	Unit
Weave frequency	The frequency that weaves recurring.	0.2 - 4	Hz
Amplitude	Weaving amplitude	1 – 20	mm
Velocity	TCP linear moving velocity along with original trajectory	1 – 20	mm/sec
Period	Distance between sibling points with same phase position	Read only	mm
Orientation	The direction of circling when TCP offset against original trajectory	Anti-Clockwise Clockwise	

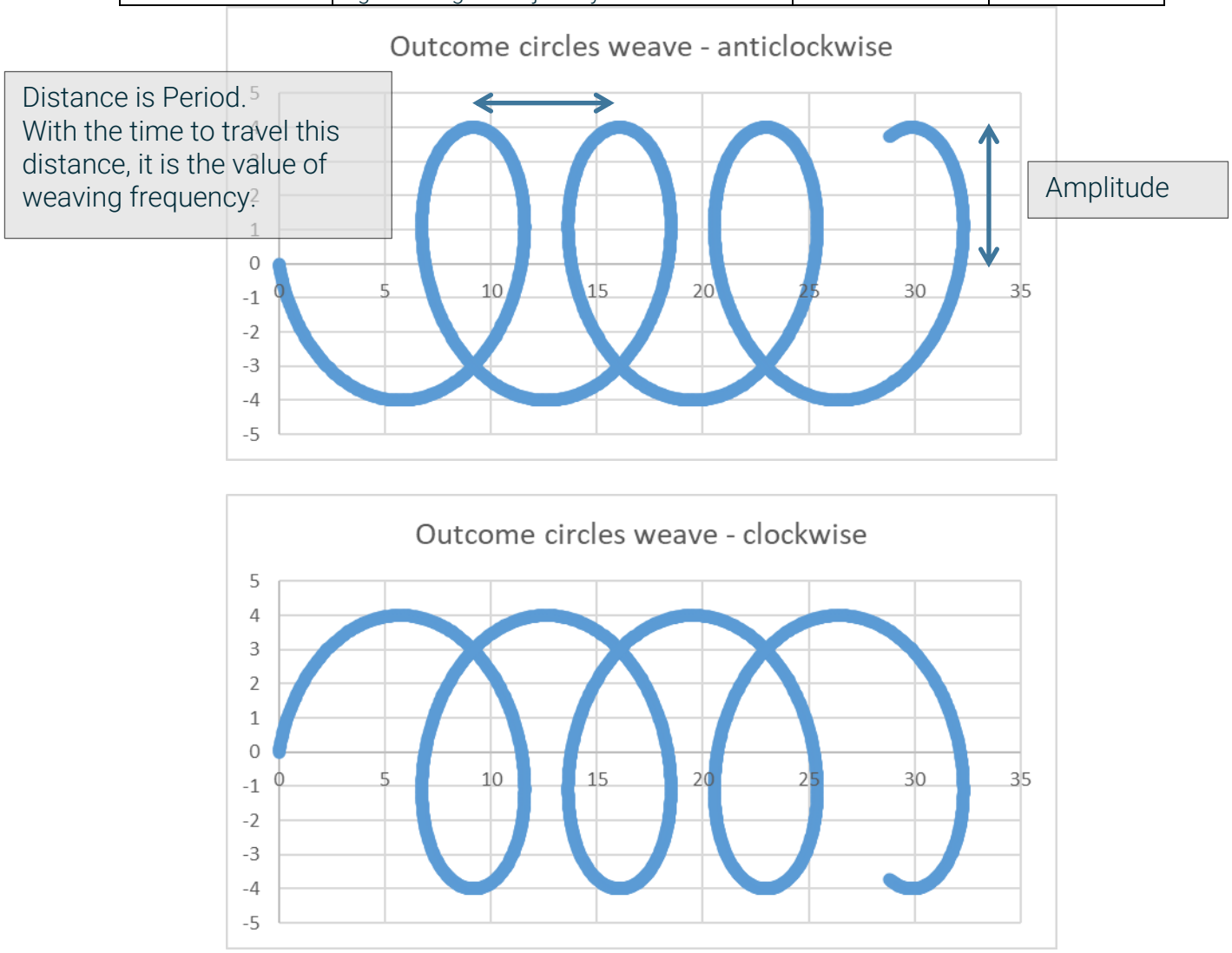
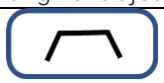
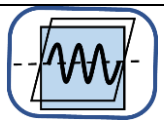


Figure 14 circles weaving configuration.

6.1.2 Zigzag

For zigzag weaving pattern,

Name	Function	Range	Unit
Weave frequency	The frequency that weaves recurring.	0.2 - 4	Hz
Amplitude	Weaving amplitude	1 – 20	mm
Velocity	TCP linear moving velocity along with original trajectory	1 – 20	mm/sec
Pause left	 Time remains on -y axis offset in Tool coordinate	0.02 - 1.0	sec
Pause right	Time remains on +y axis offset in Tool coordinate	0.02 - 1.0	sec
Elevation	 For weaving pattern rotation generated along with forwarding axis	-45 - 45	Degree [°]
Azimuth	For weaving pattern rotation generated alogn with tool Z axis	-45 - 45	Degree [°]

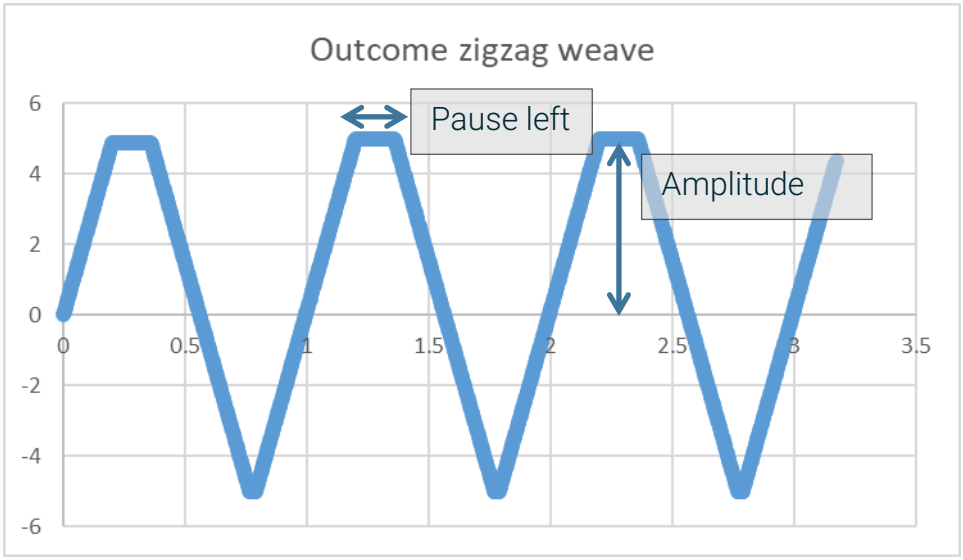


Figure 15 zigzag weaving pattern configuration

6.1.3 Triangle

For triangle weave pattern, explained in Figure 16.

Name	Function	Range	Unit
Weave frequency	The frequency that weaves recurring.	0.2 – 4	Hz
Amplitude	Weaving amplitude	1 - 20	mm
Velocity	TCP linear moving velocity along with original trajectory	1 - 20	mm/sec
Duty_1	Adjust pattern geometry in vertical direction, when value equals 0.25, it is balanced on +y and -y in Tool coordinate.	0.1 - 0.4	
H_ratio	Adjust pattern geometry in horizontal direction, higher value result in more extensive overlap with previous circle.	1.0 - 6.0	

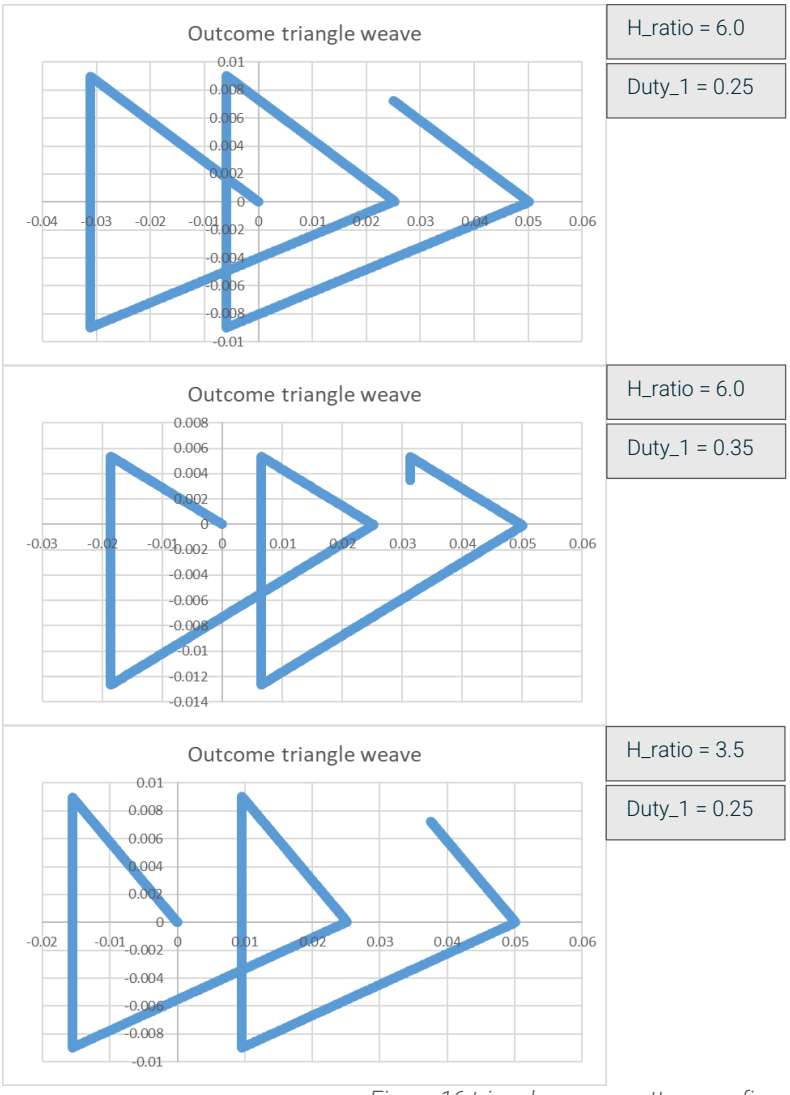


Figure 16 triangle weave pattern configuration

6.1.4 Sinusoidal

For Sinusoidal weave pattern, explained in Figure 17.

Name	Function	Range	Unit
Weave frequency	The frequency that weaves recurring.	0.2 – 4	Hz
Amplitude	Weaving amplitude	1 - 20	mm

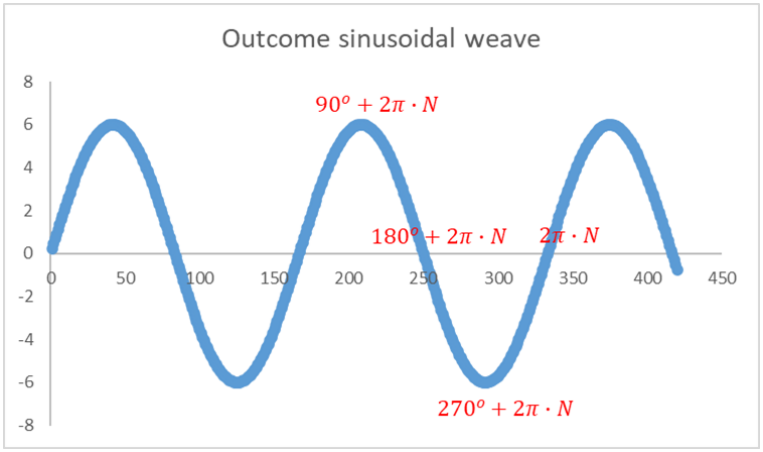
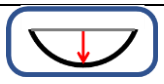


Figure 17 Sinusoidal weave pattern configuration.

6.1.5 Pendulum

For Pendulum weave pattern, it is a spatial weaving implementation that varies offsets both in tool Y direction as well as Z direction, explained in Figure 18 and Figure 19.

Name	Function	Range	Unit
Weave frequency	The frequency [denoted as *f*] that weaves recurring. **The weaving period T equals 1/f + PauseLeft + PauseRight**.	0.2 - 4	Hz
Amplitude	Weaving amplitude	1 – 20	mm
Pause left	Time remains on -y axis offset in Tool coordinate	0.02 - 1.0	sec
Pause right	Time remains on +y axis offset in Tool coordinate	0.02 - 1.0	sec
Depth	 The string depth in Z direction	It varies in function of amplitude, and maximum value is half of defined amplitude.	mm

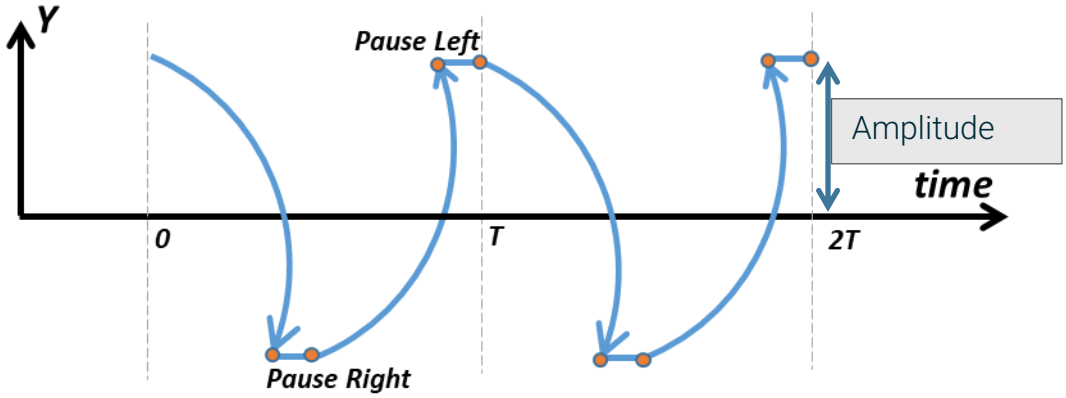


Figure 18 Pendulum weave pattern configuration Y-Time.

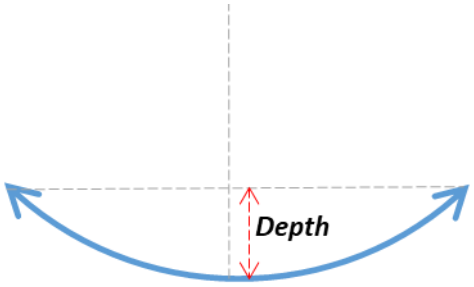


Figure 19 Pendulum weave pattern configuration Y-Z

6.2 dl_angle_RO_ext, dl_offset_RW_ext, dl_weave_enable_RO_ext,

This global ***dl_angle_RO_ext*** variable [ReadOnly, Type real, Unit Rad] is available indicating current angle that is running inside of sinusoidal weave pattern, as shown in below Figure 17. This public value would be used when using i.e., arc sensing will need to collect process data at certain ***dl_angle_RO_ext*** [ReadOnly, Type Boolean,] value in a cyclic way.

The global ***dl_weave_enable_RO_ext*** [ReadOnly, Type[x,y,z,rx,ry,rz], Unit[m,m,m,rad,rad,rad]] is used to indicating when path_offset is activated.

The global ***dl_offset_RW_ext*** variable [ReadWrite] is capable to collect 3rd party device measuring data that would affect path offset, for instance, a line laser guiding device needs to adjust robot motion in function of offset to the welding seam, and ***dl_offset_RW_ext*** can be used to apply the real-time offset into current weaving.

7. Error instruction

CATEGORY	NUMBER	DESCRIPTION
Warning	WNG001	The Weave type need to define.
Warning	WNG002	Reserved.
Warning	WNG003	Reserved.
Warning	WNG004	Pause left and right sum time is too long in consideration with current Weave frequency value.
Error	Error001	Pause time sum tool long, or weave frequency too high. This is only for Zigzag type.

Error	Error002	Weave frequency too small or velocity too high, this is only for Triangle type.
Error	Error003	Compile error: name 'Thread_XXXX_han', this is because WeaveON and WeaveOFF type selections are not same.
Error	Error004	Path offset: Not enable, there are several reasons that can lead to such error, one most possible cause is the Blend radius setting is huge. To be conservative, we recommend this value is no bigger than 2 mm.
Error	Error005	There is a relative huge value bias between WeaveON velocity value and real value.
Error	Error006	Path offset: Maximum acceleration limit exceed. This may be caused by many reasons, one most possible reason is the WeaveOFF not properly started, try click on this node and other node, and re-launch program again.
Error	Error007	String depth is too large comparared to amplitude.
Error	Compile error	Compile error: name xxxxx is not defined. Check if license password is cleared in Installation Node.

8. Script Function list

8.1 Function table

NAME	Description	Dependancy	Version
weave_circles (weave_para_st)	Calculate circles type offset in realtime	weave_circles_init.script mod.script weave_para_structure	1.0.0
weave_triangle(weave_para_st)	Triangle type offset generator	weave_triangle_init.script mod.script weave_para_structure	1.0.0
weave_zigzag(weave_para_st)	Zigzag type offset generator	weave_zigzag_init.script mod.script weave_para_structure	1.0.0
mod(number, divisor)	Re-implementation of complementation calculation	null	1.0.0
weave_circles_init(weave_para_st)	Circle initialization	weave_para_structure	1.0.0
weave_zigzag_init()	Zigzag initialization		1.0.0
weave_triangle_init(weave_para_st)	Triangle initialization	weave_para_structure	1.0.0
weave_para_structure	Weave parameters definition	null	1.0.0

Attention:
All global variables used in this urcap scope follow a format of 'dl_xxxx'.
To avoid interference in program, change another format.

User can call URCap function via ExpressionEditor to customized own Robot Program, shown in Figure 21.

8.2 Weave parameter structure

```
# ADDRESS           DESCRIPTION                 TYPE          INITIAL_VALUE        UNIT
# 0-9 are general parameters
# weave_para_st[0]  ctrl_frequence              integer       500                  Hz
# weave_para_st[1]  weave_frequence             integer       1                    Hz
# weave_para_st[2]  amplitude[maximum 0.006]    real          0.005                M
# weave_para_st[3]  pi                          real          3.14159              na
# weave_para_st[4]  velocity                    integer       0.008                M/Second
# weave_para_st[5]  path_offset_type            Integer       1                    na
# weave_para_st[6]  reserved
# weave_para_st[7]  reserved
# weave_para_st[8]  reserved
# weave_para_st[9]  reserved

# 10-19 are process related parameters
# weave_para_st[10] Orientation[0-L,1-R]        boolean       True                 na
# weave_para_st[11] TimeStamp_ini               real          0                    Second
# weave_para_st[12] Angle_ini                   real          0                    Degree
# weave_para_st[13] x_TCP_ini                   real          0                    M
# weave_para_st[14] y_TCP_ini                   real          0                    M
# weave_para_st[15] Dwell_left[minimum 0.02]    real          0.2                  Second
# weave_para_st[16] Dwell_right[minimum 0.02]   real          0.2                  Second
# weave_para_st[17] duty_1                      real          0.25                 M/Second
# weave_para_st[18] Tolerence                   real          0.002                Second
# weave_para_st[19] H_ratio[1~6]                real          6                    na
```

Figure 20 weave parameter data structure

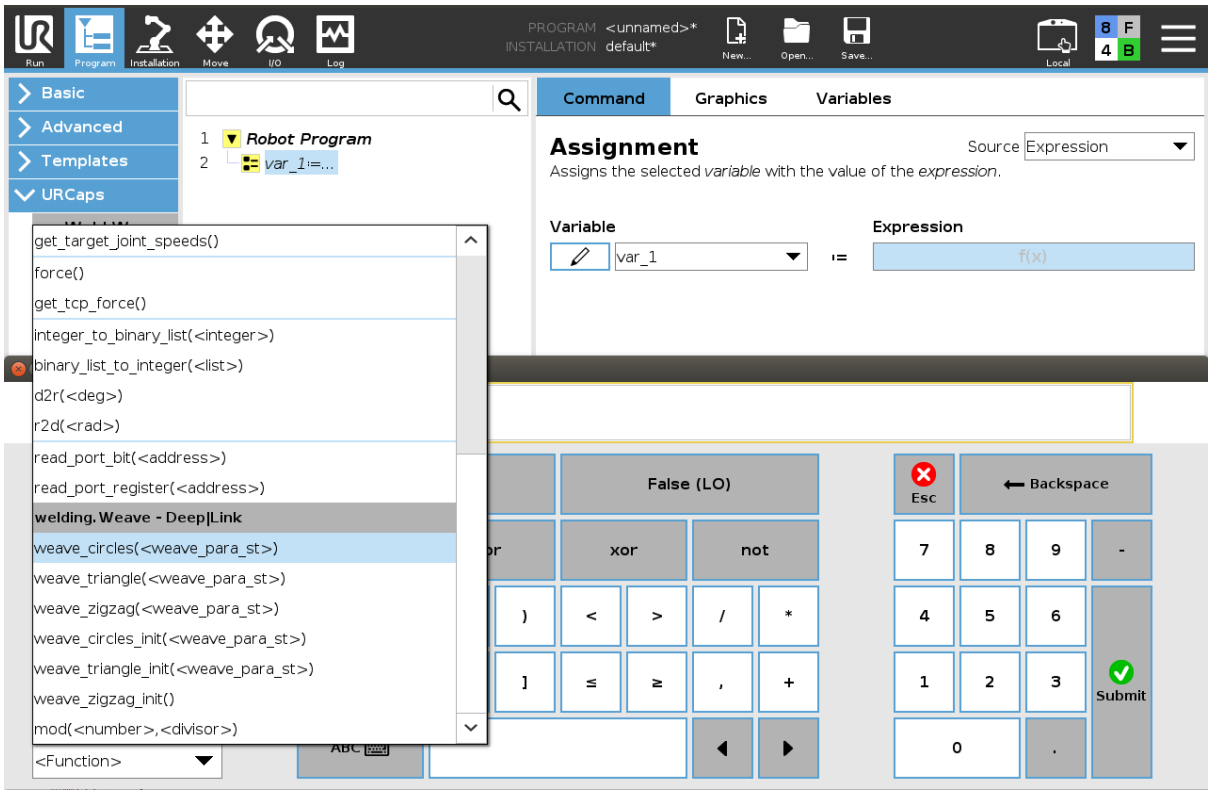


Figure 21 call URCap functions in ExpressionEditor, this feature is available only when license cleared.

9. Revision notes

Version	Description	Time
1.0.0	Initial release	2022-Oct-24
1.0.1	Update global variable format, Add urcap function in ExpressionEditor. Add Polyscope log-tab message.	2022-Nov-14
1.0.2	Add simulation function.	2022-Nov-19
1.0.3	Update interface. Add QuickStart program node	2023-Jan-06
1.0.4	Add sinusoidal weave pattern. Fix warning message box display bug.	2023-May-12
1.0.5 Build005	Fix double weave node thread error bug. Add translation CN.	2023-June-03
1.0.5 Build010	Add Copyright section, Remove Script line of simulation enable, Simplify instruction of adding before Start Assignment. Improve translation CN. Improve threads runtime performance.	2023-July-15
1.0.5 Build022	Add azimuth and elevation rotation feature for Sinusoidal & Zigzag pattern.	2023-August-03
1.0.5 Build024	Fix activation bug	2023-October-31
1.0.5 Build025	Add trial-activation interlock	2023-November-09
1.0.5 Build026	Optimize WeaveOn node interface by converting combobox to button. And add extra buttons for faster programming with graphic interface. Remove damping effect of ZIGZAG and CIRCLES. Adjust ZIGZAG pause safety percentage to 90%	2023-December-05
1.0.6.Build002	Add PENDULUM pattern. Update Chinese translation.	2023-December-18
1.0.6.Build003	Optimize activation approach.	2023-December-27
1.0.6.Build006	Fix offset error when returning zero at trail.	2024-April-17
1.0.6.Build007	Fix error of ProgramNode deformation. Add MAC info in Installation node.	2024-May-07
1.0.6.Build009	Add Sinusoidal pattern pause parameter. Adjust Pendulum pause compensation.	2024-June-15
1.0.6.Build012	Fix error on ArcSensing final stage. Fix Sinusidal pattern pause error when frequence is larger than 2. Add pokayoke in Pendulum pattern panel. Switch to ursim-5.15.0.126572	2024-July-12
1.0.6.Build022	Update activation experience. Add simulator variant. Fix installation icon deformation issue.	2025-April-30
1.0.6.Build023	Upgrade Polyscope base to 5.19.0+. Upgrade URCap API base to 1.15.	2025-May-31

	Upgrade URSIM support version to 5.19.0.1210631. Adjust Sinusoidal & Pendulum weaving pause MAX limit which equals dynamically 5/frequency. Remove Weave QuickStart Node following an effort of merging Weave URCap with Power URCap. Change default Weave type to Sinusoidal pattern.	
1.0.6.Build024	Fix Zigzag weave offset error at specific frequency. Add Pendulum depth validation in function of amplitude. Fix Pendulum depth validation error in Sinusoidal weave	2025-July-16
1.0.6.Build025	Streamline non-activation behavior.	2025-December-22

*****Chinese Documentation / 中文手册 *****

1. 简介

这个产品是为 Universal-Robots 定制开发的软件插件产品，运行在 Universal-Robots 示教器上。这个软件实现了多种摆动形状的运动轨迹。得益于精确的**周期控制，**它提供**可编程的摆动波形，**同时保持稳定的 TCP **工艺速度**可以最大限度保证焊接质量。它现在可以提供 **圆形螺旋摆动，三角形摆动，Z 字摆动，正弦摆动，和时钟摆动。**
Deep-Link 的团队正致力于增加更多的可编程摆动形状。开发这个产品的目的是提供一个**模块化的插件**，可以紧密集成在客户的机器人程序中加快现场编程速度，降低编程难度，提高摆动焊接质量。这个产品最大限度使用示教器 Polyscope 中原生指令，保证了用户**一致的使用体验**和**最大的灵活性。**

此 URCap 产品对机器人软件版本要求：

Universal Robots
e-series: 5.19 或更新的版本。

2. 安装

第 1 步: 点击屏幕右上方的汉堡菜单

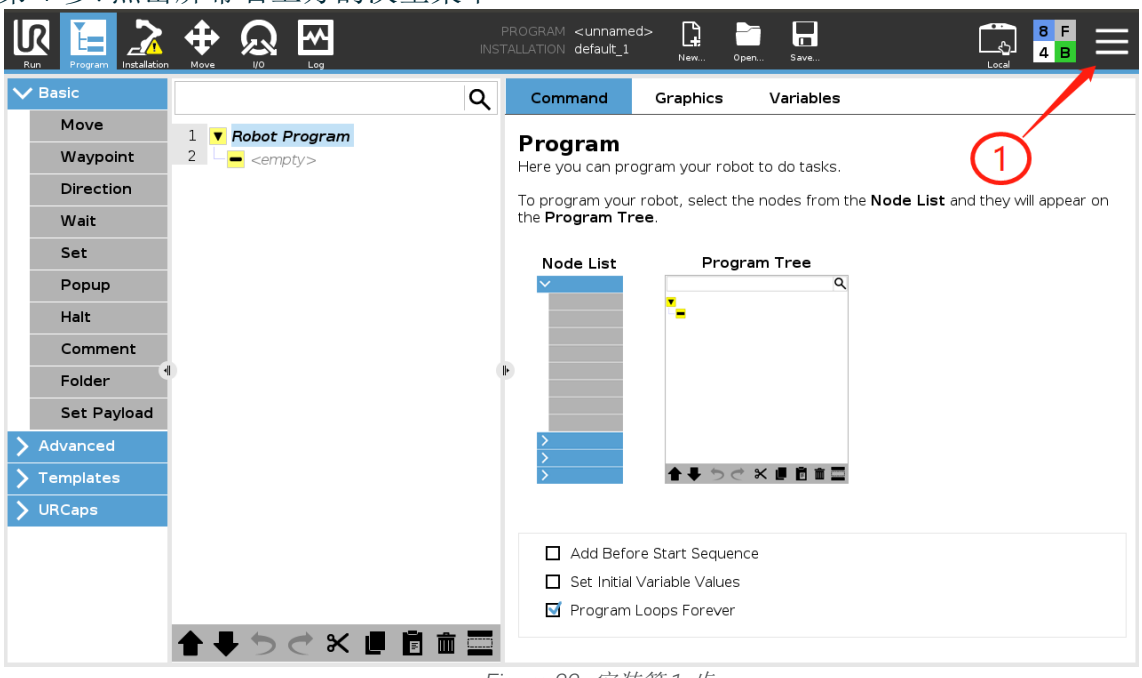


Figure 22 安装第 1 步

第 2 步：选择 **设置 / 系统 / URCaps**，随后点击屏幕下方的 “**+**” 按钮。

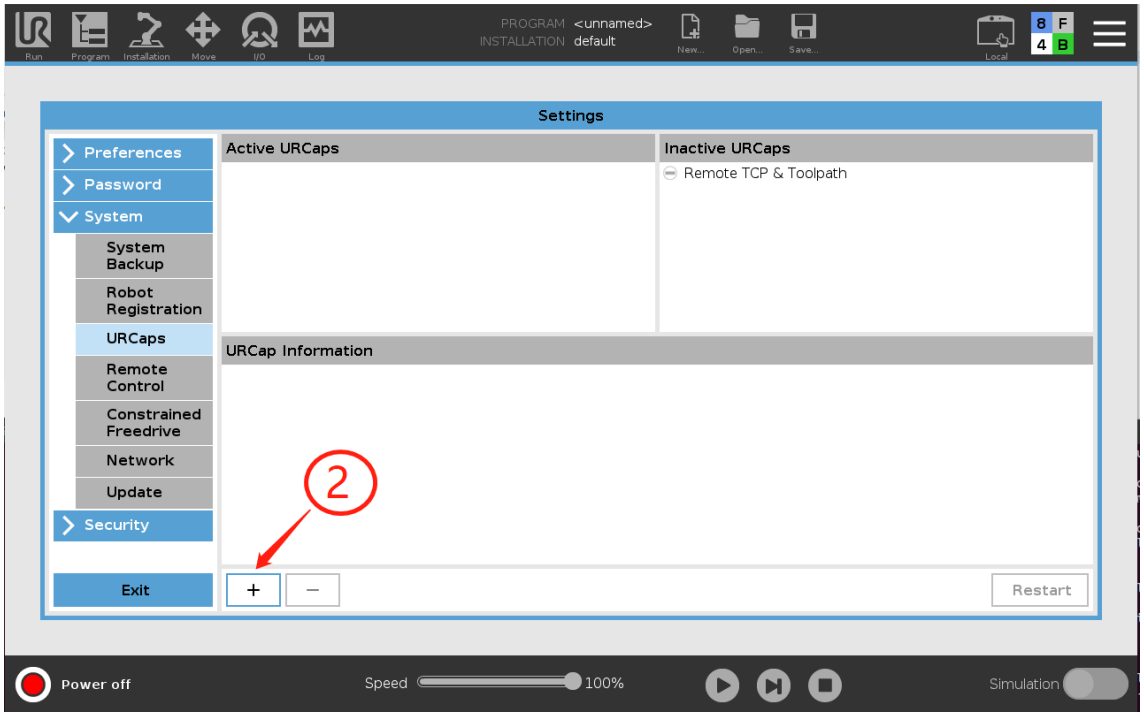


Figure 23　安装第 2 步

第 3 步：从文件浏览器中选择 <welding.Weave-1.0.x .urcap> 文件并点击**打开** 按钮，随后选择 **重新启动**。如图所示 Figure 24。

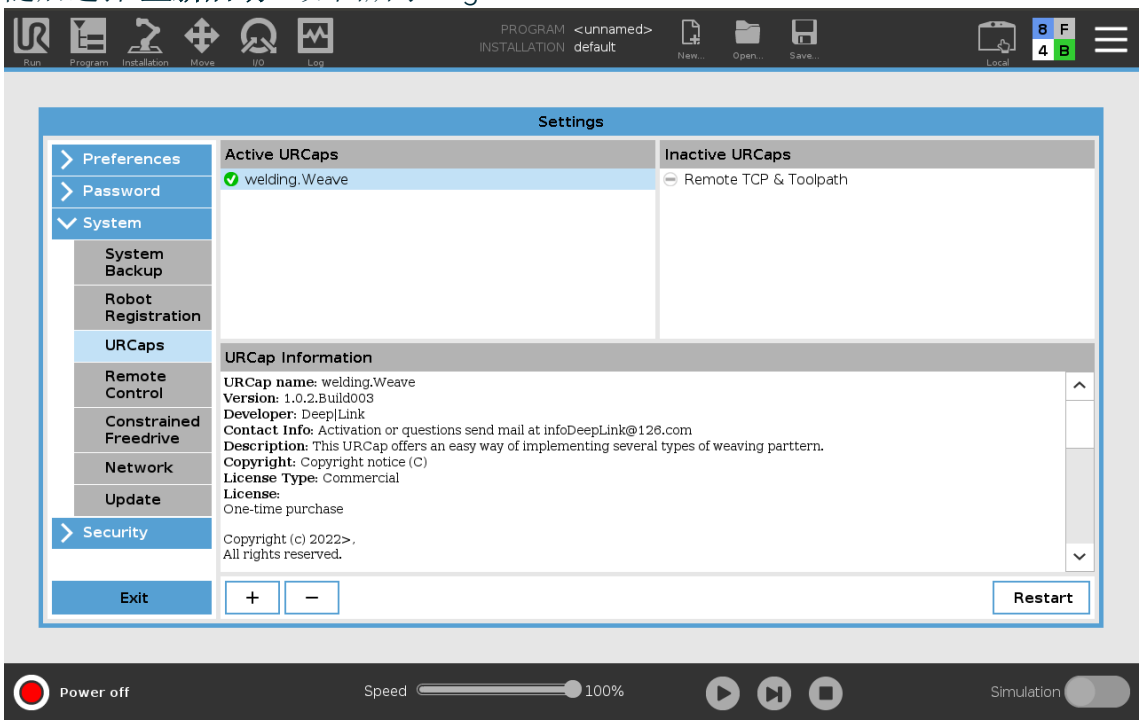


Figure 24　安装 URCap 后重启机器人

第 4 步：点击 **安装设置** 并输入激活码，直到 Figure 25 显示激活成功。

Figure 26　安装第 4 步，激活 URCap

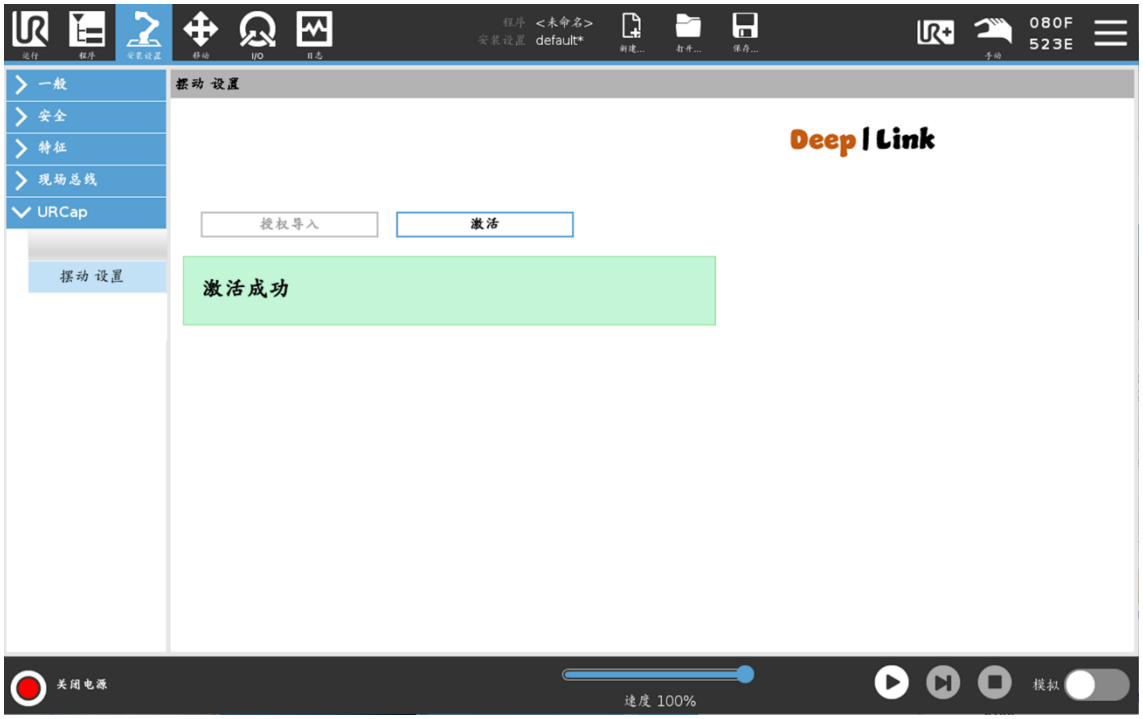


Figure 27　URCap 激活成功

第 5 步：最后一步是配置工具坐标系，请查阅 Universal-Robots 产品手册的示教工具坐标系方法，并保证 TCP 的 X 轴指向运动轨迹矢量。如图所示 Figure 28。

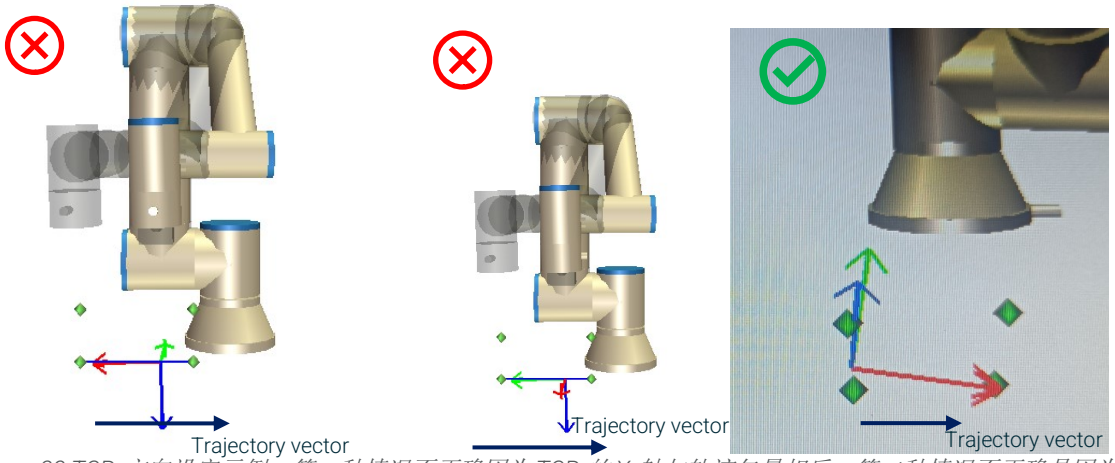


Figure 28 TCP 方向设定示例，第一种情况不正确因为 TCP 的 X 轴与轨迹矢量相反；第二种情况不正确是因为 X 轴与轨迹矢量不平行；第三种情况是正确的。

第 6 步：点击屏幕 **保存 / 保存安装设置为..** 以保存之前进行的安装设置操作，以上安装完毕。

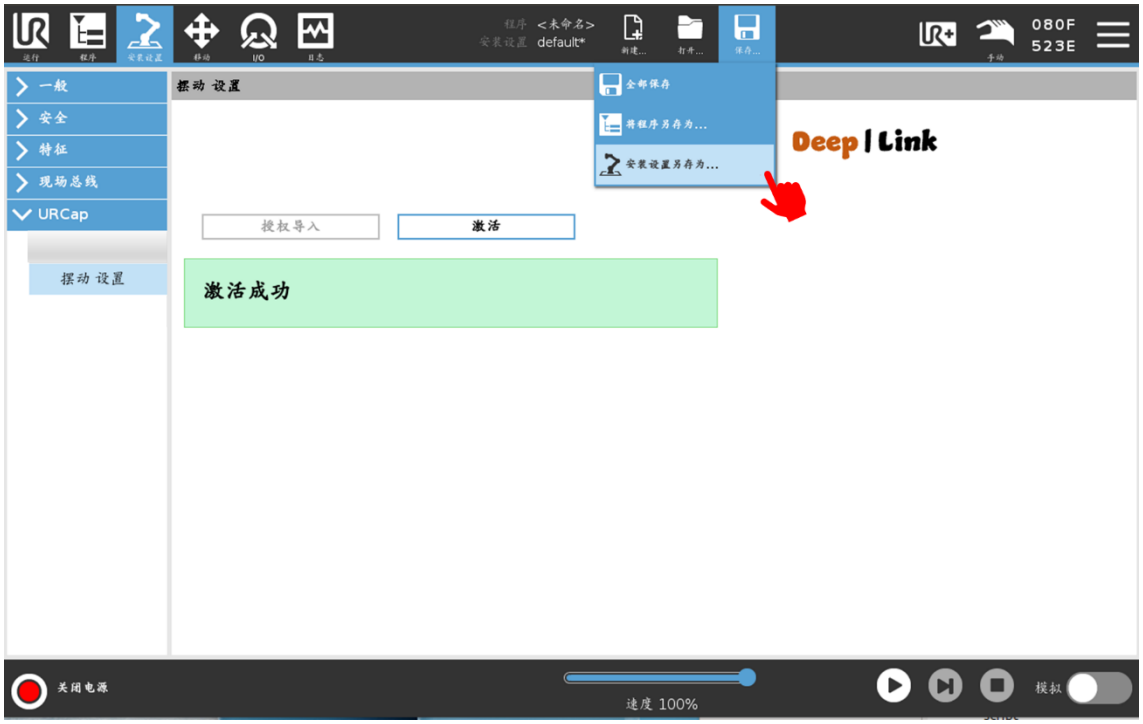


Figure 29 保存安装设置

3. 激活

请将您的激活需求或问题发送邮件至：info@deep-link.cn

4. 场景配置

这一章节将介绍 URCap 编程节点。

第 1 步：创建一个路径：

路点	移动类型	速度	加速度	交融半径
Waypoint_1	MoveJ/L	Unlimited	Unlimited	Unlimited
Waypoint_2	MoveP	40 mm/s	100 mm/s^2	0
Waypoint_3	MoveP	8 mm/s	30 mm/s^2	0
Waypoint_4	MoveJ/L	Unlimited	Unlimited	Unlimited

注意： 8 mm/s 是摆动时机器人沿轨迹矢量的线速度。
推荐使用 MoveP 作为摆动路段的运动类型，因为 MoveL 将导致 TCP 运动不够平滑并容易导致加速度超限错误。

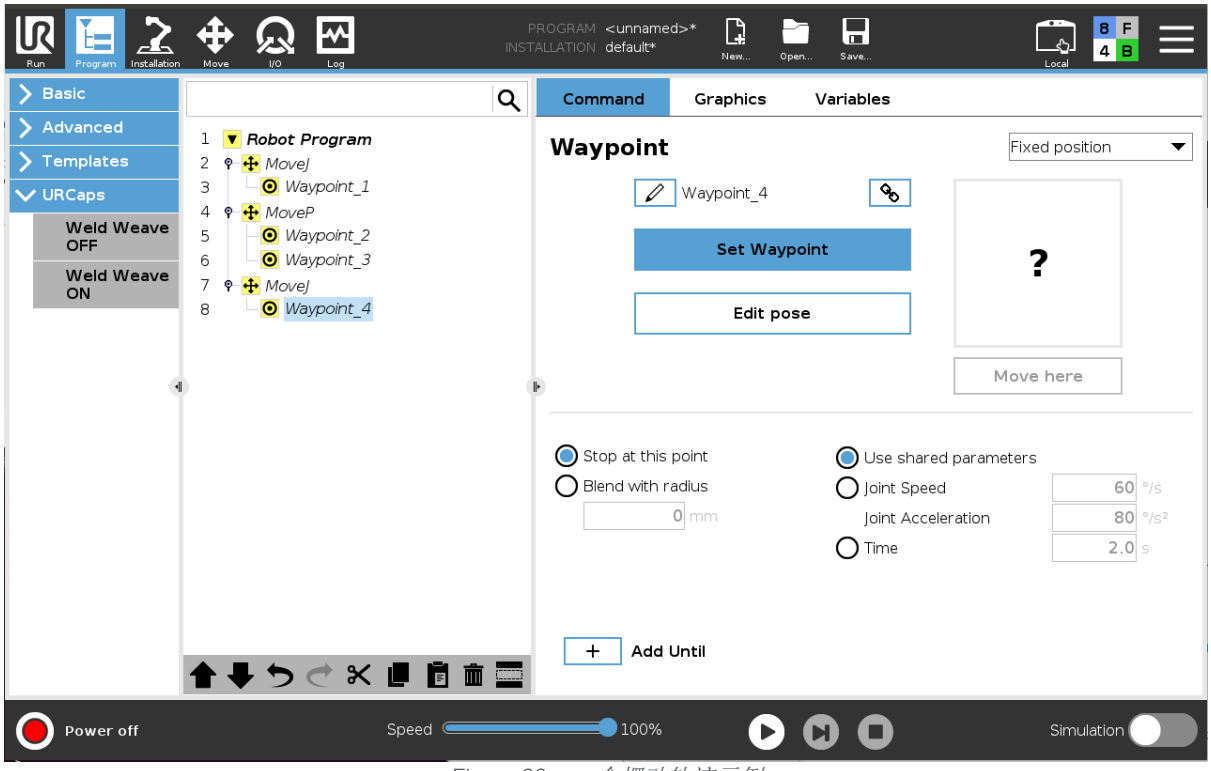


Figure 30　一个摆动轨迹示例

第 2 步：在 MoveP(Waypoint_2)之后插入**摆动开始**，在 MoveP(Waypoint_3)之后插入**摆动停止**节点如图所示 Figure 31 摆动开始和摆动结束节点在程序中应该始终配套使用。

注意： ON 和 OFF 节点应该始终配套使用。不推荐嵌套插入 On/Off 节点。

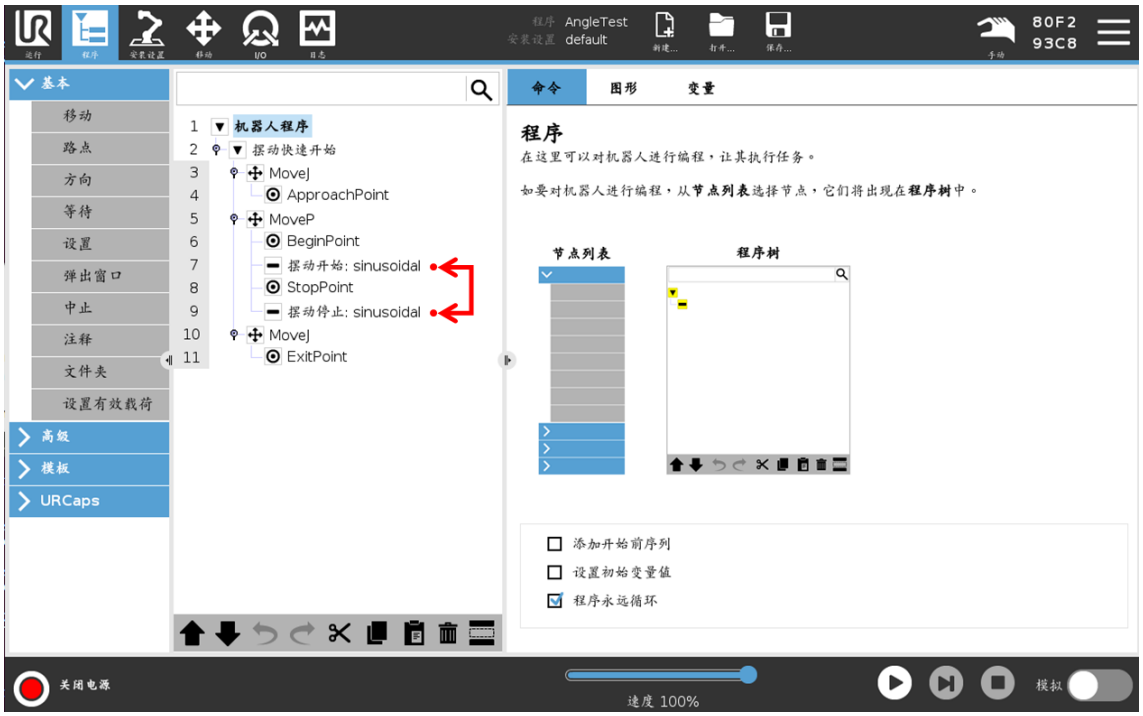


Figure 31 摆动开始和摆动结束节点在程序中应该始终配套使用。

第 3 步：开始配置 <摆动开始> <摆动结束> URCap 程序节点。

注意：需要保证 摆动停止节点中的摆动类型 应与摆动开始节点中的摆动类型 一致，如图所示 Figure 33。

Figure 32 摆动开始节点设置

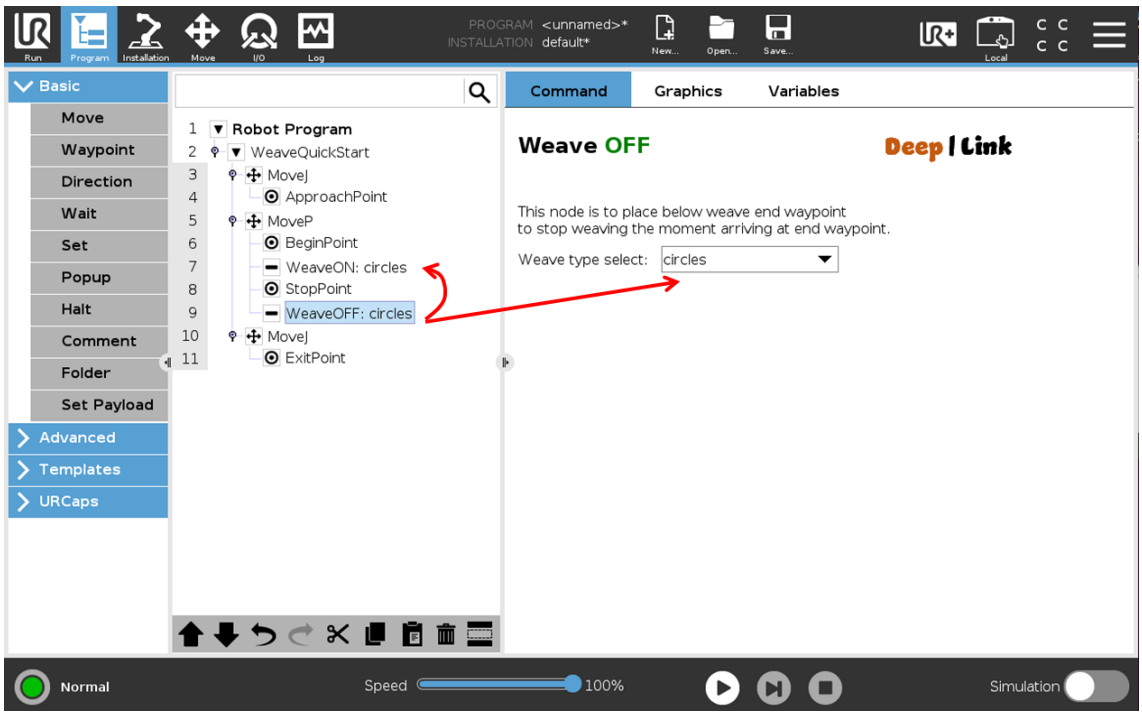


Figure 33 摆动结束节点的摆动类型选择应与对于的开始节点一致。

第 4 步：检查摆动开始节点中的报警信息 [如有报警，节点显示未完成状态]。

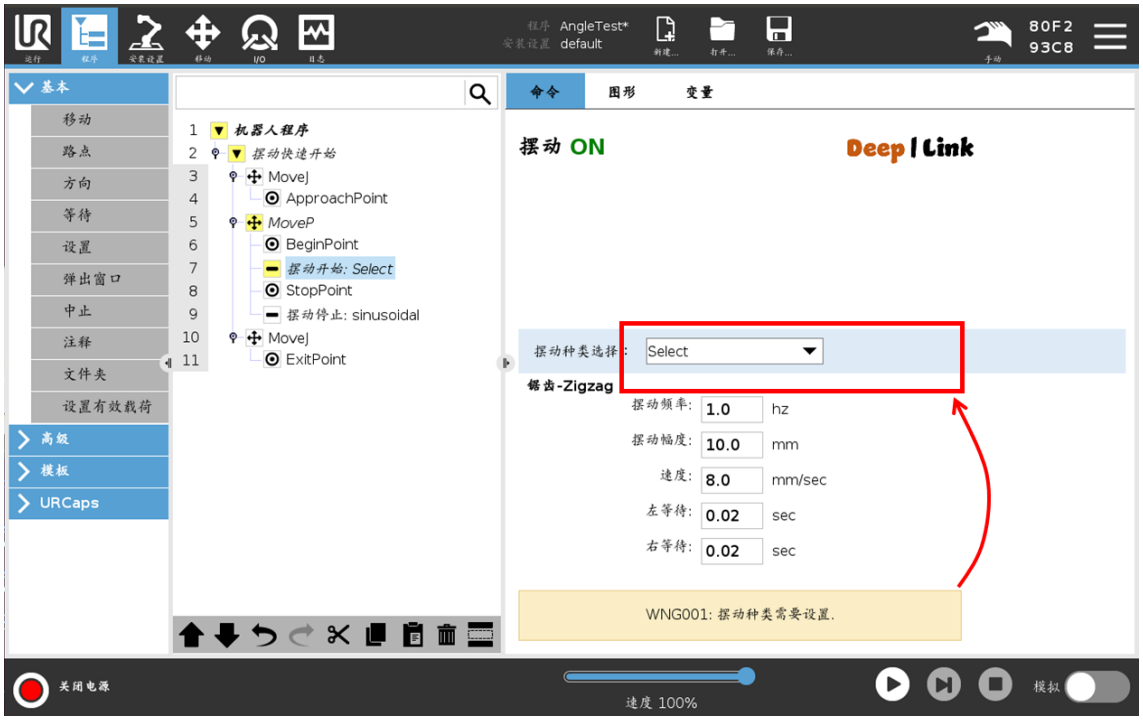


Figure 34 报警信息框

第 5 步：请再一次确认摆动开始节点中的速度与实际轨迹 MoveP 中运动速度一致，随后可以点击屏幕下方的▶ 按钮运行程序。

5. Quick start 快速开始

QuickStart 节点提供一个配置好的摆动模板，用户插入后只要示教 4 个路点即可以开始运行程序。

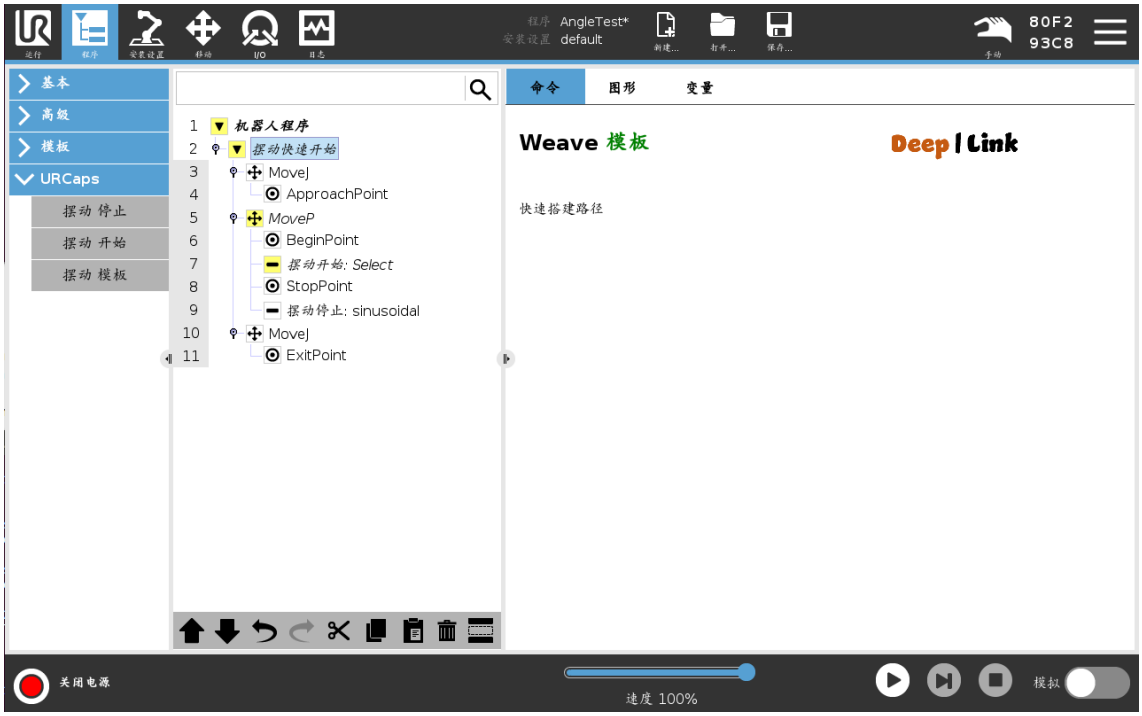


Figure 35 快速开始节点

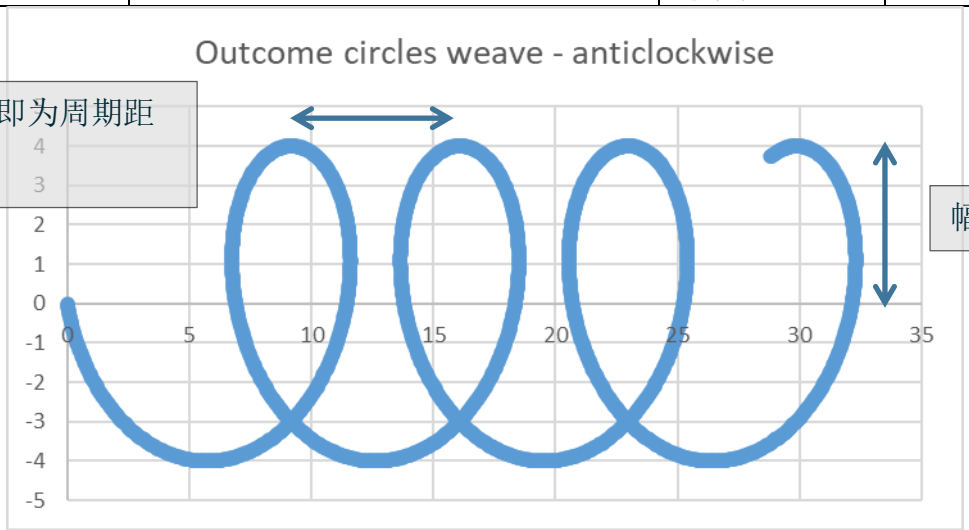
6. 摆动配置

6.1 摆动形状配置

6.1.1 圆形螺旋摆动

对于圆形螺旋摆动波形，可配置参数如下表和 Figure 36 所示：Figure 14 circles weaving configuration

名称	功能	范围	单位
摆动频率	摆动的频率，值越大则单位时间内摆动往复次数越多	0.2 - 4	Hz
幅度	摆动单侧幅度值	1 – 20	mm
速度	轨迹矢量的 TCP 速度	1 – 20	mm/sec
周期距离	相邻周期同相位的点的距离	只读	mm
方向	圆弧往复旋转的方向	顺时针 / 逆时针	



图中指示的距离即为周期距离。

幅度

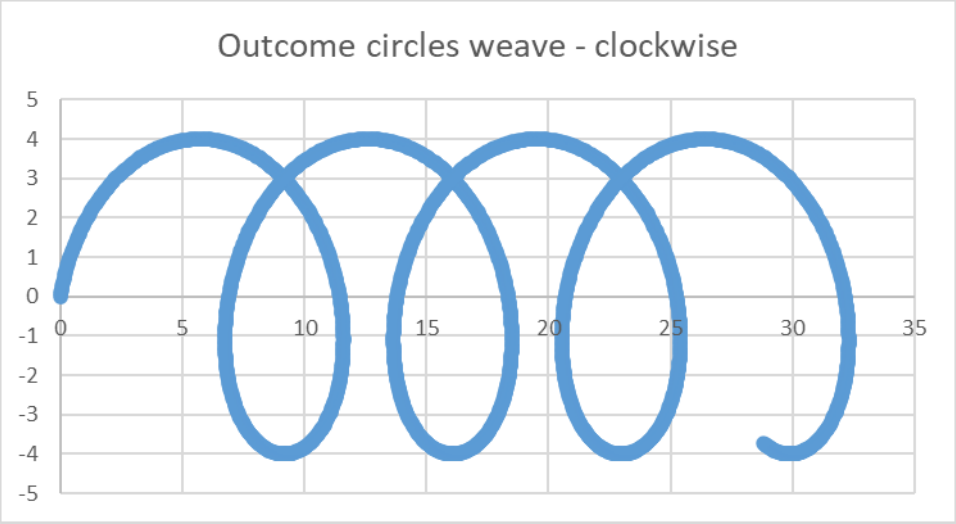
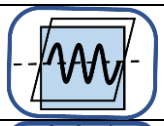


Figure 36 圆弧摆动配置

6.1.2 Z 字摆动

Z 字摆动配置，如 Figure 37 所示。

名称	功能	范围	单位
摆动频率	摆动的频率，值越大则单位时间内摆动往复次数越多。	0.2 - 4	Hz
幅度	单侧摆动幅度	1 – 20	mm
速度	轨迹矢量的 TCP 速度	1 – 20	mm/sec
左等待	TCP 在左侧停留时间	0.02 - 1.0	sec
右等待	TCP 在右侧停留时间	0.02 - 1.0	Sec
操作角 Elevation		-45 – 45	度［°］
摆动方向 Azimuth		-45 – 45	度［°］

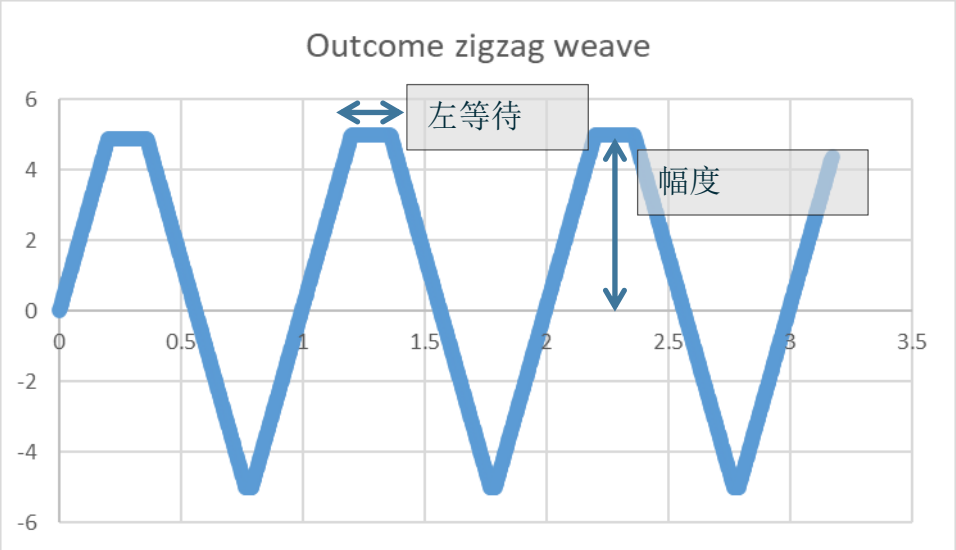
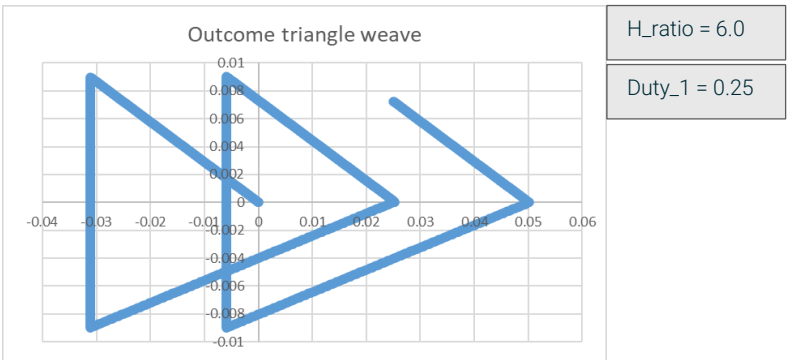


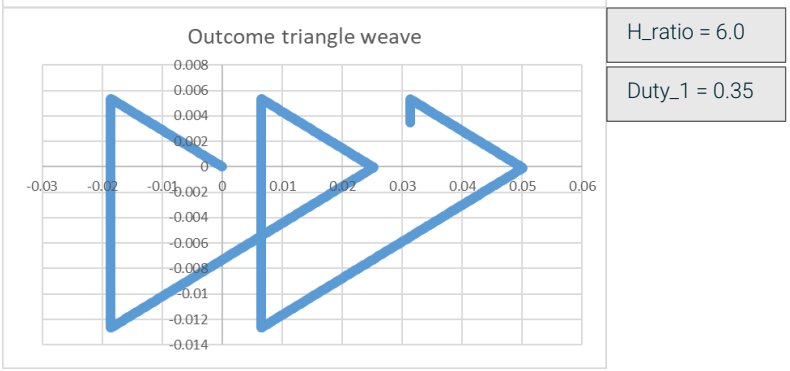
Figure 37 Z 字摆动配置

6.1.3 三角摆动

三角摆动配置，如图所示 Figure 36

名称	功能	范围	单位
摆动频率	摆动的频率，值越大则单位时间内摆动往复次数越多。	0.2 - 4	Hz
幅度	单侧摆动幅度	1 – 20	mm
速度	轨迹矢量的 TCP 速度	1 – 20	mm/sec
Duty_1	调整摆动左右方向的形状，等于 0.25 时左右对称。	0.1 - 0.4	
H_ratio	调整摆动前进方向的形状，数值越大，叠加越多，摆动得越密集。	1.0 - 6.0	





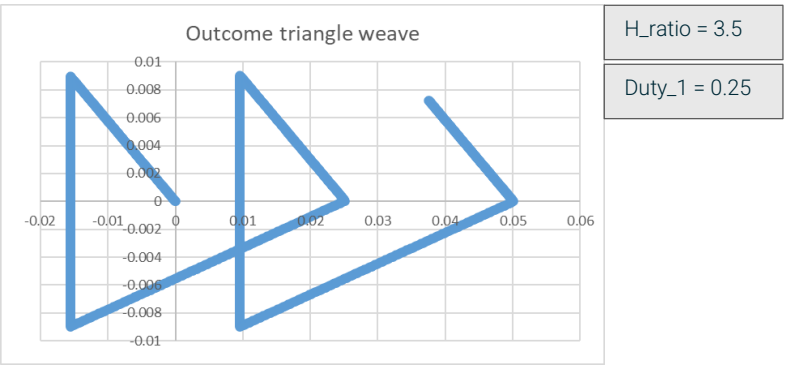


Figure 38 三角摆动配置

6.1.4 正弦摆动

正弦摆动如 Figure 39 所示。

名称	功能	范围	单位
摆动频率	摆动的频率，值越大则单位时间内摆动往复次数越多。推荐 1.5	0.2 – 4	Hz
幅度	单侧摆动幅度。推荐 3	1 - 20	mm
左等待	TCP 在左侧停留时间。推荐 0.02	0.02 - 1/(频率*2)	sec
右等待	TCP 在右侧停留时间。推荐 0.02	0.02 - 1/(频率*2)	sec

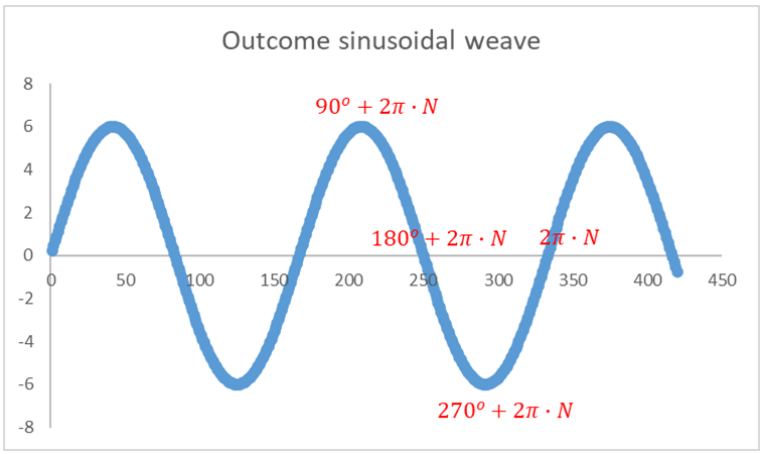
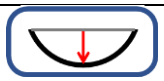


Figure 39 正弦摆动配置

6.1.5 时钟摆动

时钟摆动设置如 Figure 40 和 Figure 41 所示。

名称	功能	范围	单位
摆动频率		0.2 - 4	Hz

	摆动的频率[字母 *f* 代表]，值越大则单位时间内摆动往复次数越多。摆动周期为 f 倒数+左等待+右等待。		
幅度	单侧摆动幅度。	1 – 20	mm
左等待	TCP 在左侧停留时间	0.02 - 1.0	sec
右等待	TCP 在右侧停留时间	0.02 - 1.0	sec
弦深	 摆动圆弧的弦深	最大值等于幅度设定值的一半。例如：幅度设定值为 6，弦深范围为[0, 3]	mm

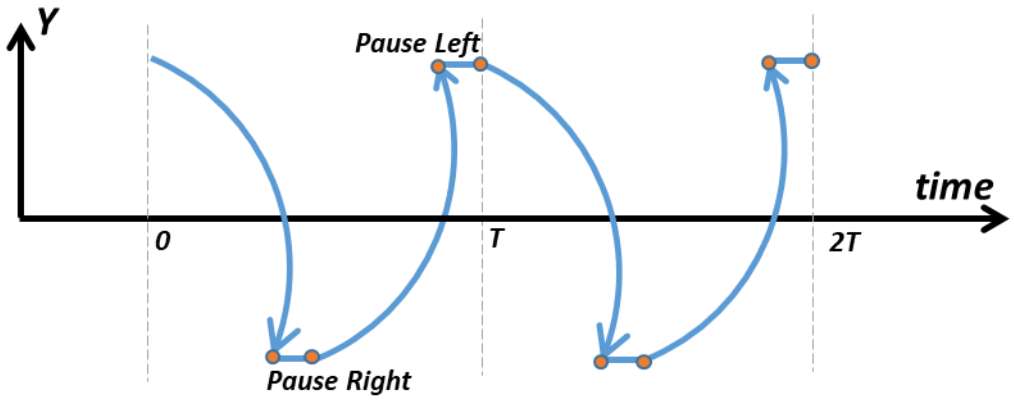


Figure 40 时钟摆动 Y-时间变化关系

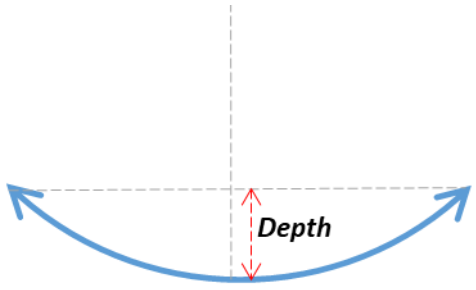


Figure 41 时钟摆动 Y-Z 变化关系

6.2 全局变量 dl_angle_RO_ext, dl_offset_RW_ext, dl_weave_enable_RO_ext,

dl_angle_RO_ext 是一个[只读, 实数, 单位: 弧度]的全局变量，它实时更新摆动正弦波形中的相位，如图所示 Figure 39，它对于一些工艺技术如焊接的弧压跟踪将会非常有用。

dl_weave_enable_RO_ext [只读, 布尔量] 用来指示 path_offset 功能被激活状态。

全局变量 *dl_offset_RW_ext* [可读可写, [x, y, z, rx, ry, rz], 单位[m, m, m, rad, rad, rad]]
用来接受第三方设备的偏移数据，例如焊接中的激光跟踪设备需要实时对机器人输入偏移量，在 *dl_weave_enable_RO_ext* 为 True 的情况下可以写入这个变量进行偏移。

7. 报警信息

类别	序号	描述
报警	WNG001	摆动形状需要选择
报警	WNG002	Reserved.
报警	WNG003	Reserved.
报警	WNG004	左等待和右等待时间之和太大。
错误	Error001	仅对于 Z 字摆动，左右等待时间太长。
错误	Error002	对于三角摆动，摆动频率太小或速度太快导致无法达到期待的摆动形状。
错误	Error003	编译错误：名称 'Thread_XXXX_han'，这是由于 WeaveON and WeaveOFF 中摆动形状选择不一致。
错误	Error004	路径偏移：没有激活，有多种原因可导致这个结果，其中最可能的一种情况时交融半径设置过大。最可靠的一个建议时交融半径的值不超过 2mm。
错误	Error005	WeaveON 中设定的速度和机器人实际速度偏差过大。
错误	Error007	弦深值过大。
错误	编译错误	编译错误 名称 xxxxx 没有定义， 这可能是由于安装界面的授权未激活导致。

8. 脚本函数信息

8.1 函数表

名称	描述	依赖	版本
weave_circles (weave_para_st)	Calculate circles type offset in realtime	weave_circles_init.script mod.script weave_para_structure	1.0.0
weave_triangle(weave_para_st)	Triangle type offset generator	weave_triangle_init.script mod.script weave_para_structure	1.0.0
weave_zigzag(weave_para_st)	Zigzag type offset generator	weave_zigzag_init.script mod.script weave_para_structure	1.0.0
mod(number, divisor)	Re-implementation of complementation calculation	null	1.0.0

Attention:
All global variables used in this urcap scope follow a format of 'dl_xxxx'.
To avoid interference in program, change another format.

User can call URCap function via ExpressionEditor to customized own Robot Program, shown in Figure 21.

weave_circles_init(weave_para_st)	Circle initialization	weave_para_structure	1.0.0
weave_zigzag_init()	Zigzag initialization		1.0.0
weave_triangle_init(weave_para_st)	Triangle initialization	weave_para_structure	1.0.0
weave_para_structure	Weave parameters definition	null	1.0.0

8.2 摆动参数数据格式

```
# ADDRESS           DESCRIPTION                TYPE          INITIAL_VALUE        UNIT
# 0-9 are general parameters
# weave_para_st[0]  ctrl_frequence             integer       500                  Hz
# weave_para_st[1]  weave_frequence            integer       1                    Hz
# weave_para_st[2]  amplitude[maximum 0.006]   real          0.005                M
# weave_para_st[3]  pi                         real          3.14159              na
# weave_para_st[4]  velocity                   integer       0.008                M/Second
# weave_para_st[5]  path_offset_type           Integer       1                    na
# weave_para_st[6]  reserved
# weave_para_st[7]  reserved
# weave_para_st[8]  reserved
# weave_para_st[9]  reserved

# 10-19 are process related parameters
# weave_para_st[10] Orientation[0-L,1-R]       boolean       True                 na
# weave_para_st[11] TimeStamp_ini              real          0                    Second
# weave_para_st[12] Angle_ini                  real          0                    Degree
# weave_para_st[13] x_TCP_ini                  real          0                    M
# weave_para_st[14] y_TCP_ini                  real          0                    M
# weave_para_st[15] Dwell_left[minimum 0.02]   real          0.2                  Second
# weave_para_st[16] Dwell_right[minimum 0.02]  real          0.2                  Second
# weave_para_st[17] duty_1                     real          0.25                 M/Second
# weave_para_st[18] Tolerence                  real          0.002                Second
# weave_para_st[19] H_ratio[1~6]               real          6                    na
```

Figure 42　摆动参数数据格式

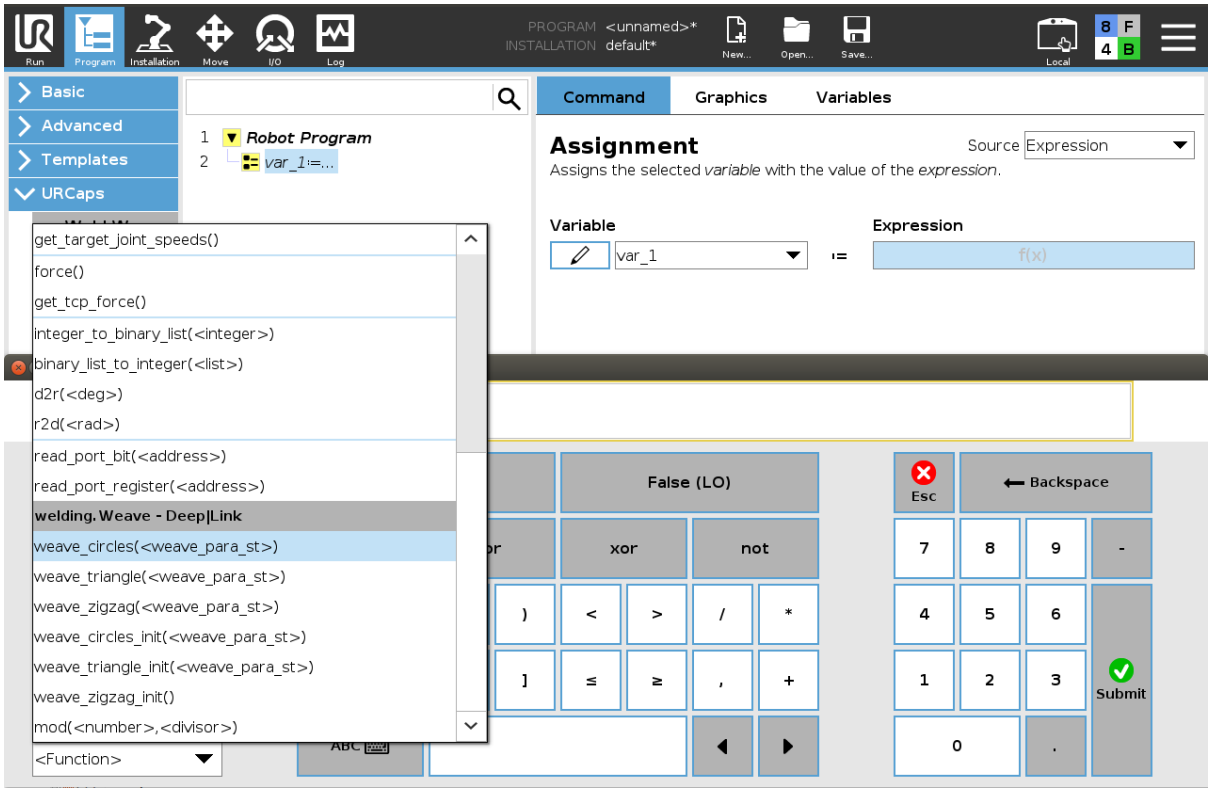


Figure 43　在 ExpressionEditor 中调用 Deep-Link 开放的函数。