

Standard Bots

Programming Manual



Standard Bots Programming Manual

Standard Bots

Revised August 28, 2026

Table of Contents

1	Introduction	3
1.1	Welcome!	3
1.2	Contacting Standard Bots Support	3
1.3	Intended Use & Limitations	3
1.4	Warnings & Risks	4
1.4.1	Introduction	4
1.4.2	Installation	4
1.4.3	Operator Safety	5
1.4.4	Transportation	5
1.4.5	Cybersecurity	5
2	Safety	7
2.1	Overview of Safety Features	7
2.1.1	Response Time	8
2.2	Safety Settings	9
2.2.1	Speed Limits	9
2.2.2	Speed Modifier % Slider	11
2.2.3	Customizing Speed Limits For A Step	12
2.3	Safety I/O	14
2.3.1	Configuring Safety Inputs	15
2.3.2	Configuring Safety Outputs	16
2.4	Collisions & Protective Stops	17
2.4.1	Adjusting Collision Sensitivity	17
2.5	Setting the Robot's Payload Mass	17
3	Software Overview	19
3.1	Connecting to the Robot	19
3.1.1	Connecting Via iPad	19
3.1.2	Network Settings Page	19
3.1.3	Home Page	21
3.1.4	Footer	21
3.2	Jogging the Robot	23
3.2.1	Jogging Tooltip	23
3.2.2	Jogging Joints	24
3.2.3	Entering Exact Values	25
3.3	Managing Equipment	26
3.3.1	Custom Base/ Mounting Angles	30
3.3.2	Vertical Lift	31
3.4	Robot Settings	33
3.5	Singularities	34
3.6	Routines	35
3.6.1	Routine Creation Example	35
3.6.2	Running Routines	48
3.6.3	Accessing the Routine's Schema	53

3.6.4	Loop Settings	55
3.6.5	Waypoints	55
3.6.6	Stop Early If	57
3.7	Advanced Routine Functionality	58
3.7.1	Variables	58
3.7.2	Javascript	60
3.7.3	Grid Position Lists	62
3.7.4	Freeform Position Lists	65
3.7.5	Push Mode	68
3.7.6	Prompt User	70
3.7.7	Decision Logic	74
3.7.8	Machine Tending Application	75
3.8	External IP Communication	87
3.8.1	Haas Ethernet Integration	87
3.8.2	Modbus/TCP Client	93
3.8.3	Modbus Routine Examples	100
3.9	Global Space & Variables	103
3.9.1	Introduction to Global Space Items.	103
3.9.2	Creating a Global Space	103
3.9.3	Creating a Global Variable	105
3.10	Exporting/Importing Routines	107
3.10.1	Introduction	107
3.10.2	Exporting Routines	107
3.10.3	Importing Routines	109
3.11	Updating Your Routine to Use V2 Move Steps	112
3.12	Recovery Mode	114
3.13	Using Run in Background steps in Routines	117
3.14	Palletizing with pallet base and boxes	118
4	Vision	129
4.1	Locate	130
4.1.1	Use Cases	130
4.1.2	Setup	131
4.2	Classify	144
4.2.1	Use Cases	144
4.2.2	Setup	144
5	External Control Overview:	149
5.1	Configuring EtherNet/IP Communication	152
5.1.1	Introduction	152
5.1.2	Configuration Steps	152
6	Offline Updates Overview:	165
6.0.1	Step by step	165
7	Appendix A - Error List	169
8	Appendix B - FAQ	172
9	Appendix C: How to direct wire robot	177
9.0.1	With an Ethernet Cable:	177
9.0.2	Direct Connection (using a USB-C Cable)	183

Chapter 1

Introduction

1.1 Welcome!

Congratulations on your new Standard Bots collaborative robot.

At Standard Bots, we believe robots can elevate people's lives — but only if people can use them. We created the RO1 to be an affordable robot that anyone can use, yet is still capable of tackling the toughest, most complex challenges.

This manual is meant to include everything needed to get you up and running. But if you ever need help, we're here to assist!

1.2 Contacting Standard Bots Support

- **Email:** support@standardbots.com
- **Phone:** 1-888-9-ROBOTS
- **Address:** Standard Bots, 35 Garvies Point Rd, Glen Cove, NY 11542

1.3 Intended Use & Limitations

The Standard Bots RO1 robot and control box are intended to be used in applications where products need to be picked and placed or applications where a repetitive process can be completed with compatible tooling. The RO1 robot and controller are intended to be used within the environmental constraints outlined in this manual. The RO1 robot and controller are intended to be implemented together, the robot is not intended to be used with any other controller and the controller is not intended to be used with any other robot.

The RO1 robot provides various safety settings and measures that, when implemented correctly and evaluated with a risk assessment, allow the robot to work in an unguarded collaborative environment in close proximity to humans.

The RO1 robot is not designed to work in applications outside the bounds of this manual and any implementation violating the indented use shall be deemed misuse. These applications include but are not limited to:

- Medical applications
- Applications in an explosive environment
- Applications where ingress protection above IP54 is required
- Applications with improper safety integration or where a risk assessment has not been created and evaluated
- Applications requiring a high degree of food safety

Standard Bots expressly denies any liability or expressed or implied warranty claims arising from intentional or unintentional misuse.

1.4 Warnings & Risks

1.4.1 Introduction

This manual is not a comprehensive guide to designing and installing a robot cell. This manual does not cover the selection and integration of 3rd party components that may be required to complete a safe installation. The solution must be designed according to the standards provided by the governing body in the country where the solution will be installed. Standard Bots is not responsible for ensuring the installation of the robot meets the standards for any given installation. The integrator assumes all responsibility for ensuring the robot installation meets all applicable standards and safety guidelines.

1.4.2 Installation

- Make sure to install the robot and all electrical equipment according to requirements and specifications in this manual (see Chapter 2).
- The RO1 control box should only be connected to secure networks.
- **Risk Assessment:** The RO1 is provided as a partially completed machinery. As such, the robot should only be used as part of a cell after a comprehensive risk assessment has taken place.
 - Aside from the robot itself, such a risk assessment may consider the robot's payload, pinching hazards, any toxic or hazardous substances in the cell, and risks involved with sharp end effectors.
 - Risk assessments must cover the entire scope of the integration, including components provided by other suppliers.
 - The risk assessment will help determine any speed or force limits to set up on the robot and what safety accessories (like light curtains or area scanners) can be used. These settings are covered in Chapter 4.
 - Anti-gravity (hand-guided) mode should only be used if risk assessment approves its use.

- Refer to ISO 10218-2 and ISO/TS 15066 for comprehensive guidelines.
- Only trained, instructed, and otherwise qualified personnel may conduct cleaning, maintenance and repair of the robot.

1.4.3 Operator Safety

- Always ensure that the robot is well-maintained and in good mechanical condition before operating it. Always inspect it before use.
- Don't wear jewelry or loose clothing when working with the RO1. Tie back any long hair.
- Never open the control box cabinet door during operation. Never feed wires through the door; feed them through the rubber slot on the bottom of the box.
- Don't expose the robot to strong magnetic fields. This may damage the robot or cause malfunctions.
- Depending on other noise in the environment where the robot is used, hearing protection may be required.
- Never attempt to make any repairs, adjustments, or inspections while the robot is running.
- Operators should stay alert and focused when operating the robot. Avoid distractions and keep attention on the task at hand.

1.4.4 Transportation

WARNING – Proper Transportation Required

To prevent damage during transport, the RO1 robot **must be placed in the supplied Pelican case** with all joints positioned in the **designated “box position.”** Transporting the RO1 outside of these conditions may result in damage to the robot .

1.4.5 Cybersecurity

The Standard Bots RO1 is able to connect to both wired and wireless internet in order to facilitate remote access, troubleshooting and software updates. It is imperative to follow best cybersecurity practices, as you would with any industrial device capable of being accessed online.

1. Secure communication: Make sure the communication channels used by your robot are encrypted and secure, so that unauthorized individuals cannot access the robot's data or control its actions.
2. Strong passwords: Set strong passwords for your robot's user accounts and ensure that they are changed regularly. Use a password manager to help you create and manage secure passwords.
3. Regular updates: Keep your robot's software and firmware up-to-date, as updates often contain security patches that address vulnerabilities.
4. Physical security: Ensure that physical access to the robot and tablet is restricted, and that the robot is stored in a secure location when not in use.
5. Network security: Keep your robot on a separate network or VLAN, and restrict access to that network to only authorized users.

6. Regular testing: Conduct regular security testing to identify any vulnerabilities and ensure that your robot is secure.

By following these best practices, you can help ensure the security of your robot and protect it from potential cybersecurity threats.

Chapter 2

Safety

2.1 Overview of Safety Features

The RO1 robot includes many safety features for operating in either collaborative or industrial applications. The exact implementation of these features will depend upon the application and risk assessment. The final performance level of the system will depend on the integration and must be calculated by the integrator.

The RO1 has parameters for the below settings that can be fully customized for the end application and allow for integration with a wide variety of industrial safety components.

- Joint Collision Thresholds
- Joint Velocity, Acceleration, and Torque limits
- Emergency Stop Inputs
- Safety IO / OSSD

2.1.1 Response Time

The RO1 responds to safety events within the tolerances in the below chart:

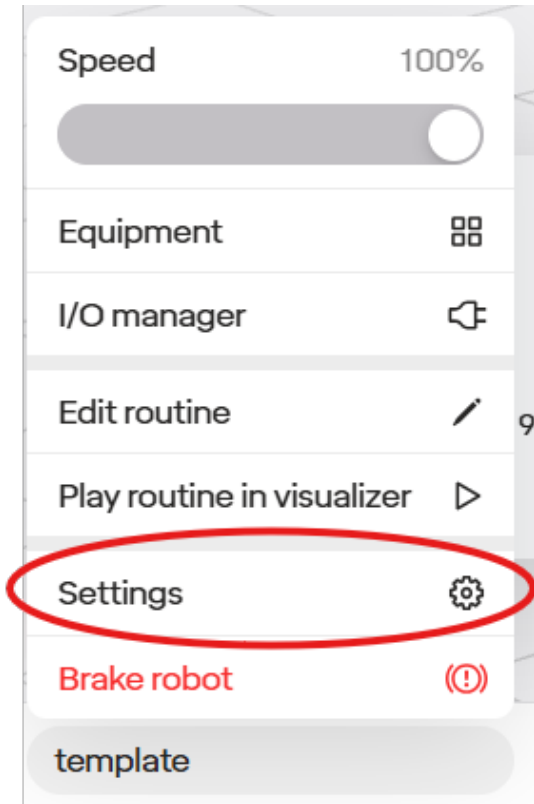
Safety Input Event	Worst Cast Detection Time	Worst Cast Power Off Time	Worst Case Response Time
Internal Emergency Stop	50ms	600ms	600ms
External Emergency Stop	50ms	600ms	600ms
External Safety IO Slow Speed Input	50ms	N/A	600ms
External Safety IO Emergency Stop	50ms	600ms	600ms
Tablet or Browser E-Stop	1000ms*	1600ms*	1600ms*

*Network Latency Dependent

Safety Output Event	Worst Case Response Time
Robot E-Stop	50ms

2.2 Safety Settings

Safety Settings can be accessed by tapping the robot menu, then Settings > Safety.



Settings are locked by default; click the “Unlock” button and re-enter the robot PIN to make the settings here editable.

Safety settings do not take effect until you tap “Apply Settings.”

2.2.1 Speed Limits

How Limits Work

The RO1 has several levels of limits; the robot is always gated to the **lowest** of all of these:

- **Global Limits:** defined in Safety settings, which affect tooltip speed and the acceleration, velocity, and torque of joints.
- **Speed Modifier % Slider:** the robot can be slowed down on an as-needed basis (for instance, to trial a routine) without requiring access to Safety Settings.
- **Step-Level Limits:** the robot can also apply an alternate set of speed limits only for the duration of a specific step.

All of these settings are described below.

Editing Global Limits

The first tab of safety settings allows setting limits on the speed and acceleration of both the tooltip and the joints of the robot.

Using Factory Presets By default, you will be presented with the choice of several presets.

The “default” (middle) setting restricts the robot to a tooltip speed of 0.75m/s, which is safe for most collaborative settings. Several others are available; if there are factors that may require more caution (such as the robot’s payload or end effectors), it may be desirable to select a more conservative setting.

Remember that many factors can affect whether a given speed is safe. Before putting the robot into production, you should always conduct a full safety assessment per ISO 10218-2 to determine the proper values for these settings and any other mitigations required.

The screenshot shows the 'Safety Settings' window with the 'Limits' tab selected. At the top, there is a warning: 'Remember to conduct a full safety assessment per ISO 10218-2 to verify limits and safeguards required for application.' Below this is a toggle for 'Use Factory Preset' which is currently turned on. A horizontal slider is shown with four positions: 'Safest' (blue), 'Safer' (green), 'Default' (yellow, with a white diamond marker), and 'Faster' (orange). Below the slider, a note states: 'Default: Best for collaborative or hybrid settings where humans are trained and aware of the robot and the environment is free of sharp or dangerous objects.' At the bottom, there are three buttons: 'Tooltip', 'Full Speed', and 'Slow Speed'. The 'Last Updated' timestamp is '11/21/2023, 8:59:34 AM'. At the bottom right are 'Cancel' and 'Apply Settings' buttons.

To review the values set for these factory presets, you can scroll down.

Custom Limits To edit these limits, switch off “Use Factory Preset”; the limits should then become editable:

The screenshot shows the 'Safety Settings' window with the 'Limits' tab selected. The 'Use Factory Preset' toggle is turned off. The settings are organized into two sections: 'Tooltip' and 'JO'. Each section has two columns: 'Full Speed' and 'Slow Speed'. The 'Tooltip' section shows 'Max Velocity' with values of 0.75 m/s (Full Speed) and 0.5 m/s (Slow Speed), with a 66% usage indicator. The 'JO' section shows 'Max Velocity' with values of 4.7 rad/s (Full Speed) and 2.35 rad/s (Slow Speed), with a 50% usage indicator. 'Max Acceleration' is set to 1 rad/s² for both speeds, with a 100% usage indicator. 'Max Torque' is set to 346 N-m (Full Speed) and 100 N-m (Slow Speed), with a 28% usage indicator. The 'Last Updated' timestamp is '11/21/2023, 8:59:34 AM'. At the bottom right are 'Cancel' and 'Apply Settings' buttons.

The following can be customized, both for when the robot operates at its full speed and in a “slow”

mode that can be triggered via Safety I/O settings:

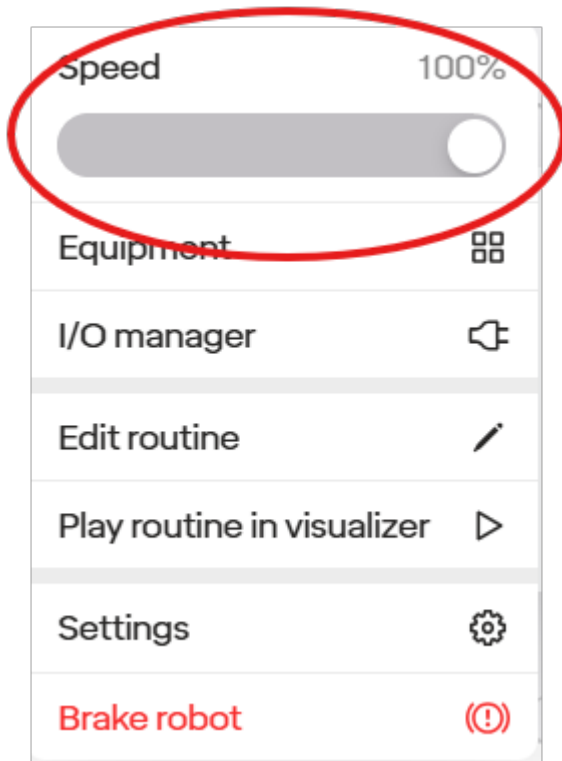
- **Tooltip Speed:** This limits the speed of the tooltip in Cartesian space.
- **Per joint:**
 - **Max velocity:** This is the maximum angular change in the joint's position per second permitted.
 - **Max acceleration:** This affects the rate at which the joint can change speed. If this is too low, the may not be able to achieve the maximum velocity otherwise permitted during the course of a movement.
 - **Max torque:** The maximum torque that the motor in the joint can exert.

2.2.2 Speed Modifier % Slider

The robot can be slowed down from its maximum as needed inside the robot menu.

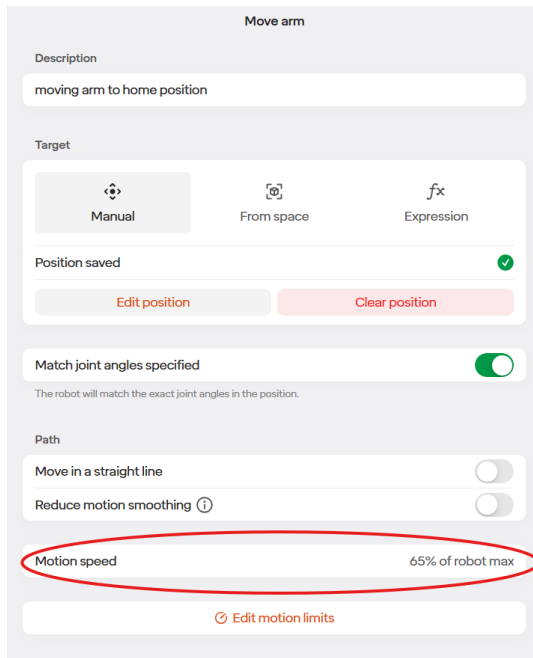
This is useful for trying out a routine before putting it into production.

Setting this slider only applies a cap to the maximum limits that would otherwise be set in Safety Settings; movements that were already slower than that are not affected.



2.2.3 Customizing Speed Limits For A Step

When editing a “Move” step, the “Motion Speed” shows the speed of this specific step. This is useful to notice for delicate movements or when required for safety. To change the speed, tap the “Edit Motion Limits” icon.



The screenshot shows the configuration interface for a "Move arm" step. The "Description" field contains "moving arm to home position". The "Target" section has three options: "Manual" (selected), "From space", and "Expression". Below the target options, it says "Position saved" with a green checkmark. There are two buttons: "Edit position" and "Clear position". A toggle switch for "Match joint angles specified" is turned on. Under the "Path" section, there are two toggle switches: "Move in a straight line" and "Reduce motion smoothing", both turned off. At the bottom, the "Motion speed" is set to "65% of robot max", which is circled in red. Below this is a button labeled "Edit motion limits".

By tapping “Edit Motion Limits”, several options are available. The motion can be restricted to percentage of the maximum (as with the global speed modifier in the robot menu), or to a new set of custom limits.

Note that these limits apply whether the robot is operating at full speed or at any “slow” speed mode defined in safety settings. The robot will always apply the lowest of all limits in effect. It is also not possible to set torque limits on a per-step basis.

Motion Limits

×

What limit should apply to the robot's motion speed during this step?

- ☐ Automatically choose motion speed (recommended)
No specific limit for this step
- ☒ Limit to percentage of robot maximum
All maximums otherwise specified in Safety Settings will be capped to this percentage

100%  100%

- ☐ Fully customize limits for tooltip and all joints
Custom limits entered here will still be gated by the top-level robot Safety Settings.

Tooltip	Limit		
Max Velocity	0.75	m/s	 100%
Joint 0			
Limit			
Max Velocity	4.7	rad/s	 100%
Max Acceleration	1	rad/s ²	 100%
Joint 1			
Limit			
Max Velocity	4.7	rad/s	 100%
Max Acceleration	1	rad/s ²	 100%
Joint 2			
Limit			
Max Velocity	5	rad/s	 100%

Save Changes

Cancel

2.3 Safety I/O

In the safety Settings there are two tabs for I/O, Safety Inputs and Safety Outputs. Safety inputs allows you to tie in devices such as area scanners and key switches to discrete inputs onto your control box. Safety outputs allows the robot to indicate safety-critical elements of it's status over discrete outputs to other devices like PLCs and other moving equipment that should stop when the robot does.

2.3.1 Configuring Safety Inputs

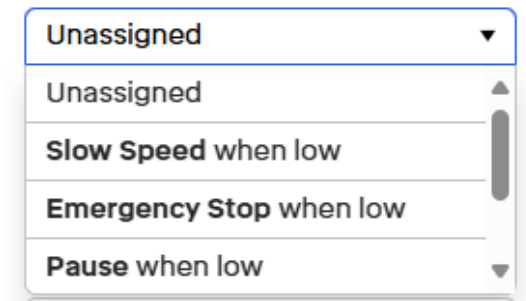
The sixteen 24V inputs on the control box are all safety-rated and can be used in pairs. Safety devices are connected in pairs to eliminate the possibility of a stray signal keeping the robot operating while actually in an unsafe condition. Therefore, if either of the paired inputs is below 24V, it will be treated as a “low” signal.

The screenshot shows the 'Safety Settings' window with the 'Inputs' tab selected. It includes a descriptive paragraph about safety inputs, a toggle for 'Teach Pendant Safety Inputs', and a table for 'Assignable Inputs'. The table shows two input pairs: 'IN 1 + IN 2' and 'IN 3 + IN 4', both currently set to 'Low'. The 'IN 3 + IN 4' pair has a dropdown menu set to 'Slow Speed when low' and an 'Auto-Reset' checkbox. At the bottom, there is a 'Last Updated' timestamp and 'Cancel' and 'Apply Settings' buttons.

Assignable Inputs	Current Value	Safety Function Assignment
IN 1 + IN 2	Low	Assigned to: External control
IN 3 + IN 4	Low	Slow Speed when low ☐ Auto-Reset

The following options are available:

- **Safety Function Assignment:** Any ports can be assigned to:
 - **Emergency Stop** when low: Triggers a category-0 stop that brakes robot and cuts power to arm.
 - **Pause** when low or when high: Pauses the currently-running routine but does not brake the robot.
 - **Slow Speed** when low: Makes the robot observe the “Slow Speed” set of limits defined in the “Limits” tab. This is useful, for instance, with an area scanner, to force the robot to move at a collaborative speed when people are nearby.
 - **Reset Safeguards When High:** This will reset the effects of any safeguard that does not have auto-reset (see below). This has no effect on emergency stop.
- **Auto-Reset:** This controls whether the effect of the function, once triggered, should go away (i.e. when we move back from low to high) or whether it requires a separate, explicit Reset. This is useful, for instance, with a sensor on a door to a fence around the robot: one would not want the robot to re-engage or speed up if the door is closed behind someone.



2.3.2 Configuring Safety Outputs

Additionally, the sixteen outputs on the control box are all safety-rated and are also used in pairs to eliminate the possibility of a stray signal keeping the robot operating while actually in an unsafe condition.

Safety Settings

Unlock

Limits
Inputs
Outputs
Collisions

The robot can indicate safety-critical elements of its status over its discrete outputs for use with devices like safety PLCs and other moving equipment (like 7th axis devices) that should stop when the robot does. All outputs on the control box are safety-rated and those not assigned here can be used for other purposes.

Assignable Outputs	Current Value	Safety Function Assignment
OUT 1 + OUT 2	Low	Unassigned
OUT 3 + OUT 4	Low	Unassigned
OUT 5 + OUT 6	Low	Unassigned

Last Updated: 11/24/2025, 3:01:39 PM

Cancel
Apply Settings

The following options are available:

- Emergency Stop
- Reduced Speed mode
- Full Speed mode
- Robot is moving
- Robot is still

2.4 Collisions & Protective Stops

When the robot is running a routine and encounters a collision, it will stop. The routine can be re-started with the “Play” button in the top toolbar. Every time a collision occurs, it will also log the time and force measured, which can be reviewed in the Notification (bell) section.

Collisions are detected with two methods: 1) by monitoring the current coming out of the joint motor to measure torque and find discrepancies, 2) by monitoring an IMU (accelerometer) inside each joint.

2.4.1 Adjusting Collision Sensitivity

Depending on the environment and safety requirements, it may be desirable to change the sensitivity of collision detection. This can be done in the “Collisions” section of Safety IO.

This allows adjusting the thresholds used for both methods: the torque shock threshold (in newton-meters) and the acceleration threshold used by the IMU (in m/s^2).

Safety Settings

Limits I/O **Collisions**

ⓘ Remember to conduct a full safety assessment per ISO 10218-2 to verify limits and safeguards required for application.

Collision Detection

The robot will trigger a protective stop when it detects a collision, either based on an unexpected change in torque or acceleration, as measured in each joint. Adjust the settings below to change the sensitivity of collision detection.

Joint	Torque Shock Threshold
J0	90 N·m - +
J1	100 N·m - +

Last Updated: 6/22/2023, 2:46:58 PM

Cancel Apply Settings

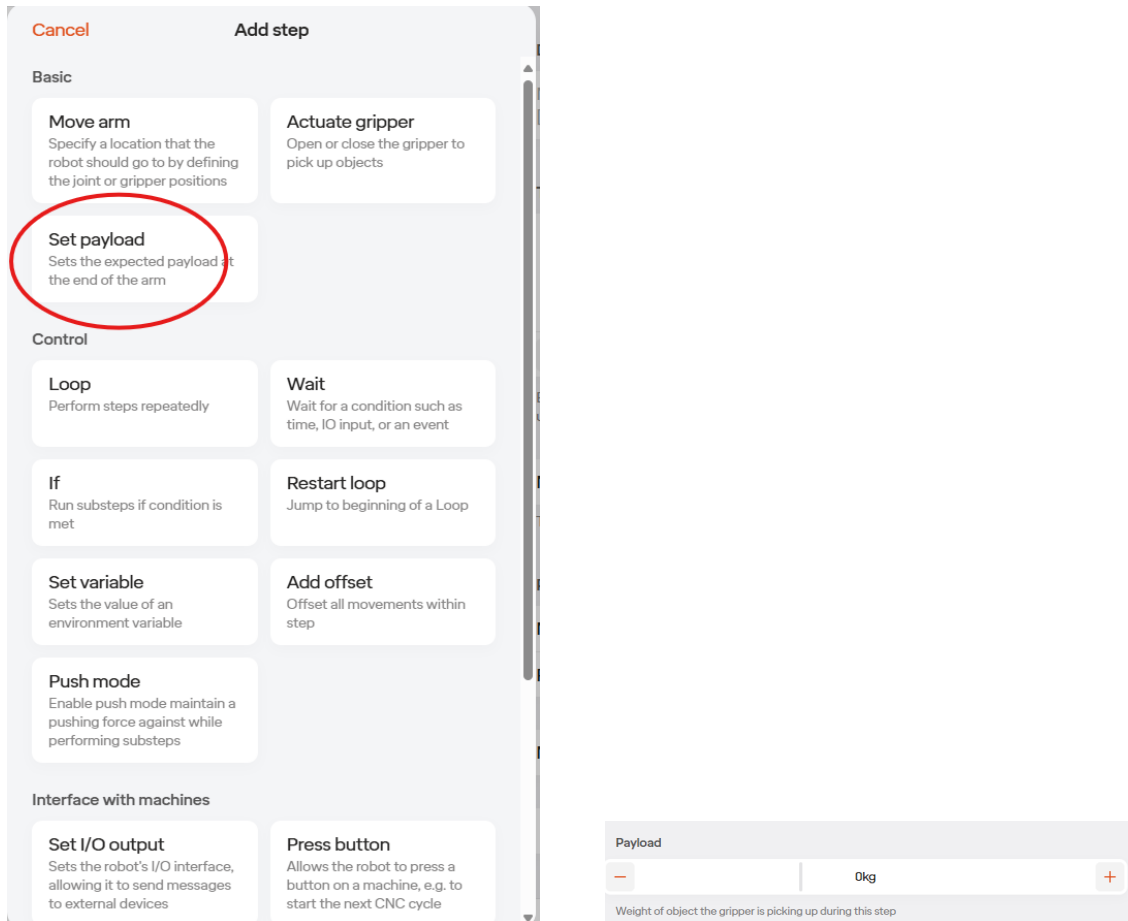
2.5 Setting the Robot’s Payload Mass

The robot’s knowledge of its payload mass affects its ability to balance and sense collisions. It’s important for overall safety to configure the payload as part of setting up a robot cell with the RO1.

There are two ways to set the payload:

1. By adding a **Set Payload** step to a routine
2. By setting the **payload** parameter within an **Actuate Gripper** step. This is useful for situations where the payload changes as a result of the gripper.

The payload mass entered should account not only for the mass of the payload itself but also that of any attached end effectors. The spec sheet for the end effector should include this information.



Chapter 3

Software Overview

3.1 Connecting to the Robot

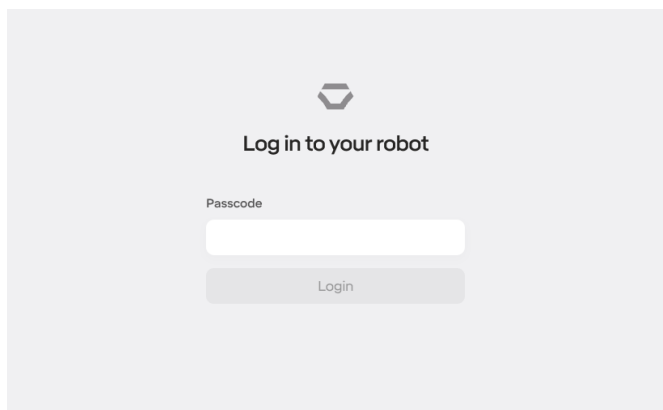
3.1.1 Connecting Via iPad

The included iPad can be connected to multiple RO1 robots (if needed) after pairing with them.

The home screen of the Standard Bots app shows the list of paired robots and their connection status. Tapping on the “...” button for a robot allows you to: * delete that robot * re-configure it (i.e. to change its network settings, PIN, etc) * show a details screen that can allow you to troubleshoot connectivity issues.

If a robot is powered on but not showing up here, after confirming the tablet is on the same network, check its details and contact customer support.

To connect to your robot, enter your PIN to log in. The default PIN is 0000 or 000000:

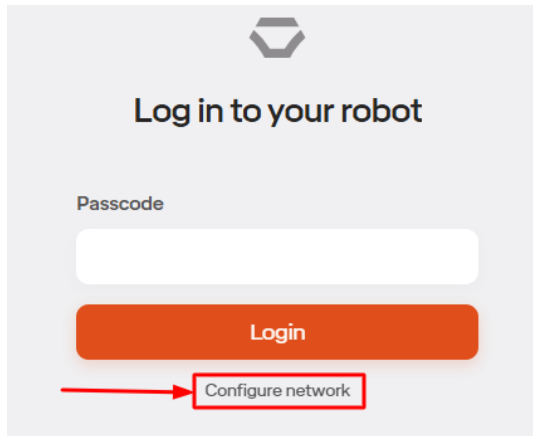


3.1.2 Network Settings Page

To change the Network settings for the robot, you can do this directly using a web browser without having to use the iPad App. You can change the settings directly via the Network Settings Page. If you

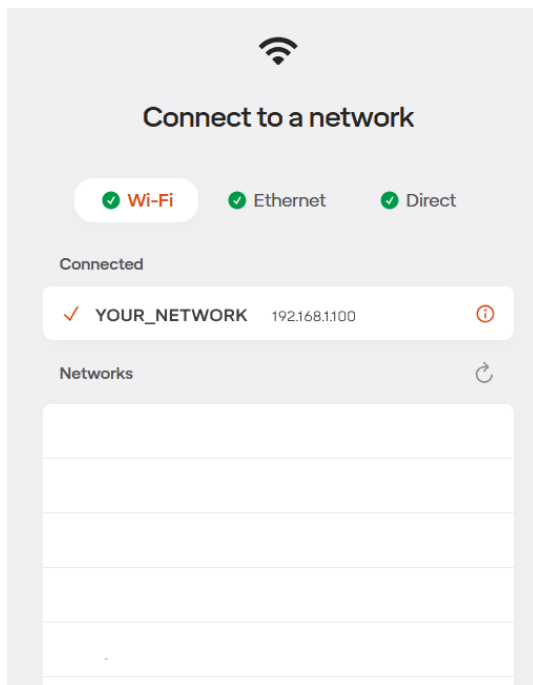
use a USB-C network adapter connected to your iPad see 12 Appendix C: How to direct wire robot.

To change the network settings, open up a web browser and connect to your robot. If you are already logged in, click Log Out at the top. Then click “Configure Network” below the Login button.



You will be asked for your passcode.

The Network Setup page will then look like this:



To enable any network check the checkbox beside it.

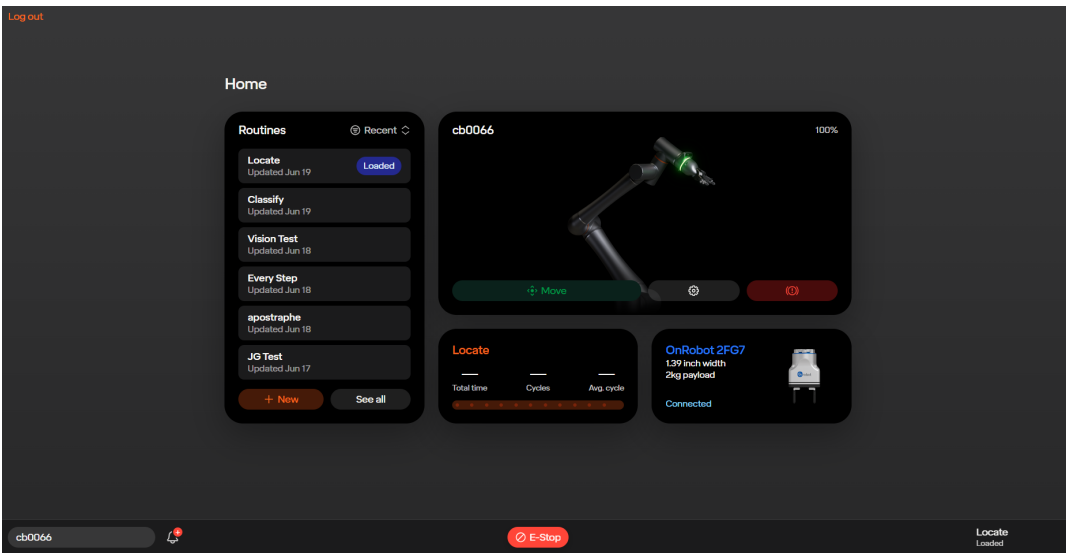
Wi-Fi - this is recommended to be your company's Wi-Fi network so that the robot can be connected to program PCs etc.

Ethernet - this is the RJ45 port on the bottom of the Control Box and is usually the machine network

Direct - this is the USB-C connection on the bottom of the Control Box

3.1.3 Home Page

The Home Page shows the routines, the current robot, and the end effector that is connected to the robot. In the “routines” tab, you are able to load, edit, and create new routines for the robot. In the tab that shows the connected robot, this is where you can go to move the robot.



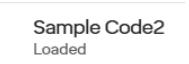
3.1.4 Footer

Once connected to an RO1, the software has a set of global controls in the footer of every page:



There are several important features worth calling out:

Element	Name	Description
	Notifications	Contains a log of failures, errors, and other notifications that have occurred when running the robot. If there are unread items here, there will be a red dot on this icon.

Element	Name	Description
	E-Stop	Trigger an emergency stop and brake the robot.
	Robot Menu	See status of current robot and access settings (see below).
	Loaded Routine	Shows what routine is loaded onto the robot.

Robot Statuses

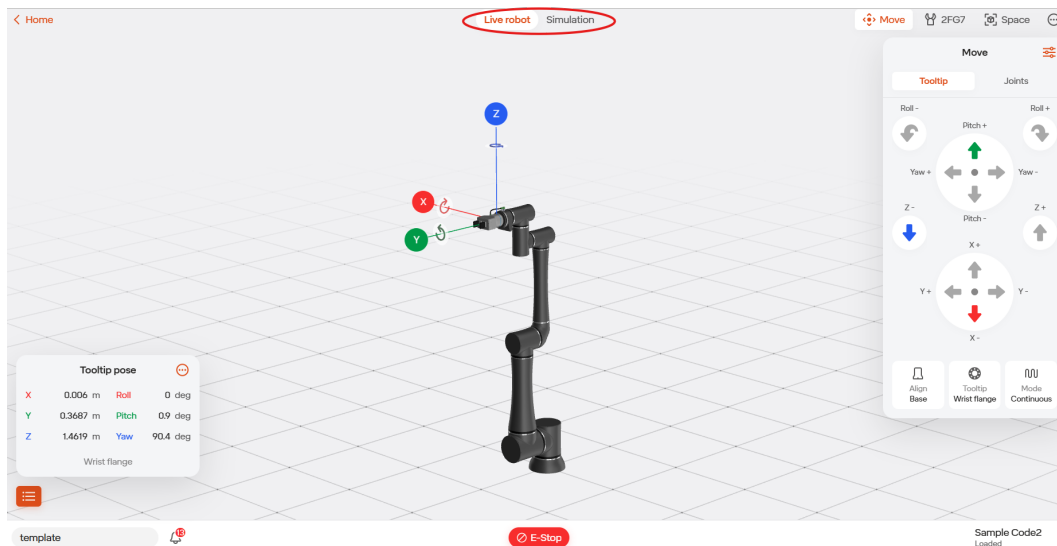
The robot menu displays the name of the robot and may show several status indicators:

Status	Description
DISCONNECTED	The tablet is not able to connect to the robot.
LOADING	The tablet is trying to connect to the robot.
MOVING	The robot is being manually jogged now.
ANTI-GRAVITY	The robot is being moved in Anti-Gravity (hand-guided) mode.
RECOVERING	The robot has exceeded joint limits and can be moved back within limits now.
RUNNING	The robot is running a routine.
PAUSED	The robot is paused in a previously-running routine. You can press 'Play' to resume it.
E-STOP	The robot is in Emergency Stop mode. Reset the E-Stop button on the control box and any external E-stop buttons.
BRAKED	The robot is braked (either from having manually braked it or from a previous E-Stop). It can be unbraked in the Move screen.
FAILED	The robot has encountered a collision or has been affected by some other hardware issue.
(blue dot)	A software update is available. Check "Software Update" under "Settings".
INSTALLING	The robot is installing a software update.

3.2 Jogging the Robot

The “Move Robot” tab brings you into the Move page.

This shows a visualization of the RO1. The switch on the top allows you to switch between visualizing (and controlling) the real robot or a simulated robot.



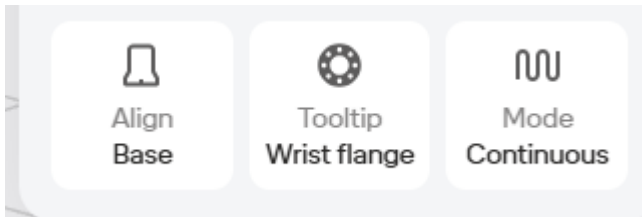
At the right of this screen are several useful tabs:

Icon	Tab Name	Description
	Move	Menu to access the “Joints” and “Tooltip” Tabs
	Joints	Controls position of individual joints.
	Gripper	Controls any connected gripper
	Space	Allows defining points and grids that can be referenced in the routine
	I/O	Shows the status of the I/O ports and allows manually toggling them. Hidden behind the three dots icon.
	Tooltip	Jogs the robot’s tooltip in Cartesian space

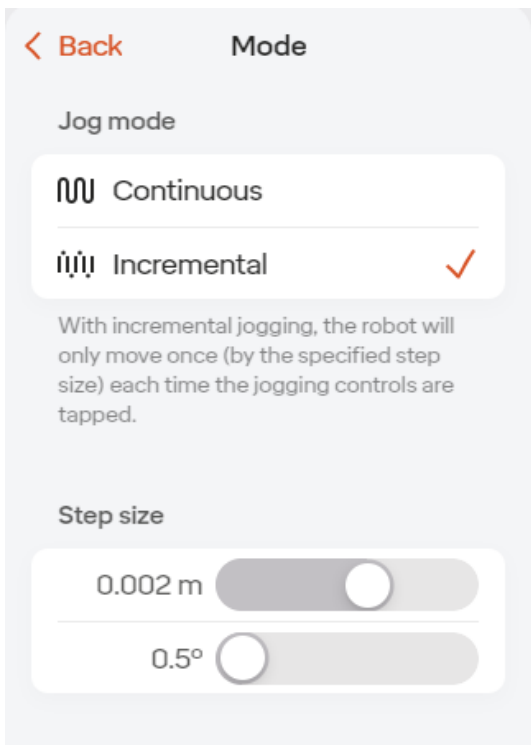
3.2.1 Jogging Tooltip

When jogging the tooltip, you are able to control the X, Y, Z coordinates of the tooltip, as well as the roll, pitch, and yaw.

This can be switched between “Base” (where the reference frame is the robot base), “Tooltip” (where the reference frame is on the robot’s tool flange) and “World” (where it is the same as the base unless you change the robot mounting angle for the base). Moving to tooltip mode can be useful for maneuvering the tooltip in and out of tight spaces.



Selecting the “Mode” icon allows you to switch between continuous and incremental jogging. With continuous jogging, holding down a move robot icon will allow the robot to move as long as the icon is held down. With incremental jogging, when tapping on a move robot icon, the robot will only move a specified distance, this can be useful when fine tuning positions.



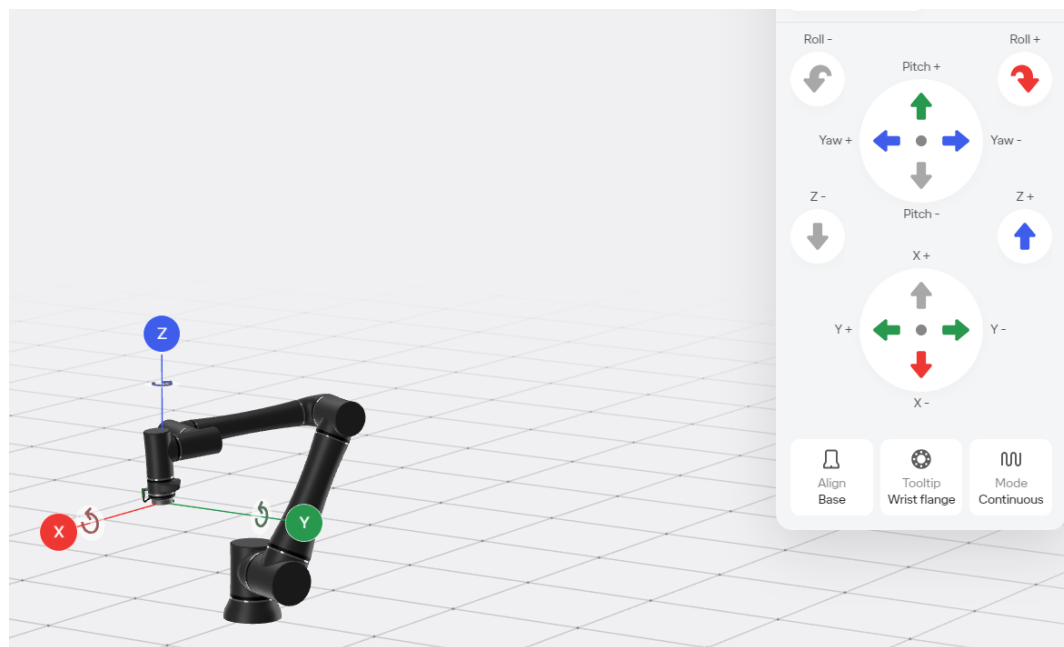
For safety, manually jogging the robot is always capped at a speed below the maximum that will apply when running. To make the robot jog even slower, you can change the global speed % slider in the robot menu.

3.2.2 Jogging Joints

The robot’s joints can also be moved directly. This can be useful in avoiding collisions and in planning moves to minimize cycle time.

When moving the robot, there could be times when the robot cannot further move in a direction when

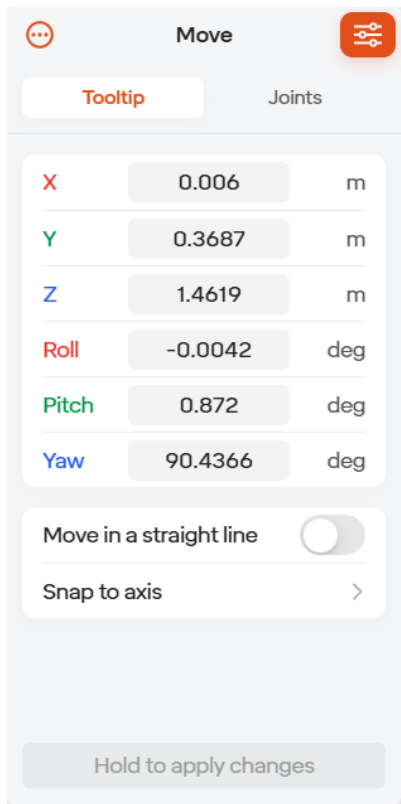
the robot is not fully extended (certain actions will be grayed out). When this happens, check your Jog Joints tab to make sure there isn't a joint that is at it's joint limit.



J0	—	-18.4	+
J1	—	-0.1	+
J2	—	87.1	+
J3	—	-359.9	+
J4	—	258.1	+
J5	—	37.9	+

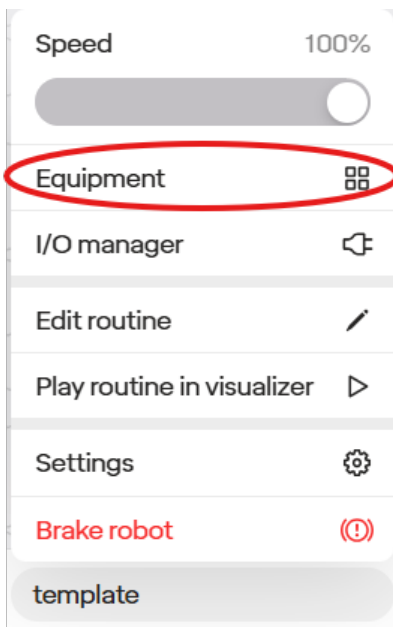
3.2.3 Entering Exact Values

When jogging in either mode, if you want to enter a more precise value, you can press the  button in the top right. Once you've entered the values you want, hold down the "Hold to Apply Changes" button at the bottom of the panel.



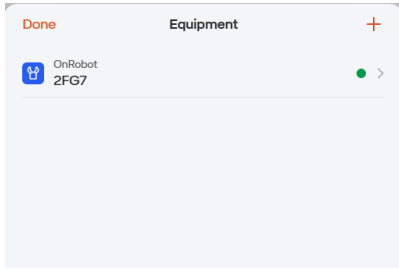
3.3 Managing Equipment

To manage equipment in the cell, including grippers, 7th axis devices, CNC machines, and more, open “Equipment” from the robot menu.

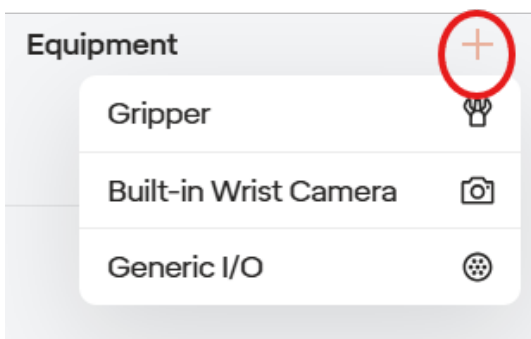


The equipment configured on the robot can be seen in the list on the left. To add new equipment, press the plus sign button in the upper right. Various settings for each connected device can be controlled

on the right.

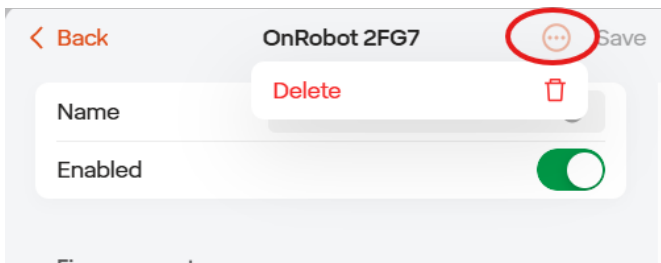


Currently, only one gripper can be added at a time (except when using the onRobot Dual Quick Changer). When adding a single tool on a single changer, you do not need to define a changer and instead simply add the tool.

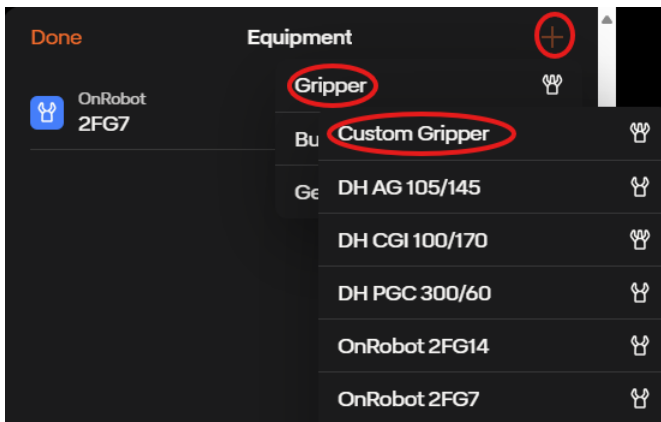


To remove equipment select the equipment that you would like to remove.

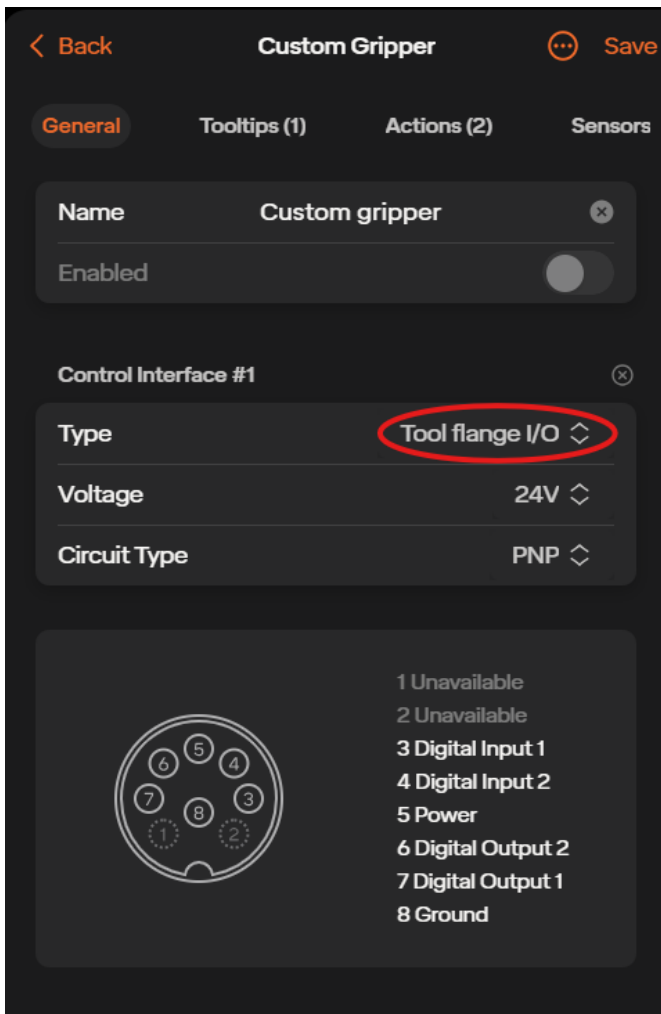
Then select the 'circle with the three dots' icon at the upper right of the tab to reveal the delete icon.



Certain m8 equipment will need access to the flange inputs and outputs. To make your customer gripper for this equipment, to go to the Equipment tab, select the robot's name in the lower left corner of the page and then select Equipment. Once here select the plus icon, then gripper, then custom gripper.

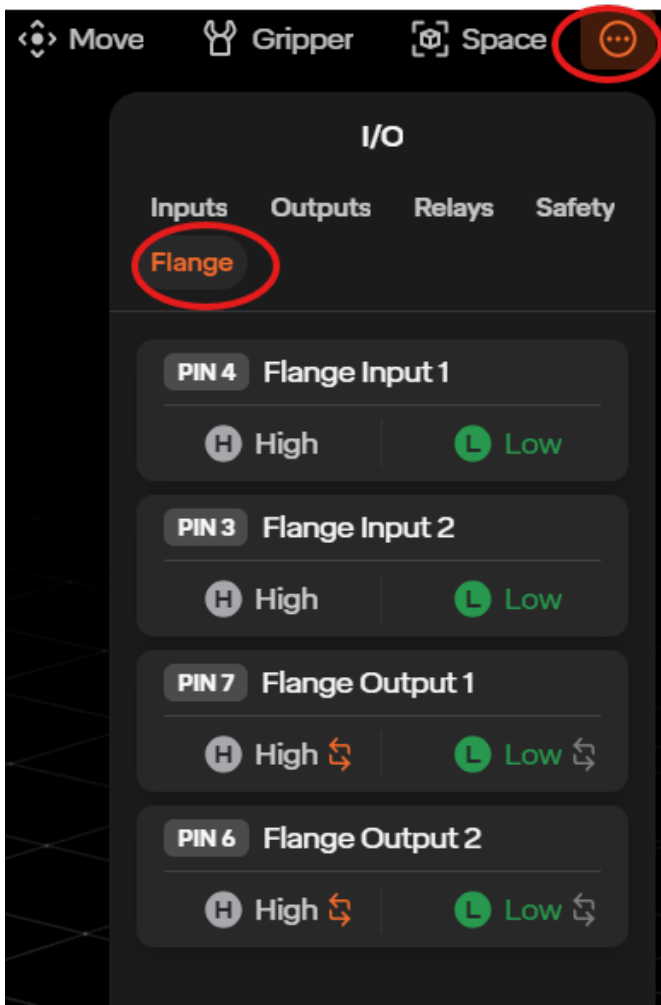


In the custom gripper page, select Add Interface, and then change the Type to Tool Flange I/O. Here you can view the pin map, change the voltage from 24V to 12V and change the circuit type from PNP to NPN

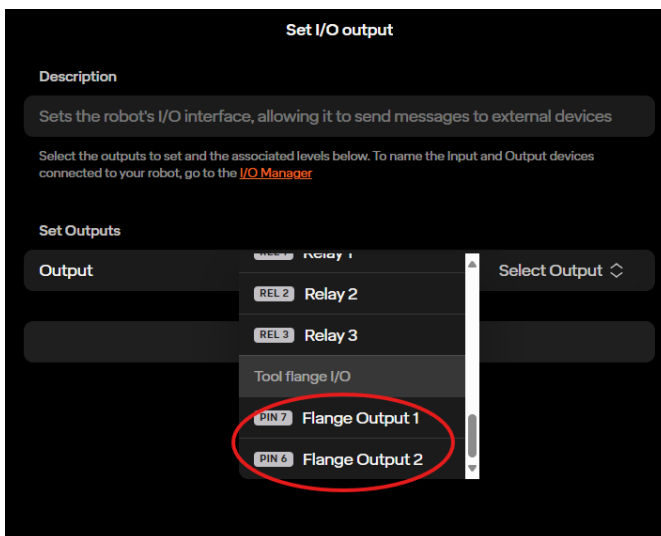


In the Tooltips tab you can add a new tooltip and adjust the TCP offset and rotation of the equipment. In the Actions tab you are able to control the actions such as actuate, reset, clamp, unclamp. You can also change the control from the control box I/O to the tool flange I/O. And in the sensors tab you can add sensors and change their input ports.

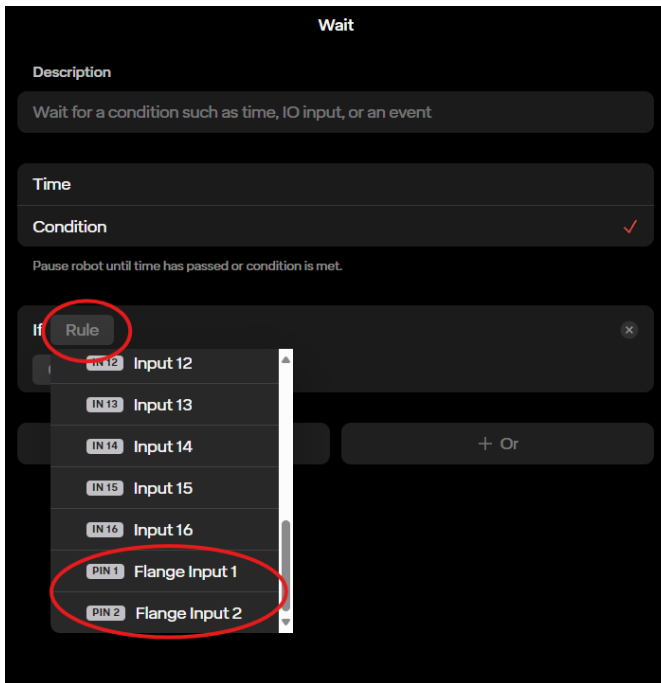
To access the flange I/O in the move robot page, go to the three dots icon in the upper right, select I/O, then select Flange.



To access the flange outputs in the routine editor page, go to Add Step, then select the Set I/O Output step. On the right side of the page there will be a list of outputs to select. At the bottom of the list will be the flange outputs.



You are able to access the flange inputs for routines as well. For example, in a wait step, select Rule, then I/O, then at the bottom of the list you can find the flange I/O. So in this wait step, the routine will wait until one of the flange I/O goes high or low.



3.3.1 Custom Base/ Mounting Angles

As of recent software versions, you are now able to add your custom base/ custom mounting angles to your robot's equipment page. If you do not see the option to create a custom base, please reach out to the Standard Bots support team.

Select the plus icon at the top right of the Equipment Page. There you will see an option called Custom Base/Mounting.

You will then be brought to the Custom Base creation tab. There you are able to change the mounting positions of the robot, as well as the rotation of the robot. You will also see some presets for the rotation. Those are upright (the standard), wall mount, ceiling mount, and custom.

< Back
Custom Base/Mounting
⋮ Save

Name Custom Base (2)
Enabled

Mounting position

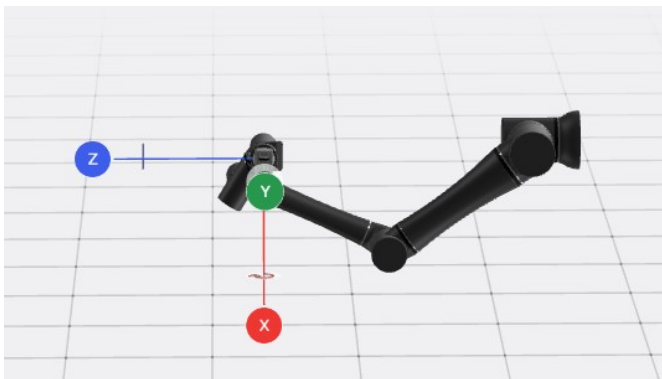
X	0	in
Y	0	in
Z	0	in

Mounting rotation

Preset	Upright	
Roll	0	deg
Pitch	0	deg
Yaw	0	deg

When done making changes, select the save icon in the upper right.

Once saved, you should see the robot model on your screen change to what you set your custom base/ mounting angles to be.



3.3.2 Vertical Lift

Step 1: Adding Lift Equipment In the UI, go to the bottom left hand corner and go into your equipment manager. Once there, select the “+” icon. Afterwards you can select the Ewellix Lift. Select the save icon and the lift will be added to your equipment list. Please reach out to support if you are unable to

add the lift to your equipment list.

<

Back

Ewellix Lift TLT

⋮

Save

Name

Lift TLT

✕

Enabled

Serial port path

/dev/ttyUSB0

✕

Stroke

700mm

⬆️⬆️

Step 2: How to use the Move Lift Step

Go to your routine and add the “Move Vertical Lift” step.

Cancel

Add step

Basic

Move arm

Specify a location that the robot should go to by defining the joint or gripper positions

Move vertical lift

Move vertical lift to position

Set payload

Sets the expected payload at the end of the arm

In the Lift step you can see the device selected as well as a target where you enter how high you would like the lift to move

Move vertical lift

Description

Move vertical lift to position

Device

Lift TLT

Target

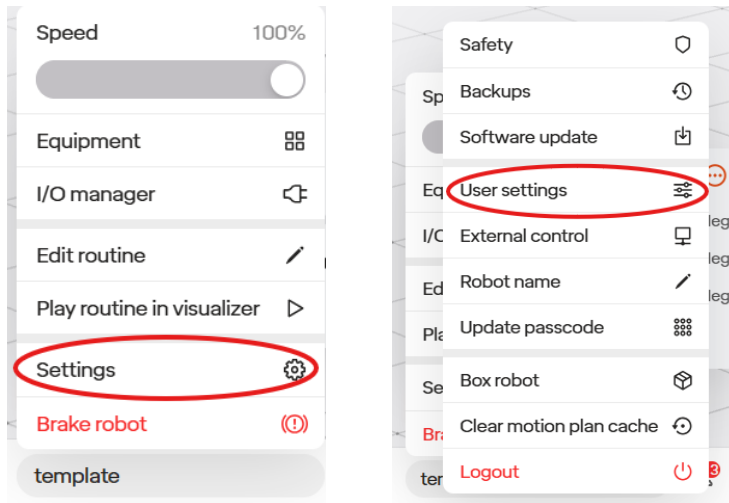
625 mm–1325 mm

fx

mm

3.4 Robot Settings

Settings for the robot are accessed via the robot menu. User interface settings are accessed separately within settings via the hamburger menu.



The following options are available on the RO1:

Section Name	Description
Safety	Speed limits, configuration of safeguard devices connected to I/O ports, and collision sensitivity settings
PIN	Update the robot's PIN
Robot Name	Name displayed in UI to distinguish it from any other RO1s you may be using.
Software Update	Allows checking for and installing any available software updates
Interface Settings	Allows switching between imperial and metric units for displaying lengths in the UI
Box Robot	Places robot into pose that allows it to fit in Standard Bots' provided foam case in case it needs to be returned.

3.5 Singularities

A singularity is a configuration in which the robot end-effector becomes blocked in certain directions. A robot is unable to maintain a constant velocity while passing one.

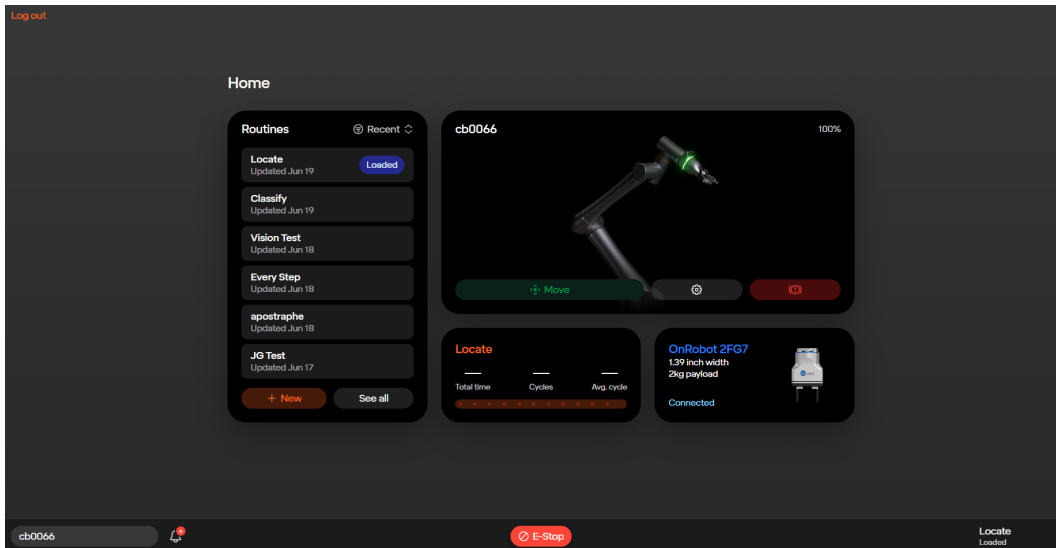
The RO1 handles this by never planning motion paths through singularities. If the robot is programmed in a manner in which it will encounter a singularity, the UI will display a “Motion Planning Failed” error and the robot will not attempt the movement.

To ensure routines are not created in which the robot will encounter a singularity, the user interface requires a play-through of any new routine in a simulation mode before running it on a real system. This ensures all motion plans are valid before running on a physical robot arm.

3.6 Routines

Standard Bots uses an intuitive “no-code” approach to programming the RO1 robot. The programs in the robots are referred to as “Routines”. The routine reads as a story. Routines can be developed and tested in simulation without moving the actual robot. The robot can store multiple routines. Routines contain all information the robot needs to complete the programmed task including moves, speeds, setting/reading IO, communicating with external equipment and more.

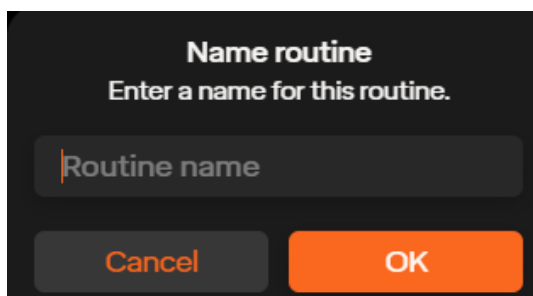
3.6.1 Routine Creation Example



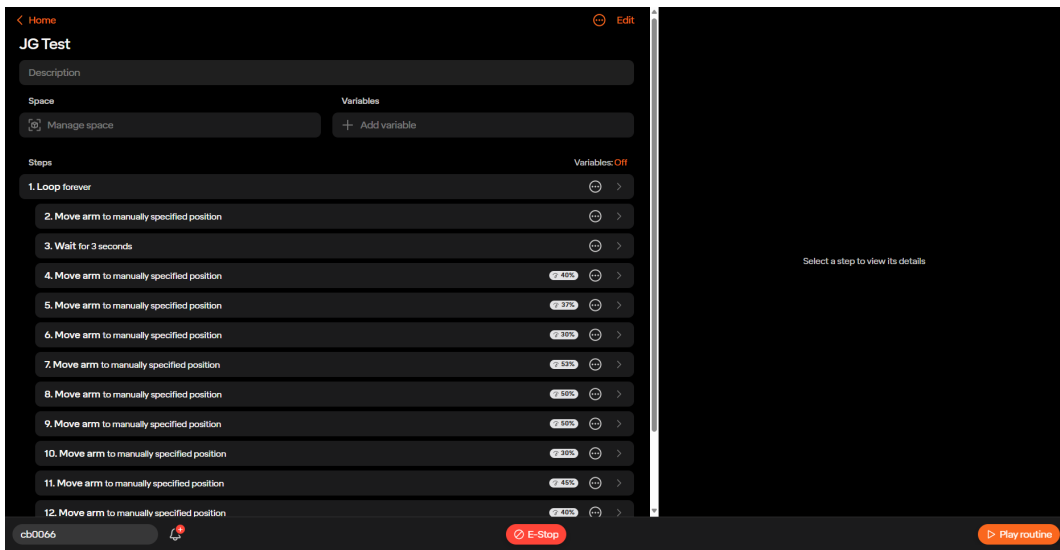
To get to the Routines area of the interface, return to the home page.

Existing Routines will be shown in the main window, and can be viewed and edited by clicking on them.

To create a new routine select New.



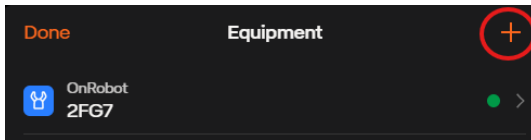
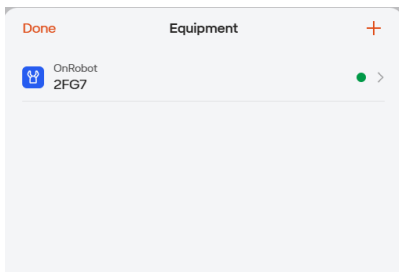
To create a new routine, give it a name under Routine Name.



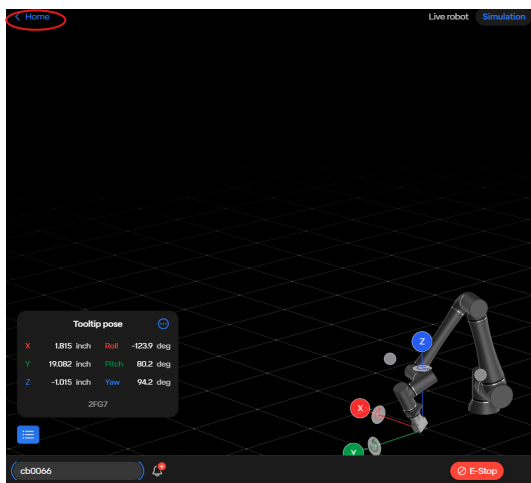
The routine interface is where you can add a description of your routine, and to edit and create steps.

- 1.Return to the **Space**.
- 2.Return to **Routine** list.
- 3.The **Routine** name.
- 4.Go to the **Space**.
- 5.Add variable to use in the **Routine**.
- 6.Lock or unlock editing.
- 7.Load this **Routine** onto the robot.
- 8.Add a new **Step** into the **Routine**.
- 9.List of **Steps** in the **Routine**
- 10.List of **Step** types that can be added to the routine.

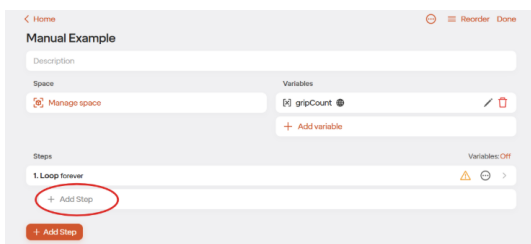
If you are using a Standard Bots-supported tool, first add it using the manage equipment button on the robot page. If you are not using a supported tool, no need to add it. It will be controlled over I/O directly.



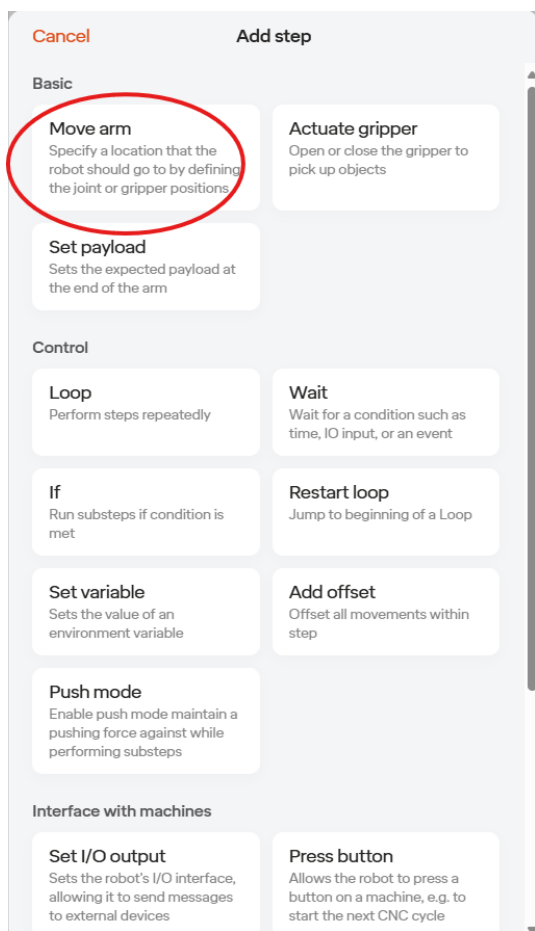
You will be directed back to the equipment tab to add the equipment. Click the +add to add the equipment. Select the equipment you have from the list. For our example, select OnRobot 2FG7. When adding a single tool on a single changer, you do not need to define a changer and instead simply add the tool.



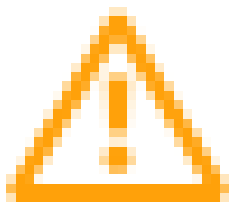
Select the back arrow in the upper left to go to the Home Page and then select on your routine to return to the Routine Page.



Click Add Step to begin creating the routine.



The menu will show all available step types with descriptions. For the sample routine, we will start with a Move Arm Step. A Move Arm step is how you tell the robot to go to a position. Click the Move Arm step to add it to the Routine.



The Move Arm step will be added to the first line of the Routine. The yellow ! Indicates that an action is required for that step. Click the step to edit the step.

Note: As of our most recent software release, the move arm step has been changed with the additions of waypoints and blending

Move arm

Description 1

Move arm through 0 segments

Targets 2

+ Add waypoint

Tap the teach button on the wrist to quickly add waypoints here.

Advanced

Default speed limits 3100% of max speed >

Default Blending 4None >

1. Editable description for this Step
2. Adding waypoints within this step
3. Changing the default speed limits for the waypoints within this step
4. Changing the default blending for the waypoint within this step

Below is the waypoint steps that are within the move arm step

Waypoint

Description 1

Waypoint to manually specified position

 2
Manual

 3
From space

 4
Expression

Position saved ✓

Edit position

Clear position

Match joint angles specified 5 ☒

The robot will match the exact joint angles in the position.

Path

Move here in a straight line

6 ☐

Stop here

7 ☒

Blending

8 None >

Stop here is required because this is the only point in the move.

1. Editable description for this step
2. Specify the position for the robot to go to based on the current position.
3. Specify the position for the robot to go to based on a saved position from the Space.
4. Specify the position for the robot to go to based on a math function and/or variables.
5. The robot can sometimes reach a point in multiple ways, this tells the robot to use this exact orientation when going to this point.
6. Moving to this position in a straight line
7. With multiple waypoints you get the option to stop at a position, or to pass through it. Note that the final waypoint step within a move arm step will always stop
8. Edit blending for this specific step

Waypoint

Description

Move the robot to a specific position in space


 Manual


 From space


 Expression

Target Select space ▾

Match joint angles specified ☒

The robot will match the exact joint angles in the position.

Path

Move here in a straight line ☐

Stop here ☒

Blending None >

*'Stop here' is required because this is the only point in the move.

For our example we are going to select From Space. Defining a point in the Space allows you to use it in multiple places in the routine. Defining a point in the Space also lets you more easily modify the point in the future.

< Home
⋮ Reorder Done

Manual Example

Description

Space


Manage space

Variables

+
Add variable

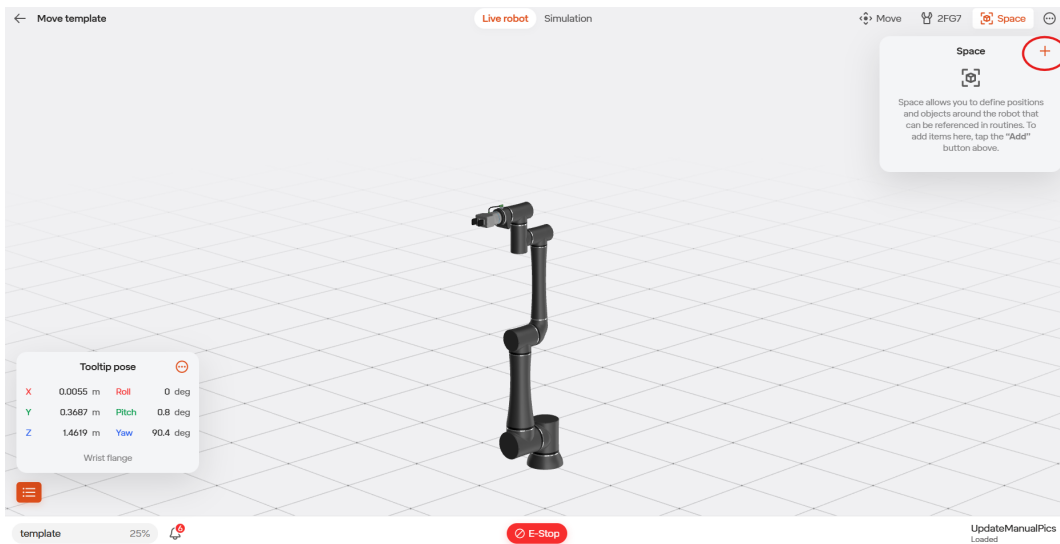
Steps Variables: Off

1. Loop forever ⋮ >
2. Move arm through 1 segment 100% ⋮ >
3. Waypoint Move the robot to a specific position in space ⚠ ⋮ >

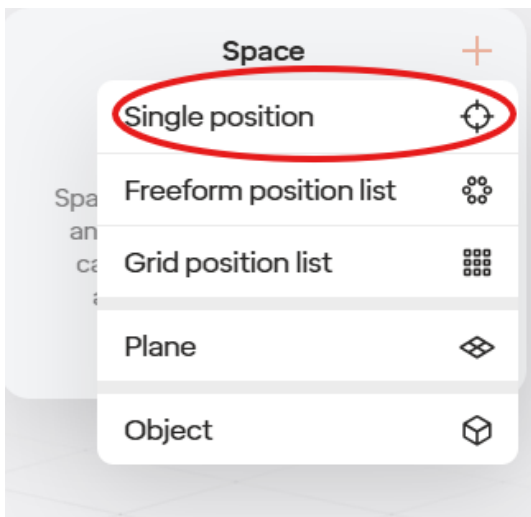
+ Add Waypoint

+ Add Step

Select “Manage Space” to create Spaces



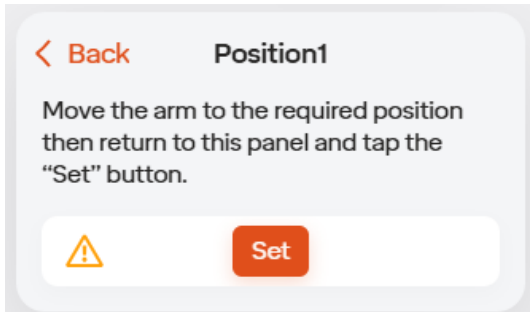
You will be brought back to the Move Robot view under the Space option in the upper right. Click the + button to add a new variable position.



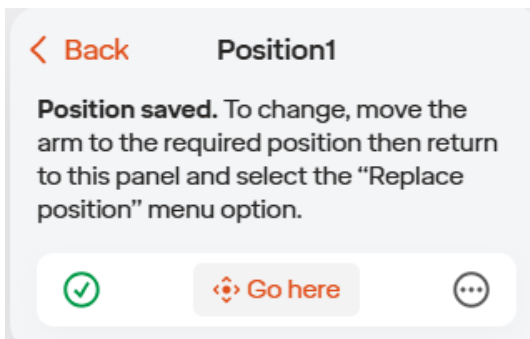
Select Single Position to create a new saved position.

1. Give the position a name.

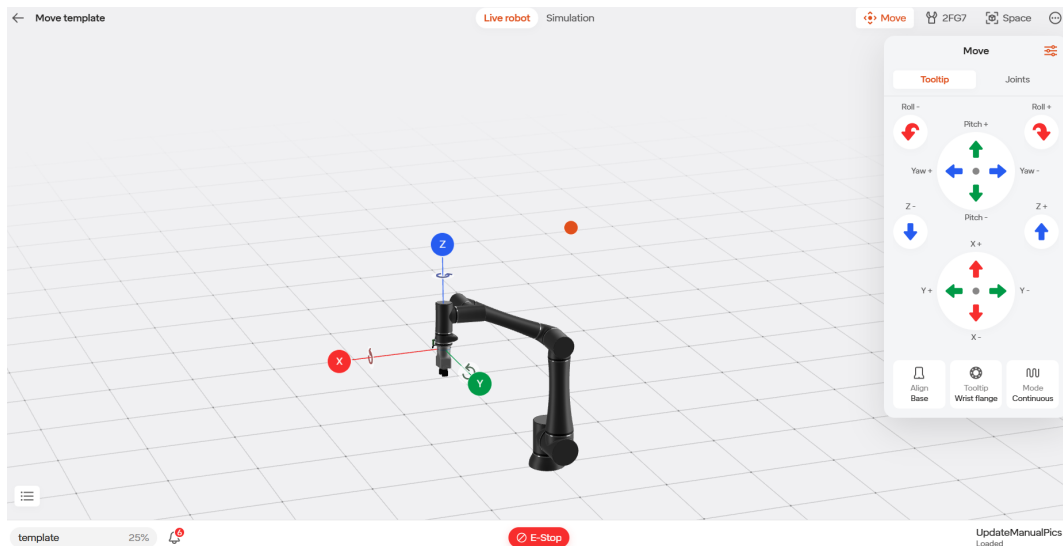
2. Select Create.



1. Set will set the position named “Position 1” to the current robot location.

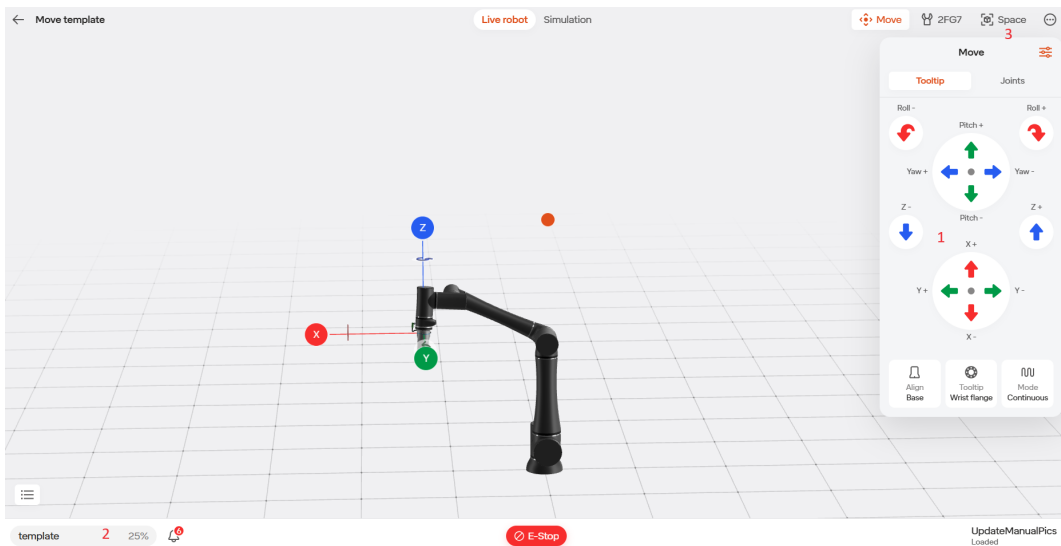


2. “Go Here” will drive the robot to the saved position if held down.

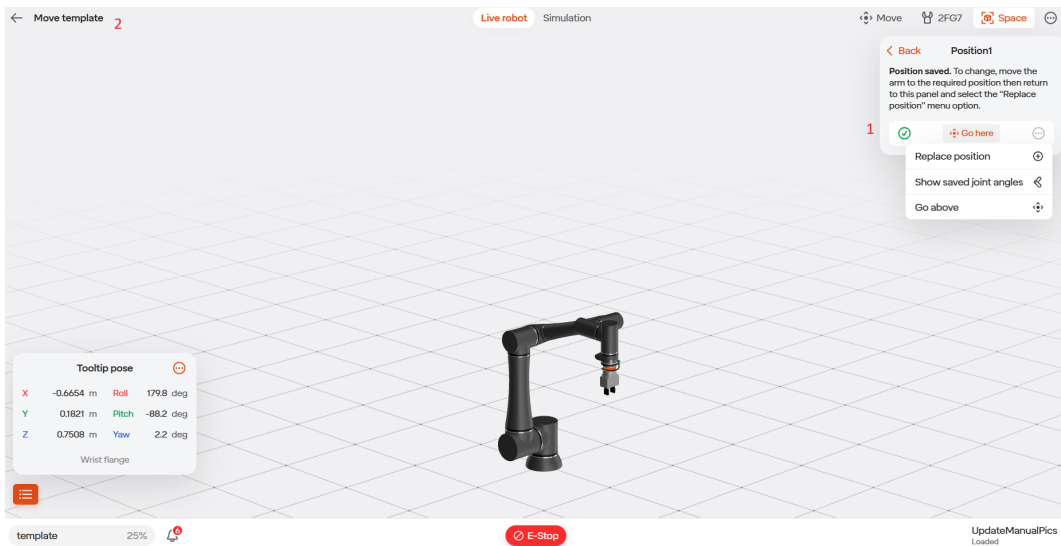


3. Use the Tooltip jogging area to move the robot to the desired initial position.

4. We are going to set a safe position above our work area for the robot to go to every time we start the routine.



1. Jog the robot to the desired position using the arrows.
2. If needed, use the menu to change the robot speed.
3. Once you are in the desired position, use the Space icon to return to the Space positions.



1. Set the position to the variable "Position 1".
2. Select the back arrow to go back to the routine.

Waypoint

Description
Move the robot to a specific position in space

Manual From space Expression

Target **Position 1** Select space

Match joint angles specified ☒

The robot will match the exact joint angles in the position.

Path

Move here in a straight line ☐

Stop here ☒

Blending None >

Stop here is required because this is the only point in the move.

Select our new saved position “Position 1” from the drop down.

Create a step that is Below Position 1 and name it “Position 2”. Then add another waypoint step below the first waypoint step and add the new position

Manual Example

Description

Space Position 1 Position 2 Manage space

Variables + Add variable

Steps Variables: Off

1. Loop forever
2. Move arm through 2 segments 100%
3. Waypoint to Position 1
4. Waypoint to Position 2

+ Add Waypoint

+ Add Step

Waypoint

Description
Waypoint to Position 2

Manual From space Expression

Target Position 2

Match joint angles specified ☒

The robot will match the exact joint angles in the position.

Path

Move here in a straight line ☐

Stop here ☒

Blending None >

Stop here is required because this is the last point in the move.

We have now completed our first Move arm step. Next add an Actuate Gripper step by selecting the three dots icon on the Move arm step and selecting “Add step Above”.

Add step

Cancel

Basic

Move arm
Specify a location that the robot should go to by defining the joint or gripper positions

Actuate gripper
Open or close the gripper to pick up objects

Set payload
Sets the expected payload at the end of the arm

Often you will want to set your tool configuration at the beginning of the routine as you do not know how it will be left every time the robot stops. Select Actuate Gripper.

Select the Actuate Gripper the step to edit the step.

Grippers can be changed from the Equipment Manager in the Robot settings.

1. Grip kind: Grip with force
2. Stop routine if grip fails: ☒
3. Grip Direction: ☒ Inward ☐ Outward
4. Width: 0.071 m
5. Force: 20N
6. Payload: 0kg
7. Grip tolerance: 2 mm

Difference between actual diameter and measured diameter at the end of the step to consider the grip successful

Configure end effector target

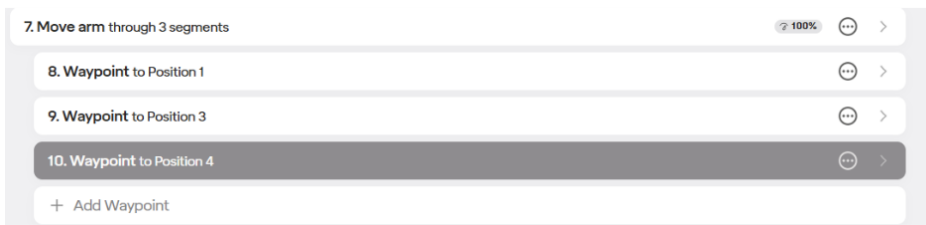
1. If we want to close the tool with some force select the Grip object with force checkbox. Since we are opening, we will change this option to Move To Position.
2. Set the GRIP DIRECTION to inward.
3. Set the gripper WIDTH, since we are opening, we will set it wide open
4. Set the FORCE to grip with. We will leave at 20N as we are opening.
5. Set the PAYLOAD for the target. Since we are just opening, this will be left at 0kg
6. Set the TOLERANCE for the gripper to be considered successful. This will be left at 2mm.
7. Click Configure End Effector Target.
8. The live robot page will open. Press the “Hold to Apply Changes” and then “Confirm End Effector Position” in the upper left. You’ll then be brought back to the Routine page.

Next add a step below the move arm step that closes the gripper to simulate grabbing an object. The routine will look something like this.



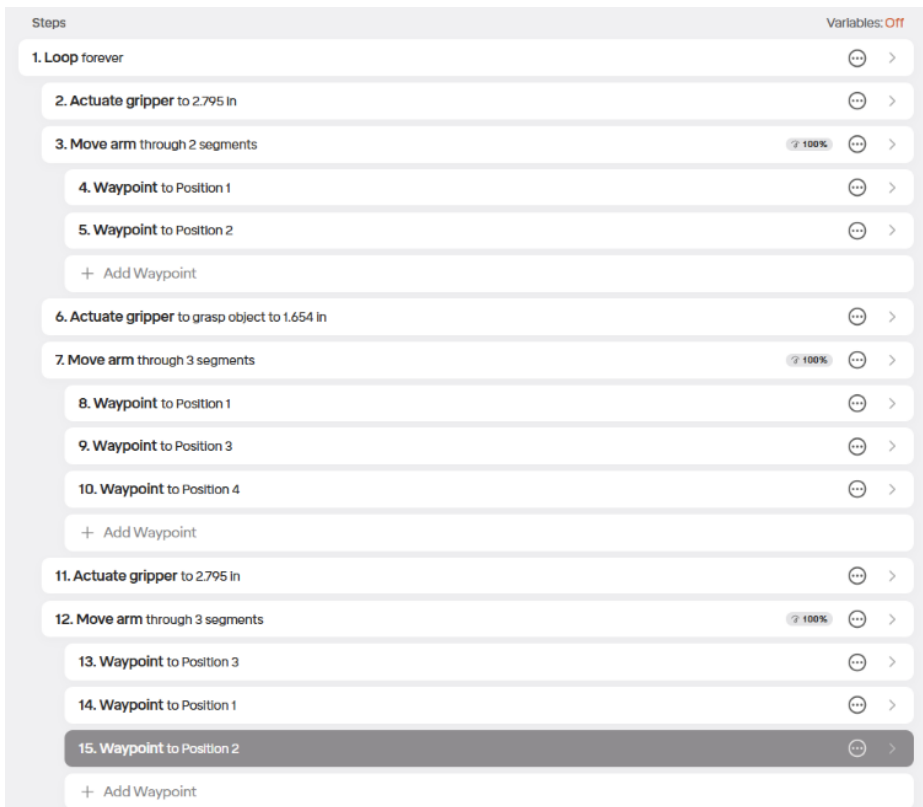
Next we will create two more positions that will “place” the object we picked up. Name those positions “Position 3” and “Position 4”. Position 3 will be above Position 4.

We will then add another move arm step that moves from position 2 (the previous position) to position 4. The move arm step should look similar to this:

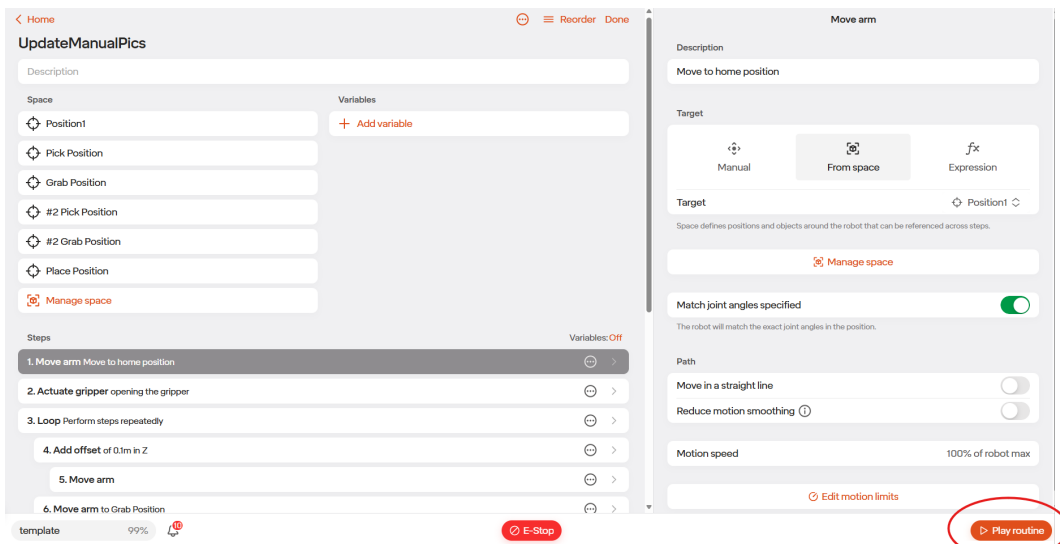


Next add another gripper step, this time opening the gripper to “drop the part”. Once that is done add another move arm step that moves through the positions to position 2.

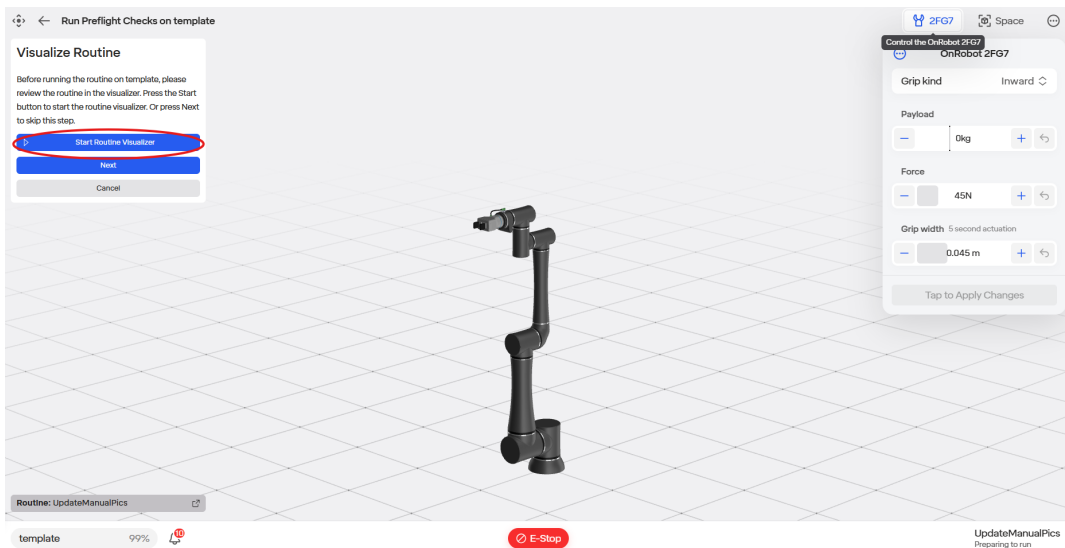
When done the final routine should look similar to this:



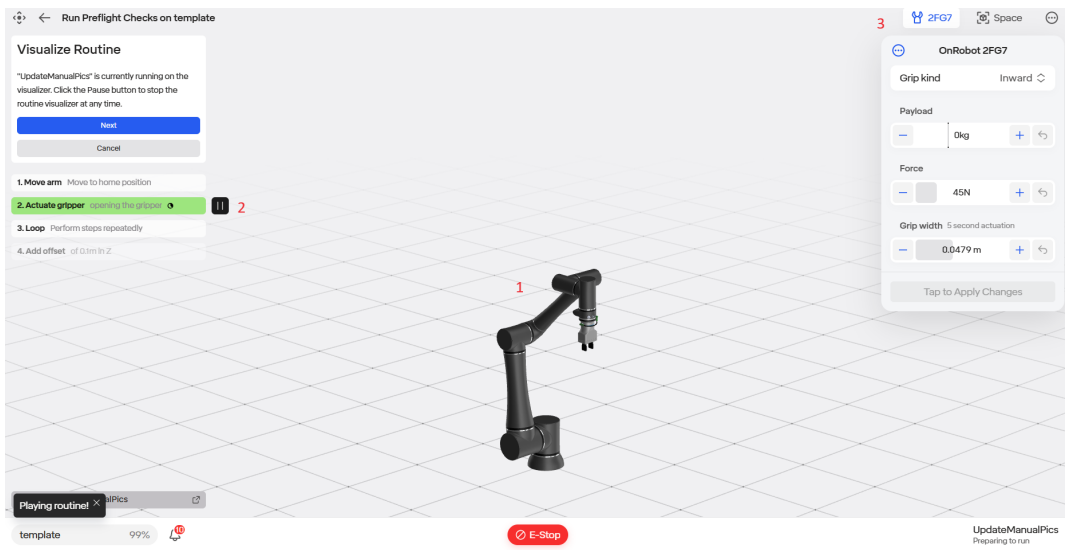
3.6.2 Running Routines



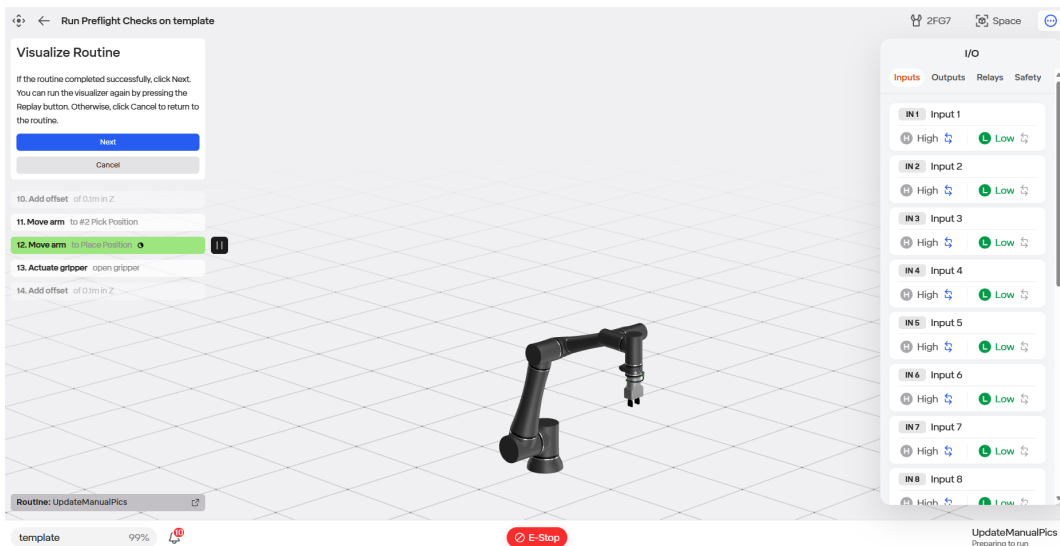
To test the routine, select the Play Routine icon.



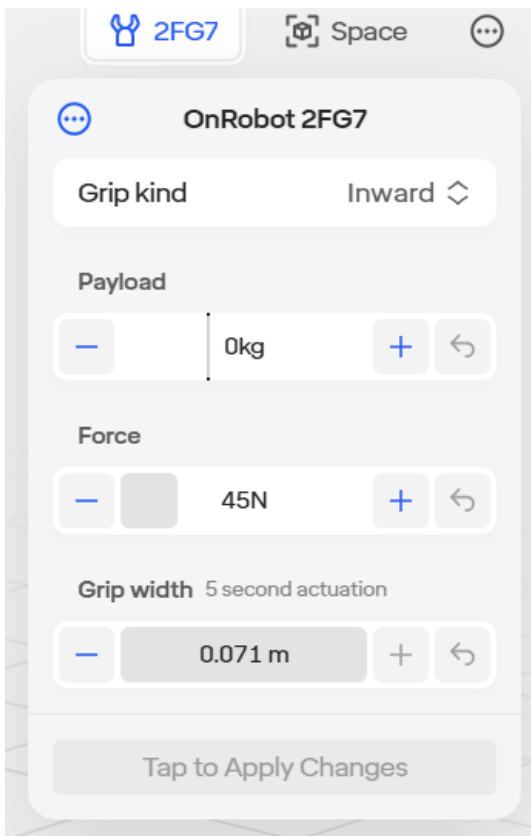
Select Start Routine Visualizer.



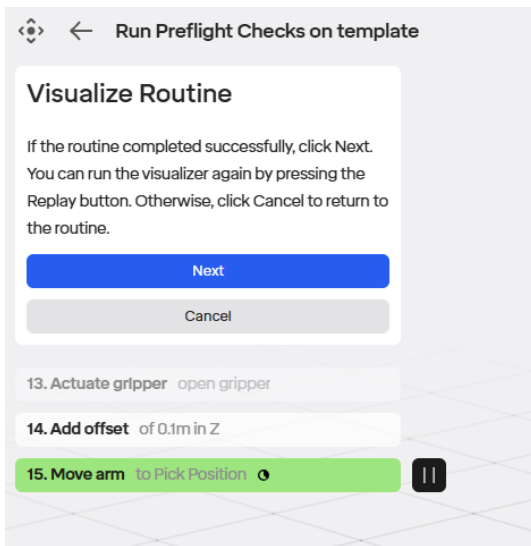
1. The robot will start completing the routine in a virtual environment.
2. The step the robot is currently completing will show on the left side of the screen.
3. The value of all Variables is shown in the upper right.
4. The routine will only loop once the first time. If you hit the "resume routine" button again it will loop indefinitely.



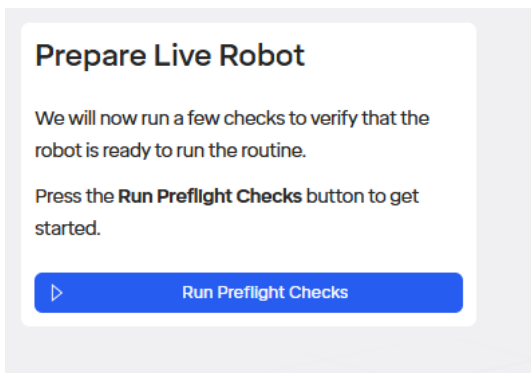
1. The **I/O** menu (hidden behind the three dot icon in the upper right) can be selected to show the IO values as the routine is running.



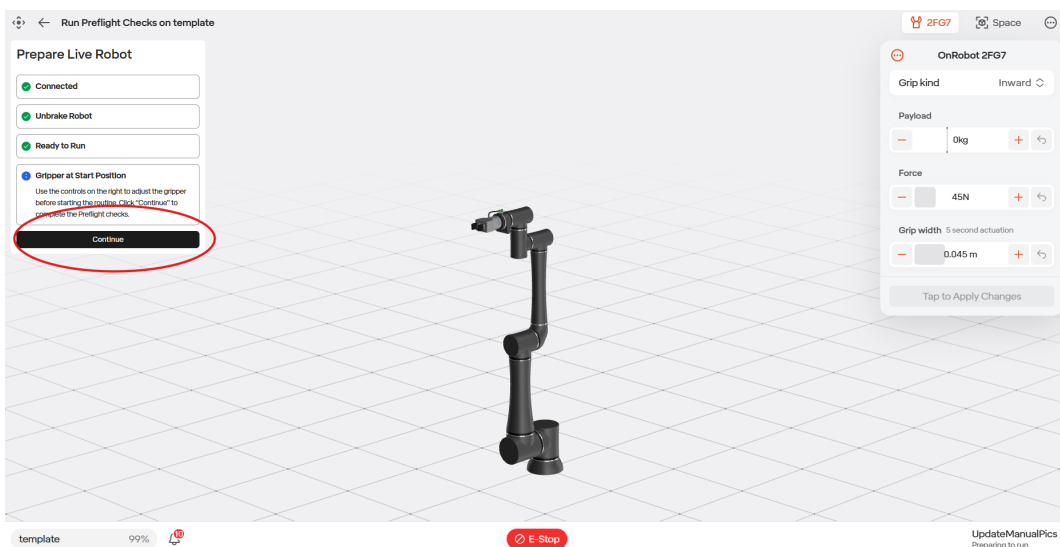
- 1.The **Gripper** menu can be selected to show the tool settings as the routine is running.



If the routine looks correct, it is ready to run on the physical robot. Select the blue **Next** button.



Select Run Pre-flight Checks.



Ensure the tool is in an acceptable position based on the start of your program. In our case we start by opening, so ensure the gripper is empty if you don't want to drop the product.

Prepare Live Robot

✓ Connected

✓ Unbrake Robot

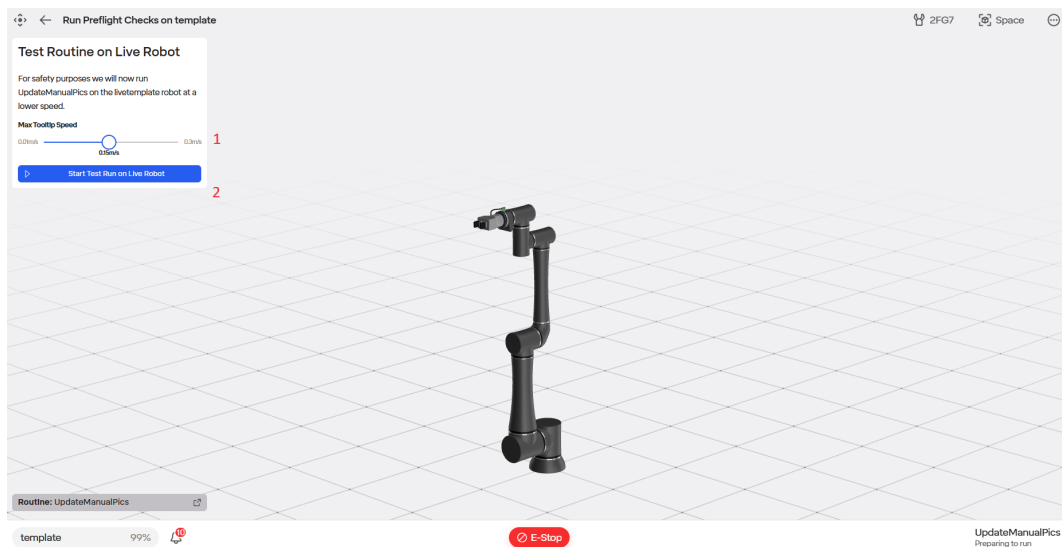
✓ Ready to Run

✓ Gripper at Start Position

All preflight checks on template have completed successfully and the live robot is now ready for its test run of the routine.

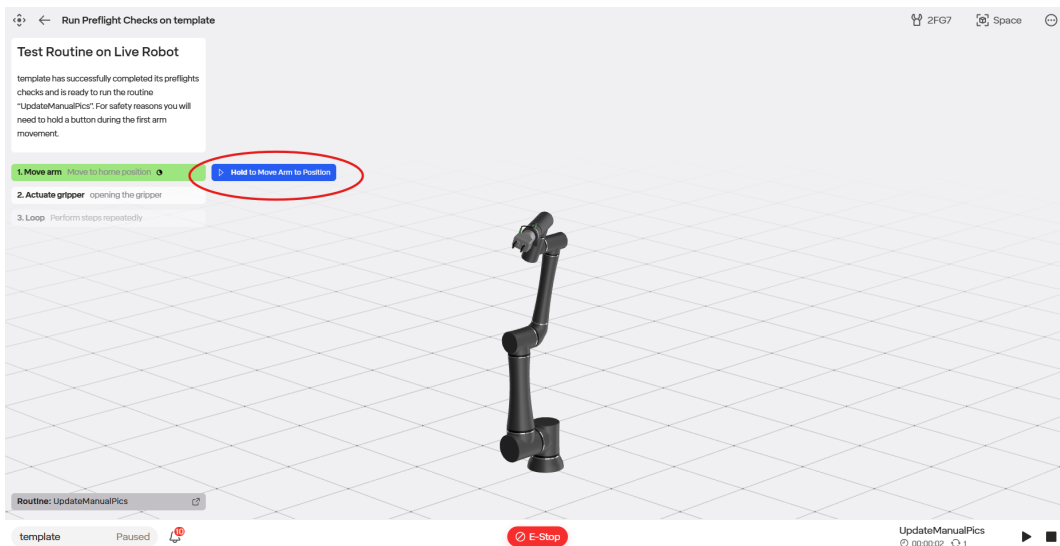
Test Run

Click the blue Test Run button.



1. Select the maximum speed you want the end of the arm to move for the initial run by setting Max Tooltip Speed.

2. Click the Start Test Run on Live Robot button.

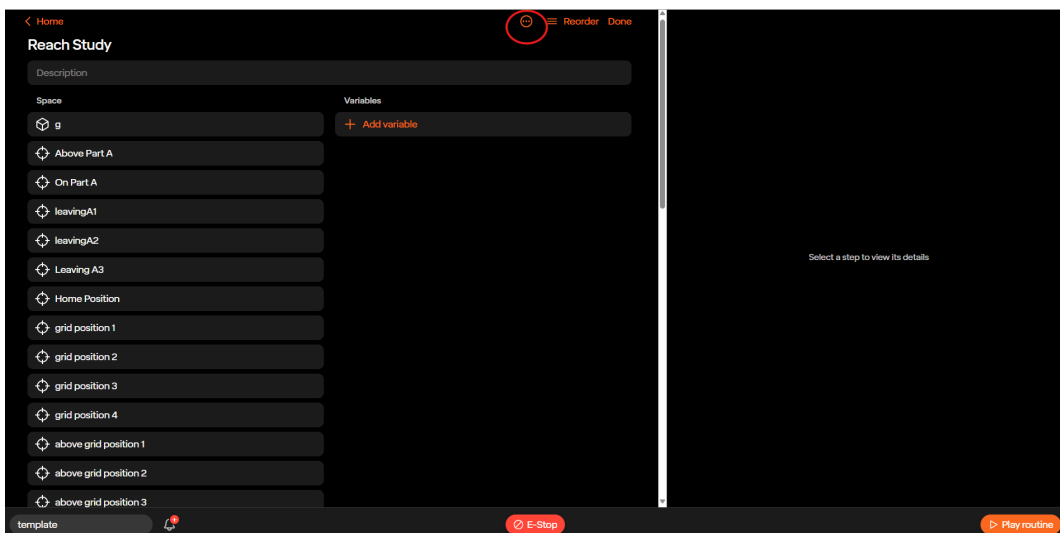


1. Click and hold the Hold to Move Arm to Position button to move the robot to the first position in the routine.
2. Watch the robot to ensure there are not going to be collisions during this move.
3. The routine will run at the specified reduced speed once and stop.
4. If the routine looks good, you can now play again with the “resume routine” icon that will appear in the upper right, and it will run at the speed specified in the menu in the lower left.

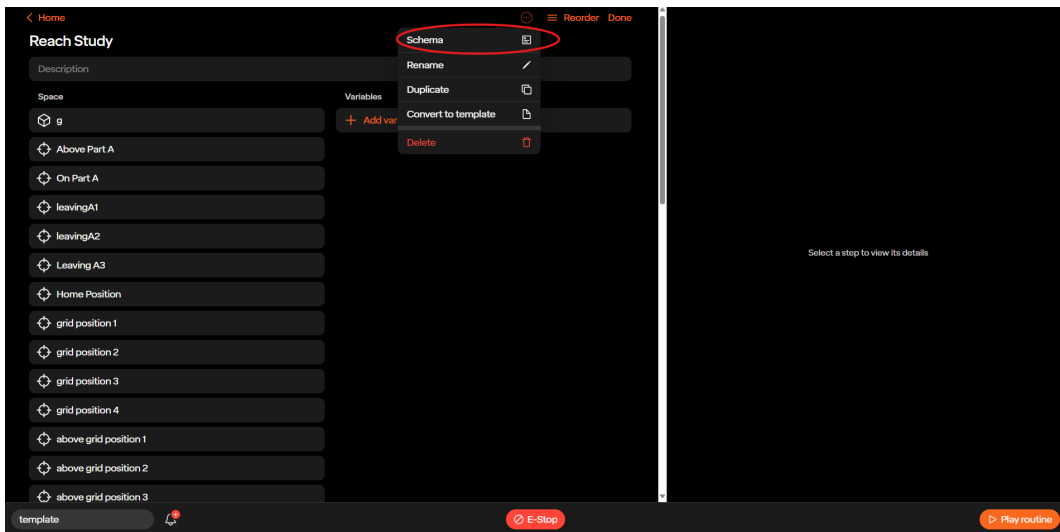
3.6.3 Accessing the Routine’s Schema

In the Standard Bots UI, you are able to access the routine’s schema. This allows you to copy, delete, and make changed to the schema of your desired robot routine.

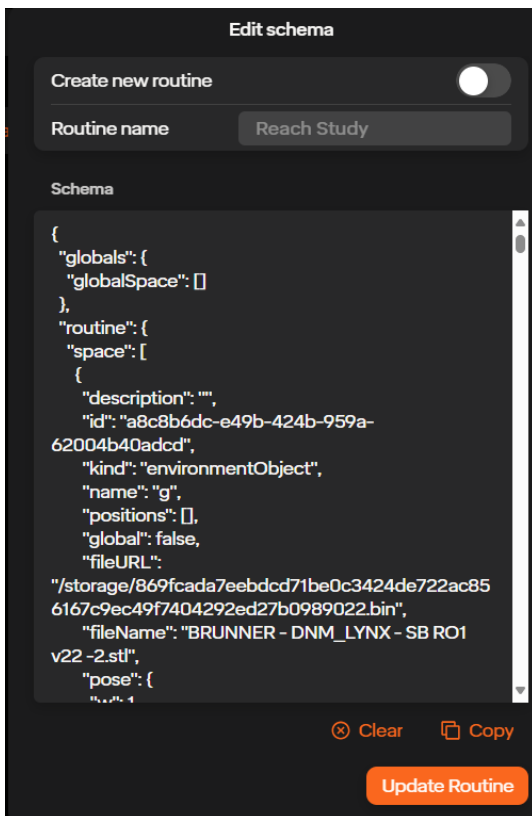
1. To access the routine’s schema, first go to the routine editor page of your desired routine.
2. Next, select the three dots icon at that is located at the top of the page.



3. Then select the Schema icon



- Once here you can access and make changes to the schema of your routine. At the bottom of the page are three buttons: Clear, Copy, and Update Routine. The Clear button deletes the schema, the Copy button copies the entire schema to your clipboard, and the Update Routine button saves the changes made to the schema. At the top of the page is a Create New Routine button. This allows you to create a separate routine with the changes you made instead of changing the existing routine.



3.6.4 Loop Settings

With the preset loop settings, this routine will loop forever until stopped. If you would like to change the loop setting, select the first step.

Loop

Description

Loop forever

Loop forever ✓

Loop specific number of times

Use rules

Main Loop

Automatically assign main loop ✓

Manually assign main loop

Choose one loop in the routine to be the main loop. This loop will be used when counting the routine cycles.

Automatic defaults to the outermost loop in your routine. Manual overrides the default main loop assignment.

For Loop under Basic Options you can choose to:

- Loop Forever: Loop until the program is stopped by the user.
- Loop specific number of times: Loop the number of times specified before going to the next step.
- Use Rules: Allows you to use math and variables to set the number of times the loop runs.

For our case leave as Loop Forever

•Under Main Loop you can choose:

- Automatically assign Main Loop: Set the loop that is to count the number of times the routine runs as the one index most left in the routine window.
- Manually assign Main Loop: Set the loop that is to count the number of times the routine runs manually in the Loop setting using the checkbox below.

3.6.5 Waypoints

As shown above in the example routine, you are able to create waypoints by creating Spaces and then by selecting that Space as a target in a Move Arm step. There are three other ways you can move through a waypoint: Manual, Expression, and Relative.

Manual is the easiest out of all the options. With that step selected you edit position and then move the robot to where you want that position to be and then save it. It is the quickest to create however it is the most limited, as you are not able to easily use this position throughout your routine as if you saved it as a Space.

Waypoint

Description

Waypoint to manually specified position

Manual

Space

Relative

Expression

Position saved 

Edit position

Clear position

Match joint angles specified 

The robot will match the exact joint angles in the position.

Path

Move here in a straight line 

Stop here 

Blending None >

With Expressions you are able to move to a position based off of an expression that you created. This can be accessed by selecting the Fx icon. To see more information on how to use expressions, go to section 5.7.1 to learn about Variables.

A relative waypoint step is similar to a waypoint step using a Space, however it has more features to allow you to make more complicated movements easier. Relative movements use Spaces as the tooltip position to move to but here you are able to change the Offset, Rotation, and with what respects the position is based on. With offsets you are able to use offsets in the XYZ plane based on the target position you select. Also with Rotation you are able to offset the pitch, roll and yaw from the position. These allow you to make multiple positions based off of one Space that you created. “With Respect To” allows you to base this position off of the base of the robot, the tool at the end of your robot, or of a plane that you created.

Waypoint

Description

Waypoint to pos1


Manual


Space


Relative


Expression

Target

 pos1

Offset

None >

Rotation

None >

With respect to

Base >

Match joint angles specified

☐

Tooltip to move

Automatic >

The robot will match the exact joint angles in the position.

Path

Move here in a straight line

☐

Stop here

☒

Blending

None >

Relative waypoints are the most recent way added to move waypoints. If you do not see the Relative icon in your Waypoint step and if you would like to use it, please contact Standard Bots Support.

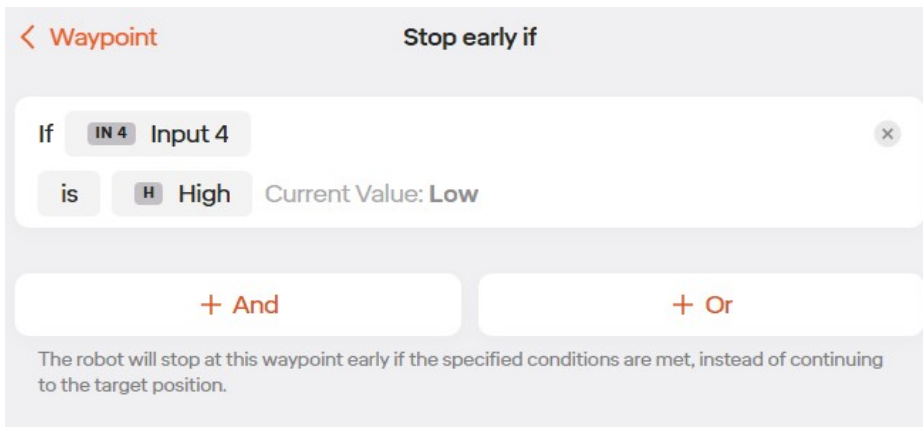
3.6.6 Stop Early If

With newer software releases, the waypoint steps now have a “Stop Early If” feature. Please reach out to support if you do not have this feature and would like access to it.

With the feature, you are able to configure conditions that will cause the robot to stop at this waypoint, before reaching the target position.

This can be configured by having inputs, expressions, or expressions reach the desired conditions

57

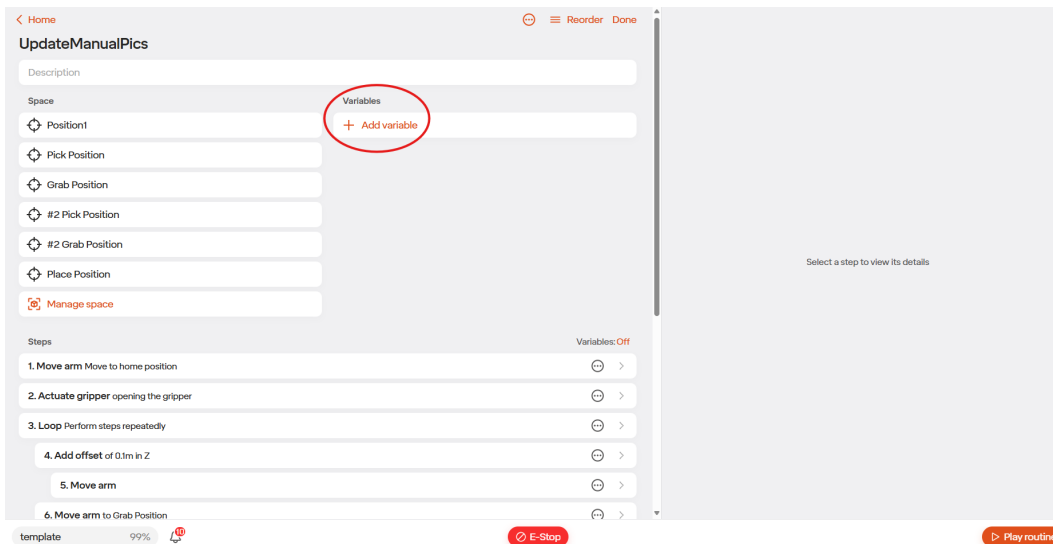


3.7 Advanced Routine Functionality

3.7.1 Variables

Variables can be used to assist with a variety of tasks within a routine such as:

- Storing a part offset for use in multiple Add Offset steps.
- Keeping track of how many parts have been picked in a given row.
- Keeping track of how many layers have been placed on a pallet.



To create a variable, locate the variables are in the routine editor.

A dialog box titled "Variable" with "Cancel" and "Save" buttons. It contains two input fields: "Name" with the value "increment_size" and "Initial value" with the value "0.005". There are small icons (a cross and three dots) to the right of each input field.

When creating a variable, be sure to set an initial value. If no value is set, the variable will be set to 0. The value in the initial value box will be used set for the variable every time the routine is started.

A "Target" selection menu with three options: "Manual", "From space", and "Expression". The "Expression" option is highlighted with a red circle. Below the options is a text input field with a three-dot icon. To the left of the input field is a tooltip that reads: "Expressions allow the robot to dynamically determine the position and using math." To the right of the input field is a list of variable categories: "Expressions Reference", "Step Variables", "Environment Variables", "IO Input Variables", "Equipment", and "Space". Each category has a corresponding icon.

To use a variable in the routine, click the “Fx expressions” button in the given area of the step. There are several sets of variables available.

Expressions Reference: Help menu on how to construct expressions

Step Variables: Data from each routine step that can be used, for example the number of times it has been executed in a loop.

Environmental Variables: Variables created by the user in the routine.

IO Input Variables: Access to each of the 16 24VDC inputs, 0 if off 1 if on.

Gripper State: For supported grippers, inputs based on current state, for example closed or open.

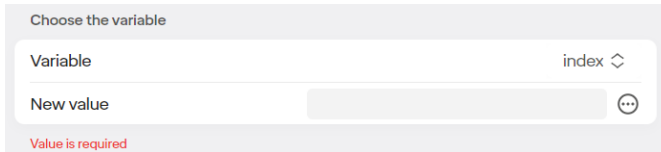
Space: Data set in the space, such as saved positions.

Note: If you are using a variable, you must add it using the “Fx expressions” button and then go to the three dots icon to find what variable you want to use (custom variables made will be found in “Environment Variables”). Do not manually type it in. Manually typing the variable will not include the correct prefix and will result in an error.

Here is an example of how to “add 1” to the value of a variable:

1. select the “set variable” step

2. select the variable you wish to add to (this example uses a variable named “index”)



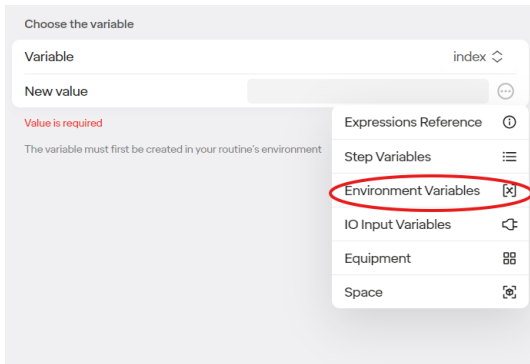
Choose the variable

Variable index

New value

Value is required

3. select the three dots icon and click “environmental variables”.



Choose the variable

Variable index

New value

Value is required

The variable must first be created in your routine's environment

Expressions Reference

Step Variables

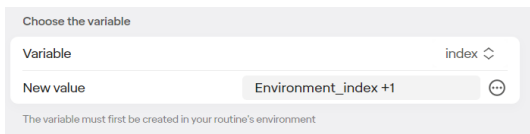
Environment Variables

IO Input Variables

Equipment

Space

4. Then select the variable you wish to change, it should automatically appear in the value bar. Notice how it's shown as “Environment_...”, typing in the name of your variable into the bar will not work.
5. Then put “+1” in the value bar.



Choose the variable

Variable index

New value Environment_index +1

Value is required

The variable must first be created in your routine's environment

6. Now every time the routine gets to this step, the value of “index” will increase by one.

This doesn't have to be only for addition, other math functions can be done as well.

3.7.2 Javascript

The Standard Bots routine editor provides the necessary instructions that address the requirements for the majority of applications. Should you need to incorporate more advanced functions, the routine editor allows you to write your own JavaScript into Loops, Add Offsets, Move Arm steps, and more.

Expressions Reference



How to use Expressions

Expressions allow you to dynamically define data based on the variables present in your routine. Expressions support any expression from the JavaScript language, which can be used to construct mathematical expressions, access lists of values, or create values based on conditions.

Examples include:

- `loopCount * 12` to compute an offset of 12mm per loop count for stacking
- `currentStep == 'left' ? leftPositionList : rightPositionList` to use the the left position list if stacking boxes on the left, and the right one otherwise.
- `Math.sin(angle) * armLength` to calculate the z offset along a circular path.
- `positionList[loopCount]` to access the entry in a position list corresponding to the loop
- `positionList[0]` to access the first step in a position list (lists in JavaScript start with 0).

You can access the above help menu by clicking “Fx” in any text window in the routine editor and selecting “Expressions Reference”.

In the above example, this loop will iterate until the loop has iterated more than the value set in an environmental variable, `partsPerRow`. After the loop has iterated more times than the value of `partsPerRow`, the routine will move to the next step after what is contained in this loop.

Loop

Description

Perform steps repeatedly

Loop forever

Loop specific number of times

Use rules ✓

If f× Expression

Environment_environmentBoxCountOnPallet <= 16 && Environment_...

+ And

+ Or

Main Loop

Automatically assign main loop ✓

Manually assign main loop

Choose one loop in the routine to be the main loop. This loop will be used when counting the routine cycles.

Automatic defaults to the outermost loop in your routine. Manual overrides the default main loop assignment.

Loop

Description

Perform steps repeatedly

Loop forever

Loop specific number of times

Use rules ✓

If f× Expression

!OnPallet <= 16 && Environment_environmentBoxCountOnPallet > 8

+ And

+ Or

Main Loop

Automatically assign main loop ✓

Manually assign main loop

Choose one loop in the routine to be the main loop. This loop will be used when counting the routine cycles.

Automatic defaults to the outermost loop in your routine. Manual overrides the default main loop assignment.

In the above example, the code is checking if a variable boxCountOnPallet is less than or equal to 16 but more than 8. If the above condition is true, whatever is indented below the If statement in the routine will execute.

Some Examples of JavaScript Code are Below:

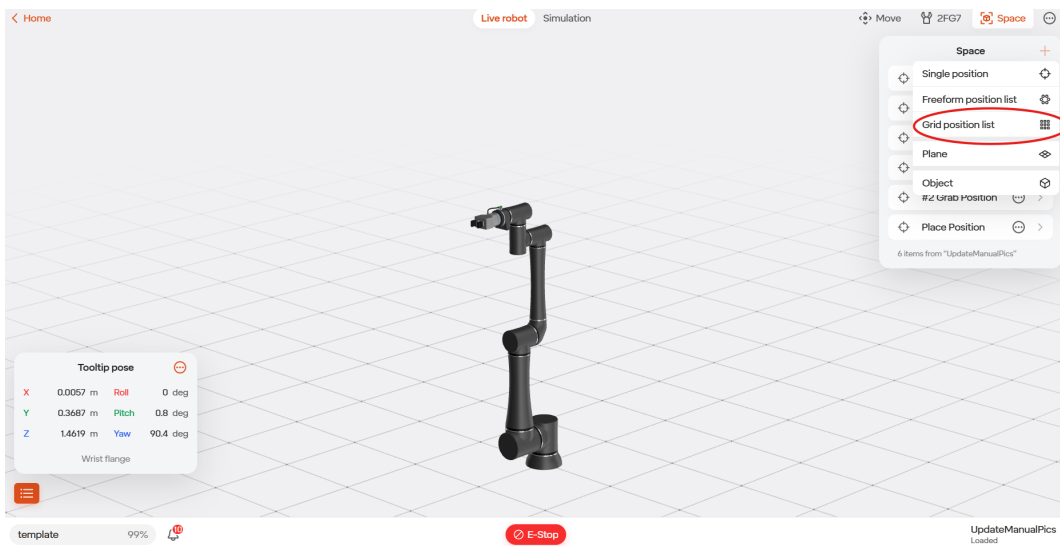
Command	Description
>	Greater Than
<	Less Than
>=	Greater Than or Equal To
<=	Less Than or Equal To
==	Equal To
===	Strictly Equal To
!=	Not Equal To
!==	Strictly Not Equal To

Basic Math Symbols Apply “+” for Addition, “-” for Subtraction, “*” for Multiplication, and “/” for Division.

Programming Example: We want to wait for Variable 1 to equal 18. That would be created in a If Fx Expression as follows: Environment_variable1 == 18 where Enviroment_variable1 is Variable 1 and we are waiting for it to be equal to 18 to continue the program.

3.7.3 Grid Position Lists

Grids are often used in robot programming when a large number of parts needs to be processed by a robot. Using the Grid Position List functionality when creating a routine can save significant time and effort.



To create a grid, go to the Move Robot area, go to the Space (square icon), select the Plus icon, and then select Grid Position List.

Back

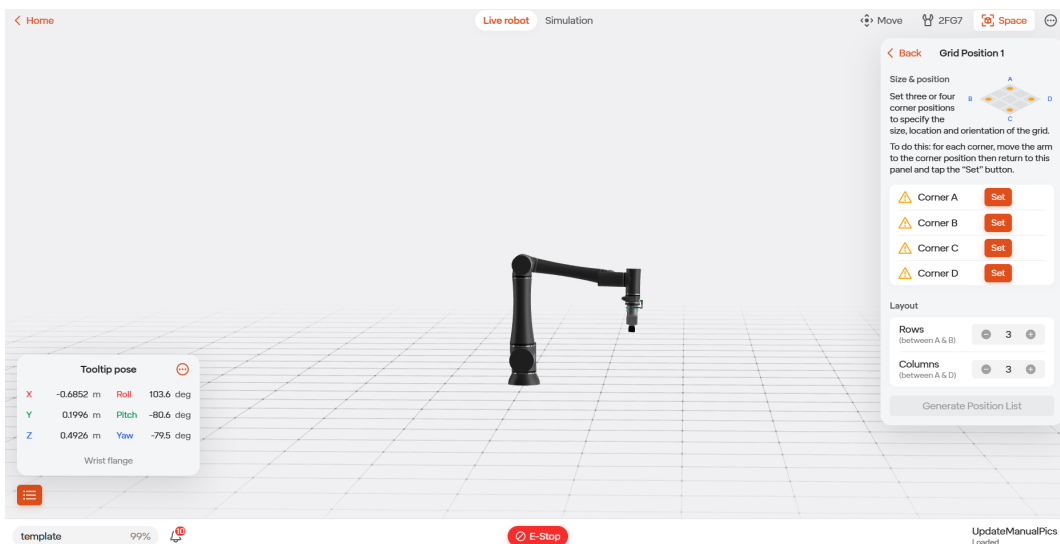
New item

Name

Description

Create

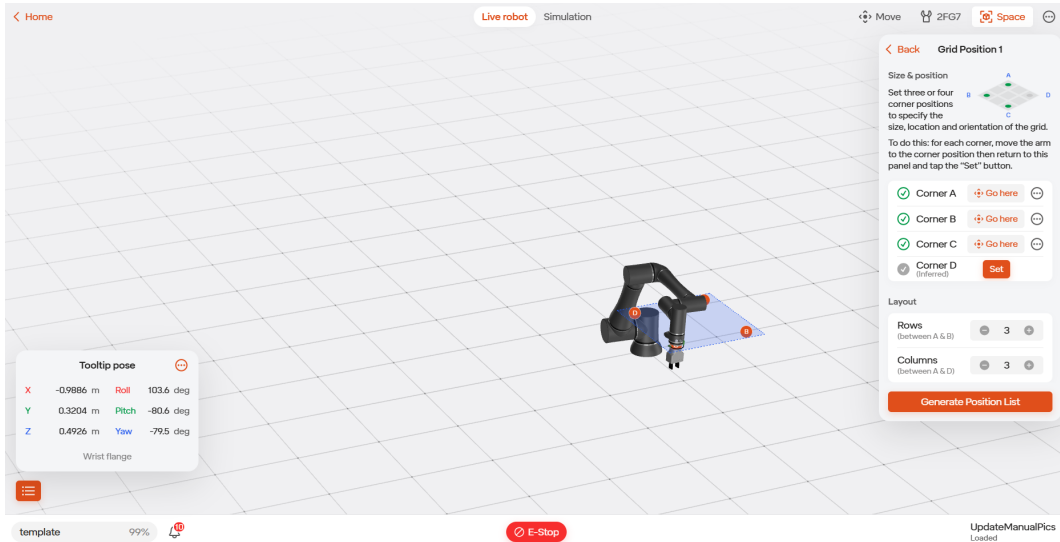
Give a name to the Grid Position List.



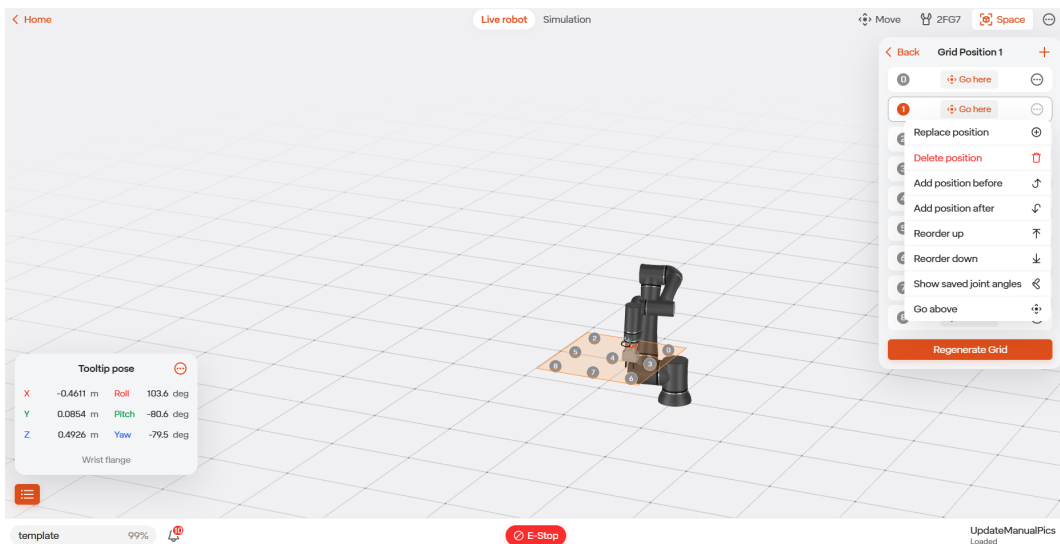
To create the grid, first drive the robot to a corner of the grid using the Jog Robot functionality in the

Move Robot view. Set the position. Repeat for 3 corners of the grid.

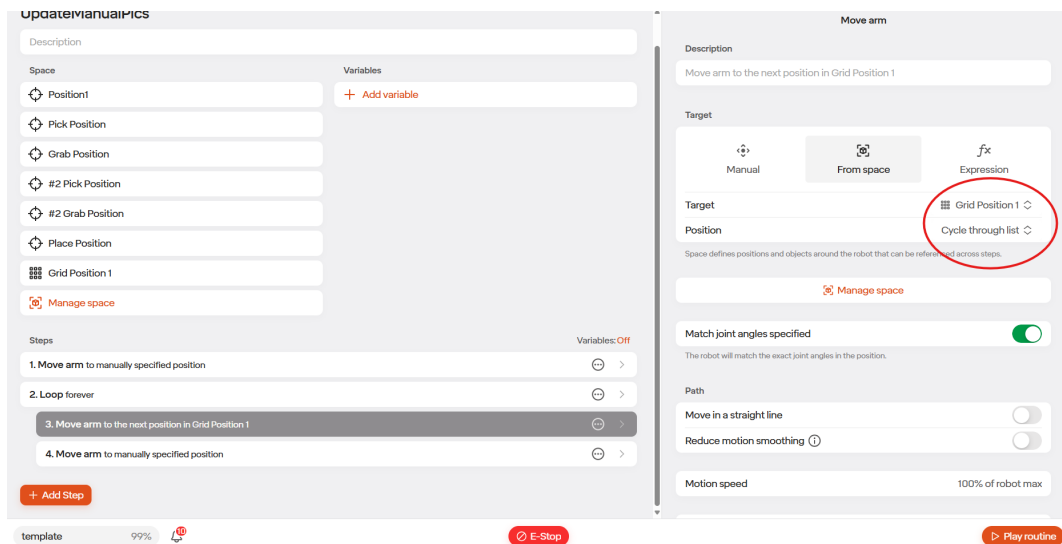
Under Layout, select the number of rows and columns in the grid. This will be the layout of the parts within the grid.



Select “Generate Position List”.



After the grid is generated, you can go to the positions using “Go Here”, tune them by using “Replace Position” (hidden behind the three dot icon), or exit.

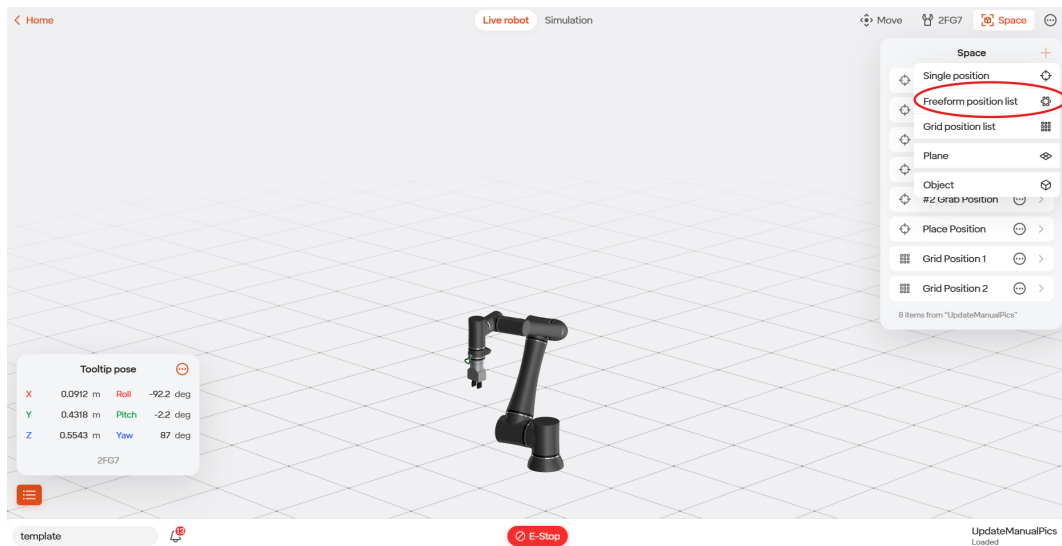


To use the grid, use a Move Arm command within a loop and select the grid within the “From Space” option. The robot will move to the positions in the grid, in order, each time the loop executes.

To program a grid with multiple layers, use an “Add Offset” to set an offset to the grid positions based on the number of times the loop has executed.

3.7.4 Freeform Position Lists

A Freeform Position List is a list of positions in an array. This can be useful when programming if you need to go to positions in order in a loop that are not in a grid, or indirectly address positions.



To create a Freeform Position List, go to the Move Robot area, go to the space (square icon), select the Plus icon, and then select Freeform Position List.

You can touch up the positions using “Go Here” and “Replace” (which is hidden behind the three dots icon). Once finished simply exit.

The screenshot shows the 'Move arm' configuration interface. The 'Description' field contains 'Move arm to the next position in Freeform Position Lists 1'. Under the 'Target' section, three options are available: 'Manual', 'From space' (which is selected and highlighted with a red circle), and 'Expression'. Below these, the 'Target' field shows 'Freeform Position Lists 1' with a dropdown arrow, and the 'Position' field shows 'Cycle through list' with a dropdown arrow. A 'Manage space' button is located below the 'Position' field. Further down, there is a 'Match joint angles specified' toggle switch which is turned on. Under the 'Path' section, there are two toggle switches: 'Move in a straight line' and 'Reduce motion smoothing', both of which are turned off. At the bottom, the 'Motion speed' is set to '100% of robot max'.

To go to the positions in order, use a Move Arm command and select the list from the Space.

The screenshot shows the 'Move arm' configuration interface with the 'Expression' option selected under the 'Target' section. The 'Description' field now contains 'Move arm based on expression Space_freeformPositionLists1 [Environment_index]'. The 'Target' field shows the expression 'Space_freeformPositionLists1 [Environment_index]' with a three-dot menu icon to its right. Below this, there is a note: 'Expressions allow the robot to dynamically determine the position to move to, referencing variables and using math.' The 'Match joint angles specified' toggle switch remains turned on. Under the 'Path' section, the 'Move in a straight line' and 'Reduce motion smoothing' toggle switches remain turned off. At the bottom, the 'Motion speed' is still set to '100% of robot max', and an 'Edit motion limits' button is visible.

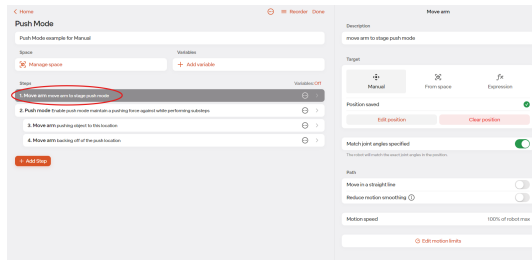
To access positions in the list not in order, use the “Expressions” option as shown. In this example we are going to the position in the list based off the environmental variable “index”.

3.7.5 Push Mode

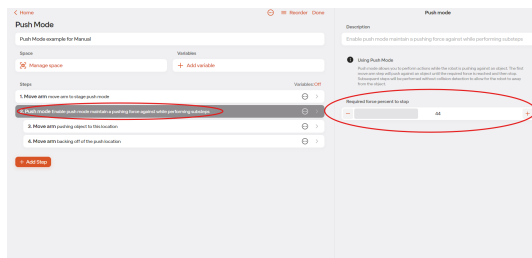
Push Mode allows you to perform actions while the robot is pushing against an object. The first move arm step will have the robot push against an object until the custom force requirement was reached. That step is finished once the force requirement is reached. Then the next step would be moving the robot away from the object. The steps are performed without the collision detection, allowing the robot to detect the needed amount of force.

Here is an example of implementing Push Mode into a routine:

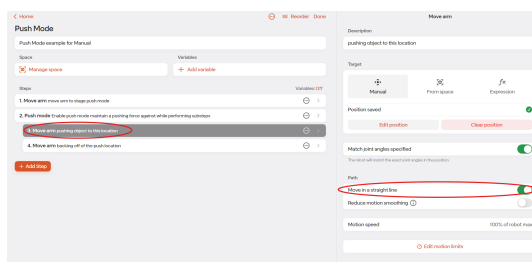
Step 1 in this example is used to move the robot into the initial position to push the object.



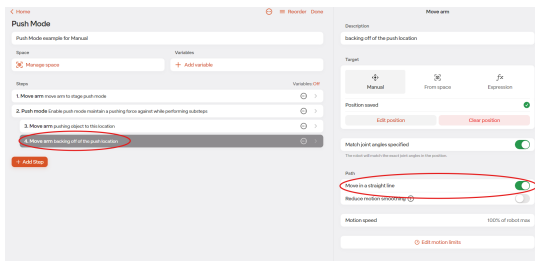
Step 2 in this example is enabling the push mode. Set the required amount of force for the robot to stop.



Step 3 in this example is the robot pushing the object to it's final position. The robot will stop when it detects the selected force percentage. It is required that “move in a straight line” is selected for this step to work.



Step 4 in this example is the robot arm backing away from the object. This step is done still under Push Mode because Push Mode turns off collision detection. Once the routine is out of Push Mode, collision detection is back on and if the robot is still pushed against the object an error will happen. “Move in a straight line” will still need to be selected for this step.



Once the robot is backed away from the pushed object, the steps after can leave Push Mode.

3.7.6 Prompt User

With the Prompt User step, the routine will pause and the operator will be prompted to take an action. Those actions are Confirm, Choice, or Freeform.

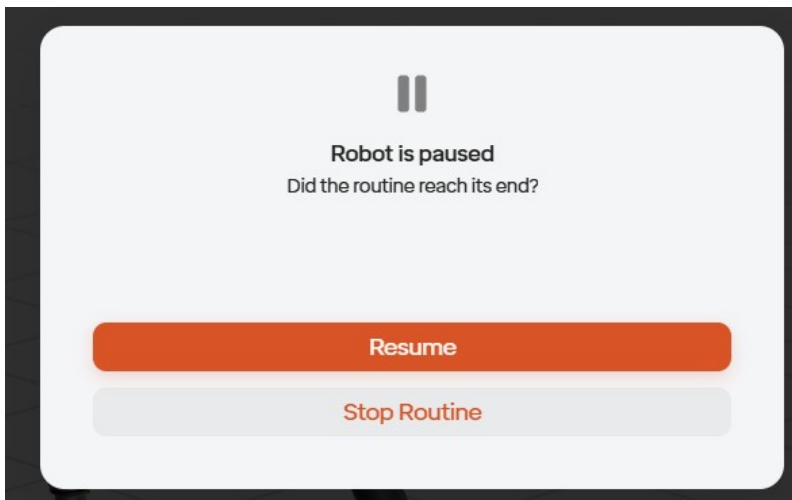
The screenshot shows the 'Prompt User' configuration interface. It has a title 'Prompt User' at the top. Below it is a 'Description' field with the text 'Routines are paused. The user will be prompted to take an action.' Below that is a 'Message' field, which is currently empty. Underneath the message field is the 'Input type' section, which contains three buttons: 'Confirmation' (with a checkmark icon), 'Choice' (with a list icon), and 'Freeform' (with a notepad icon). Below these buttons is a small explanatory text: 'The robot will display the message and just ask the operator to confirm before resuming the routine.' At the bottom, there is a 'Resume delay interval' field with a value of '5' and a unit of 'sec'. Below this field is another small explanatory text: 'The robot will wait this long to resume the routine after the operator completes the prompt.'

Each action allows you to add a message. The message will appear with the action when the routine pauses.

There is also a Resume Delay Interval. This is the amount of time the robot will wait to resume the routine after the operator has completed the prompt.

Confirmation:

When Confirmation is selected, the operator is given two answer's respond with, "Resume" and "Stop Routine". In the example image below, "Did the routine reach its end?" is the message that was written when editing the step.



Choice:

When Choice is selected, you are able to add options for the operator to select. The examples in the image below are “Yes” and “No”.

Prompt User

Description

Ask "Did the routine reach its end?"

Message

Did the routine reach its end?

Input type

☒ Confirmation ☒ **Choice** ☐ Freeform

Yes

No

[Add option](#)

The robot will display the message and ask the operator for a choice between the options specified here

Resume delay interval sec

The robot will wait this long to resume the routine after the operator completes the prompt.

A screenshot of a user interface dialog box. At the top, there is a pause icon (two vertical bars) and the text "Input required" followed by "Did the routine reach its end?". Below this, there are two radio button options: "Yes" and "No". At the bottom, there are two buttons: "Resume" and "Stop Routine". The "Stop Routine" button is highlighted in red.

Freeform: When Freeform is selected, the operator has to enter a response in order to resume the routine.

A screenshot of a user interface dialog box, similar to the one above. It has a pause icon and the text "Input required" followed by "Did the routine reach its end?". Below this, there is a text input field with the placeholder text "Respond here...". At the bottom, there are two buttons: "Resume" and "Stop Routine". The "Stop Routine" button is highlighted in red.

If Statements:

You are able to use the result of the Prompt User step in an If Statement. The Prompt User step can be used in the If Statement by "Variables" and then your Prompt User step.

When using Confirmation, you are able to choose between "Confirmed" and "Not Confirmed" in the If statement.

If

Description

Run substeps if condition is met

If **4. Prompt User: Confirm** ×

is **Confirmed**

Confirmed ✓

Not confirmed

+ Or

When using Choice, you are able to choose between whatever options you made in the Prompt User step (the options made in the example image are Yes, No, and Maybe)

If

Description

Run substeps if condition is met

If **4. Prompt User: Choice** ×

is **Value**

Yes

No

Maybe

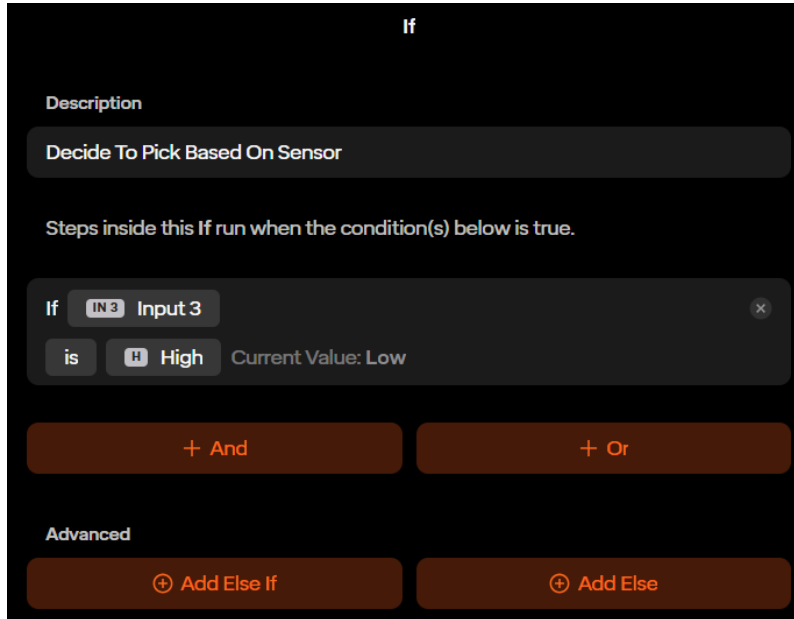
+ Or

3.7.7 Decision Logic

If Statement - Else If - Else:

To create branches and decisions in your program you can use the If / Else If / Else logic construction.

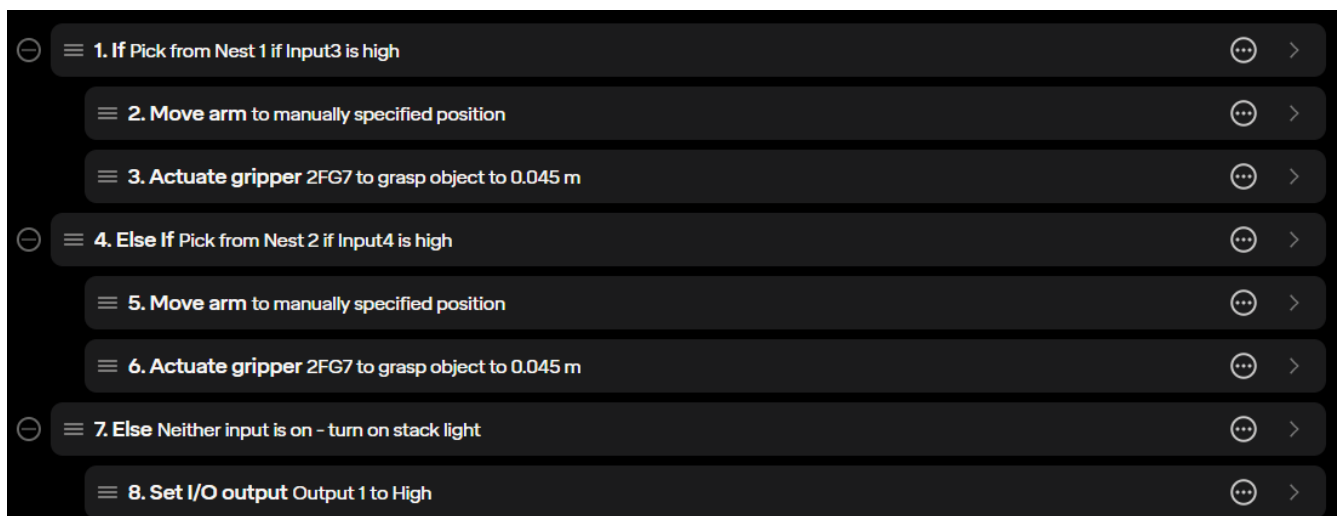
Add an **If** step to your program.



The IF statement has logic that when it evaluates to true will run more steps in your program inside the IF block.

For more advanced logic you can add an Else by clicking the “Add Else” button. Any steps in the Else section are run if the initial IF is false.

For further functionality you can add one or more “Else If” sections by clicking the “Add Else If” button.



3.7.8 Machine Tending Application

Introduction:

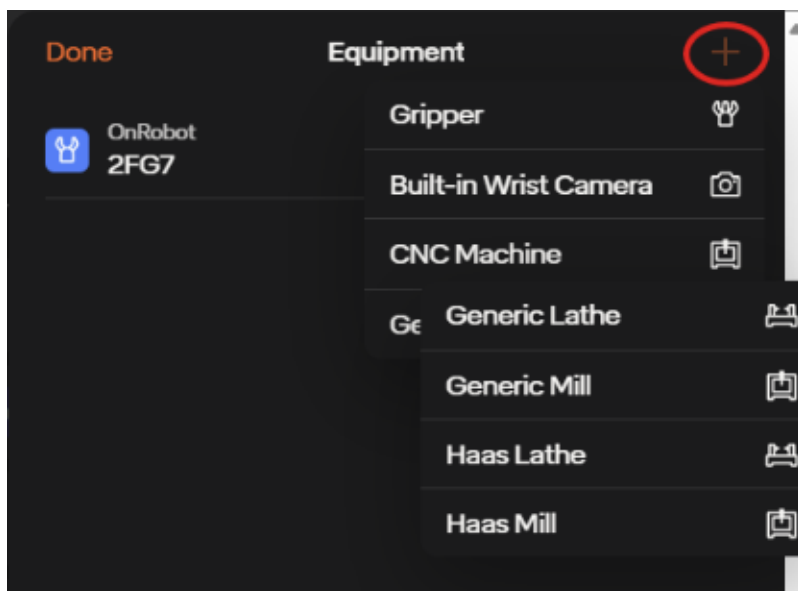
Standard Bots offers machine tending solutions by the ability to add a CNC machine into your robot's equipment. If you are using a Haas machine and are interested in using the Haas Ethernet Integration system, go to chapter 5.7.6.

Required components:

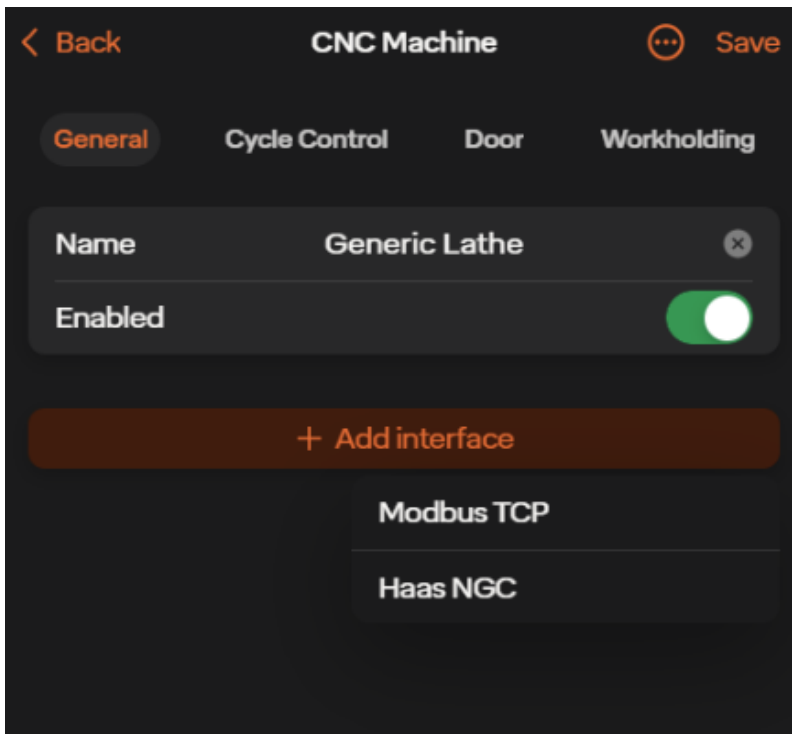
- Vices/vice adapters
- Robot end effector
- conveyors/feeders (if needed)
- loading/unloading stations
- Safety devices (if needed)
 - Area scanners
 - Light curtains and mirrors
 - Safety rate start buttons
 - Fence interlocks

Integrating CNC Machine:

To add the CNC machine into your robot's equipment, tap on the robot's name at the lower left corner of the screen, tap the tab that's labeled Equipment, then in the equipment tab press the orange plus icon in the upper right. Here you will see the list of the different CNC machines (general lathe, general mill, Haas lathe, Haas mill)



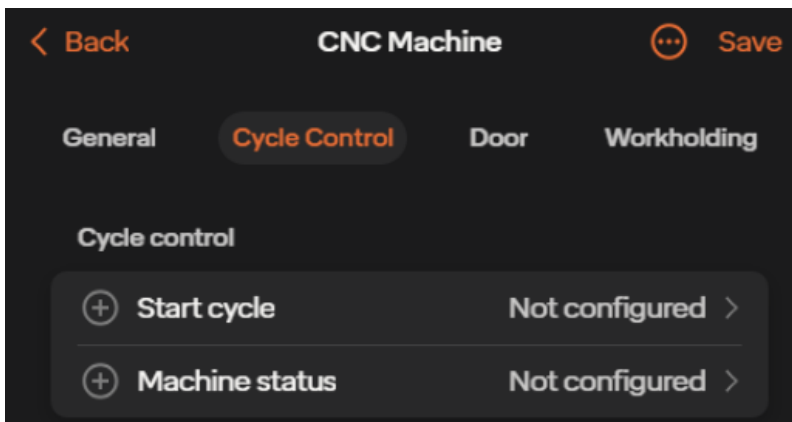
If you are using a generic lathe or mill, select the desired one. Once doing so you will be brought to the generals page. On the generals page you will be able to name your equipment, enable/disable your equipment, and add Modbus TCP and Haas NGC if desired.



You can then select Cycle Control which can be seen on the top of the page. To tend a machine, the robot requires the ability to start the machine and to know when the machine is finished.

If the machine and the robot cannot be connected directly, these tasks can be accomplished by manually pushing buttons and by reading screens/lights with the robot's camera.

In the Cycle Control page you will see Start cycle and Machine status.



In the Start cycle page you will be able to choose which output or relay is used to start the machine's cycle as well as if it goes high or low. You will be able to choose whether the output is set or if it is just a pulse. If pulse time is chosen you will also be able to choose the length of the pulse time. At the bottom of the page you can test the start cycle as well as add more outputs if needed. Make sure to save any changes in the upper right

< Back Start cycle Save

Action	Start cycle
Control via Pulse Unassigned to High X	Control box IO Pulse time: 0. sec

Add output Test Start cycle

In the Machine status page you can set the inputs for machine running as well as machine faulting. For machine running, the robot will have assigned inputs which will monitor the machine's status during its routine. For machine faulted, the robot can have assigned inputs which will monitor the machine for a fault status.

< Back Machine status Save

Machine running

Read via: Control box IO

Running when: Unassigned is High

The robot will monitor the machine's status during "Run Program" and "Monitor Machine" steps.

Machine faulted

Read via: Control box IO

Faulted when: Unassigned is High

The robot will monitor the machine for fault status during cycles. The robot can also be configured to automatically stop according to various conditions in Safety settings.

You can then backout and select the Door icon which can be seen on the top of the page next to Cycle control. Here you can set outputs which can control the opening and closing of the machine's door. You will be able to choose whether the output is set or if it is just a pulse. At the bottom of the page you can add more actions if needed.

Back

CNC Machine

⋮ Save

General

Cycle Control

Door

Workholding

Name

Autodoor

✕

Action

Open

◇

Control via

Control box IO

◇

Pulse

Unassigned

◇

to

High

✕

Pulse time

0.

sec

Add output

Test Open

Delete

Action

Close

◇

Control via

Control box IO

◇

Pulse

Unassigned

◇

to

High

✕

Pulse time

0.

sec

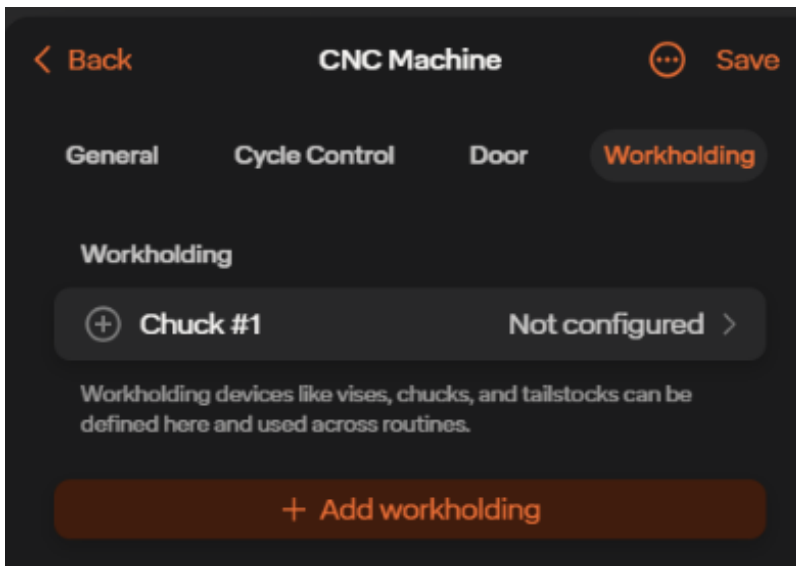
Add output

Test Close

Delete

+ Add action

Next, in the Workholding page, you can set workholding equipment in your machine which the robot can send outputs to. At the bottom of the page you can add more workholding equipment if needed.



The actions for the workholding equipment are separated into Clamp and Unclamp. For each one you are able to choose the output from the control box, its state, and whether that state is a pulse or if it's set. You are able to add more outputs to each action if needed. At the bottom of the page you can add more actions if desired.

Workholding

Name: Chuck #1

Clamp

Control via: Control box IO

Pulse Unassigned to High

Pulse time: 0. sec

Add output Test Clamp Delete

Unclamp

Control via: Control box IO

Pulse Unassigned to High

Pulse time: 0. sec

Add output Test Unclamp Delete

+ Add action

When done, make sure to save your machine. Your machine will now be shown in the equipment tab. To make any new changes to it, simply click on your machine's name.

Equipment

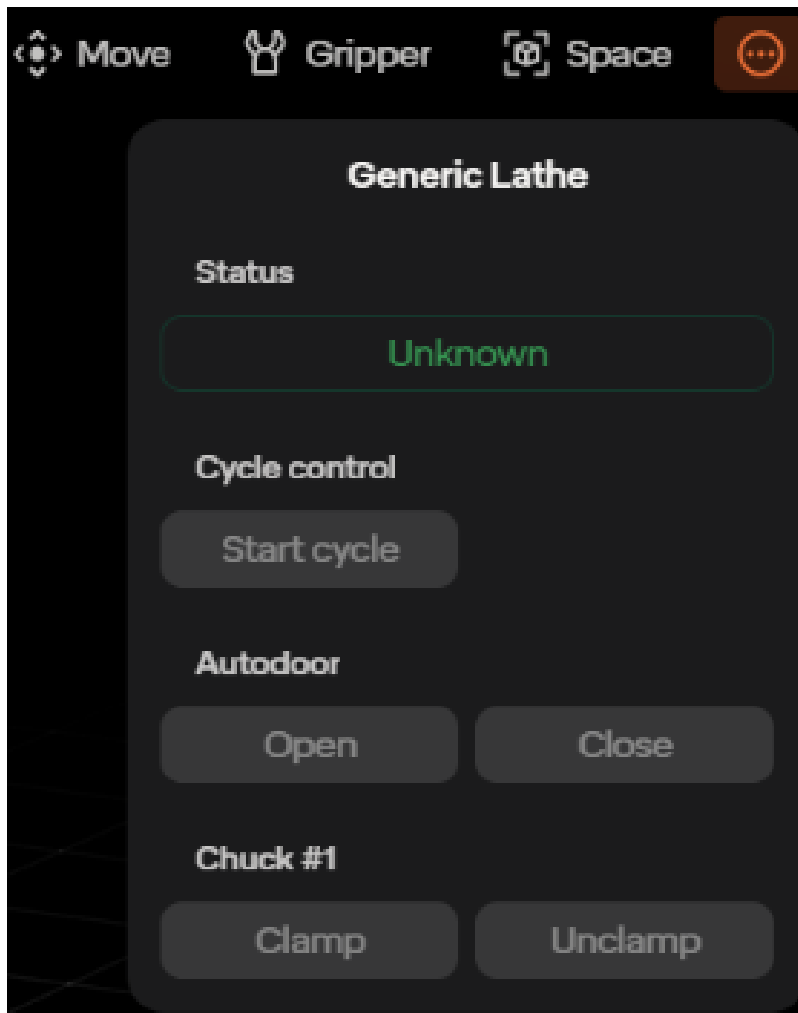
Generic Lathe

OnRobot 2FG7

Accessing CNC Machine:

Once your machine is added to your equipment list, you are able to access it on your move robot page. Once there, go to the three dots icon in the upper right corner of the page and select your machine.

Here you can see the status of your machine, as well as the ability to start the cycle, open and close the doors, and open and close your workholding equipment.



Wiring Your Robot Control Box to a Generic CNC Machine:

If you are using a generic CNC machine then you will have to hard wire your robot's control box to your CNC machine. This will be done by utilizing your control box's and CNC's inputs and outputs. And due to both machines being on separate commons you will need the use of relays in order for them to successfully communicate with each other.

Here is an example of how to connect you robot's control box to your CNC machine with the use of relays:

Robot and CNC machine relay diagram

Notes for Example:

Start Cycle= Robot Output 1
high

CNC Running= Robot Input 1
high

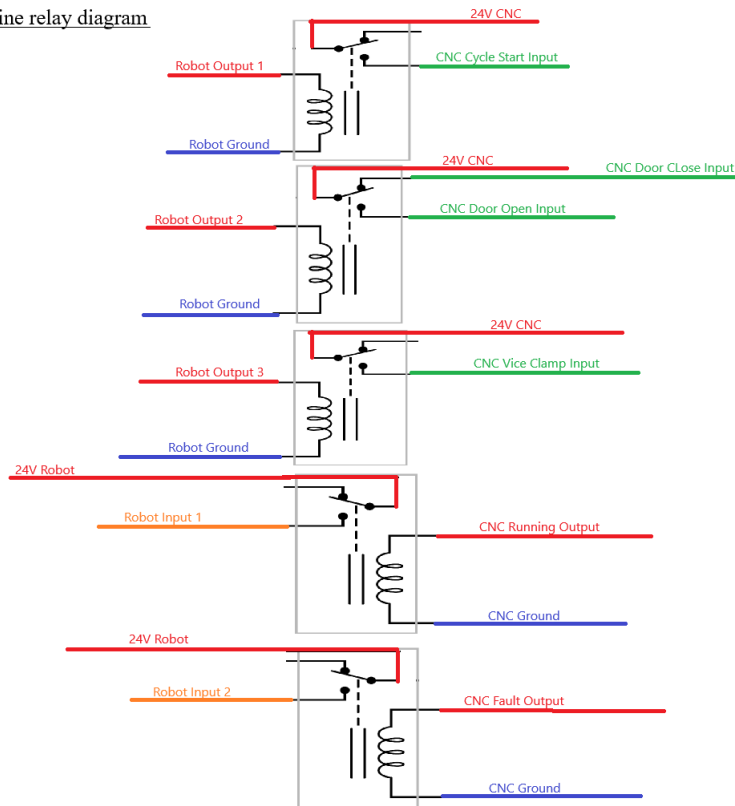
CNC Faulted= Robot Input 2
high

CNC Door Open= Robot Output 2
2 High

CNC Door Close= Robot Output 2
2 Low

CNC Clamp= Robot Output 3
High

CNC Unclamp= Robot Output 3
Low

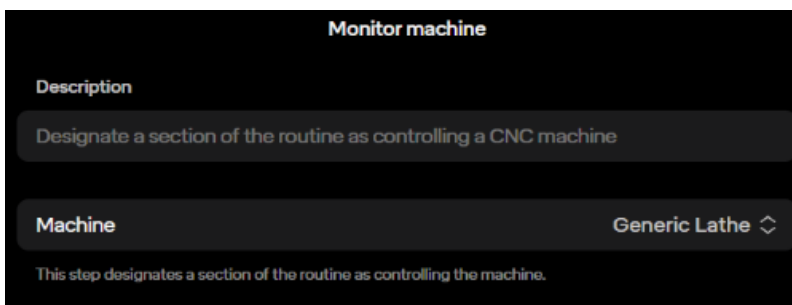


Creating a Routine- Generic CNC Machine:

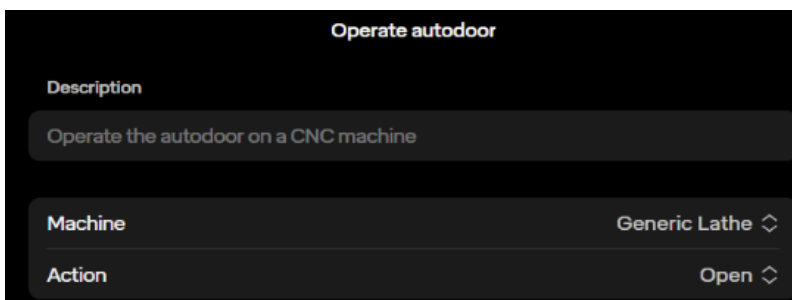
In the routine editor page you can find tasks that are made for a machine tending routine. These steps are: Monitor Machine, Operate Autodoor, Run Program, and Operate Workholding.



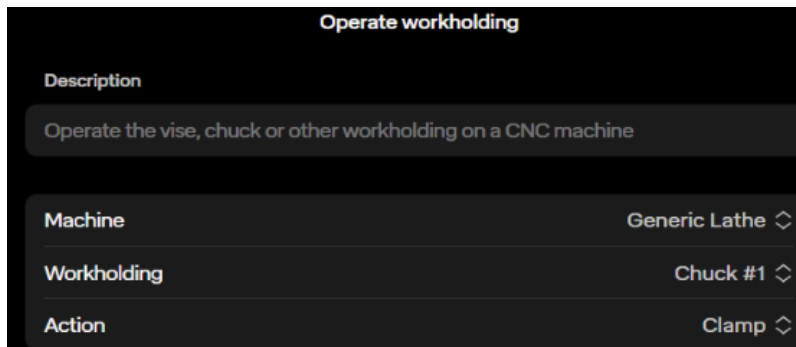
For the Monitor Machine step, you are able to choose the machine you created earlier. You dedicate this section of the routine to controlling your machine.



For the Operate Autodoor step, you can select the machine, as well as choose the action you want to perform with the door (open or close).



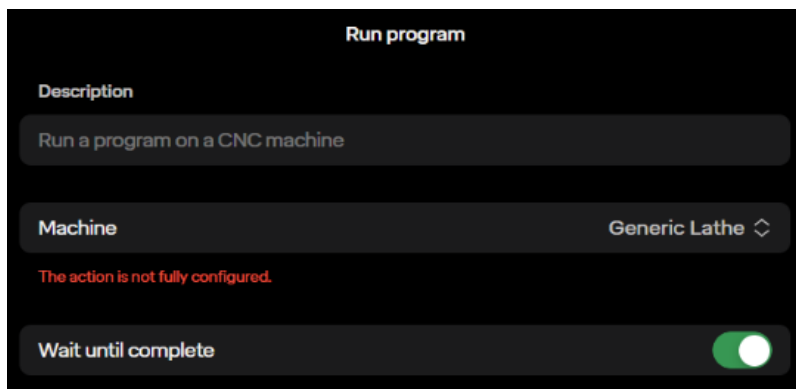
For the Operate Workholding step, you can select the machine, the workholding piece, as well as choose the action you want to perform with the workholding equipment (clamp or unclamp).



The screenshot shows a configuration window titled "Operate workholding". It contains a "Description" field with the text "Operate the vise, chuck or other workholding on a CNC machine". Below this are three rows of configuration options: "Machine" set to "Generic Lathe", "Workholding" set to "Chuck #1", and "Action" set to "Clamp". Each option has a dropdown arrow on its right side.

Field	Value
Description	Operate the vise, chuck or other workholding on a CNC machine
Machine	Generic Lathe
Workholding	Chuck #1
Action	Clamp

For the Run Program step, you are able to select your machine that you wish to run. You are also able to choose whether you want the robot's routine to pause until the machine is done.



The screenshot shows a configuration window titled "Run program". It contains a "Description" field with the text "Run a program on a CNC machine". Below this is a "Machine" dropdown menu set to "Generic Lathe". A red error message "The action is not fully configured." is displayed below the machine selection. At the bottom, there is a toggle switch for "Wait until complete", which is currently turned on (green).

Field	Value
Description	Run a program on a CNC machine
Machine	Generic Lathe
Wait until complete	Yes (toggle is on)

Creating a Routine- Haas Machine:

There are also Haas specific steps you can use if you are using a Haas machine. These steps are Monitor Haas Machine and Run Haas Program.

The Monitor Haas Machine step specifies a section of the routine to control the Haas machine. Here you have to enter the ethernet endpoint for the Haas. Here you can also choose whether you want Cell Safe on and if you'd rather use RS232 serial instead of ethernet.

Monitor Haas machine

Description

Specify a section of the routine as controlling a Haas machine accessible over Ethernet, being interrupted by errors from the device and sending a Cell Safe command

Address

Address not specified

Enter the ethernet endpoint over which the Haas device is accessible, e.g. `http://192.168.1.50:5551`. The haas-ip can be found in "Settings>Network" on the Haas control. The haas-port is the port set for "Machine Data Collection Port" (setting 148).

Cell safe

Send Haas cell safe signal when running nested steps. This allows the robot to control the Haas even when the door is open. For example, to open the door or open/close the chuck.

Use RS232 serial instead of ethernet

The Run Haas Program step runs a program on the Haas machine that's connected over ethernet. Here you will have to enter the program name for the Haas to call (ex: 900.nc). You will also have to once again enter the ethernet endpoint for the Haas. You can choose if you want Cell Safe on, to wait on this step until the Haas is complete, and if you'd rather use RS232 serial instead of ethernet.

Run Haas program

Description

Run a program on a Haas device connected over Ethernet

Program name

Enter the program name saved on the Haas to run, e.g. "900.nc". The programname must contain numbers only and end in .nc

Address

Address not specified

Enter the ethernet endpoint over which the Haas device is accessible, e.g. http://192.168.1.50:5551. The haas-ip can be found in "Settings>Network" on the Haas control. The haas-port is the port set for "Machine Data Collection Port" (setting 148).

Disable cell safe

☐

Disable cell safe during this program. With cell safe disabled, the program will not run if the doors are open.

Wait until complete

☒

If enabled, this step will not wait until the program is finished running before continuing the routine. If you disable this setting, use some way (such as a Wait step) to wait until the program completes before running a new Haas program.

Use RS232 serial instead of ethernet

☐

For more information on Haas Ethernet Integration, go to chapter 5.8.1.

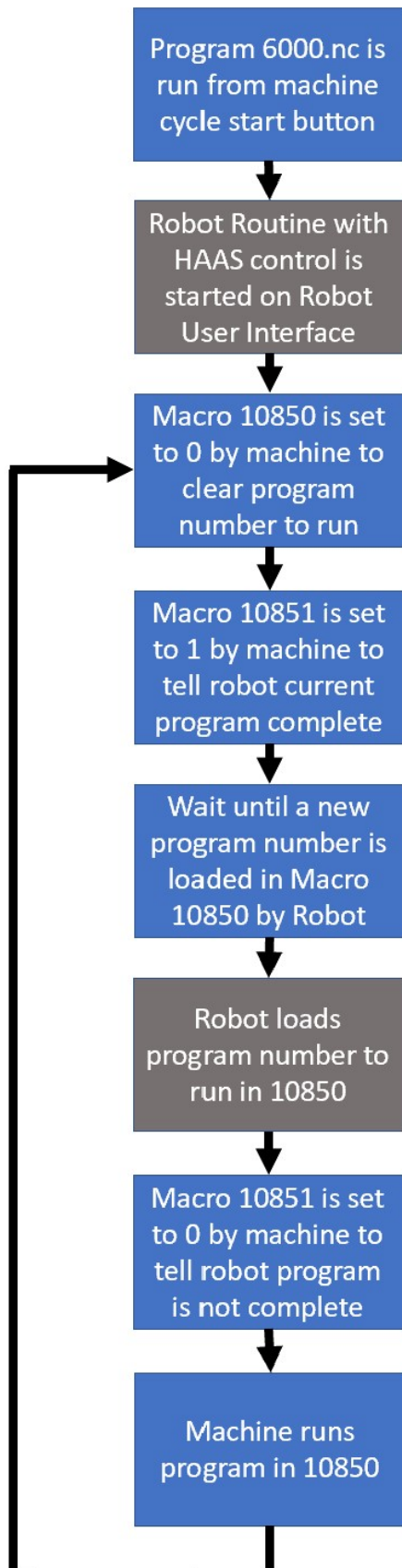
3.8 External IP Communication

3.8.1 Haas Ethernet Integration

Introduction

For Haas machines that support Next Generation control (generally 2017 or newer machines) Standard Bots provides a first-class integration suite that allows for ethernet communication between the Standard Bots robot and Haas machine. This allows for quicker setup and more flexibility with less wiring.

Standard Bots provides a set of sample .nc files to get started with an ethernet Haas integration. The integration works by running a main program on the Haas machine (in the example this program is 6000.nc). It then uses Macros 10850 and 10851 to complete handshaking between the Haas machine and Standard Bots robot as shown below:



Please be sure to review the comments in the sample code. G103 P1 is a required step in the 6000.nc main sample code to ensure the integration that happens through the macros functions correctly. Do not remove this step. If needed, add a G103 P0 as well as several empty lines after to the beginning the machine programs you create to run from the robot (called through macros from 6000.nc) to re-enable lookahead during those programs and ensure the fastest program speed. A P0 will disable block limiting, allowing the machine to look ahead as much as it wants. A number option will limit the number of blocks to look ahead.

Requirements:

- Haas machine with Next Generation control
- Network connection between machines (Either Wireless or Wifi)
- The “Cell Safe” signal must be held high using the “Monitor Haas Routine” step anytime the door is open. If the signal is dropped with the door open the Haas machine will go into Feed Hold and will need to be reset and then cycle start must be hit on the Haas again.
- HAAS sample programs and Standard Bot Routine Example Schema located at: <https://drop.ui.com/42515024-05a9-4d51-946f-d34295fb45ed>

Network Configuration:

In order to ensure a reliable solution the network between the robot and Haas machine must be configured correctly. Setting a static address without consulting the router/network configuration can cause duplicate IP addresses and other issues which will result in the robot not being able to run the Haas programs remotely. The Haas machine must have a known set IP address reachable by the robot for the integration to work through power cycles. There are several acceptable methods to setup a network between the robot and Haas machine:

1. Configure the Haas machine on a wireless network. The IP address must be set to static on the machine with an address outside of any DHCP range or must be configured to always have the same IP address through the router configuration.
2. Configure the Haas machine on a wired network with router. The IP address must be set to static on the machine with an address outside of any DHCP range or must be configured to always have the same IP address through the router configuration. Allow the robot to DHCP through the network router.
3. Run a single cable between the robot and Haas machine. The Haas machine must have a static IP address set manually. Contact Standard Bots Support for assistance setting a static IP address on the robot on the same subnet as the Haas machine.

Integration:

1. Ensure program numbers 6000.nc-6005.nc are available on the Haas machine.
2. Load Standard Bots sample NC files onto the Haas machine.

Settings And Graphics

Graphics Settings **Network** Notifications Rotary Alias Codes

Wired Connection Wireless Connection Net Share

Wired Network Information

Host Name	HAASMachine	DHCP Server	*
Domain		IP Address	*
DNS Server	*	Subnet Mask	*
Mac Address		Gateway	
DHCP Enabled	OFF	Status	UP

NAME		VALUE
Wired Network Enabled	>	On
Obtain Address Automatically	>	Off
IP Address		
Subnet Mask		
Default Gateway		
DNS Server		

Warning: Changes will not be saved if page is left without pressing [F4]!

F3 Discard Changes F4 Apply Changes

3. If using a wired connection, run an ethernet cable from the Ro1 cabinet to the Haas machine cabinet. The ethernet port on the Ro1 box is on the underside of the IO connection points in the box. The ethernet port on the Haas for Machine Data Collect is usually on the upper left when looking at the back of it. To get to the ethernet settings on the Haas navigate to Settings→Network→Wired/Wireless Connection.
4. Set the Haas machine up to communicate with the robot. The robot and Haas machine must be on the same subnet (3rd set of numbers in IP address) and be on the same wireless or wired network to communicate with each other. To ensure Haas can talk to the robot even through a power cycle the Haas must be set to a static IP address (no DHCP) or have network provisions made to ensure it is always the same IP address.
5. Enable machine data collect on the Haas (settings 143) and set the port to 5551.
6. If you are using a wired connection without DHCP, use the Standard Bots tablet application to set the IP address. You may need to log out of the User Interface by going to robot name menu in the bottom left then selecting Logout.
7. Load the sample “haas test” program provided by Standard Bots into the routine editor by going to Actions-> Edit Routine Schema and pasting the contents of the text file in. Then select “Create

Routine”.

The screenshot shows the 'Monitor Haas Machine' step configuration. The step is selected in a list on the left. The main configuration area has a title 'Monitor Haas Machine'. Under 'CELL SAFE', the 'Send Haas Cell Safe signal while running nested steps...' checkbox is checked. Below it, the 'Use RS232 Serial instead of Ethernet' checkbox is unchecked. The 'ADDRESS' field contains the text 'http://192.168.215.175:5551'. A small text box below the address field explains the format: 'Enter the ethernet endpoint over which the Haas device is accessible in the format http://haas-ip:haas-port, for example you might enter http://192.168.1.50:5551. The haas-ip can be found in 'settings->network' on the Haas control, and the haas-port is the port set for 'Machine Data Collection Port' (setting 148)'.

8. Edit the first step of the sample routine. Leave the “Send Haas Cell Safe” button checked. This allows the robot to run routines with the door open, which is often required (IE, when you need to close the door).
9. Update the IP address of the Monitor Haas Machine step to match the IP address of the Haas. Ensure the port remains at the end of the address. The format is: `http://xxx.xxx.xxx.xxx:port` . I.e., `http://192.168.215.175:5551`. Save the step.
10. Go back and edit Step 4, Run Haas Program.

The screenshot shows the 'Run Haas Program' step configuration. The step is selected in a list on the left. The main configuration area has a title 'Run Haas Program'. Under 'Enter the step description', the text 'to run the Haas test program (2020)' is entered. Below this, the 'PROGRAM NAME' field contains '2020.nc'. Under 'DISABLE CELL SAFE', the 'Disable cell safe during this program...' checkbox is checked. Below it, the 'WAIT UNTIL COMPLETE' checkbox is checked. Below that, the 'Use RS232 Serial instead of Ethernet' checkbox is unchecked. The 'ADDRESS' field contains the text 'http://192.168.215.175:5551'. A small text box below the address field explains the format: 'Enter the ethernet endpoint over which the Haas device is accessible in the format http://haas-ip:haas-port, for example you might enter http://192.168.1.50:5551. The haas-ip can be found in 'settings->network' on the Haas control, and the haas-port is the port set for 'Machine Data Collection Port' (setting 148)'.

11. The “haas test” program will attempt to run 2020.nc. In many Haas machines, this is the spindle warmup program. If needed, change this program number to a different program under “Program Name”.
12. Leave “Wait until Complete” checked. This will ensure the routine stays at the current step until the Haas has completed the program sent above under “Program Name”.
13. Update the IP address of the Monitor Haas Machine step to match the IP address of the Haas. Ensure the port remains at the end of the address. The format is: `http://xxx.xxx.xxx.xxx:port` . I.e., `http://192.168.215.175:5551`. Do not use 127.0.0.1 as that is the default loopback address. Save the step.
14. Start program 6000.nc on the Haas using the standard cycle start button on the Haas.

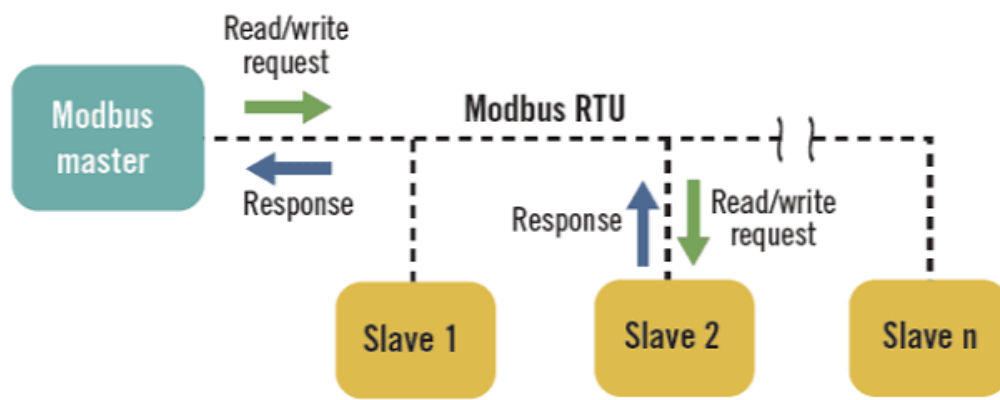
15. Start the sample program on the robot by running the routine.
16. The spindle warmup program (or alternative program chosen above) should now run. If it does not work verify the 10850 macro variable on the Haas is updating to the program number you put in the Run Haas Program step. If it is not, verify communication between robot and machine. The Haas machine must be running the 6000.ns]c program with no errors first, then you must start the robot program.

3.8.2 Modbus/TCP Client

One way for the robot to communicate with other devices is through Modbus, one of the oldest and most widely used communication protocols in industrial settings. Modbus provides a common language that enables different devices to communicate with each other, allowing devices from different manufacturers to work together. A Modbus Controller initiates communication and manages the data exchange between the compatible devices.

Modbus can operate over various physical layers, including serial communication methods like RS-232, RS-485, and RS-422, as well as Ethernet. Modbus TCP allows Modbus communication to occur over TCP/IP networks, enabling it to coexist alongside other Ethernet protocols.

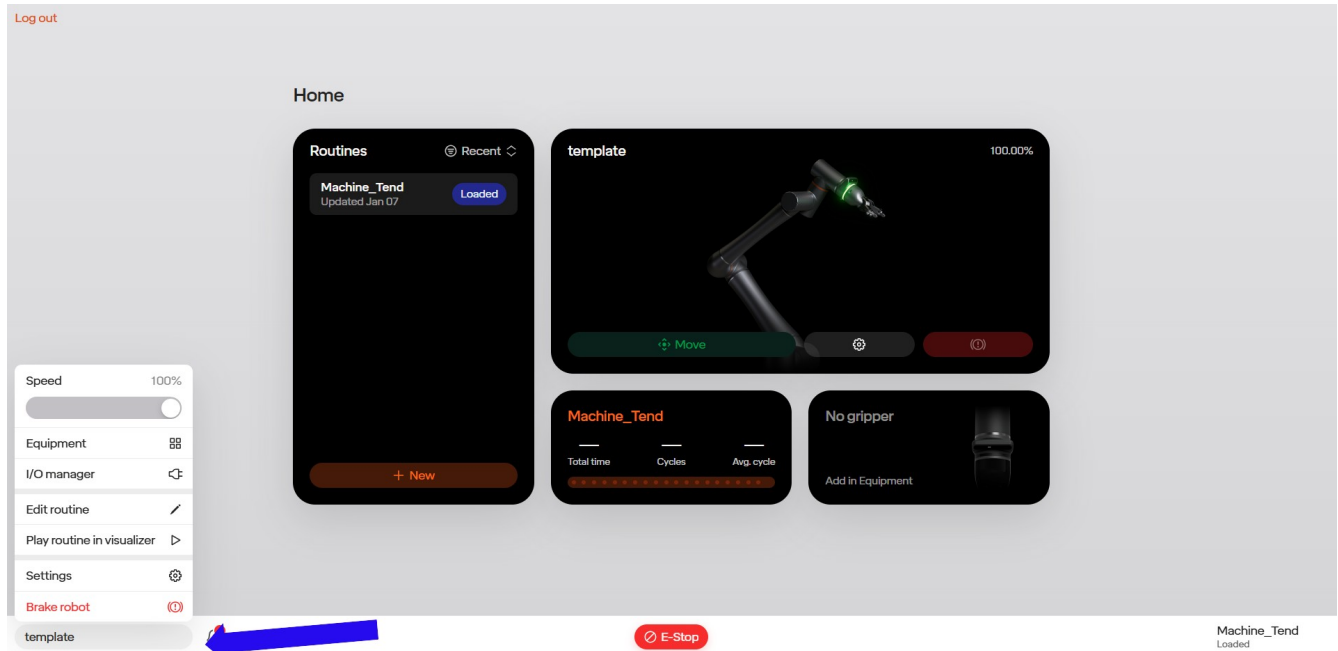
Please note: the robot is the client/master, so it would need to connect to the server/slave.



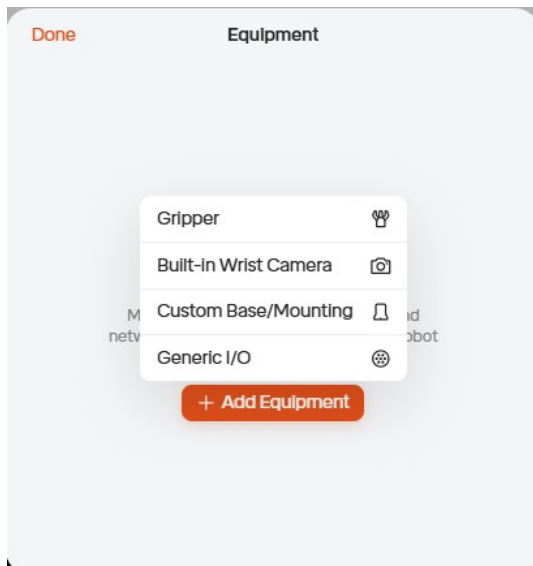
Setting up Modbus

Utilize the following steps to setup Modbus.

Step1) To setup Modbus communication with your RO1 click on the menu in the bottom left corner and select equipment.

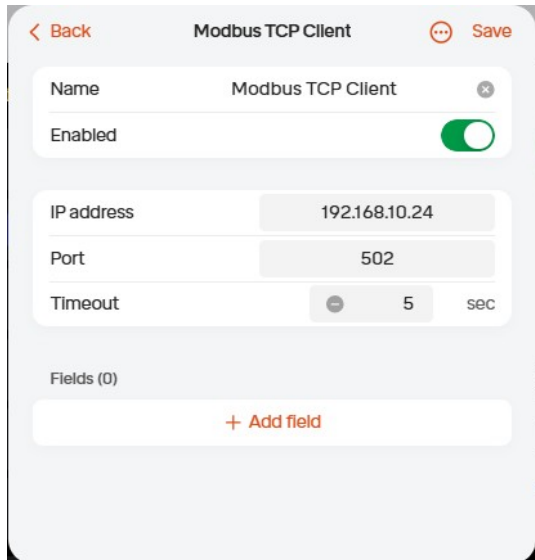


Step 2) Click on add equipment, generic I/O, Modbus TCP Client.



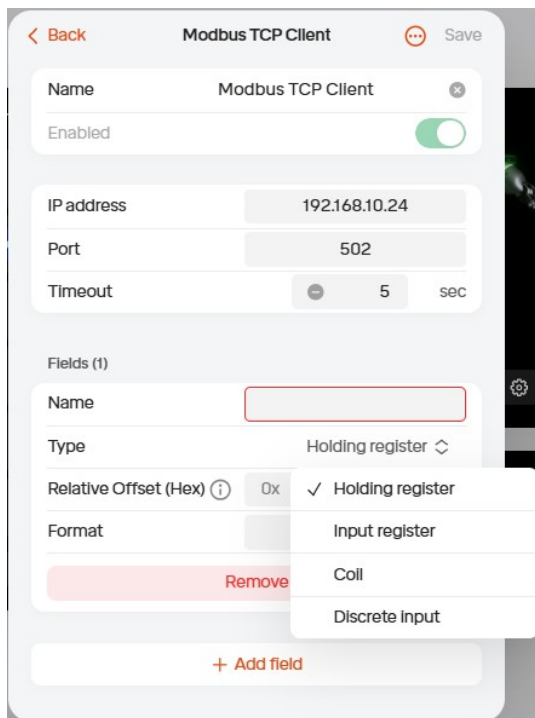
Step 3) Provide a Name for the Modbus TCP Client, IP Address, Port #, and Timeout.

Note: For this demonstration we will be using an IP address of 192.168.10.24, Port 502, as well as a 10 second timeout. This will vary based on your application.

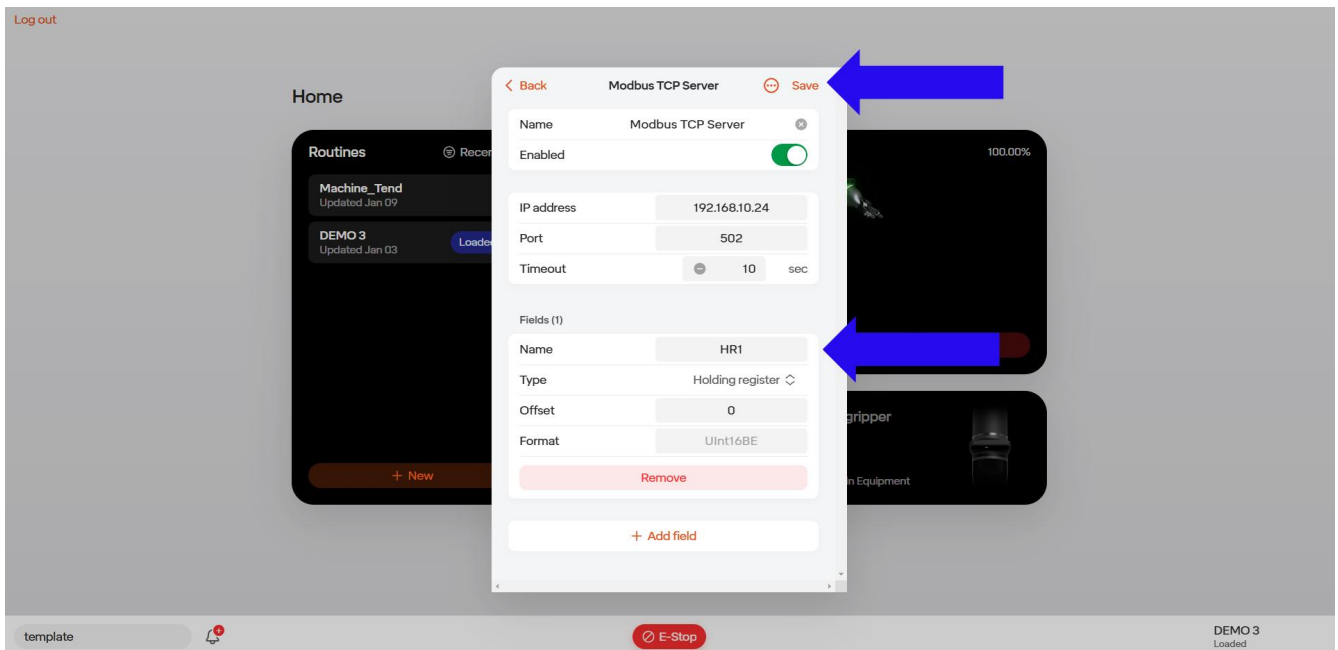
A screenshot of a mobile application interface titled "Modbus TCP Client". At the top left is a "< Back" button, and at the top right is a "Save" button with a red circular icon. Below the title bar, there is a "Name" field containing "Modbus TCP Client" with a close icon (x) to its right. Below the name field is an "Enabled" toggle switch, which is currently turned on (green). Further down, there are three input fields: "IP address" with the value "192.168.10.24", "Port" with the value "502", and "Timeout" with a value of "5" and a unit of "sec". Below these fields, it says "Fields (0)". At the bottom, there is a red button with a plus sign and the text "+ Add field".

Step 4) Click on add field and select the field type you would like to use. You can choose from the following types: Holding Register, Input Register, Status Coil, or Input Coil.

Note: For this demonstration we will be using a Holding Register.

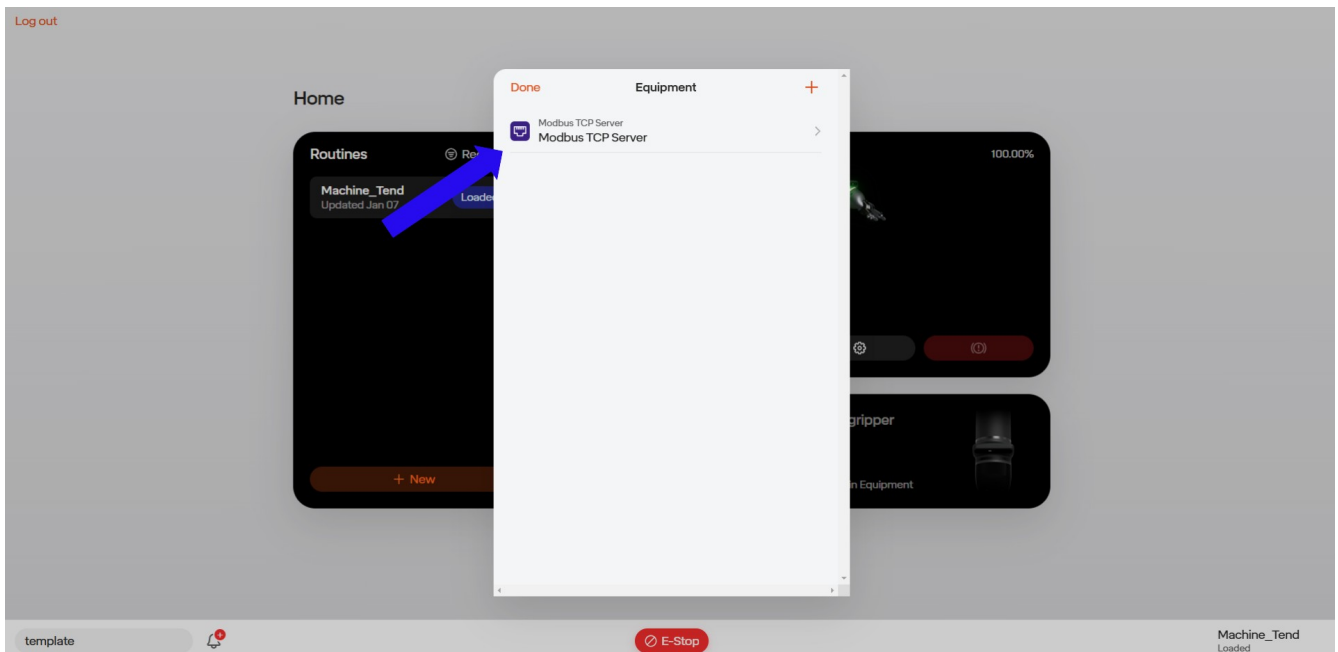
A screenshot of the same mobile application interface, but now with a field added. The "Fields (1)" section is visible. The first field has a "Name" field (empty), a "Type" dropdown menu set to "Holding register", a "Relative Offset (Hex)" field with a value of "0x", and a "Format" dropdown menu. A red "Remove" button is next to the field. At the bottom, there is a red button with a plus sign and the text "+ Add field".

Step 5) Provide a name for your field. When complete click save in the top right corner. Note: For this demonstration we will be naming the Field HR1 as we will be setting up a holding register. The offset will also remain zero for this demonstration.

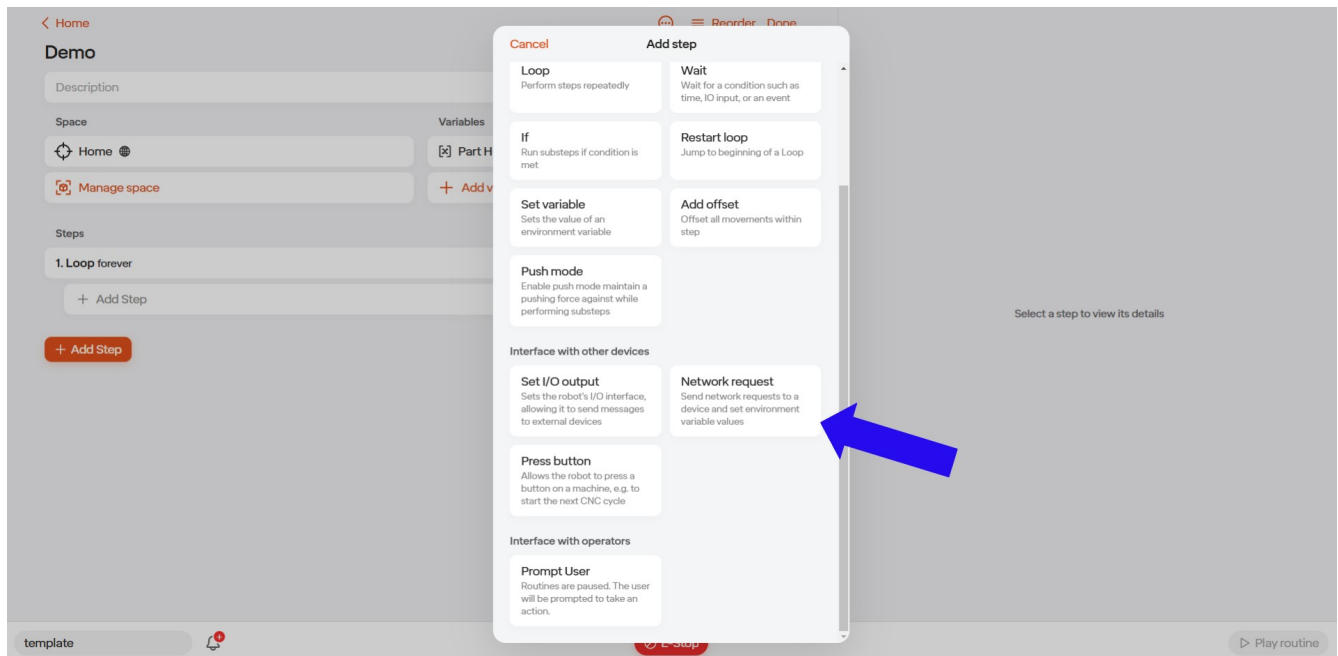


Note: To add additional fields repeat steps 4 & 5 as desired.

Step 6) Once you have selected save you should see your modbus tcp client added to the equipment list.

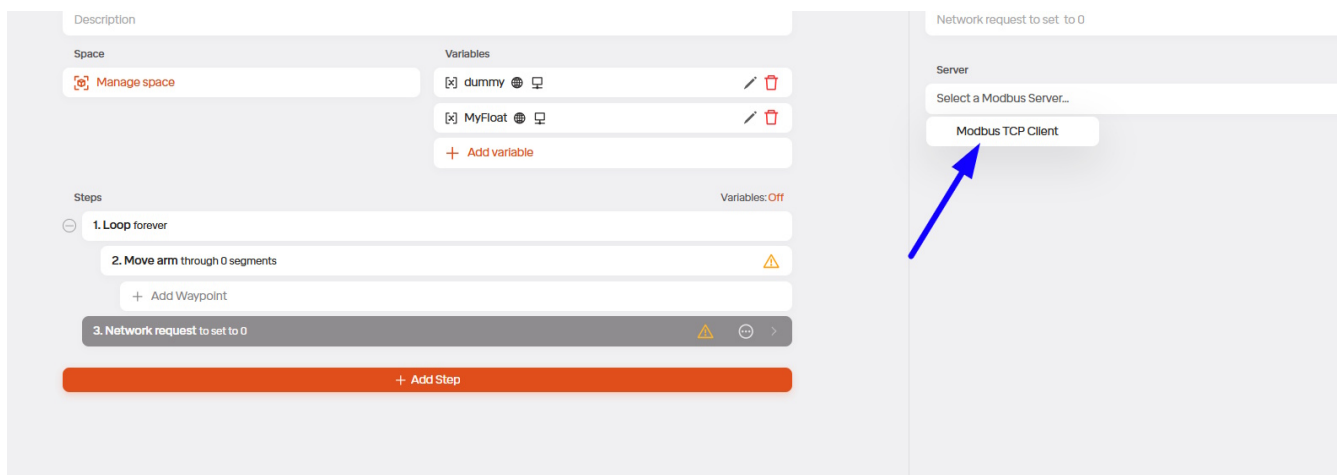


Step 7) Once you have added a Modbus TCP Client to your equipment you can now utilize it in a routine. Open up the routine and add a new step. Select “Network Request”.



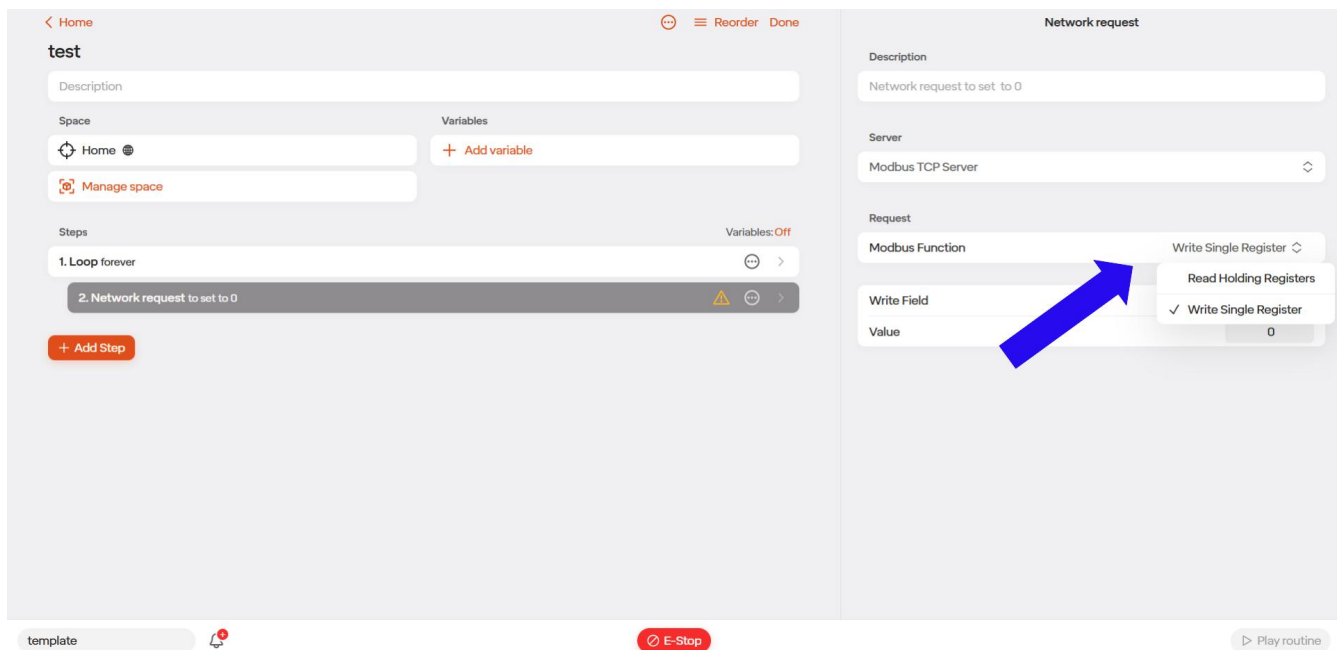
Step 8) Select which Modbus Client you would like to use.

Note: For this demonstration only one modbus client has been set up.



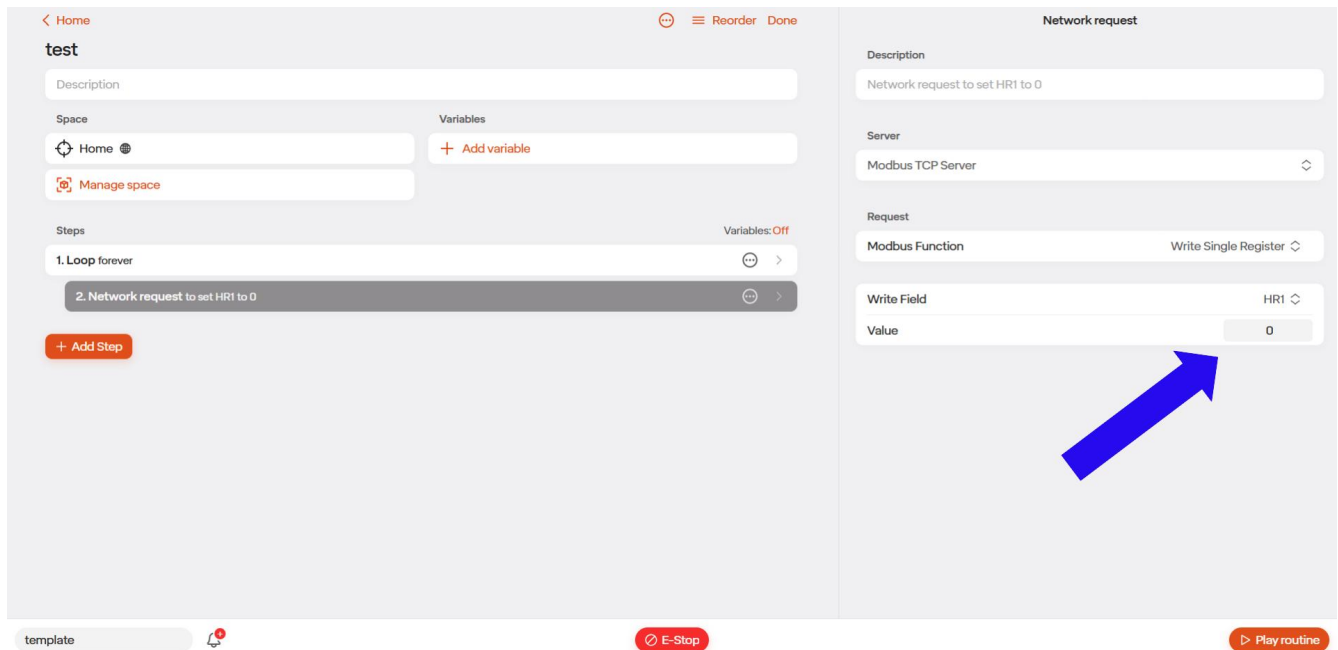
Step 9) Select the type of Modbus function you would like to request. The user can select either Read holding register or write single register.

Note: For demonstration purposes we will be selecting “Write Single Register”.



Step 10) Select what field you would like to write to, and what value you would like to write.

Note: HR1 was selected this was setup in step 4 & 5 We will be writing a value of 0 to HR1. This will be used to tell the external device that we are done with a particular step.



Step 11) Next we will repeat steps 7,8,&9 however this time we will setup a read holding register request. Our read field will be HR1 and we will want this to be sent to our "Receive In" variable.

Note: See *Creating a Global Variable* section for setting up variables.

< Home

⌵ ✓ Reorder Done

test

Description

Space

Home

Manage space

Variables

Receive In

+ Add variable

Steps

1. Loop forever

2. Network request to set HR1 to 0

3. Network request to save HR1 to variable Receive In

+ Add Step

Variables: Off

Network request

Description

Network request to save HR1 to variable Receive In

Server

Modbus TCP Server

Request

Modbus Function

Read Holding Registers

Read Field

HR1

To Variable

Receive In

template

E-Stop

Play routine



3.8.3 Modbus Routine Examples

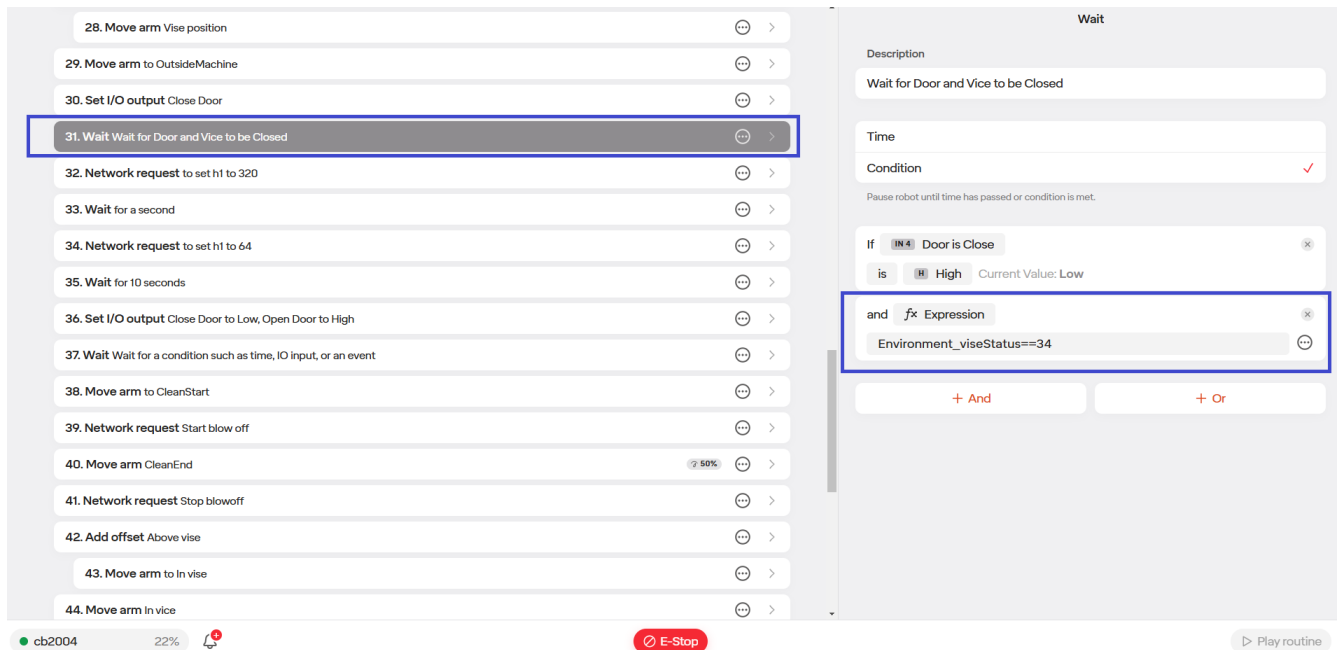
Reading from a Register Example

Note: This example demonstrates how Modbus can be employed to read a value from a single register and assign that data to a variable within the routine for subsequent use or decision-making. This approach proves beneficial later in the routine, as the value of the “vice status” variable can be leveraged to inform critical decisions.

Summary: Step 24 illustrates the process of reading the holding register, h1, following the closure of the vice in Step 22. The retrieved value is subsequently assigned to the Vice Status variable, as highlighted in Figure 1. *Note: Vice status variable was setup prior to adding the network request step.*

The screenshot displays a software interface for configuring a Modbus routine. On the left, a vertical list of steps is shown, with steps 22, 23, and 24 highlighted by a blue box. Step 24 is '24. Network request to save h1 to variable Vice Status'. On the right, a detailed view of the 'Network request' step is shown. It includes a 'Description' field with the text 'Network request to save h1 to variable Vice Status', a 'Server' dropdown set to 'Modbus TCP Server', and a 'Request' section. The 'Request' section contains three fields: 'Modbus Function' set to 'Read Holding Registers', 'Read Field' set to 'h1', and 'To Variable' set to 'Vice Status'. At the bottom of the interface, there is a status bar with a red dot and the text 'cb2004', a red 'E-Stop' button, and a red 'Play routine' button.

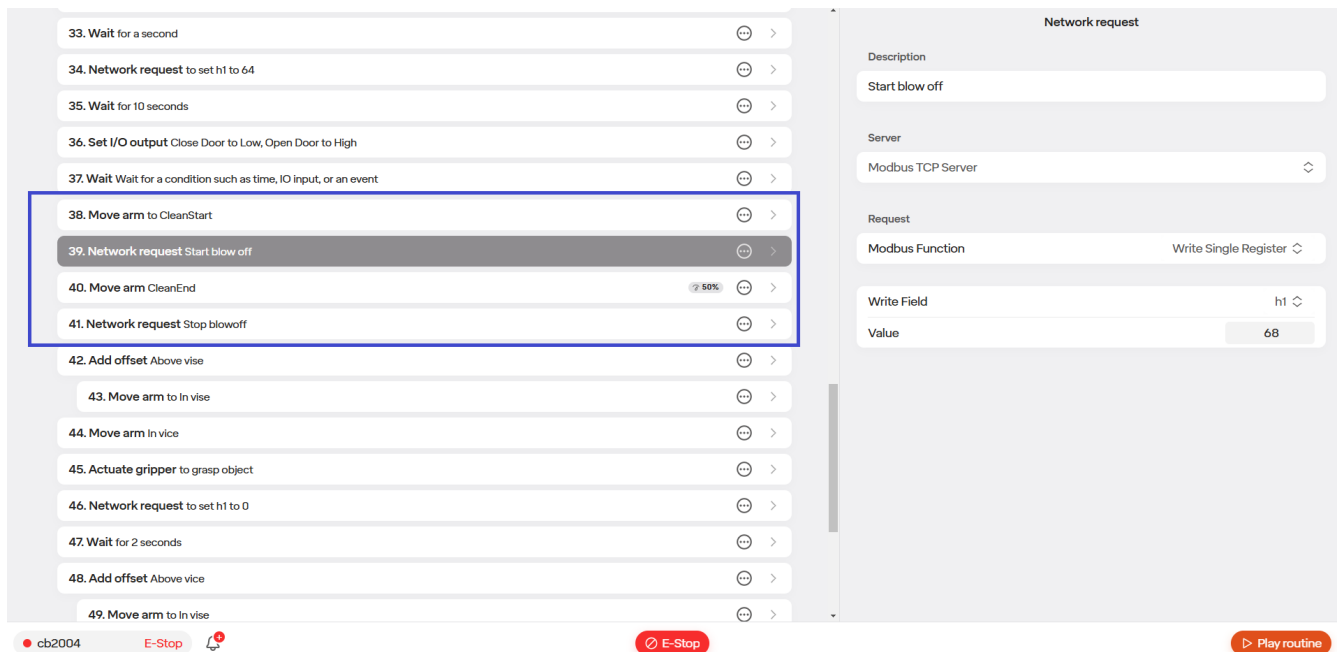
The retrieved value is then utilized in Step 31, in which the program is looking for the CNC Machine Door and Vice to be closed before continuing. For purposes of this example the value of 34 is used to indicate the vice is closed. This can be seen in Figure 2.



Writing to a Register Example

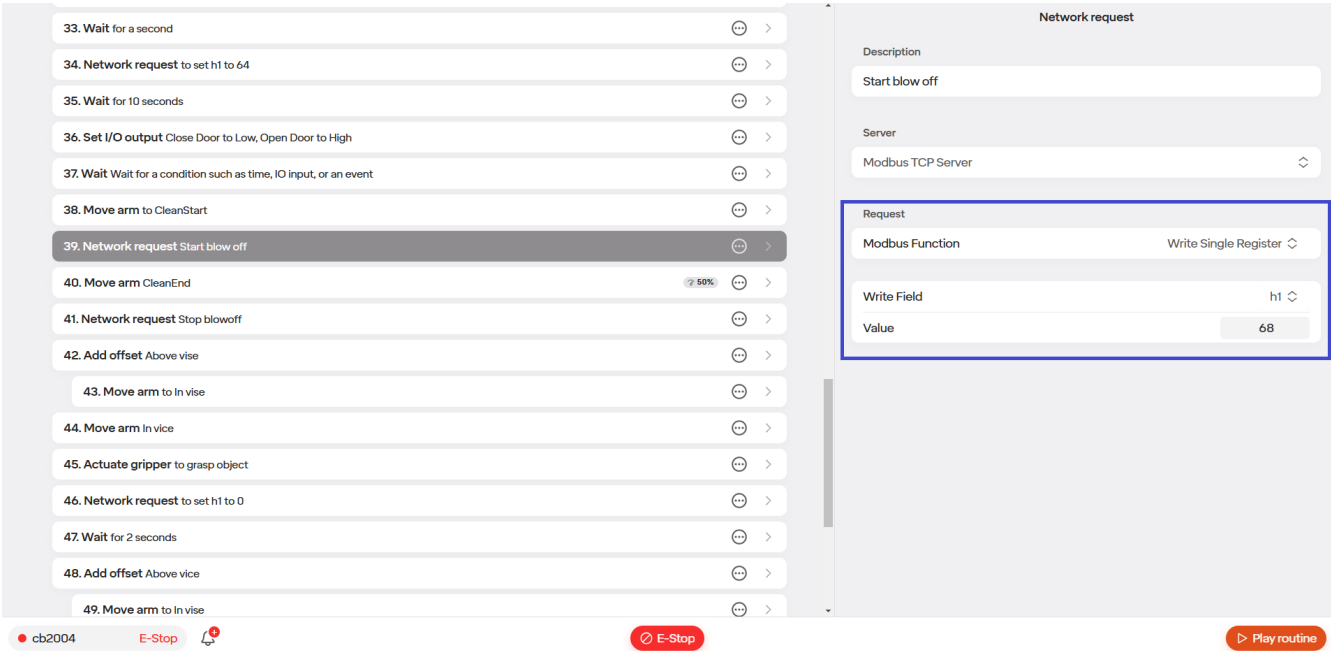
Note: This example illustrates a machine tending application, where Modbus is utilized to write a value to a single register for controlling an air blow-off nozzle mounted on the end effector. The primary function of the blow-off tool is to remove any debris from the part before the robot proceeds to pick up the finished component.

Summary: Steps 38-41 illustrate the process of writing a value to a holding register within a routine. This is clearly highlighted in Figure 1.



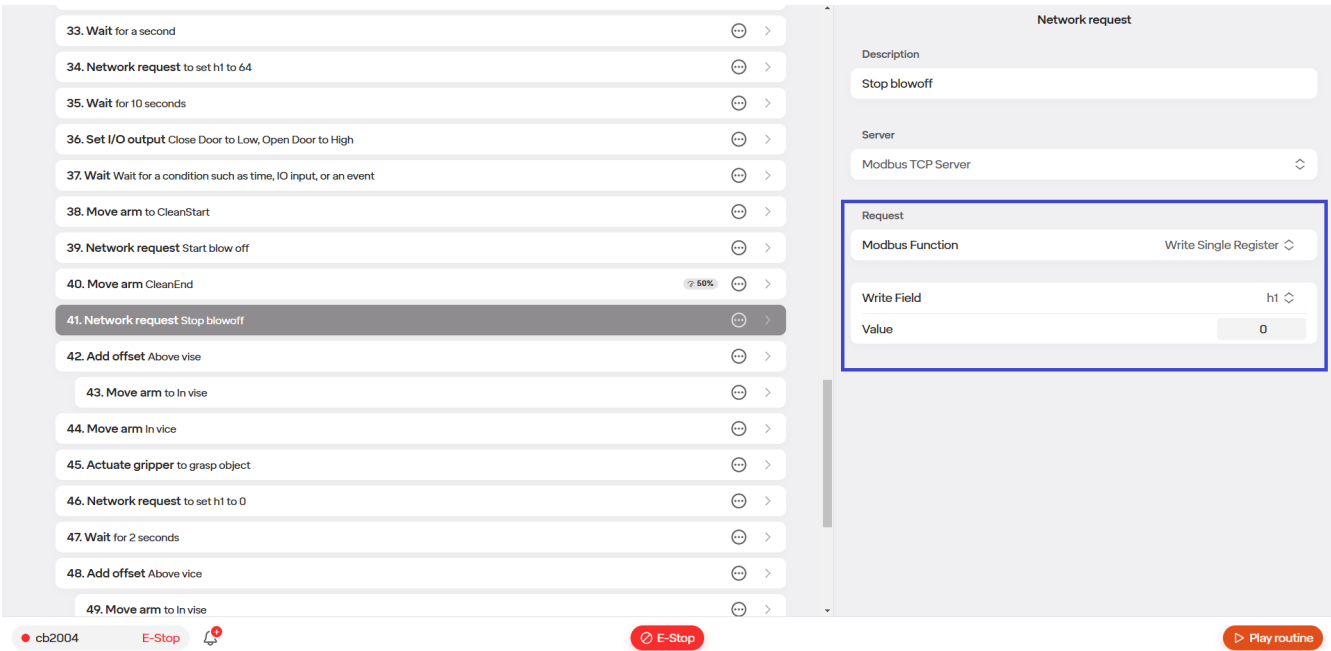
In step 39 a value of 68 is written to the holding register, h1, to turn on the valve for the blow off tool.

This can be seen in Figure 2.



The robot arm then moves to its end location in step 40.

In step 41 a network request is then made again to shut the valve for the blow off tool off again. This is accomplished by writing a value of 0 to the holding register h1. This can be seen in Figure 3 below.



3.9 Global Space & Variables

3.9.1 Introduction to Global Space Items.

(Note: section was written using software update version main 4608.)

The following section defines and explores use cases for global space items as well as global variable items.

Global Space Items

Global space items allow users the ability to access a saved robot position in space across all routines.

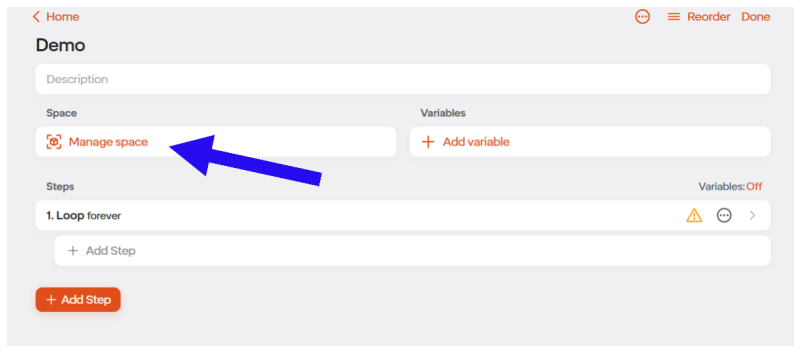
Types of Spaces

- Single Position
- Freeform Position List
- Grid Position List
- Plane
- Object

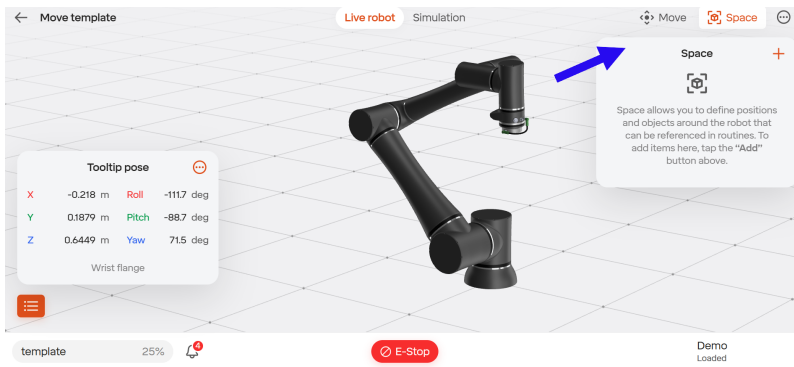
3.9.2 Creating a Global Space

Note: for this example we will be making a single position global.

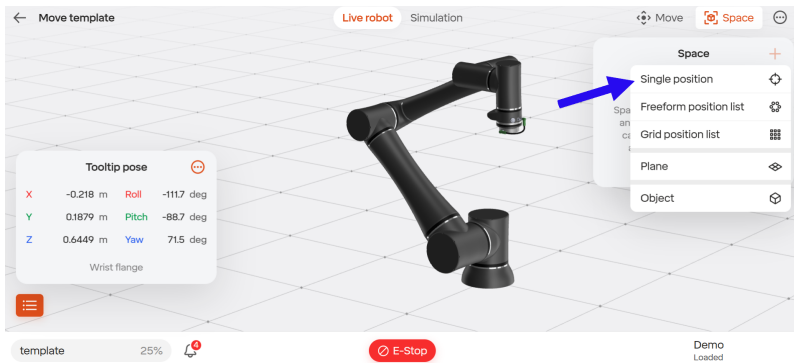
Step 1) In your routine click on “Manage Space”



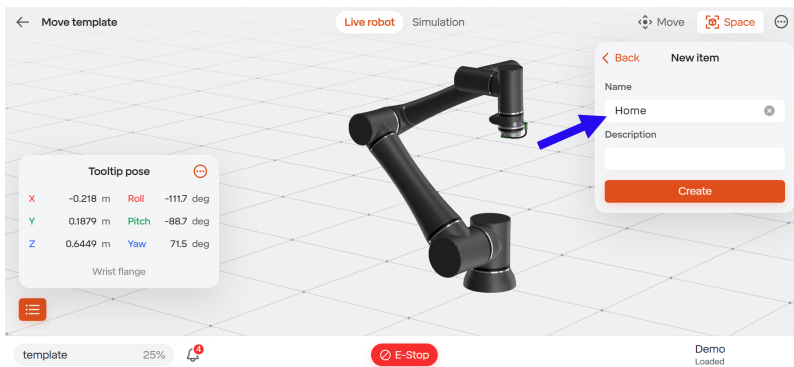
Step 2) Move the robot to the desired position and select “+” from there you will be prompted to choose the type of space you would like to add.



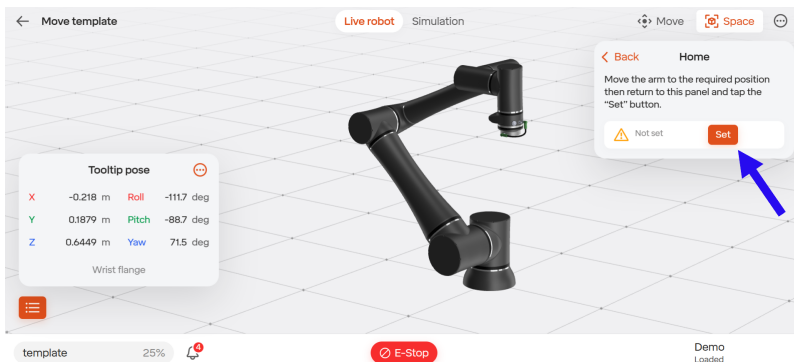
Step 3) Click "Single Position"



Step 4) You will be prompted to name your position. (For this example we will name the position "Home".)

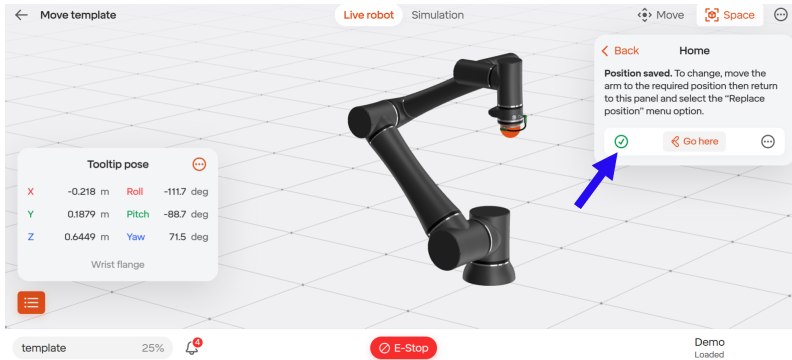


Step 5) After naming the space you will be asked to set its position.

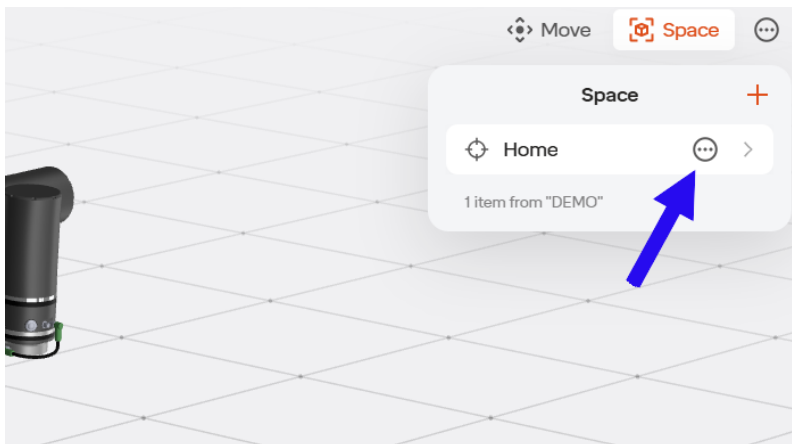


Step 6) Confirm the position was set, indicated by a green check mark that will appear on the left hand

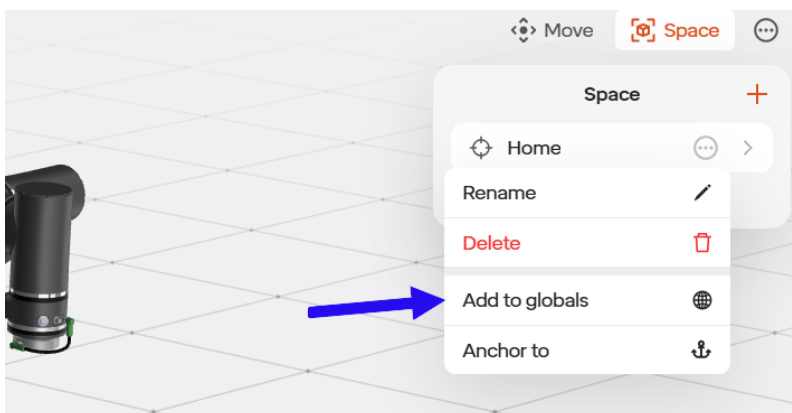
side.



Step 7) Once the space has been set, hit BACK found in the top right corner. This will take you back to the main space menu. Once you have returned to the main space menu, locate the space you would like to make global and click on the options for that space. The options button will be indicated by the ellipsis menu icon.

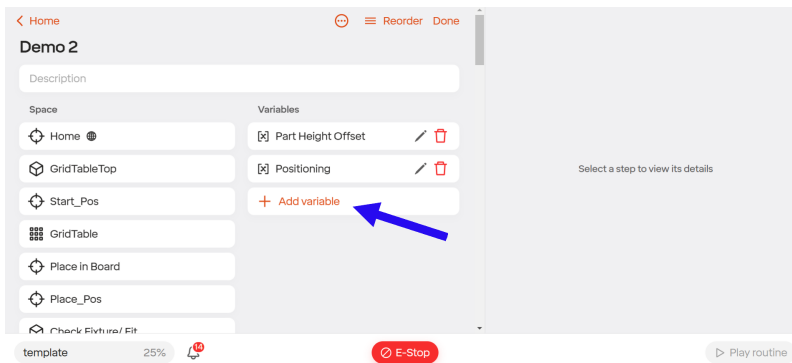


Step 8) Once in the menu drop down select "Add to Globals" and this space will be able to be accessed across all routines.



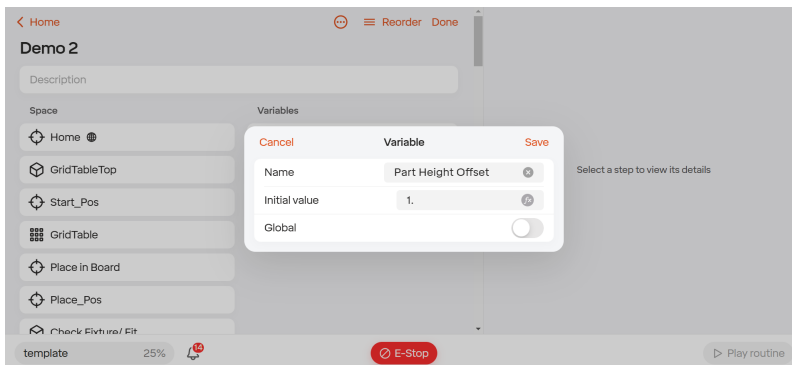
3.9.3 Creating a Global Variable

Step 1) In an open routine select add variable.

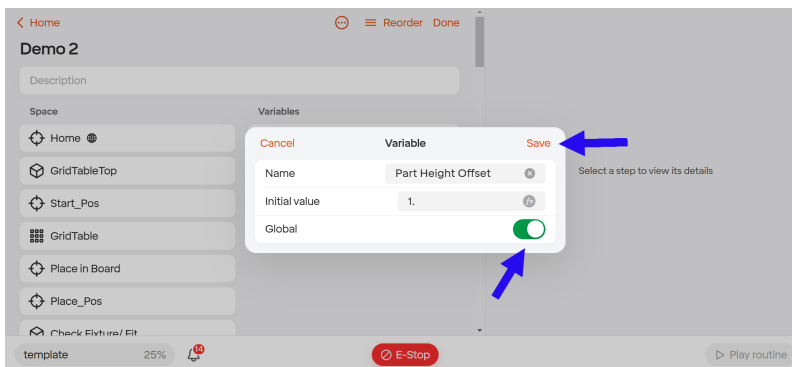


Step 2) Provide a name and an initial value for your variable.

Note: In this example we are making a part height offset and our initial value will be 1.



Step 3) To make this variable global click the switch that is labeled global. This should turn green. Click Save.



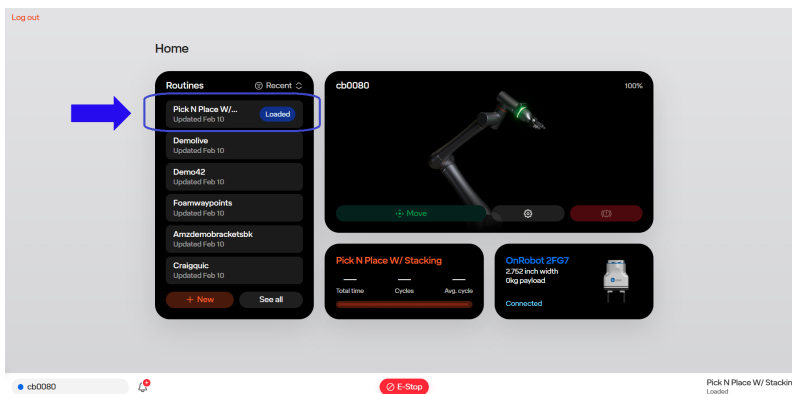
3.10 Exporting/Importing Routines

3.10.1 Introduction

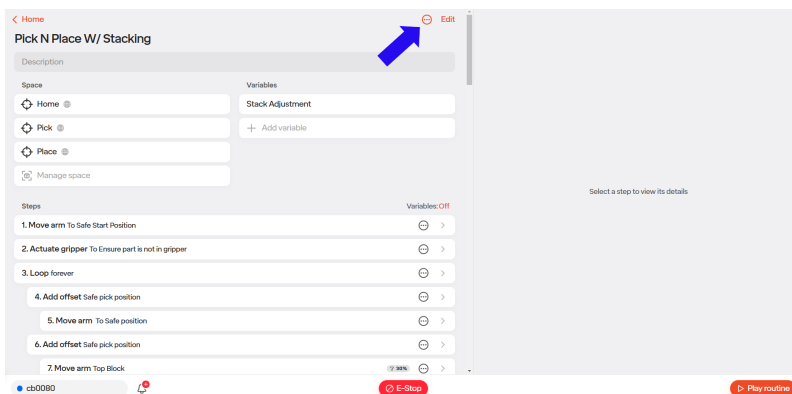
This section provides a detailed guide on exporting and importing routines to and from your robot. These operations can be performed either through the Standard Bots App or by directly accessing the robot via its web address.

3.10.2 Exporting Routines

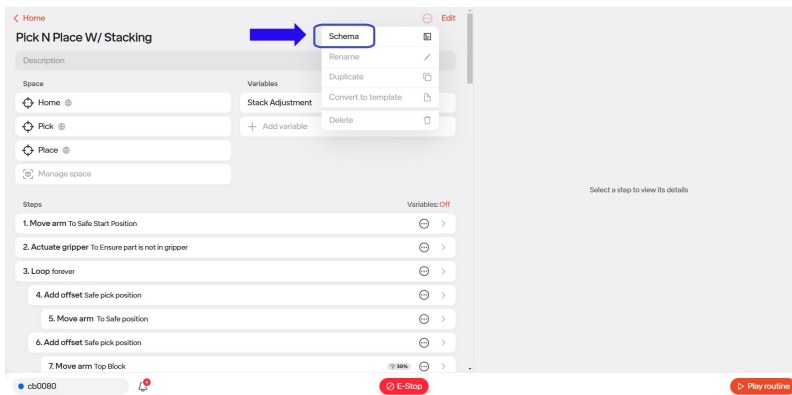
Step 1: Access the routine you wish to export. In this example, we will be exporting the routine titled “Pick N Place W/Stacking.”



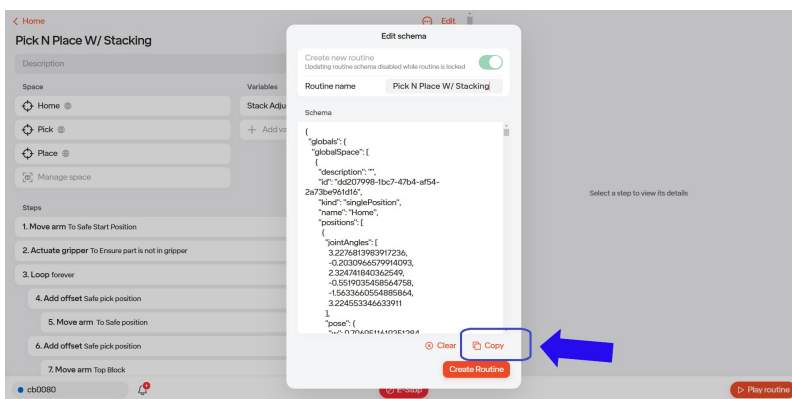
Step 2: After opening the routine, locate the “Options” button at the top of the page. Click this button to reveal a drop-down menu.



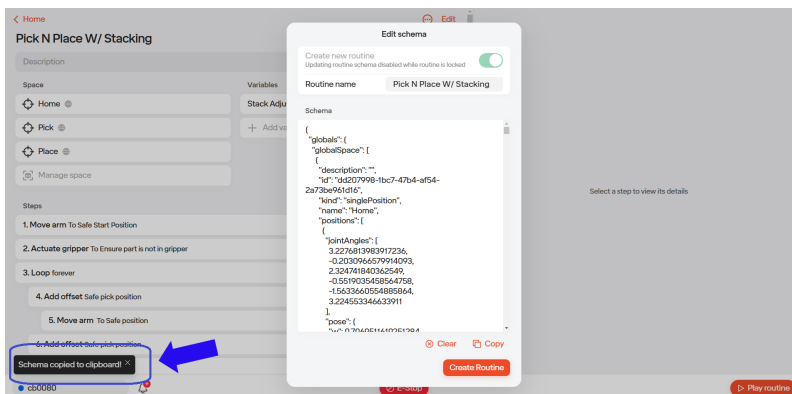
Step 3: From the drop-down menu, select “Schema.”



Step 4: Once the Schema is open, locate and select the “Copy” button. This will copy the Schema text file to your clipboard.



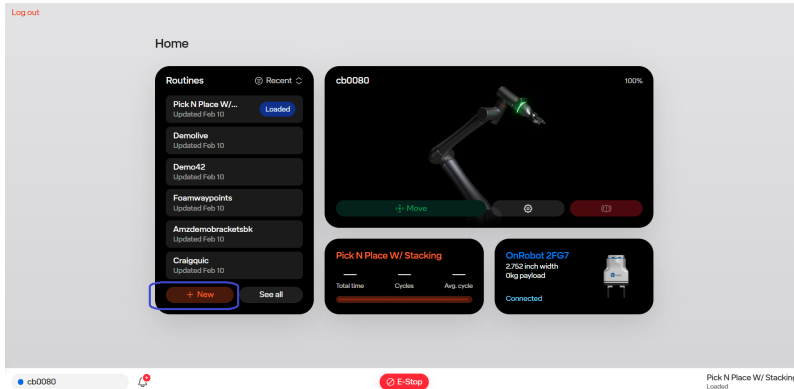
A confirmation message will appear in the lower-left corner of the screen once the selection is made.



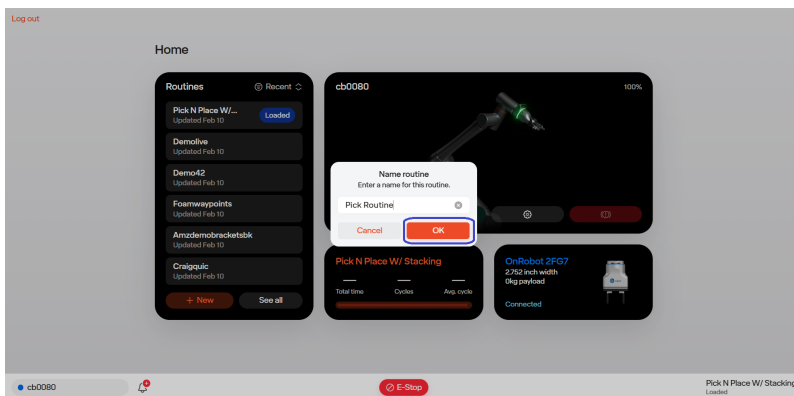
At this point, you can paste the text file into any word processing software, allowing you to save and store the file for future use.

3.10.3 Importing Routines

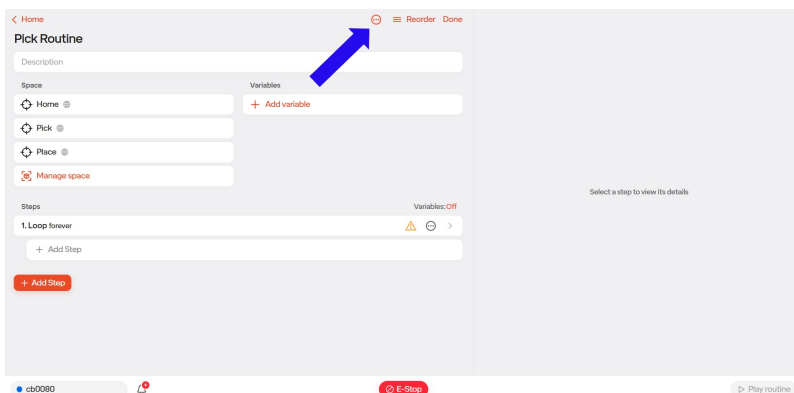
Step 1: From the main dashboard, click the “+ New” button.



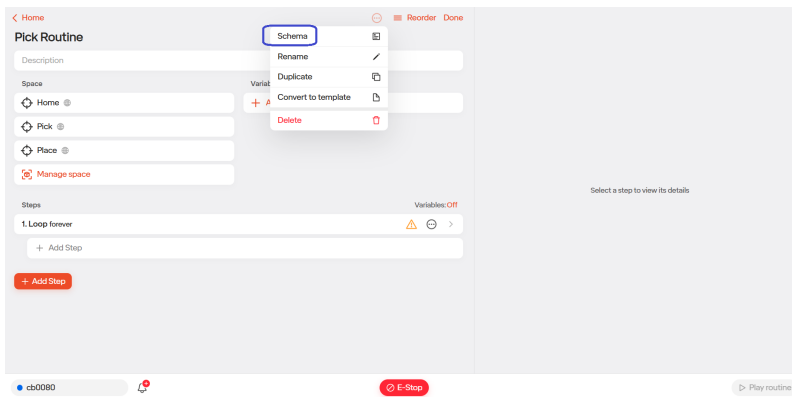
Step 2: Enter a name for your new routine and click “OK.” For this demonstration, we will name the routine “Pick Routine.”



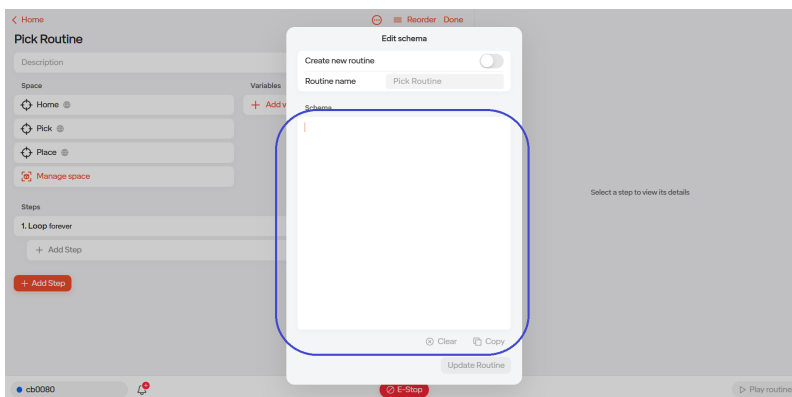
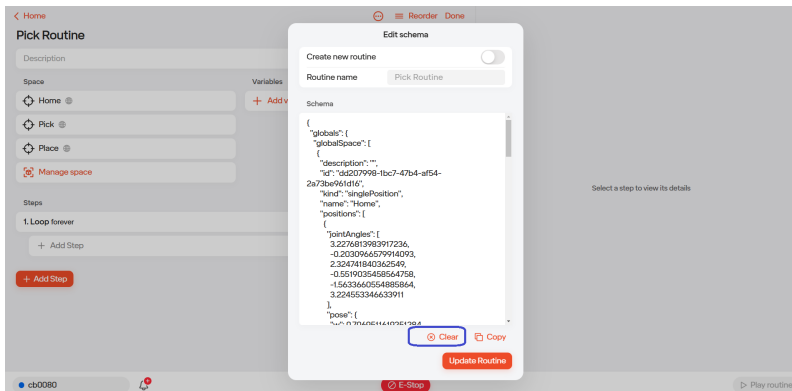
Step 3: With the new routine open, locate the “Options” button at the top of the page. Click this button to display a drop-down menu.



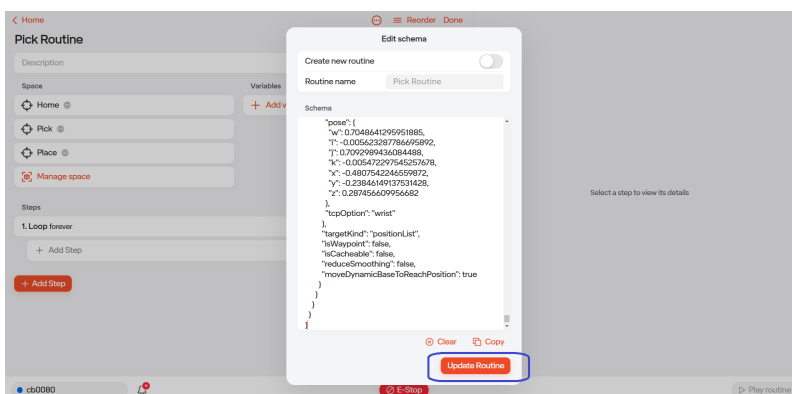
Step 4) From the drop-down menu, select “Schema.”



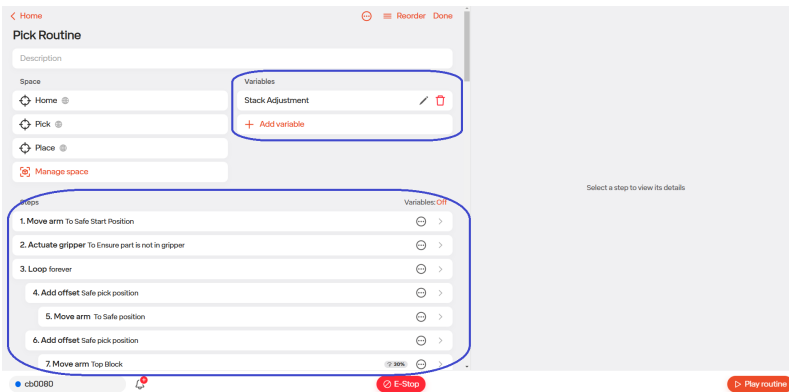
Step 5: When the Schema is open, locate the “Clear” button at the bottom of the popup and select it.



Step 6: After the Schema has been cleared, paste the desired routine into the Schema window and click “Update Routine.”



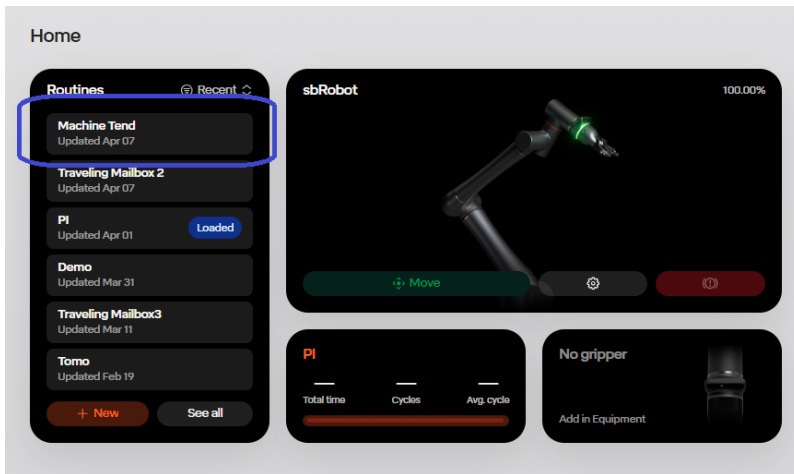
Step 7: After clicking “Create,” the routine will be updated, reflecting all spaces, variables, and steps from the imported schema. *Note:* In this example, global variables were used, which is why no visual change was observed.



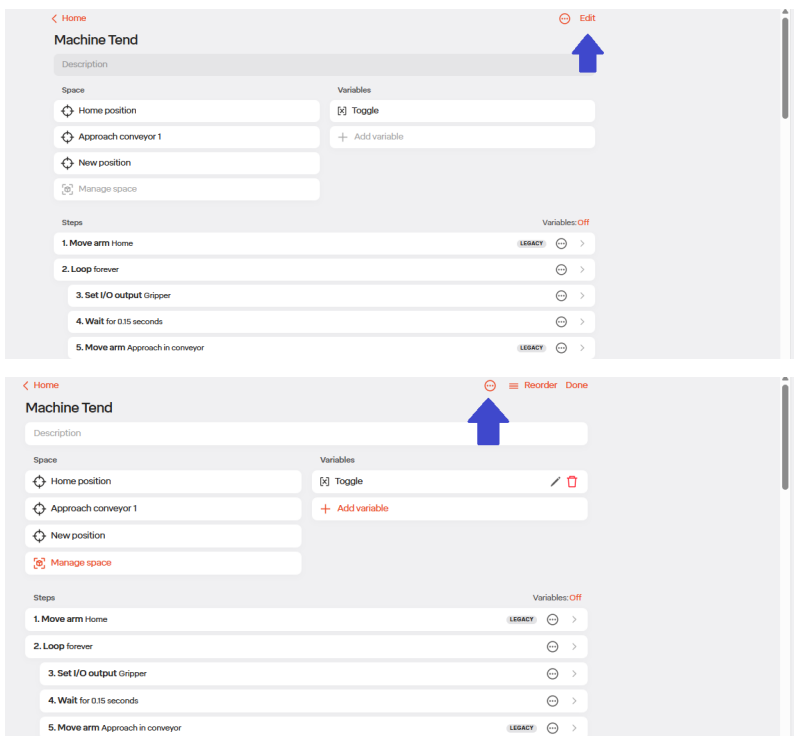
3.11 Updating Your Routine to Use V2 Move Steps

In this section we will discuss how to upgrade/migrate existing routines to use the improved waypoint functionality. *Note: This applies only to routines created prior to the release of the Waypoint Move feature V2. The Waypoint feature requires the feature flag to be enabled. If you need assistance activating it, or are unsure if it is already active please contact support at support@standardbots.com*

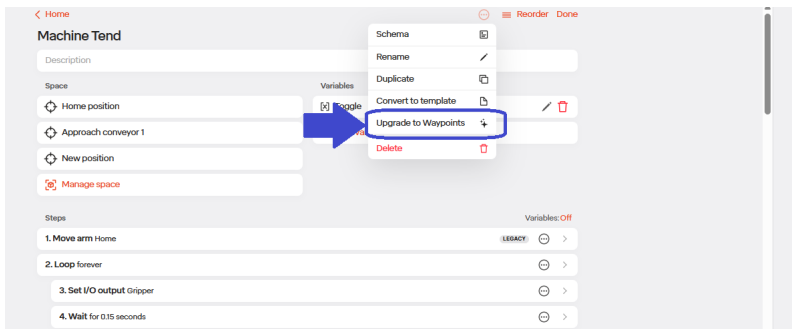
Step 1: Open the Routine Navigate to and open the routine you wish to upgrade. *Example: For this guide, we will use the routine titled “Machine Tend”.*



Step 2: Access Edit Options Click the “EDIT” button, then select the “Options” button located at the top center of the routine screen to open the dropdown menu.

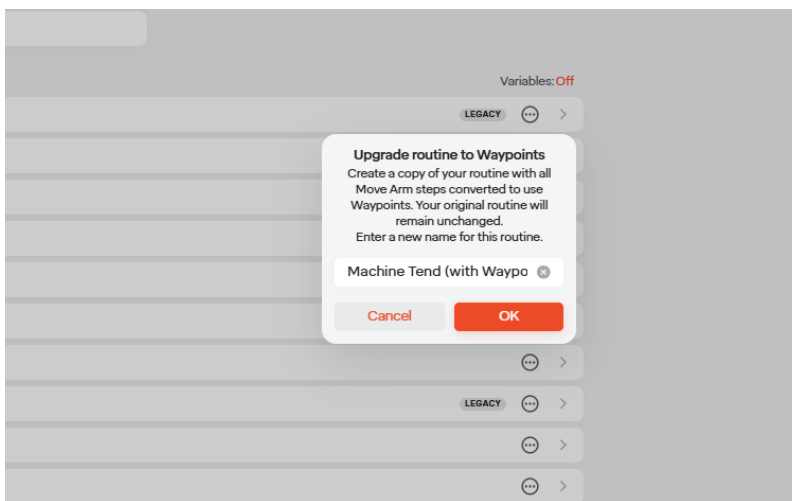


Step 3: Select Upgrade Option From the dropdown menu, choose “Upgrade to Waypoints”.



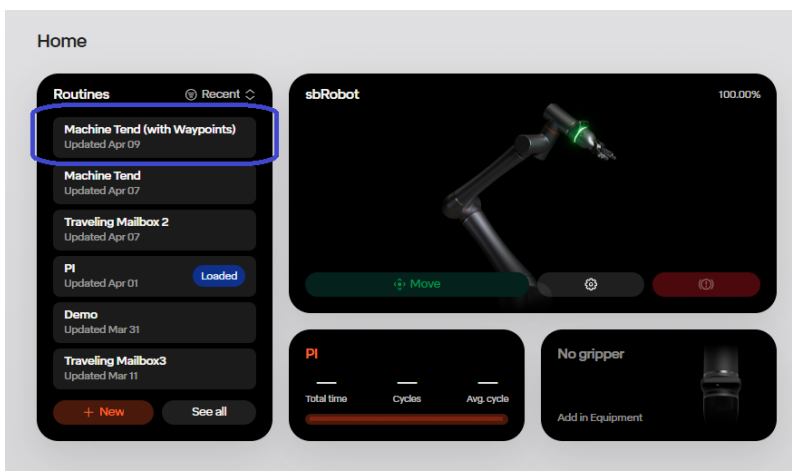
Step 4: Name the Upgraded Routine Enter a new name for the upgraded version of the routine.

Note: This process creates a duplicate of the original routine. The original will remain unchanged. Click “OK” to confirm.



Step 5: Locate the Upgraded Routine Return to the main dashboard. The newly upgraded routine will appear at the top of the routines list.

Note: If the original routine was created prior to the release of Waypoints V2, its steps may be labeled as “Legacy”. These labels will be removed automatically after the upgrade is completed.



3.12 Recovery Mode

To access the recovery mode through the user interface, follow the steps below: Navigate to the user interface. Within the routine click edit and select manage space. Tap the three dots menu located in the upper right corner. From the drop down menu, select recovery mode. Click the start recovery to begin the process. Once the system has entered recovery mode, you will be able to modify the live payload, if required.

Recovery

Change payload if required.

Live robot payload

-

0kg

+

↶

Save

Brake status				Torque	
✓	J0	-	0.2	+	
✓	J1	-	-25.5	+	
✓	J2	-	93.4	+	
✓	J3	-	22.3	+	
✓	J4	-	-90.6	+	
✓	J5	-	-174.2	+	

All joints recovered

Exit recovery

On the left side of the display is the brake status. Here there will be a green check mark once the robot is safely recovered. On the right side of the display is the torque bar. This is a quick visual indicator of how much load is on that joint. By using the “+” and “-” buttons beside J0, J1,J2, J3, J4, and J5 you will be able to recover your robot. These controls let you manually adjust joints to safely recover from

a fault, collision, or unexpected stop. Once all joints are recovered, you can exit recovery and resume normal operation.

3.13 Using Run in Background steps in Routines

The Run in Background step lets you execute a group of non-motion actions in parallel while the main routine continues to the next steps. It's ideal for things like timers, I/O handling, data logging, or condition monitoring that shouldn't block robot movement.

How to use it (UI)

Step 1: Open the routine and click **Edit**.

Step 2: In the step list, place the cursor where the background work should begin.

Step 3: Click **Add step** then choose **Run in Background**.

Step 4: A new **Run in Background** block should appear. Click inside the block.

Step 5: Add the steps you want to run concurrently (e.g., Wait/Delay, Set/Read I/O, variable updates, loops/conditions). - Keep motion out of this block (No Move Arm step).

Step 6: (Optional) Rename the block to describe its purpose (e.g., "Monitor Clamp Sensors").

Step 7: Add/continue steps after the block, these steps will run immediately while the background block is active.

Simple Examples: Run in Background

- Repeat until **part_detected = true** - Read DI **Part_Sensor** → if ON set **part_detected = true**
- Wait 100 ms

Main routine

- Move Arm to Pick Position
- Wait Until **part_detected = true** - Close Gripper
- Continue...

3.14 Palletizing with pallet base and boxes

This section explains how to set up a pallet base along with the pallet boxes you will need for palletizing.

Step 1: Add a Pallet Base Open your routine and click **Edit > Manage Space**. Click the **+** icon in the upper-right corner next to Space. Select **Pallet Base**. Enter a name and description, then click **Create**. Then define the pallet dimensions: adjust the corner position(X, Y, Z values) also you need to set the pallet size manually or use a preset pallet size. Once finished, click **Back** to return to the main space menu.

Position from Robot

Corner to position

R 

X

320

mm

Y

-600

mm

Yaw

0

deg

Pick the corner closest to the robot & infeed. The pallet will “grow” from this point as you adjust its size.

Floor-to-Pallet Distance

Z

0

mm

Pallet Size

Preset

None 

Length

1219

mm

Width

1016

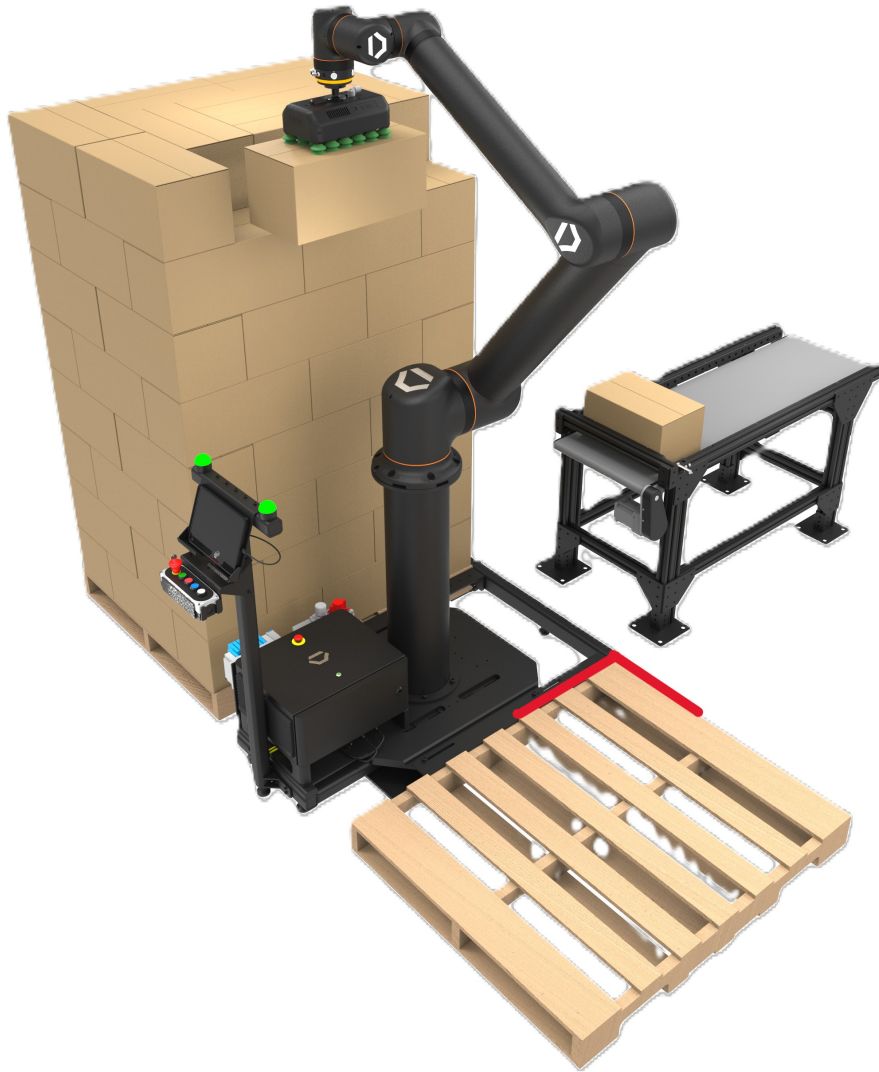
mm

Height

135

mm

In the image below, highlighted in red around the corner shows the corner location of the pallet relative to the base of the robot



Step 2: Add Pallet Boxes From the Space menu, click the **+** icon and select ***Pallet Boxes***. Enter a name and description, then click create. There are three different parts of this setup first you will need to configure the ***set up***: First you will select the correct pallet and pallet size under the ***parent*** section. Then under the ***Box approach distance*** you will then set the approach distance which is how far the approach waypoints are from the target box.

< Back Pallet Box Left

Setup

Boxes

Layers

Parent

Pallet base

Left Pallet 

Size

1016 mm x 1219 mm

This is only used to determine the size for patterns and where the layers will be visualized here. In the robot's routine, a pallet may be built on any base.

Box Approach Distance

XY (Plane)

50

mm

Z

350

mm

Approach distance specifies how far the approach waypoint is from the target box, with XY applied independently in each horizontal direction per the chose pattern, and Z applied vertically.

Define Box Properties Click the ***Boxes*** tab in the middle of the menu, then click the ***+*** icon in the

upper-right corner. Here you will then need to define the dimensions, weight, and pickup positions for boxes (SKUs) that will be placed in the selected pallet. Before defining the dimensions be sure to have the robot in the pick position above the conveyor to be sure the box is in the right orientation. When defining the box Dimensions you need to remember that dimension ***A*** is the side of the box that will be parallel with the EOA camera while dimension ***B*** is the side of the box that will be perpendicular to the camera and dimension ***C*** will be the height of the box.



Align the gripper tooltip to the box's  top center. The boxes "A" dimension is defined by the robot  camera.

Pickup Position



Go here



Dimensions

A 420 mm

B 300 mm

C 300 mm

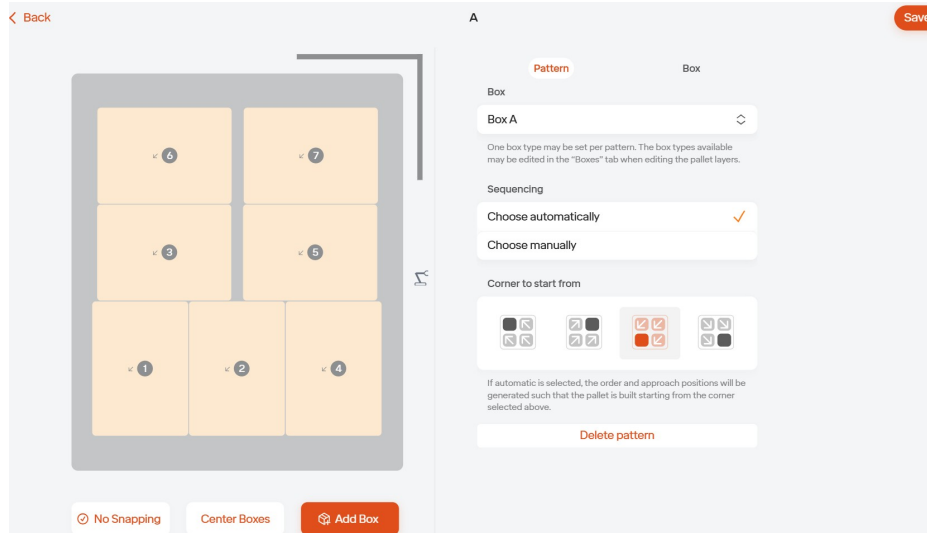
Weight 1 kg

Advanced

A offset 0 mm

B offset 0 mm

Create Box Layers Go to the ***Layers*** tab and click the ***+** icon under patterns to create a new pattern. Once here you will first need to select the correct box you will be using for this certain pattern. Then pick the sequencing in which to place the boxes automatically or manually along with selecting the corner in which to start from. Once this is done you can begin to click the ***Add Boxes*** button on the bottom left side of this window, you can also use the ***No Snapping*** button to freely move the boxes, if not they will automatically be placed next to each other at whatever distance that you have set.





Align the gripper tooltip to the box's  top center. The boxes "A" dimension is defined by the robot  camera.

Pickup Position



Go here



Dimensions

A 420 mm

B 300 mm

C 300 mm

Weight 1 kg

Advanced

A offset 0 mm

B offset 0 mm

Finalize the Layout In the ****Layers**** tab, drag and drop each ****pattern**** into the ****pallet layout**** in the order of which the boxes will be placed on the pallet. Once all layers are placed, your pallet base and boxes are fully configured and ready to go.

 **Back** **Pallet Box Left**

Setup

Boxes

Layers

Patterns (4)

A

B

C

D

+

Pallet Layers (3)



2

Pattern A



1

Pattern B



0

Pattern A



Create patterns then drag them to the list above. Layers listed on top will be on top in the pallet.

Once this is finished you should move the robot to each box position to be sure the robot can reach and

so the software can validate that the robot will be able to reach each position correctly.

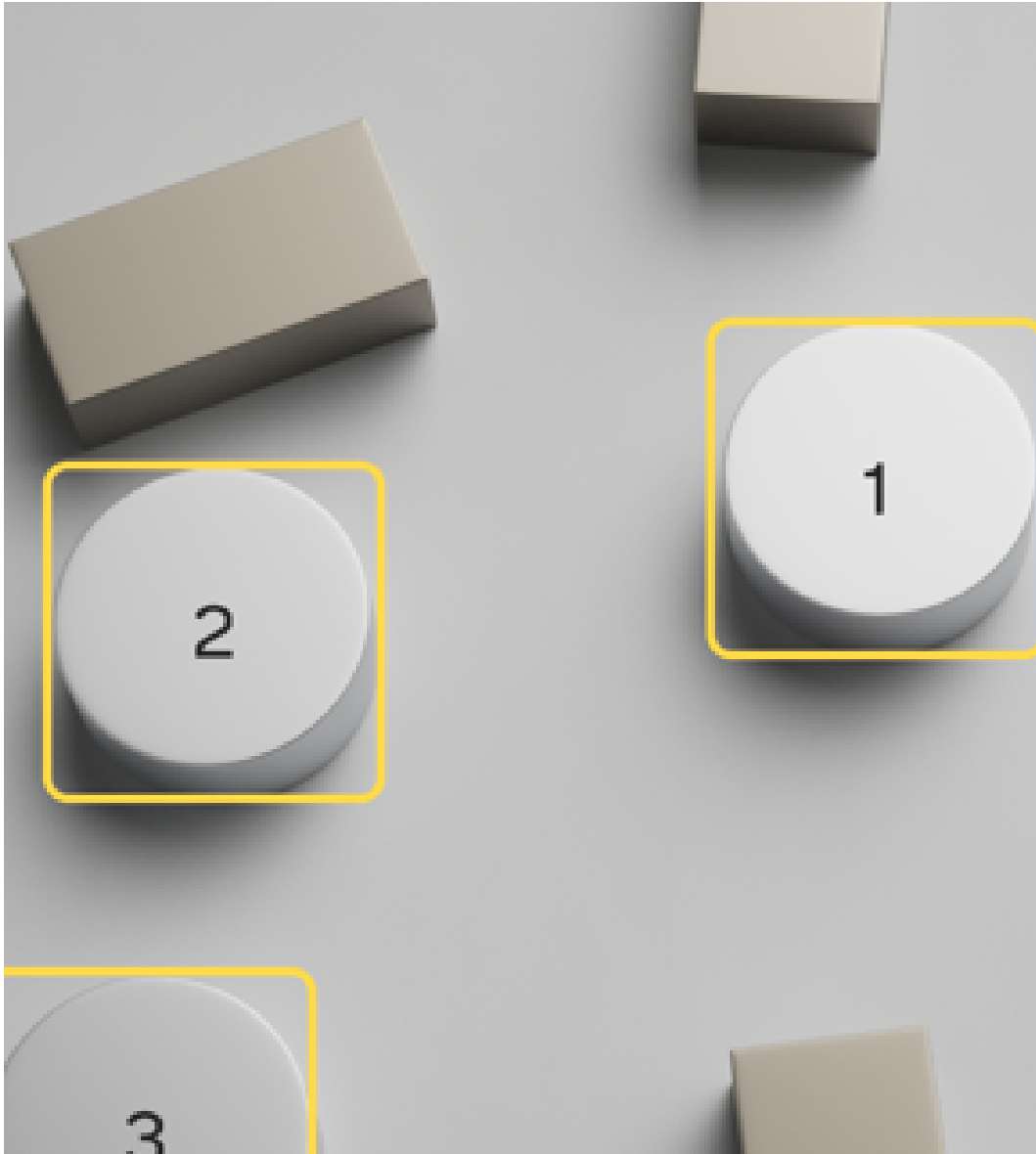
Chapter 4

Vision

Note: Vision is an optional paid feature. If you are interested in incorporating this into your robot, please contact Standard Bots for further assistance.

- **Email:** live-support@standardbots.com
- **Phone:** 1-888-9-ROBOTS

4.1 Locate



4.1.1 Use Cases

Standard Bots Locate feature is designed to locate items on a 2D flat plane to allow for picking and placing them using the integrated wrist camera. Items to be picked do not need to be organized but must be distinguishable from each other. Locate is not designed for 3D applications (stacked items). 3D functionality will be an optional upgrade provided at a later date. Locate accuracy is dependent on how far the robot is from the plane, but at closer distances $\pm 1/8$ in is possible depending on lighting and calibration.

4.1.2 Setup

Required Items

- Robot with Camera
- USB Cable connected between robot and camera. Cable will be located inside the control box from the factory.
- Calibration spike (Provided by Standard Bots or custom)
- Vision calibration grid: <http://camera-calibration.standardbots.com/>
- Accuracy Calibration Grid (Optional): <http://accuracy-calibration.standardbots.com/>
- Flat plane to pick from
- Items to pick
- Tooling to pick items

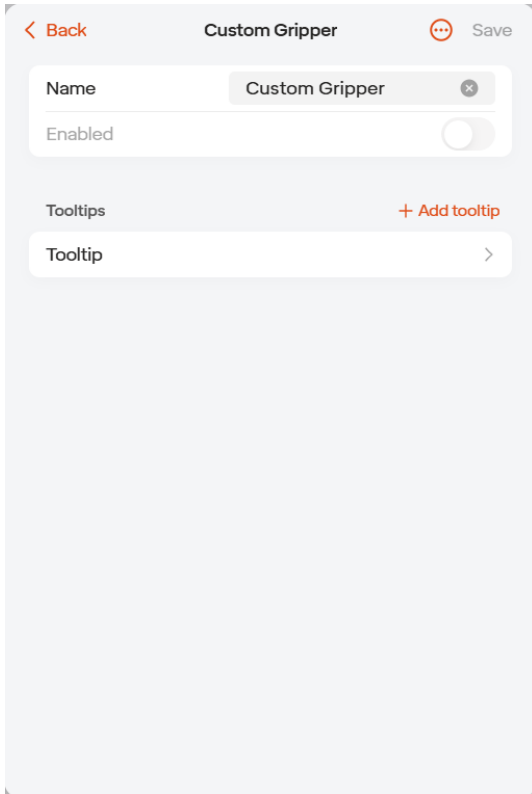
Camera Setup

1. Go to the Move Robot view, check to make sure that you are Live Robot mode not Simulation mode.
2. If needed, unbrake the robot.
3. Select the three dots icon in the upper right, then select the camera icon. If you do not have this icon, contact Standard Bots Support.
4. You should see the camera view. If you cannot. Ensure the provided USB cable is connected between the robot and control box. If it is, try rotating the orientation of the usb-c plug 180 degrees in the robot.
5. Go to “Equipment” in the robot menu (Menu has the robot name in it and is located in the lower left of the screen).
6. Click the “+” icon in the upper right of the window.
7. Add the Built-in Wrist Camera.
8. You do not need to perform any calibration here.
9. Click save in the upper right of the window.
10. Click done in the upper left of the window.

Calibration Spike Setup

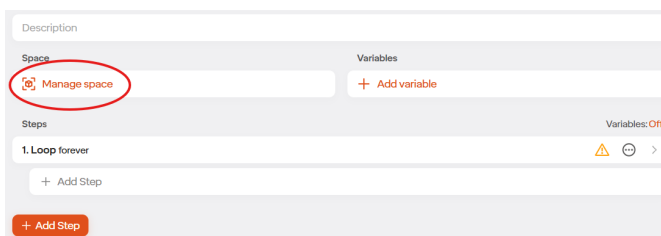
1. Go to the Move Robot view (located in the upper right of the screen).
2. Go to “Equipment” in the robot menu (Menu has the robot name in it).
3. Click the “+” icon in the upper right of the window.
4. Select “Custom Gripper”.
5. Select “Tooltip”.

6. If using the Standard Bots provided spike, set the z height to 50mm. If using another spike, set the offsets accordingly.
7. Click save in the upper right.
8. Click save again in the upper right.

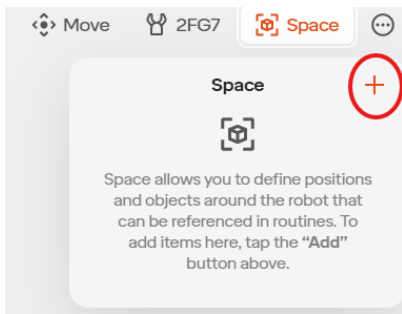


Space Setup

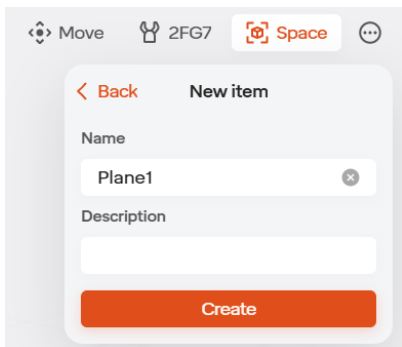
1. Create a new routine, or open the routine you want to run locate in.
2. Select “Manage Space”.



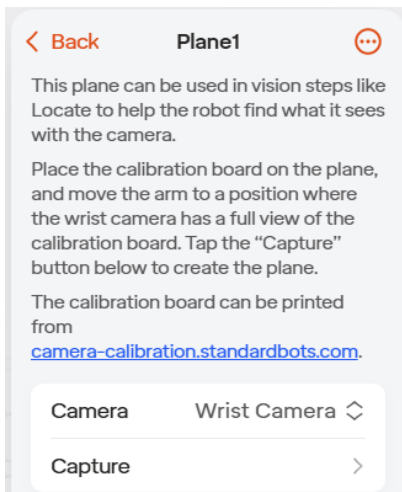
3. Go to “Space” in the Move Robot view (square icon located in the upper right of the screen).
4. Click the “+” icon in the upper right of the “Space” window in on the right side of the User Interface.



5. Select “Plane” in the list that appears. If you do not have this option contact Standard Bots Support.
6. Name the plane and select “Create”.



7. Select “Create with Camera” and then select “Capture”.



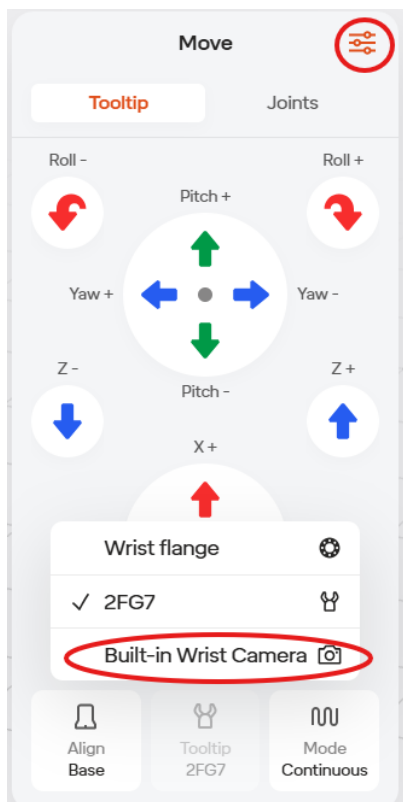
8. Place the larger, full 8.5 x 11 size calibration board on your plane.



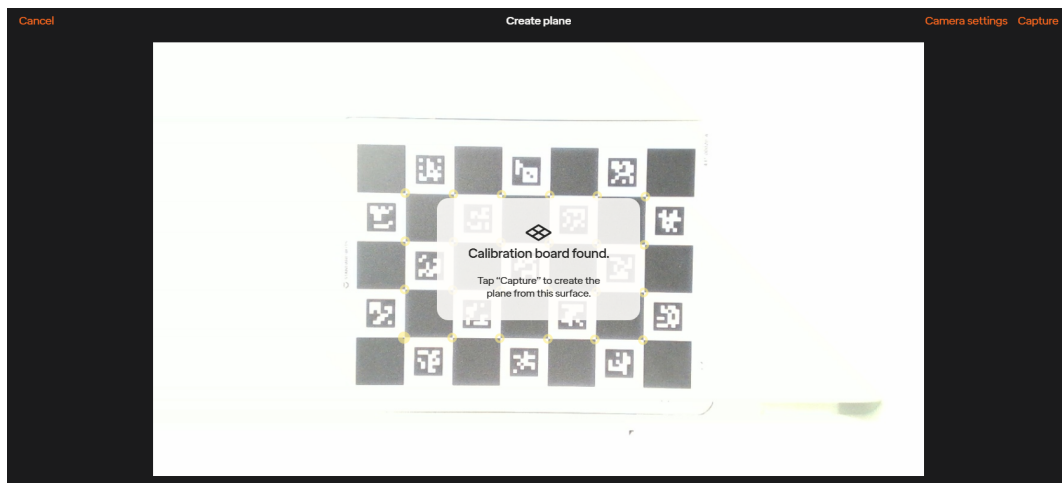
9. In the “Tooltip” jogging view, ensure the calibration spike custom tooltip is the current tooltip.



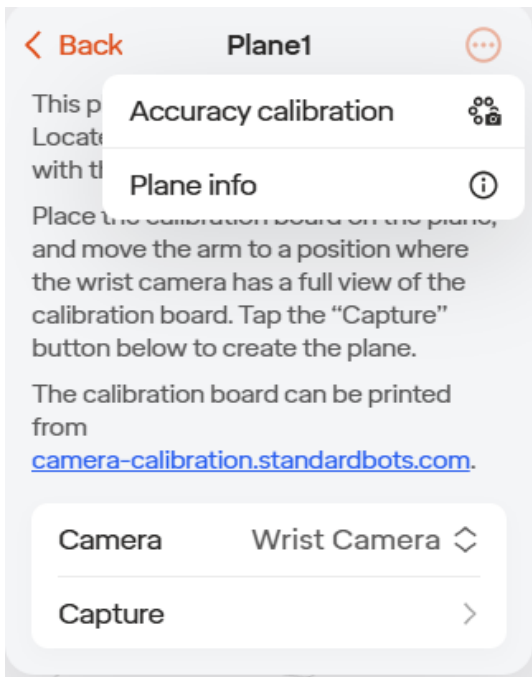
10. Drive the robot so you can see the calibration board with the camera. You can see the camera view while jogging by selecting the camera icon. Ensure the joint positions make sense and don't appear at risk of hitting your plane.
11. Under “Tooltip” jogging you can select the “Wrist Camera” frame. You can then select the slider icon in the top right of the tooltip jogging view. Using “Snap to Axis” you can align the wrist camera to the robot base.



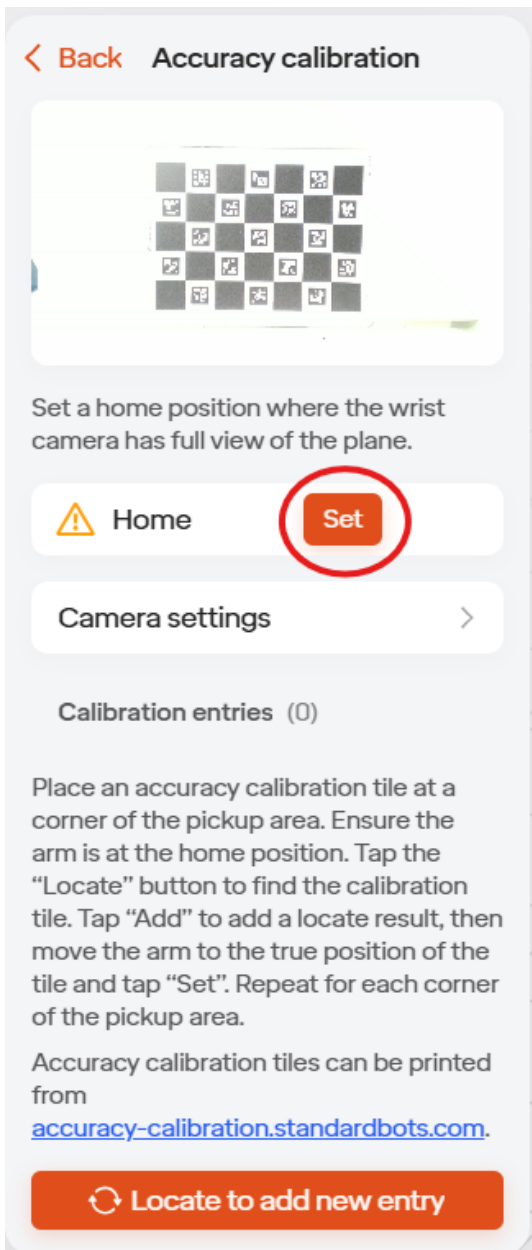
12. Once you have the calibration grid in view, get the robot as close to it as possible while still seeing the whole grid.
13. Go back to the “Space” in the bottom right.
14. Select “Create with camera”.
15. Select “Capture” in the upper right of the window. If capture is greyed out ensure you can see the whole grid, it is the correct grid and the lighting is good enough. “Calibration Board Found” will appear on the screen.



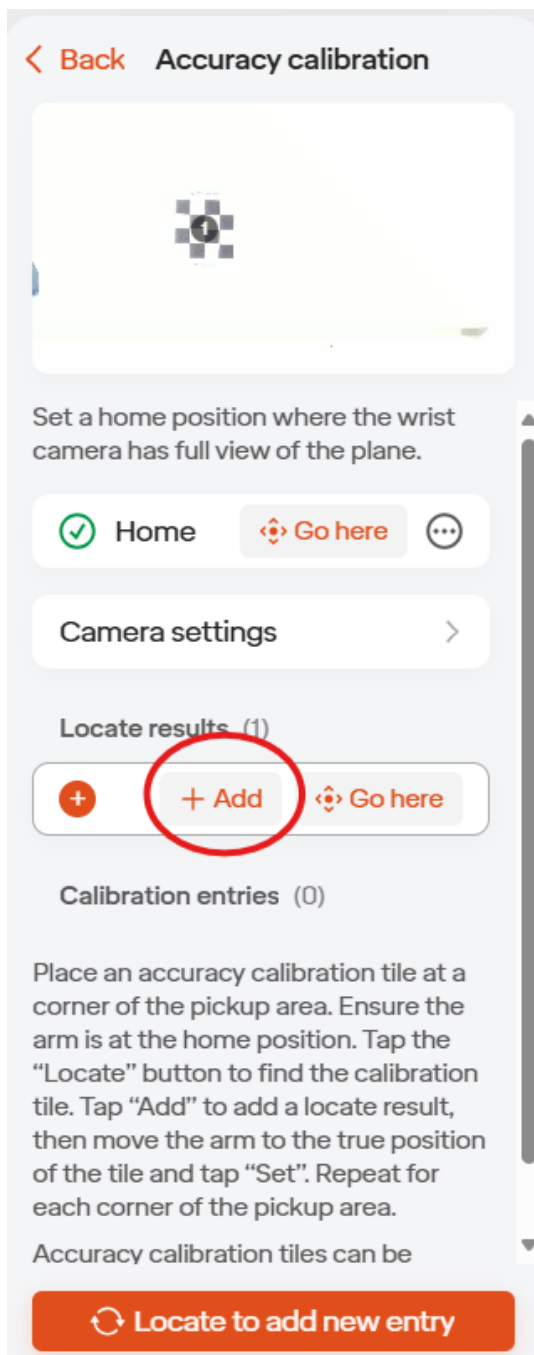
16. Click the 3 dot menu in the upper right of the plane calibration view.
17. Select Accuracy Calibration.



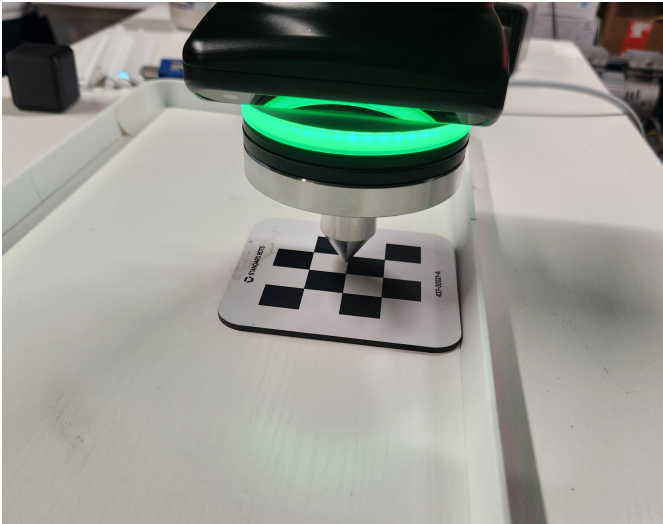
18. Jog the robot so you can see the whole work area. Ensure you have your custom tooltip selected in the bottom right of the tooltip jogging view.
19. Go back to the plane setup in the space view .
20. Click "Set" to create your home position in the plane view in space.



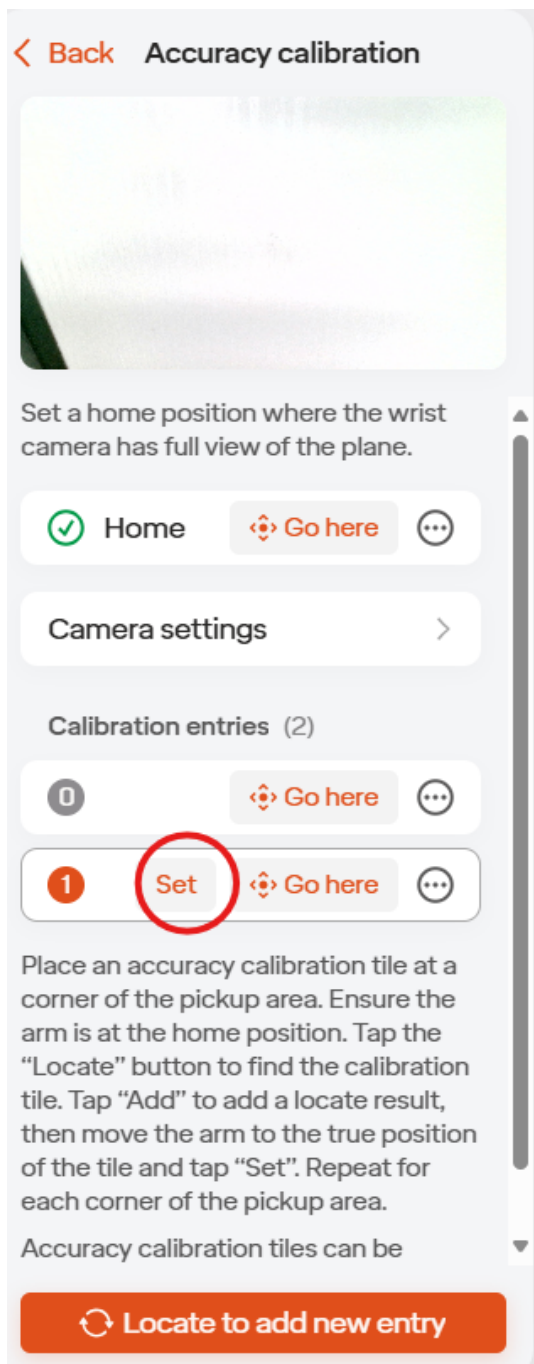
21. Place one of the smaller checkerboard calibration images in the robot view.
22. Select "locate to add new entry".
23. Add the entry. Ensure you do this before navigating away from the space window to jog.



24. Hold down "Go Here". Robot will drive above the grid.
25. Jog the robot to the center of the grid with the point of the spike touching the center. You can do this via the UI or with Anti-Gravity. Ensure the grid does not move. It is best to do this with the robot going at a slow speed.



26. Go to the Space window again, and click “Set” for this position.

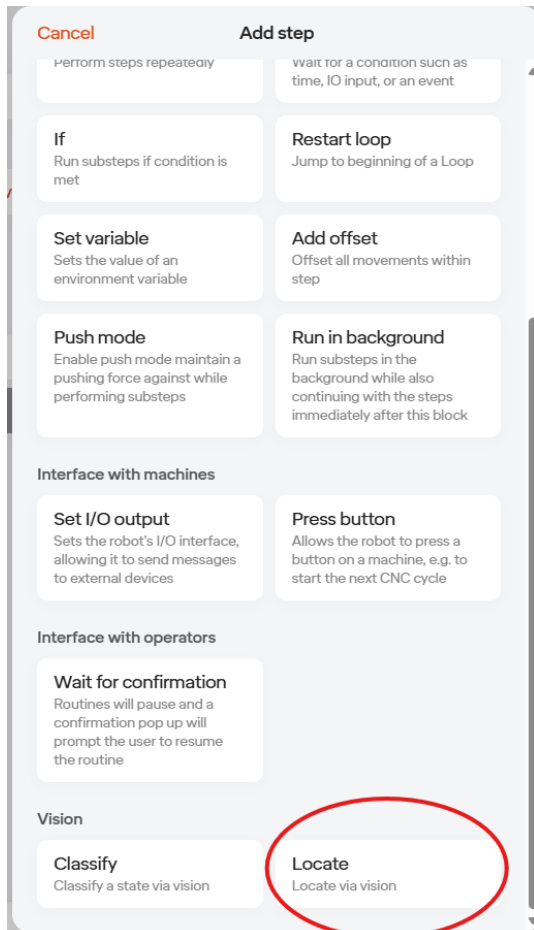


27. Bring the robot view back to the home position and repeat at least 4 times around your work area.
28. The plane is now ready to use in a Locate step.

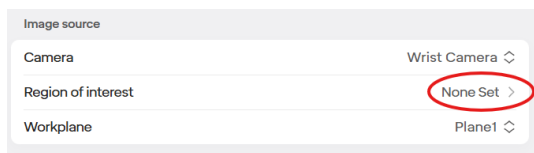
Routine Setup

1. Setup a tooltip or select a standard tool under equipment manager to use in your routine.
2. Go to the routine you created your plane in.
3. Leave the existing main loop and add all the below steps in this main loop.

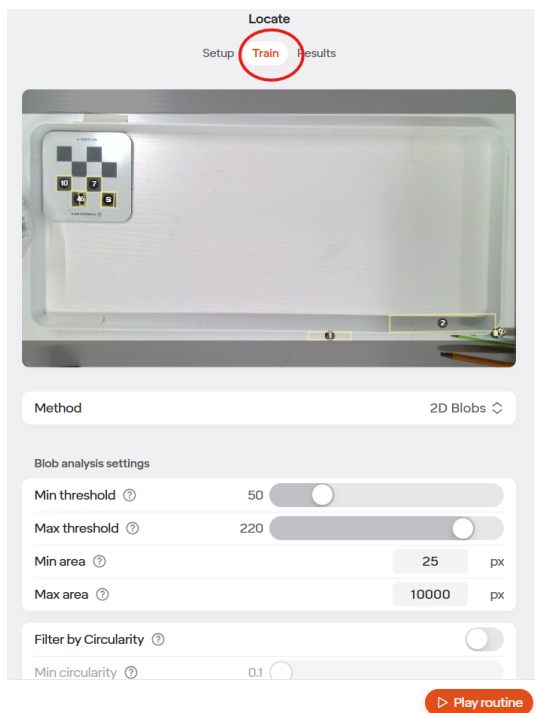
4. Setup a Move To Position positioned such that the camera can see the work area.
5. Add a Locate step (found at the bottom of the “Add Step” window).



6. Under the Setup tab of the locate step, set a region of interest. This is the area you want to pick from.



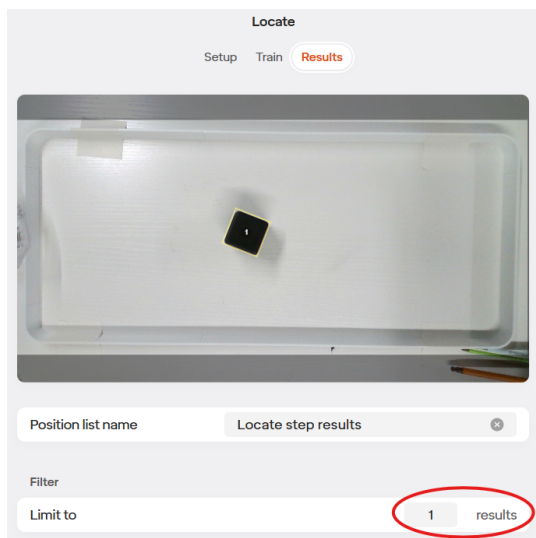
7. Under workplane, select the plane created in previous steps.
8. If needed adjust the camera settings so the items are visible. Generally the defaults are a good starting point.
9. Under the Train tab (located at the top of the Locate window), setup your blob or shape settings.



10. If using blobs, adjust the 0-255 Min and Max threshold greyscale value and Min and Max area values until just the items you wish to pick are indicated in the above image.

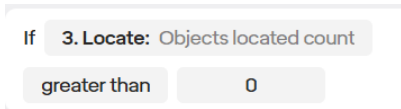
If using Shapes, Teach the image by taking an image and highlighting the items you wish to pick. Set the bounding box around the item. Adjust the % match required until all items you wish to pick are indicated in the above image.

11. Under results (located to the right of the Train tab), if needed limit the number of items locate should find.



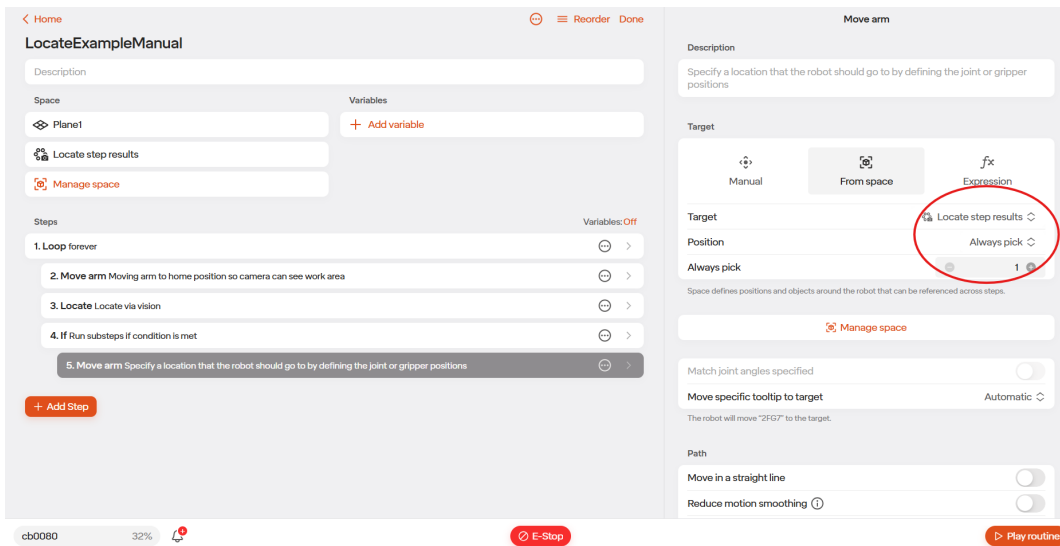
12. If needed, in results set a transform to apply to the located position when the robot goes to pick. Generally, it is good to start with a larger z offset of 40mm or so and tune from there.
13. Add an if statement below the locate step.

14. Set the if statement to say “IF Objects located count greater than 0”. Objects located count is automatically created for you when you add a locate step. This will ensure the robot only tried to pick if there are items to pick.



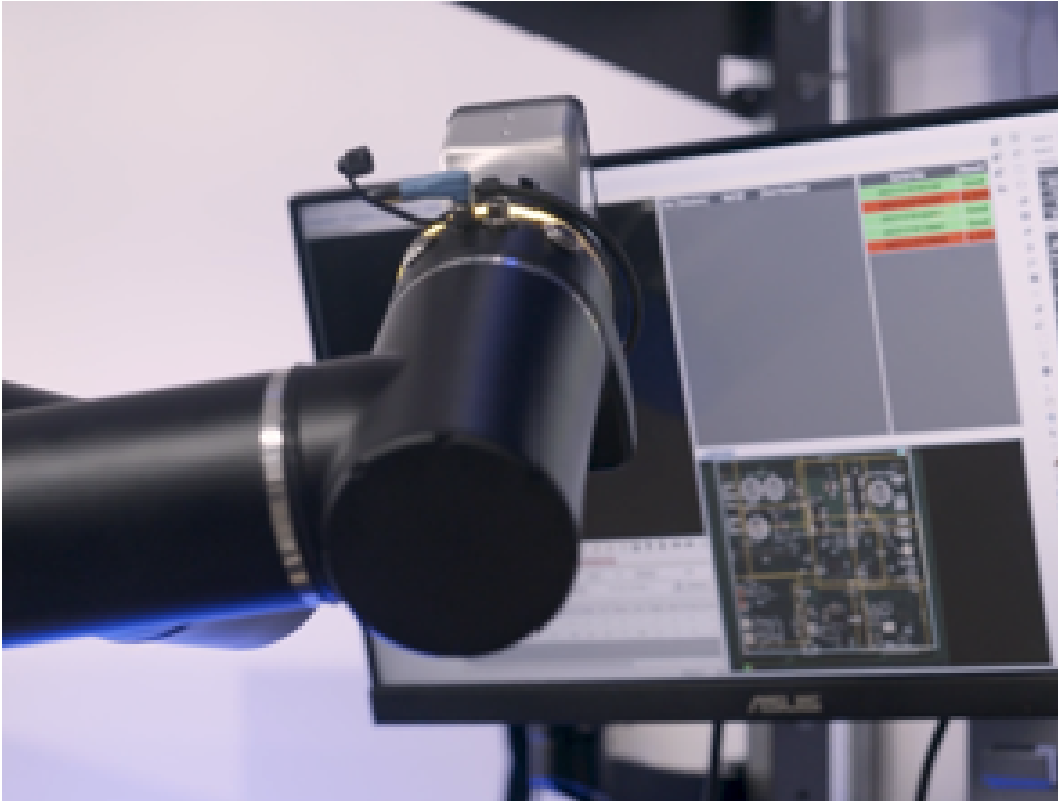
15. Add a Move Arm.

16. Select “From space” in the Move Arm command. Select “Locate Step Results”. Set it to always pick 1. This will pick the first item in the locate step results list.



17. Fill out the rest of the routine as desired. You likely will want to do an add offset to to the locate results item before driving to it, etc.

4.2 Classify



4.2.1 Use Cases

Standard Bots Classify feature is designed to allow the robot to look at an item and determine its state by comparing the current view to reference images. Classify can be used to look at a light stack to determine the machine state or to look at a HMI screen to know when a program is finished. Classify is not suited for detailed inspection / metrology applications.

4.2.2 Setup

Required Items

- Robot with Camera
- USB Cable connected between robot and camera. Cable will be located inside the control box from the factory.

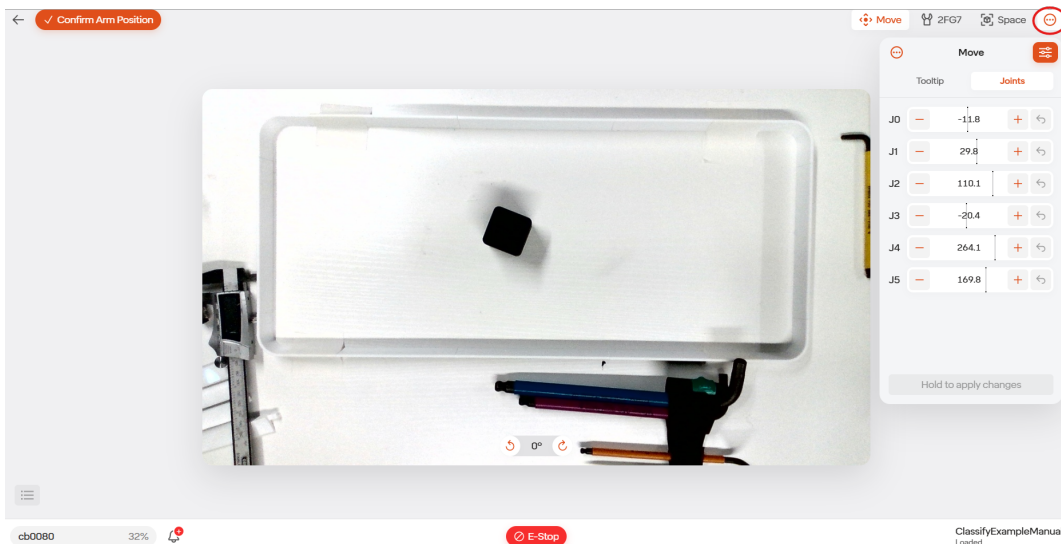
Camera Setup

1. Go to the Move Robot view, make sure the Robot view is in Live view and not Simulation.
2. If needed, unbrake the robot.
3. Select the three dots icon in the upper right, then select the camera icon. If you do not have this icon, contact Standard Bots Support.

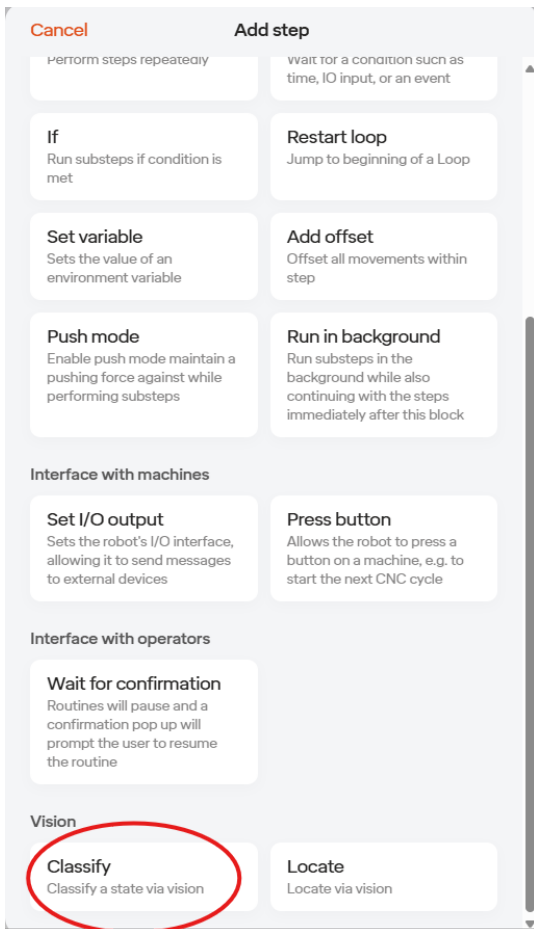
4. You should see the camera view. If you cannot. Ensure the provided USB cable is connected between the robot and control box. If it is, try rotating the orientation of the usb-c plug 180 degrees in the robot.
5. Go to “Equipment” in the robot menu (Menu has the robot name in it, located in the lower left of the screen).
6. Click the “+” icon in the upper right of the window.
7. Add the Built-in Wrist Camera.
8. You do not need to perform any calibration here.
9. Click save in the upper right of the window.
10. Click done in the upper left of the window.

Routine Setup

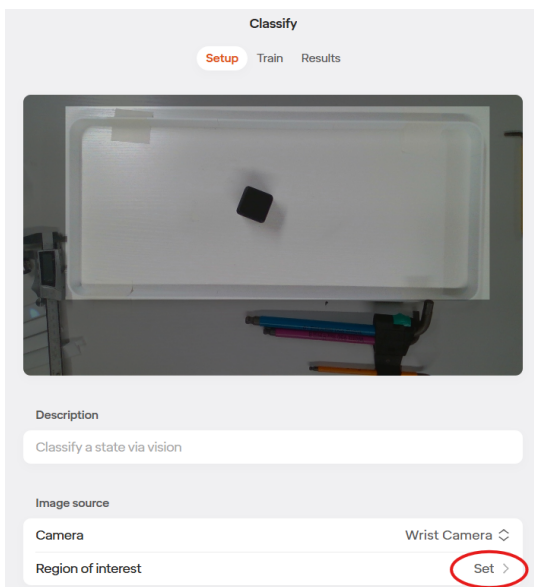
1. Create a new routine, or open the routine you want to run classify in.
2. Add a “Move Arm” step and set the position such that the camera can see the item you want to determine the state of. You can see the camera view while jogging by selecting three dots icon located in the upper right, then by selecting the camera icon.



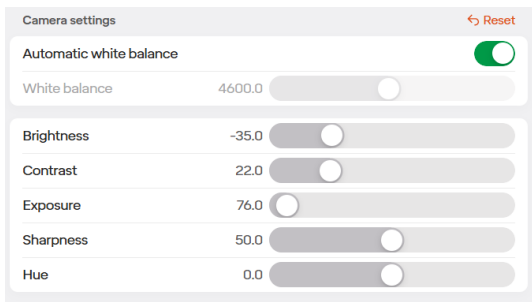
3. Add a “Classify” step (located at the bottom of the Add Step Window).



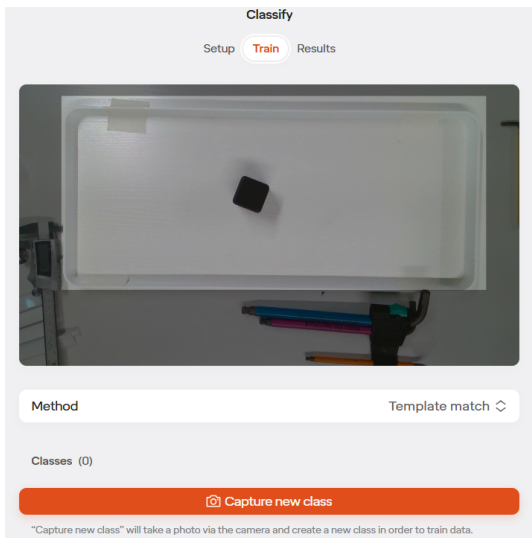
- Under “Setup” set the Region of interest to be only the part of the screen that is going to change state. Move the box outline and when done select “Crop”.



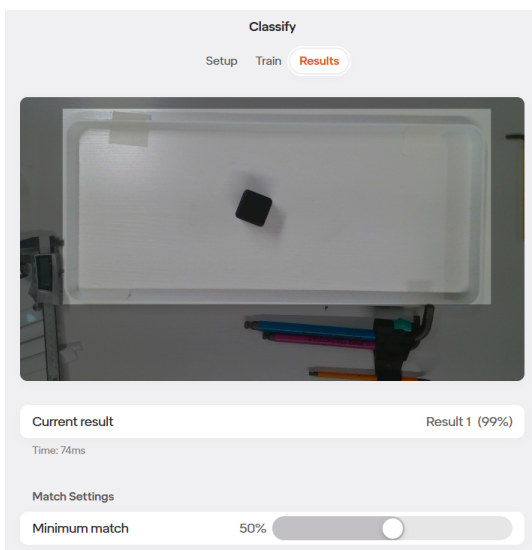
- You can adjust camera settings if needed, but the defaults work for most applications.



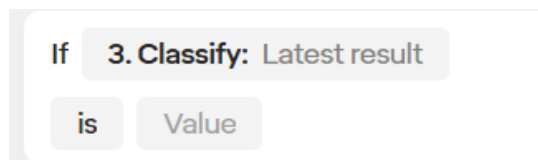
6. Under “Train” in the Classify step (located at the top of the Classify window), Capture a new class for each state and provide it a name.



7. If desired, add a fallback class to default to if no state is matched.
8. Under “Results” (located at the top of the Classify window), test the different states to see if the right result is reported. Adjust the minimum match threshold as needed.



9. Add an IF statement below the Classify step. You can now adjust the IF to say “IF Classify Results is (State programmed in Classify)”.



10. Create the rest of the program as desired.

Chapter 5

External Control Overview:

At Standard Bots, External control refers to the ability to command and monitor our collaborative robots using external devices—most commonly a PLC or a PC based system. This setup allows our robots to function as part of a larger automated cell, communicating in real-time with other equipment such as HMIs, conveyors, presses, vision systems, and safety hardware. This setup allows for seamless communication between the robot and external systems, enabling precise coordination, automated workflows, and enhanced safety tailored to our specific applications.

Why use External Controls?

- Enables operators to start or stop robot cycles directly from HMIs via a PLC
- Coordinates robot actions with other equipment in the automated cell, such as presses, conveyors, and safety devices
- Enhances system flexibility by allowing centralized control and integration
- Supports advanced decision-making by interfacing with vision systems, AI, or higher-level control logic

The steps within our External Control and what they do:

Step
Initialization & Handshake
Robot Ready Signal
External Start Command
End-of-Cycle & Reset
Safety & Fault Handling

Common External Interfaces:

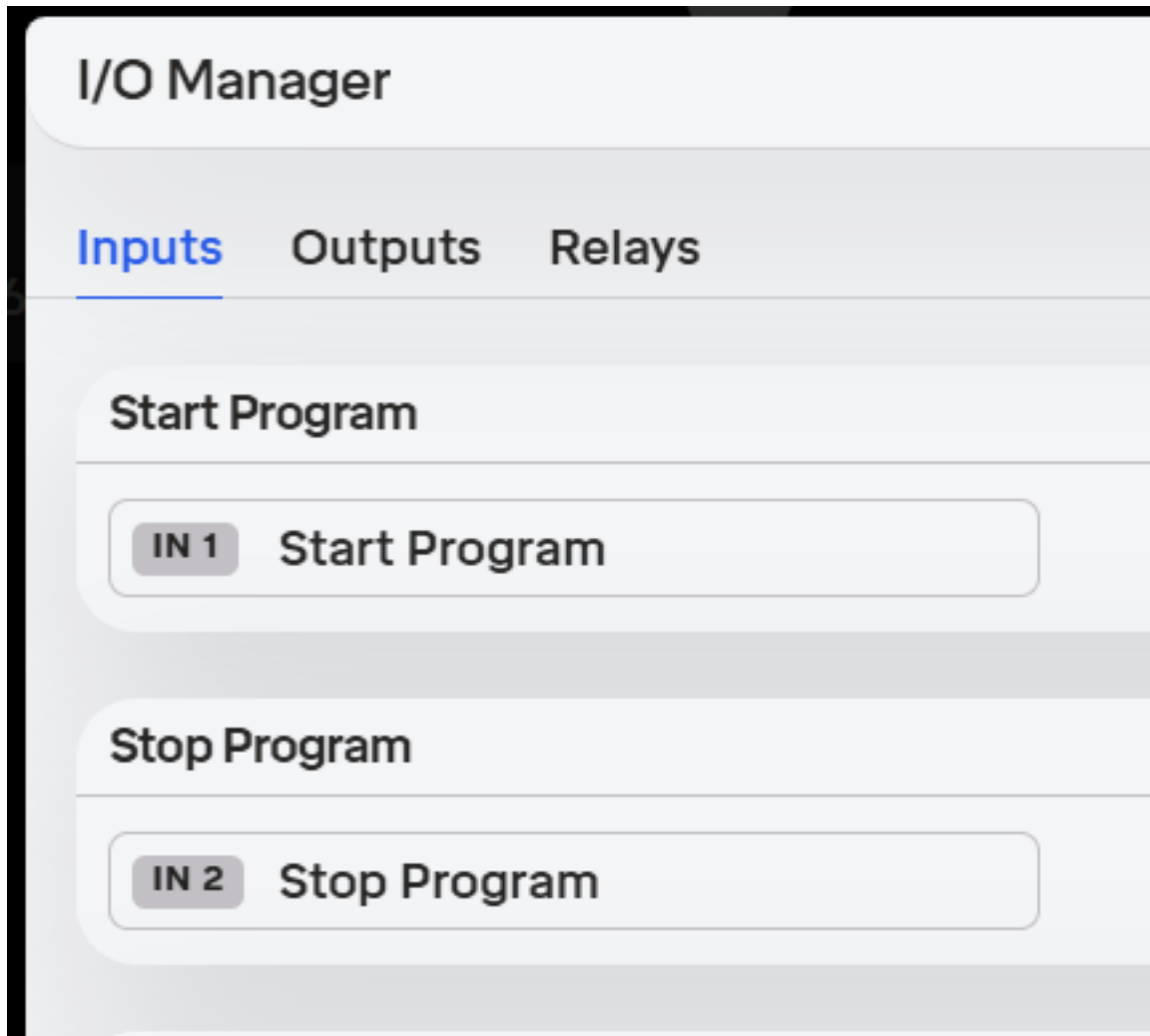
Type | Typical Use Cases | Notes
Digital I/O | Simple signals like Start, Stop, or Ready | Fast, reliable, low data volume
Modbus TCP | Connect to PLC or HMI using industrial protocol | Widely used in automation
Safety Inputs | E-stop and safety relay signals | Must be hardwired and safety rated

Example:

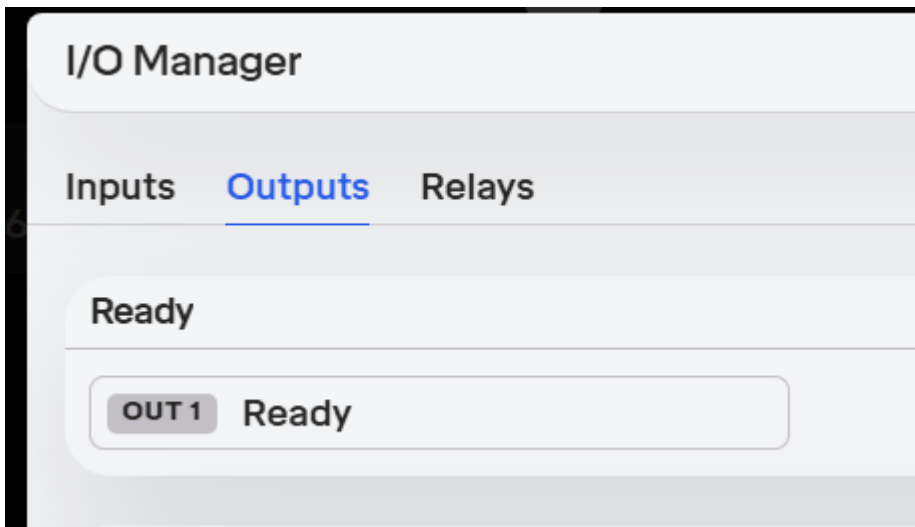
Scenario: A PLC or control box sends a signal to start or stop the RO1 when a part is present and safety conditions are met.

Hardware setup: DI1: Start signal DI2: Stop signal DO1: Robot ready (feedback to PLC)

Robot Configuration: Go to settings > I/O manager Assign DI1 > Start Program Assign DI2 > Stop Program



Assign DO1 > Ready



Set robot to "External Control Mode"

5.1 Configuring EtherNet/IP Communication

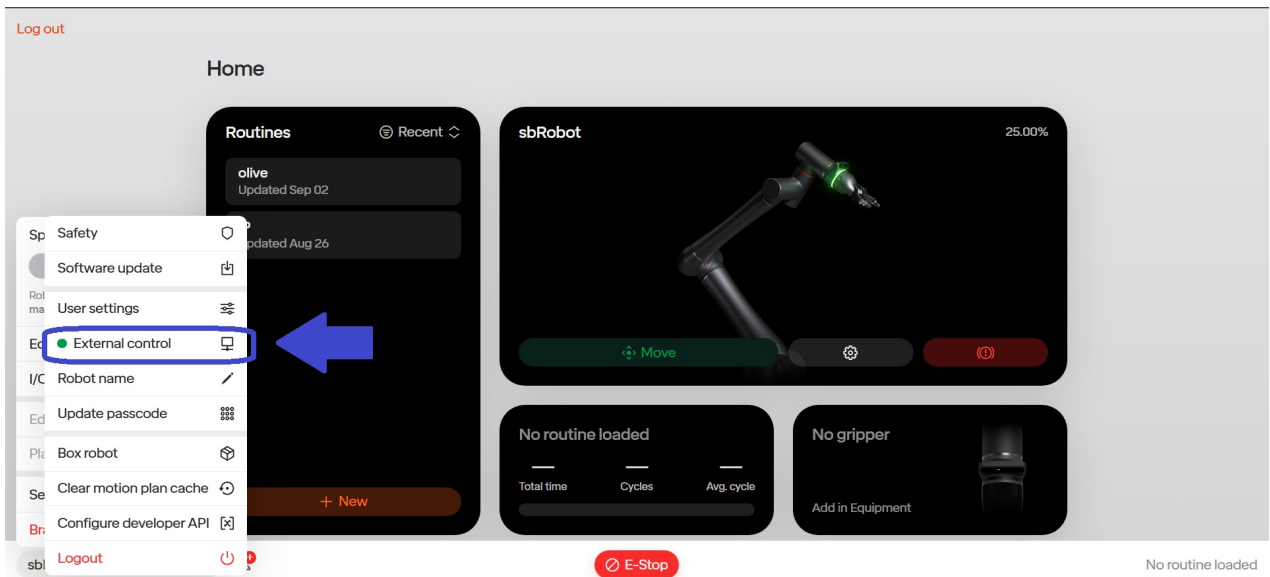
5.1.1 Introduction

This section explains how to configure the **RO1 robot** for external control via **EtherNet/IP** using **Studio 5000 Logix Designer**. The process includes enabling EtherNet/IP on the robot, installing the Electronic Data Sheet (EDS) file, adding the robot as an Ethernet device, and integrating the provided Add-On Instruction (AOI).

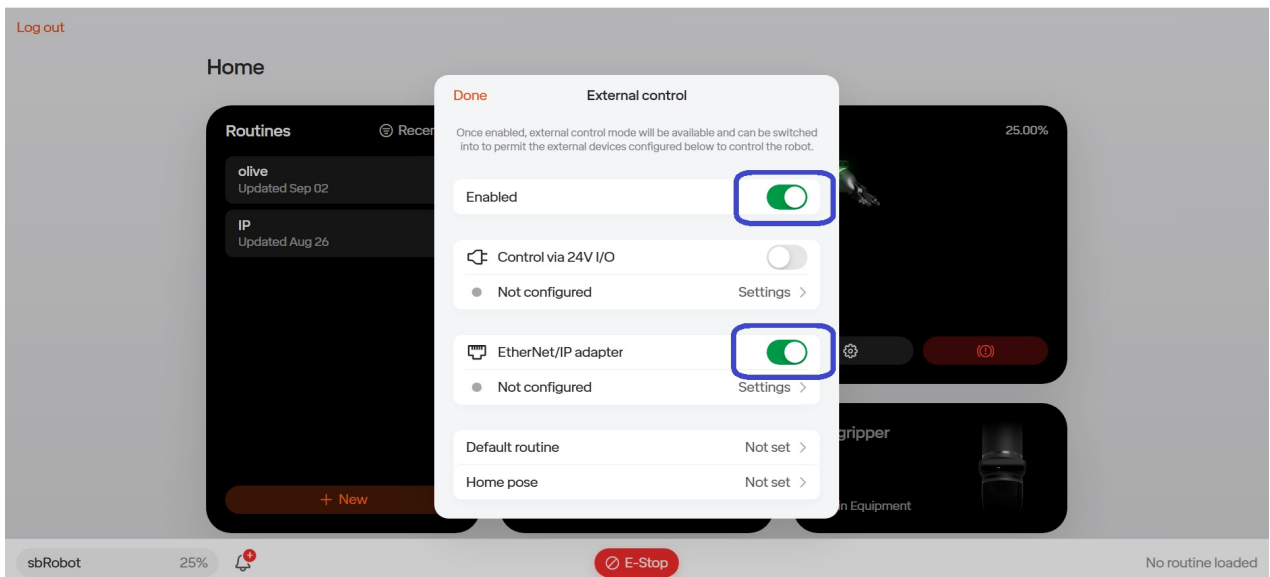
5.1.2 Configuration Steps

Step 1: Enable EtherNet/IP on the Robot

1. Connect to the robot using the **web interface**.
2. From the **Main Menu** (bottom-left corner), navigate to: **Settings > External Control**.



3. Enable **External Control** if it is not already active.
4. Toggle the **EtherNet/IP Adapter** switch to **ON**.



5. Select **Settings** to open the network configuration.

6. In the **Network** section:

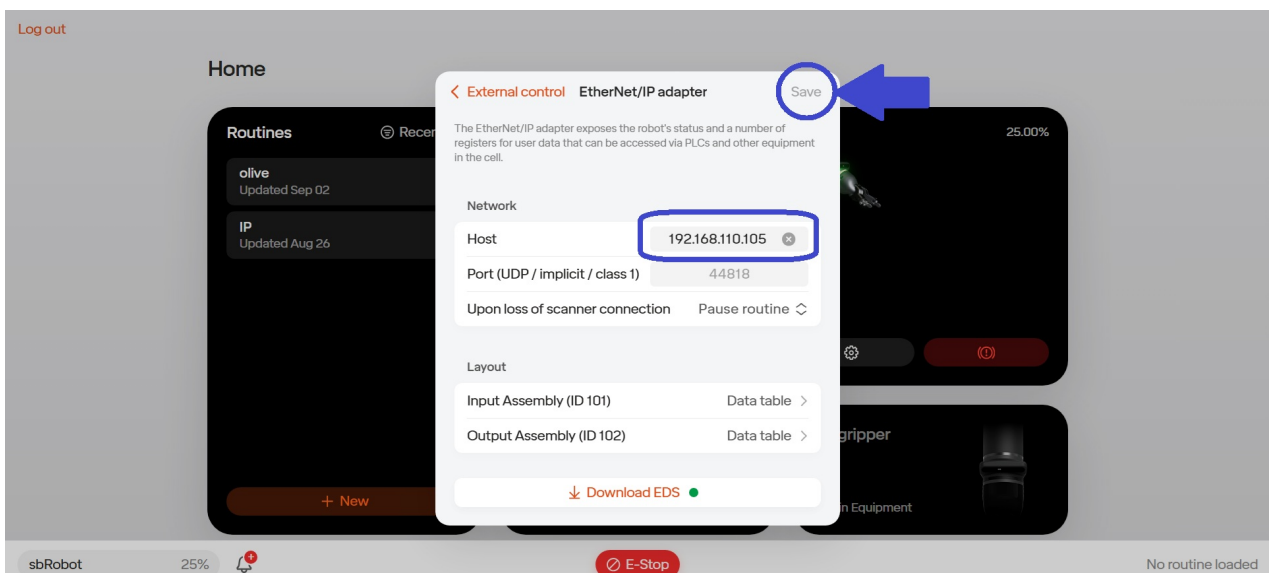
- Enter the **Host IP Address** → 192.168.110.105 (example).

Note: This needs to be the IP address of the network adapter on the robot.

Typically you should be using the Ethernet connection, not Wi-Fi or Direct. Find the IP address from the Network Setup and enter it here. This is not the IP address of the PLC.

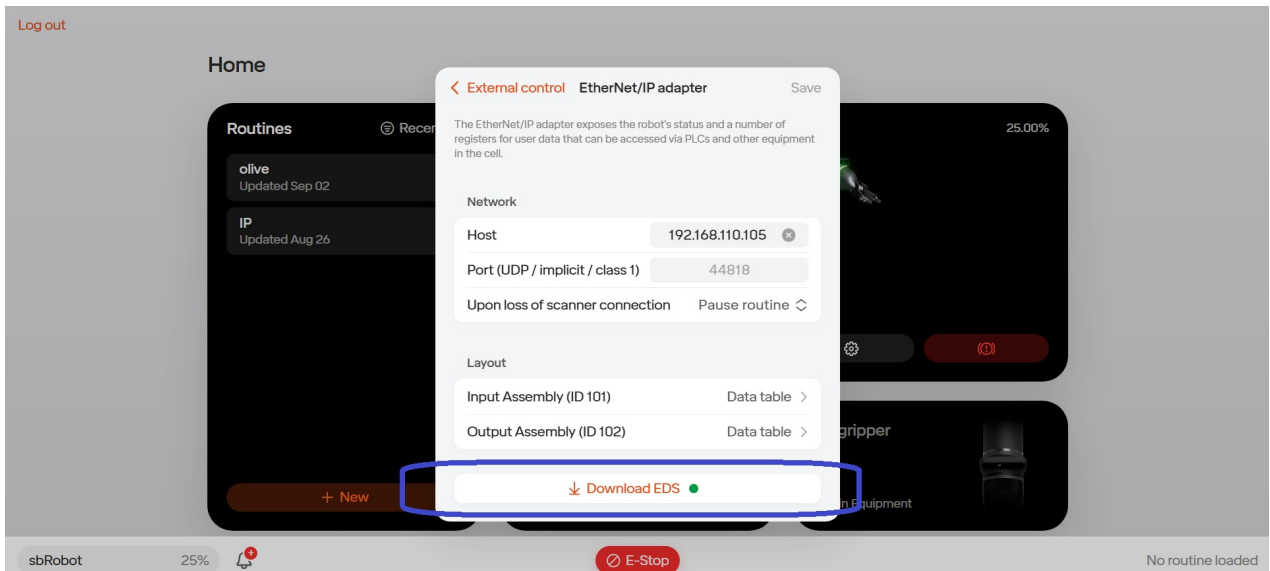
- Confirm the **Port** is set to 44818 (default for EtherNet/IP communication).

7. Click **Save**.



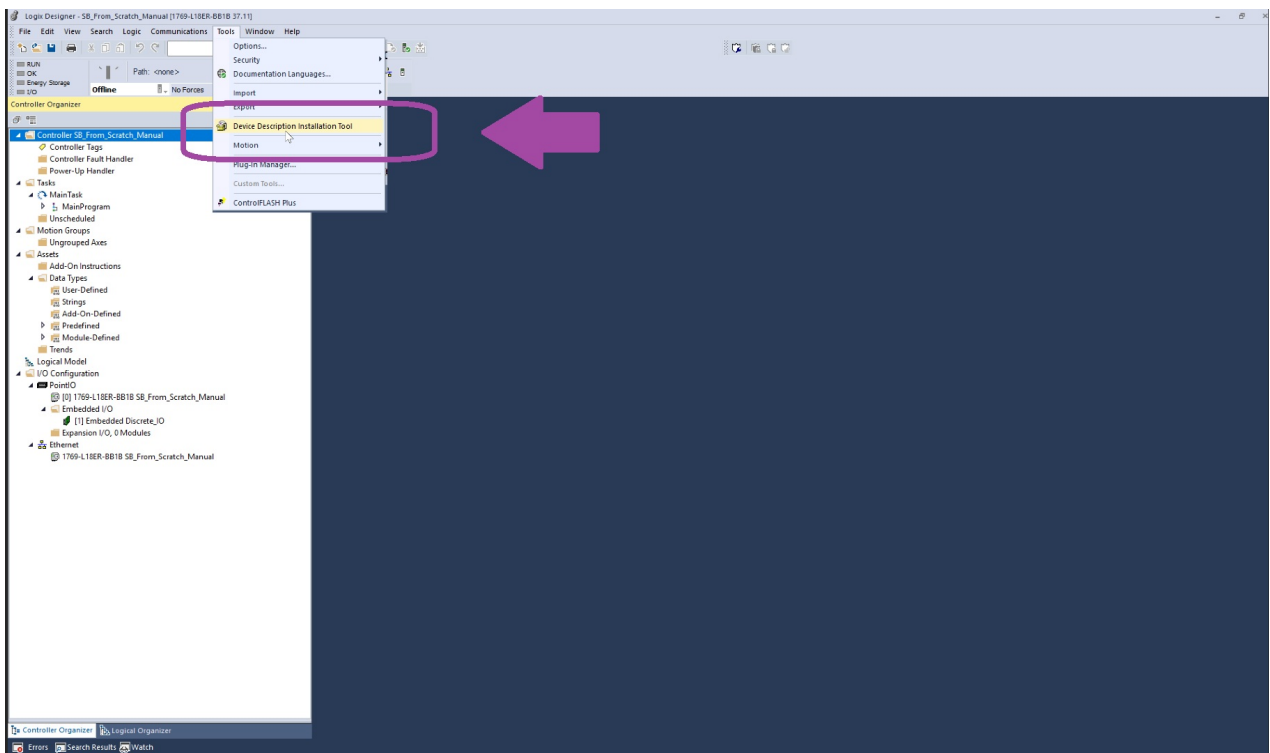
Note: At this step, you may also define variables within the input and output structure. This can be done in the **Layout** section located directly under the **Network** section.

8. Download the **EDS file**. The filename will appear similar to: robot_eds_2025-09-12T15-06-21.eds

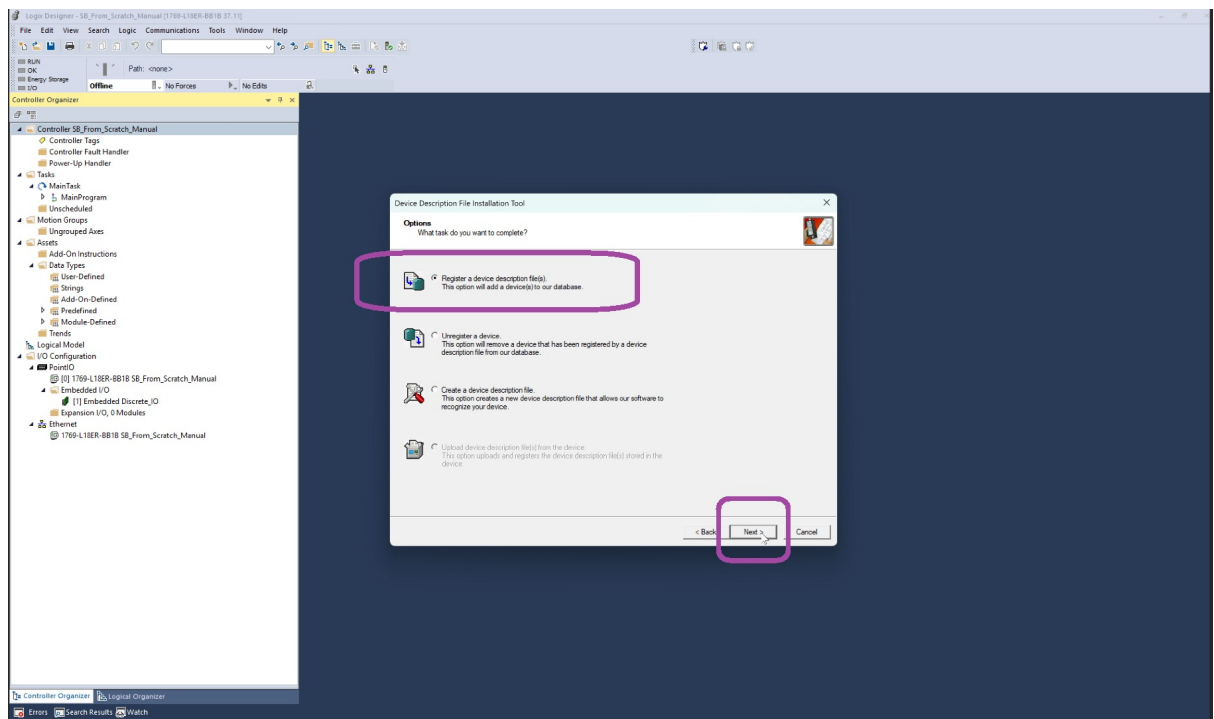


Step 2: Install the EDS File in Studio 5000

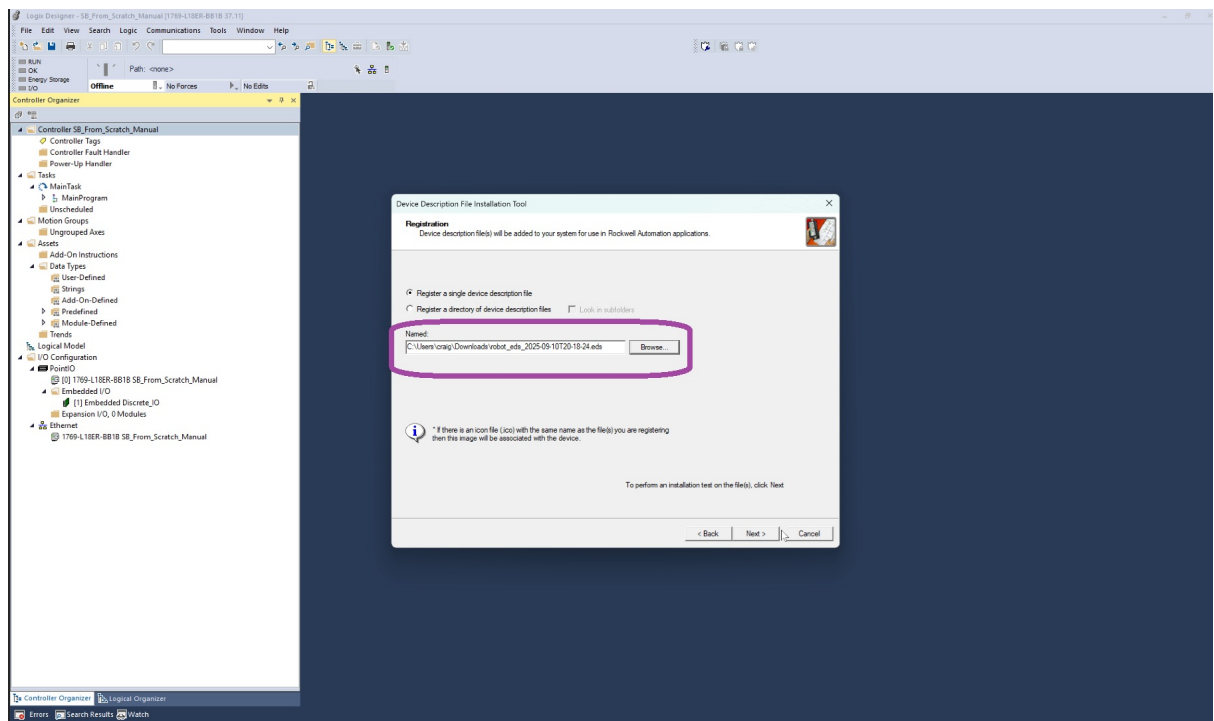
1. Launch **Studio 5000** and open an existing project or create a new one.
2. From the top menu, select: **Tools > Device Description Installation Tool**.



3. In the installation wizard:
 - Select **Register a device description file(s)** → **Next**.



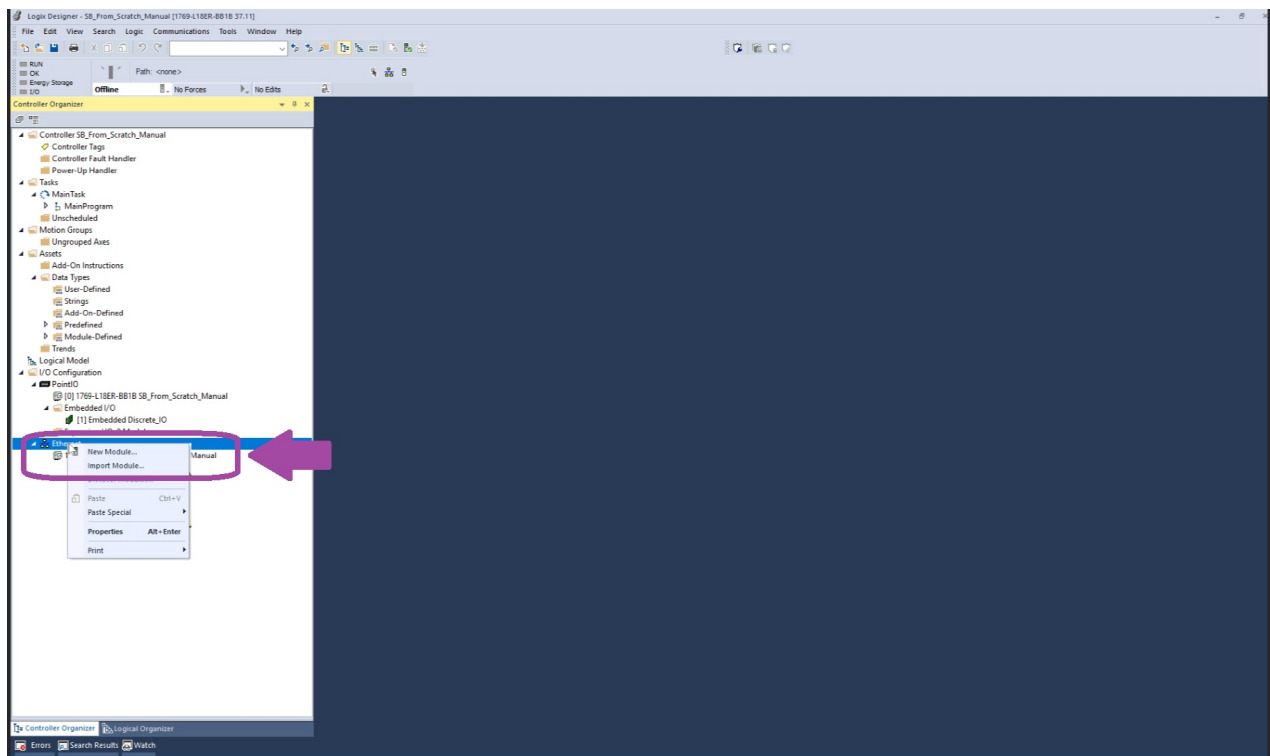
- Browse to and select the EDS file downloaded in Step 1.



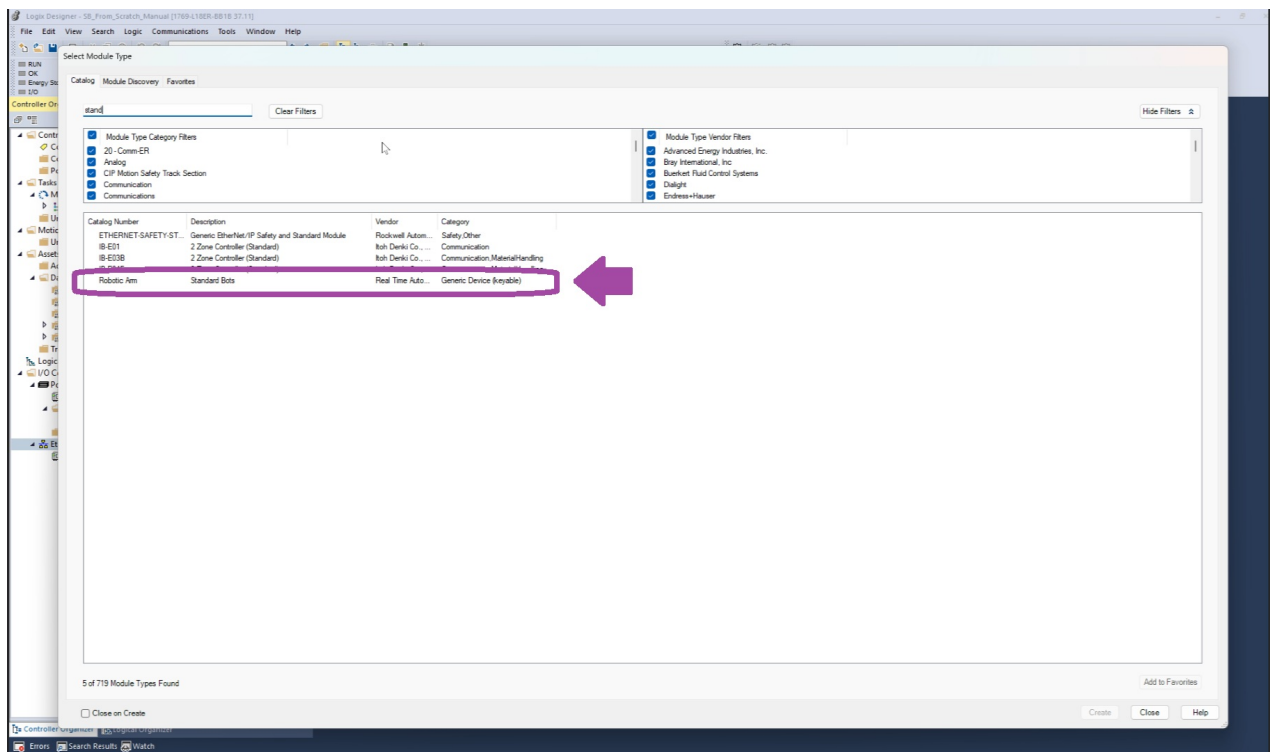
- Continue clicking **Next** until installation is complete.

Step 3: Add the Robot as an Ethernet Module

1. In the **Controller Organizer Tree**, locate the **Ethernet** branch.
2. Right-click and select **New Module**.

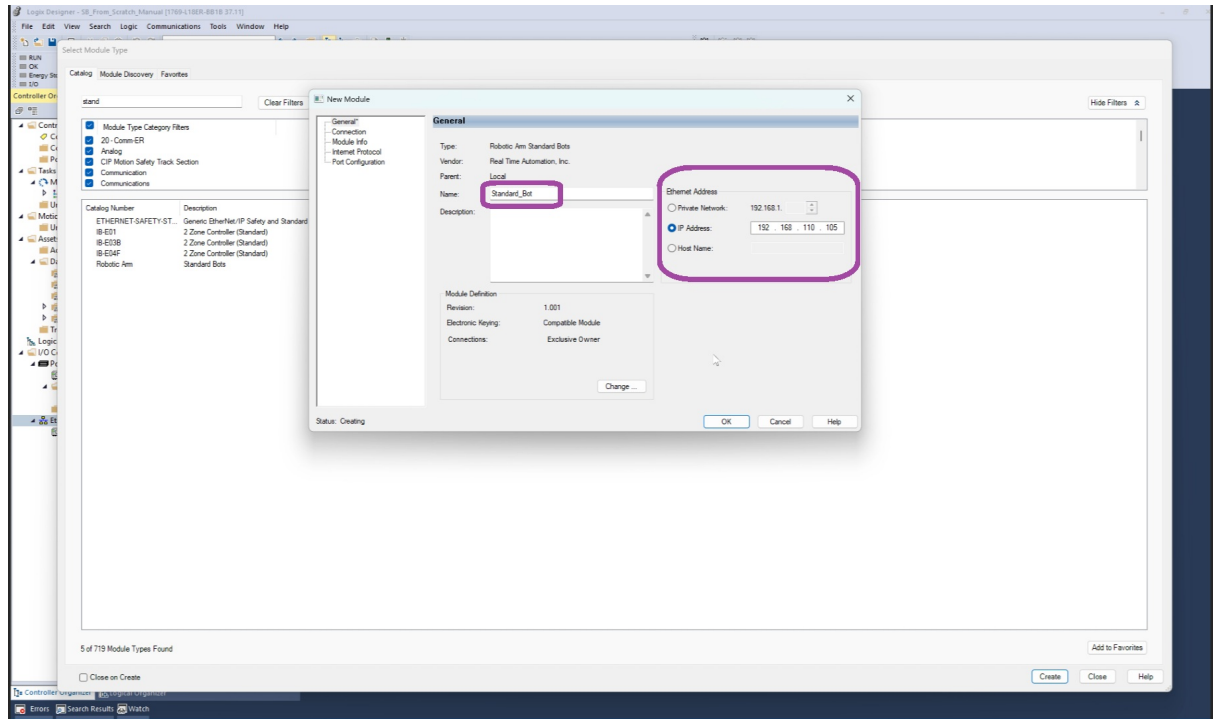


3. From the list, choose the **Standard Bots** module type.



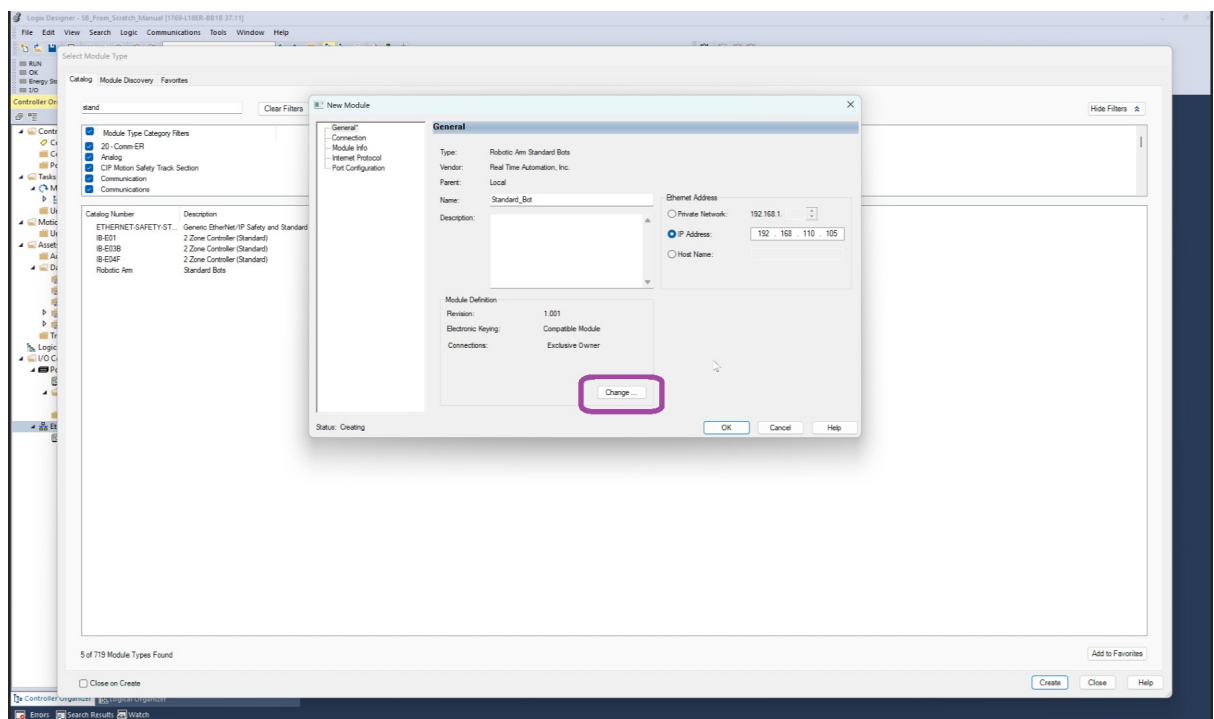
4. In the configuration window:

- Enter a **Module Name** (e.g., Standard_Bot).
- Set the **IP Address** → 192.168.110.105. (Note: IP address will be determined by user, 192.168.105 was used only for demo purposes.)

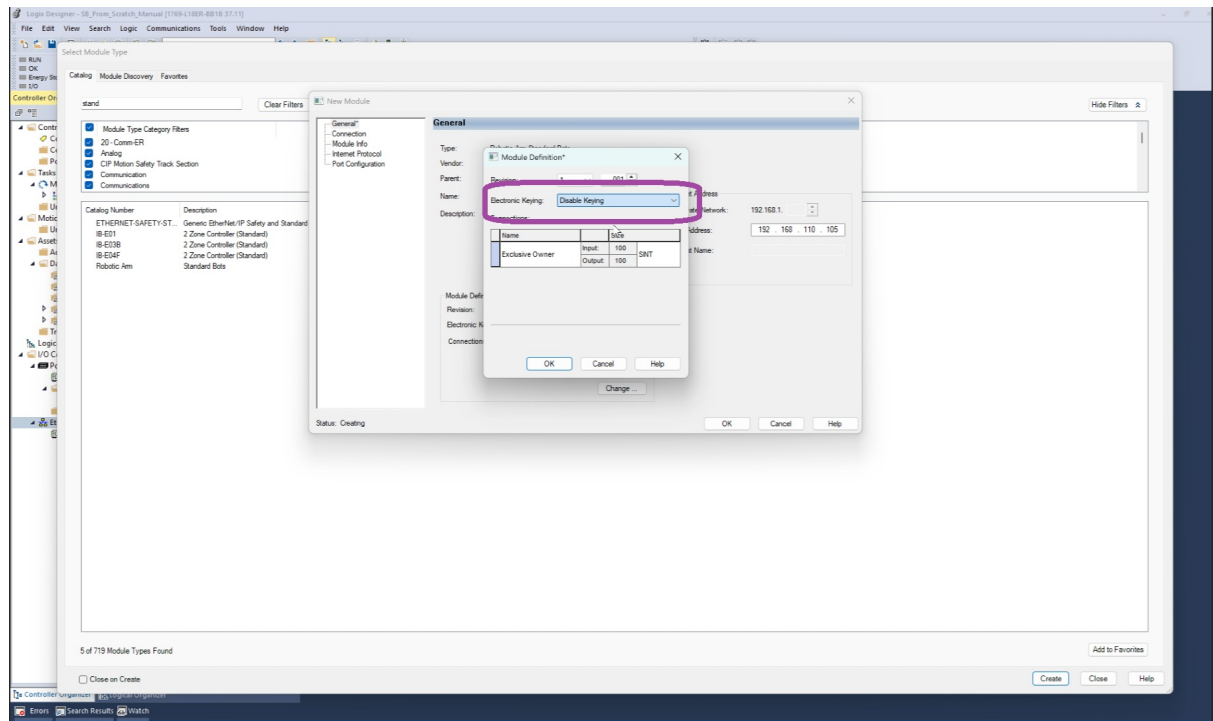


5. Modify the **Module Definition**:

- Click **Change**.



- Set **Electronic Keying** to **Disable Keying**.

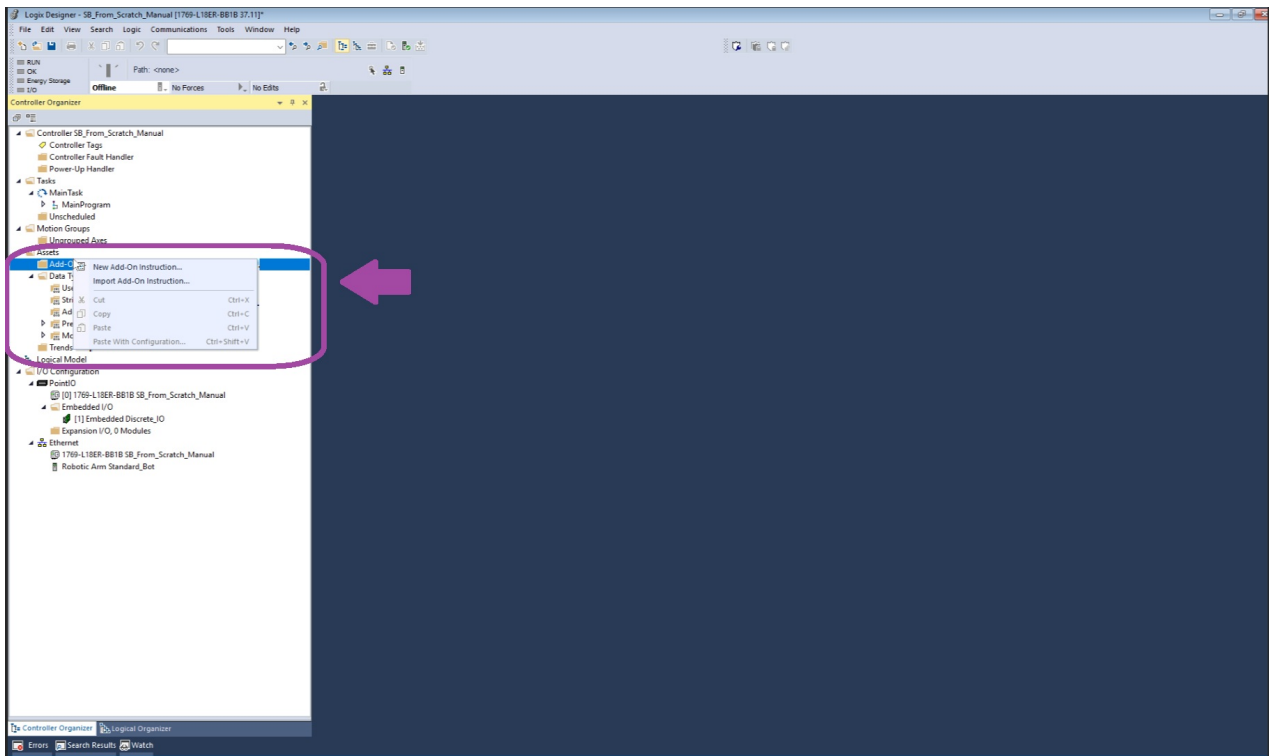


- Confirm with **OK**, then select **Close**.

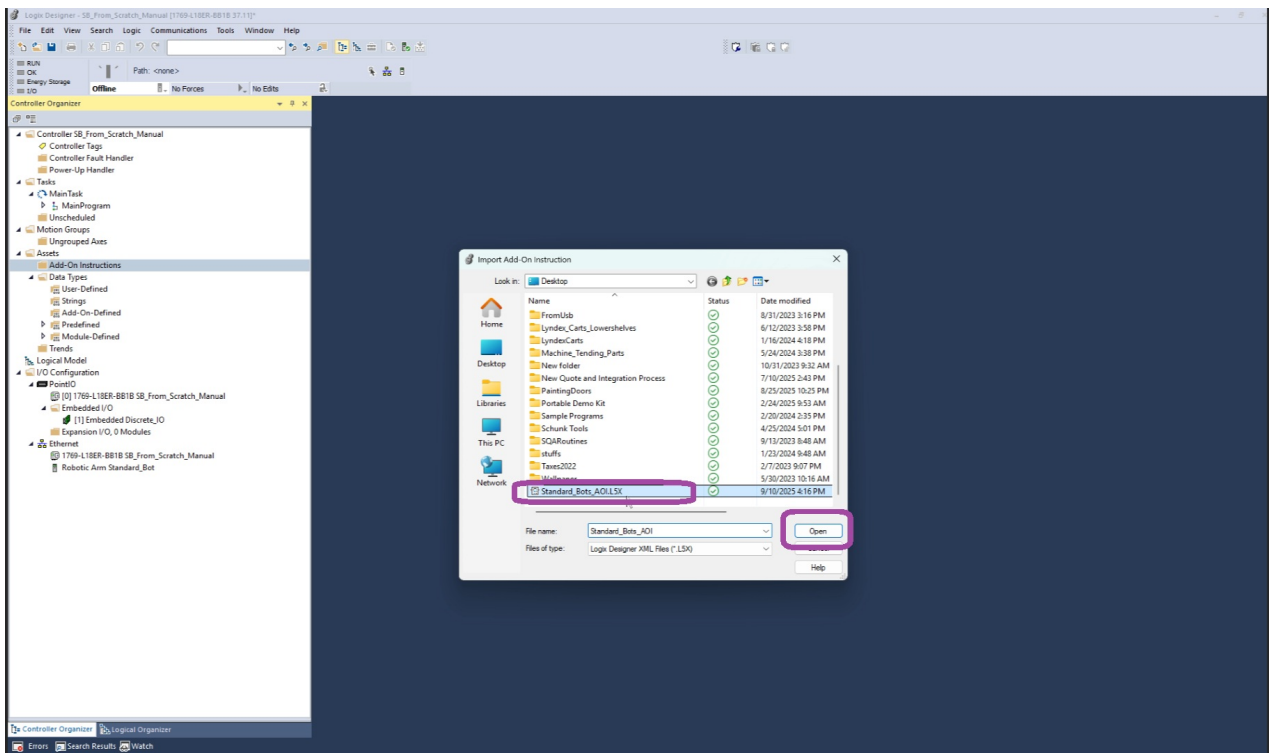
Step 4: Import the Add-On Instruction (AOI)

Note: The Add-On Instruction (AOI) is available for download at the following link: [Google Drive – AOI Files](#)

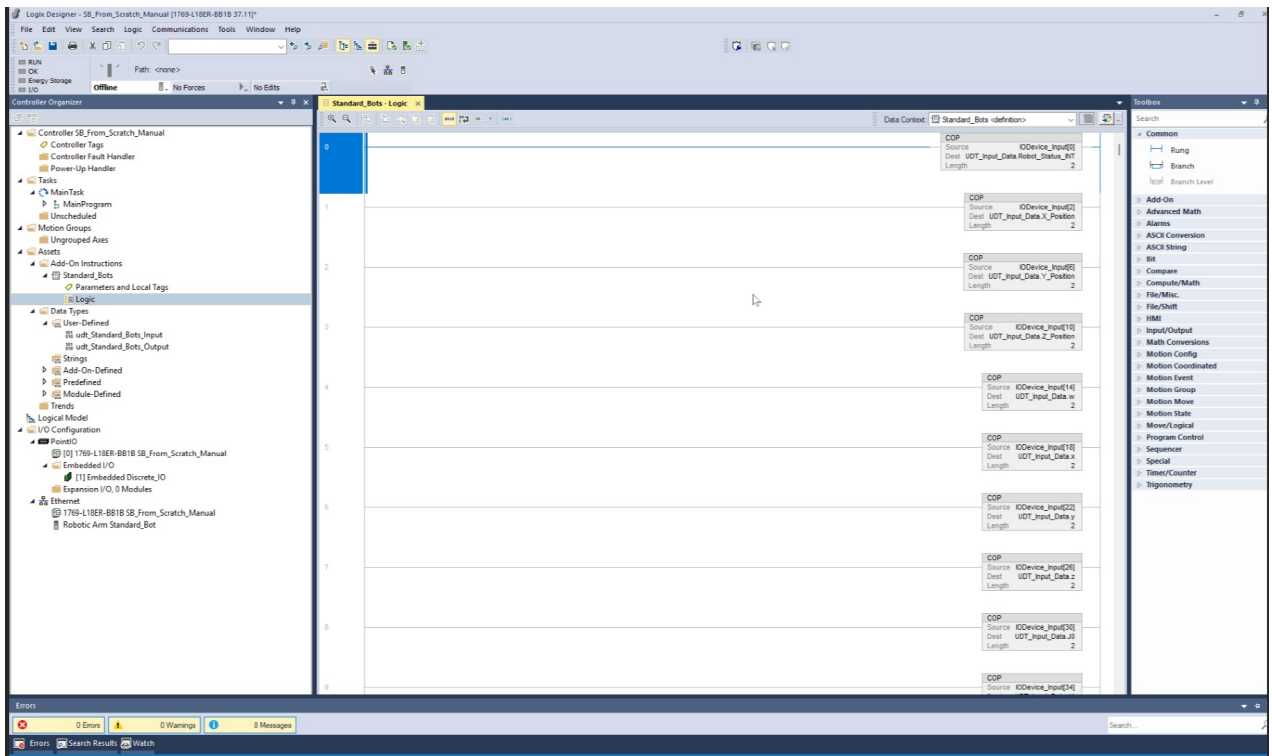
1. In the **Controller Organizer**, expand **Add-On Instructions**.
2. Right-click and select **Import Add-On Instruction**.



3. Locate and open the Standard_Bots_AOI file.
4. Verify the configuration details, then click **OPEN** > **OK**.



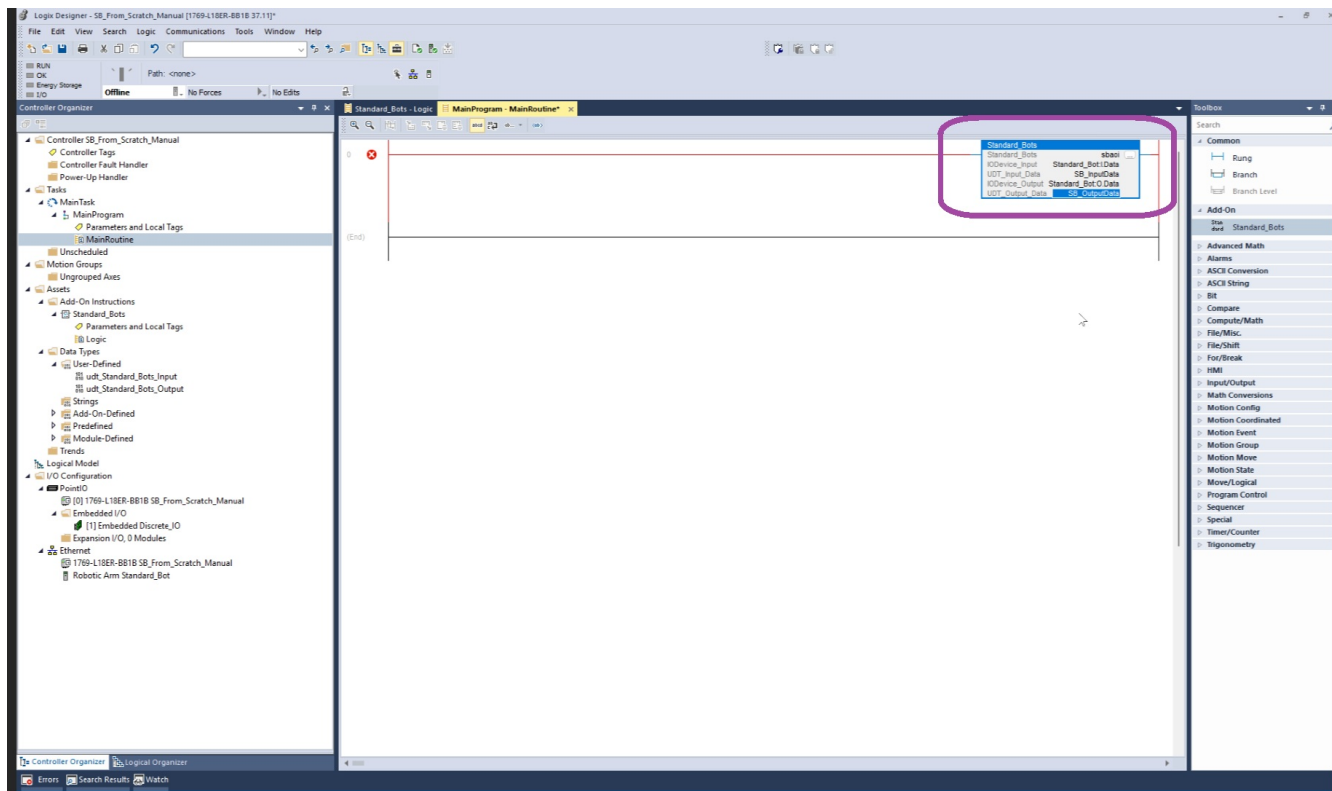
5. Confirm the AOI was imported successfully by reviewing its logic file in the Organizer.



Step 5: Configure the AOI in the Main Routine

1. Open the **Main Routine**.
2. From the **Toolbox > Add-Ons**, drag the **Standard Bot AOI** into a program rung.
3. Assign the following tags:

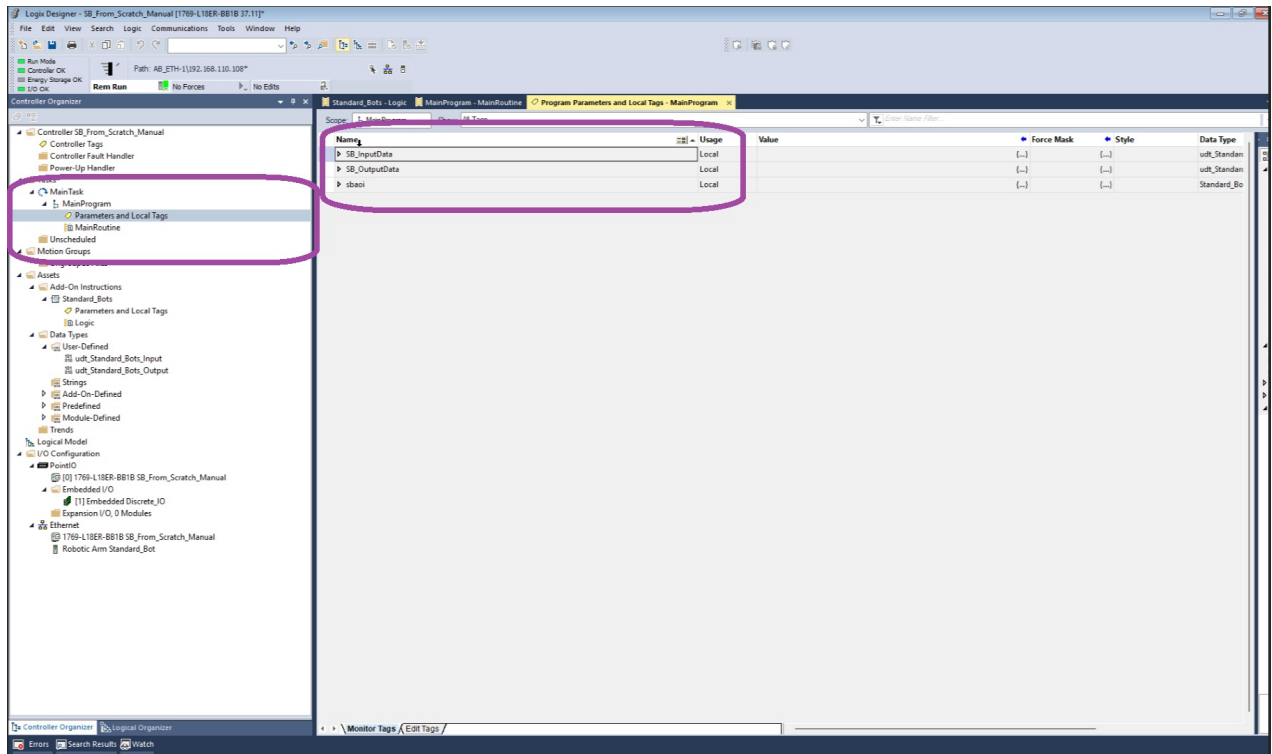
Reference	Tag Name	Example Value
Standard_Bots instance	sbaoui	Main AOI instance
IODevice_Input	Standard_Bot:I.Data	Input data mapping
UDT_Input_Data	SB_InputData	User-defined input struct
IODevice_Output	Standard_Bot:O.Data	Output data mapping
UDT_Output_Data	SB_OutputData	User-defined output struct



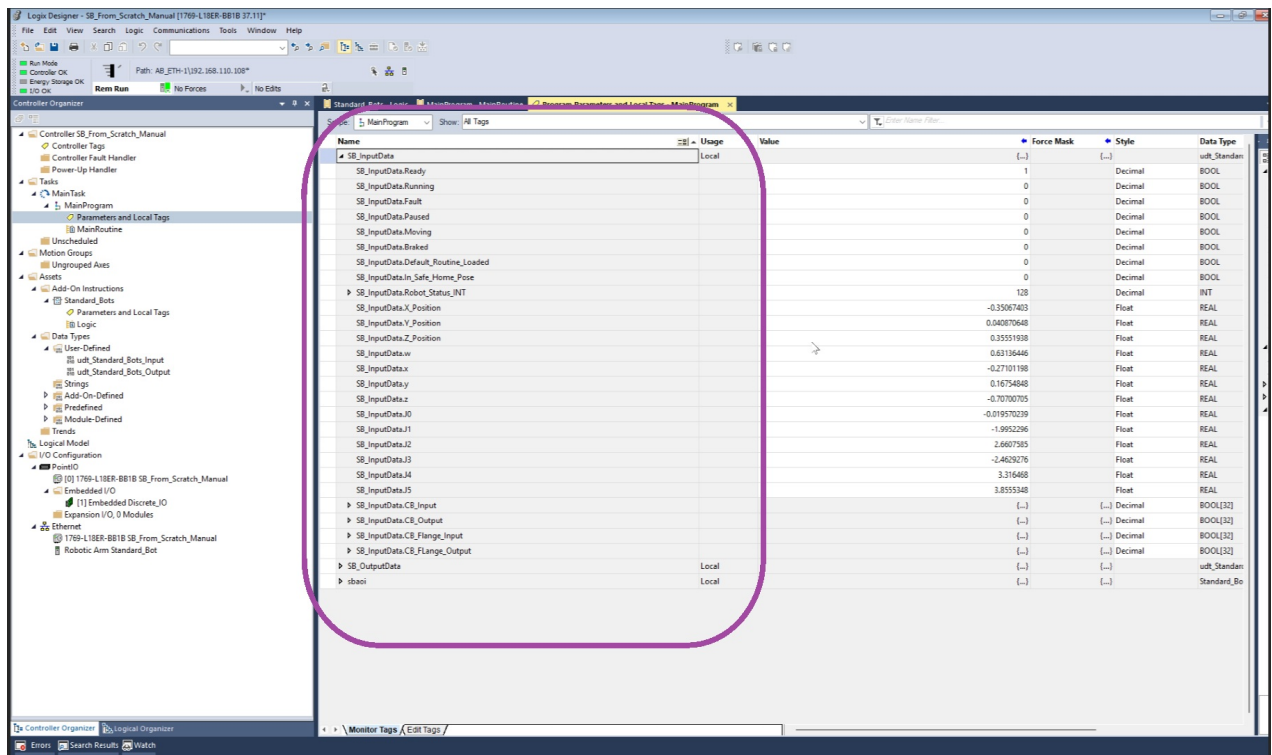
4. Download the project to the controller .

Step 6: Verify Integration

1. In the **MainProgram**, navigate to **Parameters and Local Tags**.



2. Expand the **SB_InputData** structure.



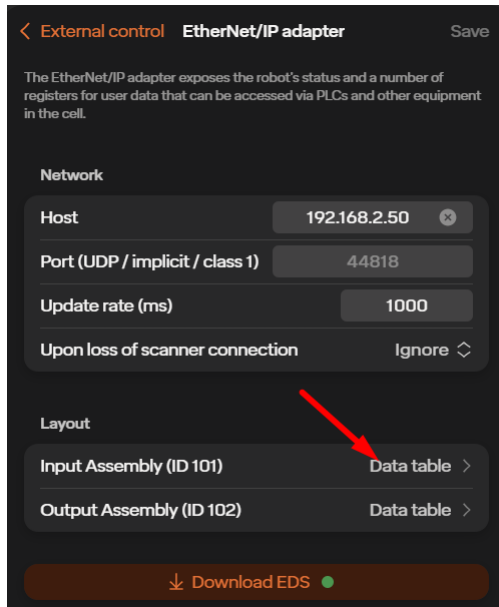
3. Confirm that all available robot control methods are accessible.

Note: Adding Custom Data

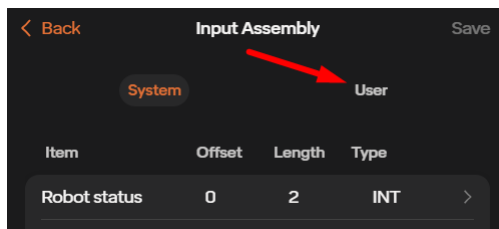
To add custom data to the data exchange, add Global Variables to your robot. Then add them to either the Input Assembly or Output Assembly through the User tab on those pages. Note that this will change the data size. After doing this, re-export the EDS file and reimport it into your PLC.

Open the Ethernet/IP settings from External Control

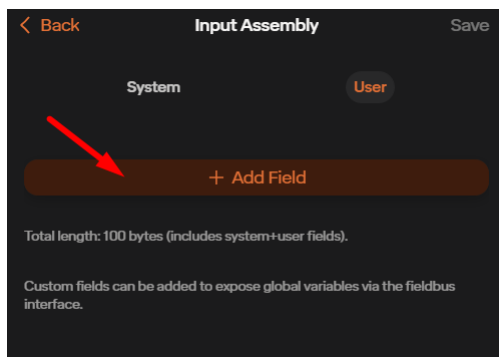
Click on either the Input Assembly or Output Assembly Data table link to add to or edit user data



Click on User at the top



Click Add Field



Select the Global Variable you want to link and select the data type you want to use.

The screenshot shows the 'New Field' configuration window. At the top, there are 'Back' and 'Add' buttons. The 'Variable' section contains a text field with 'Bob' and a dropdown menu also showing 'Bob'. Below this is a 'Description' text area. The 'Representation' section includes 'Assembly' (set to 'Input Assembly (101)'), 'Offset' (set to '100 bytes (after "Reserved space")'), 'Type' (set to 'INT'), 'Count' (set to '1'), and 'Length' (set to '2 bytes'). Red rectangular boxes are drawn around the 'Bob' variable name and the 'INT' type selection.

Click Save and make sure to export the new EDS and reimport it into your PLC as this changes the size and layout of the data structure. User variables get added to the end of the assemblies.

Conclusion

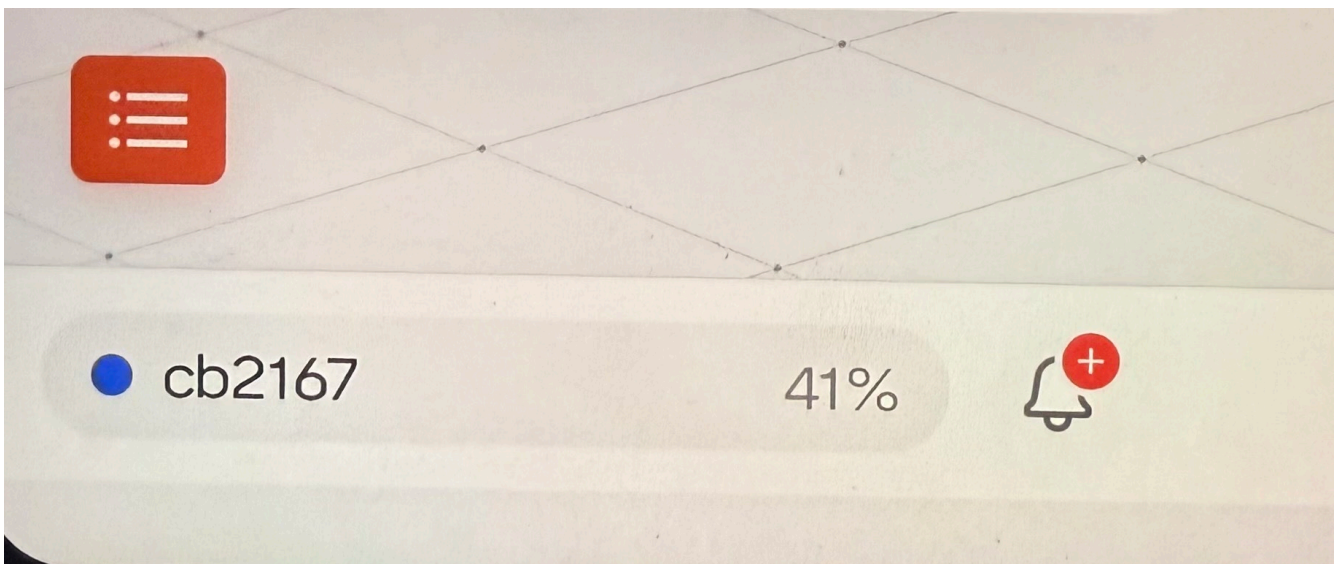
After completing these steps, the RO1 robot is fully configured for external control via **EtherNet/IP** in **Studio 5000 Logix Designer**. The AOI provides structured access to robot control functions, ensuring smooth integration with the PLC program.

Chapter 6

Offline Updates Overview:

6.0.1 Step by step

1. In the event of having to do an offline update we will send the file you will need upon request.
2. Extract the (.tar) file from the download; then copy the extracted files to the root folder of a USB drive.
 - This might take some time because it is quite a large file.
 - This is different from the Production version (PRD), but it is acceptable for now.
3. Insert the USB into the control box. (be sure that the control box is up to date with the main)
 - Wait until a blue dot appears.



4. Install the update
 - On the iPad access the UI, go to menu, click “settings” then “software update”.
 - Then on the software update tab, click on “Install”, where the USB drive has been detected.

- This will probably take around 25 min.

Speed

41%



Safety



Rot
ma



Software update



Eq

User settings



I/C

External control



Ed

Robot name



Pla

Update passcode



Box robot



Br

Clear motion plan cache



Logout



Cancel

Software update

Installed version: StandardOS 2025.08.01.4

ONLINE UPDATE

No newer version available via internet.

USB UPDATE

from "USB Flash Disk"

rox/no-op.1

build_0sapi_8EodyVeVORtEpxsk0v8w

Pending

Install

Last checked: Aug 11, 2025 at 5:24 PM

5. Once the installation is complete, confirm the new version appears as the “current version”.

- When the cb reconnects, it should automatically syn with the correct target build.

Chapter 7

Appendix A - Error List

Error	Reason	How to Fix Error
E-Stop was triggered	The E-Stop was pressed	The robot requires recovery. Navigate to the Routine Editor and press the Play button to access the Recovery Panel.
Internal communication failure		The robot requires recovery. Navigate to the Routine Editor and press the Play button to access the Recovery Panel.
Robot encountered a control system issue	Failed to receive encoder information for [Time in ms]	The robot requires recovery. Navigate to the Routine Editor and press the Play button to access the Recovery Panel.
Failed plan joint motion	Joint is already at limit of relative joint motion and can not travers any further in that direction.	The joint can not move any further in the direction of travel. The robot requires recovery. Navigate to the Routine Editor and press the Play button to access the Recovery Panel.

Error	Reason	How to Fix Error
Arm has encountered a collision	The arm collided with an object and stopped. The joint will be identified in the error, along with the detected Nm force and the threshold set force in Nm	The robot requires recovery. Navigate to the Routine Editor and press the Play button to access the Recovery Panel. Furthermore you can check the threshold settings and evaluate if the numbers are set correctly based on a safety study of the application
Unable to enter ANTIGRAVITY state		The robot requires recovery. Navigate to the Routine Editor and press the Play button to access the Recovery Panel.
CSl_collision detected	There was a collision detected. The threshold crossing limit will be displayed in m/s2	The robot requires recovery. Navigate to the Routine Editor and press the Play button to access the Recovery Panel. Furthermore you can check the threshold settings and evaluate if the numbers are set correctly based on a safety study of the application
Failed to run program: Already running program or robot-main is not running on Haas machine		Restart the robot or the CNC and verify the robot-main program is running on the CNC
Failed to receive encoder status	The E-Stop was pressed or the control box to robot cable is unplugged	Ensure Estop is not pressed and that cable between robot and control box is connected firmly
IO state timeout	The E-Stop was pressed or the control box to robot cable is unplugged	Ensure Estop is not pressed and that cable between robot and control box is connected firmly
Routine Invalid: Waypoint steps must have terminal Move Arm step	Waypoints can only be used when the next step is a Move Arm	Uncheck "Treat as waypoint" or add Move Arm step after Waypoint step.
Motion failed	Robot cannot plan a path between current position and target position, likely due to a singularity.	Adjust starting position or target position.

Error	Reason	How to Fix Error
Torque Limit Exceeded	Too much torque was recorded on the joint named in the error message	Reduce payload, speed or reach.
Joint Limit Exceeded	Joint is already at limit of relative joint motion and can not traverse any further in that direction.	The joint can not move any further in the direction of travel. The robot requires recovery. Navigate to the Routine Editor and press the Play button to access the Recovery Panel.
Error Loading Camera Feed	Vision camera does not display video feed	Check camera connection to robot. Rotate USB-C cable 180 degrees and try camera again
EOAT Not Detected	The robot is not detecting the EOAT	Check the EOAT connection to the robot. Verify that you added the EOAT into Equipment after it was connected to the robot.
Ipad will not connect to robot	There is a wireless issue between the Ipad and the robot either over WiFi or Bluetooth	Make sure the Ipad and the robot are on the same network. Verify that Bluetooth is enabled on the Ipad.
Failed to Generate Motion Plan	The robot is unable to move to a position.	Make sure the robot can reach a position. Ex: when using offsets make sure it does not go beyond the robot's reach and when using the Ewellix Lift make sure it is enabled in the equipment page.

Chapter 8

Appendix B - FAQ

Below is a list of common questions and answers for the Standard Bots system

Q: Do you need to have the robot connected to the internet to use the robot?

A: No, the robot can run on a local network or simply with a single ethernet cable connected between the robot and a PC or tablet. If the robot is not online it will still operate without issue, however remote diagnostics and automatic updates will not be available.

Q: Do you need to have the Ipad connected to the robot to run a program?

A: No, the Ipad is only needed for setting the Network setting for the robot. Once the robot is connected to the network, you can use the Ipad to control the robot or a computer using Google Chrome or by using the 24V External Control Interface.

Q: How do I connect the robot to the network?

A: The robot can be connected to the network by going to the robot menu in the Ipad application and selecting the network icon. From there you can select the network you want to connect to and enter the password.

Q: How do I access my robot from a computer?

A: If the robot is on the internet: You can access the robot from a computer by entering the robot name into the address bar of Google Chrome. You will then be prompted to enter the password for the robot. This will be cbXXXX.sb.app where XXXX is the last 4 digits of the controller serial number. This number can be found inside of the controller door.

If the robot is not on the internet: Use the Ipad application to find the ip address of the robot. Then, in the browser of a computer on the same local network of the robot type in the robot ip address port 3000 as shown in the Ipad application. Example: 192.168.110.20:3000.

Q: How do I access the network settings after initial setup?

A: To access the network settings of the robot, for example if you want to set a static IP address, use the Standard Bots application on the tablet. If you are already logged into the robot: In the User Interface on the tablet go to the Move Robot view. Go to the robot name menu in the bottom left and select “Logout”. You will now be back to the tablet application main screen. On the left side you can select your robot. Once selected, the network settings will be on the right side.

Q: How do I connect my robot to a tablet or PC directly?

A: You will need an ethernet cable to go between the robot ethernet port and your pc or tablet. You may also need a usb c to ethernet or similar adapter.

If you are simply using a wire, there will not be a router to give both devices an IP address. This means you will need to set the address manually, and set them on the same subnet.

Using the Standard Bots application on the tablet, set an IP address on the wired port of the robot under the network settings in the application. Then set a static address on the pc or tablet. Generally the address follows the format of 192.168.x.y. x would be the subnet, and both the robot and pc or tablet need to have the same number for x. Y would be unique for each device. Ex, the robot could be 192.168.110.5 and the pc or tablet could be 192.168.110.6.

Then on the pc or tablet go to the robot address you assigned in a browser tab. The format is generally 192.168.x.y:3000.

Q: Why is an axis direction grayed when the robot is not fully extended?

A: When the robot has one or more joints that are at their limit, this can prevent the robot from moving in a certain direction in the XYZ plane. The robot does not need to be fully extended in one direction to cause this. It is simple to fix, go to your move joints tab, then rotate the joint that is at it's motion limit (the joint that is maxed out in the positive or negative direction). Once done the robot should be able to move with more freedom.

Q: How do I connect my robot to a tablet or PC without setting static IP address and wires but not have the robot on the internet?

A: You can setup a simple local network for the robot using a standard wireless router. You will need to purchase a wireless router. Ensure it is not just an access point, as an access point will not assign IP addresses.

Setup a standard wireless network on the router. Do not connect the router to your LAN or internet. Connect the tablet to the wireless network. Use the iPad application to connect the robot to the new wireless network. In the tablet, check the network status for the robot Ip address, which will be in the format xxx.xxx.xxx.xxx:3000. Any device on your new wireless network can access the robot using this address. The cbxxxx.sb.app link will not work if the robot is not online.

Q: Which end effectors work with the RO1?

A: The following tools are supported directly in the Standard Bots interface:

Onrobot 2FG7 Parallel Tool

Onrobot 2FG14 Parallel Tool

Onrobot 3FG15 3 Finger Centric Tool

Onrobot Dual Changer

Schunk EGU/EGK

Standard Bots also supports any tools that are discrete IO driven, including Onrobot tools that are not listed above through the use of the OnRobot Compute Box.

Currently, Robotiq tools are not supported as they cannot be controlled over discrete IO.

Q: How do I change which tool I want to use on the RO1?

A: To change which tool is on the robot, go to the robot menu (has the robot name in an oval) and go to equipment. You can then add or delete tools.

Q: Which protocols does the RO1 support?

A: Currently, the RO1 supports 24V discrete IO and Modbus TCP. There is also a REST API, a python SDK and typescript API.

Q: What are the power requirements of the RO1?

A: The control box requires 90 ~ 264VAC, 47-63Hz. The robot draws less than 15 amps at 120V.

Q: How do I get started with a new RO1?

A: Please start the included iPad, open the Standard Bots application, and follow the instructions.

Q: What is the payload capacity of the RO1?

A: Depending on speeds and move types, the RO1 can carry up to 18 kg.

Q: What is the maximum speed of the RO1?

A: The RO1 joints can move up to 435 degrees / second.

Q: What is the maximum reach of the RO1?

A: 1.3 Meters

Q: What is the bolt pattern at the end of the arm?

A: ISO 50mm pattern. The pattern follows the standard and has 4 m6 bolts on a 50mm circle.

Q: How do I move a routine to a different robot?

A: Open the routine you would like to copy in the first robot. Select the ... menu, and select “schema”. Copy the text in the window. Open up the second robot and create a routine. Again select the ... menu and “schema”. Copy the text into the window, rename the routine and select “create routine”.

Q: How do I import a model into the environment?

A: To import a model, go to the Move Robot view and go to the space area. The icon is in the bottom right of the view and looks like a square. In this view select the “+” to add an item, and select “Object”. The UI will tell you which file formats are supported.

Q: How do I get a Developer API Token?

A: The developer API token and setting can be found in the robot menu. Go to the Move Robot page, click on the robot name, settings, Configure developer API.

Q: My robot says it is online but I cannot access it?

A:

1. Check the device you are accessing the robot from (Ipad, PC) is on the same network as the robot.
2. Check that the network the robot is connected to does not have a “splash” page you need to log into. This is often the case with guest networks. The robot does not support web based logins like this.

Q: I imported a model as an object but I cannot see it?

A:

- Try changing the scale. The import is in Meters. Often you must convert to meters by setting the scale to .001.
- Check that the file is valid

Q: How do I get the robot back into the box or case?

A:

1. Ensure all tools are disconnected and that the robot has sufficient space to move
2. Power on the robot
3. Go to the Move Robot page of the interface (where you jog the robot)
4. Click on the robot name menu (where the speed is)
5. Go to Settings
6. Go to Box Robot
7. Hold the Prepare Robot For Shipping button down to bring the robot to the box position.

Q: What should I do if I forget my robot's passcode/PIN?

A: Please reach out to Standard Bots at live-support@standardbots.com. Be sure to provide the serial number of your robot controller (e.g., cbXXXX), which you can find inside the controller door.

Q: Why are the safety settings greyed out, and how do I unlock them?

A: The safety settings will appear greyed out until access is unlocked. To unlock them, click the padlock icon labeled "Unlock" in the top-right corner of the safety settings pop-up window. You will then be prompted to enter your passcode. Once the correct passcode is entered, the safety settings for the RO1 will become editable. Don't forget to click "Apply Settings" and confirm the changes to ensure they take effect..

Q: What is the difference between "Connect" and "Connect Remotely"?

A:

- "Connect" uses the local IP to connect to your robot. It does not go out to the web
- "Connect Remotely" goes over the web. The same as if using the `cbXXXX.sb.app` link in a browser

Chapter 9

Appendix C: How to direct wire robot

9.0.1 With an Ethernet Cable:

To get started, you need the following:

- Ipad with USBC port
- This dongle: (https://www.amazon.com/Anker-PowerExpand-Ethernet-Delivery-MacBook/dp/B08C9HZ5YT/0VSifNPtOpRCm9k0Pgk81DqptbuYVAvaEKRMP-UhBFZHKuZUIuB_wz7-MWZbbQMoxg370mhMgsfV91sjZ2gcg1Kze1TwencY87-1QjF26kEMuMd8PVnUtUrRcqWQLFd-7wosHDFfaecURmNZ6fvv0Uc85HZvINhgo2RMW3)
- Any ethernet cable like this one: (https://www.amazon.com/Amazon-Basics-Ethernet-Internet-Meters/dp/B00N2VIWPY/ref=sxsts_b2b_sx_reorder_acb_customer?content-id=amzn1.sym.93dc5330-3850-4554-84ea-5a8297a3e1d7%3Aamzn1.sym.93dc5330-3850-4554-84ea-5a8297a3e1d7&crd=2H9DWCfff9-42a6-b190-141db76338df&pd_rd_w=PCIKj&pd_rd_wg=4pr6l&pf_rd_p=93dc5330-3850-4554-84ea-5a8297a3e1d7&pf_rd_r=VWW6EFT6W3F89ZJR33ZY&qid=1745500256&sbo=RZvfv%2F%2FHxD1-76dd0896-e18a-4d36-b921-52fc00e57159&th=1))

Initial Setup:

1. Power up control box
2. Connect USBC adapter to Ipad
3. Connect ethernet cable to USBC adapter
4. Connect the other end of the ethernet cable to the control box
5. Open Ipad
6. Open the standard bots app
7. Click add robot
8. Wait for the robot to show up, if it doesn't check under the cobtrol box to make sure the antennas are on the box



9. Click through until it asks about “network”
10. Go to ethernet tab

< cb2095

Network

✓ Wi-Fi

✓ Ethernet

Advanced

Assign IP

Manual ▾

IP 192.168.1.2

Subnet Mask 255.255.255.0

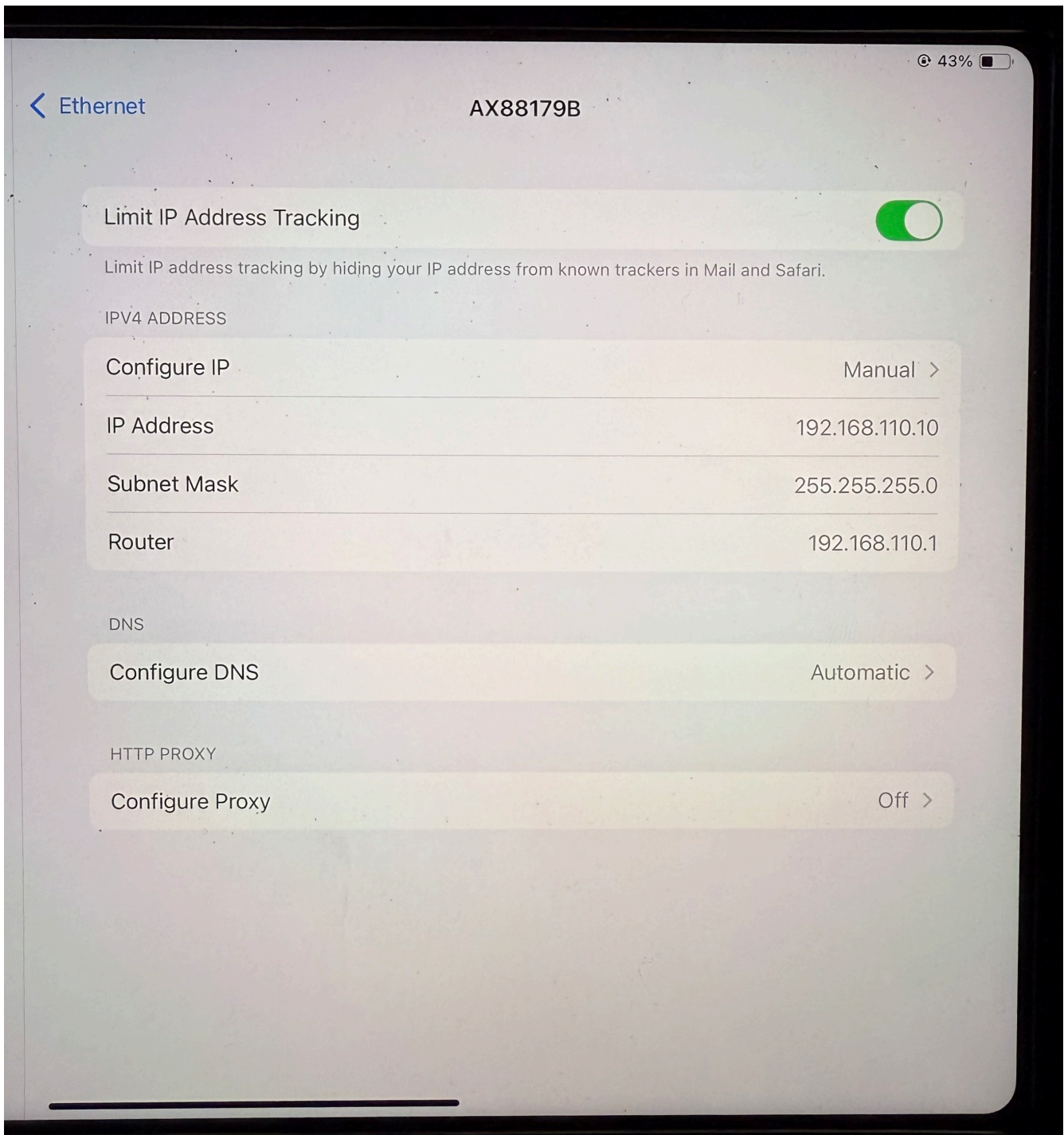
Router 192.168.1.1

Assign MAC

Automatic ▾

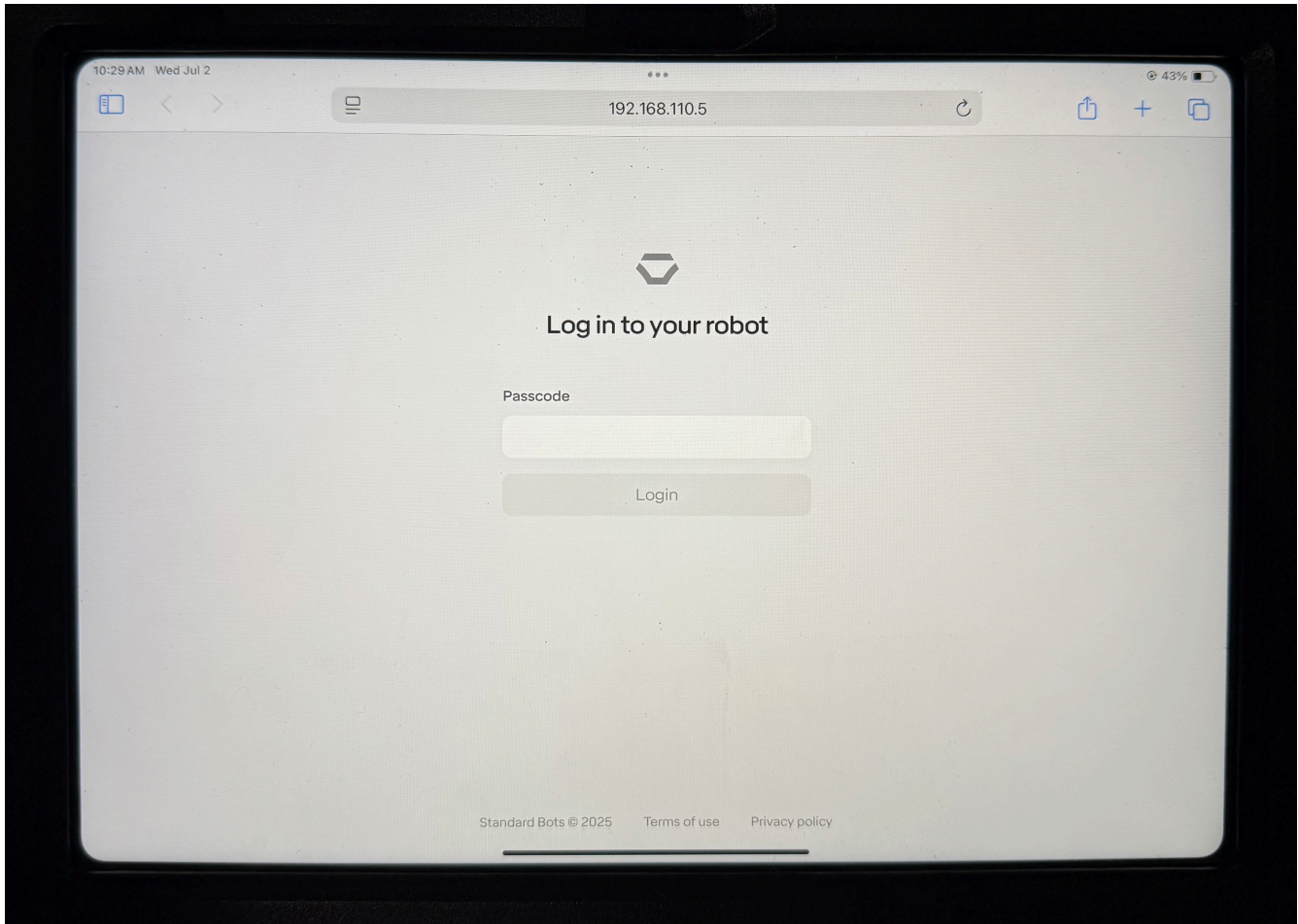
Disconnect

11. Change to these settings: Manual/Auto: Manual IP: 192.168.110.5 Subnet Mark: 255.255.255.0
Router: 192.168.110.1 Leave all other as is
12. Click connect on the bottom of the screen
13. Then on the Ipad go to settings
14. You should now see a settings for ethernet
15. Go to ethernet settings



16. Change to these settings: Manual/Auto: Manual IP: 192.168.110.10 Subnet Mark: 255.255.255.0
Router: 192.168.110.1 Leave all other as is

17. Don't forget to save these changes
18. Open safari or any web browser
19. Type in 192.168.110.5:3000
20. Hit enter
21. UI should then come up



22. Bookmark this page
23. If the Ipad goes to sleep , you may have to unplug and replug in the USBC adapter to get it to wake up. BUT you should never have to reset any IP's on the robot or Ipad as long as you use the same adapter.

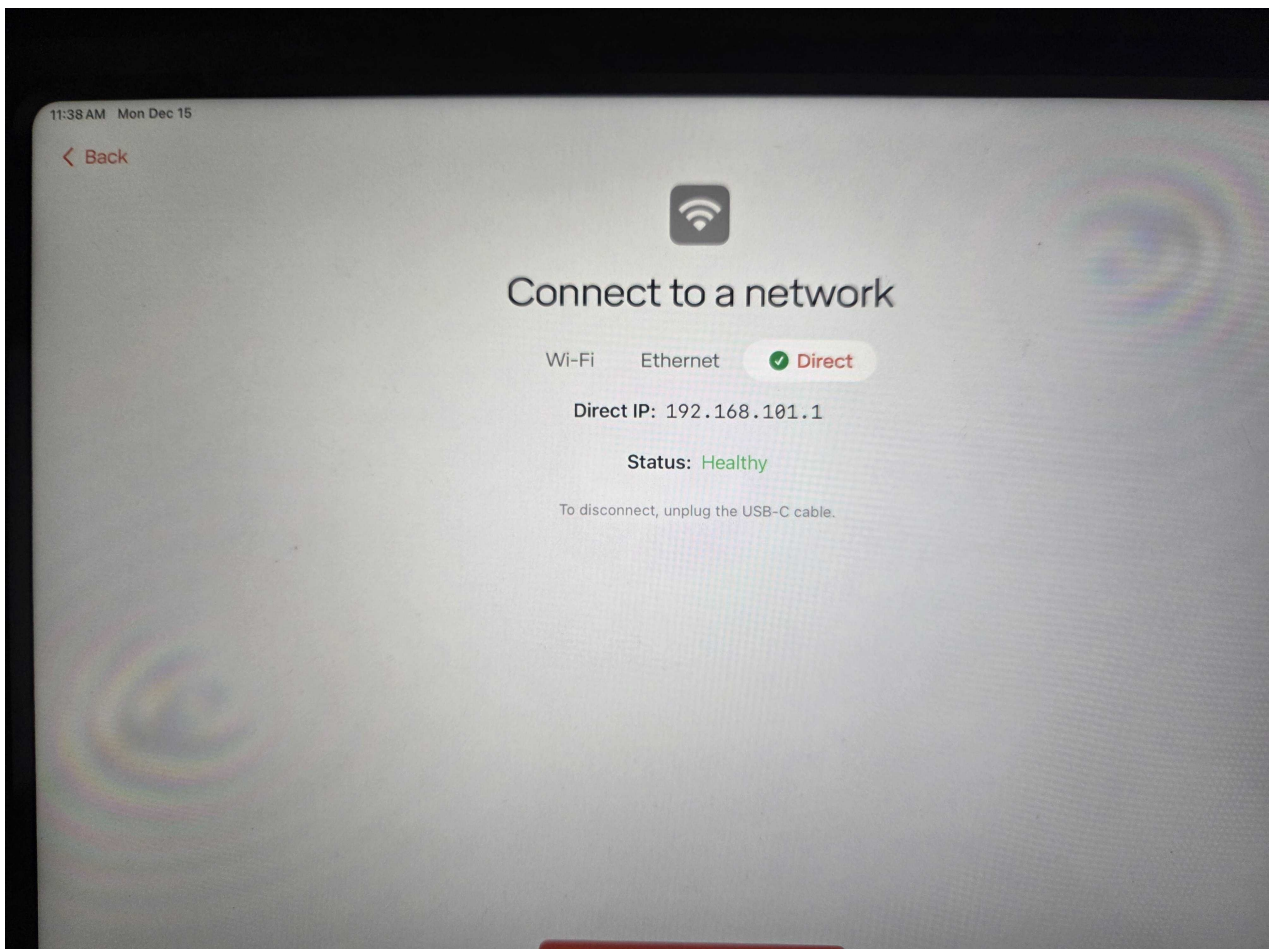
Next time you set the robot up:

1. Turn on the control box
2. Wait for the LED on top of the box to turn green, if it does not turn green after 3 min press it.
3. Connect USBC adapter to Ipad
4. Connect the ethernet cable to both the adapter and the control box
5. Go to your web browser and open the bookmark you previously saved
6. Use the robot!

9.0.2 Direct Connection (using a USB-C Cable)

Initial Setup:

1. Power up control box
2. Connect the ipad to the Tablet USB port on the bottom of the control box with a usb-c cable. At this time, the port is only one directional. If you do not see the ipad start to charge, flip the end of the cable 180 degrees on the control box side.
3. Go to the settings in the ipad and set the ipad IP to automatic (this should be the default on the ipad)
4. Open Ipad
5. Open the standard bots app
6. Click add robot
7. Wait for the robot to show up
8. Click through until it asks about “network”
9. Go to the option that says “direct”. You will see that the status above says “healthy” if the steps above were followed correctly.



10. Continue to the next page. You can now use your robot.



Standard Bots Company
80 Pratt Oval
Glen Cove, NY 11542 USA

www.standardbots.com
1-888-9-ROBOTS