

MachineLogic Application Examples

Contents

[Application Examples](#)

[Uploading an application](#)

[Conveyor Application](#)

[Linear Actuator / Gantry](#)

[Robot Application](#)

[Linear Actuator & Rotary](#)

[Actuator Application](#)

[Robot Application](#)

[Calibration Application](#)

[Template](#)

Application Examples

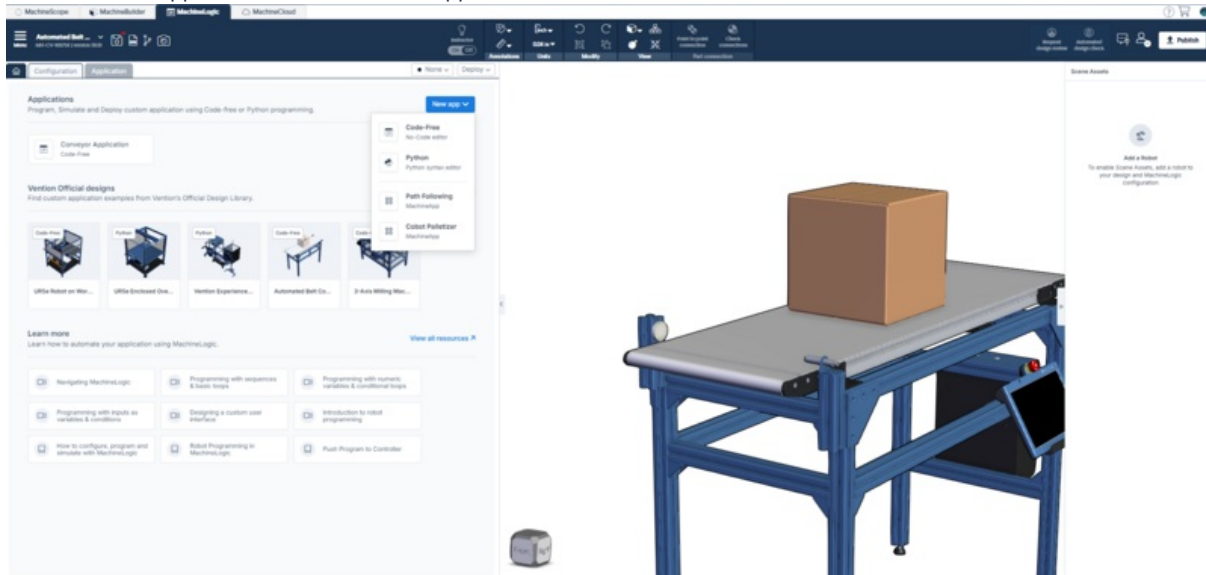
This document is intended to provide a variety of program examples that can be downloaded from this page and uploaded to any MachineLogic session. It contains examples for both code-free and Python editor for all device type supported in MachineLogic.

Uploading an application

follow the steps below to upload a new application to MachineLogic. For more information on programming and configuration in MachineLogic, [click here](#).

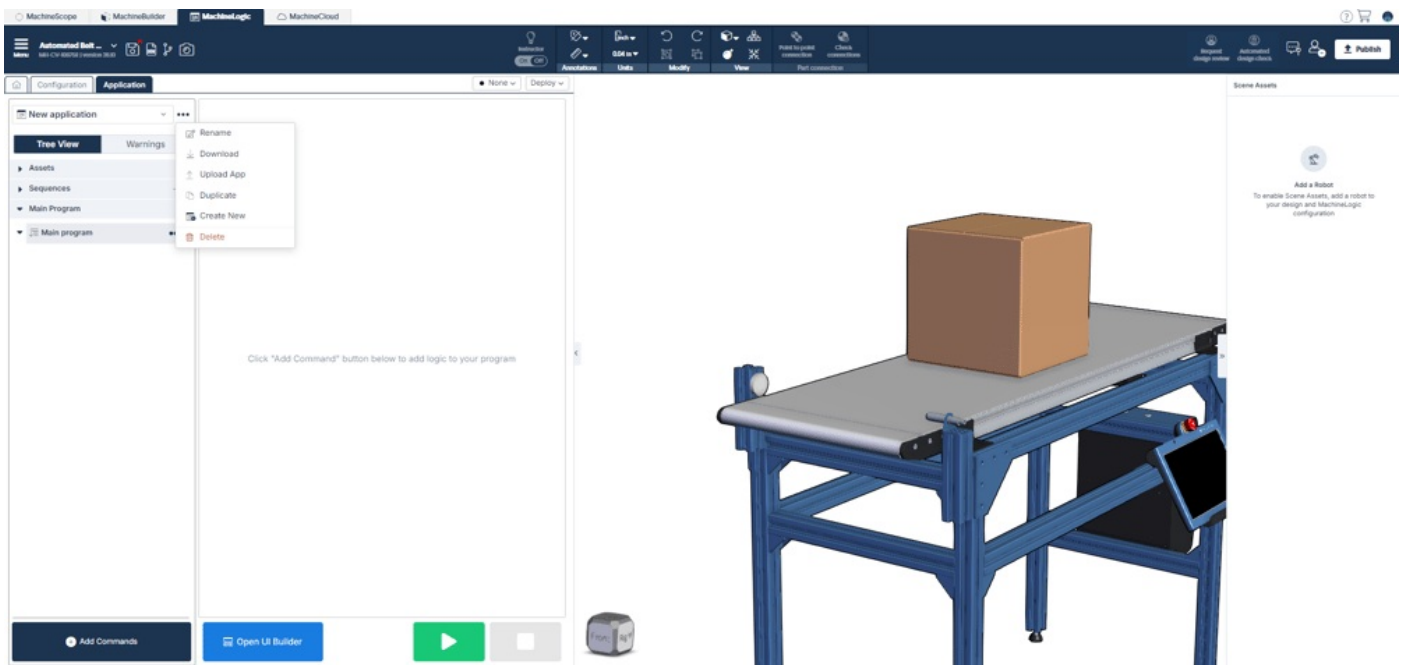
1. Creating a new application

To create a new application, click on the New App button as shown below:



2. Uploading an application

To upload an application, select the context menu and select the **Upload** Button.

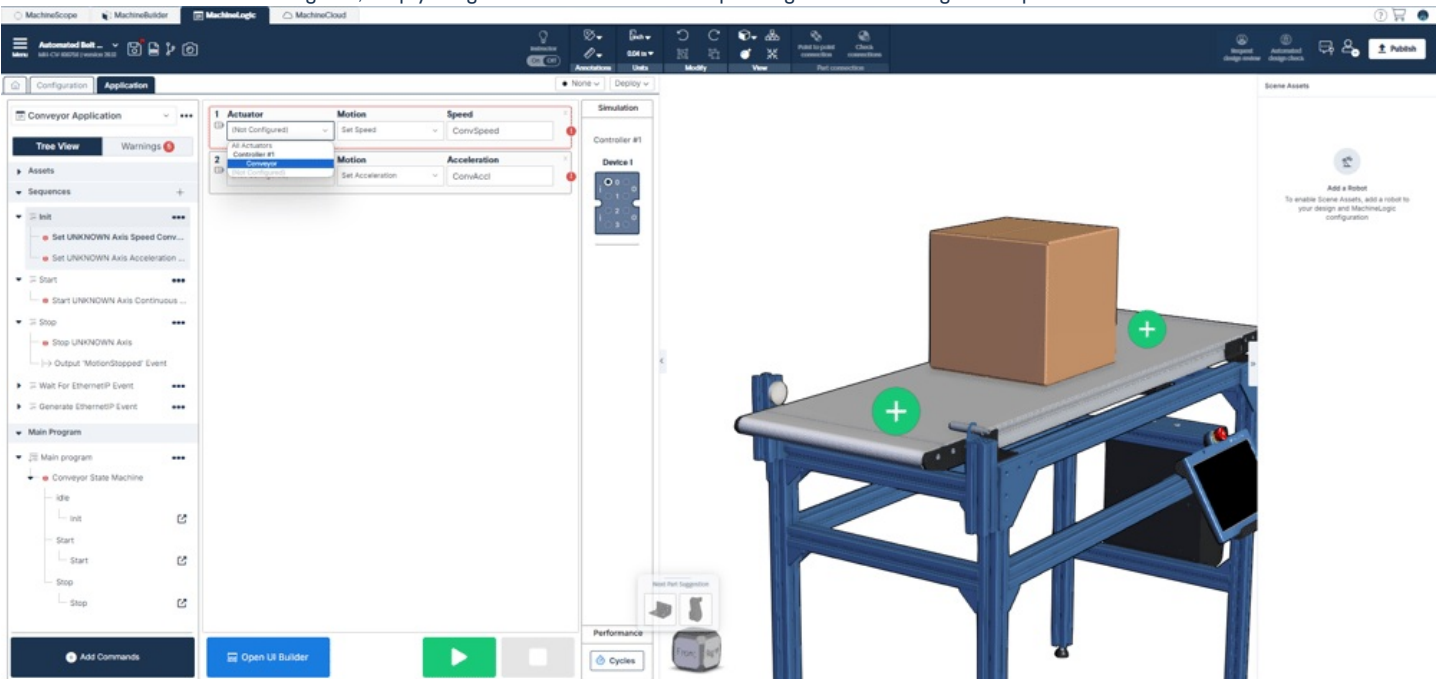


3. Assigning a configured device to an instruction.

Once an application has been uploaded, devices must be assigned to their corresponding instruction. To assign a device to a code-free instruction, it must first be configured. To learn more about configuration in MachineLogic, [click here](#)

Code-Free

Once all devices have been configured, simply assign each device to its corresponding instruction using the drop-down menu.

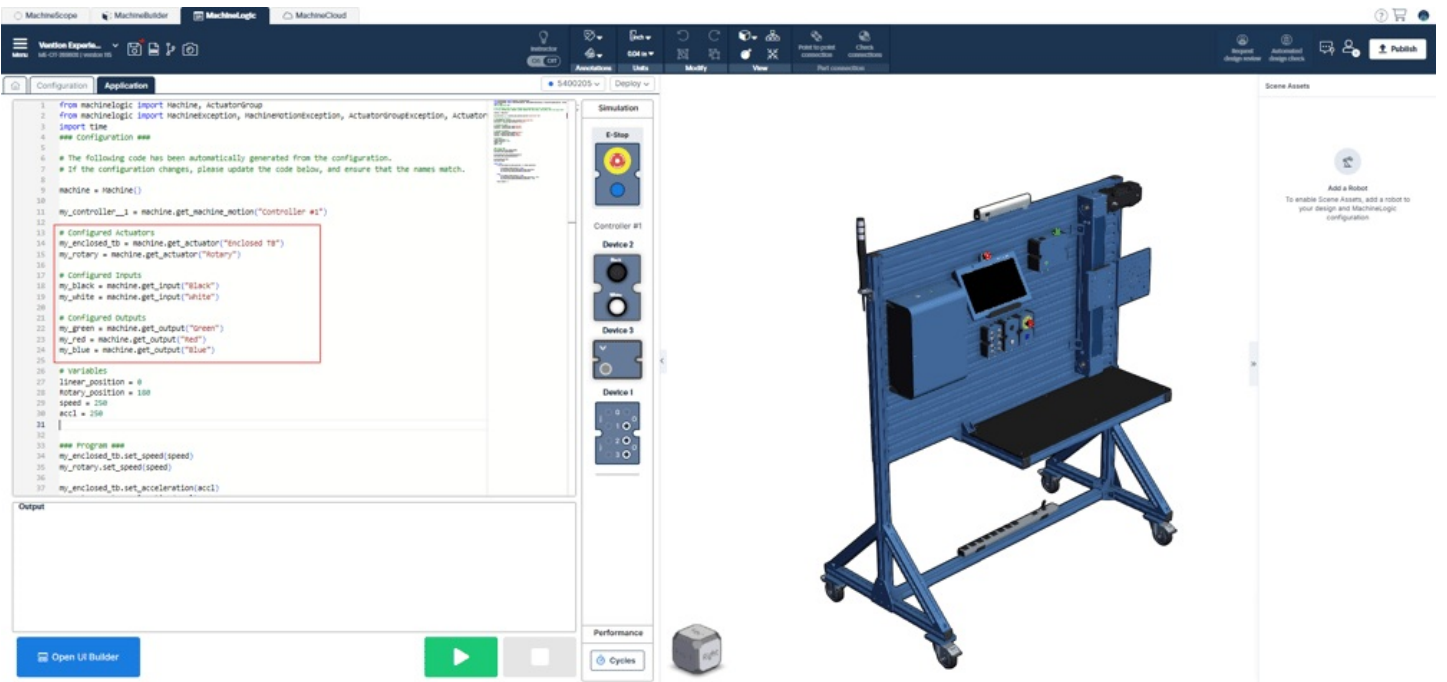


Python

Once all devices have been configured, simply assign each device to its corresponding machine class getters that appear at the header of the program:

```
my_actuator = machine.get_actuator("Actuator name from configuration")
my_digital_output = machine.get_output("Output name from configuration")
my_digital_input = machine.get_input("Input name from configuration")
```

Every time a new application is created, the machine getters are automatically populated as per the content of configuration page.



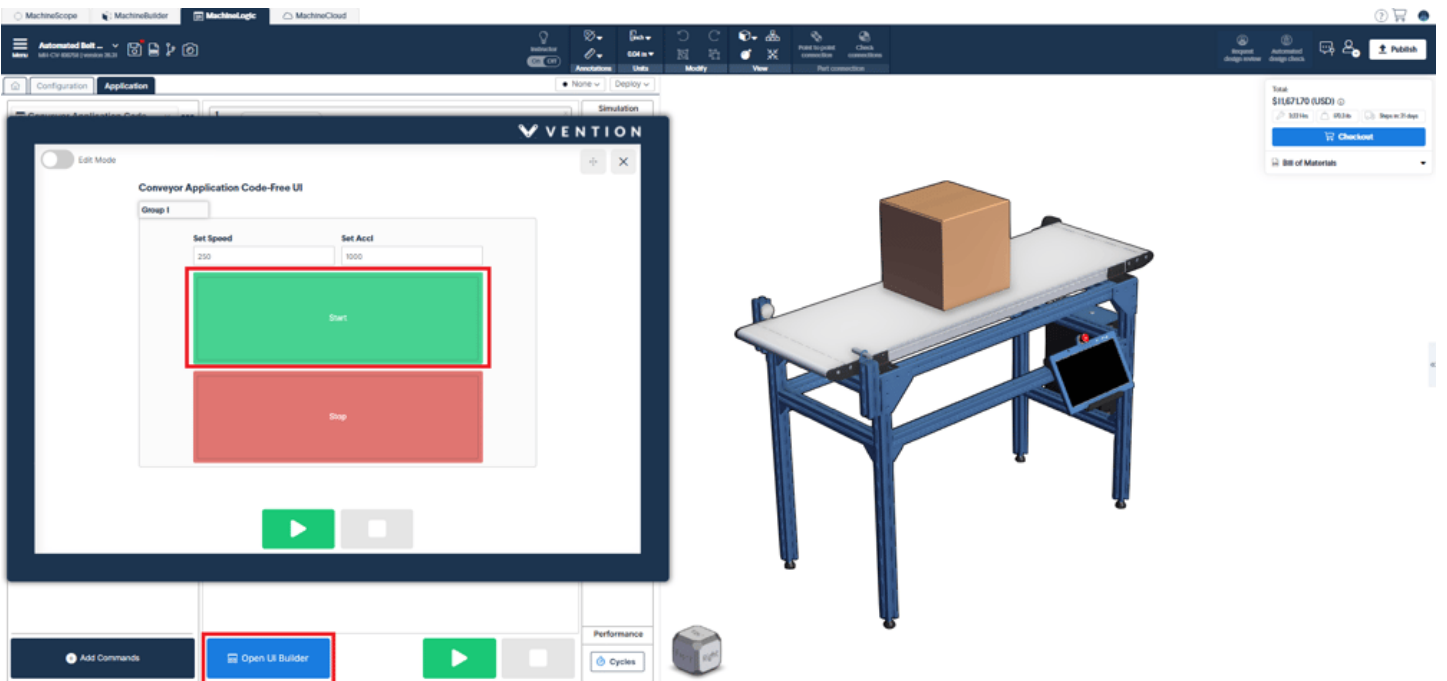
Conveyor Application

[Link to Design Example](#)

This application controls a simple conveyor that stops when a box object is detected by a beam sensor. It contains instructions for the following devices:

- Actuator (Conveyor)
- Digital Input (Sensor)

To operate the conveyor, press play and select the UI builder button to access the conveyor controls.



image

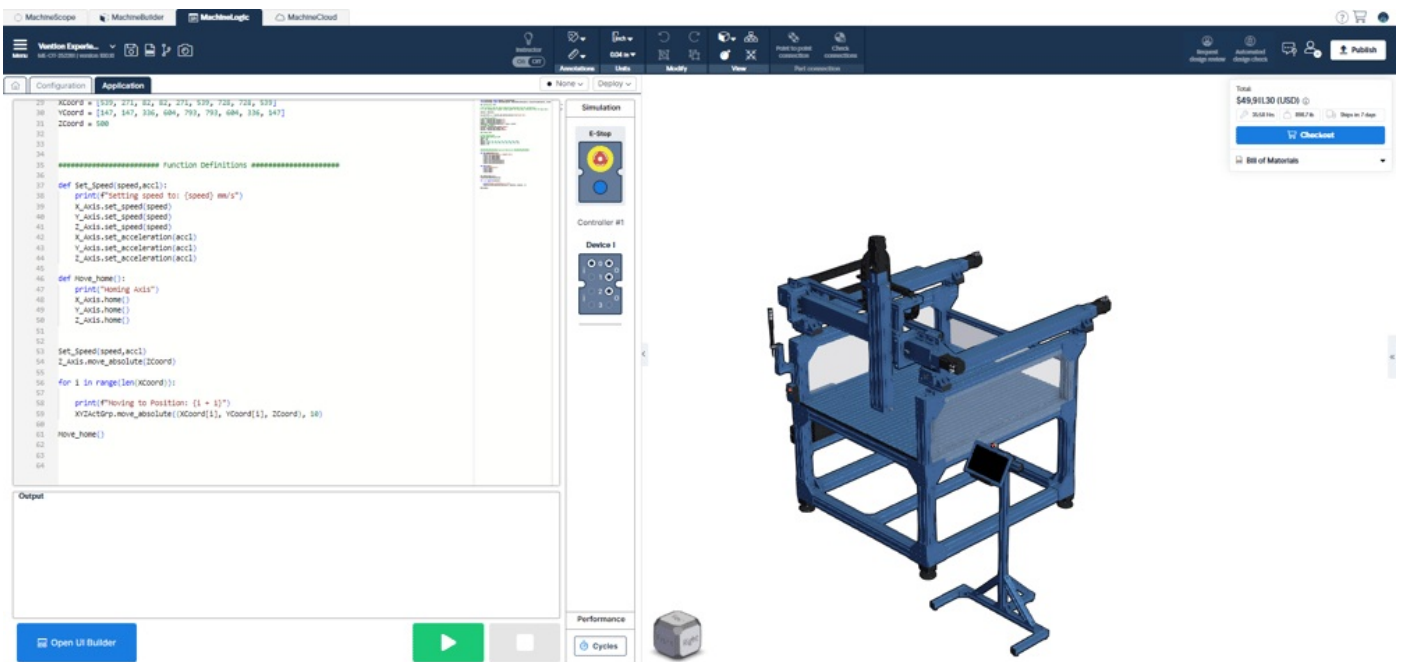
Download Code-Free Example

Linear Actuator / Gantry Robot Application

[Link to Design Example](#)

This application controls a simple Gantry composed of three linear actuators, two of which are synchronized. It contains instructions for the following devices:

- Actuator 1 (X Axis)
- Actuator 2 (Y Axis)
- Actuator 3 (Z Axis)
- Digital Input (Light Curtain)
- Output 1 (Green light)
- Output 2 (Red light)
- Output 3 (Blue light)
- To operate the conveyor, press play and select the UI builder button to access the machine controls.

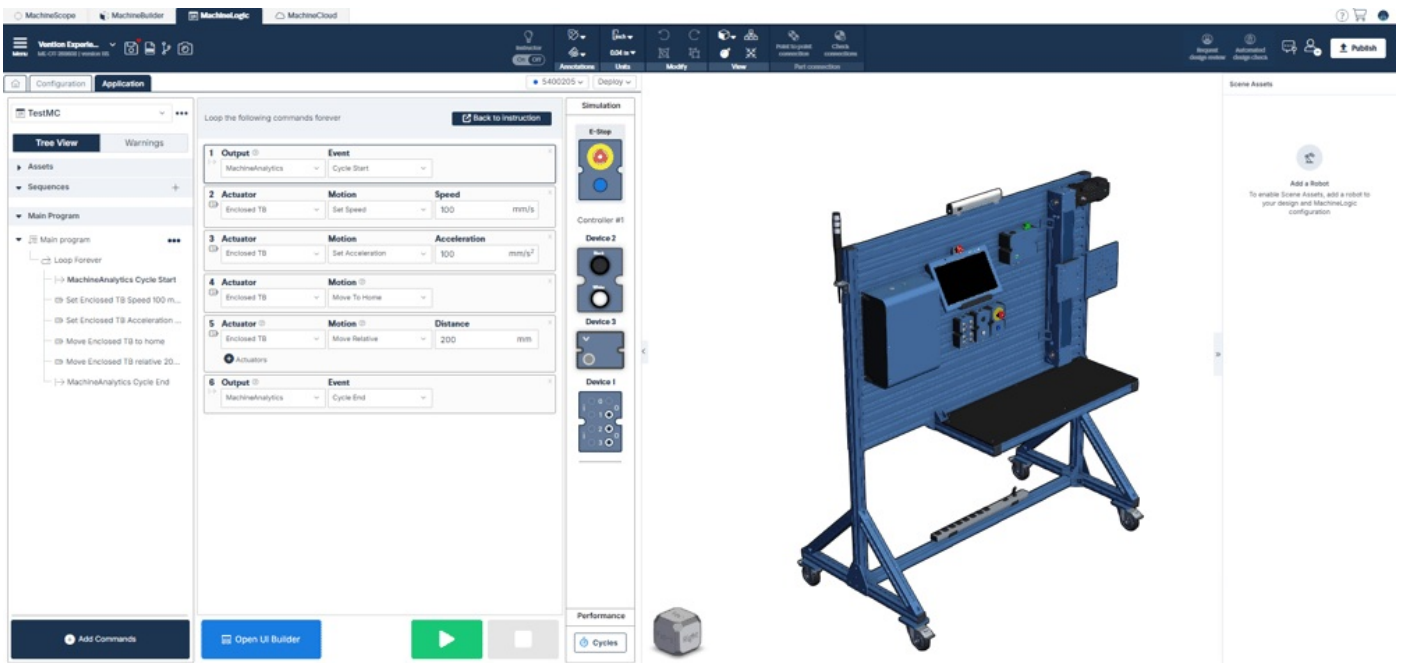
[Download Code-Free Example](#)[Download Python Example](#)

Linear Actuator & Rotary Actuator Application

[Link to Design Example](#)

This application controls a workstation composed of both a linear and rotary actuator. To Operate the application, simply select the play button. This application contains instructions for the following devices:

- Linear Actuator (Enclosed TB)
- Rotary Actuator (Rotary)
- Digital Input 1 (Black button from Push button module)
- Digital Input 12 (White button from Push button module)
- Output 1 (Green light from tower light)
- Output 2 (Red light from tower light)
- Output 3 (Blue light from tower light)
- To Operate the application, simply select the play button.



Download Code-Free Example

Download Python Example

Robot Application

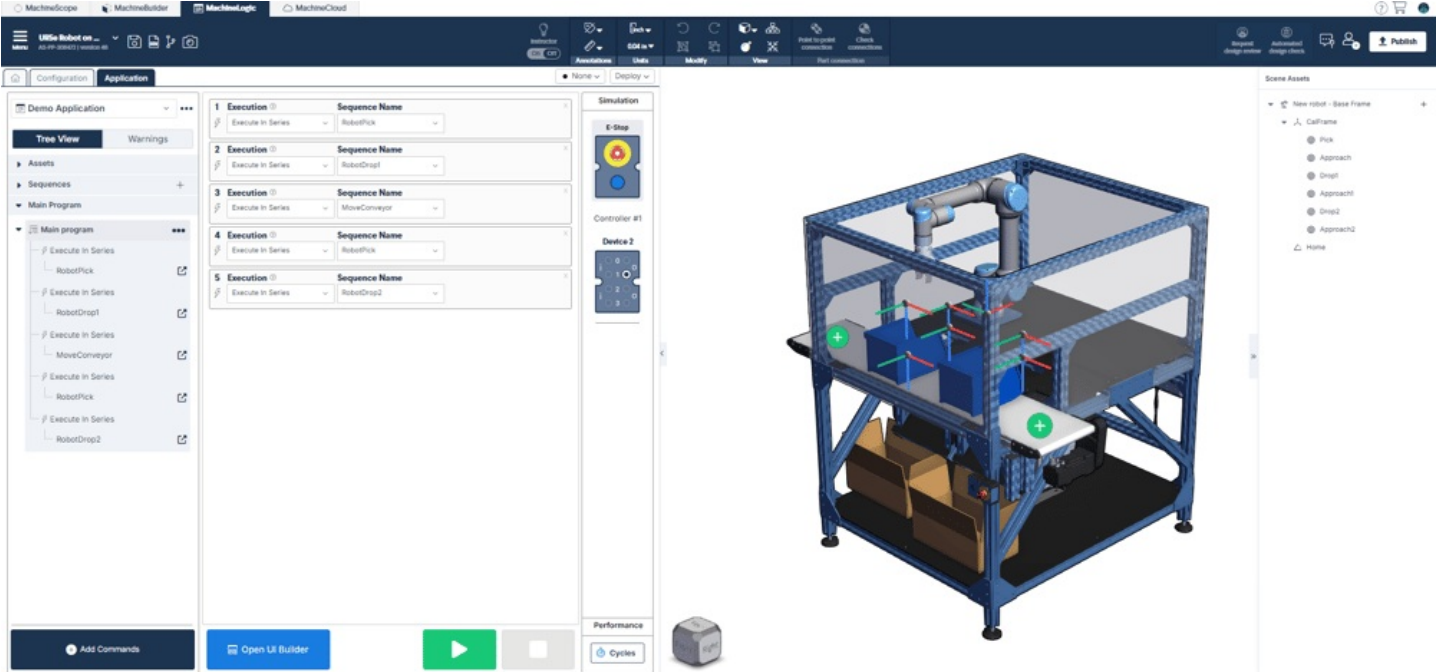
[Link to Design Example](#)

This application controls a UR10 robot and a conveyor.

This application contains instructions for the following devices:

- Robot (New robot)
- Actuator (Conveyor)
- Digital Output (Gripper)

To Operate the application, simply select the play button.



[Download Code-Free Example](#)

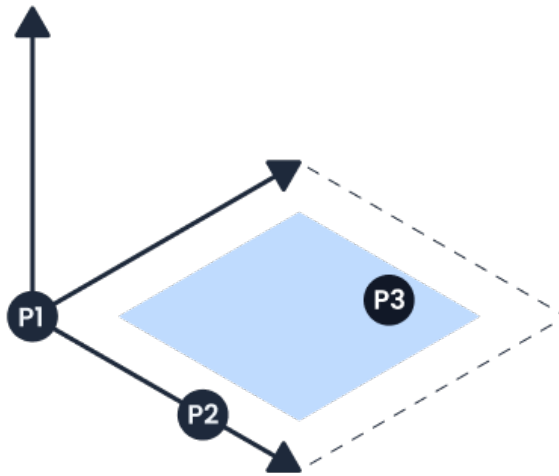
[Download Python Example](#)

Calibration Application Template

This application allows you to calibrate calibration frames on the physical machine.
2 methods are provided in this application to properly calibrate your frame on different use cases.

1. 3 Points with Origin method:

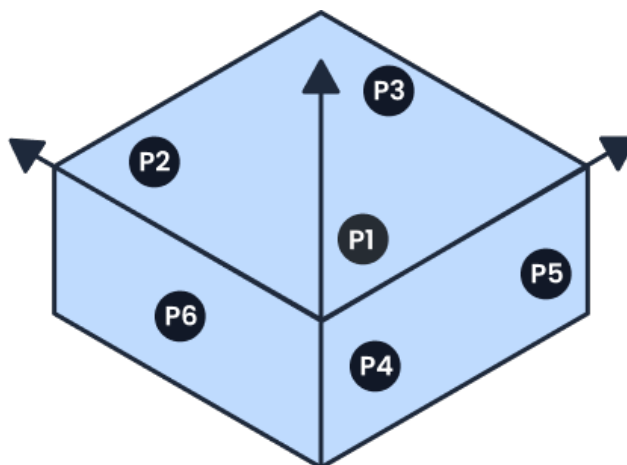
This method involves recording three specific positions within the robot's workspace by using the Tool Center Point (TCP). These positions serve as reference points for establishing the position and rotation of the calibration frame.



3 Points with Origin method

2. 6 Points - 3 Planes method:

This method involves recording six points to define three planes, allowing us to calculate with high precision the position in space of the calibration frame.



6 Points - 3 Planes method

To properly use this calibration application, you will need:

- Supported robot with MachineLogic

- MachineMotion v2.15 or higher
- Calibration Frames defined in your application
- Properly Calibrated TCP for your robot. Ideally, the TCP should be defined at the tip of your tool to ensure proper probing of the points of interest.
- The calibration application that can be downloaded with the following link:

[Download Calibration Template](#)

The steps to properly deploy the application are outlined in this user manual [How to deploy your calibration application](#)