

MachineLogic Application Library

Contents

[Application Examples](#)

[Uploading an application](#)

[MachineLogic Code-Free & Python Applications](#)

[Simple Conveyor](#)

[Linear Actuator in Gantry Configuration](#)

[Robot Application](#)

[MachineLogic Utilities Applications](#)

[Calibration](#)

[Application Template](#)

[Conveyor Process Library](#)

Application Examples

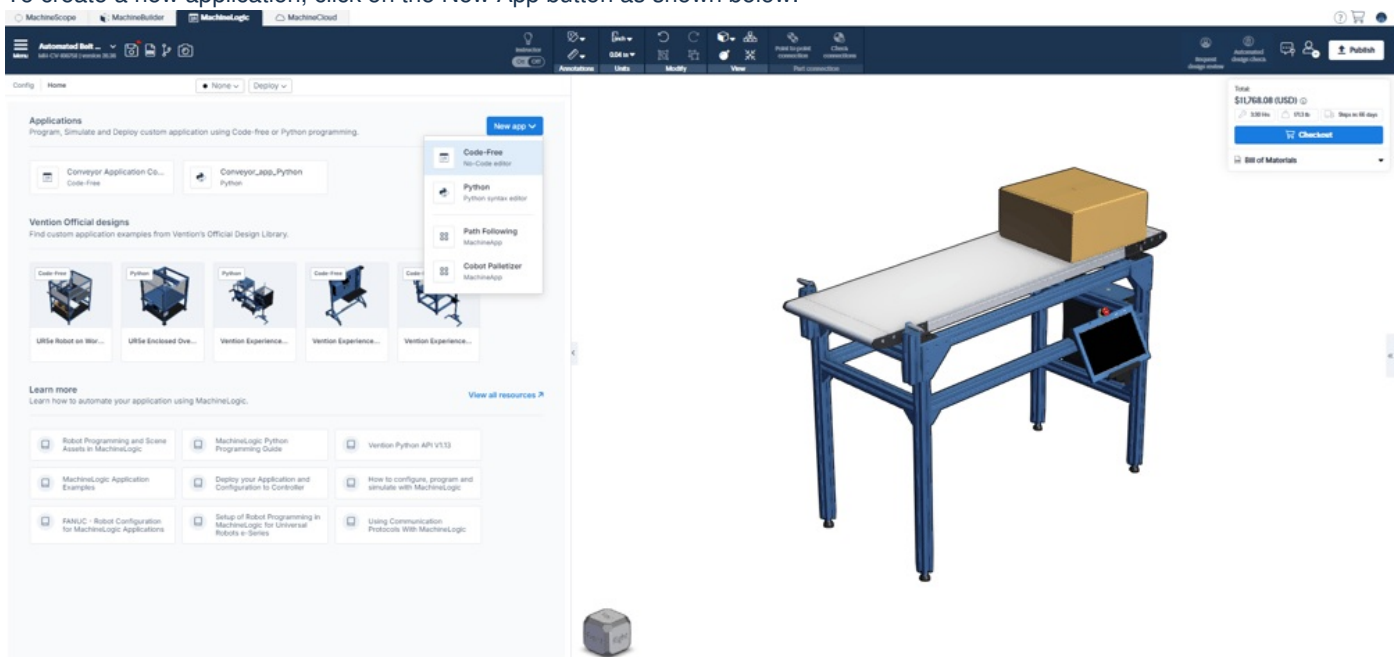
This document is intended to provide a variety of program examples that can be downloaded from this page and uploaded to any MachineLogic session. It contains examples for both code-free and Python editor for all device type supported in MachineLogic.

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](#).

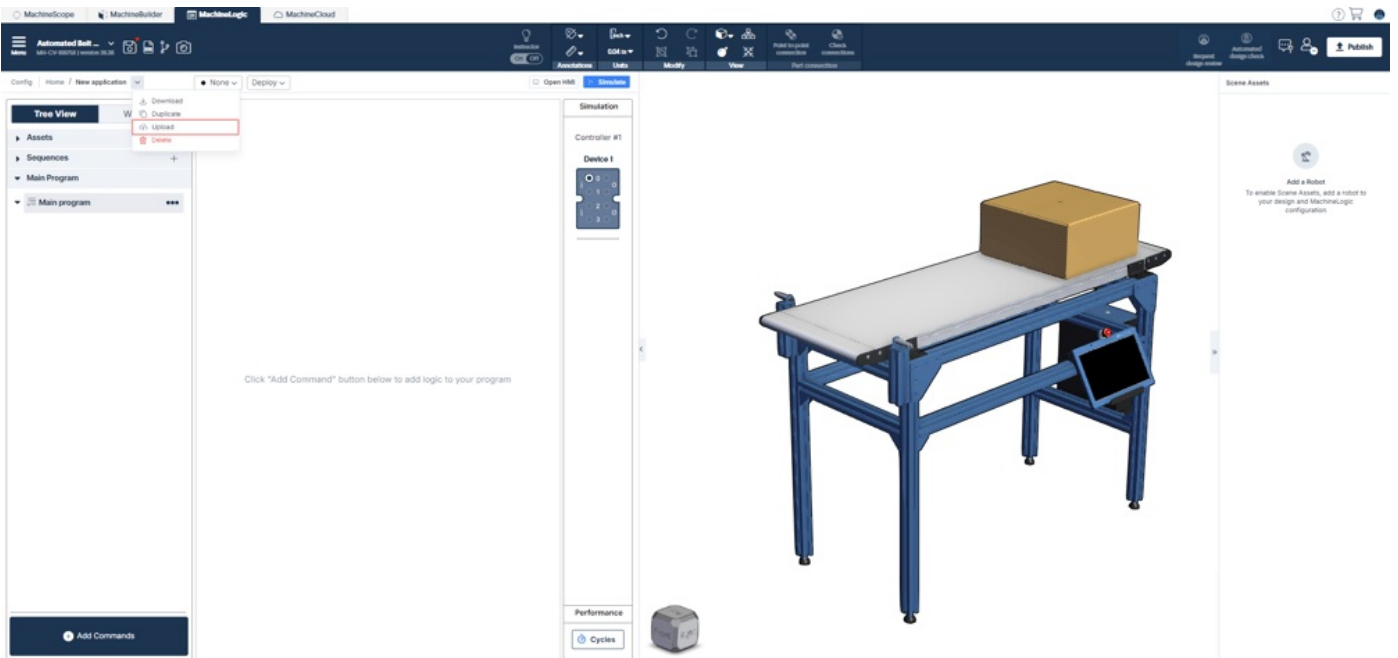
1. Creating a new application

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.

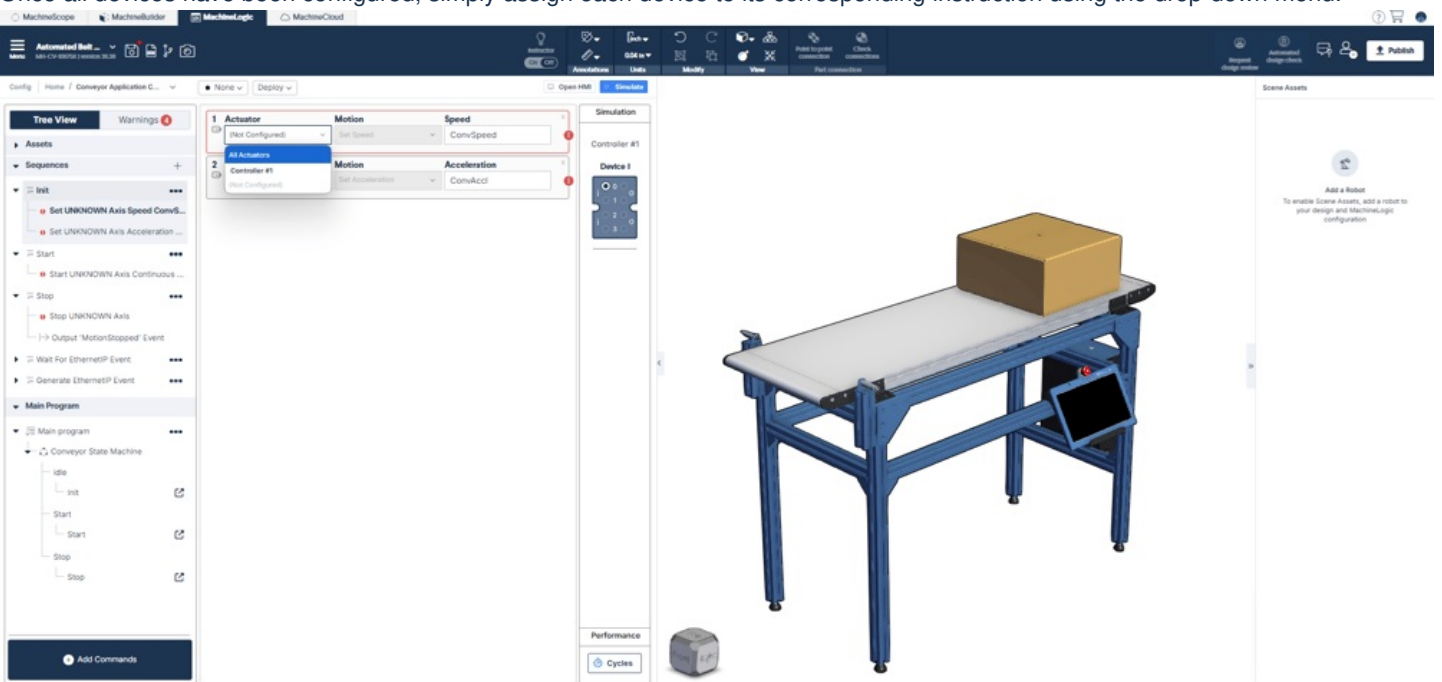


3. Assigning a configured device to an instruction.

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.

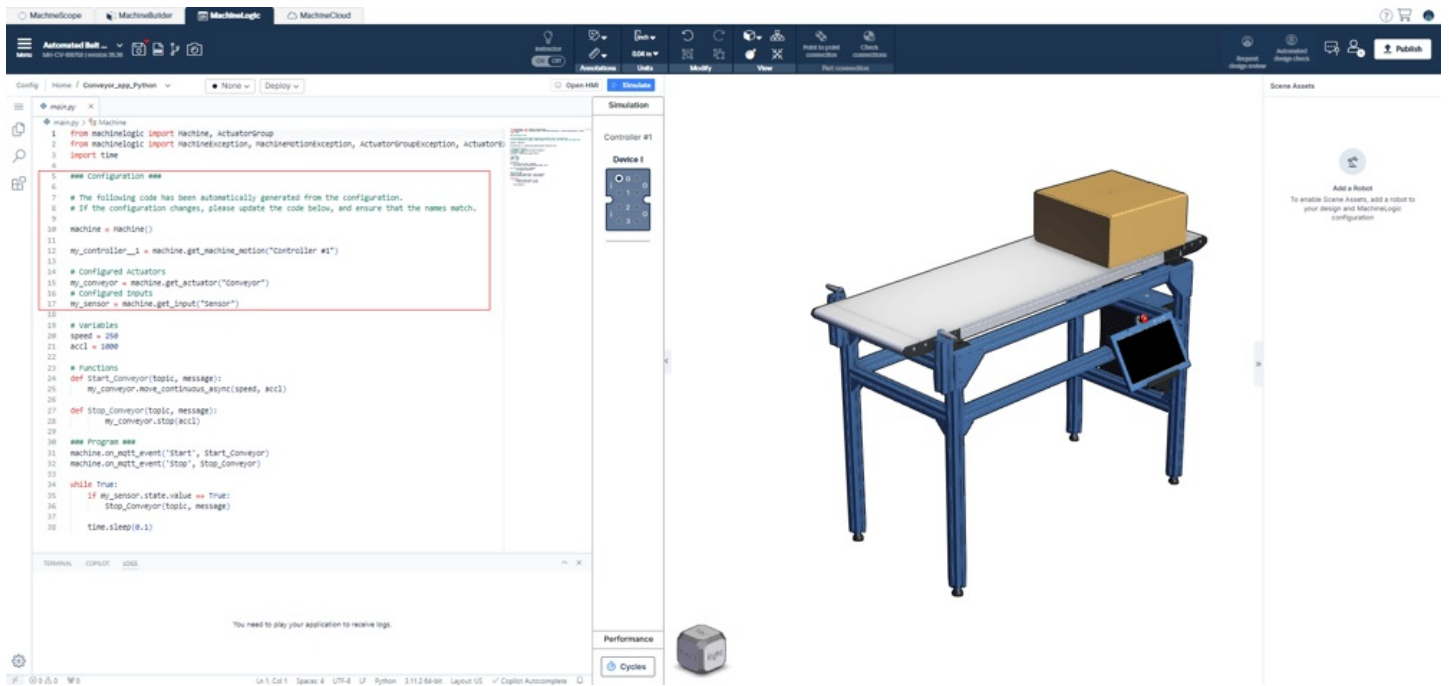


Python

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.



MachineLogic Code-Free & Python Applications

Applications below can be uploaded in the MachineLogic Code-Free or Python editor.

To learn more about code-free programming, [click here](#)

To learn more about Python programming, [click here](#)

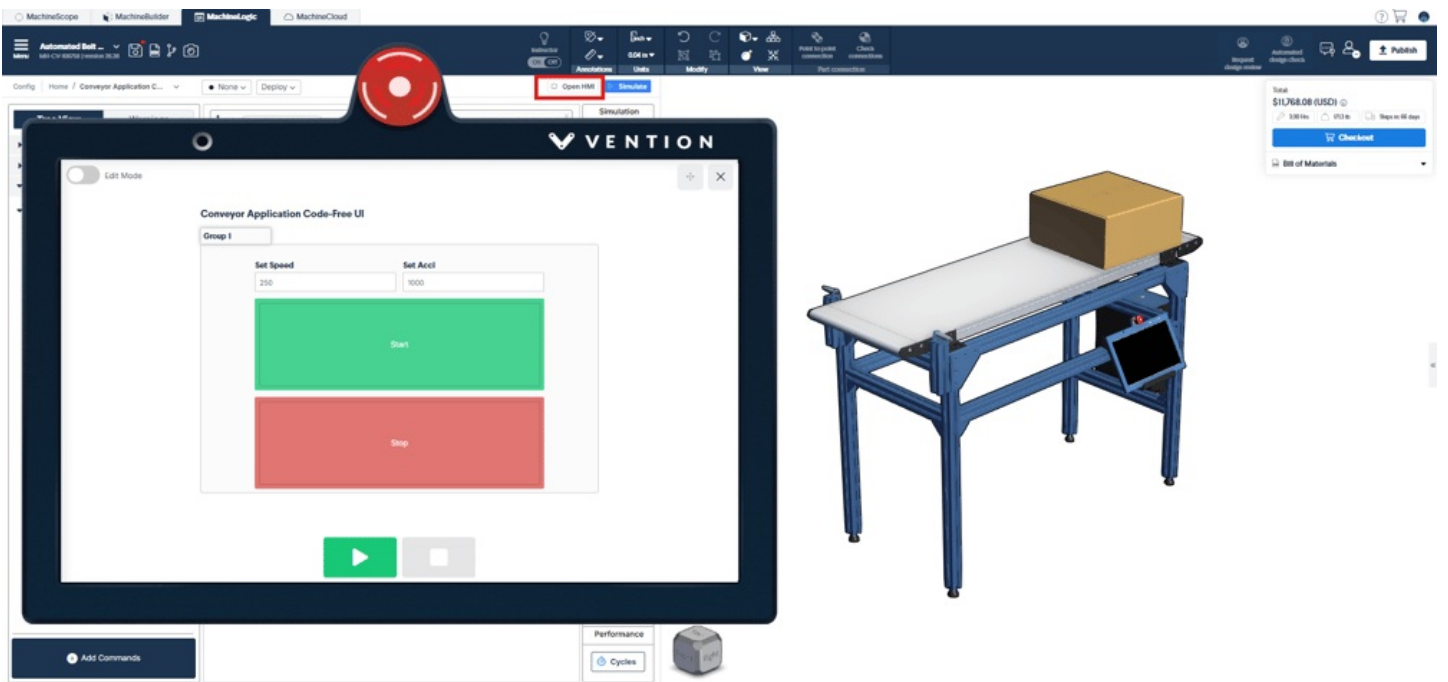
Simple Conveyor

[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 the simulate button and select the Open HMI button to access the conveyor controls.



image

[Download Code-Free Example](#)

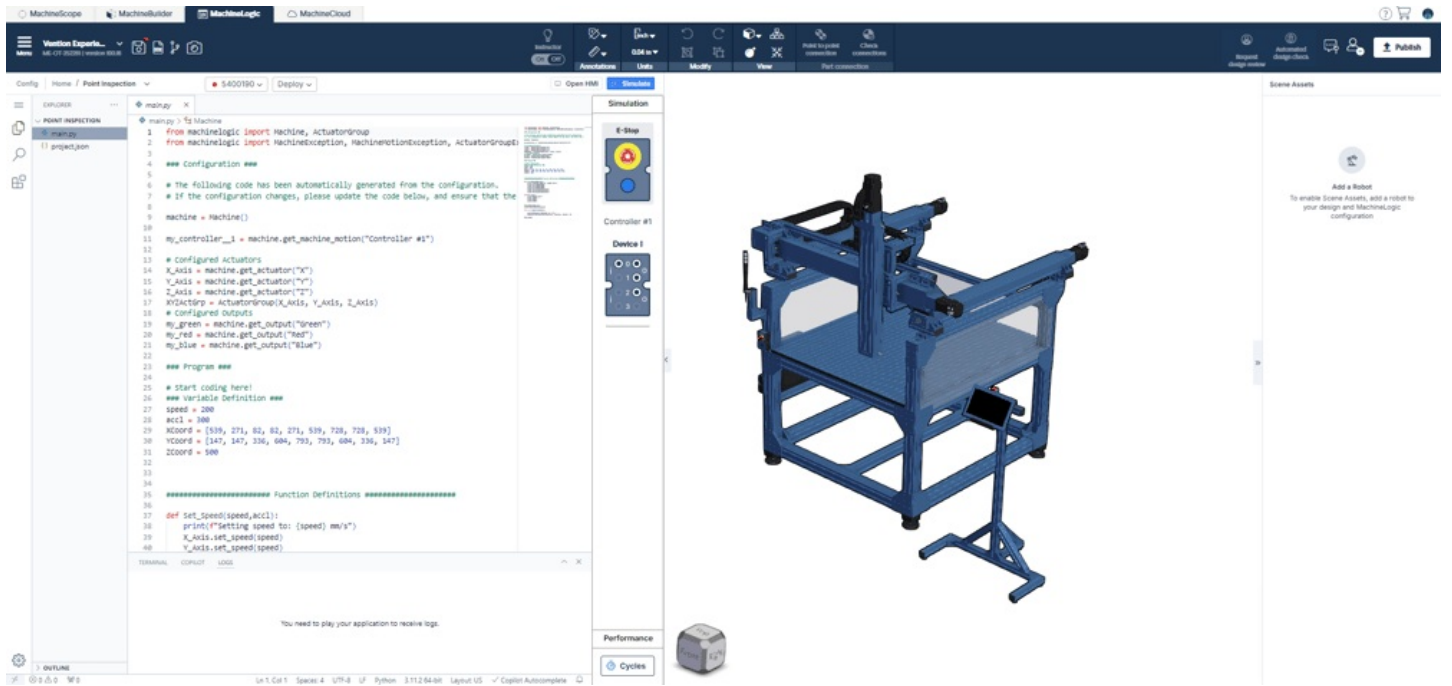
[Download Python Example](#)

Linear Actuator in Gantry Configuration

[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)



image

Download Code-Free Example

Download Python Example

Robot Application

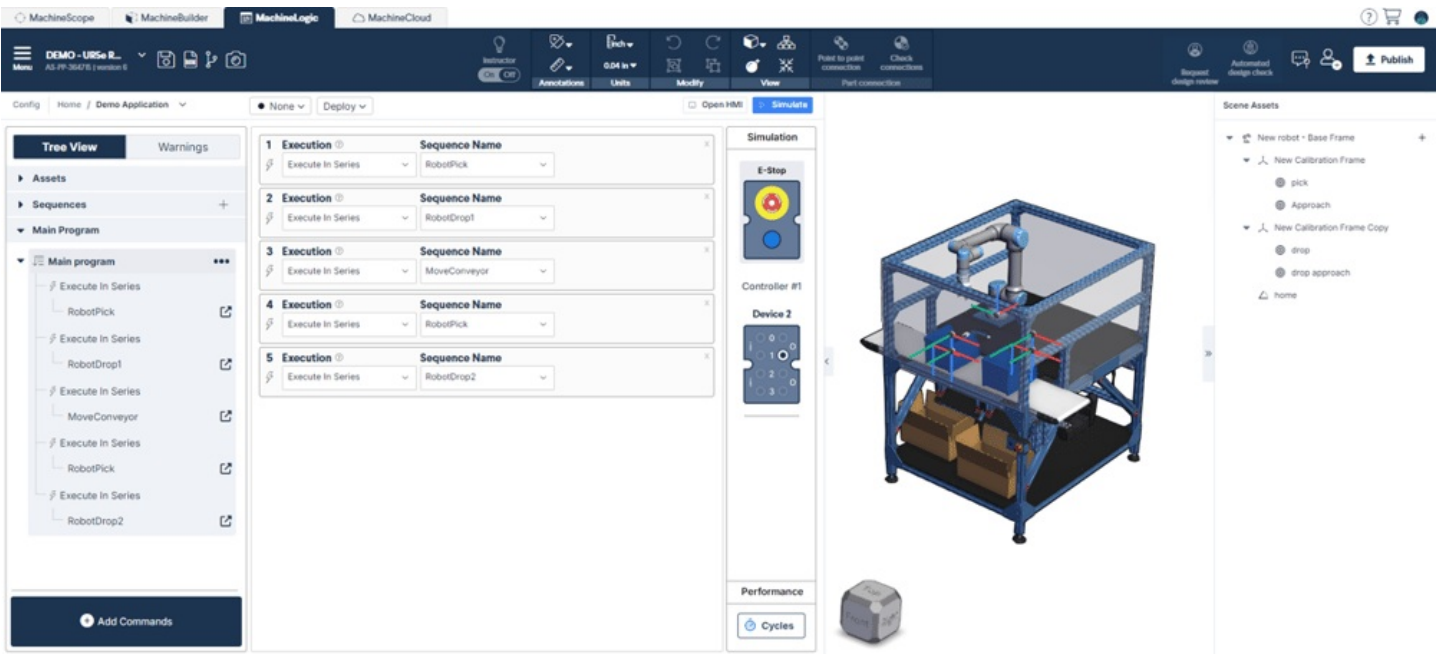
[Link to Design Example](#)

This application controls a UR10 robot and a conveyor. To learn more about Code-Free Robot programming, [click here](#)

This application contains instructions for the following devices:

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

To Operate the application, simply select the **Simulate** button.



[Download Code-Free Example](#)

[Download Python Example](#)

MachineLogic Utilities Applications

Applications below can be uploaded in the MachineLogic Python editor and contain rich operator interfaces. Use these applications as plugins that can extend the capabilities of MachineLogic.

To learn more about Python programming and custom rich Operator interfaces, [click here](#)

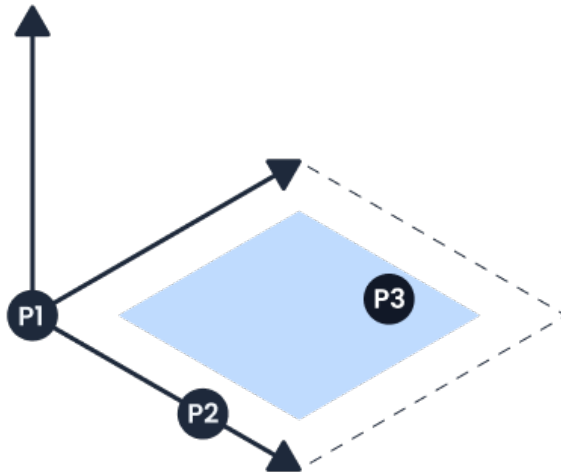
Calibration Application Template

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:**

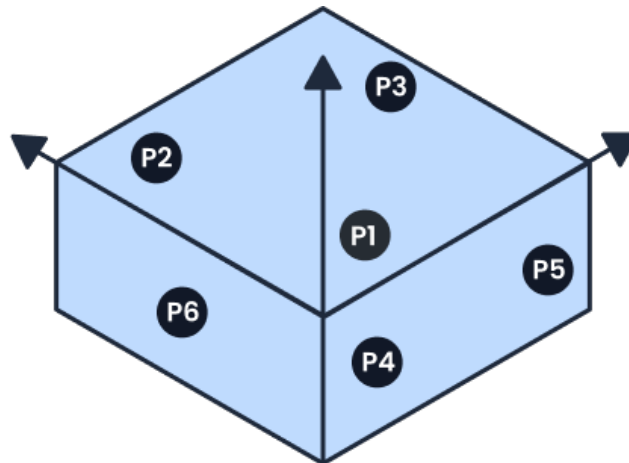
This method involves recording three specific positions within the robot's workspace by using the Tool Center Point (TCP). These positions serve as reference points for establishing the position and rotation of the calibration frame.



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

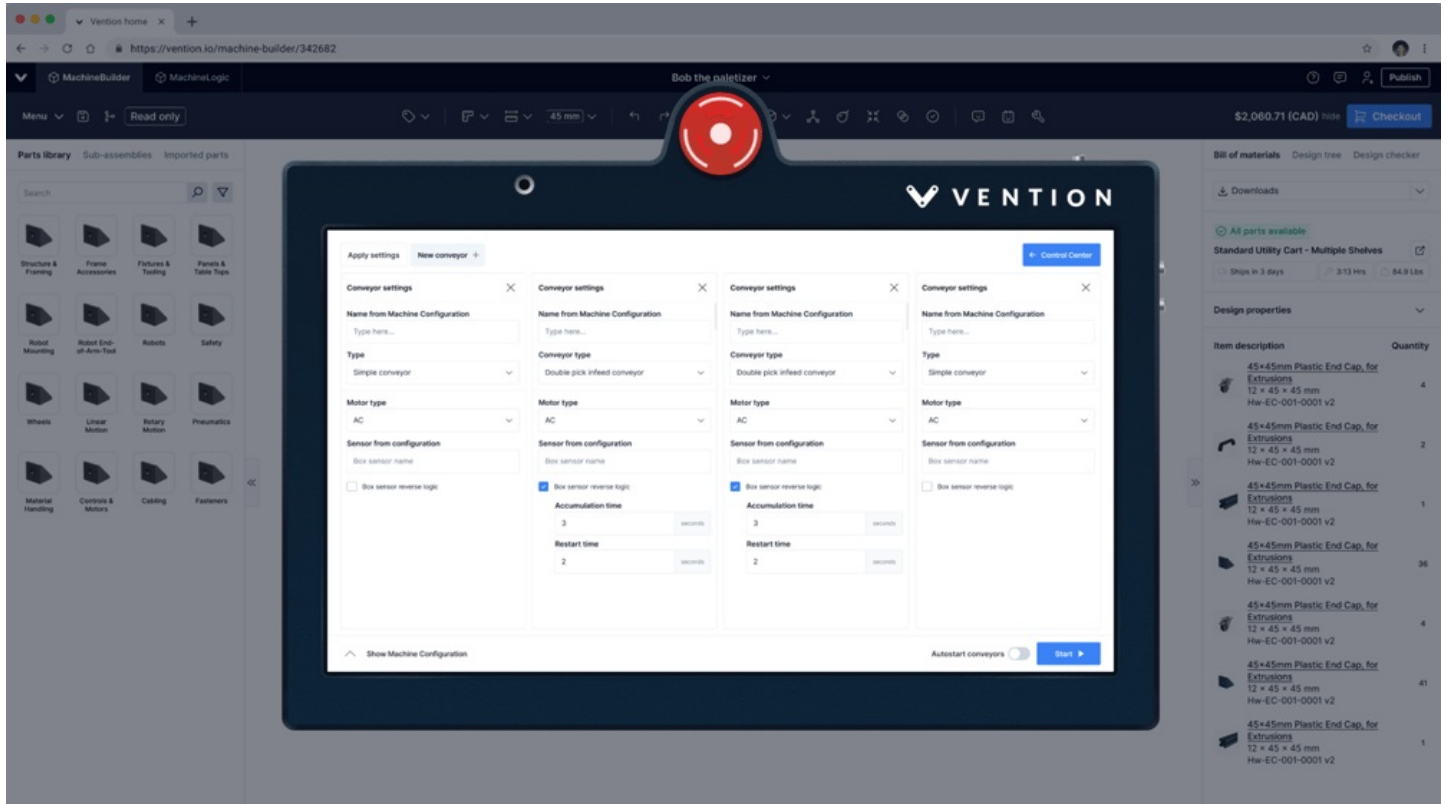
To properly use this calibration application, you will need:

- 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](#)

This application enables the instantiation and configuration of multiple conveyor processes in MachineLogic Python Applications. To use this application, upload it to MachineLogic's Python editor and open the Operator UI to create the desired conveyor processes. Conveyor processes can be created and configured from the Operator UI from the simulation environment.



Prerequisites

Before you begin, ensure that you have:

- Access to the Vention Control Center or CAD environment.
- Basic knowledge of Python, Typescript and JSON file editing, if you desire to modify conveyor behaviors.
- Configure All Inputs, Pneumatics, Actuators.
- Ensure all physical inputs, including pneumatics and actuators, are correctly configured and tested within your system. This initial setup is crucial for the seamless integration of the conveyor process. Assign a meaningful name and configuration according to your project requirements.

Conveyor Types

- Multi-Pick Infeed Conveyor (1 Pusher)
 - Sensors: 2 box detection sensors
 - Logic: When product 1 arrives at zone 1, the conveyor pauses, pusher extends and retracts. When product 2 arrives at zone 1, the conveyor pauses, pusher extends box 2 to box 1 and retracts. Robot picks both boxes, and the conveyor restarts.
- Infeed Conveyor
 - Sensors: 1 box detection sensor
 - Logic: When the product arrives, the conveyor pauses.
- Accumulating Conveyor
 - Sensors: 1 box detection sensor
 - Logic: When both sensors detect a product, the conveyor runs for a time equal to the "accumulationTime" and stops, ensuring the conveyor is completely filled with product.
- Queuing Conveyor (Zero Back Pressure)
 - Requirements: Requires a parent conveyor.
 - Sensors: 1 box detection sensor
 - Logic: When the parent conveyor stops, this conveyor continues running until the box detection sensor sees a box. Only one or two boxes can be sent to the downstream zone, depending on multipick.
- Follower Conveyor
 - Requirements: Requires a parent conveyor.
 - Logic: Replicates the motion of a parent conveyor. It moves when the parent moves.
- Simple Conveyor
 - Sensors: 1 box detection sensor
 - Logic: Stops if the IO is "ON" and runs if the IO is "OFF".
- Transfer Conveyor
 - Requirements: Requires a parent conveyor.
 - Sensors: 1 box detection sensor

- Logic: The transfer conveyor keeps moving until it detects a product. If the parent conveyor is running (free to accept a new product), then the pusher on the transfer conveyor pushes the new product onto the parent conveyor. The transfer conveyor restarts.

[Download Conveyor Process Library](#)