READY

FORGE

User Manual

page left intentionally blank

The documentation, best practices, and recommendations provided by READY Robotics do NOT constitute safety advice. Products sold through READY Robotics are not by themselves a fully integrated workcell. As required in ISO 10218-2, READY Robotics strongly recommends performing a complete risk assessment of the integrated workcell per ISO 12100. You may wish to use the methodology found in the ANSI/RIA TR R15.306 Task-based Risk Assessment Methodology.

Contents

Safety Overview

Forge/OS can be used to control a variety of robotic arms and devices. Although some robot arms are collaborative and designed to be used in proximity to people, you should never consider them inherently safe. Additionally, there are other safety considerations that come with any system that uses electric and air power for high-speed handling of parts.

Whether the arm is collaborative or not, there are some guidelines that should be considered when working with a robot arm. For more manufacturer specific safety guidelines, please refer to the manufacturer's user manuals and safety documentation.

- Know the motion limitations of your system and be sure to remove any objects that could be easily damaged or destroyed from the reach of the arm. See your robot arm's manual for specific details.
- The safest way to stop the robot arm is with the emergency stop button located on the teach pendant.
- The main power switch is **not** an emergency stop switch. Do **not** use the main power switch for emergency situations.

Note: Be sure to follow **all** the safety rules and regulations established by your facility when using Forge/OS.

Robot Safety Modes

Depending on your set up, there are different safety modes you may experience during operation. The most common safety mode is an emergency stop. If you are using a collaborative arm, the robot arm may also enter a protective stop. In Forge/OS, these safety modes trigger Robot Errors and Robot Warnings, respectively, which will prevent you from moving the robot until an issue are addressed and the stop is cleared. Attached safety devices may also trigger safety stops.

Emergency Stop

The robot arm enters an emergency stop when an emergency stop button is pressed, when a high-speed collision occurs, or when certain safety equipment is triggered. When in an emergency stop:

- The robot arm applies internal brakes to ensure rigidity.
- Attached end-effectors and devices release grip and turn off.
- Forge/OS notifies you the robot is in an emergency stop and the Mode status bars are red and display "Robot Error."

If an emergency stop occurs because of an emergency stop button, Forge/OS instructs you to turn the

emergency stop button clockwise, releasing the stop. Once the stop is released, the robot arm is ready to use.

If an emergency stop occurs because of a high-speed collision, Forge/OS instructs you to clear the stop through the teach pendant, connected to the controller of the robot arm. Follow all on-screen instructions to restart the robot arm from the teach pendant. Once the stop is released, the robot arm is ready to use.

Protective Stop

The robot arm enters a protective stop when it encounters an obstacle that triggers the human-safe force sensors inside the robot arm. When in a protective stop:

- The robot arm applies internal brakes to ensure rigidity.
- Attached end-effectors and devices release grip and turn off.
- Forge/OS notifies you the robot is in a protective stop and the Mode status bars are yellow and display "Robot Warning."

To clear the protective stop and turn on the robot arm, tap **Fix** from the **Notification** panel. See the Notification Center section for more details.

Introduction to Forge/OS

Forge/OS is the READY operating system through which you can interact with and program the robot arm and other automation devices. It enables you to create, save, and execute Tasks autonomously through the Task Canvas. It also lets you manually configure and control attached devices, and communicate job information from an automated cell to help track productivity.

Important Features

Forge/OS has several important features for configuring and controlling your automation equipment. **Hardware Configuration** enables you to select and modify attached devices. It will open when you first power on your system. **Task Canvas** allows you to assemble the blocks and flowchart to create an executable Task for all of the devices configured in Hardware Configuration. The **Control Suite** gives you access to system settings and manual controls for the robot and connected devices.

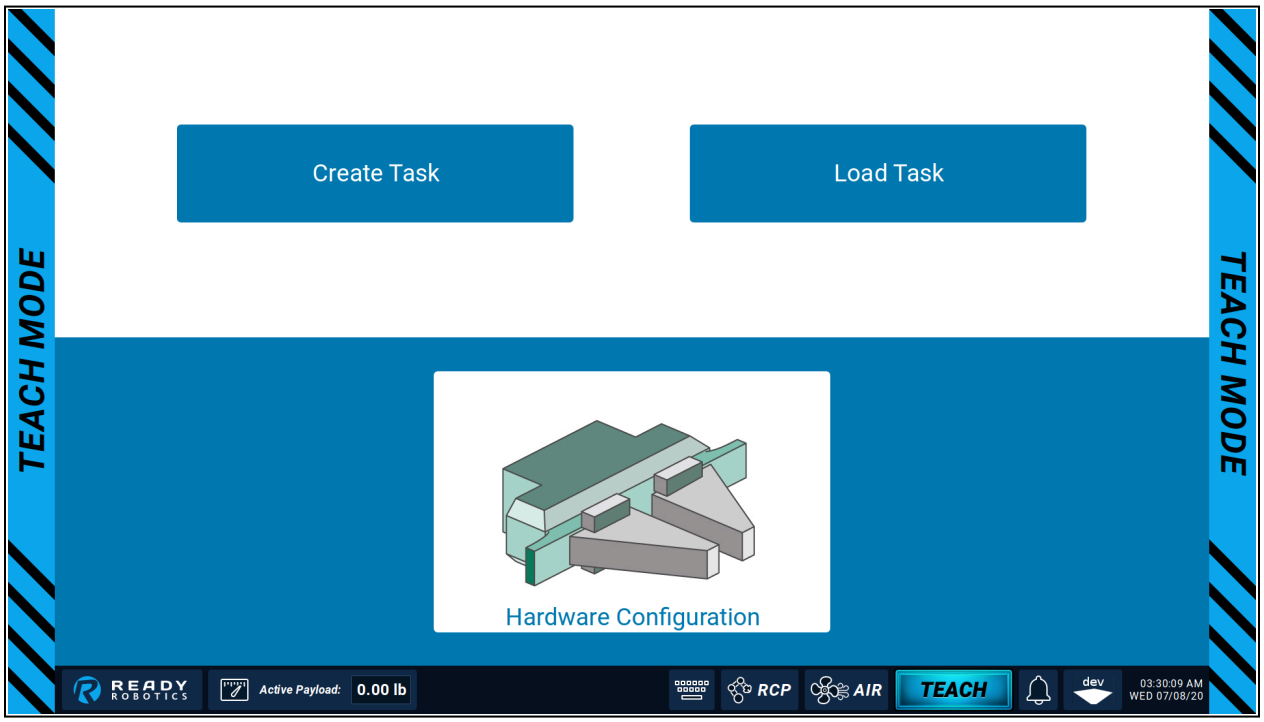
The first time you boot up Forge/OS you will go through the Initial Configuration process to set up basic system

settings, the attached robot arm, and peripherals specific to your Task. For information about the Initial Configuration of your Forge/OS system, see the next section.

After completing Initial Configuration, Forge/OS will boot to Hardware Configuration, where the system devices are setup. For information about setting up tools and devices, see the Hardware Configuration section.

The Home Screen

After initial setup and hardware configuration, you will have access to the Home Screen. From this screen, you can start a new task, load an existing task, or enter Hardware Configuration to edit the current loadout of your system. Additionally, you can access the Control Suite at the bottom of the screen, which is described in detail in the Control Suite section of this manual.

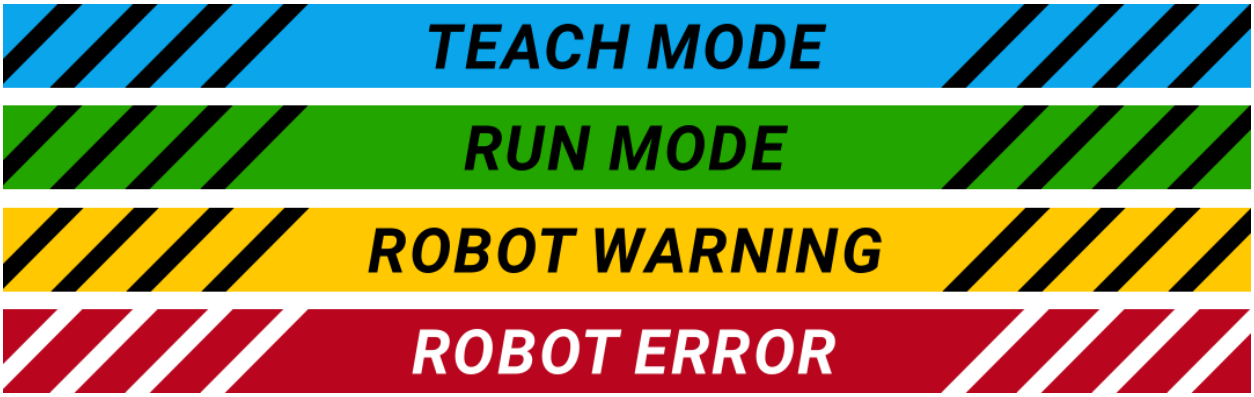


Element	Description
Create Task	Open Task Canvas with a new Task
Load Task	Access the Current Tasks screen and view the saved Tasks on your system.
Status Bar	The left and right edges of the screen display the current mode of the Forge system.

Element	Description
Control Suite	The bottom of the screen gives you access to System Settings and manual controls for the robot and attached devices.

System Modes

The current mode of the Forge system is indicated by the Status Bars on the left and right edges of the screen. If you have a TeachMate, it will also indicate the current mode with the colored LED.

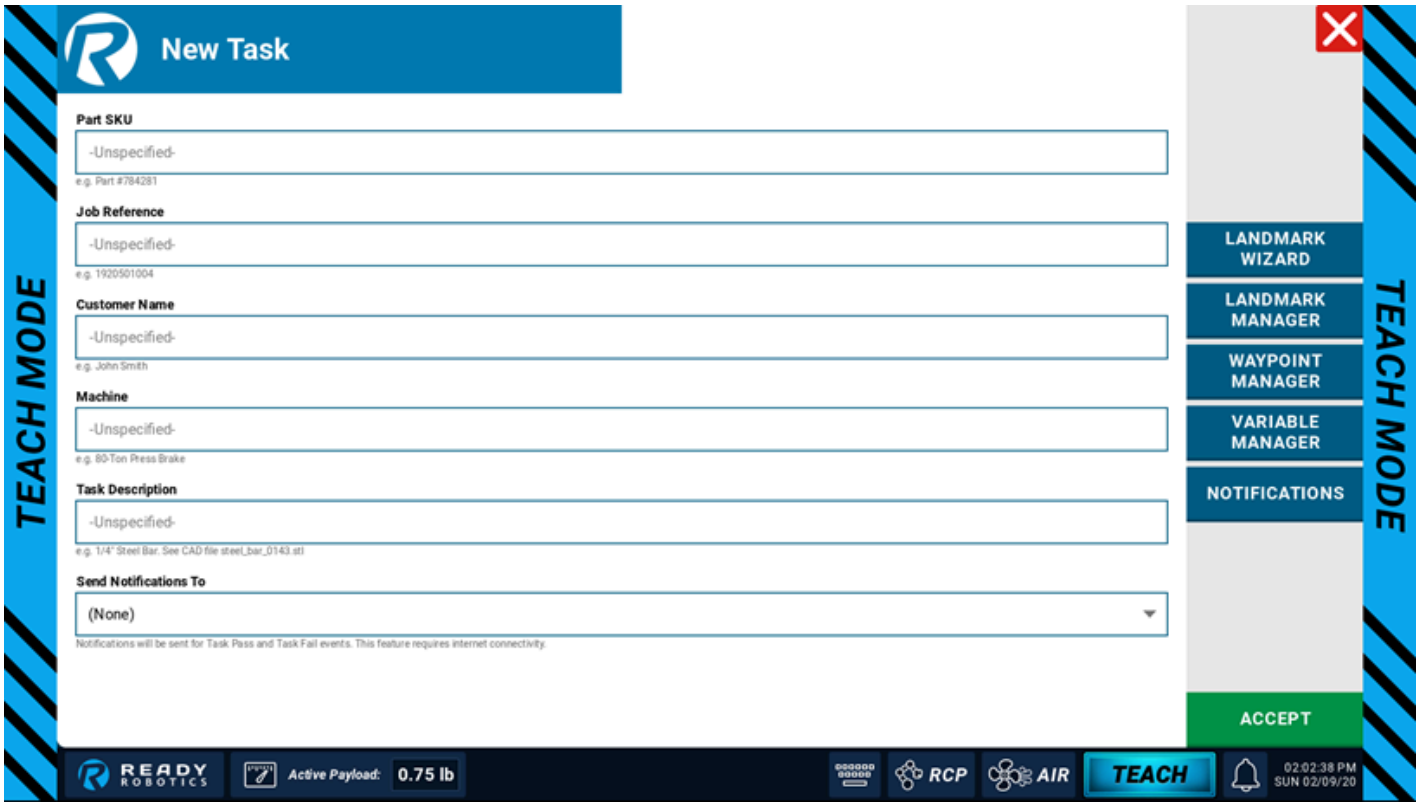


Element	Description
Teach Mode	Programming Tasks and executing Task actions at safe speed. In Teach mode, you can Step blocks on the Task Canvas and move the robot arm at a reduced speed. You can also manually control attached devices and pneumatic ports.
Run Mode	Executing Tasks at full speed. Depending on your hardware and safety configuration, all safety devices must be active and functional to enter Run mode.
Robot Warning	Cannot move the robot arm. You can access manual controls for certain devices in Robot Warning mode and you can still program all block types in Task Canvas. On collaborative robot arms, Robot Warning is also triggered by the robot arm's Protective Stop mode.
Robot Error	Indicates that something is wrong with the Forge/OS system, robot arm, or attached devices. You cannot execute any functions in Robot Error and will have limited functionality in some applications, such as Hardware Configuration.

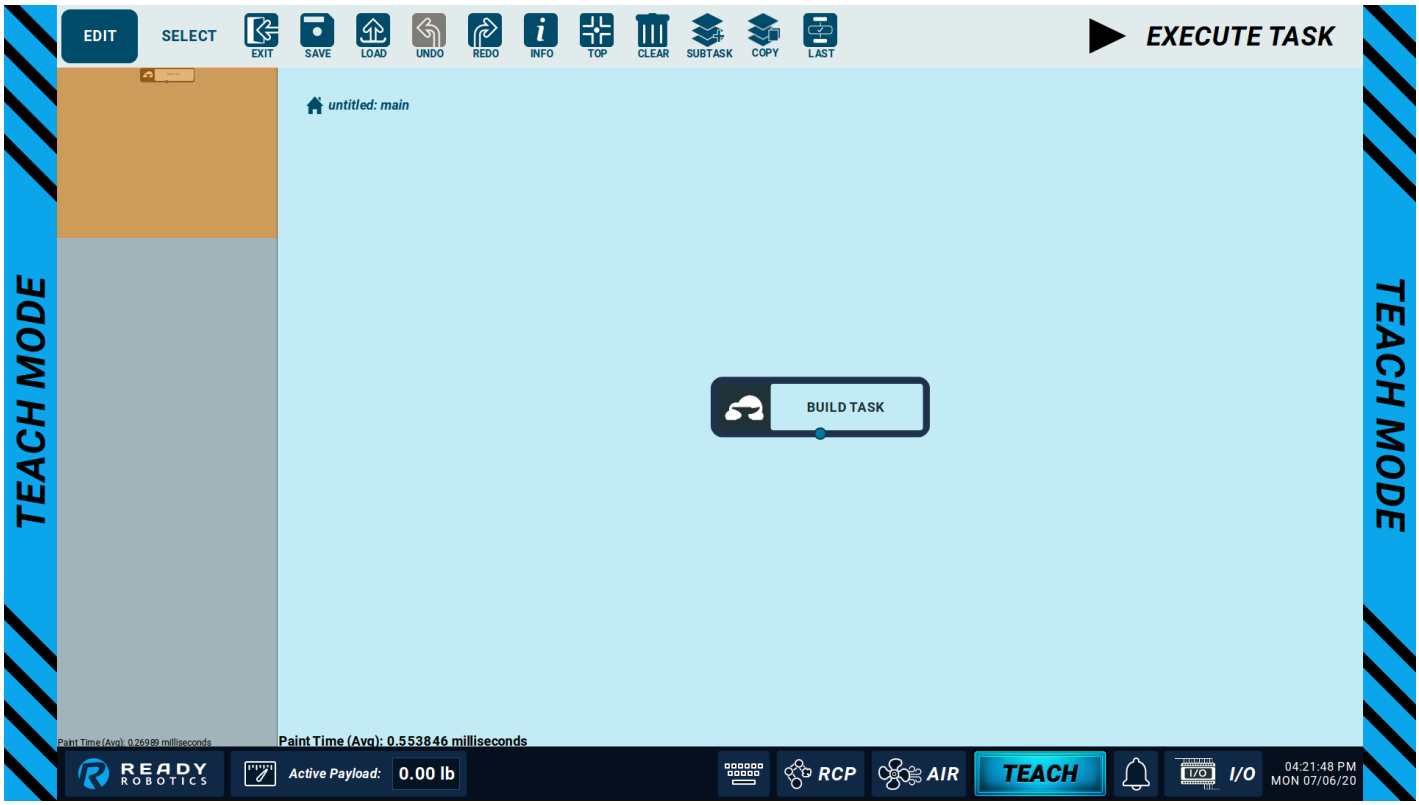
Element	Description
Guide Mode (some collaborative robots ONLY)	Allows for moving the robot arm by hand.

Creating Tasks

When you begin to create a new task, Forge/OS prompts you to enter information about the task. The information you provide about the task helps you and other operators specify and identify different tasks for different jobs, machines, parts, and customers. You can also select or provide an email address to receive notifications about the task. Actions in the task that trigger a notification send an email to the address provided.



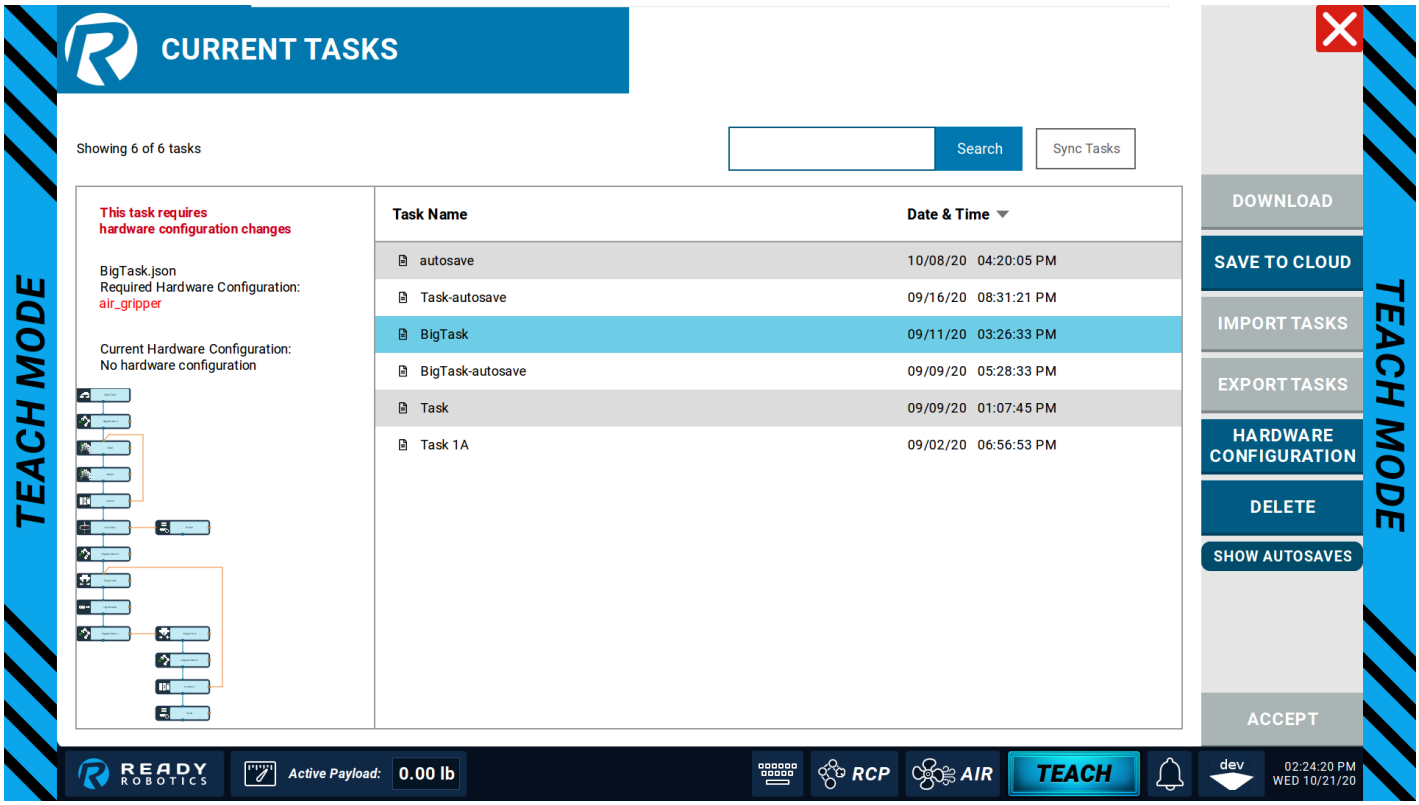
When you are satisfied with the information provided, you can tap **Accept** to open the Task Canvas, where you will program the task. You can always make changes to the Task Info later. Be sure to save your task using the **Save** button on the Menu bar at the top of the Canvas, which will prompt you to enter a name for the task.



Loading Tasks

To load a saved program, tap **Load Task** on the Home Screen. The resulting screen displays a list of tasks saved in Forge/OS, or Current Tasks. The list can be sorted alphabetically by name or by the date and time created, making it easier to find specific tasks.

When a task is selected from the list, the Load Task screen displays information about the task, including the required hardware configuration (the tooling/devices needed to execute a task) and a preview of the tasks' plan. In addition to the required configuration for the selected task, the Load Task screen displays the current configuration.



You cannot load a task without the proper configuration for that task. If the current hardware does not include the end effector or device required for the selected task, the end effector or device appears red in the Requirements section of the preview on the left side of the screen. When this happens, you will be unable to load the task. To change the loadout and load the task, you must attach the required end effector or device and update the hardware configuration (see the Hardware Configuration section for more information).

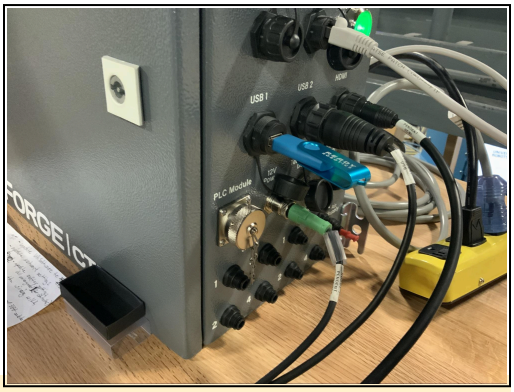
Autosaved Tasks

If your machine ever crashes or you forget to save your task, you may load an autosaved version of the task you were working on. When a task is open on the Canvas, Forge/OS automatically saves task changes. By default, autosaved tasks aren't visible in the Current Tasks screen. Tap the **Show Autosaves** toggle on the right to make autosave files visible. Autosaved tasks will display as task name-autosave. If the task wasn't given a name when the autosave file was created, the task name displays as autosave.

Note: Autosave files eventually delete themselves and are not available indefinitely.

Importing and Exporting Tasks

You can import and export tasks from an external thumb drive within the Current Tasks screen (see Loading Tasks), using the **Import Tasks** and **Export Tasks** buttons. You may use these features to:



- Back up tasks to a thumb drive
- Import a task template
- Reuse tasks across multiple systems
- Transfer tasks to a replacement system

Note: You must insert a USB flash drive into the Forge Controller to import or export tasks. The features will not be accessible unless a USB is detected.

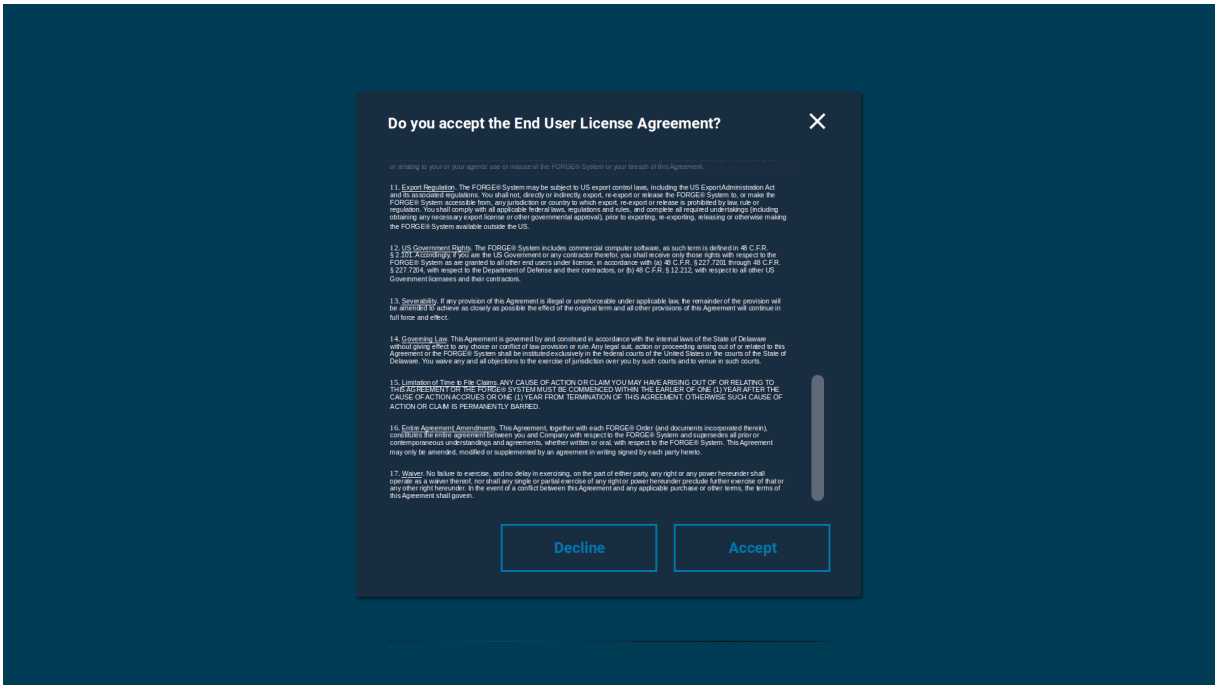
To export to a connected USB drive, tap **Export Tasks** and select the task(s) you would like to export. To import tasks from a USB drive to the controller, tap **Import Tasks** and select the task(s) to import. You can use the import feature to download a pre-programmed task template before configuring that task to your specific application.

Initial Configuration

When you power up Forge/OS for the first time, you will have to configure the system settings and set up the attached hardware. Initial Configuration provides the necessary screens for customizing your Forge system for your cell. It will guide you through the following settings prompts. Most of the settings chosen here can also be accessed and updated once the configuration is complete. For turnkey-cell specific instructions, refer to an appropriate Application Guide.

1. **End User License Agreement (EULA)** - Read and accept the Forge/OS user agreement.

   Note: After the initial setup, you will need to read and accept the EULA every time the system boots up.



2. **Connect to a Network** - Connect to the internet through WiFi or a direct connection. If your Forge system is connected to the internet, you can receive remote software updates from READY, program tasks to email or text you when certain actions occur, and send bug reports for technical support. You can skip this step for now if there are issues or if you would rather set that up after initial configuration.

3. **Create an Admin PIN** - Set up an administrator PIN for switching between the administrator and operator roles in Forge. IMPORTANT: Do not lose or forget the pin that has been created for the Forge/Station. Otherwise you will have to contact READY to reset it. The READY support number is 1(833)732-3977

4. **Select the Time Zone** - Set your time zone to update the system clock display and give READY support accurate time-stamped data when providing technical support.

5. **Configure a Robot** - Tell the Forge system which robot is in use.
6. **Force Sensor Check** - Configure an optional force sensor at the end of the robot arm.
7. **Set up a TeachMate** - Configure the optional READY TeachMate type and mounting position.
8. **Complete Forge/OS Setup** - Configure the robot arm controller to work with Forge system.
9. **Robot I/O Configuration** - Transfer the necessary information between robot arm controller and the Forge system. Refer to your robot specific Application Guide for this step.
10. **Complete Setup** - Review and accept the Initial Configuration settings to finish.
11. **Welcome to Forge/OS** - Wait for Forge/OS to load, which may take a few minutes.

Edit Initial Configuration Settings

In order to edit the steps which were skipped in the Initial Configuration, tap on the READY Robotics icon on the bottom left of the monitor.

This opens the READY menu. From there, you can access and change the following initial configuration settings:

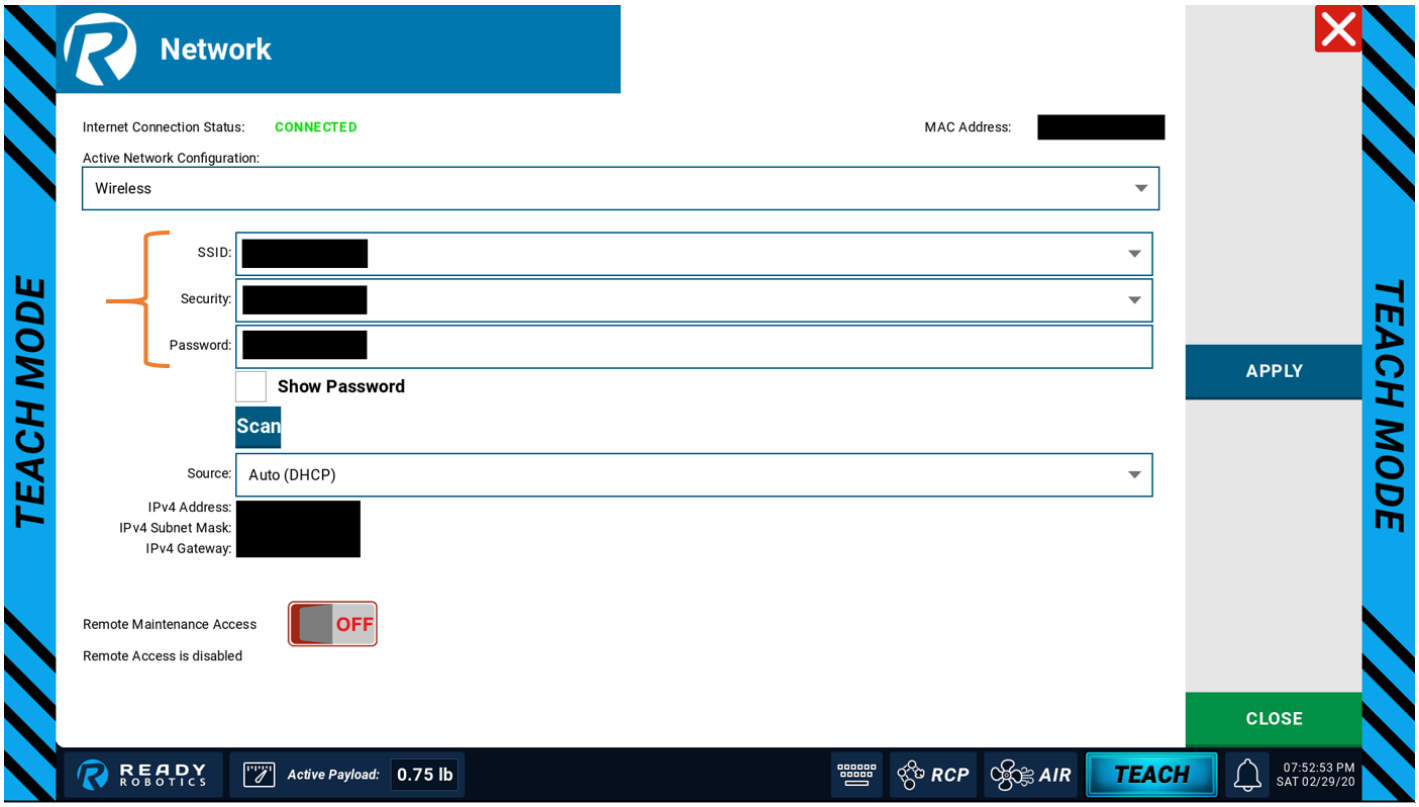
- **USER ROLES**
- **NETWORK**
- **TIME ZONES**

Network Configuration

After initial setup, the Network settings can be accessed through the READY Menu as described in the last section. This screen enables you to connect Forge/OS to the internet. Forge/OS must be connected to the internet for cloud and email services to work. An internet connection also enables remote system updates and customer support. Your Forge/OS system may have both Ethernet and WiFi access.

Use the Scan feature to generate the list of available networks. For wireless networks, be sure to select the appropriate security type. Finally, tap **Apply** to connect to the chosen network after entering the password. Then once the status changes to "connected" you may close the Network window. If you have issues

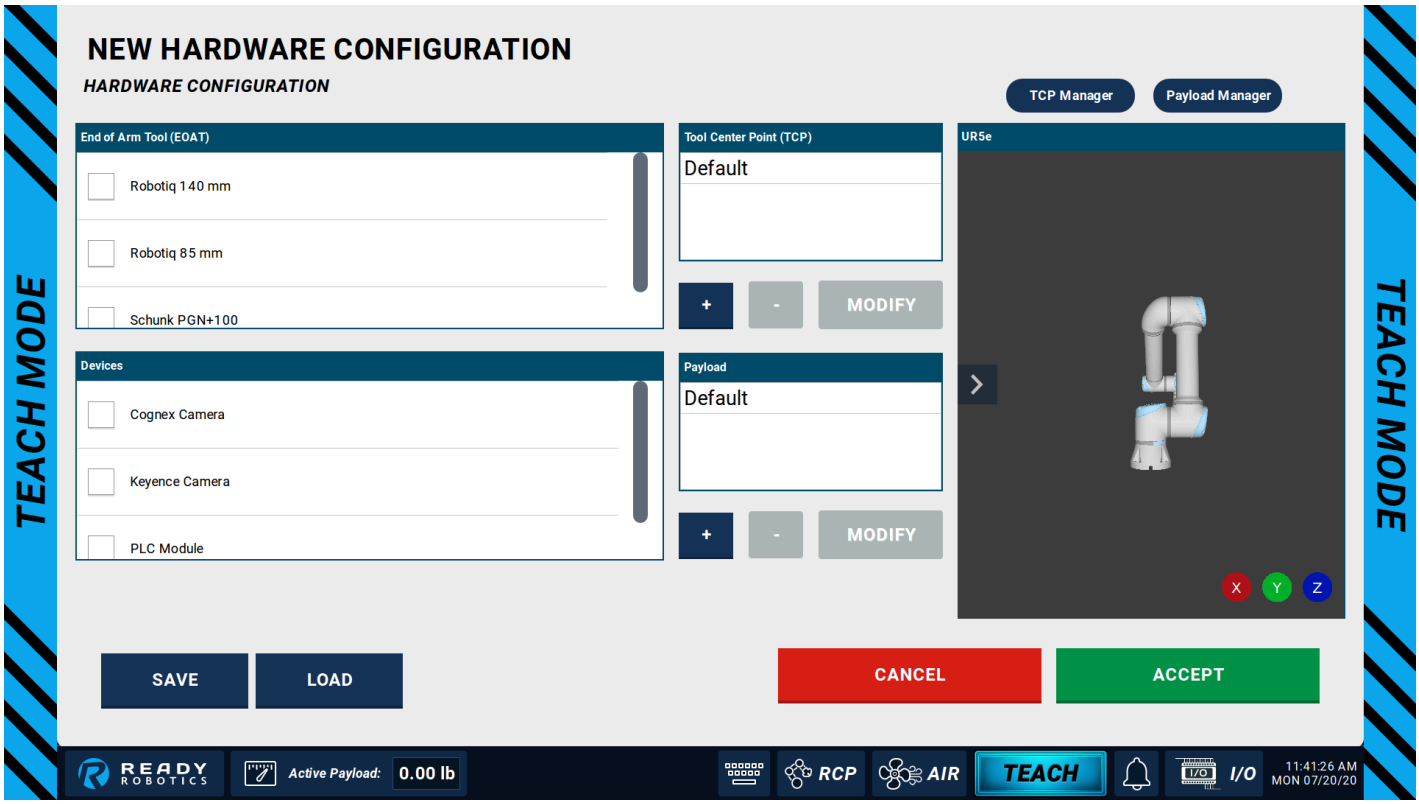
connecting to your network, first consult your facility's networking/IT manager. Then reach out to READY support if still you cannot resolve the issue.



Element	Description
Active Network Configuration	Select the type of network you would like to connect to. For WiFi connections, enter the Network name (SSID), security type, and password. If your internal network automatically assigns IP addresses, select Auto (DHCP) for source. If you are unsure about this setting, leave it as Auto (DHCP).
Remote Maintenance Access	Allows READY Support access to troubleshoot and update your system remotely. READY recommends you leave this setting OFF if your network does not have advanced security settings and only turn it on when instructed to by a READY support representative.

Hardware configuration

The Hardware Configuration defines the tools and devices attached to your Forge system. Throughout Forge/OS, different screens will present you with options for controlling only the tools and devices that are configured in the Hardware Configuration, limiting available controls to those relevant to your hardware and hiding those that are not. The options available in this application may differ depending on your robot manufacturer and model and the information you entered when configuring your Forge system for the first time.



Element	Description
Device Selection	Select pre-configured tooling and peripheral devices; create/modify and select custom Tool Center Points (TCPs) and Payloads to be used by Forge/OS.
3D Visual Rendering	View the visualization for a loaded TCP and Payload (center of mass).
Save	Save the current configuration to the Forge system. All parameters, including tools, devices, TCPs, and Payloads, will be saved to the configuration.
Load	View, load, and delete existing configurations

Element	Description
Accept	Apply the configuration to the Forge system.
Cancel	Cancel any changes and revert to the previous settings.

End of Arm Tool (EOAT)

The End of Arm Tool, or EOAT, is the tool attached to the robot tool flange. The tool can be controlled electrically, pneumatically, or otherwise. If the exact tool you are using is not shown in the available options, you may be able to control it using a tool with a similar method of actuation.

Select the checkbox next to a tool to open its configuration and tap **Accept**. Certain tools, such as pneumatic grippers, will require you to select the air ports on the Forge system to which you must attach the air hoses. Tap **Modify** for a selected gripper to change its settings.

It is recommended that you add a TCP and Payload for the EOAT to your configuration, as the Default TCP and Payloads do not account for the EOAT.

Note: The default Payload and TCP values do not account for items in the end of arm stack other than the tool, such as a force-torque sensor or tool changer. You must add the geometry and mass of these items to the pre-loaded values, as described in the sections following *Devices*.

Devices

The Device list shows tools and peripherals that the Forge system can control that are not considered EOAT. The Device list will also show custom network devices that you created from imported configuration files, if

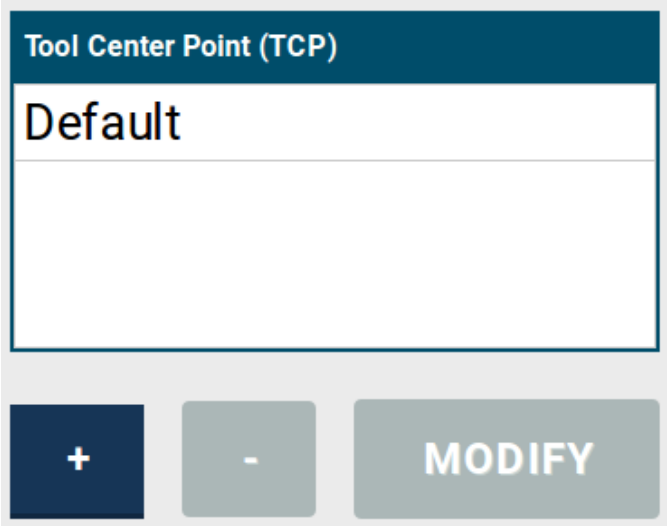
applicable.

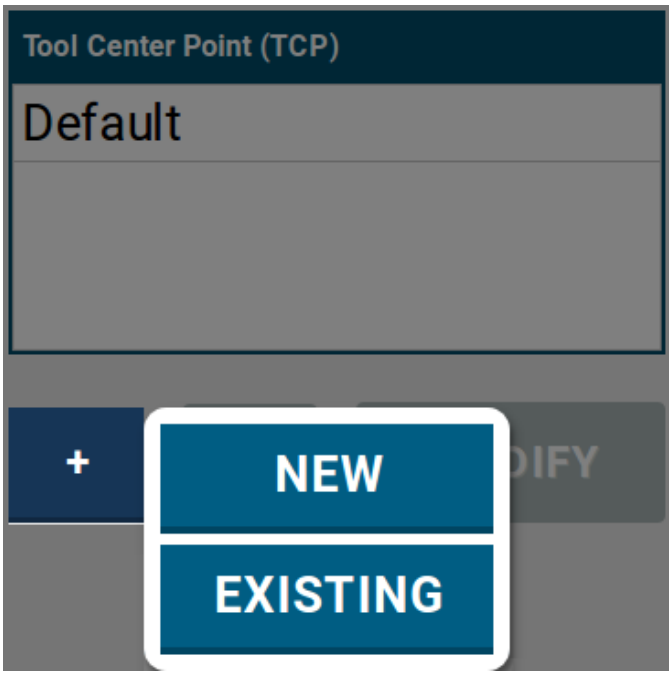
Select the checkbox next to a device to open its configuration. For instructions on configuring certain devices, see their respective Forge or OEM documentation. Tap **Modify** for a selected device to change its settings.

Tool Center Point (TCP)

The Tool Center Point, or TCP, is the point at which the robot arm assumes the attached tool is doing work. The TCP will define Waypoints, the *tool* reference frame, and the point around which the end effector will rotate. Therefore, defining the correct TCP is crucial for precise motion control. See the section *Waypoints, Landmarks and TCPs* in this manual for more on the importance of the Tool Center Point.

The factory default TCP for every robot system is at the center of the flange at the end of the robot arm. All custom TCPs are defined with respect to the default TCP. You can select a loaded TCP to view it in the 3D robot rendering. To load a TCP into the Hardware Configuration for use, tap on the **Plus** sign under the TCP selector, then tap **Existing** to choose from the existing options. To define a TCP and add it to the current configuration, tap on the **Plus** sign, then tap **New**.





Element	Description
Add (Blue Plus)	Provides the options to tap on **NEW** or **EXISTING** to create a new TCP or add an existing TCP to the configuration.
Remove (Blue Minus)	Removes the selected TCP. Removing a TCP from the configuration will not delete it from your system.
Modify	Change the values of the selected TCP. You cannot modify the default TCP.

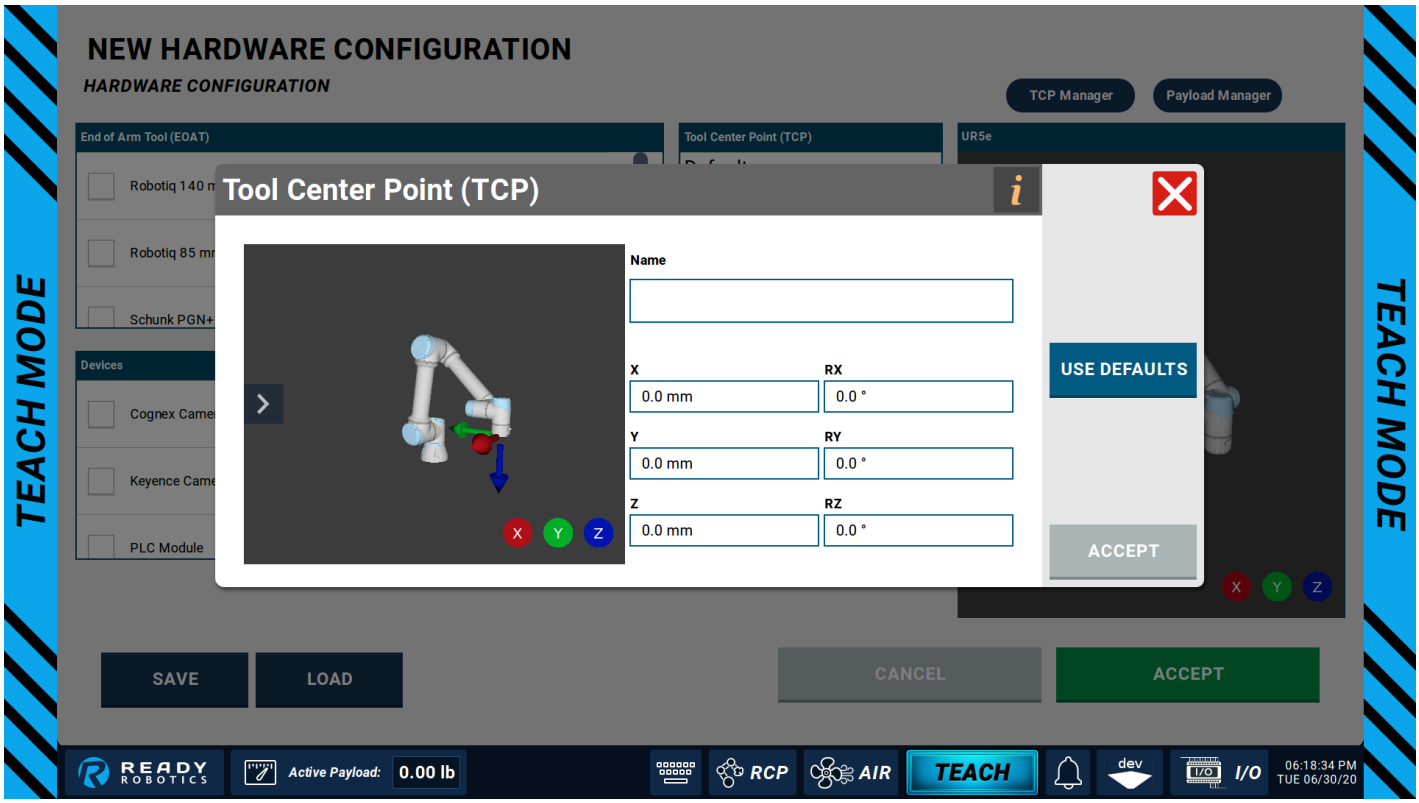
Defining a TCP

Selecting **New** or **Modify** will bring up the TCP generator. This window allows you to name the TCP and set the location and orientation of the point with respect to the default TCP at the tool flange. Name your TCP appropriately. For example, use "Schunk PGN80 - part 2 gripper" instead of "gripper."

You can enter offset values for each of the coordinate axes (**X**, **Y**, and **Z**) and the rotation values for their orientation (**RX**, **RY**, and **RZ**). Refer to the 3D rendering to visualize the location and orientation of your TCP before tapping **Accept** to create it.

With many of the pre-configured devices, you won't need to adjust the TCP orientation. But some custom end-effectors are not aligned with the Z axis. For example, you may need to use a suction cup that is mounted on a custom frame perpendicular to the robot tool flange. In this case you will need to apply an angular offset in RX

or RY. In the rendering below, if you wanted to define the positive Z axis where the positive Y axis currently is, you would rotate around the X axis by -90° (RX of -90.0°). Then positive Z would point to the left and Y would point upward.



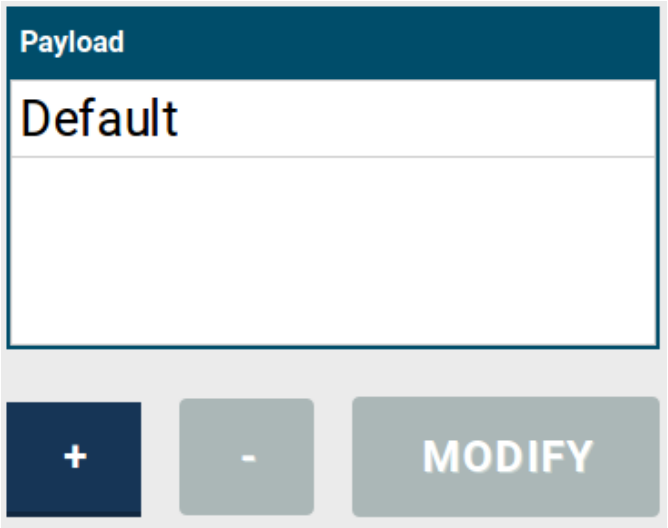
Note: You must account for the dimensions of any devices, such as force sensors or tool changers, between the robot flange and the desired TCP.

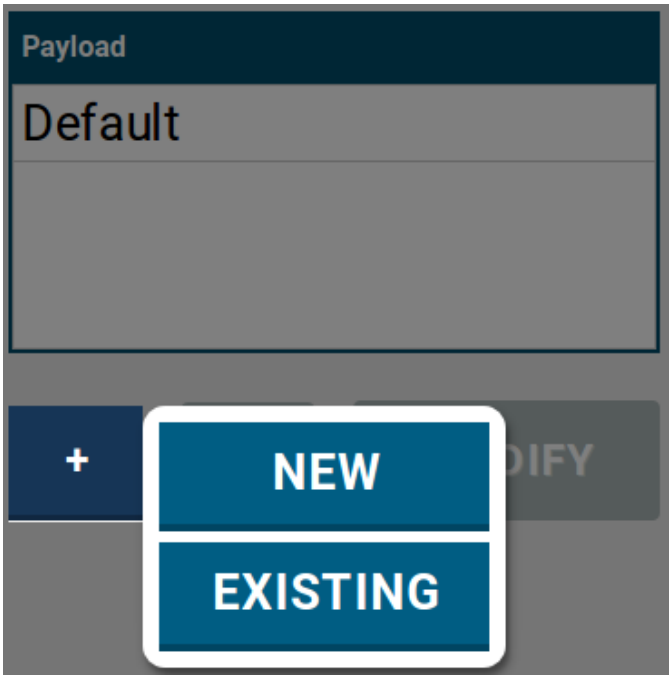
Payload

The Payload defines the mass and center of mass of all objects at the end of the robot arm. The factory default Payload is a mass of zero at the flange at the end of the robot arm. Any custom Payload you choose to define is added to the default Payload. You can select a Payload to view it in the 3D robot rendering. To load a Payload into the Hardware Configuration for use, tap on the **Plus** sign under the Payload selector, then tap **Existing** to choose from the existing options. To define a Payload and add it to the current configuration, tap on the **Plus** sign, then tap **New**.

While programming a Task, you will need to add all the TCPs and Payloads you expect the program to use in the Hardware Configuration so that they are accessible to the program. Plan accordingly and define them ahead of time when possible.

Note: It us important to supply accurate payloads as the robot arm's safety sensors may depend on the expected amount of force at the end of arm. Configured payloads that greatly differ from the actual force will cause frequent stoppages during execution



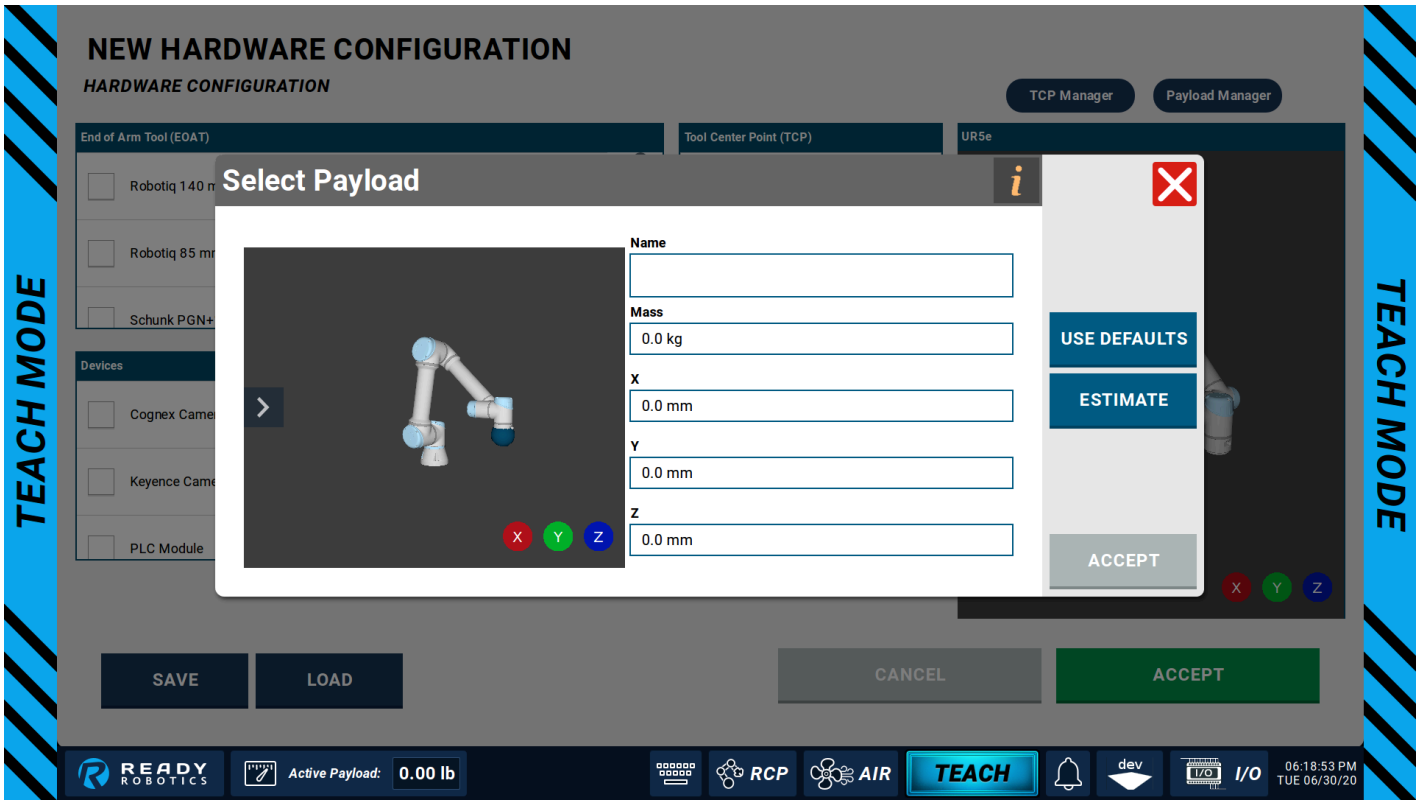


Element	Description
Add (Blue Plus)	Provides the options to tap on **NEW** or **EXISTING** to create a new Payload or add an existing Payload to the configuration.
Remove (Blue Plus)	Removes the selected Payload. Removing a Payload from the configuration will not delete it from your system.
Modify	Change the values of the selected Payload. You cannot modify the default Payload.

Defining a Payload

Selecting **New** or **Modify** will bring up the Payload generator, which allows you to name the Payload, set the mass, and set the center of mass with respect to the tool flange. Name your Payload precisely to distinguish them clearly when using them later. For example, you may have a payload for just the EOAT, for the EOAT plus pre-worked part, and for the EOAT plus finished part. You can enter offset values for each of the coordinate axes (**X**, **Y**, and **Z**). Refer to the 3D visual rendering to visualize the location the Payload center of

mass before tapping **Accept** to create it.



Note: You must account for the mass and dimensions of any devices, such as force sensors or tool changers, between the robot flange and the desired Payload.

Setting the Default Configuration

It is possible to set a default configuration that Forge/OS will load on boot. When a default configuration is set, Forge/OS will skip the Hardware Configuration application and boot directly into the Home screen.

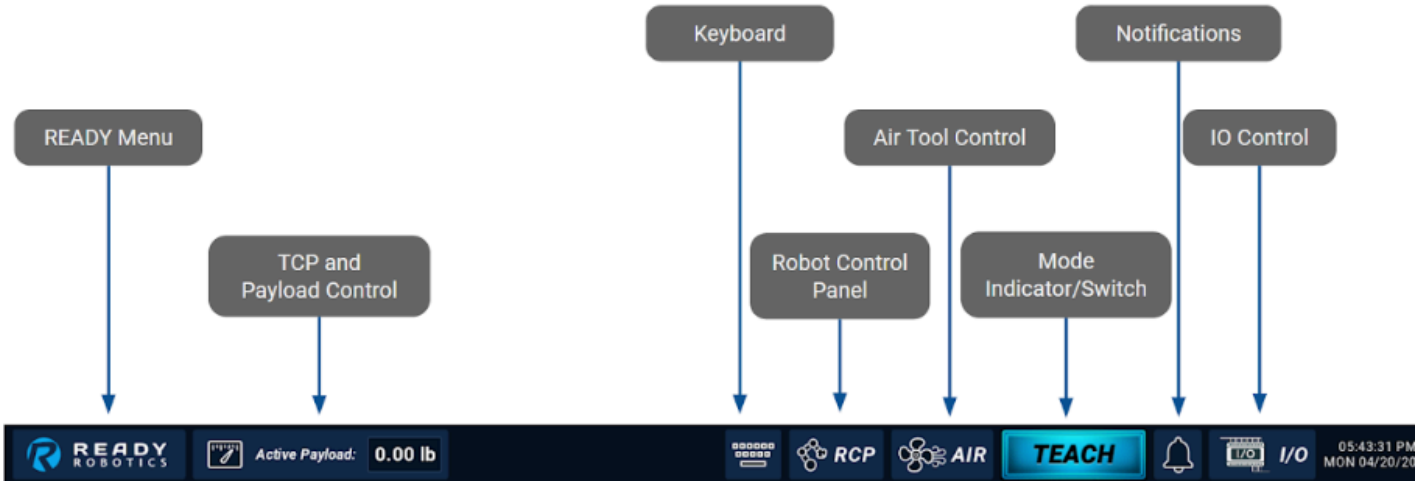
To set a default, Save a configuration by tapping **SAVE** after creating the configuration. Tap **LOAD** to open the Hardware Configuration Manager and select the saved configuration. Tap **MAKE DEFAULT** to set that configuration to load automatically when Forge/OS boots.

To clear a default configuration and set Forge/OS to boot into the Hardware Configuration application with no added tools or devices, select the default configuration and the **MAKE DEFAULT** button will change to **CLEAR**. Tap **CLEAR** and the configuration is no longer the default.

Note: If your OEM controller requires a restart or any other interaction on the OEM interface when Forge/OS configures the TCP and Payload settings, you will still need to perform these actions after Forge/OS applies the default configuration on boot.

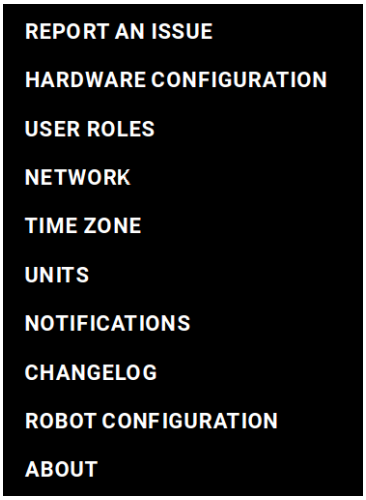
Control Suite

The Control Suite is positioned along the bottom of the Forge/OS screen and is accessible at all times except during Task execution. It gives access to system settings and manual controls for the robot and other connected devices.



READY Menu

At the leftmost end of the Control Suite is the READY Menu, which is expanded by tapping the READY Robotics logo. The READY Menu gives access to settings for Forge/OS that will apply to all applications.



Element	Description
Report an Issue	Send information about a technical issue to READY Robotics. This feature saves certain parameters and Task information, including a screenshot at the time of reporting. The information captured helps the READY support team identify specific problems when troubleshooting. If the problem is visible on the screen, READY recommends reporting the issue from the screen where the issue.
Hardware Configuration	Open the Hardware Configuration application.
User Roles	Switch between the Administrator and Operator roles; see the "User roles" section for more information.
Network	Configure the system's network connection; see the "Network Configuration" section for more information.
Time Zone	Set the local time zone for Forge/OS.
Units	Select the system of measurement for Forge/OS.
Notifications	Manage the email addresses for system notifications. Email addresses entered in this screen are accessible in certain TaskCanvas actions and events.

Element	Description
Changelog	Displays the changes in the current Forge/OS version from the previous version. If you've recently updated your system, this screen displays information about new features and bug fixes.
Robot Configuration	(Forge/Ctrl only) Restart the Initial Configuration process for your Forge/Ctrl system.
About	Displays the software version and robot driver of Forge/OS.

User Roles

User Roles enables the user to switch between the available system permission roles. The Administrator role has the highest level of access and enables full control of system settings, Hardware Configuration, Task creation and modification, and Task execution. To switch between user roles, select the role and enter the admin PIN. If you lost or forgot the Admin PIN, contact READY support for a temporary reset PIN.

The following grid shows the permission comparison between the Operator and Administrator roles:

Permission	Operator	Administrator
Execute Task	✔	✔
Access the manual robot controls on the Control Suite	✔	✔
Open saved configurations in Hardware Configuration	✔	✔
Report an issue	✔	✔
Create and modify configurations in Hardware Configuration	✖	✔
Create a new Task	✖	✔
Delete an existing Task	✖	✔
View autosaved Tasks in the Load Task screen	✖	✔

Permission	Operator	Administrator
Modify an existing Task – including updating Task information, adding, editing, deleting blocks, creating/updating Waypoints	✗	✓
Change network settings or time zone	✗	✓

Network

As detailed in the Network Configuration section earlier in the manual, you can change network connection settings from the READY Menu. Remember, Forge/OS must be connected to the internet for cloud and email services to work. An internet connection also enables remote system updates and customer support. If you have issues connecting to your network, first consult your facility's networking/IT manager. Then reach out to READY support if still you cannot resolve the issue.

Units

Forge/OS can display all values in Metric or Imperial units. From the READY menu, tap **UNITS** to open the Unit System Configuration window. Select the unit system and press **ACCEPT**.

Forge will use the following units in each system:

Measurement	Imperial	Metric
Distance	inch	millimeter
Speed	% robot max speed	% robot max speed
Mass	Pounds	Kilograms
Force	Pounds	Newtons
Torque	Foot-Pounds	Newton-Meters
Angle	Degrees	Degrees

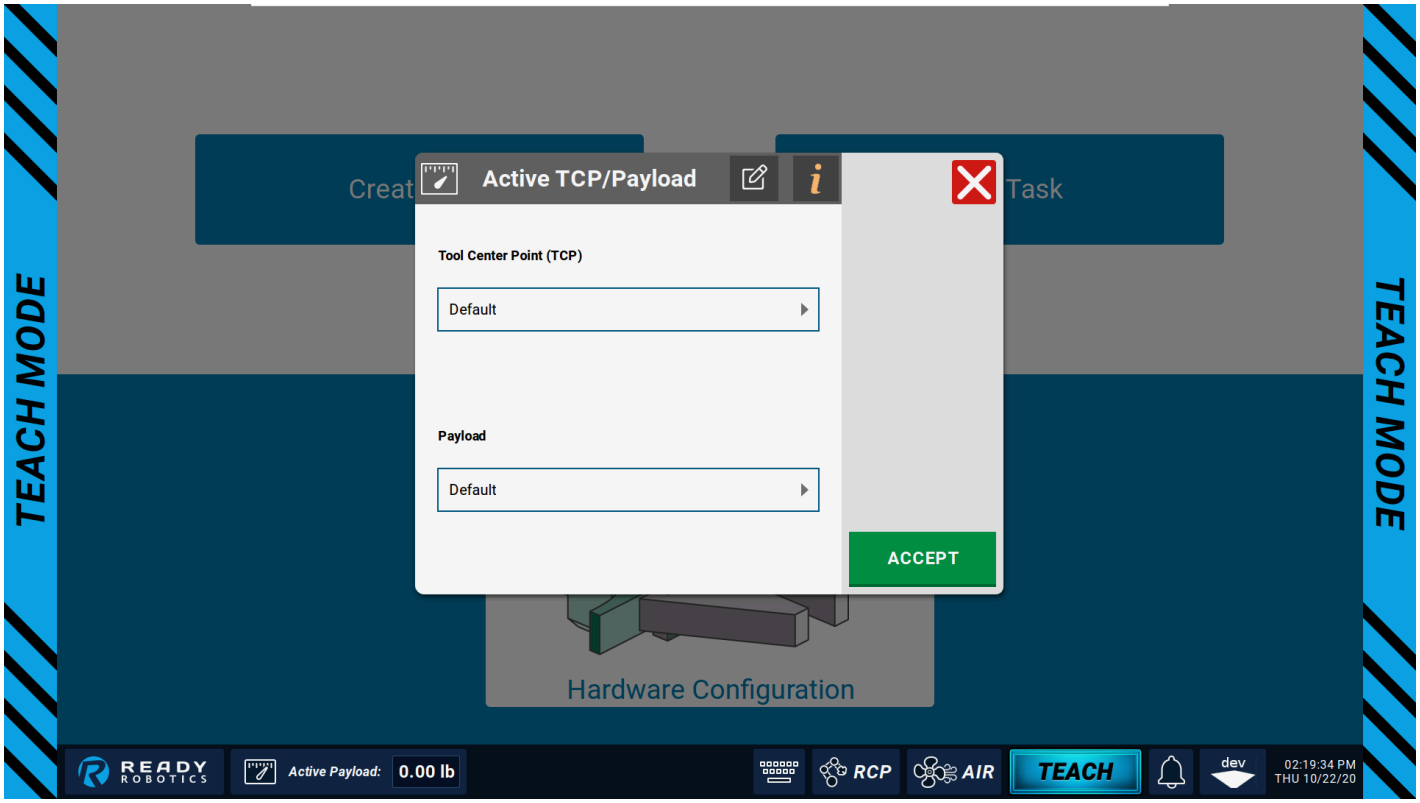
Note: Forge/OS will change the units and convert associated measurements in number fields where the units are provided by Forge/OS. For example, Forge/OS will update a relative Cartesian jump distance of 1 inch to 25.4 millimeters when the user changes the system units.

Forge/OS will not be able to convert variable values that represent real-world measurements, even if used in fields with units. For example, Forge/OS will not convert the value of a variable that is referenced in the distance field of the relative Cartesian jump, as Forge/OS cannot verify that the variable value is intended as a measurement of distance only.

Active TCP/Payload

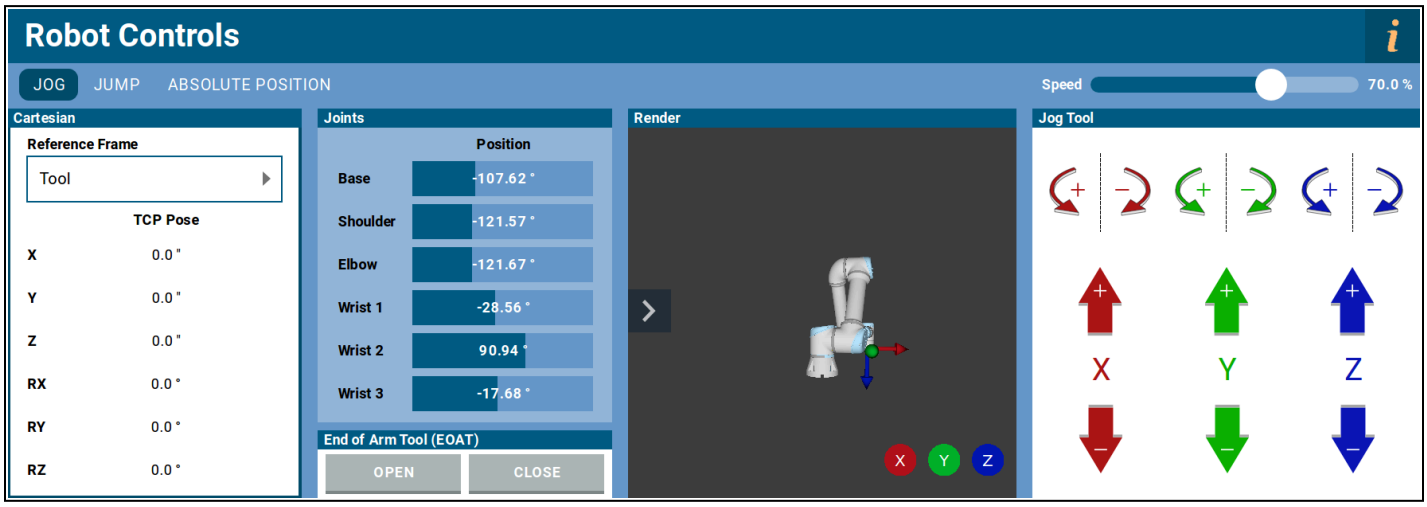
The Active TCP/Payload button on the Control Suite gives access to the robot arm TCP and Payload. Tap the button to open the Active TCP/Payload screen and select the current TCP and Payload. The available options for both parameters are those selected in Hardware Configuration as part of the current configuration.

This feature is important when you have multiple TCPs configured and you want to move the robot manually with respect to each of them while building a task.



Robot Controls Panel (RCP)

The Robot Controls Panel (RCP) is accessed by tapping on the **RCP** button on the Control Suite. It is the interface in Forge/OS for controlling the robot arm manually. The panel contains movement controls, end of arm tool controls, a 3D visual rendering of the robot arm, and the universal speed control.



There are three types of motion the RCP can instruct the robot arm to do: *jog*, *jump*, and *absolute position*. There are three tabs on the RCP, each corresponding to a type of motion for the robot arm and each including options for Cartesian and Joint motion. The 3D rendering, speed slider, and frame selection are present on all three tabs.

Tab (type of motion)	Description
Jog	Jog contains controls for jogging in both Cartesian and joint space. Tap **Cartesian** to access the individual Cartesian controls. The Cartesian pane displays the location of the TCP relative to the robot's wrist flange and has the option to change the Reference Frame. Tap **Joints** to access individual joint controls. In the Joints pane, the dark blue lines represent the joint positions relative to their physical limits which are represented by the edges of the blue position boxes.
Jump	Jump contains controls for jumping in both Cartesian and joint space. Tap **Cartesian** to access the individual Cartesian controls. Jump distance, direction, and Reference Frame can be specified in the Cartesian pane. Tap the **Joints** pane to access the individual joint controls. Enter the Jump By value to instruct individual joints to move by the Jump By amount, and tap **EXECUTE JUMP** to move the robot arm.

Tab (type of motion)	Description
Absolute Position	Absolute Jump moves the robot arm to absolute poses defined by their coordinate position from the Frame origin or by the joint positions. In the Cartesian TOOLS pane, the values represent the TCP position from the Frame origin. Tap the **Joints** pane to change the execute individual joint controls. Enter the Jump To value to instruct individual joints to move to the specified positions, and tap **EXECUTE JUMP** to move the robot arm.

Reference Frame Selection

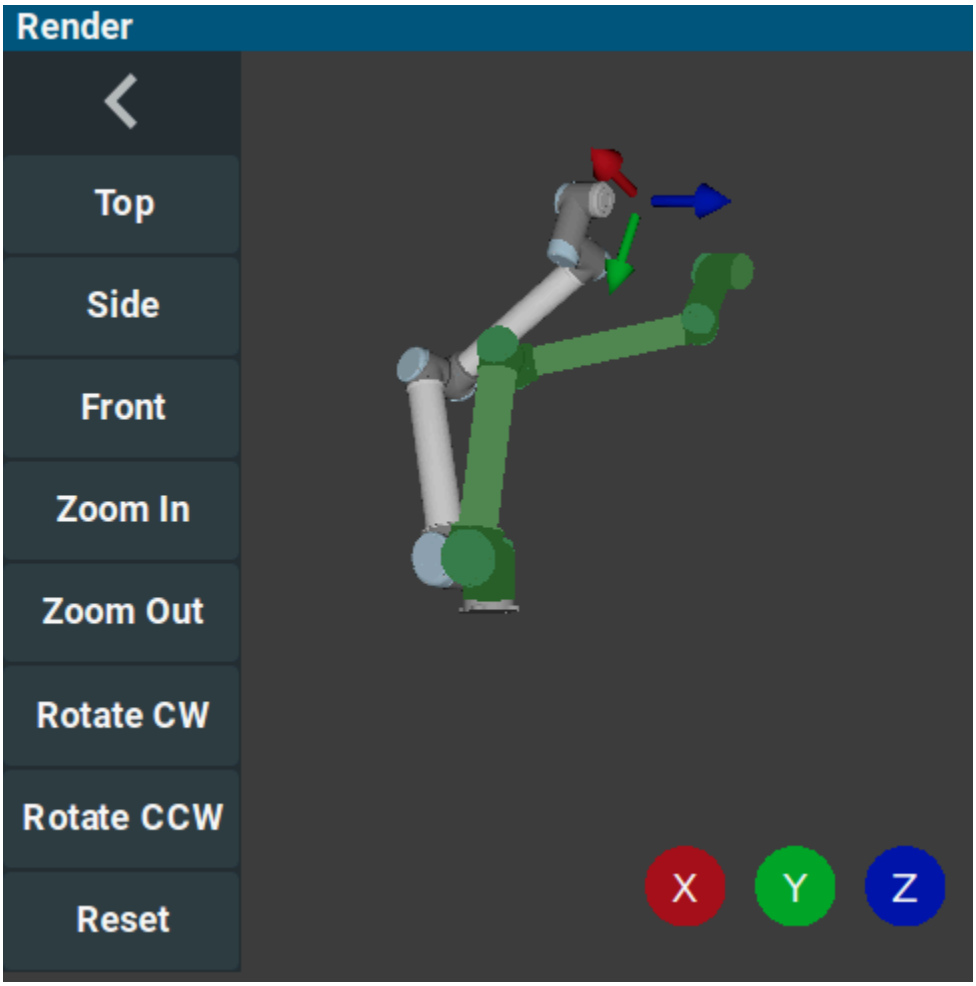
A reference frame is the set of Cartesian coordinate axes used when moving the robot arm. All motions, whether jogs, jumps, or absolute position jumps, are made with respect to the reference frame applied at the time of execution. Forge/OS comes pre-loaded with two frames, *Base* and *Tool*. User reference frames can be saved using waypoints or landmarks. The tool frame is defined by the default TCP or applied hardware configuration, so moving in this frame moves with respect to the tool's current orientation. The base frame reference is located at the base of the robot, so moving in the base frame moves the robot with respect to the unchanging base frame coordinates.

You can choose the reference frame you want to move in from the drop down selector in the Cartesian window in each of the RCP tabs. The 3D visual rendering will show the selected reference frame so that you can move the robot predictably.

3D Rendering

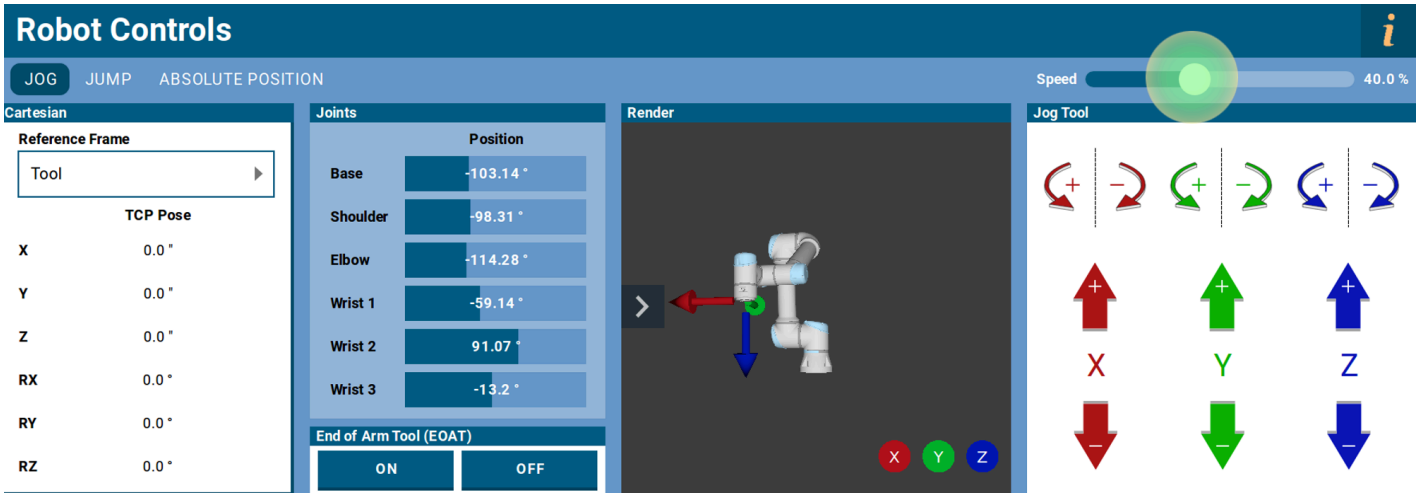
The Render box in the RCP displays the current position of the robot in a real time three-dimensional rendering. When in the jump or absolute motion contexts, the rendering displays the expected result of a jump move if values are entered in the **Move By** or **Move To** columns. The selected reference frame is also shown in the rendering as colored coordinate axes.

The rendering can be rotated by swiping left, right, up, or down across it. You can also open the View options by tapping on the arrow in the Render window to enable you to zoom in and out, rotate the visualization, and reset the view.



Speed Slider

The Speed Slider in the top-right corner of the RCP scales the speed of *all* movements the robot arm makes. When using the RCP to move the robot, this scale sets the maximum speed at which the TCP can move. Additionally, when a task program is running, this speed control scales the maximum system speed and therefore scales all programmed moves by the set percentage. So if a task includes a move block with a speed parameter of 70% and the RCP speed slider is also set to 50%, then the move will execute at 70% of the maximum system speed of 50%.



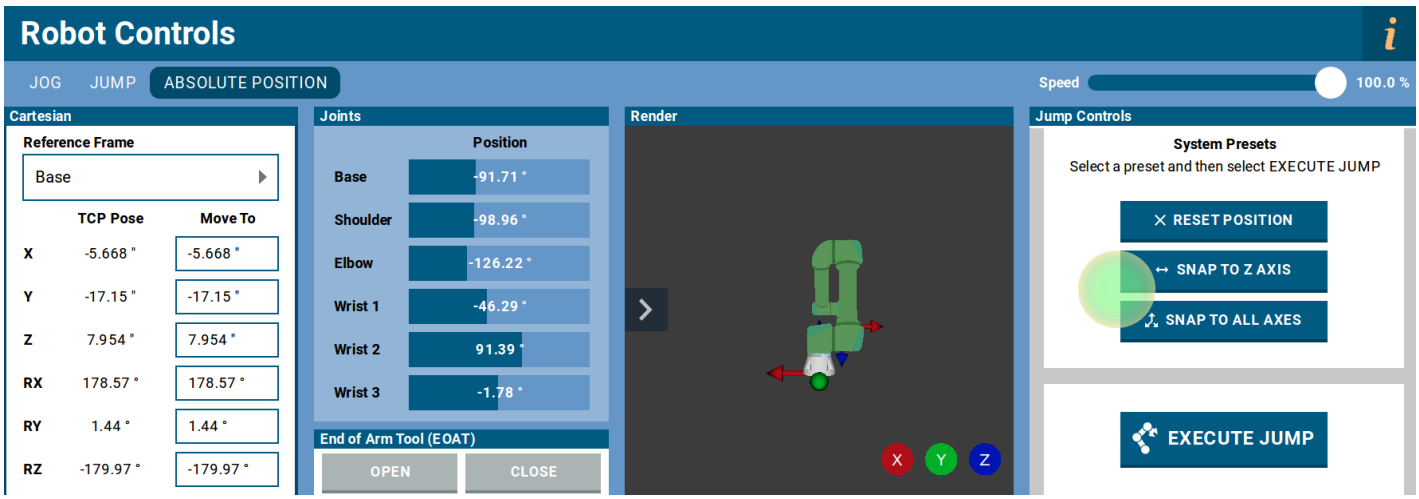
When executing a production task, set the speed to 100% to maximize efficiency. When building or debugging a task, it is good practice to set the speed control to a lower setting to prevent collisions for new waypoints or between incompatible waypoints.

Snap To Axes

Axis snapping is another important feature in the Robot Controls Panel. It allows you to align the end effector's orientation to the orientation of a selected reference frame such as *Base*, which is done to straighten the tool and help visualize where the robot will move when you give it the command. This feature is accessible in either the Jump or Absolute Position tabs on the Jump Controls window.

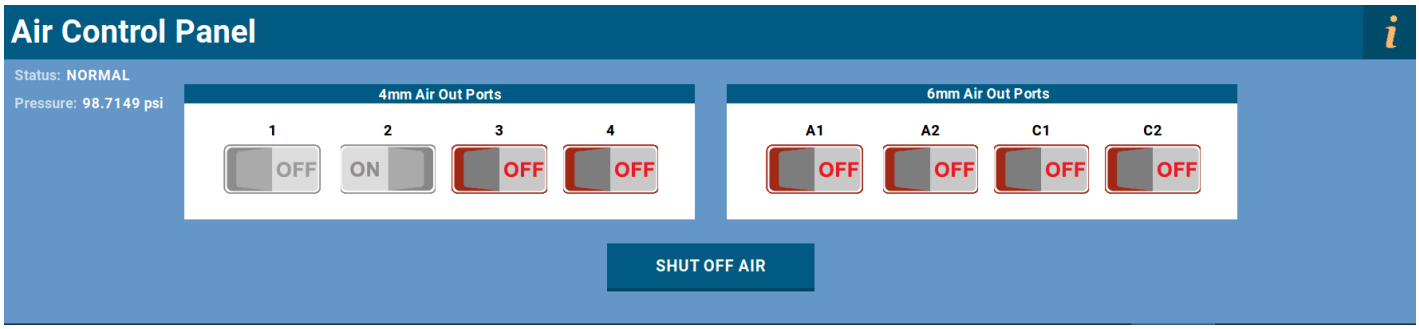
SNAP TO Z-AXIS will only align the tool's Z-axis to the reference frame's nearest axis. **SNAP TO ALL AXES** will align all of the axes. Note that the snapping functions are only accessible when a reference frame other than Tool is selected because there is no need to align the tool with itself.

Note: The Snap To buttons will align the axes of the configured tool with the selected reference frame's closest axes, so Snap To Z-Axis won't necessarily align the tool's Z-axis with the Z-axis of the reference frame. If you want to align the Z-axis of the tool to that of the base, for example, you should first move the robot into a joint position that is "nearly aligned" and then use the snap features. If you want to align the Z-axis of the tool with the X-axis of the base, you should first move the robot until the tool Z-axis is "close to parallel" with the X-axis and then use the snap features.



Air Controls Panel

The Air Controls Panel is accessed by tapping the **Air** button on the Control Suite. It displays the status of compressed air and the available pneumatic ports on your Forge/Station or Forge/Ctrl. You can toggle available air ports manually by tapping the toggle switches ON/OFF.



If you have pneumatic tools or devices configured to onboard air ports in Hardware Configuration, those ports will appear grayed out in the Air Control Panel.

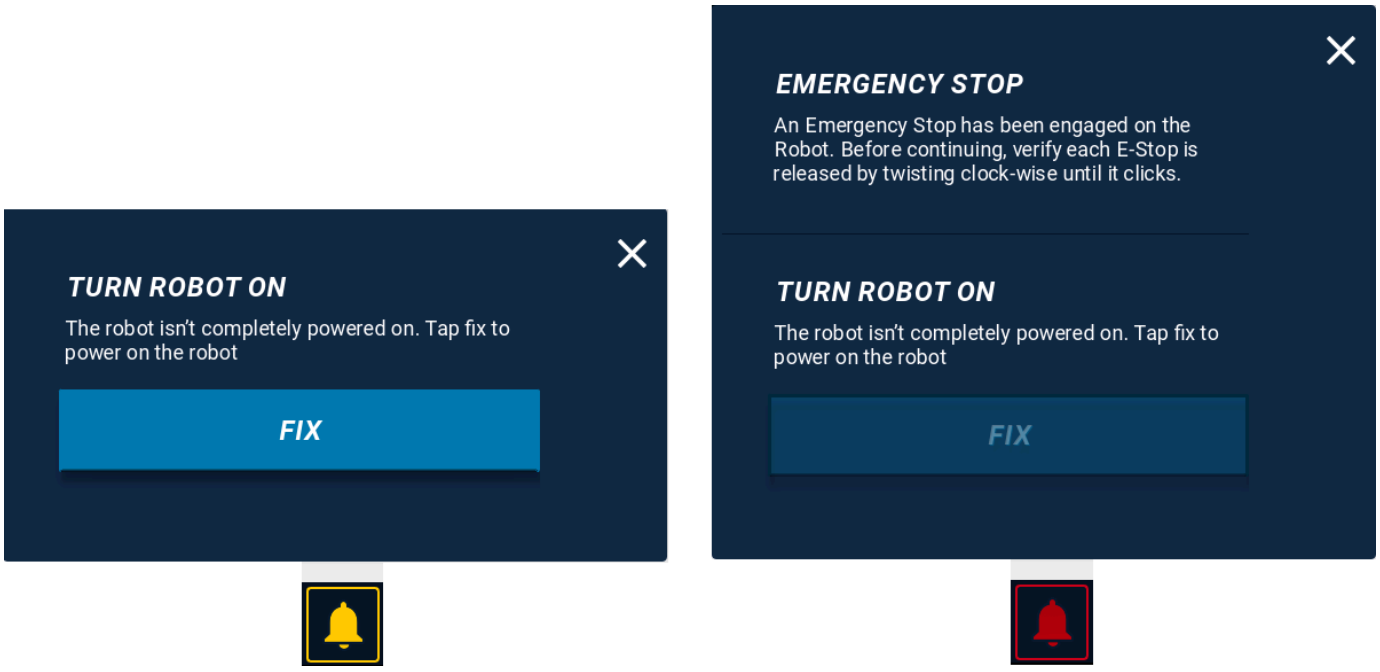
Compressed air attached to the Forge system will trigger the following states based on psi:

- High: greater than 110psi
- Normal: 85 to 110 psi
- Low: 60 - 85 psi
- Unusable: less than 60psi

Note: The pneumatic solenoids inside the Forge/Station and Forge/Ctrl are rated for a maximum of 125psi. READY recommends that you use a regulator to ensure that air pressure into the system does not exceed this value. The quality of air running through the Forge Systems should be dry air with a 5µm particle filter.

Notification Center

The Notification Center displays information about Forge/OS status, errors, and recovery. Tap the Bell icon to open the notification pop-up. If your system is in a Robot Warning state, you can power it on from the Notification Center. If your system is receiving error messages from the OEM controller, they will be displayed in the Notification Center.



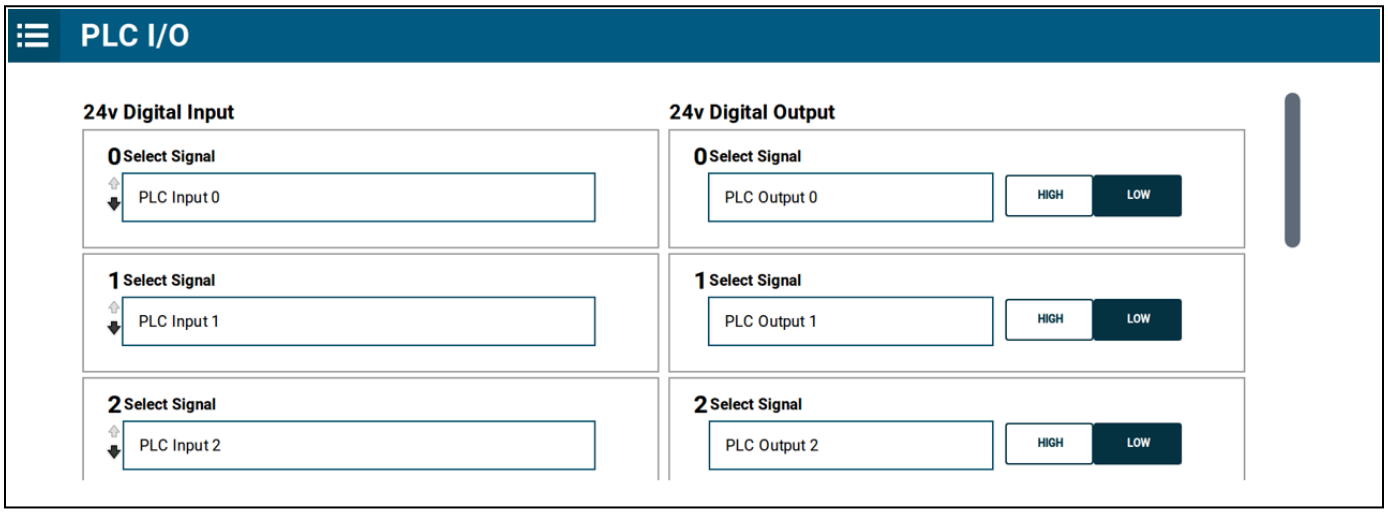
*The Notification Center displaying the **FIX** button when in a Robot Warning state, and displaying the current system status when an Emergency Stop button is pressed.*

I/O Panel

The I/O Panel displays the status and controls for attached digital input and output devices, such as the

READY PLC breakout box or the robot OEM I/O device. Tap the dropdown menu in the upper left corner to display the available devices. Some devices will only appear if they are configured as part of the Hardware Configuration.

Note: The I/O panel button is only seen and accessible on the Control Suite when PLC I/O is added in the Hardware Configuration

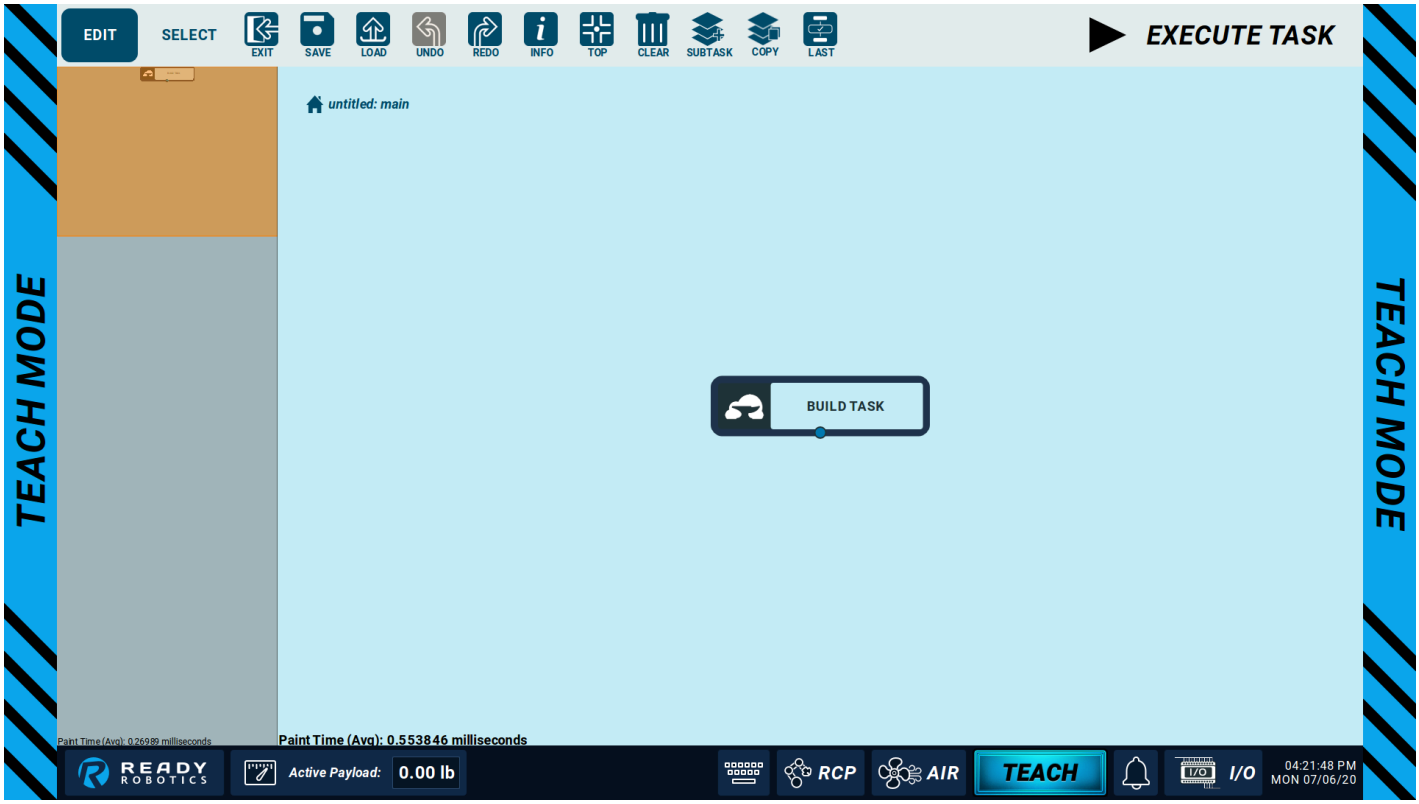


The I/O Panel displaying signals and state for the READY PLC breakout box.

Tap an input or output name field to change the name of that signal. Signal names are changed at the operating system level, meaning they will persist through different Tasks and configurations. Tap the **HIGH/ LOW** buttons to set the output signal state. Input signals display their state with the arrows to the left of the signal name.

Task Canvas

Task Canvas is the Forge/OS application that enables you to create, edit, and execute Tasks in your automation cell. You can program every primary function on the Task Canvas with "blocks", or individual commands that control everything from robot motion to digital communication and variables.



Canvas Menu

The Canvas Menu shows the primary controls for modifications to the Task and the Canvas. There are two Canvas Menus, accessible by toggling the **EDIT** and **SELECT** buttons on the left side of the menu. At the right side of the menu is the **EXECUTE TASK** Menu, covered in the Execute Task section below.

Canvas Menu: Edit

Element	Description
Exit	Leaves the current Task and returns the user to the Home screen. If the Task has unsaved changes, Task Canvas will prompt the user to save them.
Save	Saves the current Task. ***Note:*** Tasks are automatically saved every time the user makes a change to the Canvas. The autosaved Task is available in case the system unexpectedly shuts down. See the **Current Tasks** section for more information.
Load	Opens the Current Tasks screen. If the Task has unsaved changes, Task Canvas will prompt the user to save them.
Undo	Steps back the previous change made to the Canvas. Edits that can be undone include creating and deleting blocks, moving and copying blocks, and connecting and disconnecting blocks.
Redo	Steps forward the previously undone change.
Info	Displays information about the current Task such as job and part data, Waypoint, Landmark, and Variable Managers, and email notifications for Task status.
Top	Centers the Canvas view on the Root block, the first block in the Task.
Clear	Removes all blocks and parameters, such as Waypoints and Variables, and starts a blank Canvas. Be sure to save the Task before pressing CLEAR.
Subtask	Creates a new subtask Canvas. There are four buttons on the Canvas Menu for modifying Subtasks through additional Canvases. Adding Subtasks to your Task enables you to execute multiple simultaneous functions. See the Multitasking section for more information.
Copy	Enables the user to copy the selected Subtask. The canvas and all of its blocks will be copied to a new Subtask. You can copy the Main Task as well as any Subtask.
Note	Adds a Note to the selected block. You can add text to a Note and it will appear off to the side of a block. Select a block with no Note and tap **ADD NOTE** to create a new Note. Select a block with an existing Note and tap **EDIT NOTE** or **DELETE NOTE** to modify or remove the note.

Element	Description
Last	Selects and centers the view on the block that was last executing on the current Canvas when the Task stopped. The Last executed block will keep a magenta highlight when not selected.

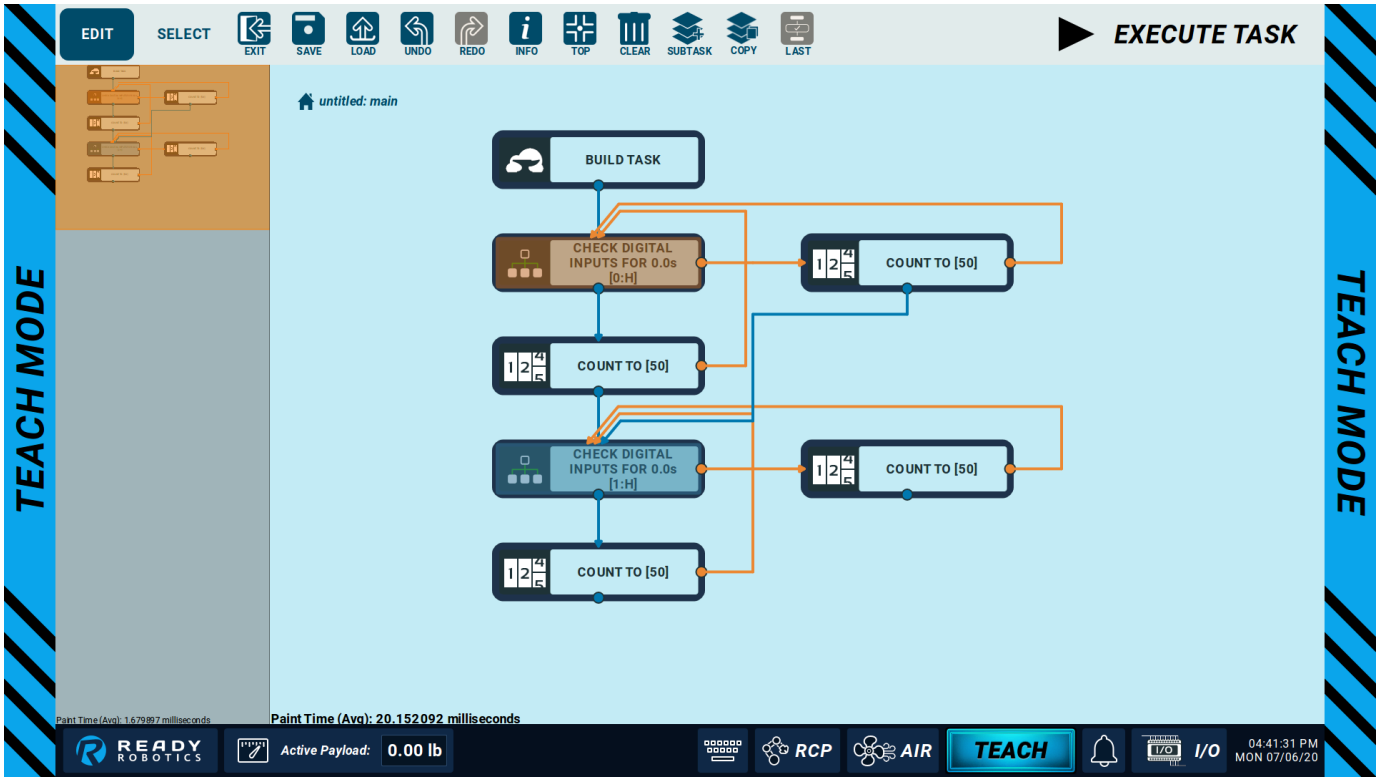
Canvas Menu: Select

Tap the **SELECT** button to change the Canvas to the Select context. Under the Select context, you can tap more than one block at a time to multi-select them. Tap a selected block again to deselect it.

Element	Description
Undo	Steps back the previous change made to the Canvas. Edits that can be undone include creating and deleting blocks, moving and copying blocks, and connecting and disconnecting blocks.
Redo	Steps forward the previously undone change.
Top	Centers the Canvas view on the Root block, the first block in the Task.
Last	Selects and centers the view on the block that was last executing on the current Canvas when the Task stopped.
Group	Creates a Container around the selected block. See the Containers section below for more information. Selected blocks must be consecutive to group them into a Container.
Clear	Removes the selected blocks.
Detached	Selects only the blocks that are not connected to the Root block through a parent pathway. Detached blocks are denoted by their transparent state.

Skipping blocks allows the Task to pass through a block without executing it. Use this feature when debugging the Task or performing a dry run to skip blocks that you only want to execute when the Task is in production. Task Canvas denotes skipped blocks with a blue or orange overlay depending on the path the Task will follow.

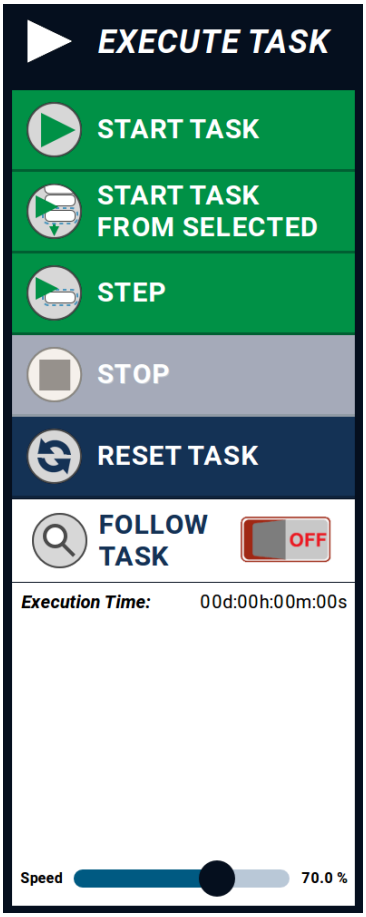
Element	Description
Bottom Skip	Skips the selected block and continues along the bottom (blue) path of the skipped block.
Right Skip	Skips the selected block and continues along the right (orange) path of the skipped block.
Restore	Removes the skipped status of a block so that Task Canvas will execute it. ***Note:*** Before executing a Task with skipped blocks, confirm the updated path of the plan will not cause the robot arm to collide or damage equipment by performing two actions that are not safe to perform consecutively.



Two skipped blocks, overlaid blue and orange. The orange block will not execute and continue along the orange path; the blue block will not execute and continue along the blue path.

Execute Task Menu

The Execute Task Menu enables you to execute the blocks in the current Task. If you have a TeachMate, when the system is executing a Task or individual block, the TeachMate LEDs will blink green and blocks on the Canvas cannot be edited. On Forge/OS systems with the READY pendant, Forge/OS must be in **RUN** mode to start a Task. On Forge/OS systems without a READY pendant, Forge/OS must be in **TEACH** mode to start a Task.

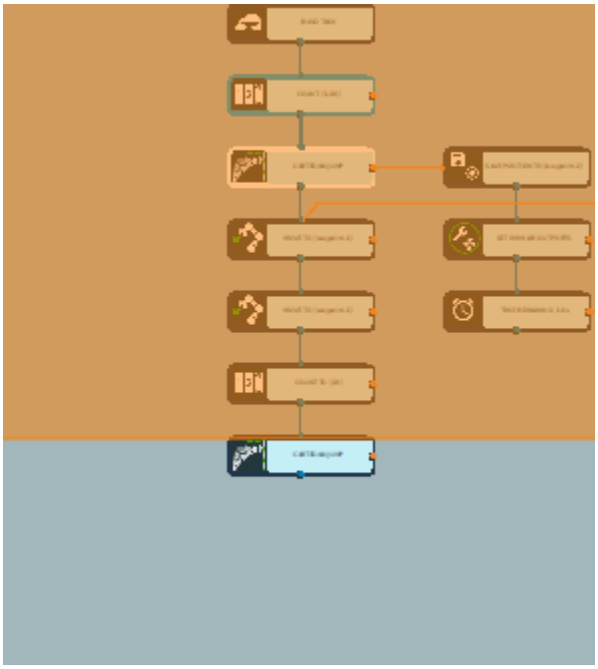


Element	Description
Start Task	Executes the Task from the Root block.
Start Task from Selected	Executes the Task from a selected block other than the Root block. The Task will run through all blocks as programmed.
Step	Executes only a single selected block. The Task will not execute any block following the selected block, but will highlight the following block on the blue or orange path based on the execution status of the select block. Press Step again to execute the next block.

Element	Description
Stop	Stops all Tasks and Subtasks immediately. Stop executes a controlled stop, which is the safest stop from the perspective of the robot arm. If the safety of a worker is at risk, use the Emergency Stop or other safety-rated systems to stop the task and robot arm.
Reset Task	Resets the iterative values or sequences of blocks such as Counter or Grid. Reset Task also resets the Task Execution Timer and text displayed within blocks on the canvas. ***Note:*** Start Task From Selected" and "Step" will not reset the values or sequences of the Task Execution Timer or blocks such as Counter, Timer, Grid, and Pattern. Start Task will cause all values to reset and start from the beginning. For example, if you stop a Task when a Grid Move is in its third position and a counter block is at three, tapping "Start Task From Selected" on any block will continue the Task with the Grid Move going to the fourth position and the counter counting to four. If you stop the Task and tap "Start Task", the Grid Move will start from the first position and the counter will start again at one, regardless of where they were stopped. Tap "Reset Task" to force these values to reset before using "Start Task From Selected" or "Step".
Follow Task	Centers the view on the executing block during Task execution. Toggle the switch OFF or interact with the Canvas to turn Follow Task off and freely navigate through the executing Task. Toggle the switch back to ON to center the view on the executing block.
Speed Slider	The Speed slider sets the maximum speed available to all robot arm motion. Any motion in Forge/OS will scale from its maximum value by the specified Speed percentage. The Speed slider does not change the speed setting within individual blocks but scales them from their set speed. For example, if the speed of a motion block is set to 50% and the Speed control is set to 75%, the motion will execute at 75% of 50% of the maximum speed of that motion type. When debugging a Task, it is good practice to set the Speed control to a lower setting to prevent collisions for new Waypoints or between incompatible Waypoints.

MiniMap

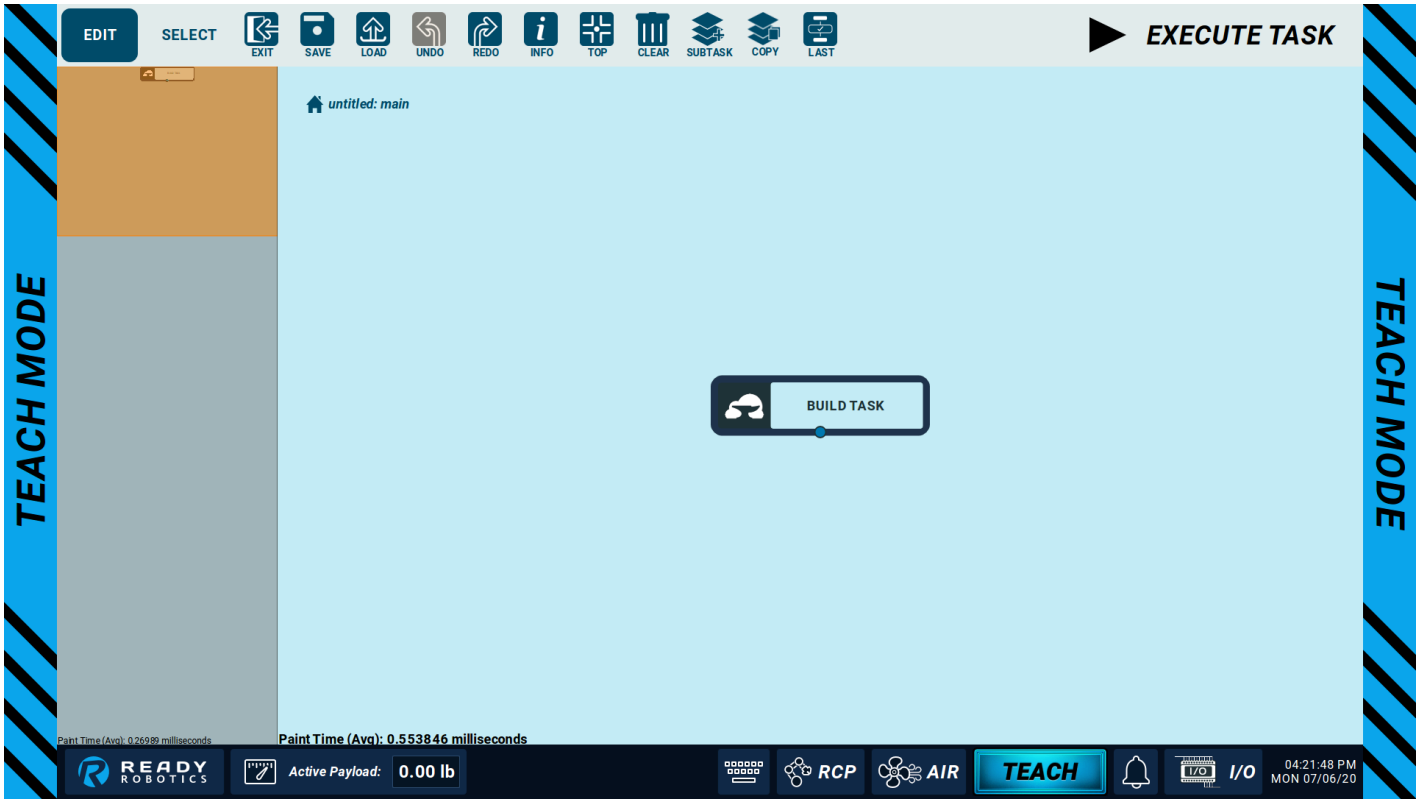
The MiniMap is located on the left side of the Canvas and displays a summary view of the flowchart and highlights the current view of the Task Canvas with an orange Frame. Dragging the orange Frame around the MiniMap changes the view of the flowchart in the Task Canvas. The MiniMap can be used while both programming and executing a Task. Blocks on the MiniMap will flash blue while the block is executing in a running Task.



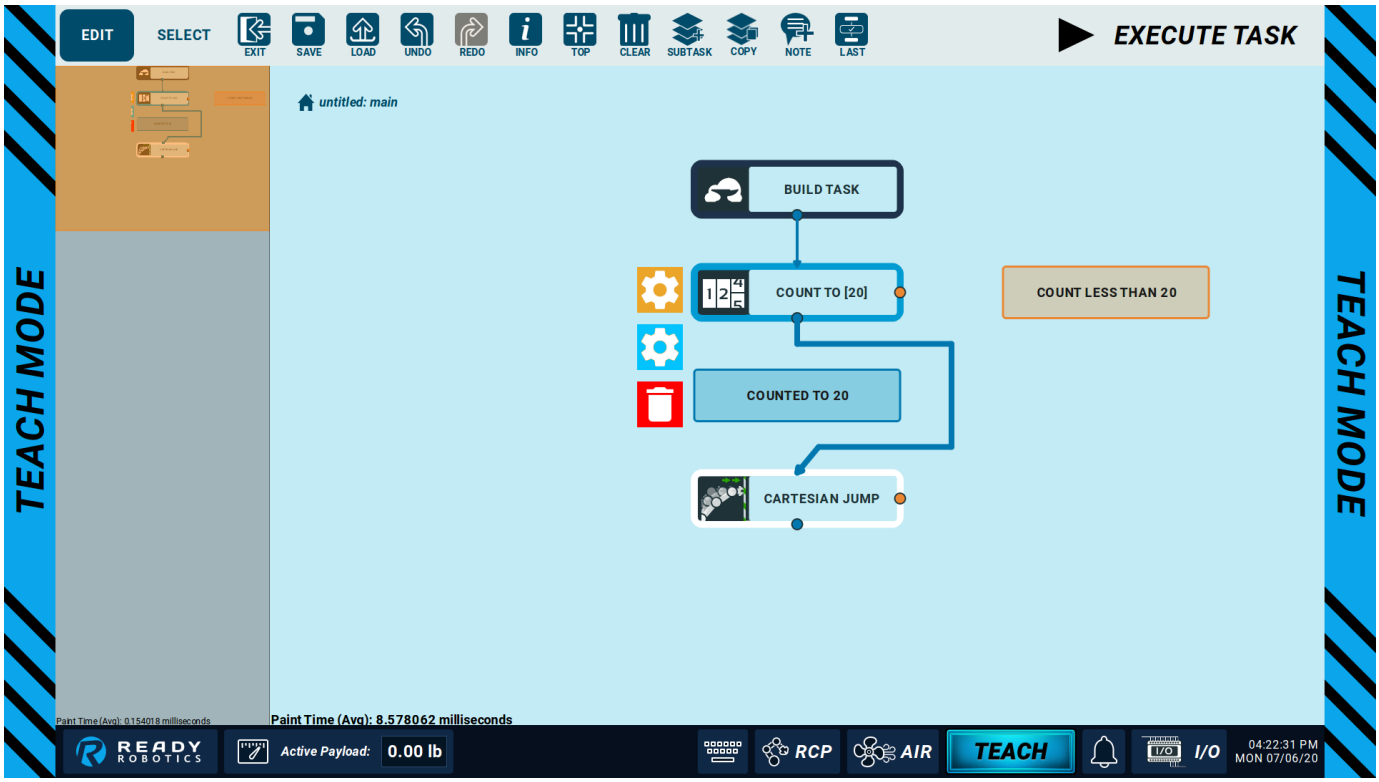
Canvas

The Canvas is where you program blocks and connect them to form the Task structure that you execute for production. There is no length limit to the flowchart and you can create multiple Canvases to execute functions simultaneously. For more information about simultaneous execution, see the Multitasking section below.

A flowchart starts at the Root Block, labeled **BUILD TASK** on the Canvas. As blocks execute, they follow the blue path or orange path to the next block and perform each function in succession. A flowchart ends when a block executes and there is nothing on its blue or orange paths.



Programming Blocks



A selected Count block shows the available sites, as well as the two settings gears.

You can add blocks to the canvas by placing them in sites. A site is a location after a block on which you can program the successive block. Some blocks have a single blue site and some blocks have a blue site and an orange site. Blocks with both a blue site and orange site will cause the flowchart to proceed down either path depending on the action or feedback for that block. The text within each site displays what action or feedback will cause the flowchart to follow that path.

Tap the **Root Block** to display its site. Tap the **ADD BLOCK** site to open the **Block Menu**. Select a block from the list and its generator will open, where you can program the settings for that particular block. Tap **ACCEPT** in the block generator and the block appears in the site. Tap any block on the Canvas to select it and display the available sites around it.

When you select a block, Task Canvas will highlight the block blue and display the options for the block.

Element	Description
Orange Settings Gear	When you select a block, Task Canvas will highlight the block blue and display the options for the block.

Element	Description
Blue Settings Gear	Opens the index, or count, of the block. Only blocks that increment as the Task executes will have a Blue Settings Gear. For all block types, Task Canvas will interpret the value you enter into the index as **the last fully executed index**. For example, if you want a Grid Move to go to the 4th position, enter 3 into the Index. Remember, you must press "Start Task From Selected" or "Step" to continue a Task at these values.
Red Trash Can	Deletes the block from the Canvas. Parameters in the block such as Waypoints and Variables will still be part of the Task and must be deleted from their Managers if you wish to remove them from the Task.

Custom Block Text

Blocks on the canvas display information about their function before, during, and after execution. You can modify this text with the Custom Block Text pop out in any block generator. Select the Custom Block Text button from the top bar to open the pop out.

The four text fields represent the states of execution on the canvas:

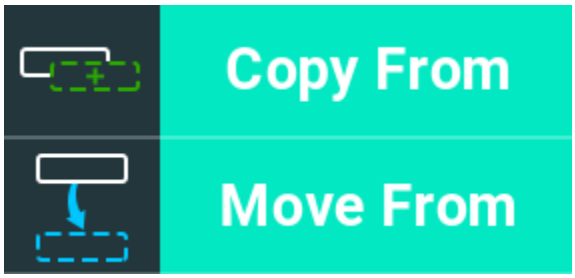
Setting	Description
Ready to execute	Before the block executes on the canvas, or anytime the canvas is reset.
Executing	While the block is executing on the canvas.
Finished Bottom	After the block finishes executing and follows the bottom outcome path.
Finished Right	After the block finishes executing and follows the right outcome path. If a block does not have a right path, this text will never appear.

Enter the text for the appropriate execution state in the field. If you leave any field blank, the default text for that state will appear on the block. Select **Use for all states** to use the “Ready to Execute” text as static text that will always appear in the block. Tap **Use Default** to clear all fields and use the default text for all states.

Copy and Move blocks

Select a site where you want to copy or move a block to. From the Robot Blocks menu, select **Copy From** or

Move From. Select the block you want to copy or move to that site. Both Copy and Move functions will preserve all of the settings within the block.



Programming Path

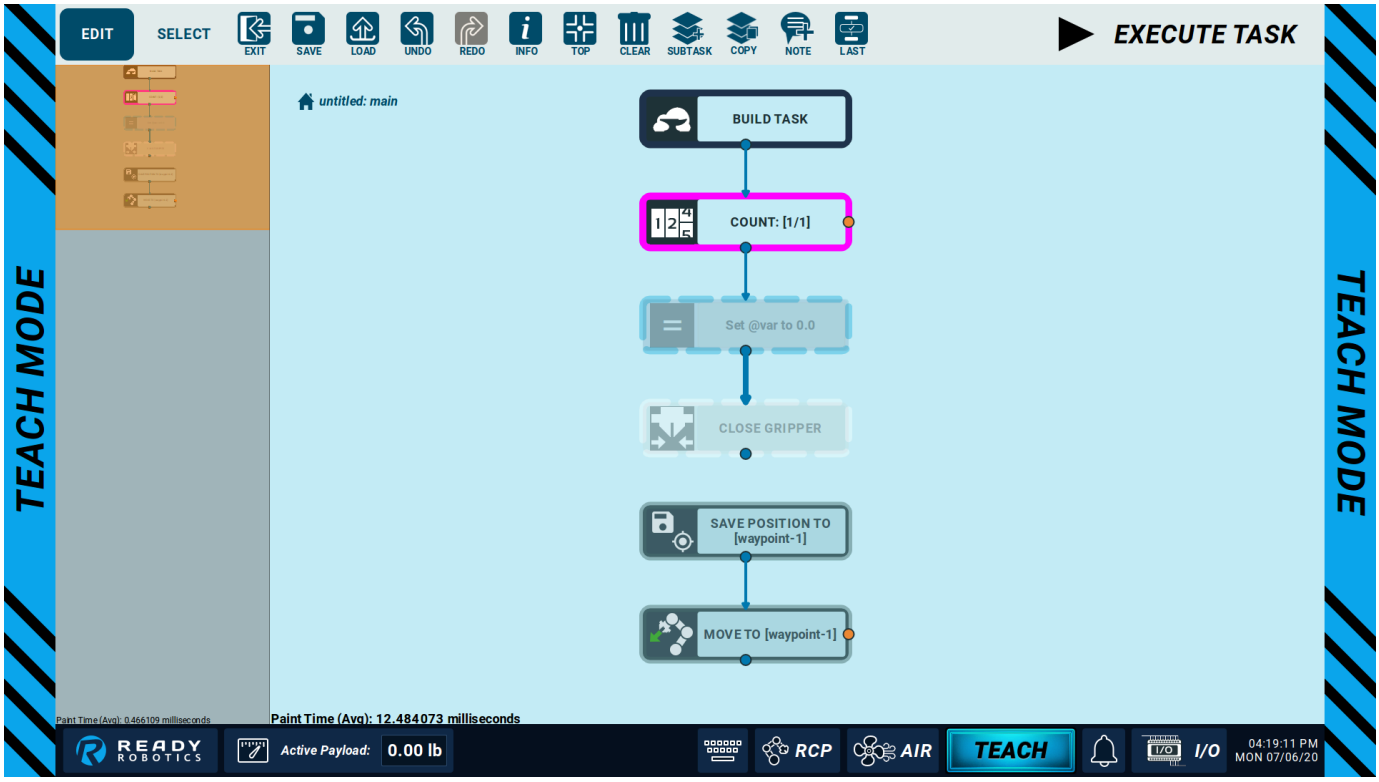
When you add a block to the site below an existing block, Task Canvas will automatically draw the path between the two blocks. You can also draw and erase paths connecting existing blocks on the Canvas from the Block Menu.

Tap the site where you want to add a path and select **CONNECT TO** from the Block Menu. Tap the block to which you want to attach that path and Task Canvas draws the path from the site to the block.

Tap a site with an existing path and select DISCONNECT from the Block Menu to erase the path.

Special Block States

There are several block states and highlight that a block may take on the Canvas.



A magenta block was the Last executed in the Task. The following two blocks are ghosted because the variable and gripper are not present. The last two blocks are detached because there is no path connecting them to the Root Block.

Element	Description
Last blocks	Last blocks are highlighted magenta and are the last blocks being executed when the Task was stopped. There can only be one Last block per Canvas and it will change when you execute and stop the task or it will lose its highlight when you press Reset Task.
Ghosted blocks	Ghosted blocks are slightly transparent and have a dotted border. Ghosted blocks are missing a necessary parameter in their settings or actuate a tool or device that is not part of the current configuration. Some ghosted blocks can be fixed by updating the parameters within them, such as a deleted variable, and some can only be fixed by removing them from the Canvas. You cannot execute a Task with ghosted blocks unless you skip them.

Element	Description
Detached blocks	Detached blocks are slightly transparent and have a solid border. Detached blocks do not have a path leading to them from the Root Block. You can Step them individually and connect them to the main path using Connect To. However they will never execute as part of the main Task unless reconnected.

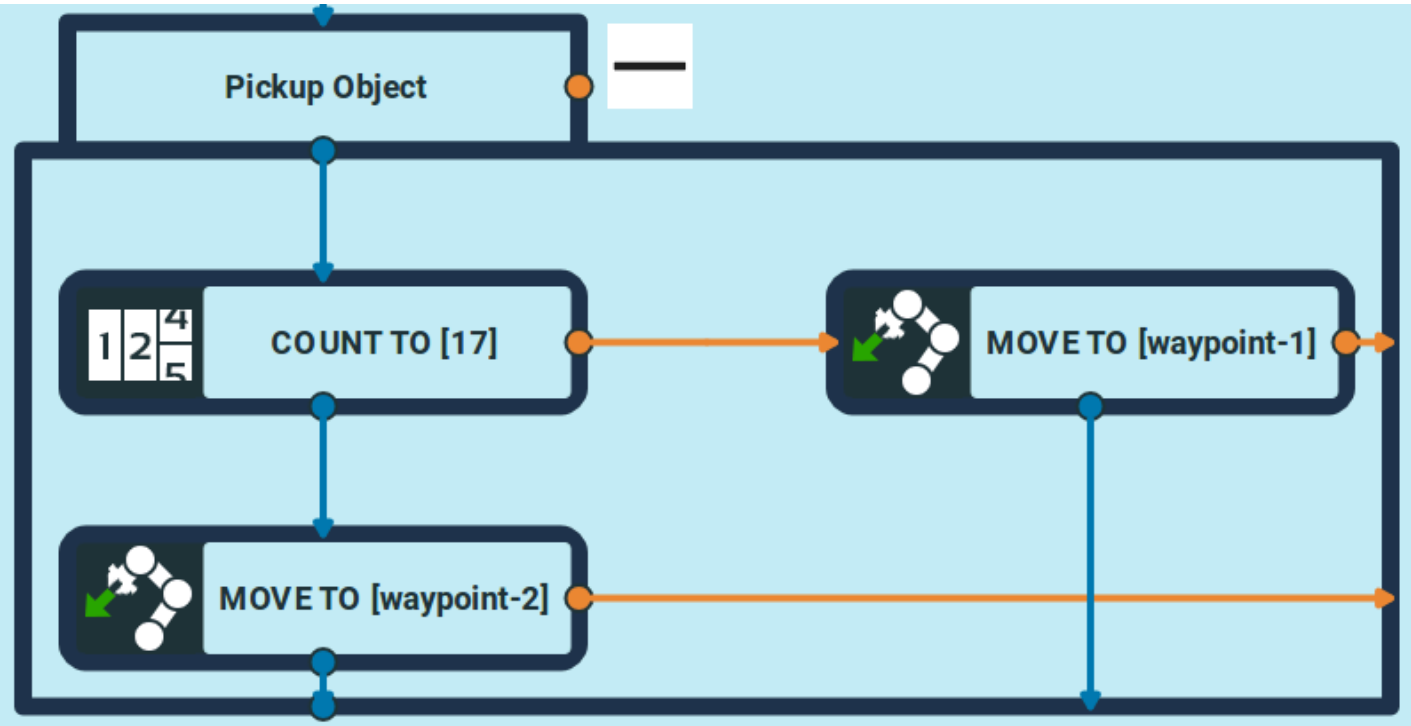
Containers

Task Canvas allows you to group blocks together into nameable, collapsible groupings called **containers**. This can greatly reduce the complexity seen in the Canvas and allow for quicker, less confusing navigation. It also allows you to reuse groups of blocks multiple times in the same task or in multiple tasks, making it so complicated functions only need to be created once.

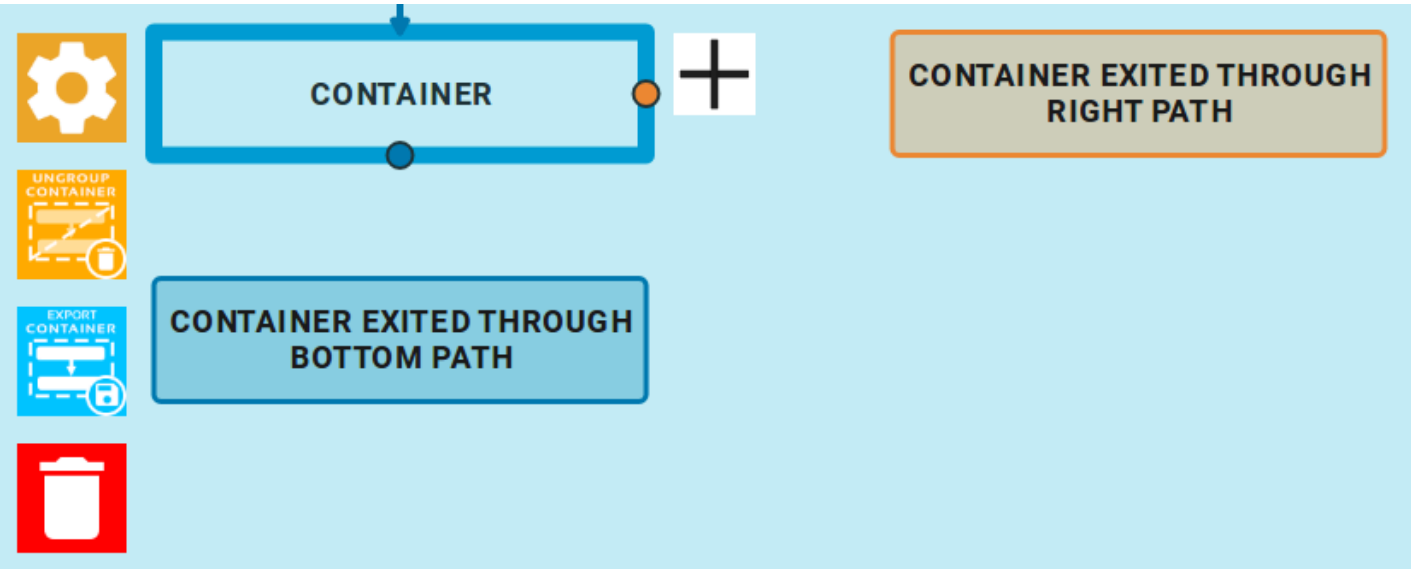
Container Behavior

A Container is treated like an individual block in that it has a single point of entry, single blue pathway, and single orange pathway. Blocks outside the container cannot connect to individual blocks inside a container. Blocks inside the container must exit through the container's pathway for the system to continue the task outside the container.

The pathways of the blocks inside the container must be connected to the container's bottom or right pathway for the system to continue the task. If a block's pathway is not connected to the container's pathway, the task ends after the block is executed.



When you select a container, it will highlight blue and display container options similar to block options.



Element	Description
Orange Settings Gear	Opens the Container settings window, which includes options to name the container, minimize it, reset internal indexing, manage containers, and import a container
Orange "Ungroup Container" Icon	Removes the grouping, effectively deleting the container while keeping the blocks in place. You will be prompted with a dialog box to confirm.
Blue "Export Container" Icon	Exports/saves the container for use elsewhere. You will be prompted with a dialog box to name the container you are saving.
Red Trash Can	Deletes the container and all blocks inside the container. You will be prompted with a dialog box to confirm.

In the settings window, there is an option to reset the current value of incremental blocks inside the container when the task exits and returns to it later. When creating a container or editing its properties, you can check the **Reset Internal Nodes** checkbox. Then, when the container is entered, the blocks inside will be reset. For instance, a Counter or Timer block inside will be reset back to zero, even if it has not reached its maximum value. Grid and Pattern blocks reset to the first position, even if they have not moved to every programmed position.

Creating Containers

There are two ways to create a container. The first way creates an empty container from the block menu like any other block. After the empty container is created, new blocks can be added in it or existing blocks can be copied or moved into it. The second method groups selected blocks, already on the Canvas, together into a container.

1. Create an empty container

You can generate a container like any other block from the block menu and add building blocks into it from there. To do this, tap on a site in the Canvas to open the block menu and tap **Container** in the **Controls** menu. Then set the properties for the empty container as needed for the task; see the Block Glossary for more information on Container settings. Expand the container by tapping on the + sign by its name and start adding blocks as you normally would.

2. Create a container from a group of blocks

You can also group together existing blocks into a container after building your task. To do this, you must select the blocks you want to group in select mode (see section on Canvas Menu), then tap **Group**. You can then edit the container properties and give it a descriptive name.

Deleting Containers

In the case that you need to delete a container, there are two options. You can delete a container along with the blocks contained in it or you can delete the container while preserving the blocks.

1. **To delete a container and its contents**, select the container on the Canvas by tapping its name, then tap the red trash icon.

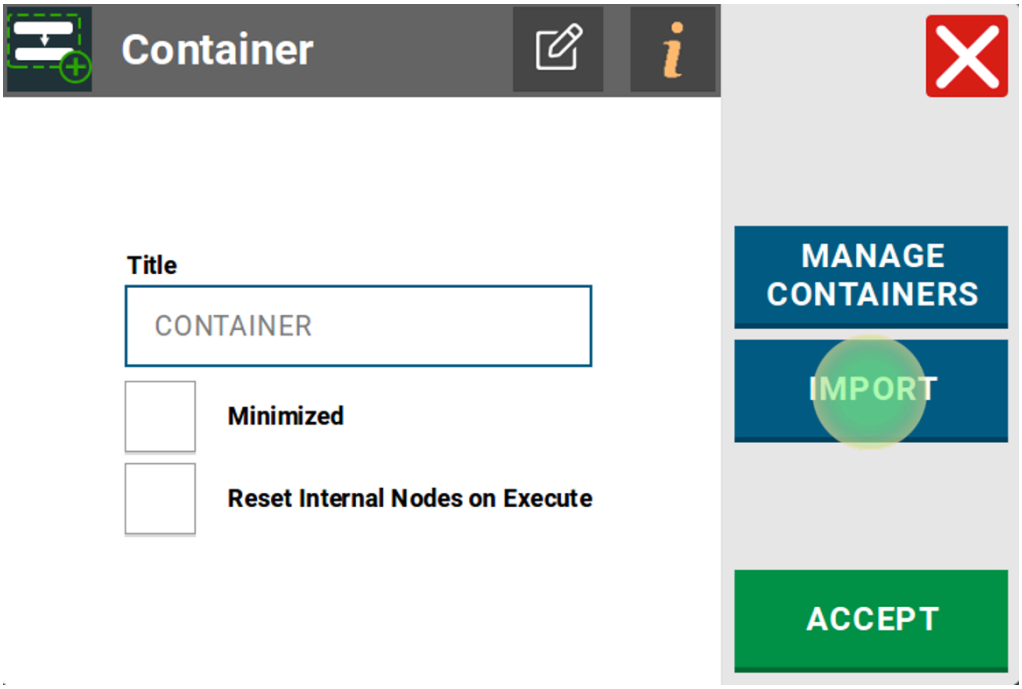
2. **To remove the container while keeping its contents**, you need to ungroup the container. Tap **Ungroup Container** to do this. Forge/OS will try to maintain the pathways where possible; however, the blocks may make different connections when they are ungrouped. Review the pathways between the blocks that were in the deleted container and the blocks that were connected to the deleted container. Adjust the connections as needed.

Reusing Containers

A container can be saved and loaded elsewhere in the same task or exported and imported into different tasks. When a container is saved, it maintains the settings of the blocks that are inside the container. This function makes it easy to copy complex sections of a task to other tasks or containers within the same task. Follow these steps to export and import a container for use in other tasks.

1. On the Canvas, select the container you want to save.
2. Tap the blue **Export Container** icon.
3. Enter the name for the container or use the default name.
4. Tap **Save**. Now. the container is saved outside of the current task and can be imported into other parts of the same task or into other tasks.

5 To import a container, add an empty container from the block menu and tap **IMPORT** in the Container settings window, then select the container that you would like to insert into your Task.



Note: Containers do not update automatically within or across tasks when you make changes to them. You must export a container to save any changes you have made and then import the container again where you want to use the updated container.

Multitasking

Tasks in Forge aren't limited to a single flowchart. Multitasking enables you to create more than one canvas and flowchart within a single Task. These "Subtasks" can run simultaneously to your Main Task, checking digital inputs, setting variable values, and running timers. Your Forge Task won't miss a single beat while the robot arm is working hard.

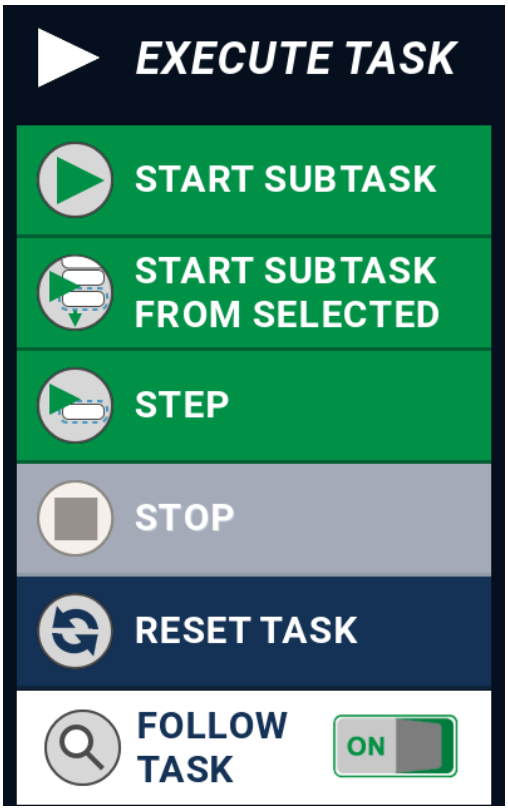
A Task with two Subtasks, displayed as tabs along the top of the Canvas.

Working with Subtasks

The most important thing to know about programming a Subtask is that they are no different from any other Task that you program on the canvas. Creating blocks and pathways, editing settings, and copying and moving blocks are all the same. A Subtask belongs to the Main Task in which it was created and will be saved in the file for the Main Task. Even parameters created in the Main Task or a Subtask will be available across those Tasks - meaning that if you create a variable in a Subtask, you can check its value in the Main Task and vice versa.

Subtasks can be started and stopped with "Start Subtask" and "Stop Subtask" blocks. The Main Task and Subtasks can start and stop other Subtasks, however the Main Task itself can only be controlled with the Start and Stop button from the Execute Task Menu. When the Main Task ends, all Subtasks end regardless of their current status.

Subtask Exectution



The Execute Task Menu will update when viewing a Subtask.

When programming and executing a Task with Subtasks, there are some changes to the canvas and execution dynamic that are important to note.

Element	Description
Subtask Execution	Subtasks will only start when a "Start Subtask" block on the Main Task or another Subtask instructs them to start.
Start Task (Main)	Start Task will start the Main Task from the Root block.
Start Subtask (Subtask)	Start Task will be replaced with "Start Subtask" on a Subtask view. It will start the visible Subtask from the Root block.
Start Task/Subtask From Selected (Main/Subtask)	Start Task From Selected will resume any blocks from Subtasks that were stopped when the Task was last stopped by the user. Forge will notify the user of what tasks are about to start. If you only wish to start the currently visible and selected Subtask, tap **Reset Task**, then **Start Subtask From Selected**.

Element	Description
Step (Main/Subtask)	Step will only execute the currently selected and visible block, regardless of what was last executing.
Reset Task	Reset Task will break the link between paused Tasks and Start Task From Selected, causing Start Task/Subtask From Selected to only start the currently selected and visible block.
Stop	Stop will stop ALL Tasks: Main and Subtasks.
Robot Arm Blocks	Robot Arm blocks can only be controlled by one running Task. If more than one Task has blocks controlling the robot arm, Forge will prevent the user from executing them simultaneously. Robot Arm blocks include all robot motion, Force Context, and Set TCP/Payload.
User Action Blocks	User Action blocks in any Task, Main or Subtask, will stop all Tasks unless the user instructs a Subtask to ignore them.
Finish	When a Finish block executes on any Task, the Main Task will also stop, stopping all Subtasks as well. A Finish block is the only way a Subtask can stop the Main Task.

Creating Subtasks

To create a Subtask in your Forge Task, press the **Add Subtask** button on the Canvas Menu. Forge will prompt you to name the Subtask. This name will display anywhere the Subtask is referenced on the canvas or in blocks that start or stop it.

When you create a new Subtask, it will appear as a tab across the top of the Canvas. You can switch between the Main Task and Subtasks at any time by selecting a tab.

When viewing a Task, changes that you make on its canvas will only occur on the visible canvas. When saving a Task, the Main Task and all Subtasks will save under the same file.

When a Task contains a Subtask, the following options will become available in the Canvas Menu.

Element	Description
Rename Subtask	Renables the user to rename the selected Subtask. Change the name and tap **ACCEPT** to save it. The name will update anywhere the Subtask is referenced, including start and stop blocks.
Copy Subtask	Enables the user to copy the selected Subtask. The canvas and all of its blocks will be copied to a new Subtask. You can copy the Main Task as well as any Subtask.
Delete Subtask	Enables the user to delete the selected Subtask. Forge will delete the Canvas and all blocks on it. Data created on this Subtask, such as Waypoints or variables, will not be deleted from the Task. Blocks that reference the Subtask, such as start and stop blocks, will enter a ghosted state until deleted or modified.

Programmatically starting and stopping tasks during Task execution

Subtasks can be programmed to start during a Task using a **Start Subtask** block. You can stop Subtasks during a Task using a **Stop Subtask** block. See the Block Glossary for more details.

Waypoints, Landmarks, and TCPs

Waypoints, Landmarks, and Tool Center Points (TCPs) are three positional parameters that define how the robot arm moves to and through its workspace.

The simplest definitions of each item are:

Term	Description
Waypoint	Location in space at which the robot arm interacts with its environment.
Landmark	Frame of reference from which the robot arm calculates a Waypoint's location.
Tool Center Point (TCP)	Location at which the robot arm assumes the working point of the attached tool is.

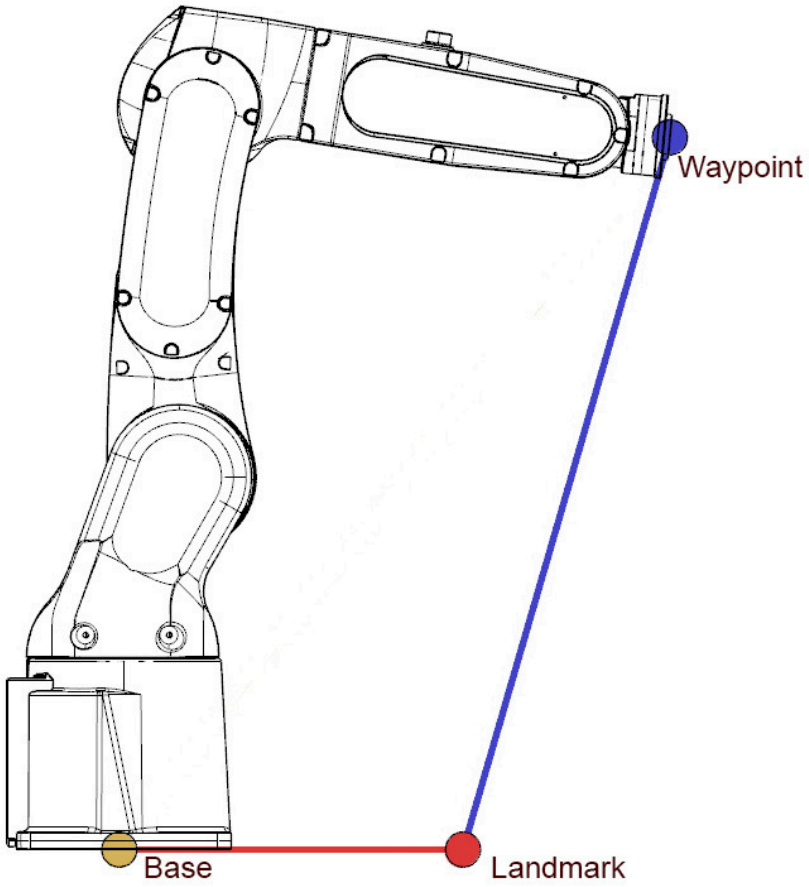
The hierarchy of how these parameters define each other looks like this:

1. A Landmark is defined from the base Frame of the robot
2. A Waypoint is defined from the Landmark acting as its origin
3. The robot arm's offset from the Waypoint is defined by the tool center point

Landmarks

Landmarks are the Frame of reference for Waypoints. The default Landmark from which all custom Landmarks are defined is Base and its position and orientation are the same as the manufacturer's origin position for your robot arm. To find the exact location of this point, consult your robot OEM's manual. You can see the position and orientation of the Base Landmark at any time by selecting "Base" from the Frame panel in the Robot Control Panel, which will update the robot rendering with the Base axes.

Custom Landmarks are best used to define the location of large objects in the robot's workspace at which more than one Waypoint will work. For example, if the robot is working within a mill and will have a number of Waypoints for interacting with the door, the vice, and the part, a recommended custom landmark would be called "Mill" and be defined at a repeatable interaction point between the robot arm and the vice.

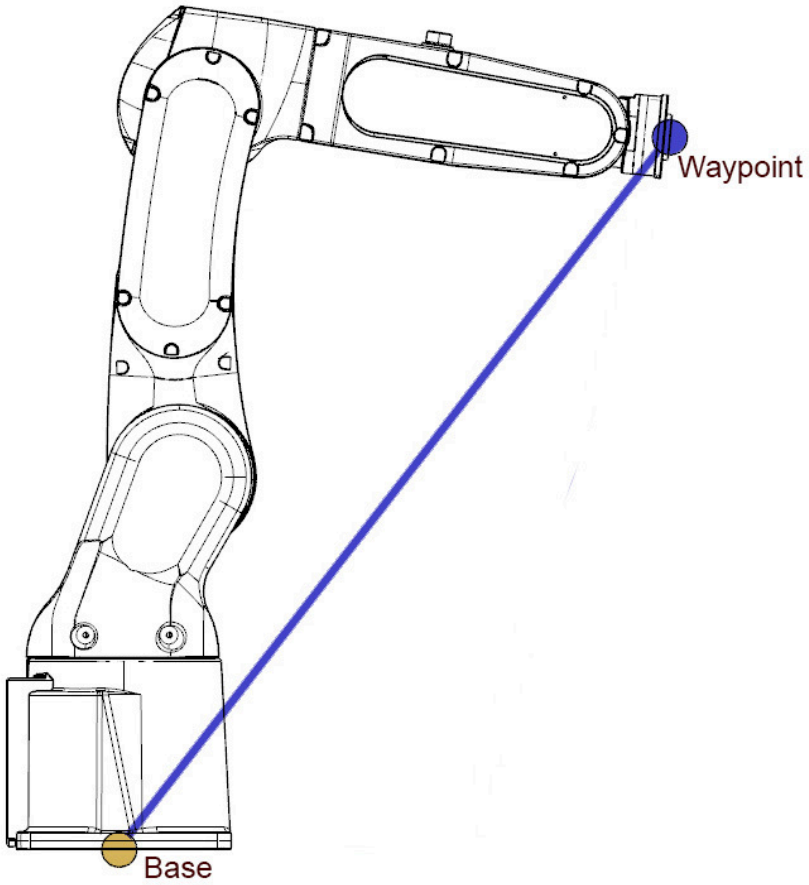


In Forge, Landmarks belong to the Task in which they were created. You can also promote a Landmark to a "shared/" Landmark which will enable you to reference it in multiple Tasks. When you change a shared/ Landmark, its pose and any Waypoints that reference it will update in any Task that uses it, even Tasks you aren't currently viewing.

Throughout this manual, the words "Landmark" and "Frame" may be used interchangeably, however they are always referencing the Landmark parameter in Forge.

Waypoints

Waypoints are the locations where the robot arm interacts with the workspace. A Waypoint will always have a Landmark, which is its origin position. By default, Waypoints will use Base as their origin.

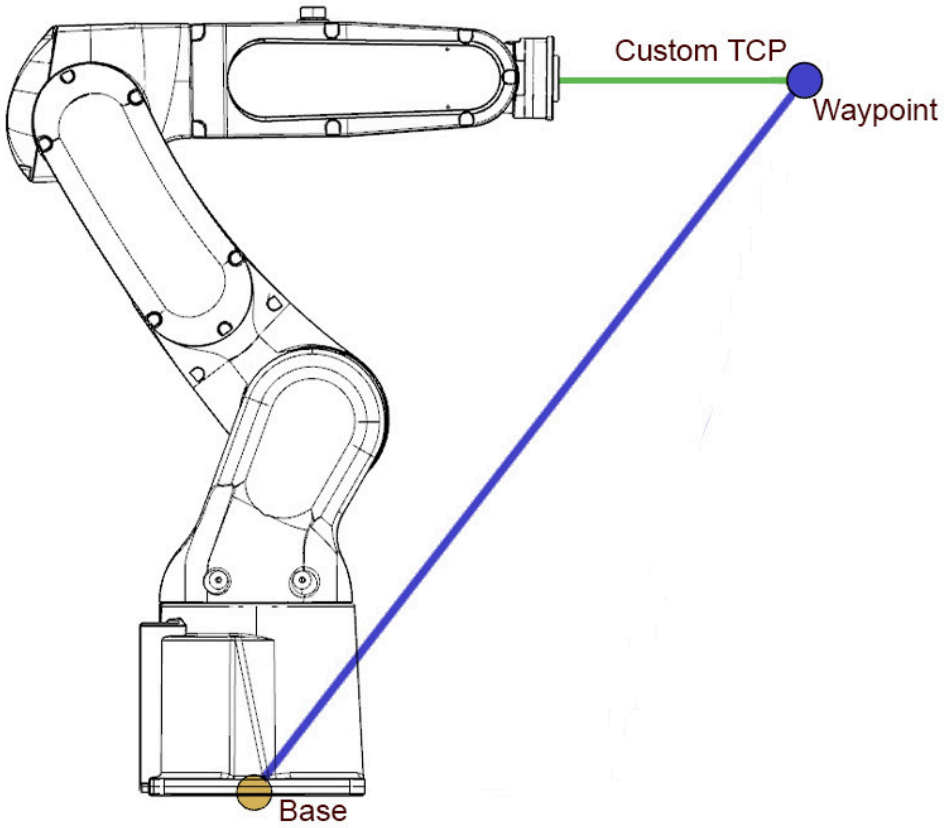


Throughout Forge, Waypoints are the most common parameter with which you will program the robot arm to move. Blocks on the canvas directing the arm to move to machine tools, iterate through a grid of parts, or curve around an obstacle will always reference Waypoints to define these motions and locations.

In Forge, Waypoints belong to the Task in which they were created. You can reference a Waypoint as many times as you want in any type of block that uses them. When you change a Waypoint, it will change everywhere that it is referenced. You can always copy or rename Waypoints to create new locations from Waypoints that you already have.

Tool Center Point (TCP)

TCPs are defined by an offset and orientation relative to the robot's tool flange which represents the working point of an attached tool. Because the working point of the tool is defined as a "Waypoint" in Forge, the TCP offsets the robot arm from a Waypoint. The "Default" TCP is at the robot arm's tool flange. When an arm makes a linear motion, the TCP is the position that the arm will move in a straight line through the shortest path possible.



Forge will always have an "active" TCP, which is the current offset that the arm assumes you are using to define Waypoints. When the active TCP changes, the location of the Waypoint will not change, but the offset of the arm from the Waypoint will update based on the difference between the TCPs. If the active TCP is "Default", then associated Waypoints and robot arm's end flange will be at the same location. Comparing the above picture to the corresponding picture for waypoint, you can see how the waypoint is in the same location in both images, however the arm pose has changed to accommodate a large TCP value in +Z.

You can set the active TCP through either the "Active Payload" on the Control Suite or by executing a "Set TCP/Payload" block on the Canvas.

In Forge, TCPs belong to configurations. To create or modify TCPs, use the Hardware Configuration application. The amount of TCPs that you can apply to a configuration will depend on the manufacturer of your robot arm. Once applied to the active configuration, a TCP can be referenced in any Task and by any Waypoint.

Forge/OS EtherNet/IP Adapter Interface

The Forge/OS EtherNet/IP Adapter Interface enables Forge/OS to communicate with a PLC device where the PLC controls and monitors the connection. Unlike the Device Manager in Hardware Configuration where Forge/OS initiates and controls the connection, EtherNet/IP Adapter Interface treats the PLC device as the master.

When communicating to a PLC device as a client, Forge/OS is pre-configured with 16 digital inputs and outputs, 16 input and 16 output integer variables, and 16 input and 16 output float variables.

EtherNet/IP Device Configuration

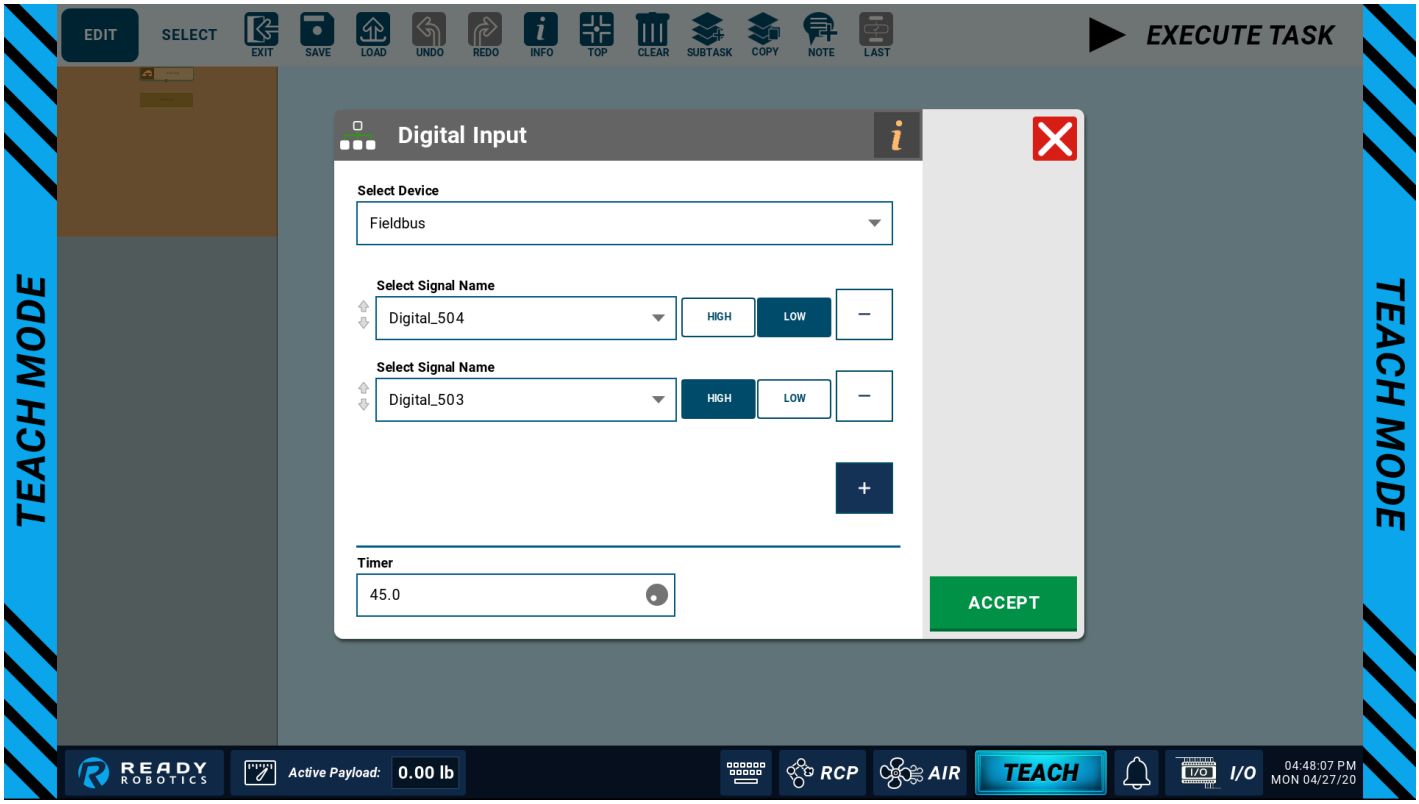
There are several parameters in your EIP device configuration that will allow it to communicate with Forge/OS as a client. These parameters should be set up in the device configuration software and may have different labels than those below.

Parameter	Values
IP Address	172.16.255.250
Input Assembly Instance or Connection Point	100
Output Assembly Instance or Connection Point	101
Configuration Assembly Instance	1
Data Type	DINT
Size or Number of Elements	Input: 33 Output: 33 Configuration: 0
Requested Packet Interval	10-3200ms, recommended 100ms
Unicast Connection over EtherNet/IP	Yes

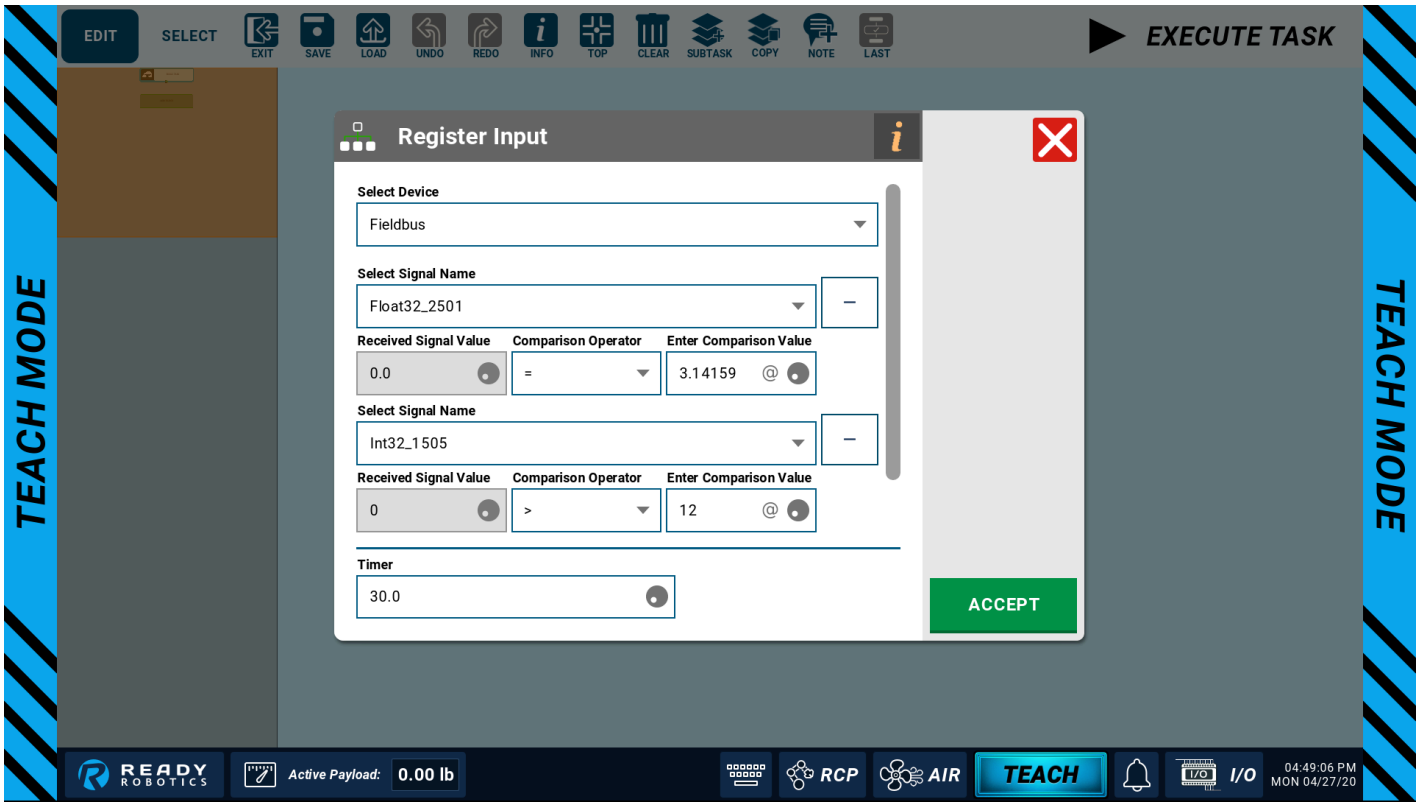
EtherNet/IP Adapter Interface and Task Canvas

Forge/OS can communicate with the master PLC device on the Task Canvas through **Digital Input, Digital Output, Register Input,** and **Register Output** blocks, under the **I/O** section of the Block menu.

In the **Digital Input** and **Output** blocks, select **Fieldbus** from the Device dropdown to populate the Signal Name dropdown with the available digital signals. For the selected signal, choose whether to check against or send a **HIGH** or **LOW** signal in the Input and Output blocks, respectively. For more information about the Digital Input and Digital Output blocks, see the PLC section.



In the **Register Input** and **Output** blocks, select the Signal Name from the dropdown and enter the check or set value for the selected signal. In the Register Input generator, the Received Signal Value field shows a real-time value from the master PLC device for the selected signal. In both blocks, you can check against or set the signal to a system variable value by tapping the **@** symbol in the Value field.



Note: If the value sent to an Integer signal comes from a float variable, Forge/OS will truncate the value when sending. As examples, 12.0, 12.1, and 12.8 will all send as **12**. -12.1 and -12.8 will send as **-12**.

Block Glossary

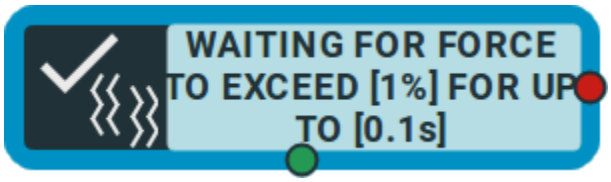
Check blocks

Check blocks check for a trigger event, such as a vision event or a force event. You can specify how the task proceeds if the event is or is not triggered.

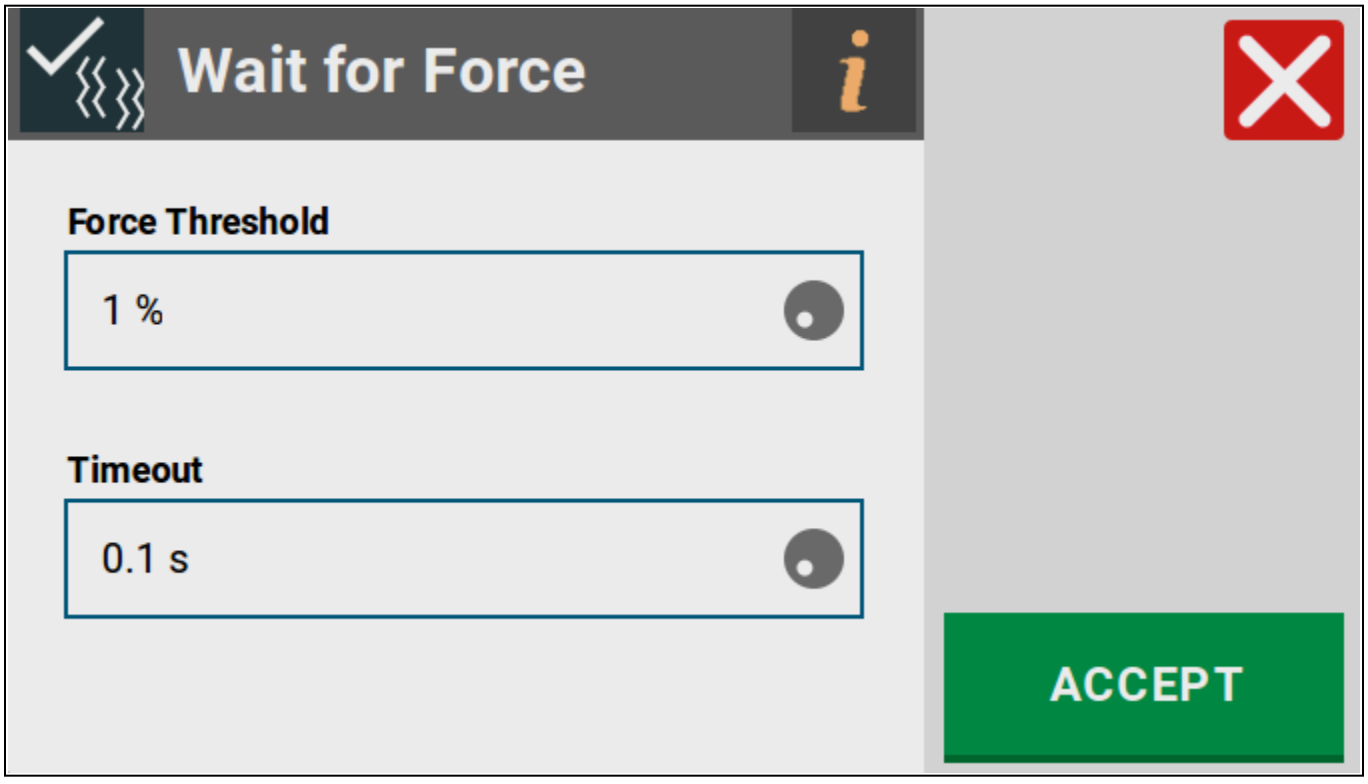
Check blocks only check for the event when the block is executed. If the event occurs while the task is elsewhere in the plan, the Check block is not triggered unless the event occurs again, or is still occurring, when the check block executes.

Wait For Force block

A Wait for Force block uses the integrated force sensor to check for force input applied to the end effector. The block follows the **FORCE OBSERVED** pathway if force is detected while the block is waiting. The **TIMEOUT EXCEEDED WITHOUT FORCE OBSERVED** pathway is followed if force is not detected before the wait time expires.



Below is the Wait for Force pop-up for adding a Wait for Force block.



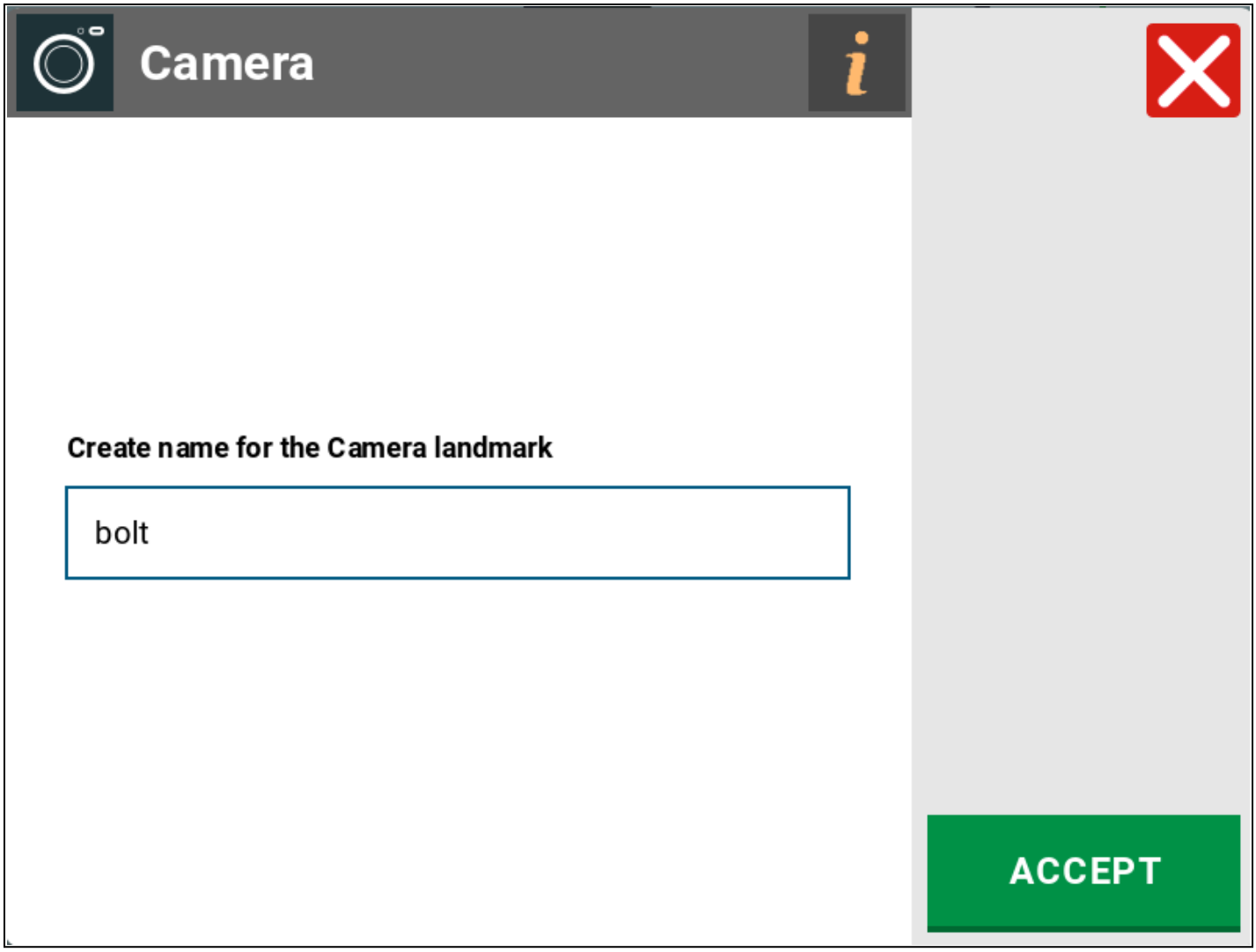
Setting	Description
Force Threshold	Set the amount of force required to trigger the task to follow the **FORCE OBSERVED** pathway.
Timeout	Set how long the block waits for force before following the **TIMEOUT EXCEEDED WITHOUT FORCE OBSERVED** pathway.

Camera block

A Camera block uses the attached camera to detect the presence of a part. The block follows the blue pathway if the part is detected by the camera. The orange pathway is followed if the part isn't detected by the camera.

When placing a Camera block, you can provide a custom label for the block. The label you create is used to name the landmark of the part's location. For example, entering "part_location" as the Camera block label changes the landmark name to "shared/part_location." If no label is provided, the Camera block label is "Detect Part" and the landmark name is "shared/camera_location."

Below is the Camera block generator which pops up when adding a Camera block.



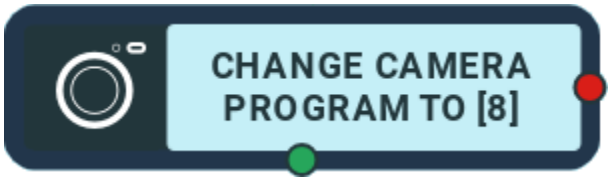
Setting	Description
Create name for the Camera landmark	Enter the name of the landmark that will be mapped to the part's location.

Control blocks

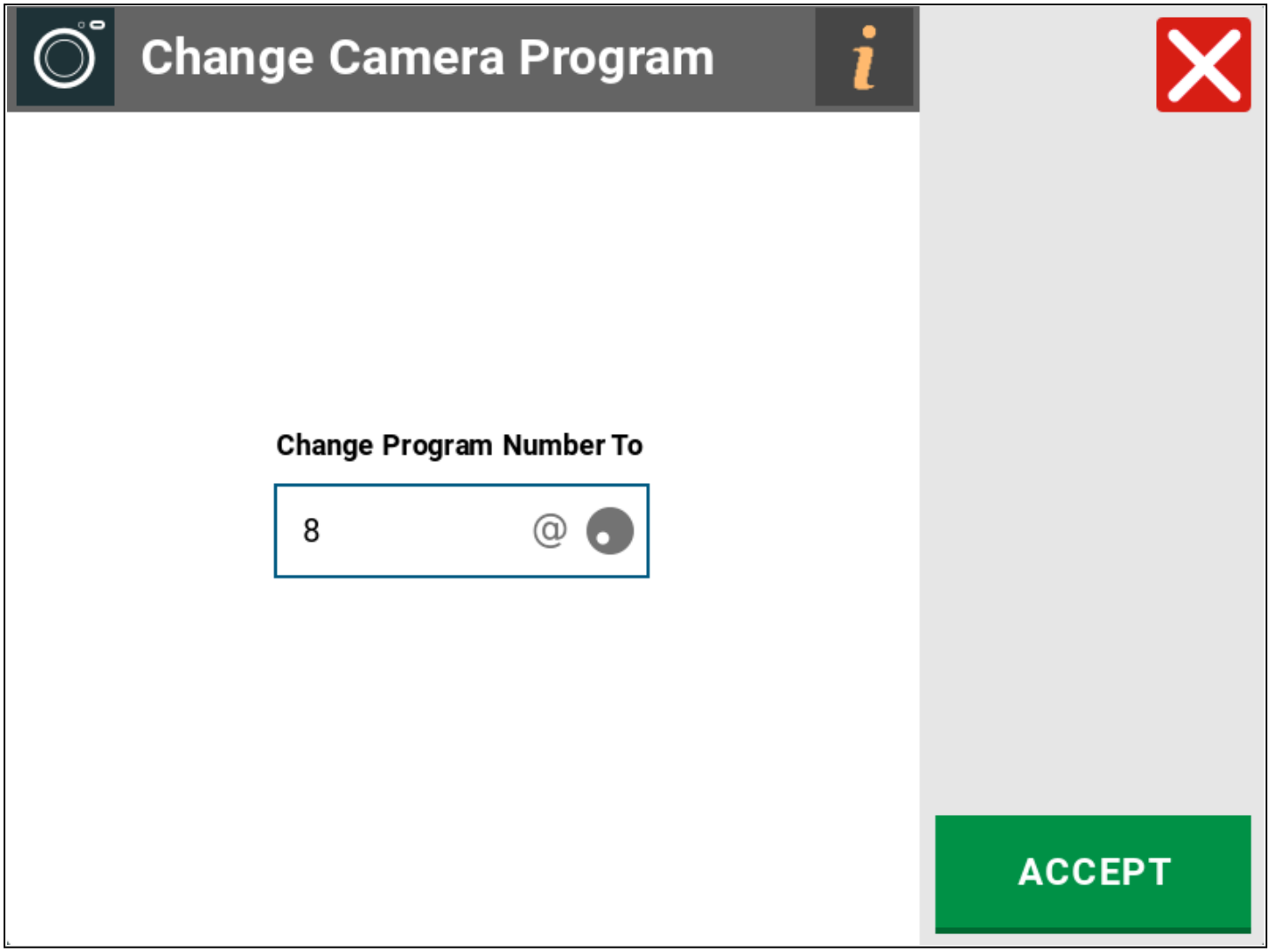
Control blocks enable you to control the behavior of the task. With control blocks, you can pause the task, repeat functions, save log messages, save waypoints, create containers, and more.

Change Camera Program block

The Change Camera Program block is only available from the Block Menu when a Keyence camera is set up in the configuration. This block is used to change the program running on the Keyence controller from Forge/OS.



Below is the Change Camera Program generator which pops up when adding a the block.



Setting	Description
Change Program Number To	Enter the program number that you wish to call from the Keyence controller.

Container

The Container option in the Blocks menu creates an empty container on the Canvas. Containers are groups of blocks that are treated as a single unit.

Setting	Description
Title	Set the name of the container.

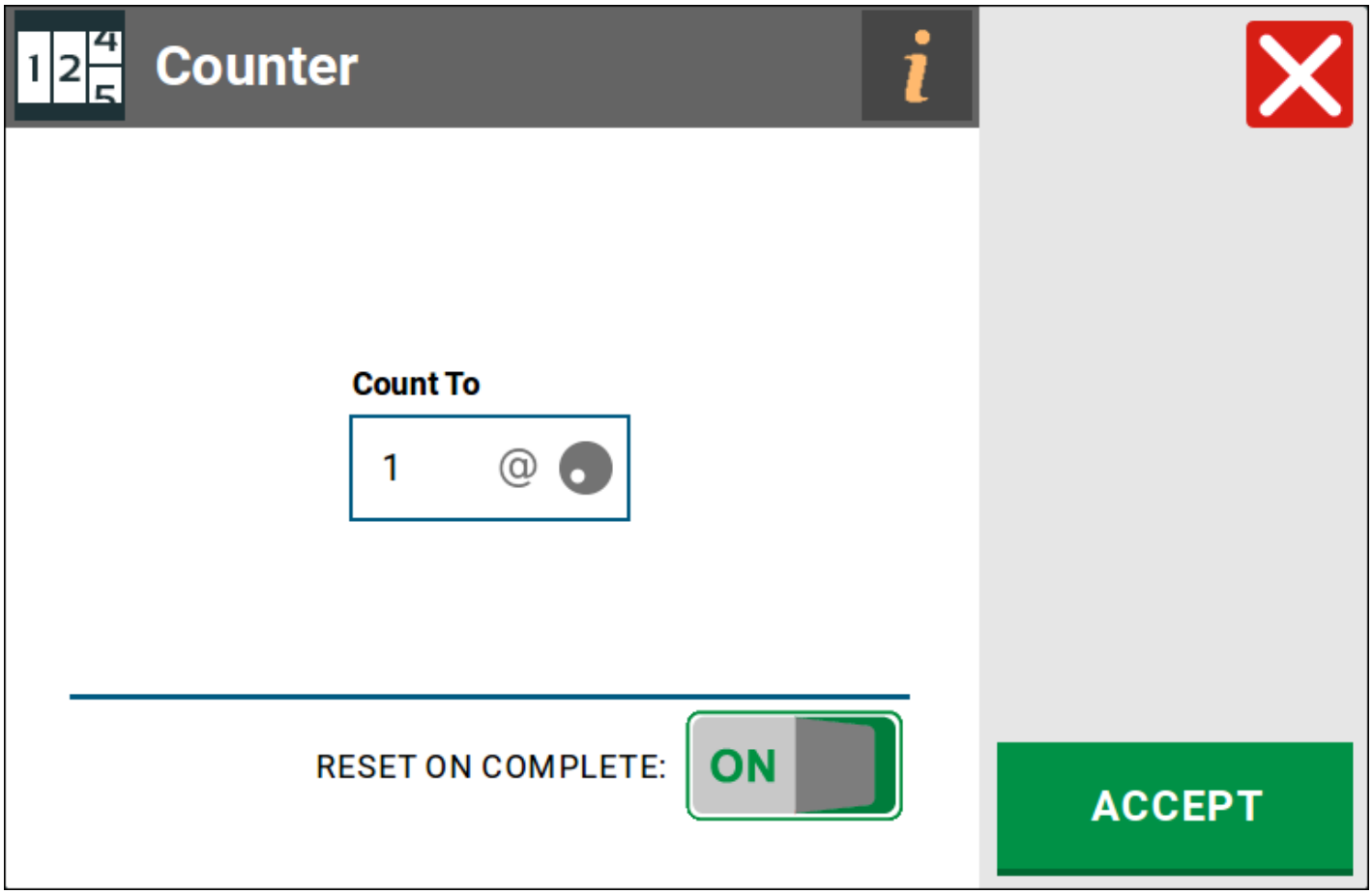
Setting	Description
Minimized	Toggle the container's state after creation. When off, the container starts in an expanded state and blocks in the container are visible on the Canvas. When on, the container starts in a collapsed state. To add blocks to the container or view blocks in the container, tap Expand (+) on the Canvas.
Reset Internal Nodes on Execute	Toggle how the values of incremental blocks (e.g. Counter, Grid) are handled when the task enters the container. When on, incremental blocks inside the container reset to their default values regardless of their current value. When off, the incremental blocks increment their values as normal and reset when complete. If an incremental block inside the container is individually set to not reset on complete, the container overrides the setting when Reset Internal Blocks on Execute is on.

Counter block

A Counter block counts to a number you define. Each time the block executes, the count increments by one. Counter blocks are a great way to create repeating loops in a task. Use the defined value to set how many times the loop repeats before moving on in the task.

The Counter block follows the **COUNT LESS THAN** pathway when the current value of the block is less than the defined value. For example, suppose you define the Counter's value as five and the block's current value is three. The **COUNTED TO** pathway is followed when the block's current value equals the defined value. For a Counter value greater than one, the block must have a both a **COUNTED TO** and **COUNT LESS THAN** pathway to function properly. If there is no **COUNT LESS THAN** pathway, the task will end as soon as the Counter block executes. Defining the Counter value as one causes the block to only follow the **COUNTED TO** pathway.

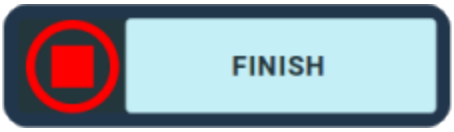
Below is the Counter modal that pops up when adding a Counter block.



Setting	Description
Counter value	Set the Counter's value by sliding the dial or by tapping the key pad and entering the value. The minimum Counter value is one; the maximum value is one million.
Reset on complete	Toggle when the Counter value resets. By default, the Counter block resets the count after the task starts following the **COUNTED TO** pathway. Counter blocks that are inside a container reset when the task leaves the container, even if the Counter has not reached the defined value. This function can be used to create Counter blocks that continue counting when a specific condition is not met but immediately reset when the condition is met.

Finish block

The Finish block marks the end of the task. While a task doesn't need a Finish block to end, it is recommended that the Finish block be used to mark where you expect the task to end. Depending on the complexity of your task, your task may use multiple Finish blocks.

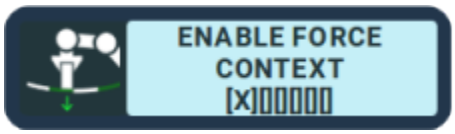


Force Context Enable and Disable block

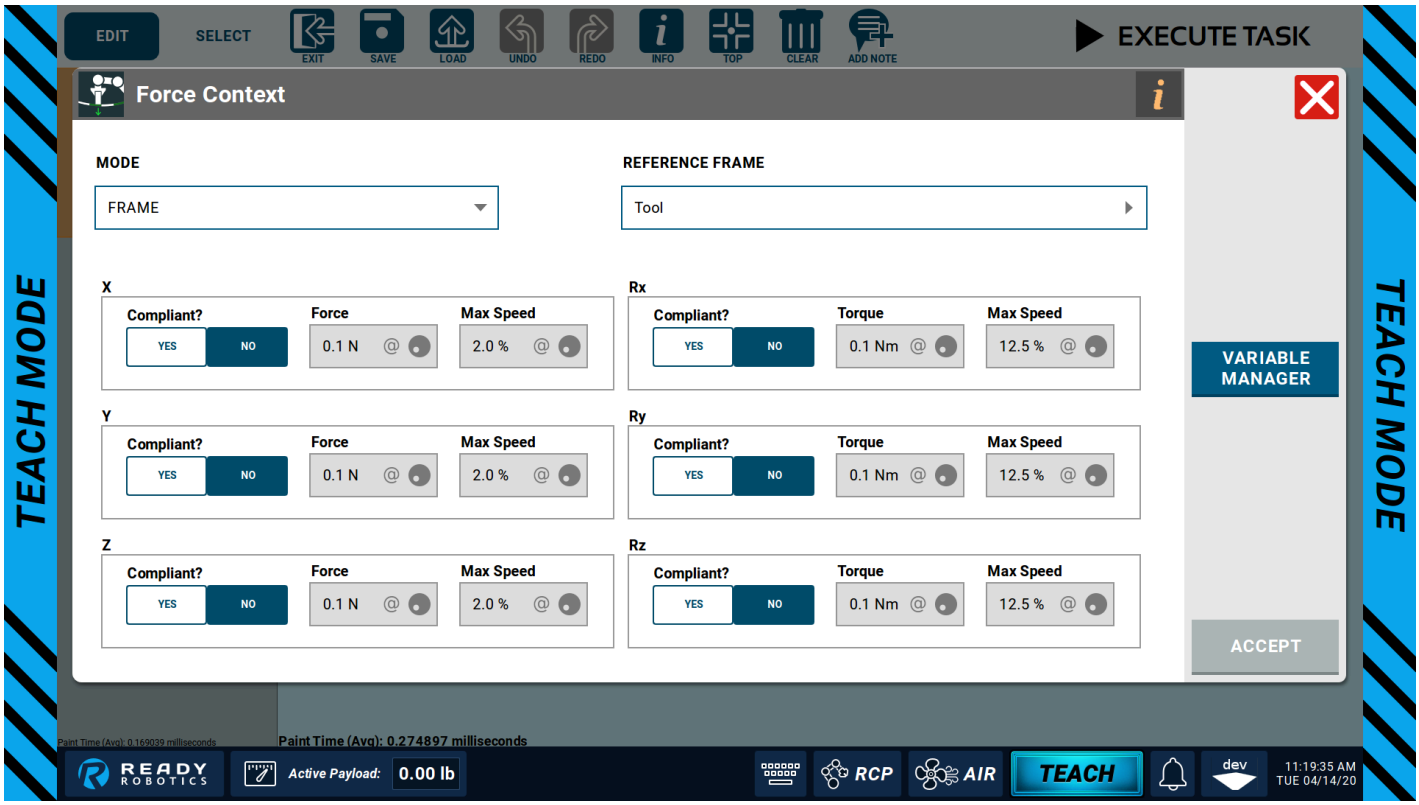
Force Context blocks define a consecutive string of blocks during which the arm applies a constant force in or around a specified primary axis. There are two Force Context blocks, Enable and Disable, which must be used together to start and end the section of the flowchart during which the arm applies the force.

Note: While the Force Context is enabled, motion blocks that use the end of arm Force-Torque Sensor may behave in unexpected ways. Although Force Context does not use the Force-Torque Sensor to execute, it is recommended that you do not program Force Context and Force moves to execute along the same axis simultaneously.

Note: To sand a curved surface, you must use a blend move to map out the contours of your workpiece with your tool. To account for the changes in the tool's orientation, the Force Context block must use Tool Frame with Mode set to Motion so that when the orientation of the tool changes, it applies force relative to that orientation.



Below is the generator that pops up when the adding an Enable block.



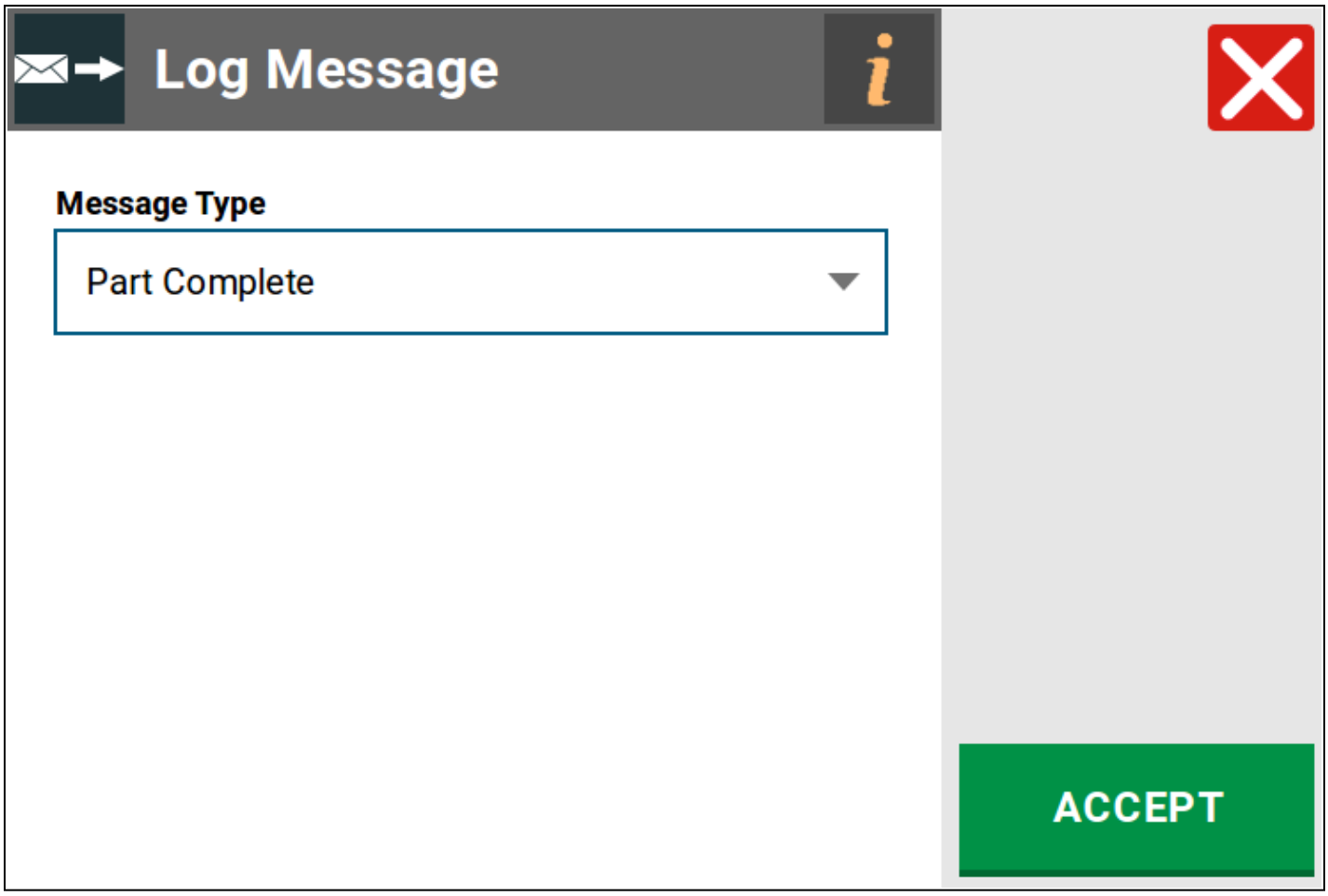
Setting	Description
Mode	Select between the two Force Modes (Frame and Motion). This determine if the applied force will be relative to a selected frame or the motion of the arm. In Frame mode, the arm will apply force in or around the selected direction of the specified frame. The frame can move with the arm or be fixed in space. In Motion mode, the arm will apply force in or around an axis based on the direction of motion of the TCP. Because the motion is based on the TCP, the Reference Frame can only be Tool.
Reference Frame	Select the frame that you want the axes of the force direction to be in relation to.
Compliant?	Select if you would like to activate Force Context in the designated axis by setting Compliance to **YES**. Otherwise leave it as **NO**.
Force	Specify the Force or Torque that the arm applies during Force Context.
Max Speed	Set the Max Speed value to limit the speed of the arm in the direction of the applied force.

Setting	Description
Variable Manager	Open the Variable Manager.

Log Message block

The Log Message block saves an internal message to note when a specific action has occurred in the task. When a Log Message block executes, Forge/OS saves a message with a timestamp and continues to the next block in the task. You are not able to view the saved message. The READY Robotics team can analyze the message and provide information on the task cycle time and material use statistics.

Below is the Log Message generator which pops up when adding a Log Message block.



Setting	Description
Message type	Select a predefined message or enter a custom message.

Robot Function block

A Robot Function block calls a programmed task or program on the robot OEM controller. The user specifies the program by its name on the OEM controller.

While the OEM program is executing, Forge will keep the flowchart on the Robot Function block until the OEM program has completed, after which Forge will continue the flowchart on the Robot Function block's blue path. If the Robot Function block cannot execute the program on the OEM controller, Forge will continue the flowchart on the block's orange path.

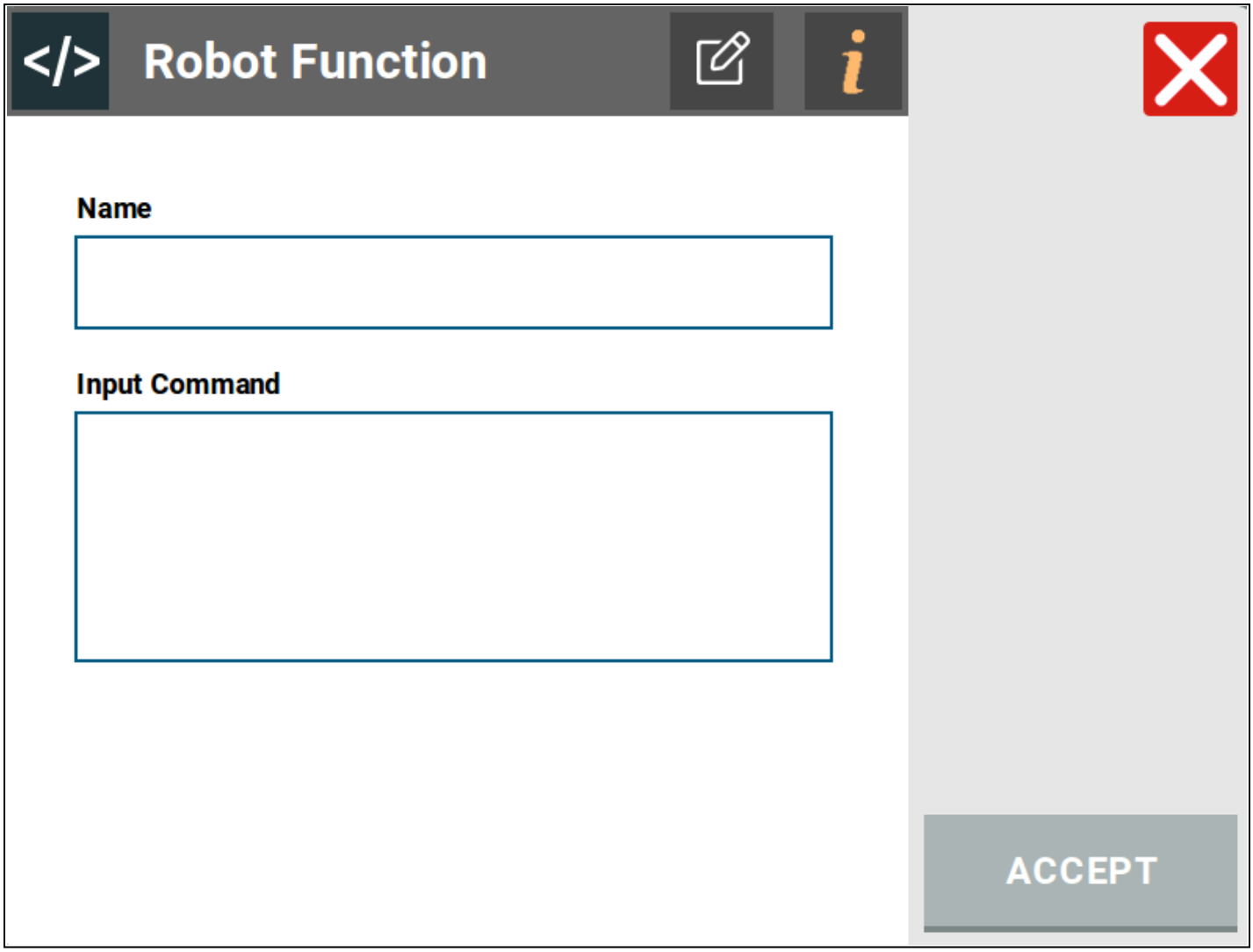
If anything on the OEM program fails or stops the robot due to an error, the Forge task will stop as well.

Note: If the called program on the OEM controller attempts to move the robot arm while Forge is moving the robot arm, both controllers will return an error and stop the task.

Note: On FANUC systems, it is possible to start a background thread on the FANUC controller that will continue executing after the controller tells Forge that the program is complete. In this case, the background thread will continue executing while Forge continues the flowchart along the Robot Block's blue path.

Note: The function called in a Robot Block must have a determinate end in its program or else the task will run indefinitely.

Below is the Robot Function generator which pops up when adding a Robot Function block.



Setting	Description
Name	Enter a Name for the program that will display in Forge. The Name field does not relate to the program name on the OEM controller, but is only local to Forge.
Input Command	Enter an Input Command that calls the program on the OEM controller. The program name in the Input Command must exactly match the name on the controller or the block will fail.

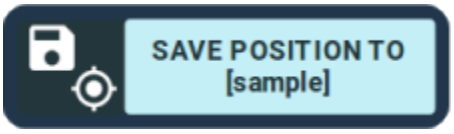
Save Position block

The Save Position block enables you to save the current position of the robot arm as a new waypoint or landmark or update to a current waypoint or landmark. A waypoint created from the Save Position block is treated like any other waypoint and can be used elsewhere in the task. The same is true for a landmark

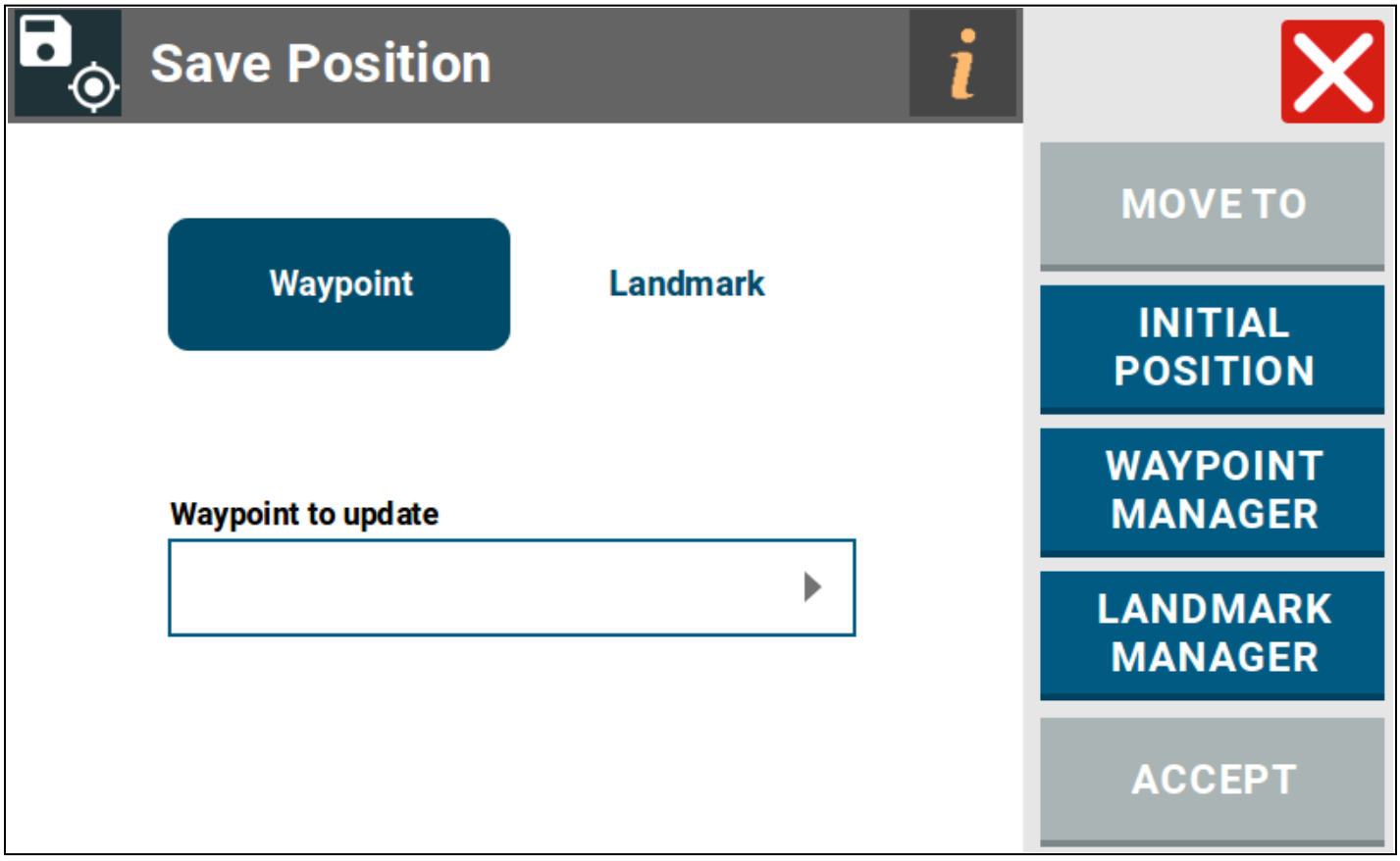
created from the Save Position block. The saved landmark can be used elsewhere in the task and referenced by other waypoints. Waypoints that reference a landmark updated by a Save Position block automatically update their positions in relation to the new location of the landmark; see the "Waypoint Based Movements" section for more information on waypoints and for more information on landmarks.

Use Save Position to create a:

- Waypoint that is expected to change but that Forge/OS needs to remember. For example, the top of a stack of parts that the robot arm is pulling from, or the location of an object on a conveyor belt that the robot arm returns to later.
- Landmark to update a sequence of motions, such as a Grid move or an operation that is the same across several shelves.



Below is the Save Position generator which pops up when adding a Save Position block.

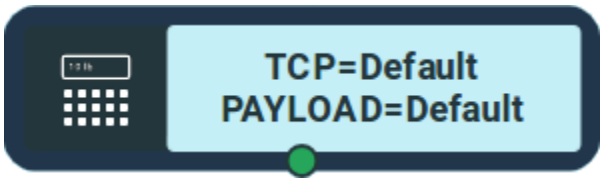


Setting	Description
Waypoint/Landmark	Select how the position is to be saved, as a waypoint or a landmark.
Waypoint/Landmark to update	Select to save the position as a new waypoint or landmark or select an existing waypoint or landmark to update. The list of existing options only displays options that match how the position is to be saved. For example, if you choose to save the position as a waypoint, only waypoints are listed under the existing section.

TCP/Payload block

The Set TCP/Payload block enables you to change the Tool Center Point (TCP) and Payload at the end of arm while the task is executing. Use this block when the robot picks up or puts down a heavy part, or changes to a tool with a different geometry.

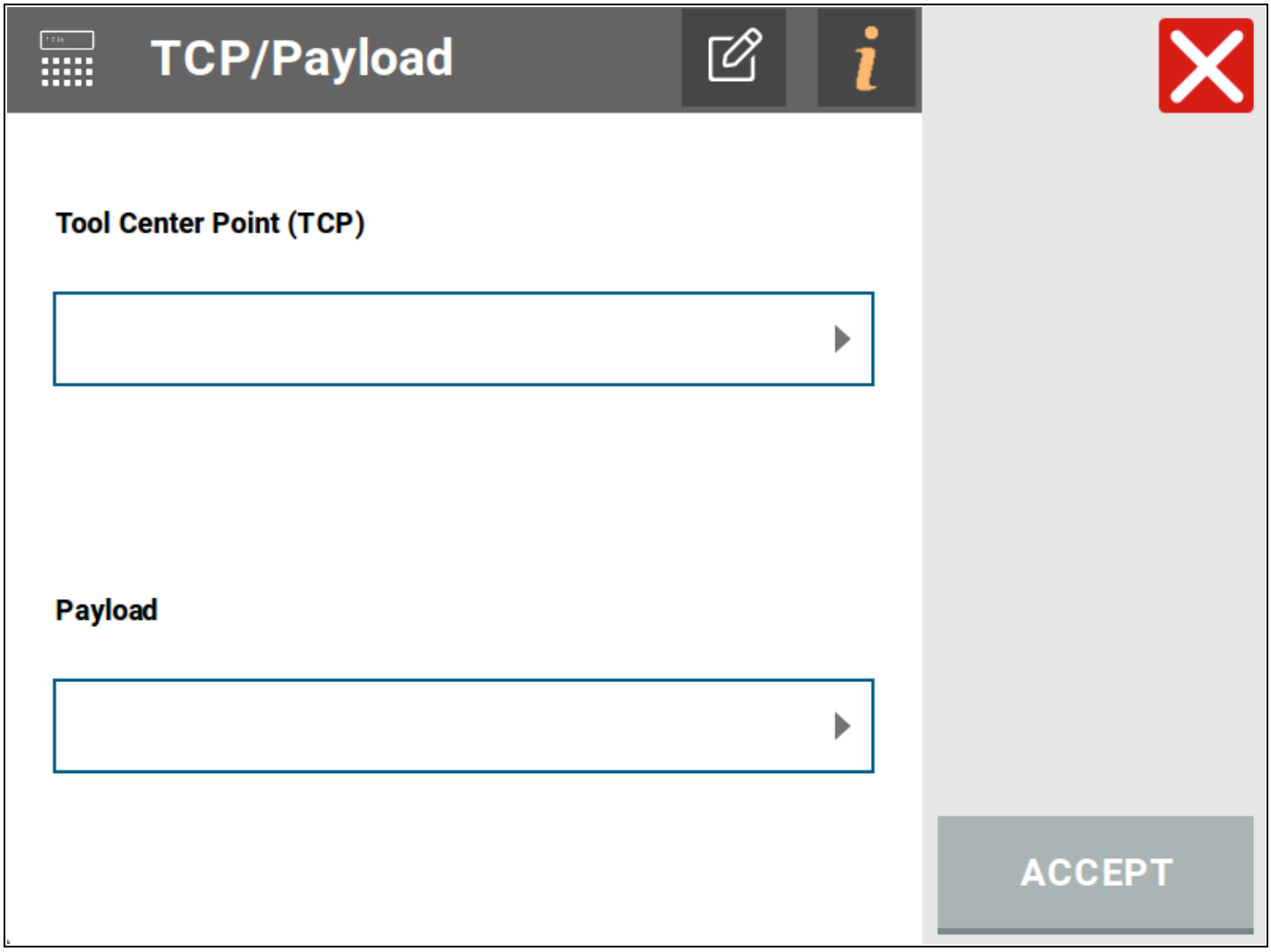
In order for a TCP or Payload to appear in their respective fields, the TCP or Payload needs to be part of the current configuration in the System Configuration Application.



Below is the TCP/Payload generator which pops up when adding a TCP/Payload block.

Note:Changing the TCP will change how the robot arm moves to waypoints and during relative jumps. Make sure that the selected TCP is the same TCP as when you programmed the waypoint or jump.

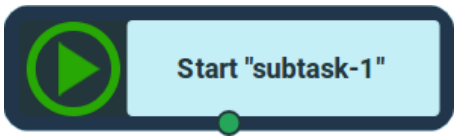
Note: Certain robot arms will interpret large differences between the programmed and actual payload as unexpected force, causing inaccurate motion and safety stops. Make sure that the selected Payload reflects the actual payload at the end of arm when the block executes.



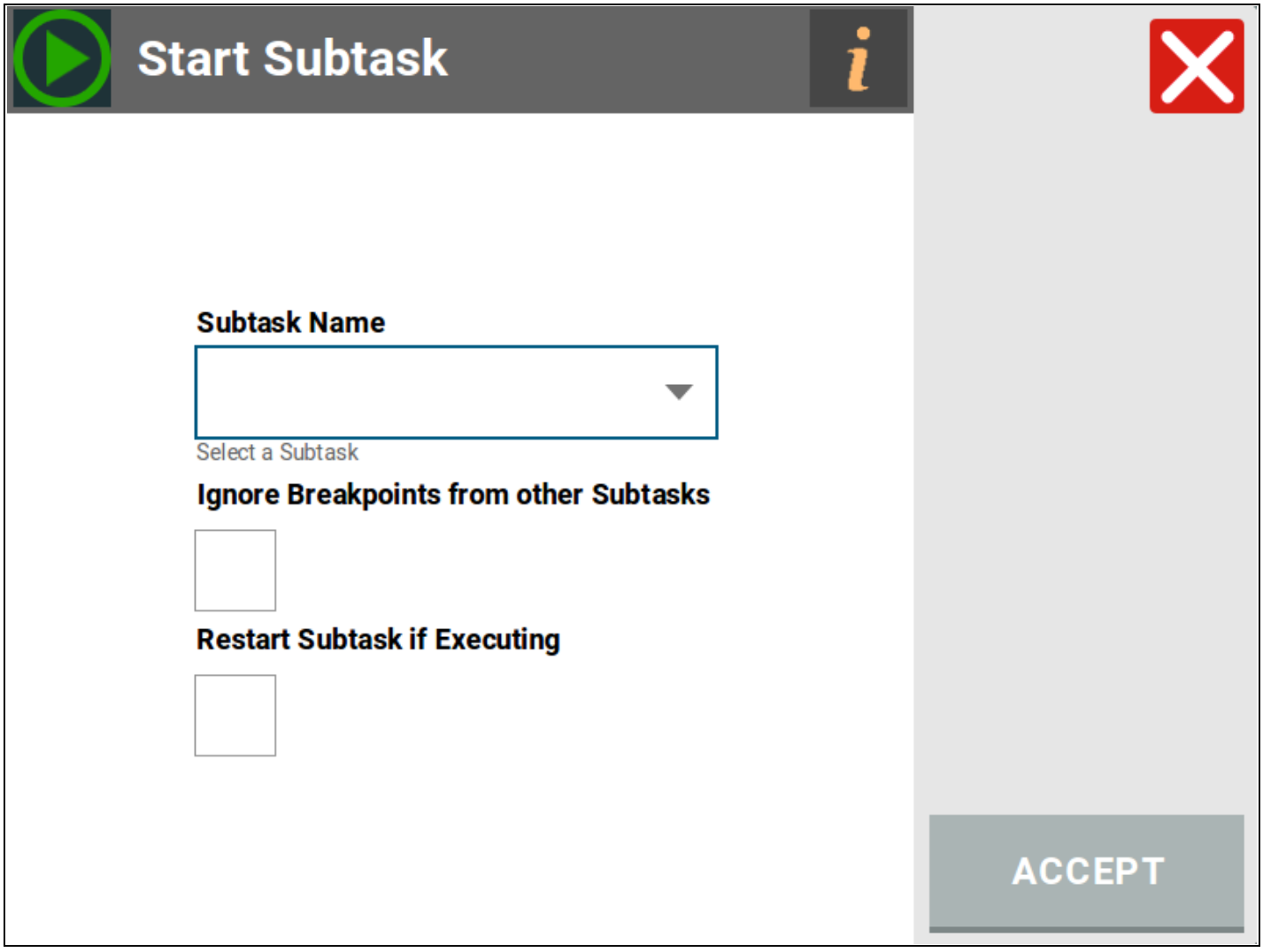
Setting	Description
Tool Center Point (TCP)	Select the available TCPs from the active Hardware Configuration.
Payload	Select the available payloads from the active Hardware Configuration.

Start Subtask block

A Start Subtask block is how an executing task starts a Subtask. A subtask will always start from the Root block when called from a Start Subtask block. Start Subtask blocks are in the Control section of the block menu.



Below is the Start Subtask generator which pops up when adding a Start Subtask block.

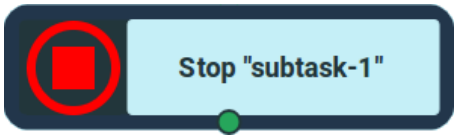


Setting	Description
Subtask Name	Select the name of the Subtask the block will start.
Restart Subtask if Executing	Give the Start Subtask block the ability to restart the called subtask from the Root block if it is already executing. If this box is unchecked and the subtask is already executing when the Start Subtask block executes, the subtask will ignore the call and continue.

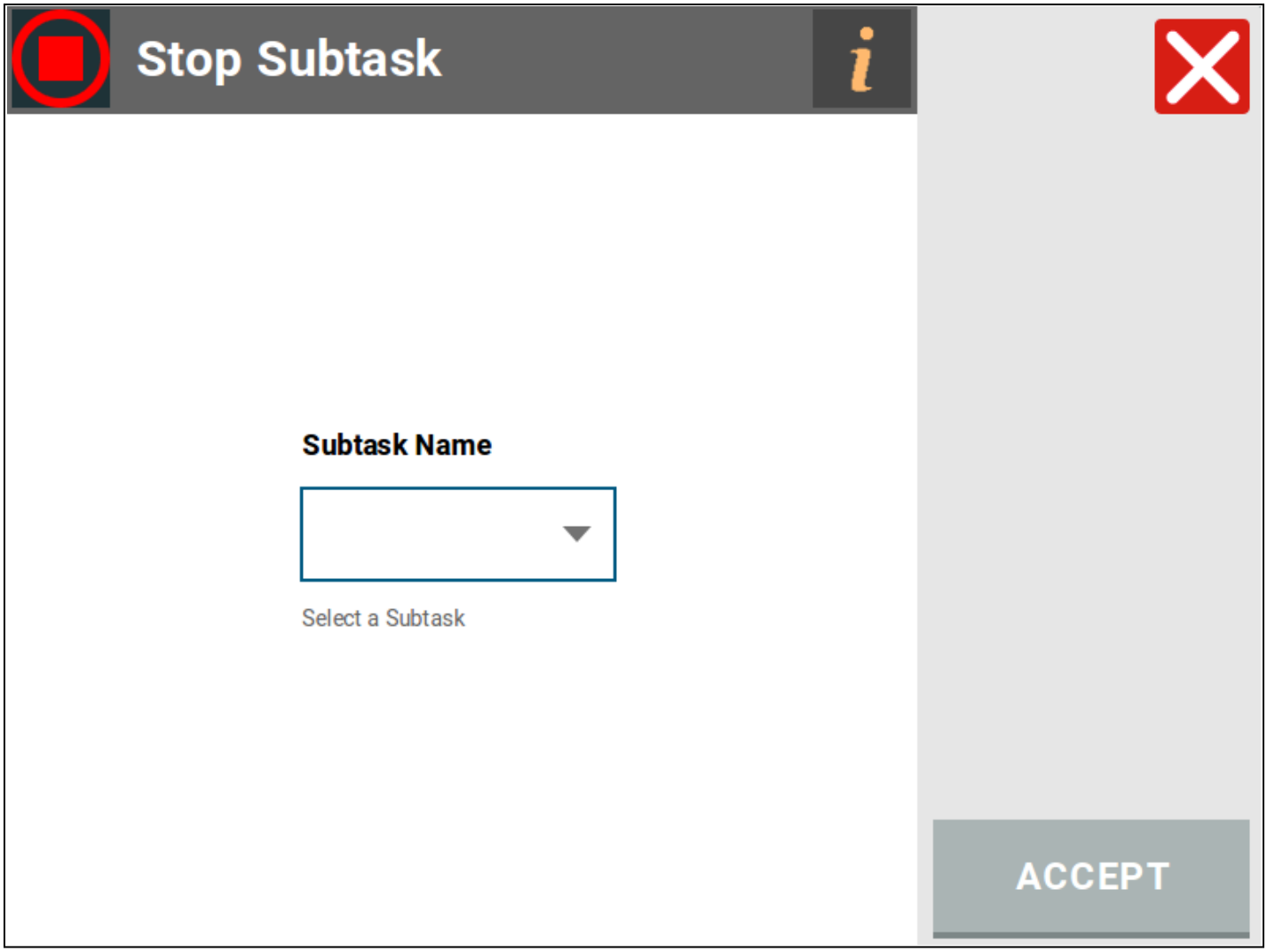
Setting	Description
Ignore User Action blocks from other Subtasks	Give the subtask the ability to continue executing if another subtask reaches a User Action block. This subtask will pause for its own User Action blocks, but ignore all others. Use this block to continue executing subtasks that could miss key information from external inputs or devices if the Task is paused for other reasons, such as notifying the user of Task status.

Stop Subtask block

A Stop Subtask block is how an executing Task stops a subtask. When a Stop Subtask block stops a subtask, the subtask will stop immediately. Stop Subtask blocks are in the Control section of the block menu.



Below is the Stop Subtask generator which pops up when adding a Stop Subtask block.



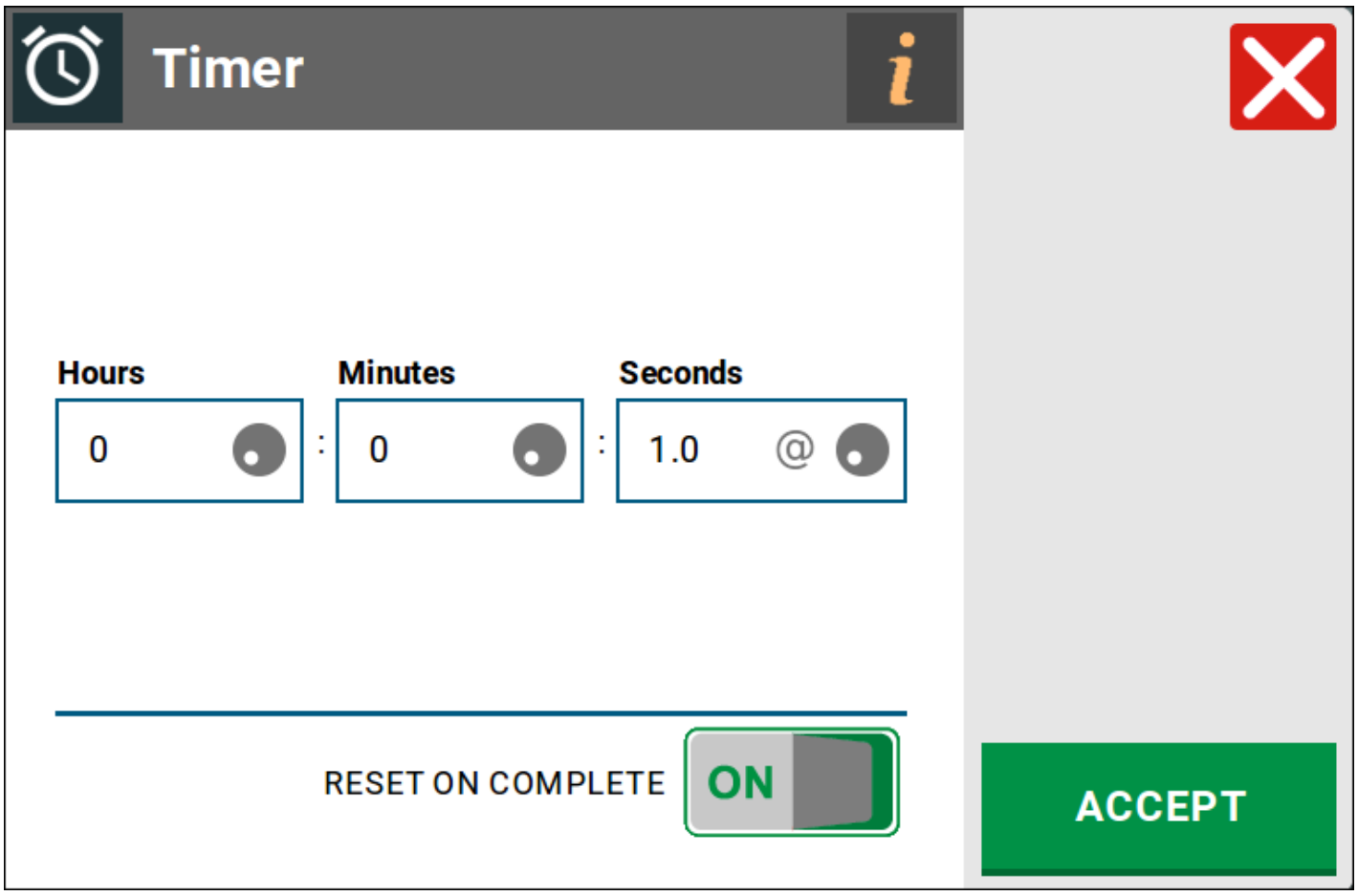
Setting	Description
Subtask Name	Select the name of the Subtask the block will stop.

Timer block

The Timer block allows you to create a subtask that occurs for a defined length of time. The timer starts with the Timer block executes and doesn't stop until the time runs out. The **ELAPSED TIME LESS THAN** path is followed while the timer is running. By default, the timer restarts the next time the Timer block executes. To create a looped function that executes while the timer is running, set the looped function on the Timer block's **ELAPSED TIME LESS THAN** path and connect the end of the function back to the Timer block. A Timer block must have both an **ELAPSED TIME GREATER THAN** and **LESS THAN** pathway to function properly. If there is no **ELAPSED TIME LESS THAN** pathway, the task ends as soon as the timer starts. If there is no **ELAPSED TIME GREATER THAN** pathway, the task ends as soon as the timer finishes.

The Timer is functionally the opposite of the Wait block. The difference is in what happens while the block is executing. While executing a Wait block, the system does not perform any actions until the Wait block ends; whereas during the execution of a Timer, the system performs any action on the Timer's **ELAPSED TIME LESS THAN** pathway until the timer ends.

Below is the Timer generator which pops up when adding a Timer block.

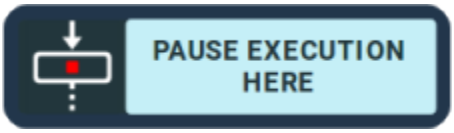


Setting	Description
Hours	Enter the number of hours, minutes, and seconds of the timer. ▪ The minimum time is 0.1 seconds. ▪ The maximum time is 23 hours, 59 minutes, and 59.9 seconds
Minutes	
Seconds	

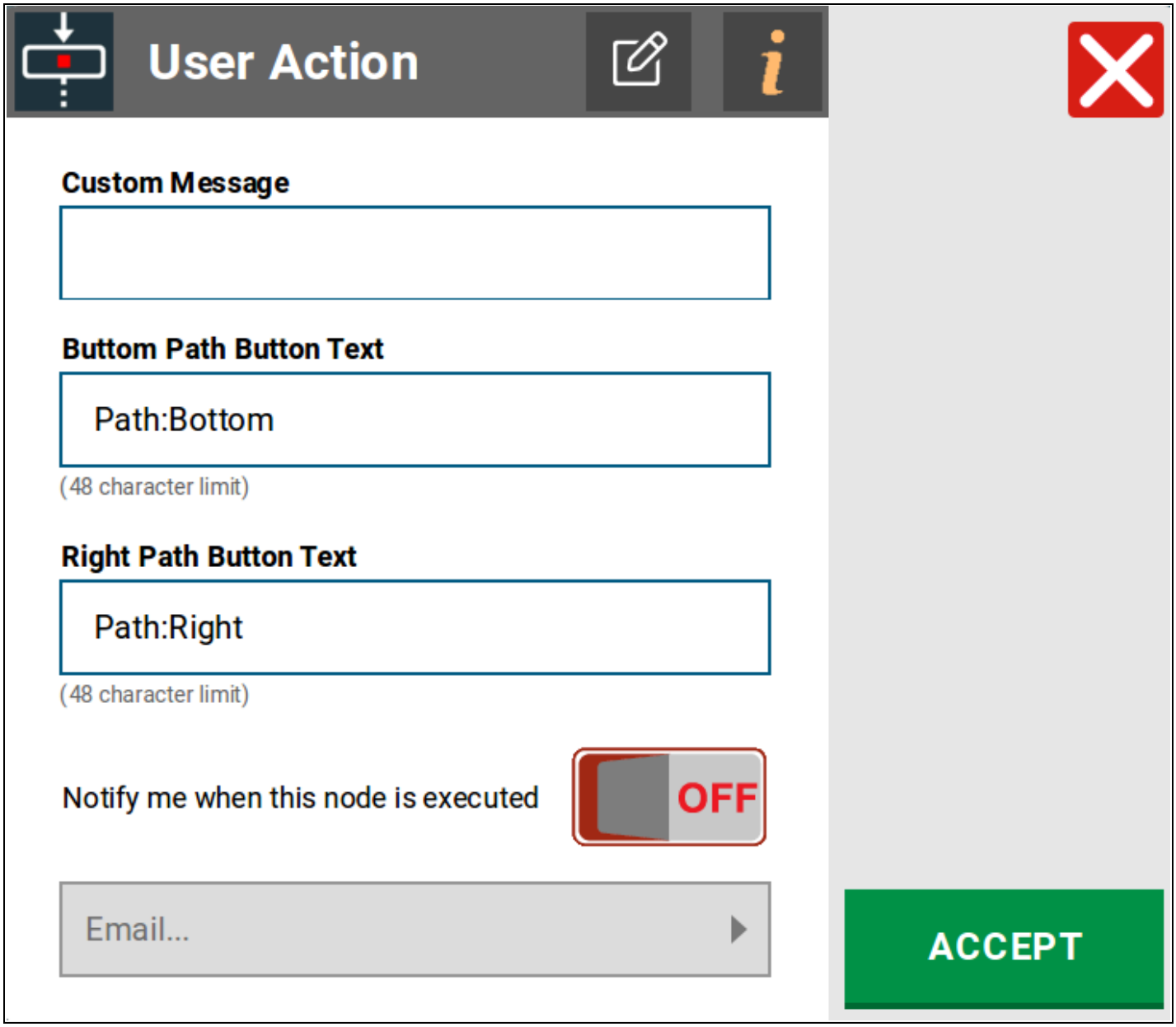
Setting	Description
Reset on Complete	Toggle the how the Timer block behaves after time runs out. By default, the Timer block resets when the timer reaches zero and the task starts following the **ELAPSED TIME GREATER THAN** pathway. When off, the task permanently follows the **ELAPSED TIME GREATER THAN** pathway after the timer reaches zero. Timer blocks that are inside a container reset when the task leaves the container, even if the Timer has not finished. This is the only way to reset a Timer block when Reset on Complete is off. This function can be used to create Timer blocks that continue counting when a specific condition is not met but immediately reset when the condition is met.

User Action block

A User Action block pauses the task until you either instruct the task to resume or quit. While paused, the robot arm does not move, and the end effector and attached devices hold their current state. You can program the User Action block to display custom text when it executes.



Below is the User Action generator which pops up when adding a User Action block.



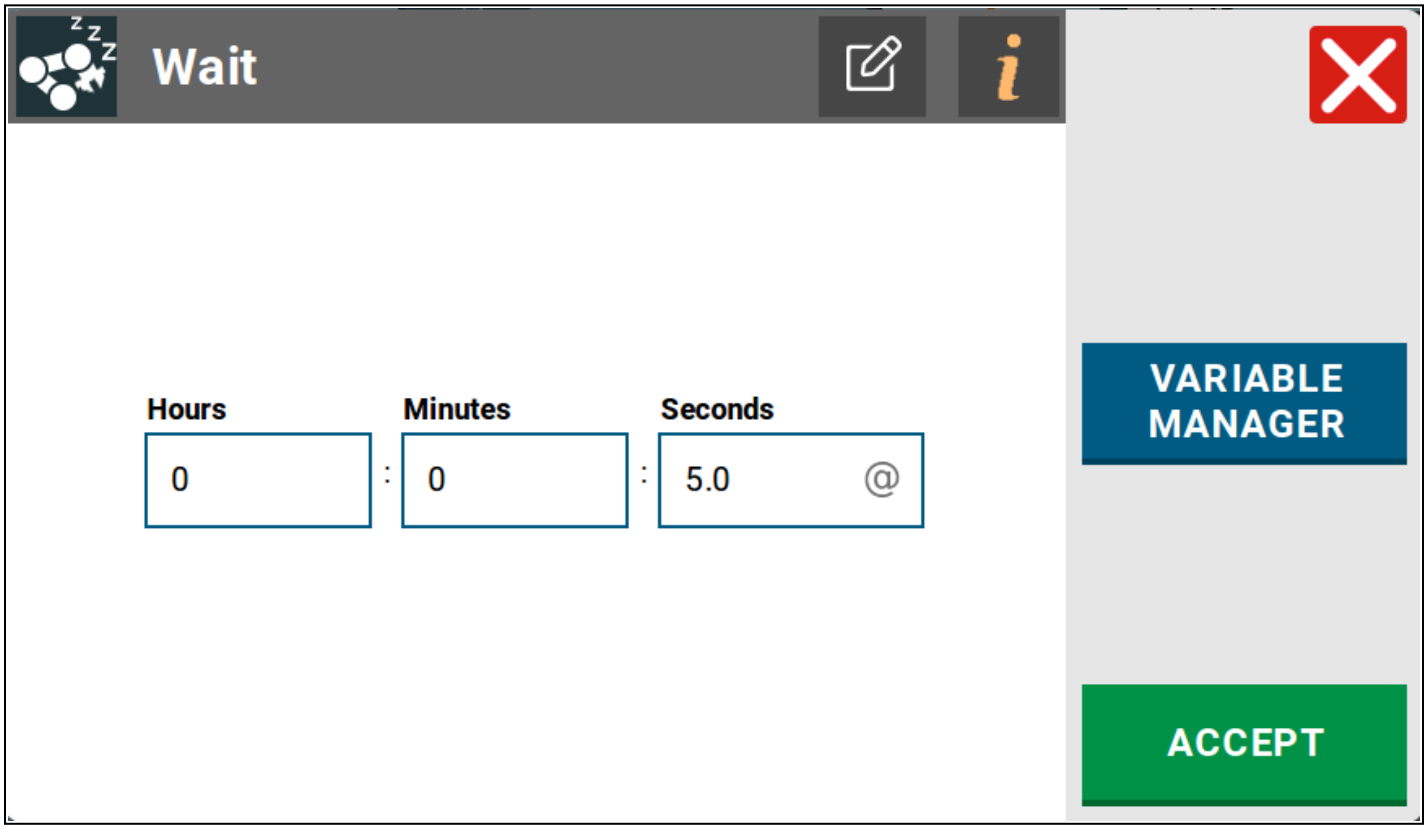
Setting	Description
Custom Message	Enter text that Forge/OS displays when the User Action block executes. If left blank, Forge/OS displays "Graph execution paused."
Bottom Path Button Text	Enter the text that appears on the button to proceed with the task along the bottom path.
Right Path Button Text	Enter the text that appears on the button to proceed with the task along the path to the right.

Setting	Description
Notify me when this block is executed	Toggle email notifications. When on, Forge/OS sends an email notification to the selected email address when the User Action block executes. See the "Manage Notification Emails" section of the "The Control Suite" chapter for information on adding, editing, and deleting email addresses.

Wait block

The Wait block instructs the system to pause, or dwell, for a defined length of time. While a Wait block is executing, the robot arm holds its position and the end effector and attached devices hold their state. When the Wait block ends, the task continues.

Below is the Wait generator which pops up when adding a Wait block.



Setting	Description
Hours	Enter the number of hours, minutes and seconds of the Wait block. ▪ The minimum time for a Wait block is 0.1 seconds. ▪ The maximum time for a Wait block is 23 hours, 59 minutes, and 59.9 seconds
Minutes	
Seconds	

Note: More than one Wait block can be used in succession if you need the system to wait for longer than the maximum time

I/O blocks

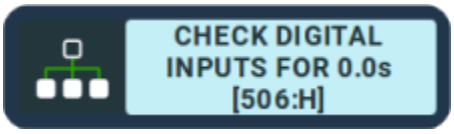
I/O blocks enable you to control the Programmable Logic Controller (PLC) Breakout Box, the robot I/O, other digital I/O, and any configured Fieldbus devices through which the system sends and receives signals. The I/O menu options are Digital I/O and Register I/O. Digital I/O is used to read and write and has control over the PLC Breakout Box, Robot I/O, and other Digital I/O. The Register I/O is used to read and write variable I/O to any configured Fieldbus devices.

For information regarding your equipment's PLC capabilities and input or output ports, consult the OEM manual or technical drawings.

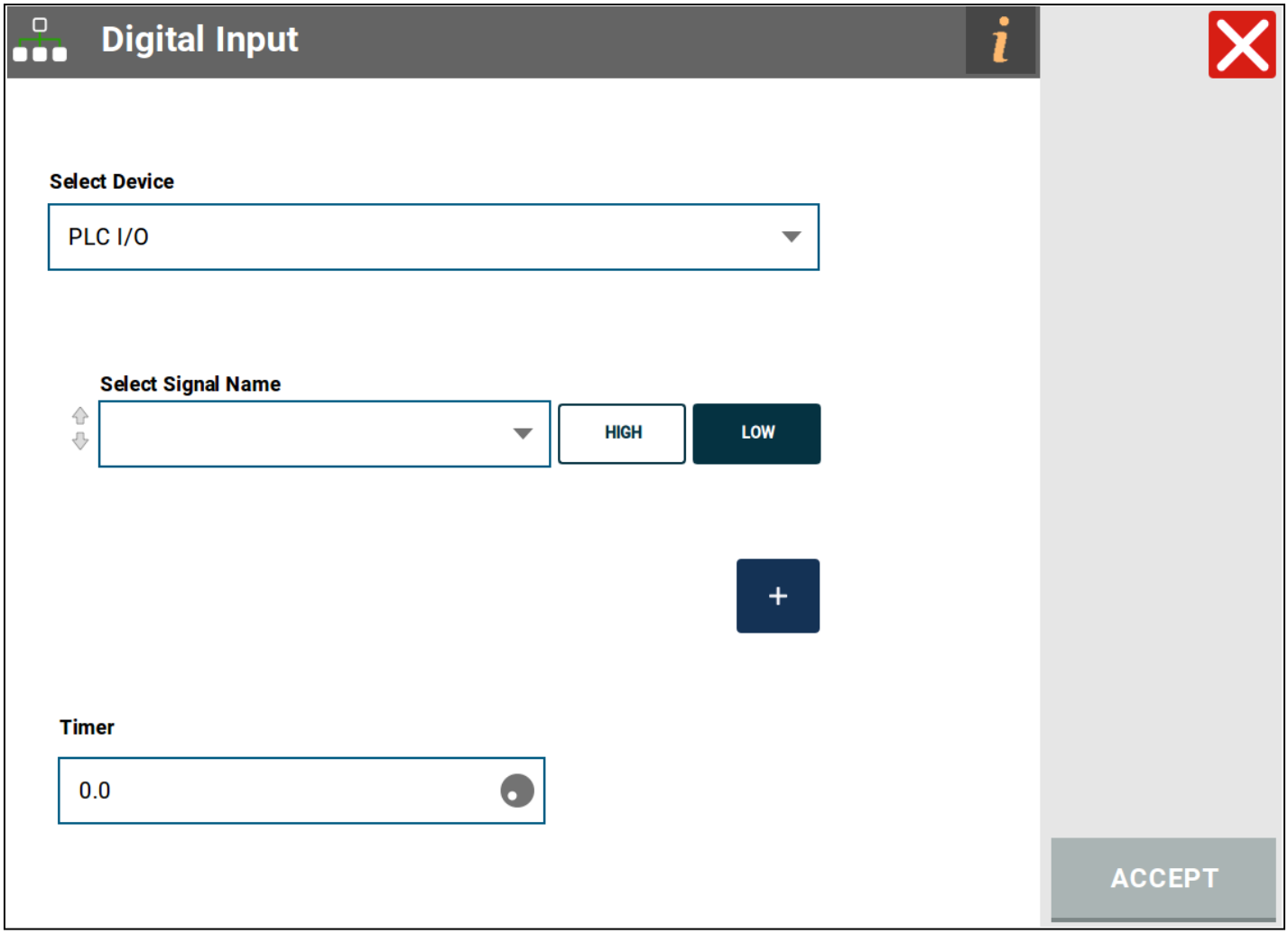
Digital Input block

The Digital Input block looks for an input of either high or low voltage on the specified channel. If the signal condition isn't met, the task follows the **INPUT CONDITIONS NOT PRESENT** pathway. If the signal condition is met, the block follows the **INPUT CONDITIONS PRESENT** pathway.

To direct the Digital Input block to constantly check for input, connect the block to itself via its **INPUT CONDITIONS NOT PRESENT** pathway. To direct the block to check for input at specific time intervals, connect its **INPUT CONDITIONS NOT PRESENT** pathway to a Wait block that follows the **WAITED FOR X** pathway back to the Digital Input block. Set the Wait block to the desired interval.



Below is the Digital Input generator that pops up when adding a Digital Input block



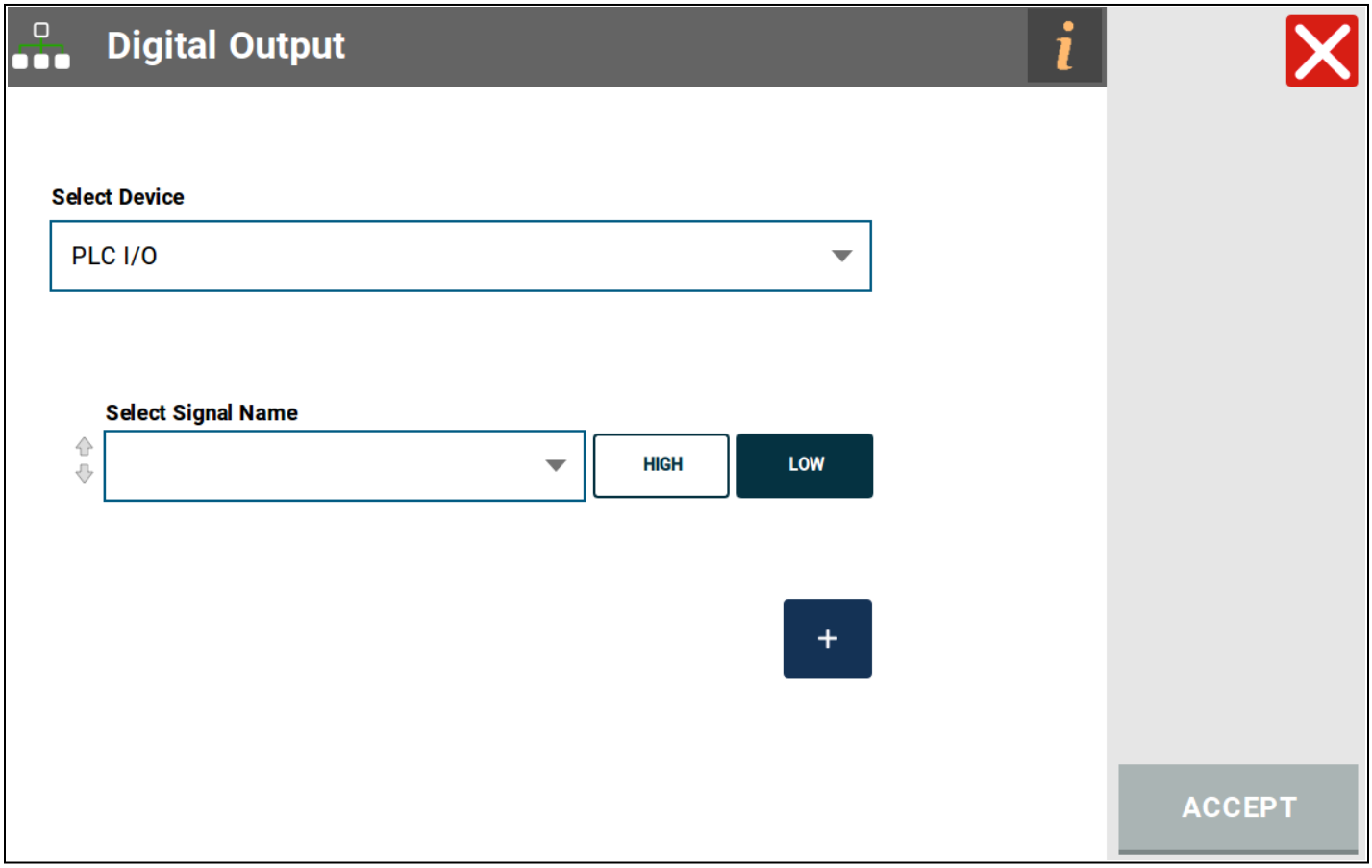
Setting	Description
Select Device	Select the device which receives the signal.
Select Signal Name	Select the port to check for a signal on.
High/Low	Toggle the voltage to read.
Timer	Set the duration of which the PLC is checking for the defined signal.
Plus	Add additional ports to check for a signal on.

Digital Output block

The Digital Output block sends a signal on the specified channel when the block executes. The voltage holds

its state until a I/O Output block executes with a change to the previously specified channel.

Below is the Digital Output generator that pops up when the adding a Digital Output block.



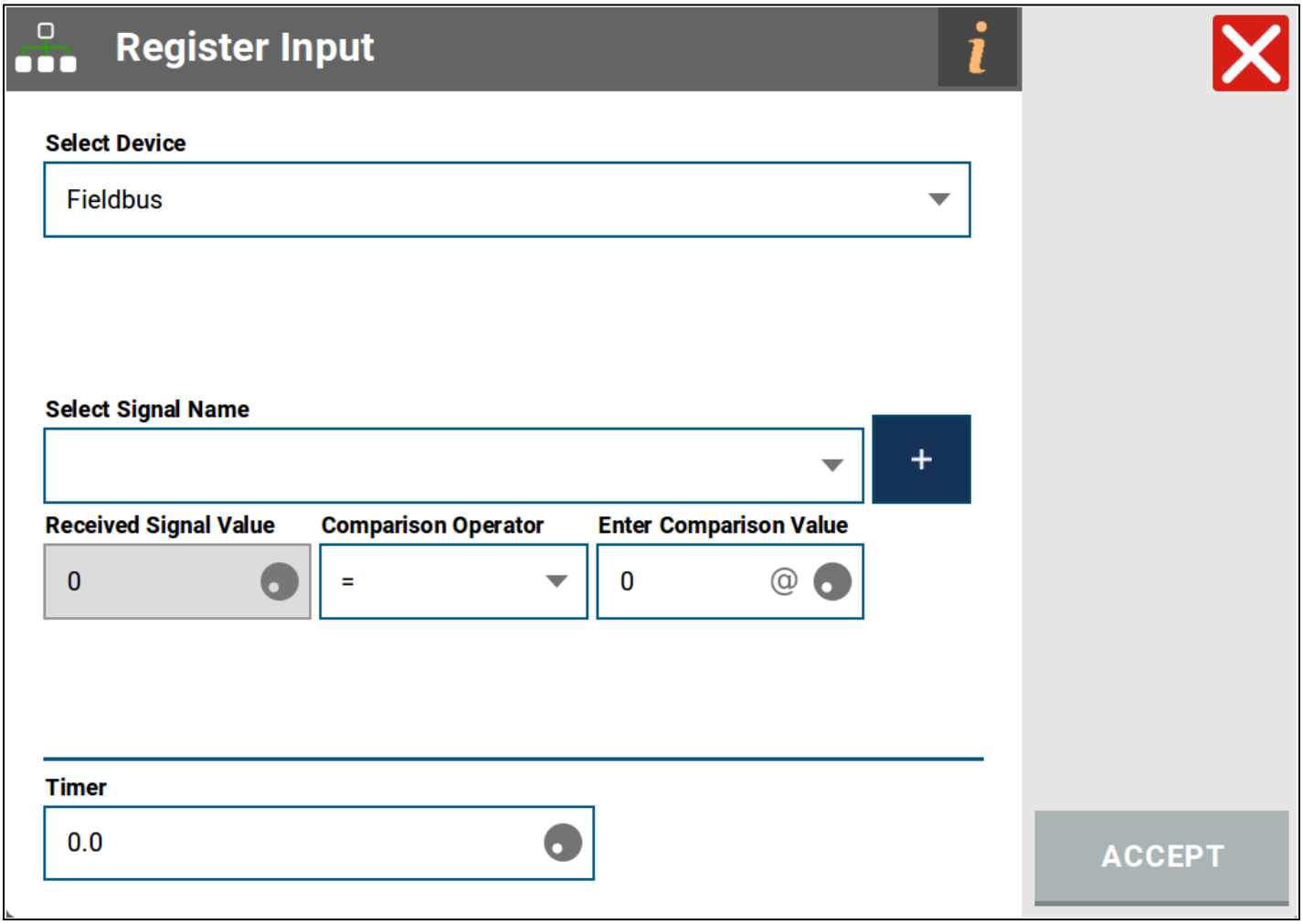
Setting	Description
Select Device	Select the device to send a signal on.
Select Signal Name	Select the port to send a signal on.
High/Low	Toggle the voltage level to send.
Plus	Add additional ports to send a signal on.

Register Input block

The Register Input block responds to variable input value on any configured Fieldbus device. The values that can be read are floats and integers. Floats are numerical values with decimal places while integers are whole numbers. Forge/OS uses IEEE Standard for Floating-Point Arithmetic which READY has limited float signals to 5 decimal places and the range to +/-99,999,999.99.

Note: Forge/OS will automatically convert data types if the received type is different from the register input type. Precision may be lost if floating point numbers are sent to integer registers. As examples, 12.0, 12.1, and 12.8 will all send as **12**. -12.1 and -12.8 will send as **-12**.

Below is the Register Input generator that pops up when adding a Register Input block.



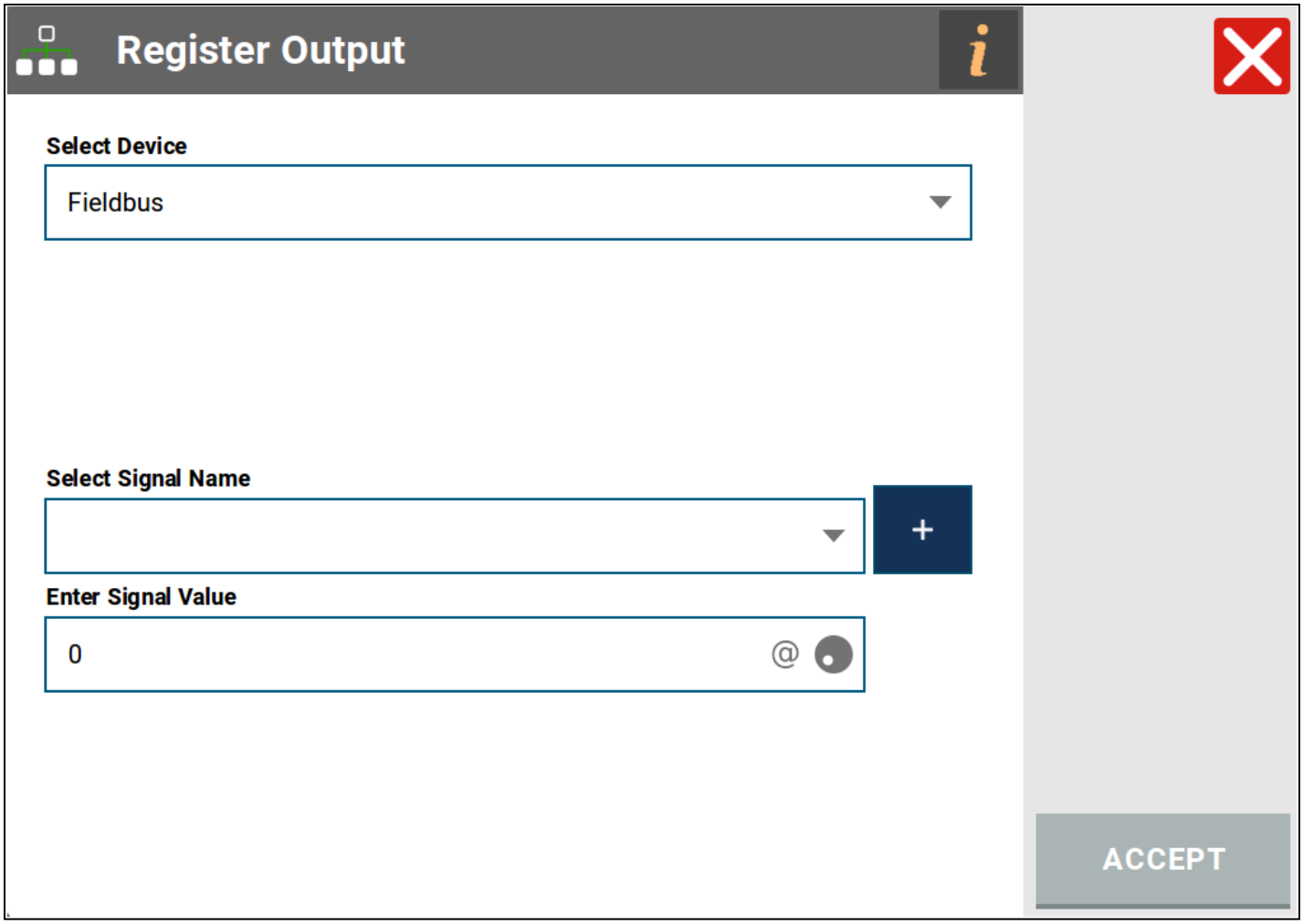
Element	Description
Select Device	Select the device which receives the signal.
Select Signal Name	Select the signal for reading the variable input value.
Add (Blue Plus)	Add additional signals.
Received Signal Value	Received Signal Value field shows a real-time value from the selected device for the selected signal.
Comparison Operator	Select the appropriate comparison operator.
Enter Comparison Value	Signal compared to from the device. The signal can be set to a system variable using the @ button.
Timer	Duration of which the selected device is checking for the variable input at the defined port(s).

Register Output block

The Register Output block sets a variable value for any configured Fieldbus device. The values that can be written are floats and integers. Floats are numerical values with decimal places while integers are whole numbers. Forge/OS uses IEEE Standard for Floating-Point Arithmetic which READY has limited float signals to 5 decimal places and the range to +/-99,999,999.99.

Note: Forge/OS will automatically convert data types if the sent type is different from the register output type. Precision may be lost if floating point numbers are sent to integer registers. As examples, 12.0, 12.1, and 12.8 will all send as **12**. -12.1 and -12.8 will send as **-12**.

Below is the Register Output generator that pops up when adding a Register Output block.



Element	Description
Select Device	Select the device which writes the signal.
Select Signal Name	Select the signal for writing the variable input value.
Add (Plus)	Add additional signals.
Enter Signal Value	Signal sent to the device. The signal can be set to a system variable using the @ button.

Logic blocks

The Logic blocks are used to manipulate and check variables so that programming logic can be implemented into tasks. Variables must be created in the Variable Manager in order to utilize these blocks.

Variable Manager

The variable manager is where you define new, edit existing, or delete variables that you want to use in your task. Through variables, you can create powerful and flexible tasks. Variables can be used to mark a point in the task based on the action that have occurred or current state of the system. For example, a variable can be set to mark a counter resetting or the moment the arm experiences feedback. The variable can be recalled later in the task and used as a decision point that changes the task's direction. In its simplest form, a variable can be used as a reference to for things like part size and count.

There are two types of variables used by Forge/OS, shared and task. This table explains the difference between a task and shared variable:

Type (Prefix)	Meaning
Task (@)	Task variables are only available in the task in which they were created.
Shared (#)	Shared variables are available in all the tasks on the Forge system.

Variables are represented using a key-value-default structure, i.e. @name(current_value, default_value). The current and default value only store numeric values that are whole numbers or numbers with decimals. Negative numbers are acceptable. This table provides more information on each part of the variable's structure:

Part	Description
Name	The key, or label, used to identify the variable. The name of a variable is created using alphanumeric charters with the exception of @ and # - those two characters are reserved to differentiate between shared and task variables.
Current_value	The value that represents the variable's current value. During the task's execution, this value can be changed or used through the use of Logic blocks.
Default_value	The variable's default value. This value is set when the variable is created and is used when the variable's current value has changed and there is a need to recall the variable's default value.

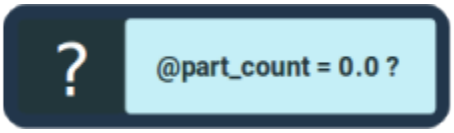
Example: A variable used to represent the length could look like: @length(5, 0). This breaks down to the following settings:

- The name is “Length”
- The current value is 5
- The default value is 0

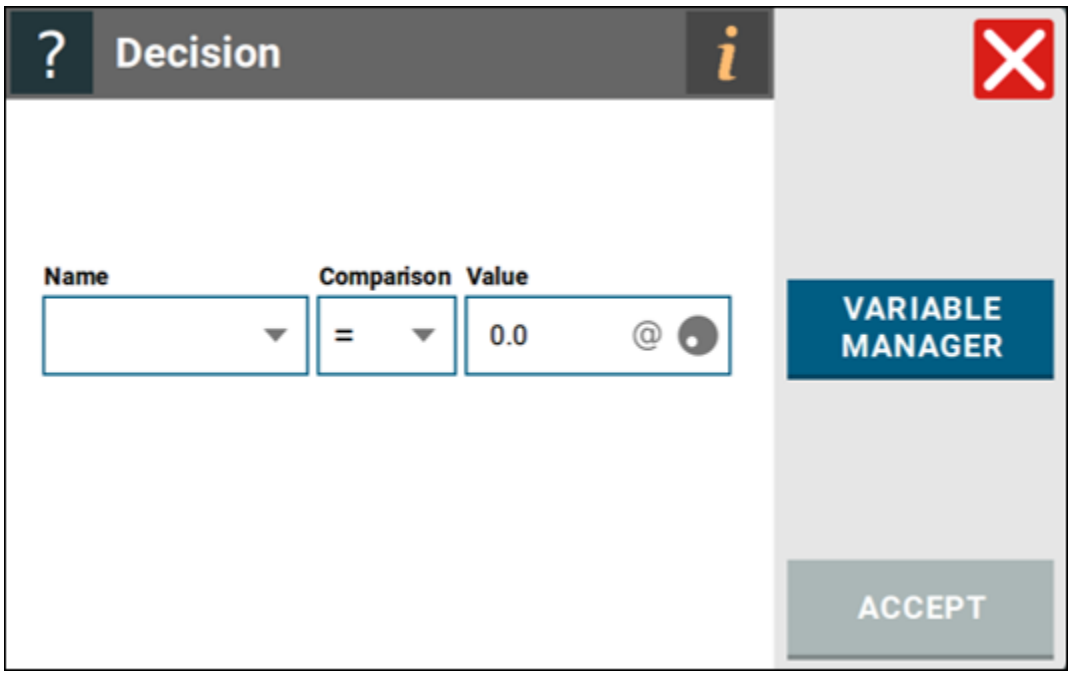
Decision block

The Decision block is used to compare a variable with a number or another variable. Comparisons can determine whether a value is greater than, less than, or equal to another value.

The Decision block follows the **Condition False** pathway when the comparison is false. It follows the **Condition True** pathway when the comparison value is true.



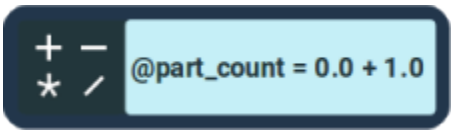
Below is the Decision generator which pops up when adding a Decision block.



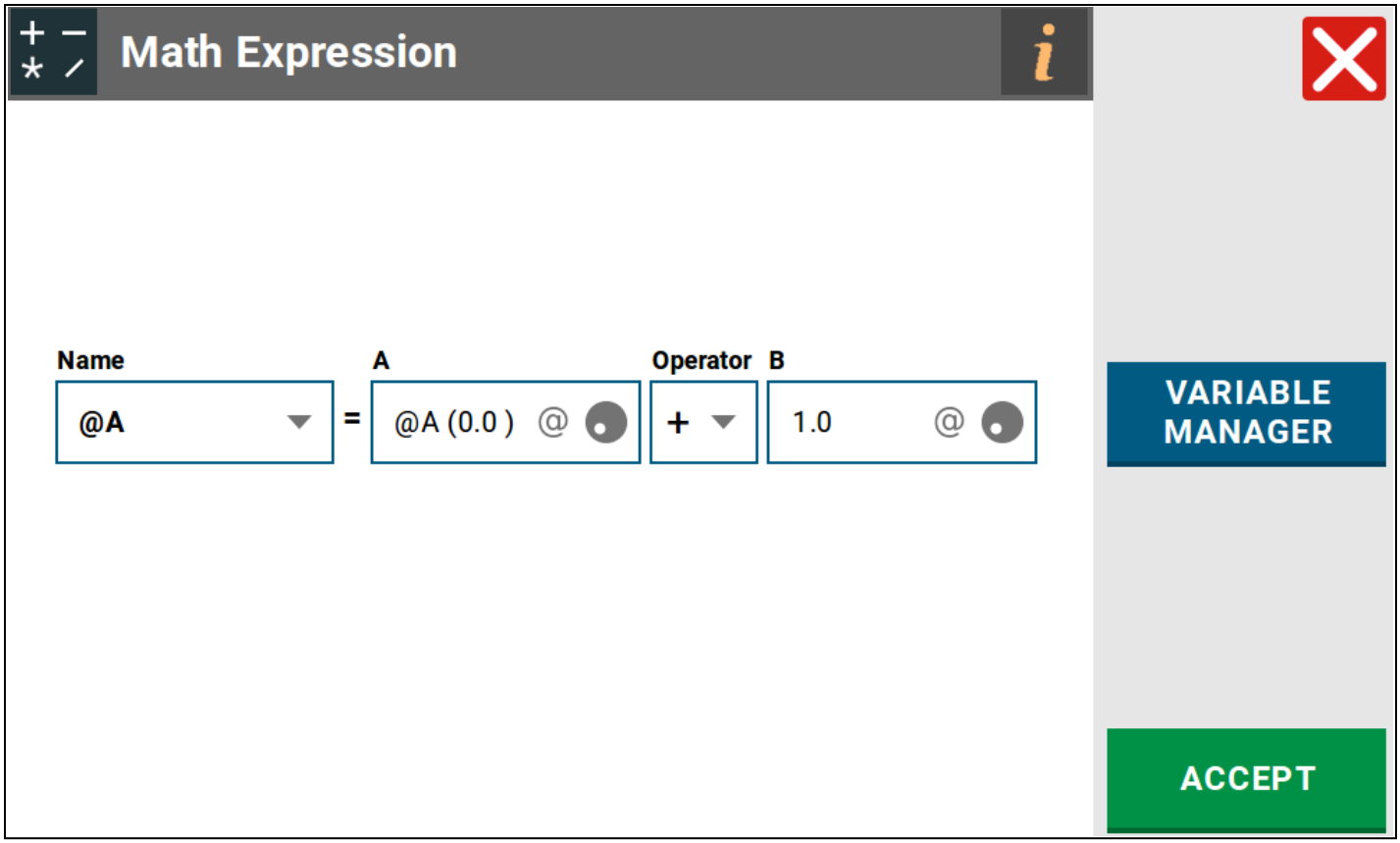
Setting	Description
Name	Select a variable that will be used in the comparison. ***Note:*** You cannot set a variable in this field or tap **ACCEPT** in this window if you have not defined any variables in your system that can be used in your task.
Comparison	Set the operator that will be used to compare the two values. Comparisons can be made using the following operators: ▪ (>) the value on the left is **greater than** the value on the right ▪ (<) the value on the left is **less than** the value on the right ▪ (=) the value on the left is **equal to** the value on the right
Value	Set the number or variable that you will compare the value in the field titled **Name** to. If you would like to set a variable in this field, tap **@**, otherwise tap the dial or anywhere else in the field to enter a number.
Variable Manager	Open the Variable Manager.

Math Expression block

The Math Expression block allows the user to execute mathematical expressions to set the values of variables. The Math Expression block is especially useful when defining variables that increment as a task runs.



Below is the Math Expression block generator which pops up after adding the math expression block to the task.

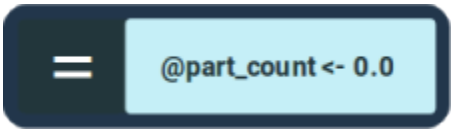


Setting	Description
Name	Select a variable that will be set to be equal to the output of the expression. ***Note:*** You cannot set a variable in this field or tap **ACCEPT** in this window if you have not defined any variables in your system that can be used in your task.
A	Set the number or variable that you will use as the value for **A** in the math expression. If you would like to set a variable in this field, tap **@**, otherwise tap the dial or anywhere else in the field to enter a number.
Operator	Set the operator that will be used to with the values for **A** and **B**. The following operators are available: ▪ (+) adds **A** to **B** ▪ (*) multiplies **A** by **B** ▪ (-) subtracts **B** from **A** ▪ (/) divides the **A** by **B**

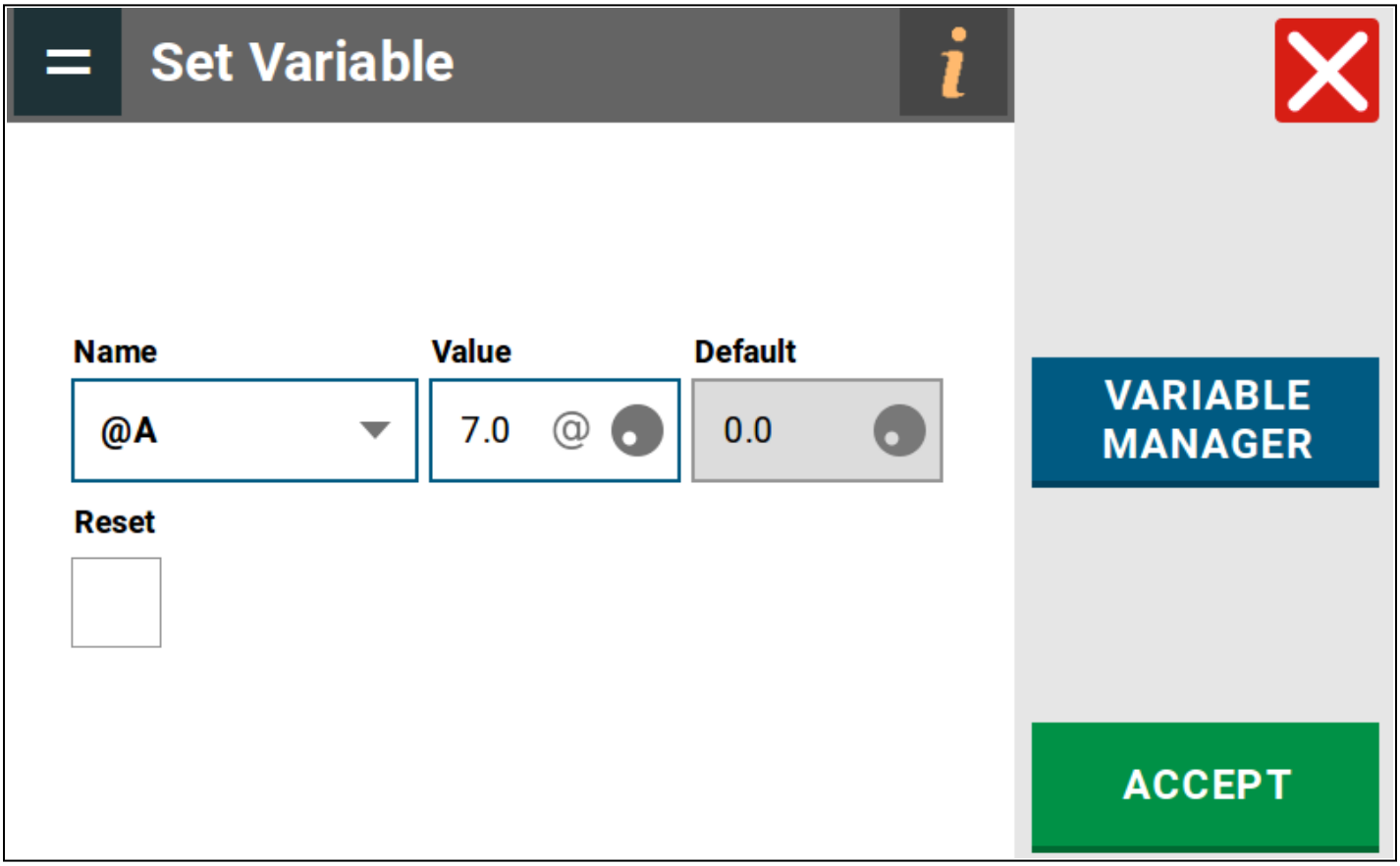
Setting	Description
B	Set the number or variable that you will use as the value for **B** in the math expression. If you would like to set a variable in this field, tap **@**, otherwise tap the dial or anywhere else in the field to enter a number.
Variable Manager	Open the Variable Manager.

Set Variable block

The Set Variable block is used to set the value of a variable. The value of a variable can be set to a number that you input or to the value of another variable. You can also reset the variable to its default value, which can be specified in the variable manager.



Below is the Set Variable generator which pops up when adding a Set Variable block.



Setting	Description
Name	Select a variable that will have its value set. ***Note:*** You cannot set a variable in this field or tap **ACCEPT** in this window if you have not defined any variables in your system that can be used in your task.
Value	Set the number or variable that you will set the variable in the **Name** field to. If you would like to set a variable in this field, tap **@**, otherwise tap the dial or anywhere else in the field to enter a number.
Default	This field cannot be edited in this window. It displays the default value of the variable in the **Name** field, which can be changed in the Variable Manager.
Reset	If this box is checked, the Set Variable block will reset a variable to its default value when it is ran. ***Note:*** If this box is checked, the **Value** field will gray out and become inactive.
Variable Manager	Open the Variable Manager.

Robot Moves blocks

The Robot Moves blocks control movement and force feedback of the robot arm. Moves are defined by the type of movement; such as moving to a specific waypoint or jogging in a defined direction. Each Robot Moves block section describes how that specific block moves and the setting that apply.

Waypoint based movements

A waypoint is an absolute location in space that is the same every time the robot arm moves to it. Waypoints are established by moving the arm to the location and saving the location. When creating a waypoint, the robot arm can be moved to the location in either Teach mode or by jogging the arm in Active mode. The important thing is that the location the arm is at when the waypoint saved is the location of the waypoint. Waypoints can be created through the Waypoint generator, which is accessible from any Robot Moves blocks that move based on waypoints; see a move block's section for information on the type of move performed.

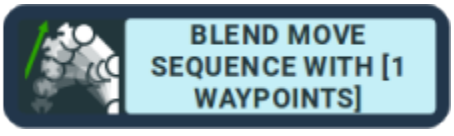
Waypoints are saved within the task they're created in. You can't reference a waypoint from another task in the current task. The only way to transfer a waypoint between tasks is to save the move containing the waypoint in a container and load that container in the new task.

Waypoints can be used multiple times in the same task. Reusing waypoints makes programming a task quicker and easier. When a waypoint that is used multiple times in the task is edited, the system prompts you to either update all usage of the waypoint or create a new waypoint. This prevents you from overwriting a waypoint used elsewhere in the task by mistake.

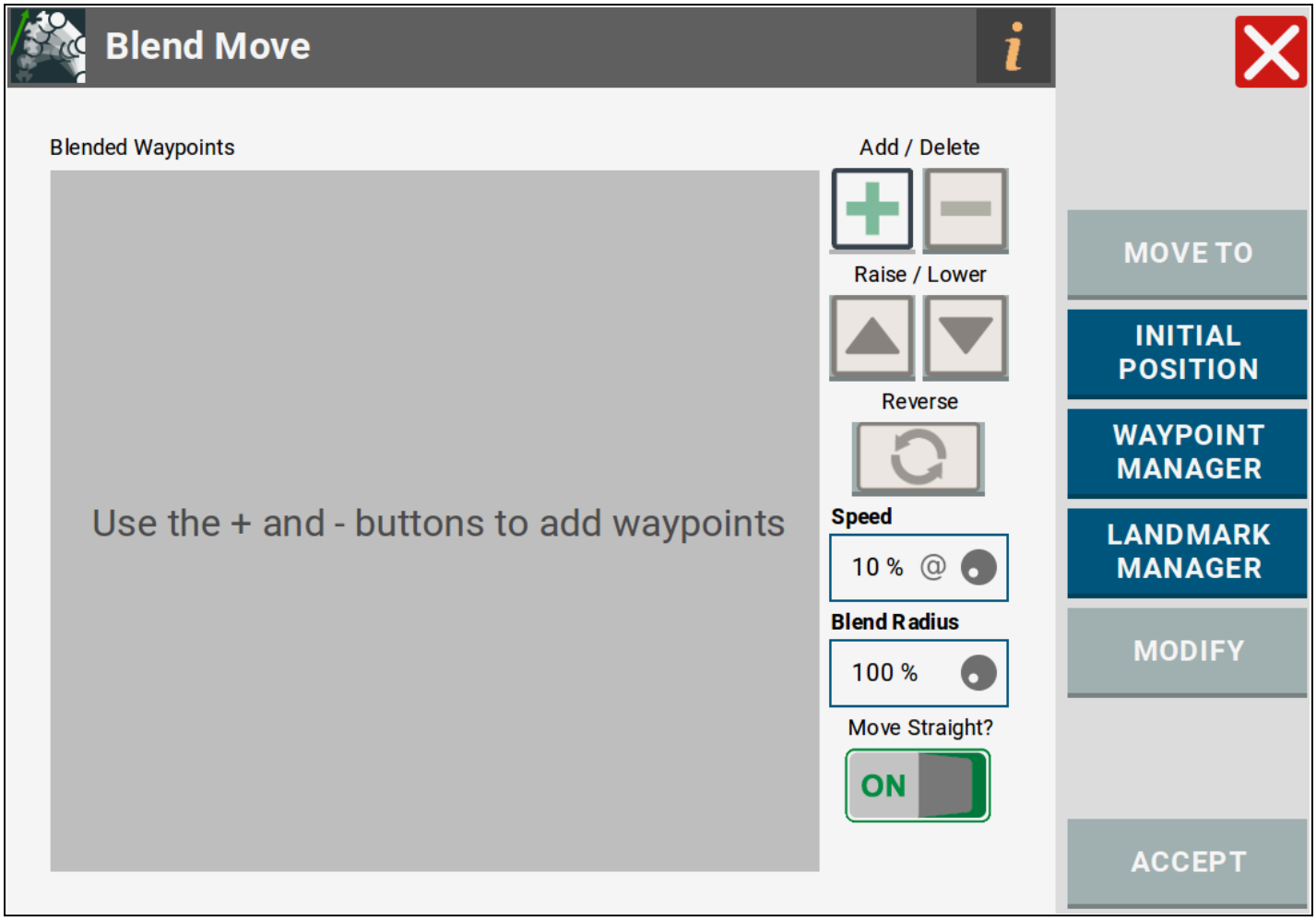
Blend block

The Blend block uses a series of waypoints that the robot arm moves to in a fluid, continuous motion. The benefit of a blend move is that it enables you to program a complex or winding motion without the robot arm pausing at each location along the path and without every waypoint requiring a separate Move block on the flowchart.

When executing a Blend block, the robot arm only reaches the exact position and orientation of the last waypoint. While moving, the arm travels close to the other waypoints in the blend but doesn't always reach them exactly. To ensure the arm moves through each waypoint's exact position and location, you can set the Blend Radius to zero.



Below is the Blend Move generator that pops up when the adding a Blend block.



Setting	Description
Blended Waypoints	List of the waypoints that are a part of the blend.
Add/Delete	Add or delete waypoints from the list of waypoints that make up the blend. When adding a waypoint, you are provided the option to create a new waypoint or select from existing waypoints; see the "Waypoint Based Movements" section for information on creating a new waypoint. Deleting a waypoint from the blend doesn't delete the waypoint from the task. To delete the waypoint from the task, use the Waypoint Manager.
Raise/Lower	Raise or lower the selected waypoint in the blend order.
Reverse	Reverse the order the Blend block moves to the waypoints.

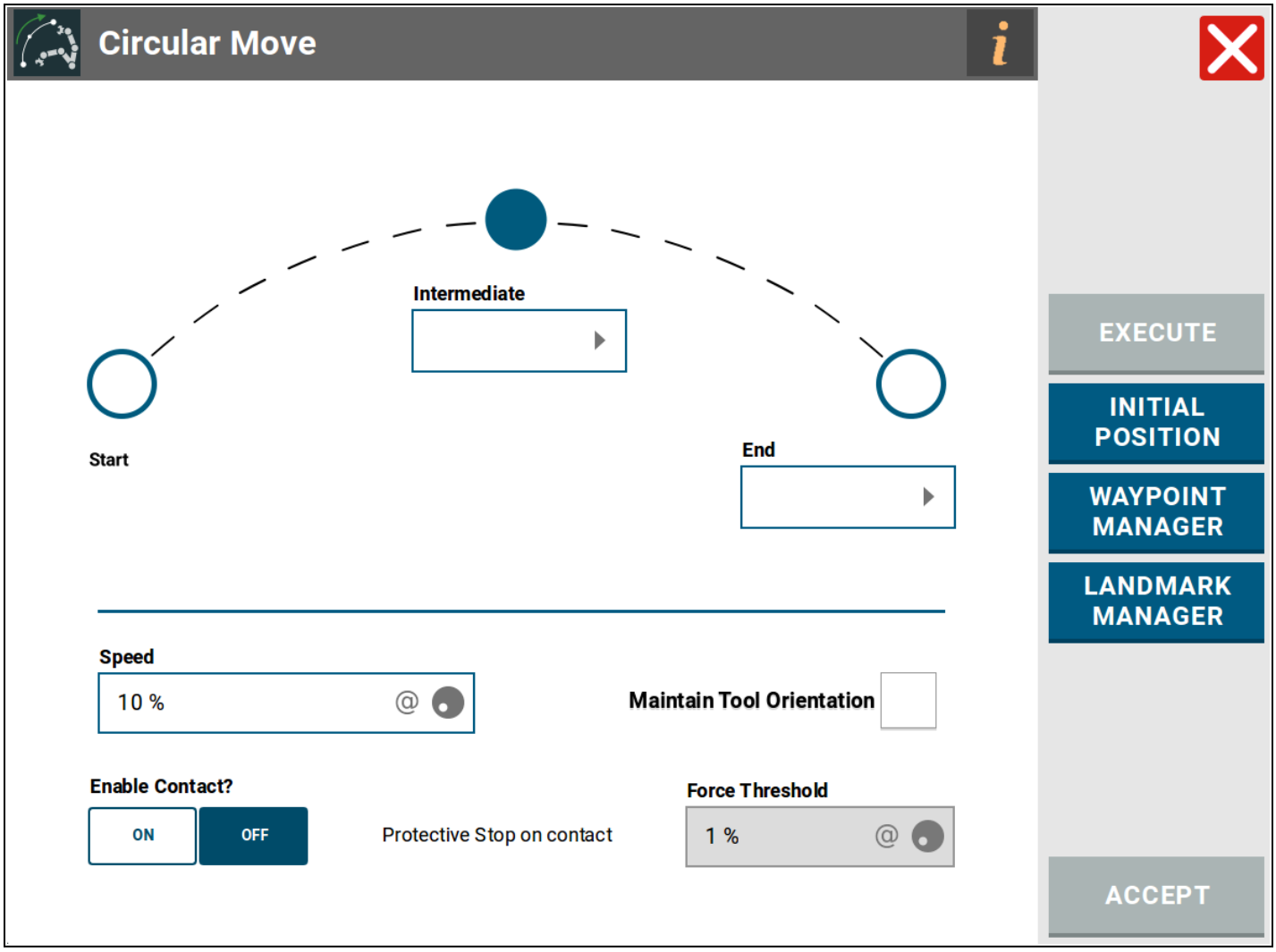
Setting	Description
Speed	Set the speed of the robot arm as it executes the blend.
Blend Radius	Adjust how smooth the motion of the robot arm is as it executes the blend.
Move Straight	Toggle how straight the robot arm's path is to the next waypoint in the blend. When on, the robot arm rotates as many joints as needed to move between the waypoints in a straight line. When off, the robot arm travels between waypoints in curved line, moving as few joints as needed.
Move To	Move the robot arm to the location of the selected waypoint.
Initial Position	Move the robot arm to the position it was in before the New Waypoint generator opened. This feature is used when the new waypoint's location is dependent on the previous waypoint's location, but the robot arm's position has changed since opening the generator.
Waypoint Manager	Open the Waypoint Manager.
Landmark Manager	Open the Landmark Manager.
Modify	Open the Modify Waypoint generator.

Circular Move Block

A circular move is a motion that the robot arm executes using three points and a constant radius. The arm will start at the current pose when the block executes and move through the Intermediate waypoint to the End waypoint in a circular arc no greater than 360 degrees. The user can instruct the arm to respond to force feedback during the motion as well as orient the TCP to the start or the end point of the motion.

Below is the Circular Move generator which pops up when adding a Circular Move block.

Note: When creating a circular move, make sure that the motion before the circular move leaves the arm in the pose you wish to use as the Start position.



Setting	Description
Intermediate	Set the intermediate waypoint which is a pose that the arm will move through between the start and end poses.
End	Set the final waypoint which is the pose where the arm will complete the motion.
Speed	Set the speed of the robot arm.
Enable Contact	Toggle whether the robot arm will check for contact during the motion. The robot arm will respond to applied force in any direction, regardless of the direction

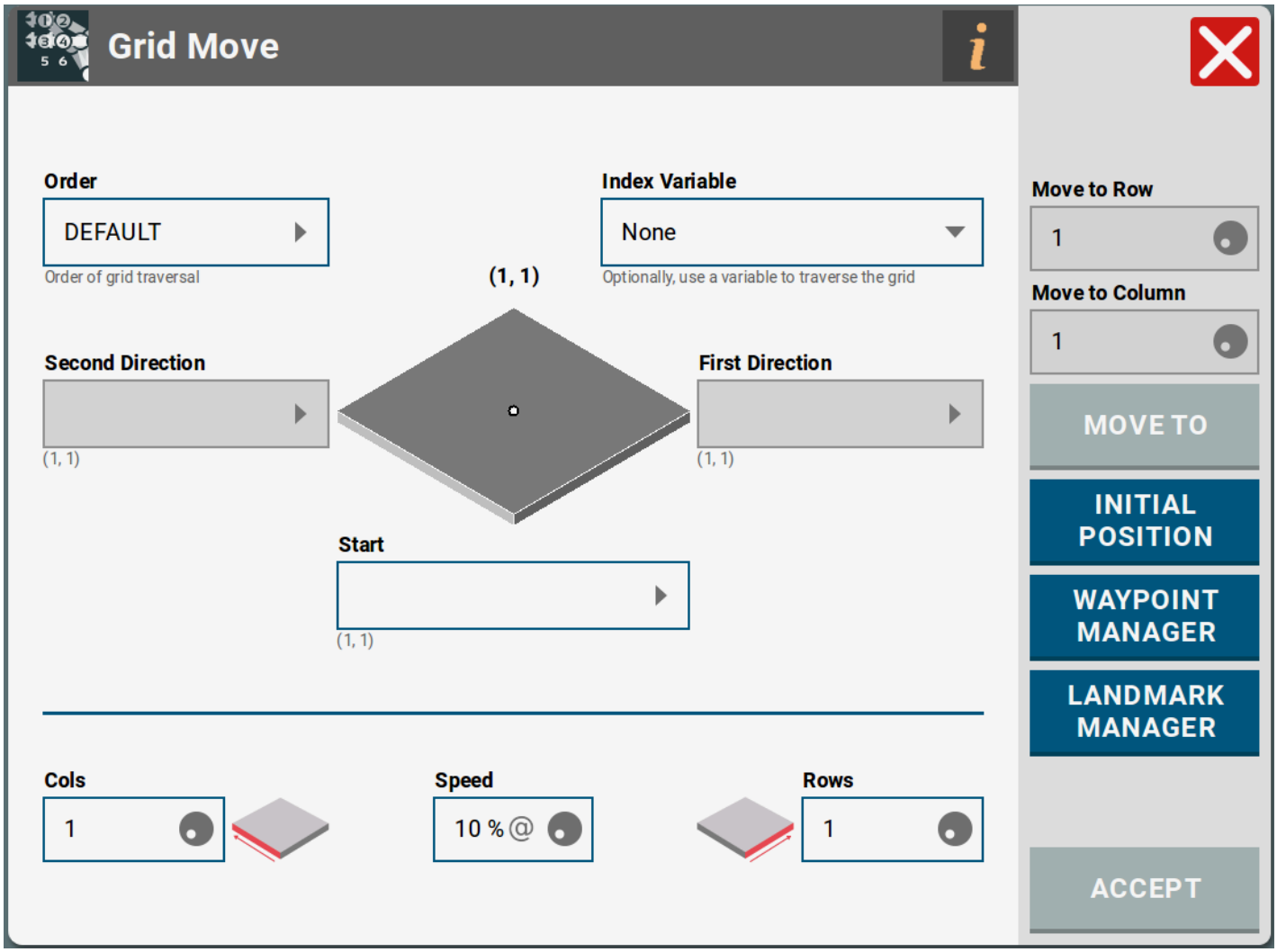
Setting	Description
Force Threshold	Set the amount of force the robot arm must experience to move along the **FORCE OBSERVED DURING MOTION** pathway. This setting is only available when Enable Contact is on.
Maintain Tool Orientation	Determine how the robot arm will orient the TCP during the motion. When unchecked, the orientation of the TCP will steadily rotate throughout the Circular move to match the orientation of the TCP at the end waypoint. When checked, the TCP will remain in the orientation that it began in throughout the Circular move.
Execute	Perform the programmed circular motion. Note that this button does not know the intended Start position and will execute the motion from the pose of the arm when it is pressed.
Initial Position	Move the robot arm to the pose it was in when the generator was opened.
Waypoint Manager	Open the Waypoint Manager.
Landmark Manager	Open the Landmark Manager.

Grid block

The Grid block creates a grid based on the boundaries and number of rows and columns you provide. For a Grid block, you provide three waypoints that represent three the corners of the grid and then the number of locations along each axis defined by the waypoints, the number of rows and columns. The system uses the data to create a series of waypoints that represent every location in the grid. Each time the Grid block executes, the robot arm moves to the next location in the grid, one at a time. After the robot arm has moved to each location, it starts over at the beginning. Grid blocks are useful for programming tasks to pick or place workpieces in a grid pattern.

When programming a Grid block to pick parts, set the waypoints at the location on the workpiece you want the end effector to grip.

Below is the Grid Move generator that pops up when the adding a Grid block



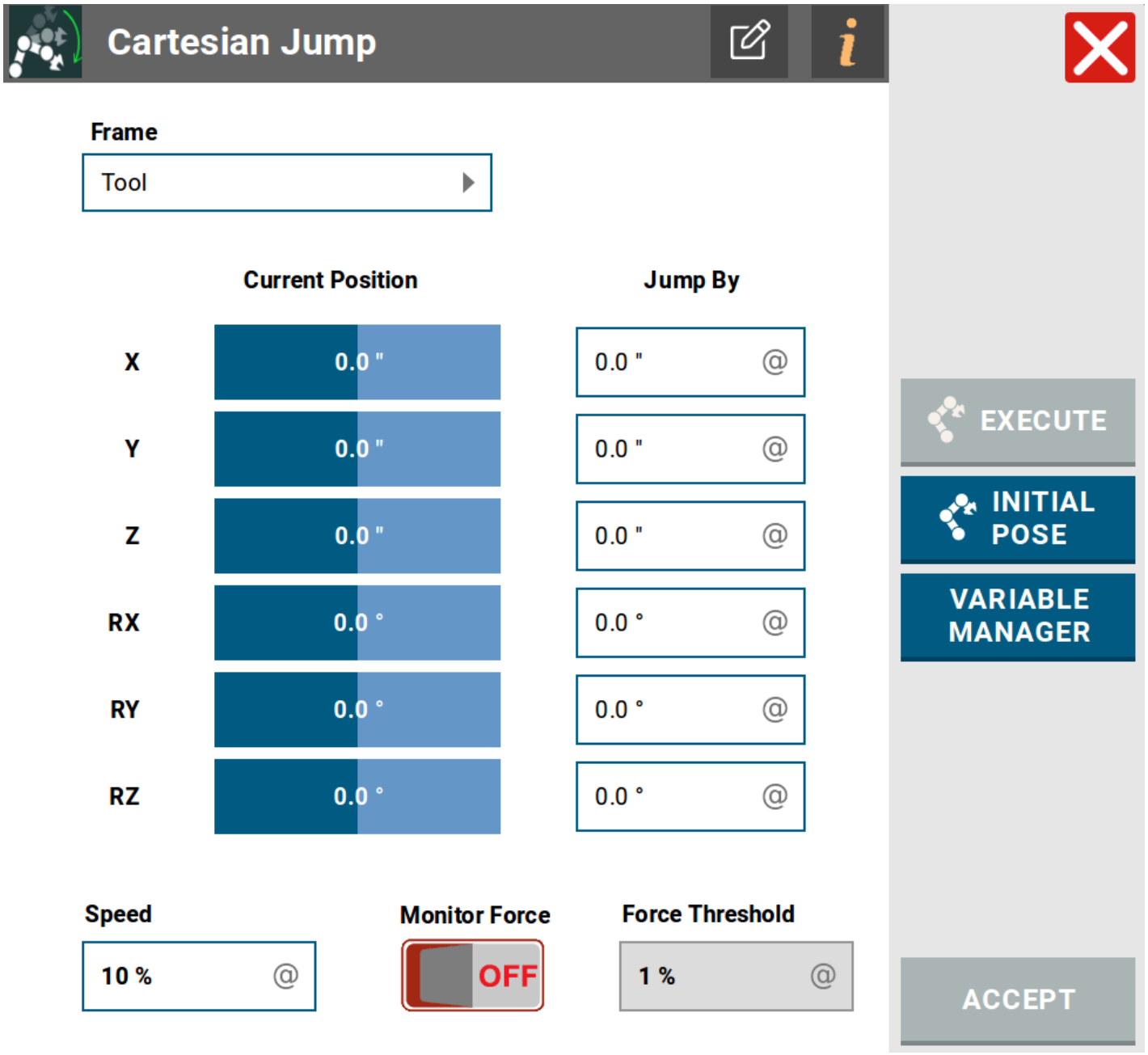
Setting	Description
Start	Select an existing waypoint or create a new waypoint that represents the start position of the grid.
First Direction	Select an existing waypoint or create a new waypoint that represents the end of the first row.
Second Direction	Select an existing waypoint or create a new waypoint that represents the end of the first column.
Order	Open the Grid Traversal Order window.
Cols	Set the number of columns in the grid.

Setting	Description
Speed	Set the speed of the robot arm as it executes the blend.
Row	Set the number of rows in the grid.
Move to Row	Moves the robot arm to the location of the provided row and column coordinates.
Move to Column	
Move To	
Initial Position	Moves the robot arm to the position it was in before the settings window opened. This feature is used when the new waypoint's location is dependent on the previous waypoint's location, but the robot arm's position has changed since opening the generator.
Waypoint Manager	Open the Waypoint Manager.
Landmark Manager	Open the Landmark Manager.

Cartesian Jump block

The Cartesian Jump block executes a motion along or around a primary axis. Cartesian Jump motions are always relative to the robot's location when the block executes. For example, if a block containing a Cartesian Jump command tells the arm to move three inches in the +Z direction, the arm will always perform this motion regardless of what waypoint it started at. The difference between a waypoint and a Cartesian Jump motion is that waypoints are specific locations the robot arm always moves to whereas Cartesian Jumps always execute the same motion.

Below is the Cartesian Jump generator which pops up when adding a Cartesian Jump block.



Setting	Description
Frame	Set the Frame that the robot will move in reference to.
Current Position	Indicates the distance of the TCP from the Reference Frame.

Setting	Description
Jump By	Set the distance, primary axis, and direction (positive or negative) in which the robot arm moves. ***Note:*** The Cartesian Jump axes may be different than those pictured if a tool offset is set in Hardware Configuration.
Distance	Set the distance of the Cartesian Jump.
Enable Contact	Toggle whether Forge/OS should check for contact during the Cartesian Jump. When off, the robot arm executes the movement for the entire distance then proceed along its **CARTESIAN JUMP MOTION COMPLETED** pathway. If force is detected, the system enters a protective stop. When on, the robot arm executes the motion until force is experienced or until it has moved the specified distance. The force must be in the opposite direction of the motion for it to register. Once force is experienced, the Cartesian Jump block proceeds along its **FORCE OBSERVED DURING MOTION** pathway.
Speed	Set the speed of the robot arm as it executes the blend. This setting is only available when Enable Contact is off.
Force Threshold	Set the amount of force the robot arm must experience. This setting is only available when Enable Contact is on.
Execute	Perform the Cartesian Jump motion with the current settings.
Initial Pose	Move the robot arm to the position it was in before the settings window opened. This feature is used when the new waypoint's location is dependent on the previous waypoint's location, but the robot arm's position has changed since opening the generator.
Variable Manager	Open the Variable Manager.

Joint Jump block

The Joint Jump block is located in the Joint Jump submenu of the block menu. It is used to move the robot's joints a specified amount. Since this robot movement is joint-based, it is useful for moving to or through robot positions that would otherwise result in singularities.

Below is the Joint Jump generator which pops up when adding a Joint Jump block.

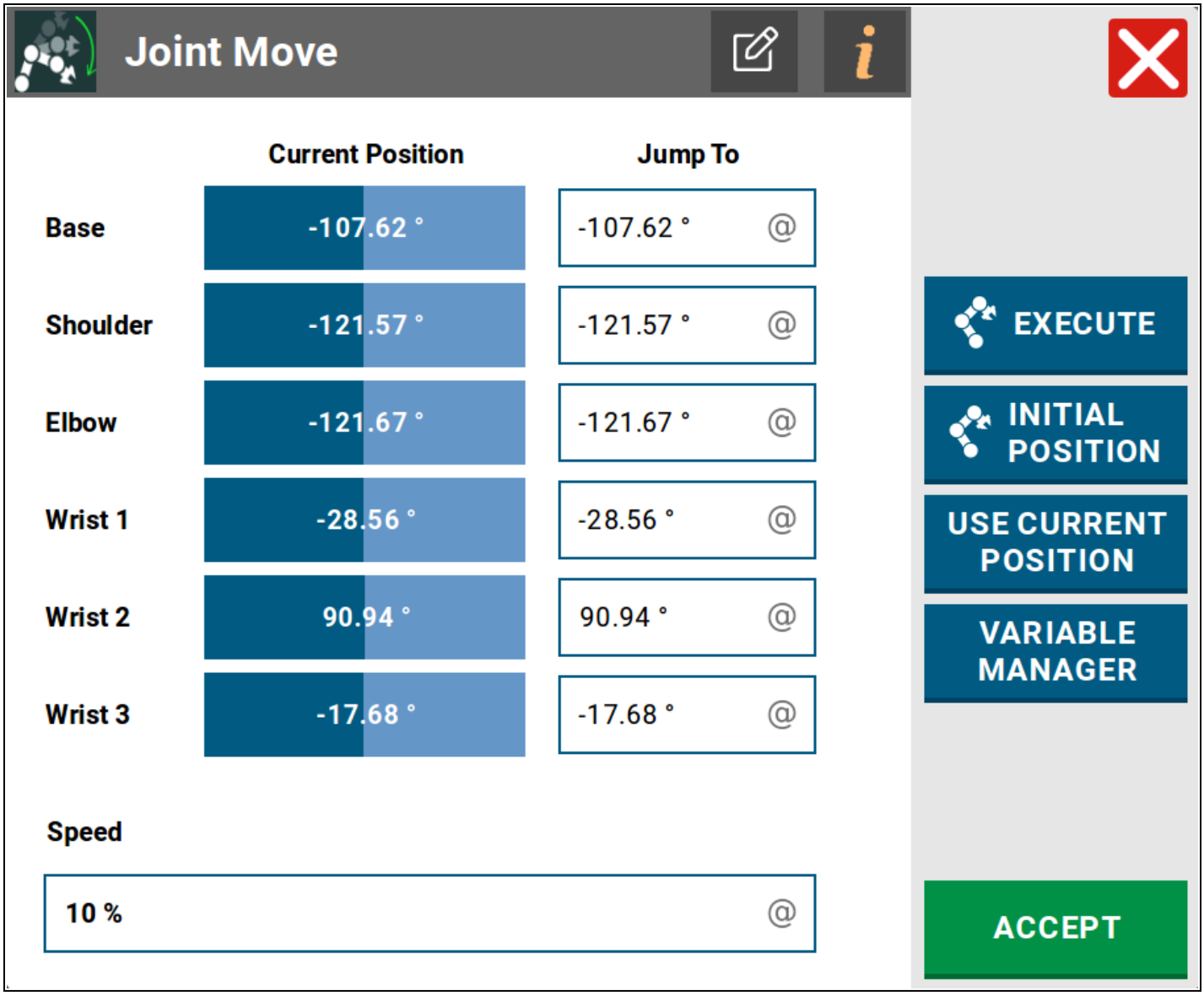
Setting	Description
Current Position	Indicates the current position of each of the robot's joints.
Jump By	Set the amount by which each joint will jump (degrees). The robot will move its joints by this much when the block is executed.

Setting	Description
Execute	Perform the robot move motion by the amounts specified in the fields for **Jump By**.
Initial Position	Move the robot arm to the position it was in when the generator opened was.
Variable Manager	Open the Variable Manager.

Joint Move block

The Joint Move block is located in the Move submenu of the block menu. It is used to move the robot to a position defined by the rotation of the robot's joints. Since this robot movement is joint-based, it is useful for moving to or through robot positions that would otherwise result in singularities.

Below is the Joint Move generator which pops up when adding an Joint Move block.



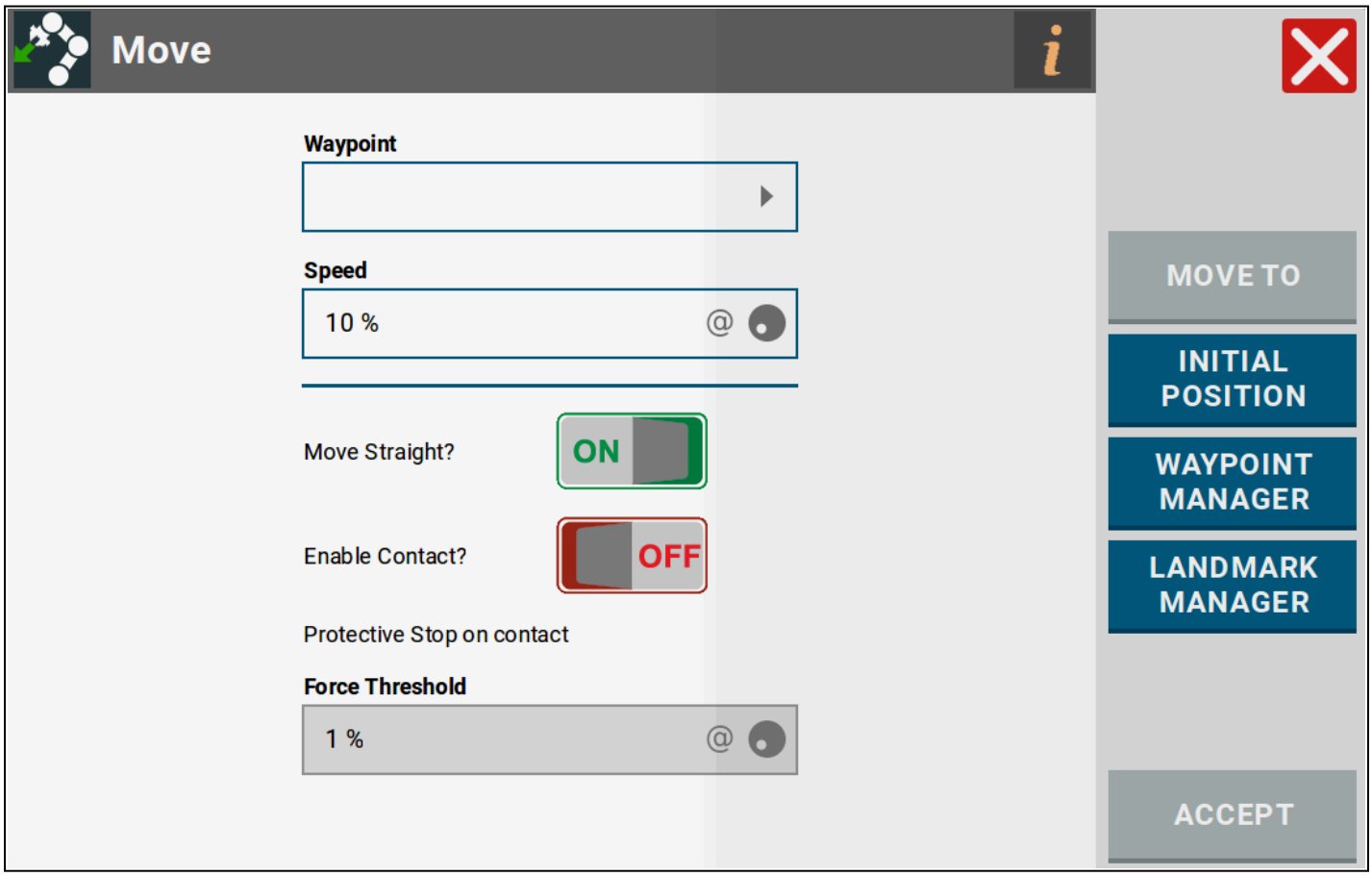
Setting	Description
Current Position	Indicates the current position of each of the robot's joints.
Jump To	Set the position of each joint. The robot will move its joints into these positions when the block is executed. When the block is first created, the values for each joint will default to the current position of the robot.
Speed	Set the speed of the robot arm as it executes the joint jump.
Execute	Perfom the robot move motion to the joint position specified in the fields for **Jump To**.

Setting	Description
Initial Position	Move the robot arm to the position it was in when the generator opened was.
Variable Manager	Open the Variable Manager.

Move block

The Move block is a single motion from the robot arm's current location to a defined waypoint. It is labeled Waypoint in the Move submenu of the Block Menu.

Below is the Move generator which pops up when adding a Move block.



Setting	Description
Waypoint	Select an existing waypoint or create a new waypoint. The waypoint used is where the robot arm will travel to when the Move executes.
Speed	Set the speed of the robot arm as it executes the move.
Move Straight	Toggle how straight the robot arm's path is to the next waypoint in the blend. When on, the robot arm rotates as many joints as needed to move between the waypoints in a straight line. When off, the robot arm travels between waypoints in curved line, moving as few joints as needed.
Enable Contact	Toggle whether Forge/OS should check for contact during the move. When off, the robot arm moves to the waypoint then proceed along its **MOTION TO TCP POSE COMPLETE** pathway. If force is detected, the system enters a protective stop. When on, the robot arm checks for force while moving to the waypoint. Once force is experienced, the Move block proceeds along its **FORCE OBSERVED DURING MOVE** pathway.
Force Threshold	Set the amount of force the robot arm must experience while executing the Move. This setting is only available when Enable Contact is on.
Move To	Move the robot arm to the location of the selected waypoint.
Initial Position	Move the robot arm to the location it was at when the generator opened was.
Waypoint Manager	Open the Waypoint Manager.
Landmark Manager	Open the Landmark Manager.

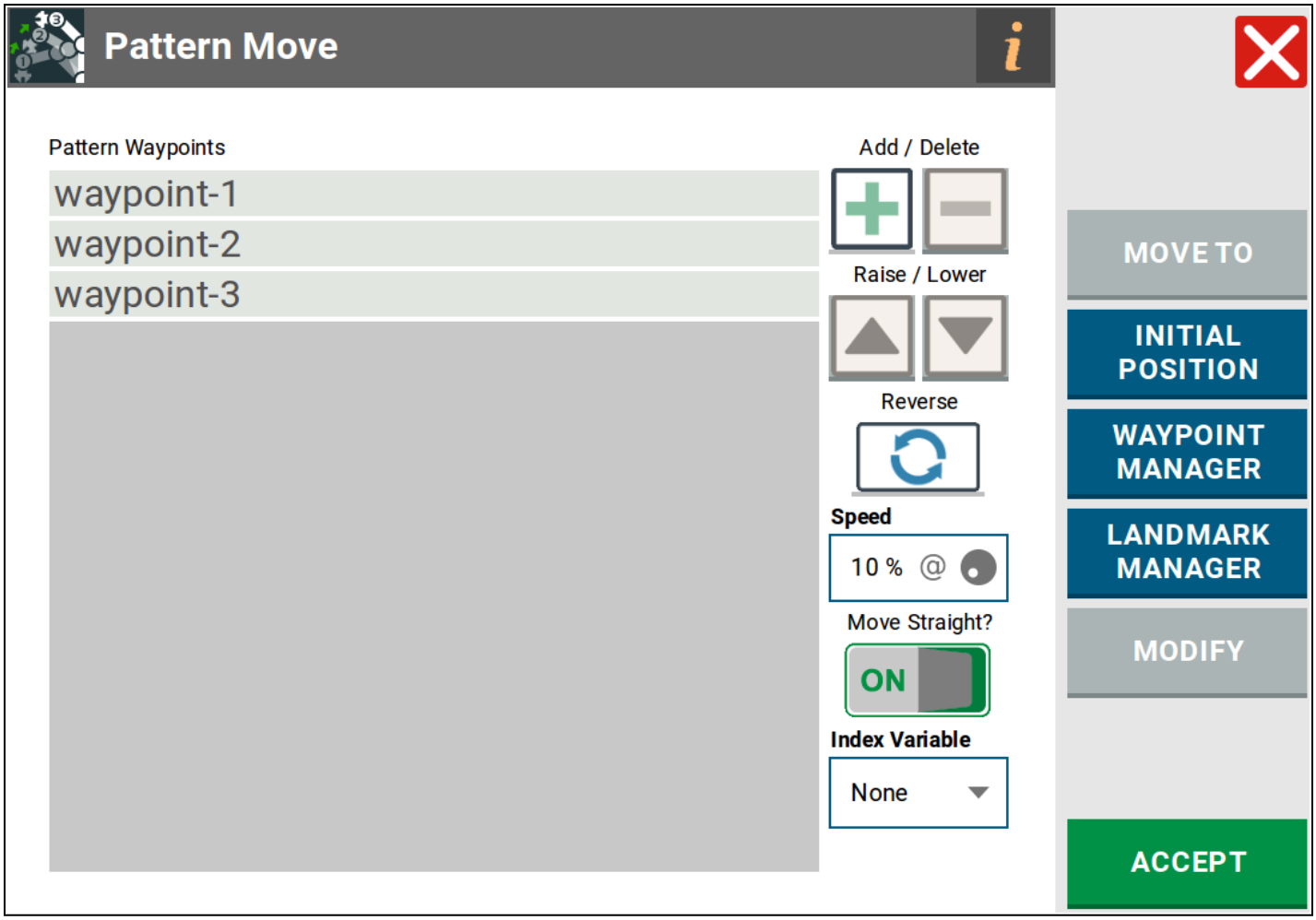
Pattern block

The Pattern block creates a list of waypoints the robot arm is to travel to. Each time the Pattern block executes, the robot arm moves to the next waypoint in the list, one at a time. After the robot arm has moved to each waypoint, it starts over at the beginning.

While the Pattern settings window is open, the TeachMate's rapid teach function can be used to add the location of the robot arm as a waypoint to the Pattern Waypoints list.

- Press the 02 button to save the location.
- Long press the 02 button to save the Pattern and add the Pattern block to the task.

Below is the Pattern generator which pops up when adding a Pattern block.



Setting	Description
Pattern Waypoints	List of the waypoints that are a part of the pattern.

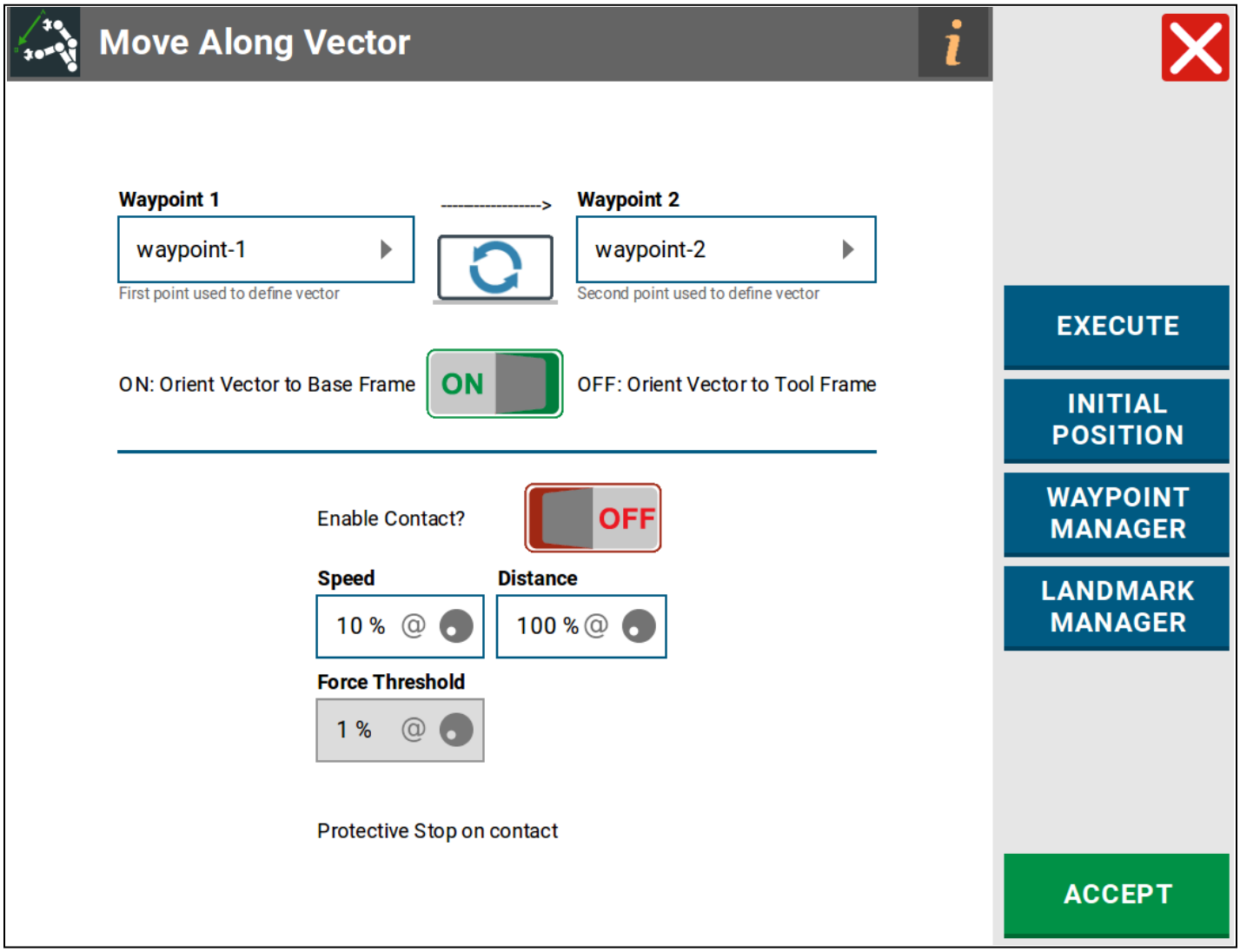
Setting	Description
Add/Delete	Add or delete waypoints from the list of waypoints that make up the pattern. When adding a waypoint, you are provided the option to create a new waypoint or select from existing waypoints; see the "Waypoint Based Movements" section for information on creating a new waypoint. Deleting a waypoint from the pattern doesn't delete the waypoint from the task. To delete the waypoint from the task, use the Waypoint Manager.
Raise/Lower	Raise or lower the selected waypoint in the pattern order.
Reverse	Reverse the order the Pattern block moves to the waypoints.
Speed	Set the speed of the robot arm as it executes the pattern.
Move Straight	Toggle how straight the robot arm's path is to the next waypoint in the blend. When on, the robot arm rotates as many joints as needed to move between the waypoints in a straight line. When off, the robot arm travels between waypoints in curved line, moving as few joints as needed.
Move To	Move the robot arm to the location of the selected waypoint.
Initial Position	Move the robot arm to the position it was in before the New Waypoint generator opened. This feature is used when the new waypoint's location is dependent on the previous waypoint's location, but the robot arm's position has changed since opening the generator.
Waypoint Manager	Open the Waypoint Manager.
Landmark Manager	Open the Landmark Manager.
Modify	Open the Modify Waypoint generator.

Vector block

The Vector block moves the robot arm in a straight vector that you create by defining the path and distance. A vector is a motion in any direction. Like a Cartesian Jump motion, a vector is the same motion every time it executes, but unlike a Cartesian Jump motion, a vector doesn't have to be along or around a primary axis.

In a Vector block, you set two waypoints that represent the start and end location of the vector. Once the vector is created, the motion of the vector is recreated any time the block executes, regardless of where the robot arm is. The waypoints still exist in the saved waypoints for the task, but the Vector block expresses the motion between the waypoints, not the location of the waypoints.

Below is the Vector generator which pops up when adding a Vector block.



Setting	Description
Waypoint 1	Select an existing waypoint or create a new waypoint that represents the beginning of the vector; see the "Waypoint Based Movements" section for information on creating a new waypoint.
Waypoint 2	Select an existing waypoint or create a new waypoint that represents the end of the vector; see the "Waypoint Based Movements" section for information on creating a new waypoint.
Reverse	Reverse the order the Vector block moves.
Orient Vector	Toggle orienting the vector between the base frame and the tool frame. When on, the robot arm executes the vector motion in the same direction, regardless of what direction the tool is facing. When off, the vector's direction changes based on the orientation of the tool. **Example:** When on, if the vector is straight down, the vector is always straight down. The end effector's orientation doesn't matter. When off, the path of the vector changes with the orientation of the end effector. If the end effector is upright, the vector moves vertically. If the end effector is rotated to the right, the vector path is rotated to the right too.
Enable Contact	Toggle whether Forge/OS should check for contact during the vector. When off, the robot arm moves the distance then proceed along its **MOTION COMPLETE** pathway. If force is detected, the system enters a protective stop. When on, the robot arm checks for force while moving the distance. Once force is experienced, the Vector block proceeds along its **FORCE OBSERVED DURING MOTION** pathway.
Speed	Set the speed of the robot arm as it executes the vector. This setting is only available when Enable Contact is off.
Force Threshold	Set the amount of force the robot arm must while executing the Vector. This setting is only available when Enable Contact is on.
Distance	Set the percentage of the Vector's distance the arm is to travel.
Execute	Perform the vector motion with the current settings.

Setting	Description
Initial Position	Moves the robot arm to the position it was in before the Vector settings opened. This feature is used when the new waypoint's location is dependent on the previous waypoint's location, but the robot arm's position has changed since opening the generator.
Waypoint Manager	Open the Waypoint Manager.
Landmark Manager	Open the Landmark Manager.
Modify	Open the Modify Waypoint generator.

Tools blocks

Tools blocks provides you control options for both the end effector as well as devices connected to the system. The available Tools blocks depend on the end effectors and devices you set up in Hardware Configuration; see the "Hardware Configuration" chapter for information on setting up end effectors and devices.

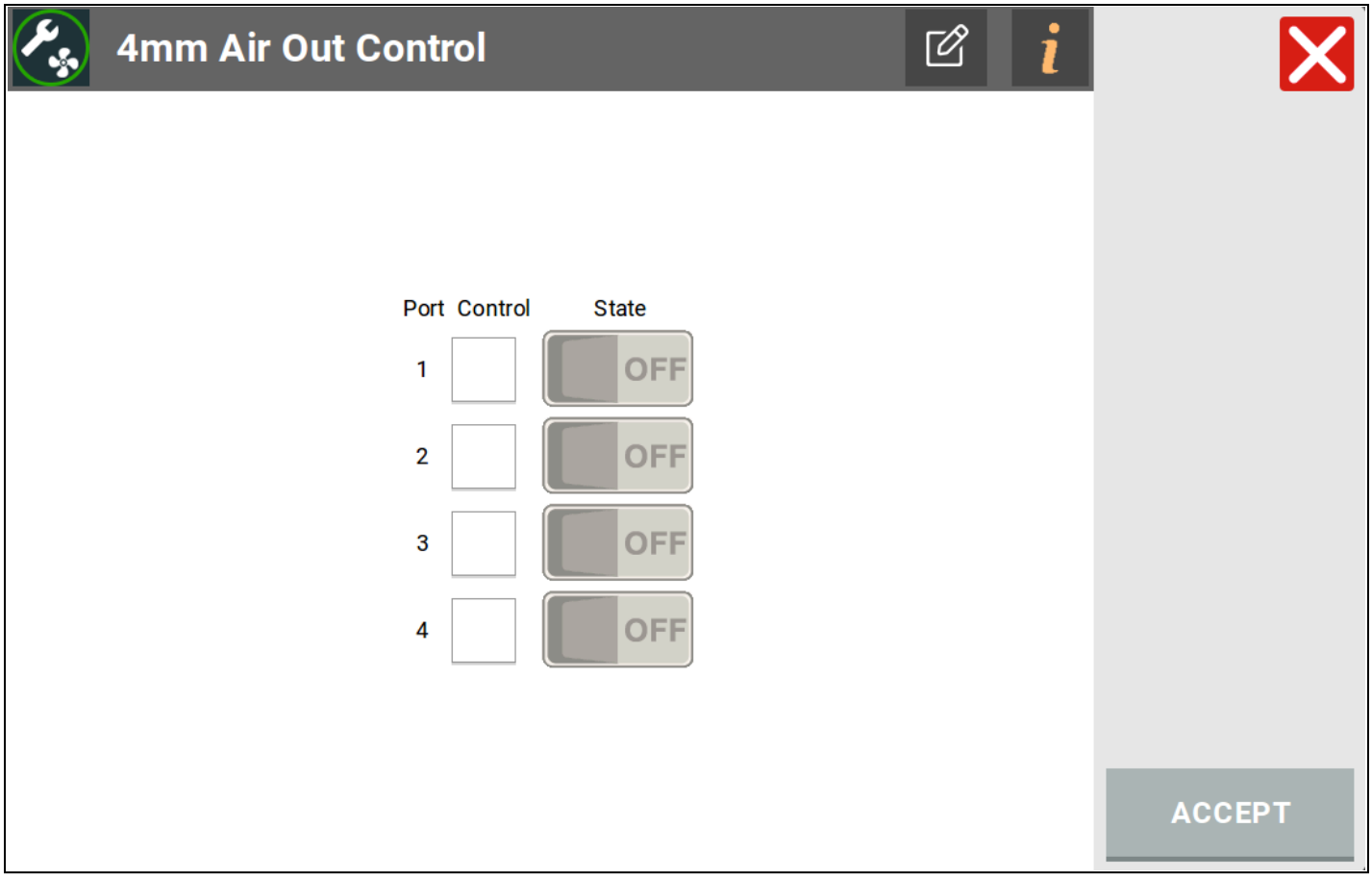
Note: Tools blocks become ghost blocks if the configuration changed while the task is open. The task cannot be executed while there are ghost blocks. If you've changed the configuration of your devices or end effector, be sure to update any affected blocks in the task; see the "Ghost blocks" section of the "The Canvas" chapter for more information about ghost blocks.

4mm Air Out Control block *(Forge/Ctrl only)*

The 4mm Air block is a dynamic control for the 4mm air outlets on the Forge/Ctrl. The available options change depending on the selected end effector in Hardware Configuration. If a pneumatic gripper is selected as the end effector in Hardware Configuration, the air outlets reserved for the gripper aren't available in the block's settings.

Upon execution of an 4mm Air block, the air outlet selected in the block's settings takes the indicated state. The air outlet holds the state until another 4mm Air block switches the air outlet into a different state. The 4mm Air block only changes the state of selected outlets. If an outlet is off and an 4mm Air block executes that doesn't have control over that outlet, the outlet holds its state.

Below is the 4mm Air Control generator which pops up when adding a 4mm Air block.

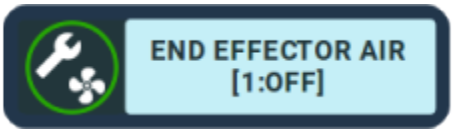


Setting	Descriptions
Control	Select the 4mm air outlet (port) on the Forge/Ctrl to control. ***Note:*** The 4mm Air block only changes the state of selected outlets. If an outlet is off and a 4mm Air block executes that doesn't have control over that outlet, the outlet holds its state.
State	Toggle the state of the air outlet (port) on or off.

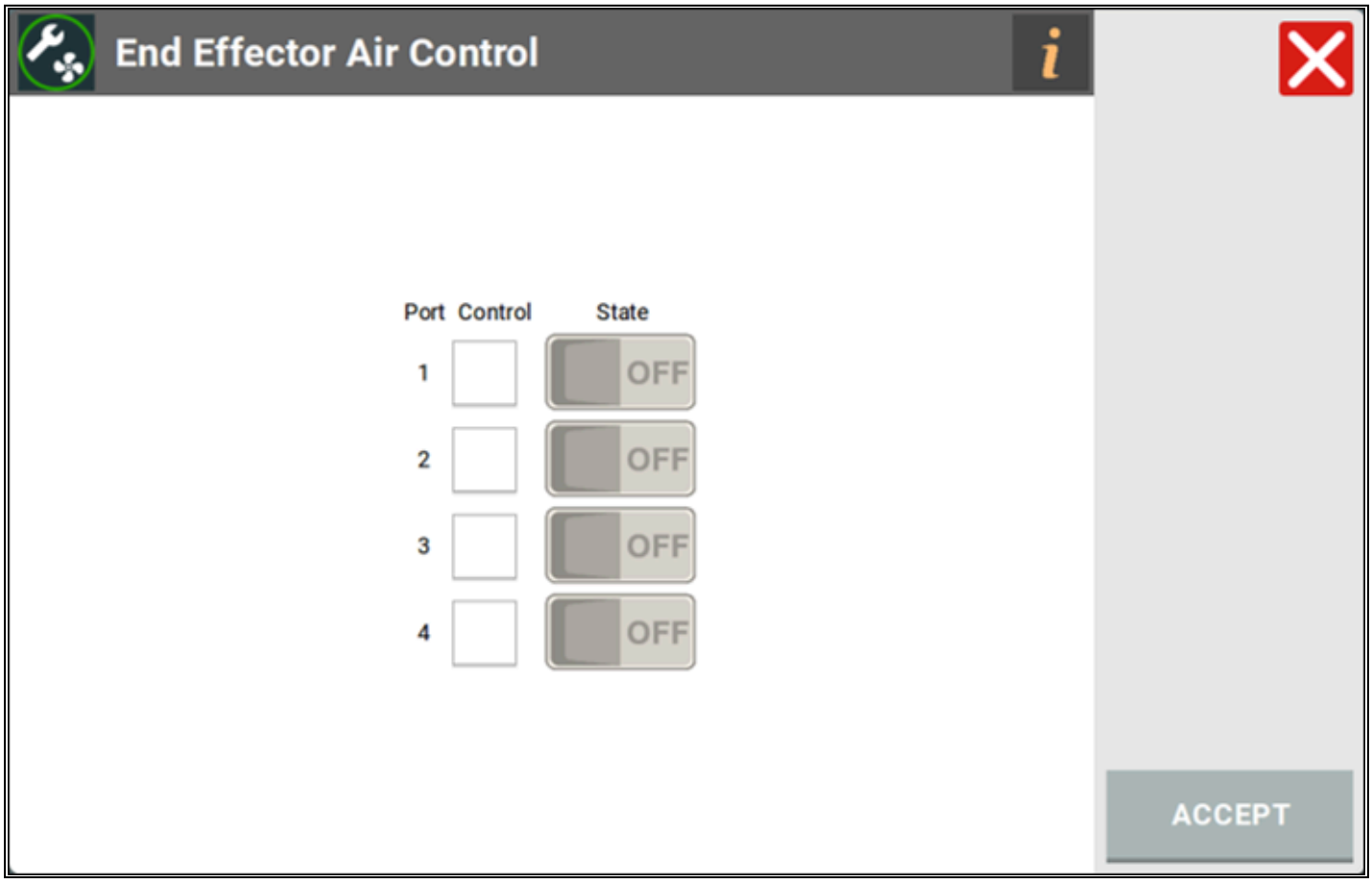
End Effector Air Out Control block *(Forge/Station only)*

The End Effector Air block is a dynamic control for the outlets on wrist 3 of the Forge/Station. The available options change depending on the selected end effector in Hardware Configuration. If a pneumatic gripper is selected as the end effector in Hardware Configuration, the air outlets reserved for the gripper aren't available in the block's settings.

Upon execution of an End Effector Air block, the air outlet selected in the block's settings takes the indicated state. The air outlet holds the state until another End Effector Air block switches the air outlet into a different state. The End Effector Air block only changes the state of selected outlets. If an outlet is off and an End Effector Air block executes that doesn't have control over that outlet, the outlet holds its state.



Below is the End Effector Air Control generator which pops up when adding a 4mm Air block.



Setting	Descriptions
Control	Select the End Effector air outlet (port) on the Forge/Ctrl to control. ***Note:*** The End Effector Air block only changes the state of selected outlets. If an outlet is off and an End Effector Air block executes that doesn't have control over that outlet, the outlet holds its state.

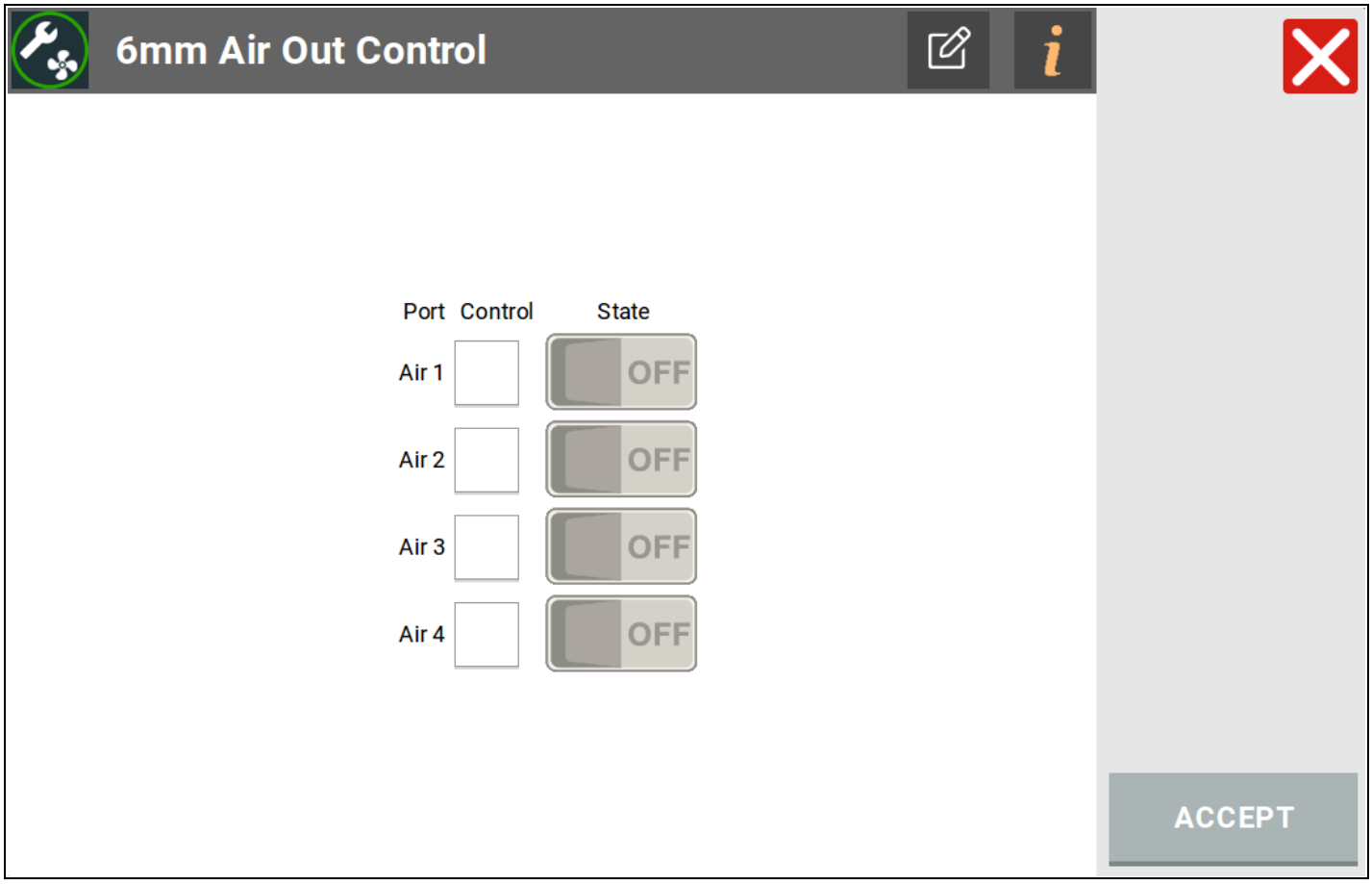
Setting	Descriptions
State	Toggle the state of the air outlet (port) on or off.

6mm Air Out Control block *(Forge/Ctrl only)*

The 6mm Air block is a dynamic control for the air outlets on the side of the Forge system. The available options change depending on the selected devices in Hardware Configuration. If a pneumatic device is selected as a device in Hardware Configuration, the air outlets reserved for the device aren't available in the block's settings.

Upon execution of a 6mm Air block, the air outlet selected in the block's settings takes the indicated state. The air outlet holds the state until another 6mm Air block switches the air outlet into a different state. The 6mm Air block only changes the state of selected outlets. If an outlet is off and a 6mm Air block executes that doesn't have control over that outlet, the outlet holds its state.

Below is the 6mm Air Control generator that pops up when the adding a 6mm Air block.



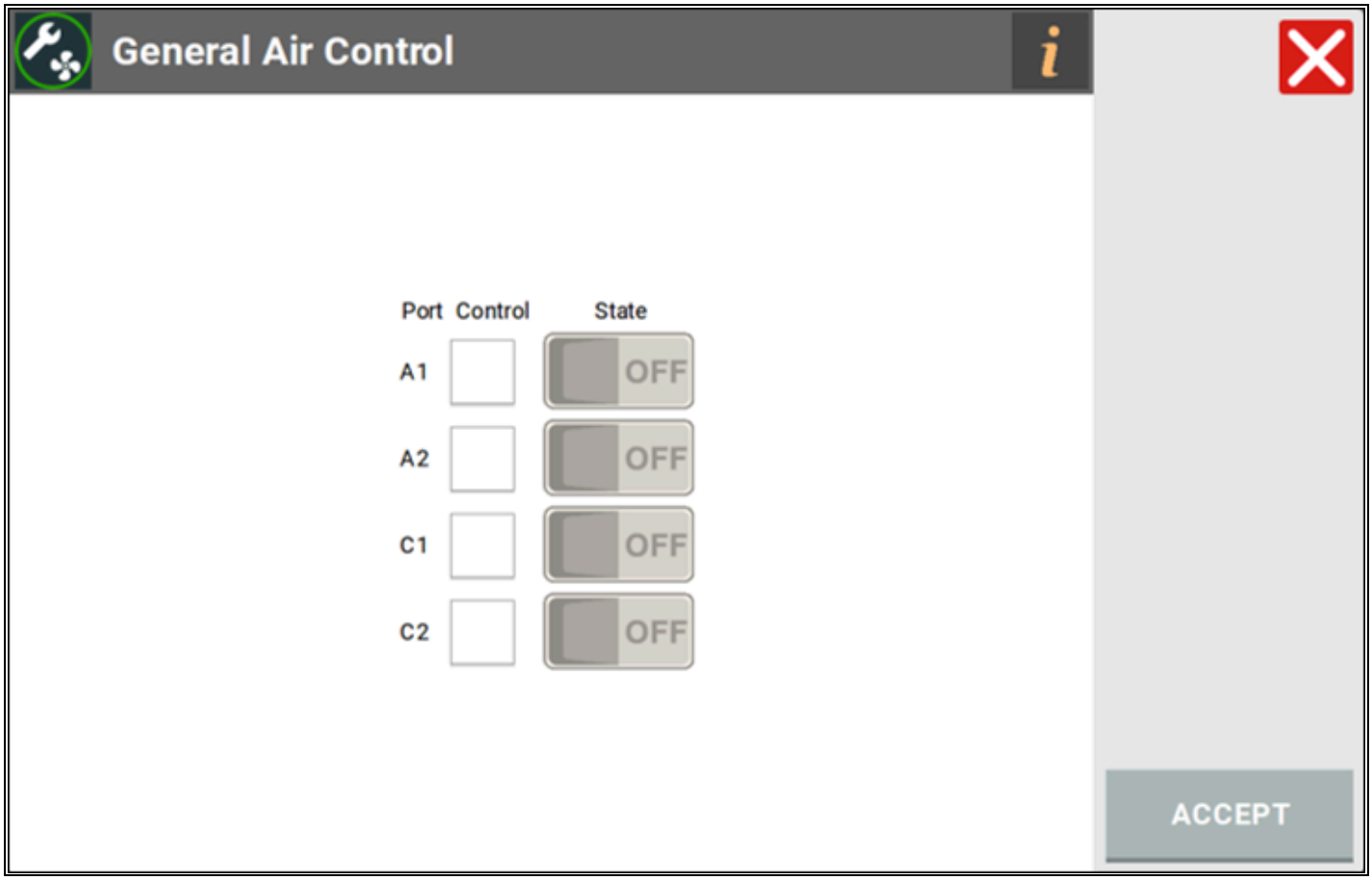
Setting	Description
Control	Select the air outlet (port) on the Forge system to control.
State	Toggle the state of the air outlet (port) on the Forge system, on or off.

General Air Out Control block *(Forge/Station only)*

The General Air block is a dynamic control for the air outlets on the Forge/Station. The available options change depending on the selected devices in Hardware Configuration. If a pneumatic device is selected as a device in Hardware Configuration, the air outlets reserved for the device aren't available in the block's settings.

Upon execution of a General Air block, the air outlet selected in the block's settings takes the indicated state. The air outlet holds the state until another General Air block switches the air outlet into a different state. The General Air block only changes the state of selected outlets. If an outlet is off and a 6mm Air block executes that doesn't have control over that outlet, the outlet holds its state.

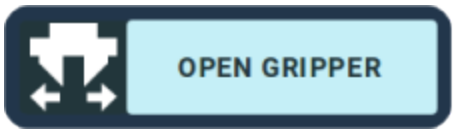
Below is the General Air Control generator that pops up when the adding a General Air block.



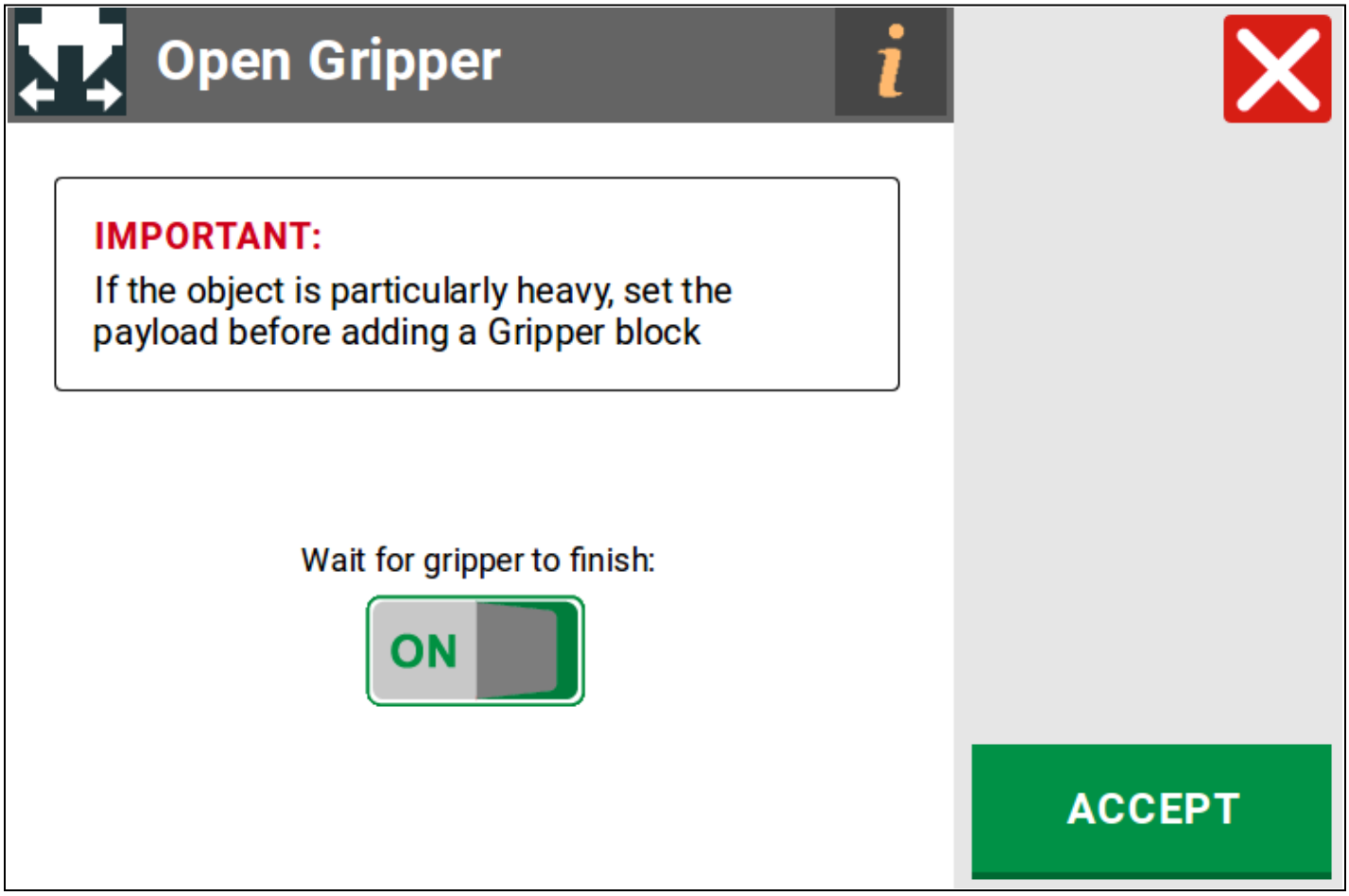
Setting	Description
Control	Select the air outlet (port) on the Forge system to control.
State	Toggle the state of the air outlet (port) on the Forge system, on or off.

Open Gripper and Close Gripper blocks

The Open Gripper and Close Gripper blocks actuate finger-based grippers.



Below is the Open Gripper generator which pops up when adding an Open Gripper block. A similar generator pops up when adding a close gripper block.



Setting	Description
Wait for gripper to finish	Toggle when the system moves to the next block in the task. When on, the system waits for the gripper to finish its action before moving to the next block. When off, the system moves to the next block in the task while performing the gripper action.

Granular Gripper Control block

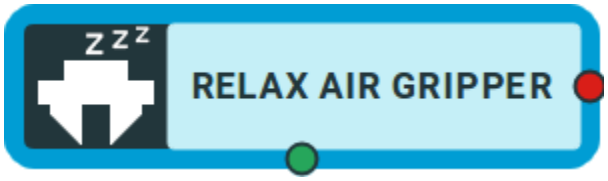
The Granular Gripper Control block enables you to customize the travel distance and force of some grippers, like the Robotiq 85 mm and Robotiq 140 mm 2-finger grippers. The Granular Gripper Control is not a Gripper

Open or Gripper Close block. When executing the Granular Gripper Control block, the system will bring the gripper fingers to the specified distance regardless of the gripper's state prior to execution, open or closed.

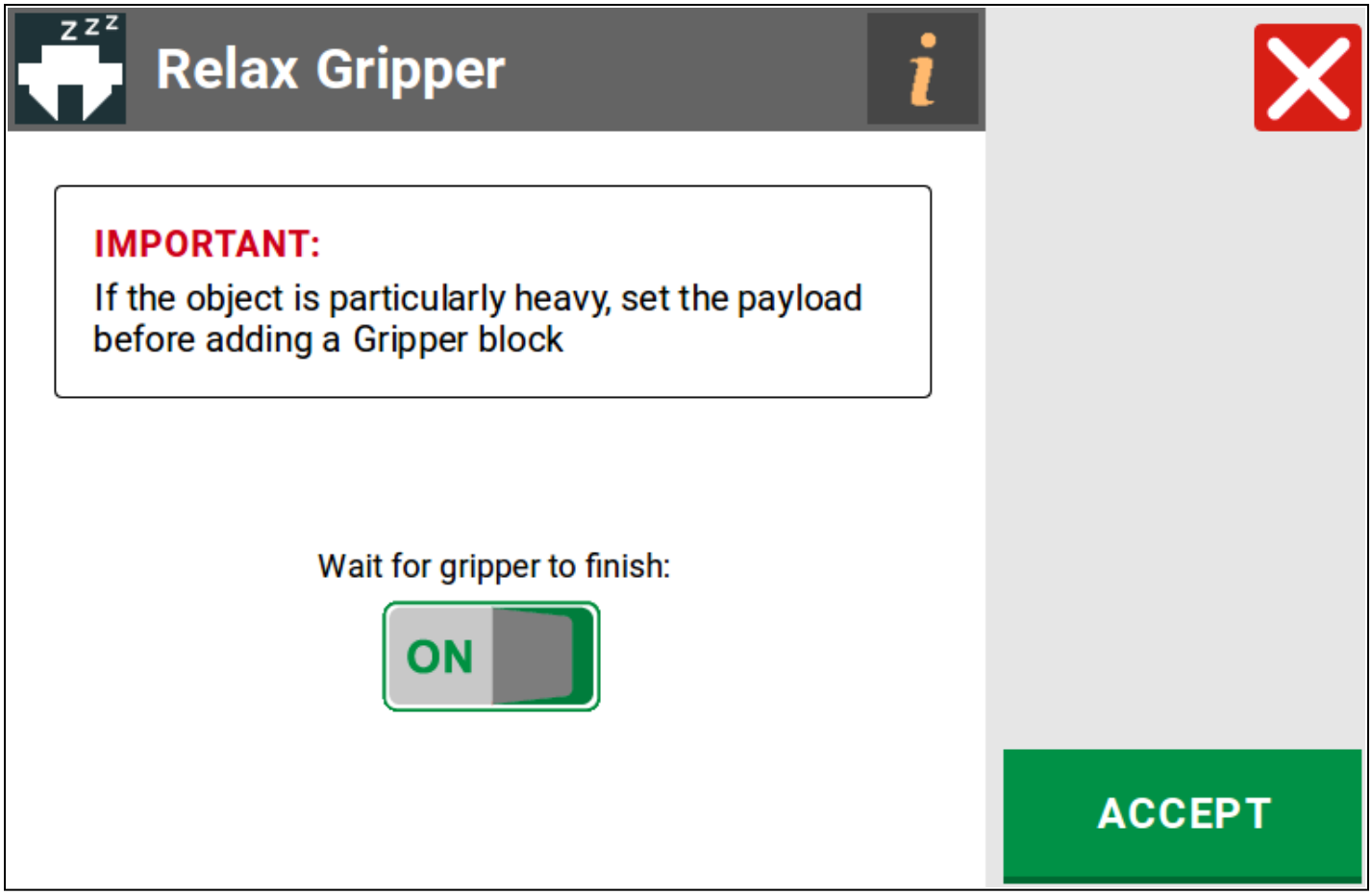
Setting	Description
Percent Closed	Set the percentage of fully closed at which the gripper stops, 100% is fully closed and 0% is fully open.
Force	Set the amount of force the gripper must feel before stopping when performing a closing action. A lower value makes the gripper more sensitive to force. A higher value makes the gripper less sensitive to force.
Wait for Gripper to Finish	Toggle when the system moves to the next block in the task. When on, the system waits for the gripper to finish its action before moving to the next block. When off, the system moves to the next block in the task while performing the gripper action.

Relax Gripper

Removes the pneumatic pressure on a gripper. When on, a pneumatic gripper, like the Schunk's 2-finger gripper, is not forced open or closed by the system. Instead, the gripper remains in its current state and can be opened or closed by applying force at the gripping point. This option is only available for some grippers.



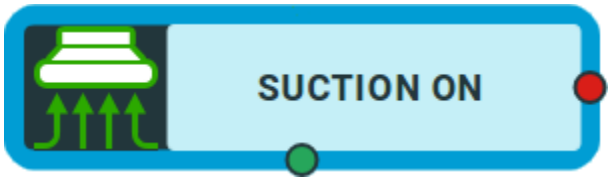
Below is the Relax Gripper generator which pops up when adding a Relax Gripper block.



Setting	Description
Waiting for gripper to finish	Toggle when the system moves to the next block in the task. When on, the system waits for the gripper to finish its action before moving to the next block. When off, the system moves to the next block in the task while performing the gripper action.

Suction On and Suction Off blocks

The Suction On and Suction Off blocks control suction grippers where air is to be on or off for all air inlets, in unison. These blocks are only available when a suction gripper is selected as the end effector in Hardware Configuration.



Press Pedal and Release Pedal

The Press Pedal and Release Pedal blocks control the pneumatic pedal press, available from READY Robotics. These blocks are only available when a pedal press is selected as a device in Hardware Configuration.

Technical Support

If you encounter a problem with a Forge system's or Forge/OS, reach out to us with the Report Issue feature via the READY menu in the lower left corner of the Canvas. READY Robotics recommends using the Report Issue feature before contacting technical support, as the issue report will help us pinpoint the problem in your system and significantly reduce the time to troubleshoot.

- Email READY Robotics: support@ready-robotics.com
- Call READY Robotics: 833-732-3977