NORTH CAROLINA STATE UNIVERSITY

Introduction to Modern Industrial Automation

ISE 589-007 – Fall 2023

ETASK: Automated Assembly Line

**Project Final Report**

*Authors:* *Instructor:*

Ethan Houser Fred Livingston, PhD

Tabitha Gardner December 7, 2023

Kasra Jalilian Khave

Ángel Miranda

Zhongyi Shao

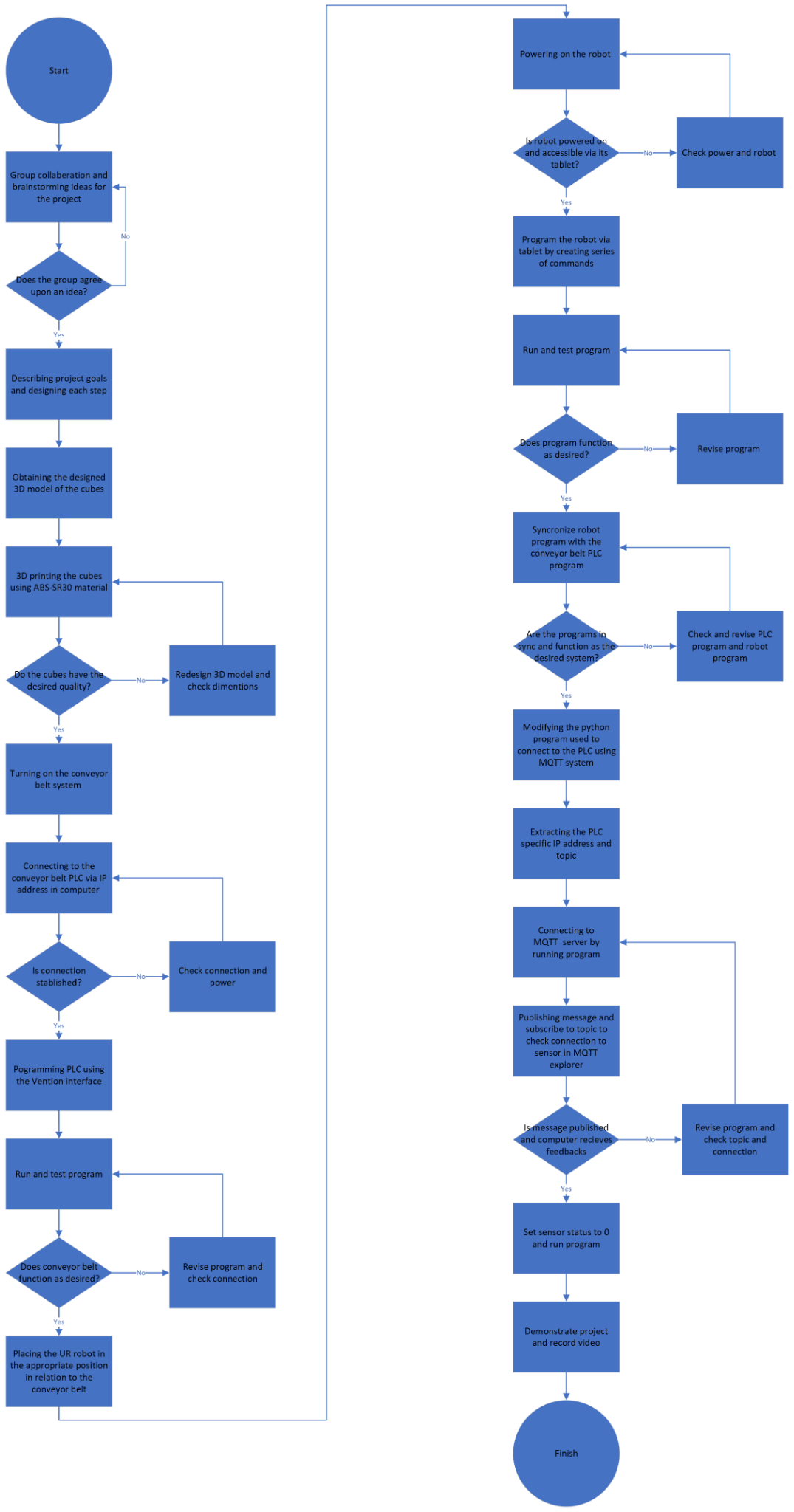


**Problem Statement**

The goal of this project was to replicate an automated assembly line by stacking three blocks to spell I-S-E. We manipulated the movement of the conveyor belt through programmable MachineLogic and developed a UR Robot sequence to perform a series of operations to stack letter blocks. Lastly, MQTT was utilized to record the start of the process and receive feedback on when the system is activated. These objectives successfully allow us to showcase our understanding of MQTT, IOT, UR robot manipulators, and Vention MachineLogic.

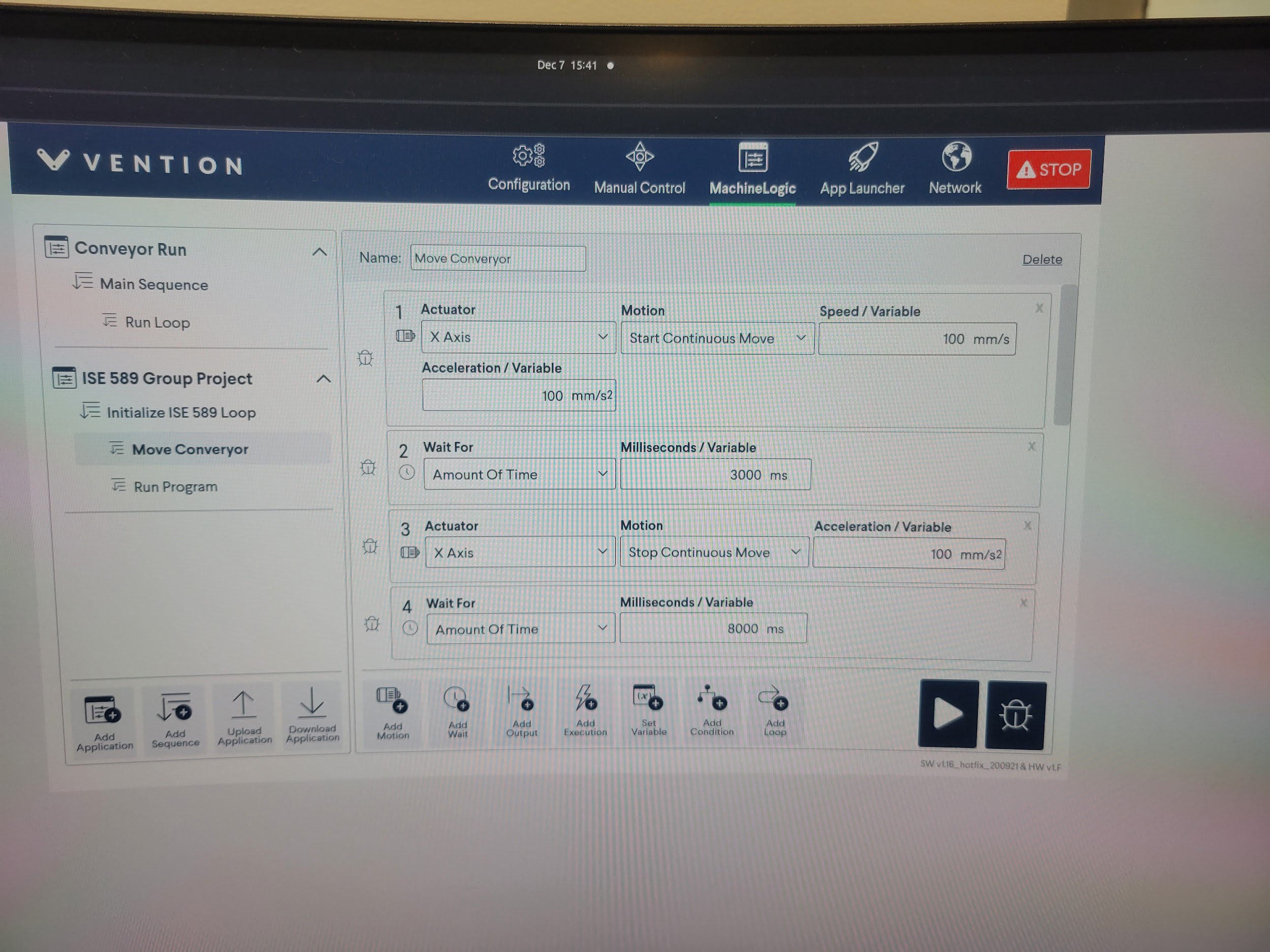
**System Design**

This project started with a group brainstorming session to determine what we would like to focus on in how we were going to execute the project. After determining our plan and meeting the checkpoints of creating our cubes, connecting to the conveyor belt, and learning how to use the UR robot, we were able to transition to a period of testing and editing to synchronize all aspects of our project. The last part of the project process was to implement MQTT and record our entire demonstration. Throughout the process of this project we faced many challenges that can be seen in the revision loops of our process, but eventually through help from the lab staff and many trails across the final weeks of the semester we were able to complete our project.

****

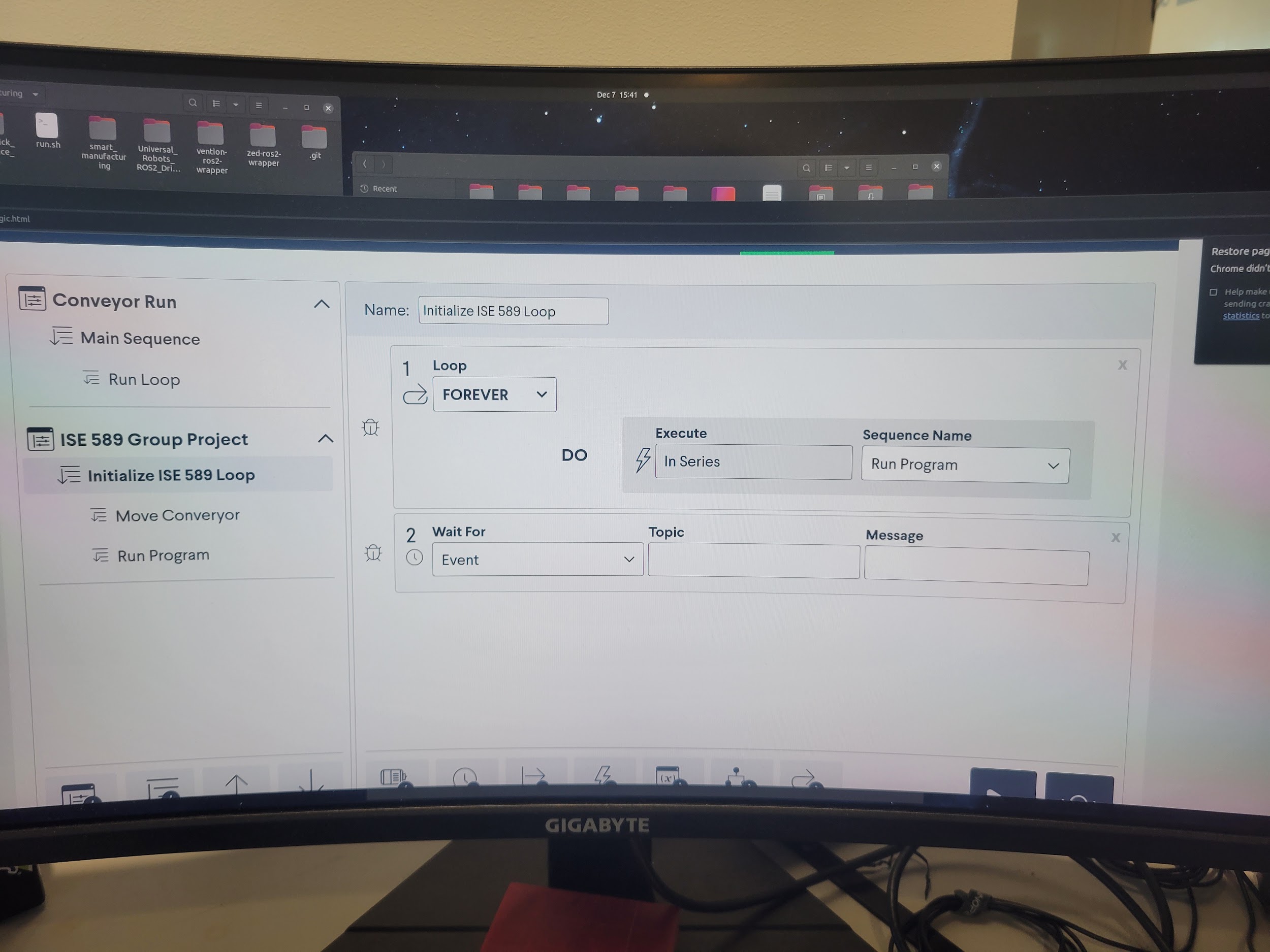
**Methodology**

With a system design in mind, our first objective after obtaining the 3D printed blocks was to get the conveyor belt running. Our system modeled a live assembly line, so we needed to program the conveyor belt to move in intervals that would allow enough time for the UR Robot to complete each assembly task. Once connected to the conveyor belt’s host IP address, we had access to a graphical user interface (GUI) which allowed us to access various functions of the conveyor system. In the GUI we created a program that would carry out all stages of our assembly process. The program itself was a series alternating between jogging the motor for 3 seconds to move the belt a desired distance and halting the belt for approximately 7-8 seconds, long enough for the robot to complete the assembly task at each station. This movement was programmed in Vention MachineLogic shown below. Figure 1, Line 1 shows our first command is to set the motor actuator to “Start Continuous Move” at 100 mm/s. This action will continue for the period defined in line 2 of three seconds. After this, the object arrives at station one where the actuator is defined to “Stop Continuous Move” for 8 seconds (Line 3&4). This sequence of starting and stopping the motor for defined intervals are repeated until the desired stations are cleared.



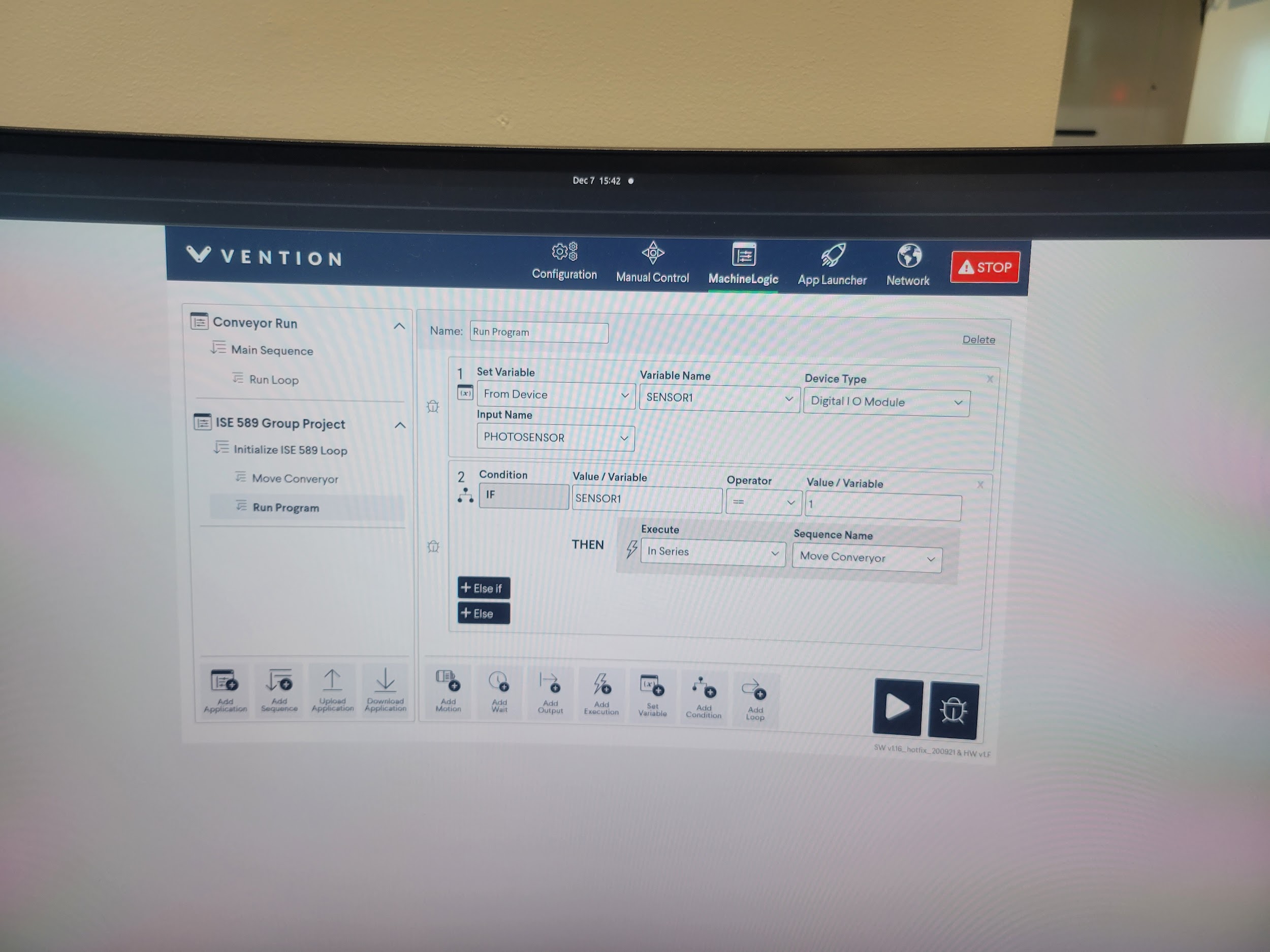
***Figure 1:*** *Move Conveyor MachineLogic*

Outside of the sequence used to design the movement of the conveyor belt, we also had to define the conditions on when to start the program to make it functional. To do this we created a stepped code that first defines a loop that will run forever. This loop as seen in Figure 2 will allow our program to be active but not run the conveyor until the sequence “Run Program” tells it to execute an action.



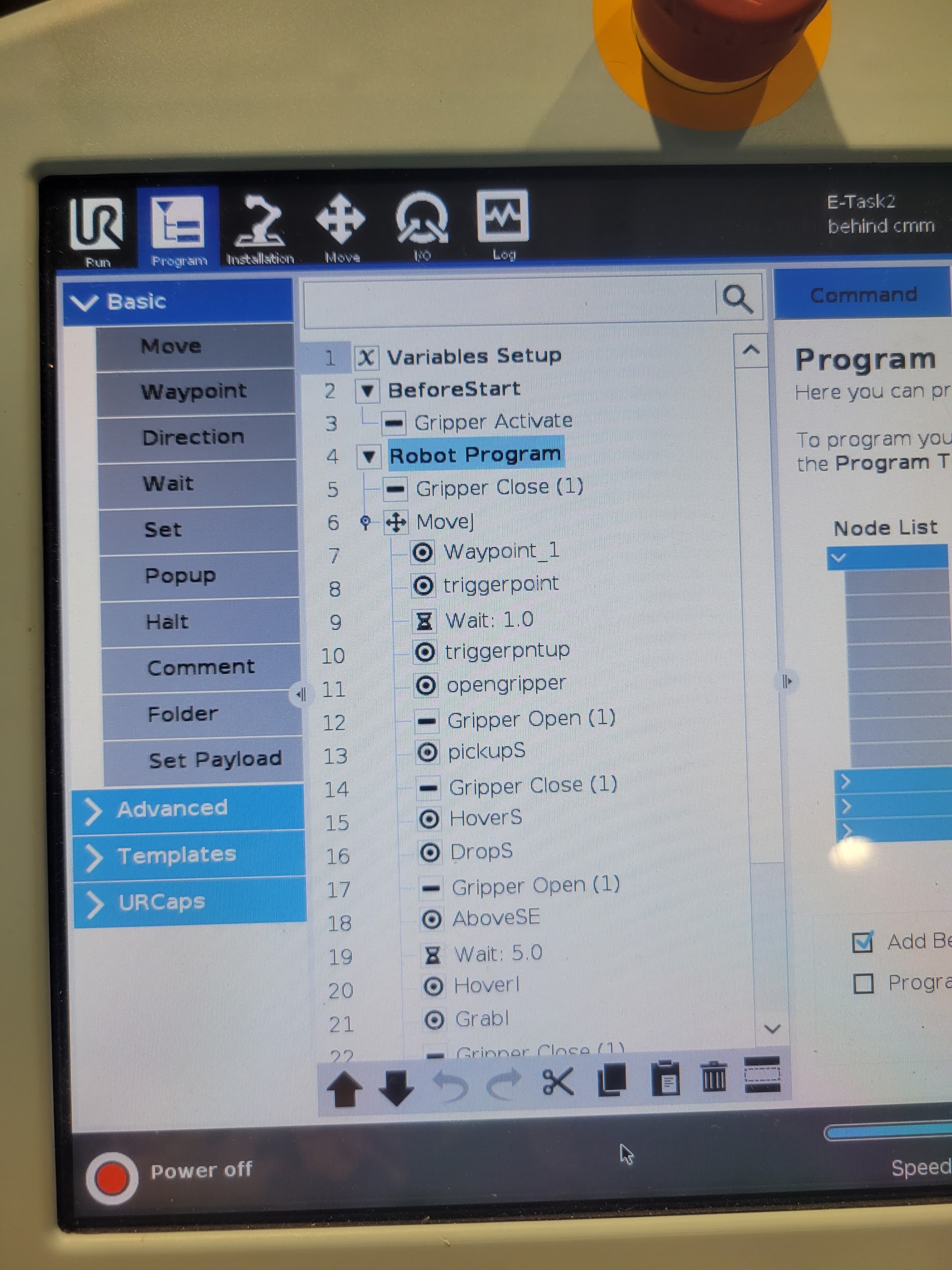
***Figure 2:*** *Initializing Step of MachineLogic*

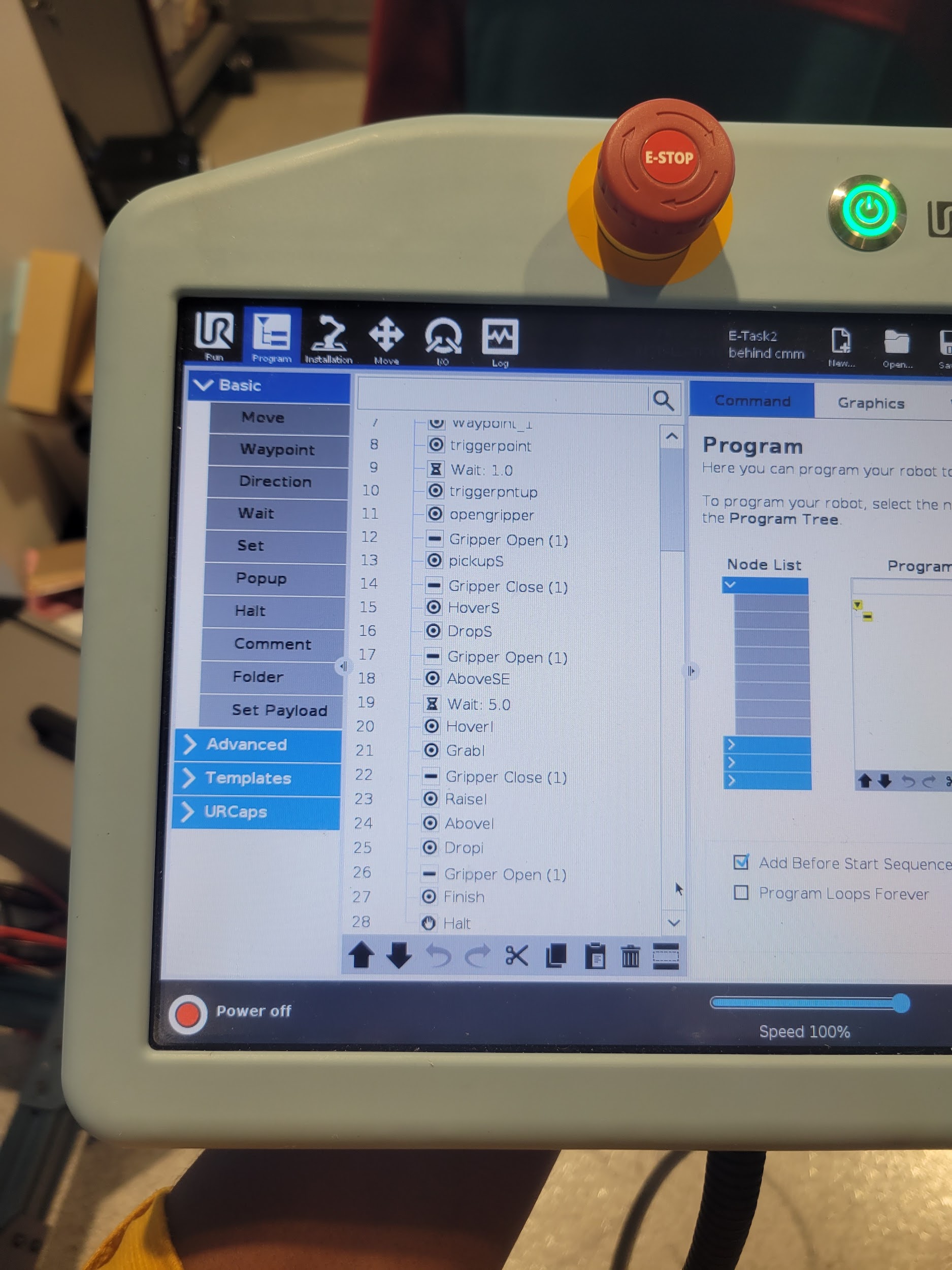
In the Run Program sequence shown in Figure 3, Line 1 we define the photosensor as an input variable and proceed to set up the conditions necessary for starting the movement sequence in Line 2. Line 2 states that if the sensor returns a value of 1 it should execute the move conveyor sequence therefore starting our conveyor belt assembly line. In effect, these levels of our code design the conveyor belt to continually check the value of the PHOTOSENSOR variable until it reads a value of 1 which triggers the conveyor belt program to begin. To synchronize the conveyor belt and UR Robot, we designed the UR robot arm to be the mechanism that trips the sensor and starts the process.



***Figure 3:*** *Run Program Condition for MachineLogic*

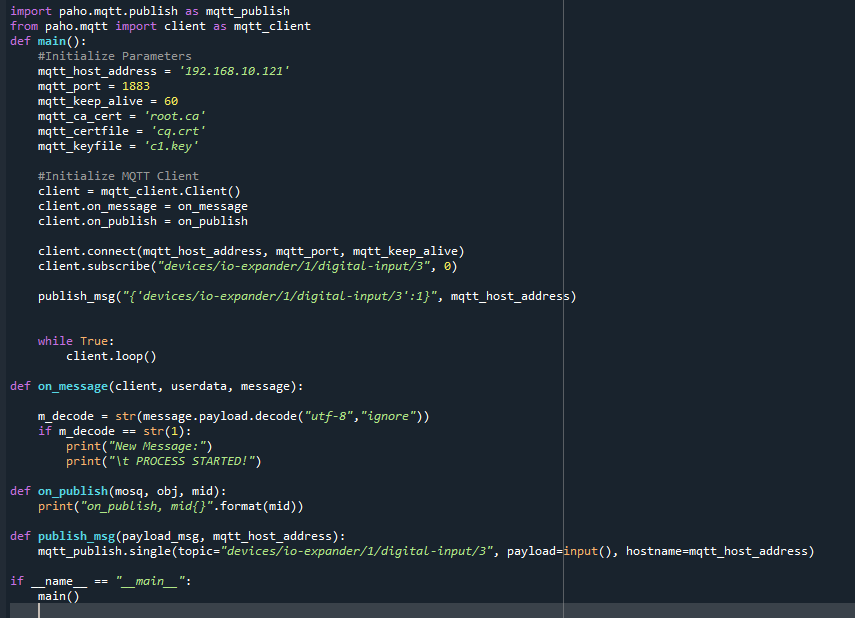
Our approach to the programming of the robot was to use the Vention machine tablet to create a series of commands to perform the task at hand. The designed series of commands included recording and saving different waypoints for the robot arm movement, closing and opening commands for the grip with the attention of grabbing and releasing the 3D designed cubes and wait commands to precisely sync the movement and actions in each stage with the conveyor belt for the system to function as a one. Furthermore, to determine and record each waypoint, we used a mixture of free-roam transformations of the robot and axis and orientational movement using the robot tablet. The first stage of the robot program, after triggering the conveyor belt sensor (Lines 7-10), was to position itself in the first stage of the assembly process. The UR Robot hovers over the S cube, picks it up, and places it over the E cube in the action sequences of lines 11-18. At this point, the first stage of the program is completed and the robot waits there for 5 seconds for the conveyor belt to move the blocks to the next stage (Line 19). Next, the UR Robot positions itself for the second stage of the assembly by hovering over the I cube, grabbing it, and positioning the cube on top of the assembled S and E cubes. The robot then opens the gripper to finish the second operation (Lines 20-26). After the second stage is completed, the robot is designed to go into a halt state for demonstration purposes only. By removing the halt command at the end of the program the robot would continue to execute the program in a loop which would imply that in a complete and real continuous assembly line, the robot has the capability to reproduce the desired product for mass assembly.

****

****

***Figure 4:*** *UR Robot Programming*

The final stage of our methodology was to implement an MQTT protocol. Once we connected to the conveyor belt’s private MQTT server, we had to identify the proper topic to subscribe to. After searching through possible topics, we were able to find the topic related to the onboard photo sensor used to trigger the conveyor belt system. The IP address and topic were then added to the Python code shown in Figure 5 to properly connect to the MQTT server and subscribe to the necessary topic. Aside from changing the topics to subscribe and publish to the sensor and editing the host address, we were able to largely rely on the Python logic explained in Lab 4.



***Figure 5:*** *MQTT Python Program*

Our python program allowed us to publish sensor readings via our Python program enabling the conveyor belt to be started virtually. However, our main point of focus for implementing MQTT was to establish a feedback system in our Python code. When the robot arm trips the sensor, a value of 1 is published to the sensor’s topic, and is received by the computer also subscribed by the Python’s interface.

**Conclusion**

The goal of this project was to replicate an automated assembly line by stacking three blocks to spell I-S-E. MQTT was utilized to record the start of the process. Then, while manipulating the movement of a conveyor belt, we used the UR Robot to pick up each block and place them in their corresponding order. At the completion of this project, we were able to successfully combine course topics to perform our tasks in unison. While each of these key course concepts provides researchers with essential knowledge and capability to perform automated tasks, combining their functionality has allowed us to unleash only a fraction of their true potential. Attached to our project submission, you will find a video demonstration of our project. Here you can see all the course concepts in action as our automated assembly line successfully constructs a tower of blocks to spell “I-S-E.”