BCIT SCHOOL OF ENERGY

Mechatronics and Robotics Engineering Technology

ROBT 4491: Mechatronics Project

FINAL PROJECT REPORT:
FESTO MPS II – REMOTE I/O

Produced By:	
Ashleen Bains	A01235466
Joshua Selikem	A01022430
Year:	Apr. 28, 2022
For:	Isaiah Regacho / Matthew Rockall

EDUCATION
FOR A COMPLEX WORLD.

BCIT

ACKNOWLEDGMENTS

BCIT's Mechatronics and Robotics program was an unforgettable experience for us. We would like to express our thanks and gratitude to the following people who have helped make this project possible.

To Chris Townsend for teaching us valuable machining skills that made the creation of the silo slips, rejection slide, and rejection slide mount possible.

To BCIT's Mediaworks for 3D printing the silo caps.

To BCIT's Print Shop for printing out our project cards and assembling our project poster.

To Brian Gaensbauer and Isaiah Regacho for helping us get started with our project, making the order of new parts possible, and their ongoing support.

Lastly, to Matthew Rockall for his continuous help with technical writing, which made this report possible.

ABSTRACT

This document details the software and hardware implementation of components, programming logic, and provides information on operating the system and its corresponding operation warnings. The objectives of this project were to:

- Control **2** stations
- Make the stations work independently of each other
- Stop the station where the problem has occurred
- Alert/notify the operator on an Human Machine Interface (HMI)
- Display what station needs servicing on the HMI
- Display what part of the station needs servicing on the HMI
- Restart the halted production line after the operator has confirmed that it is fixed

Upon completion of the project, the following objectives were reached:

- Control of **3** stations
- Program multiple state machines to carry out the wheel sorting process individually
- Make the PLC change its behaviour when a maximum wheel count is reached on all 3 stations
- Implement an HMI to carry out the tasks stated above

In the future we recommend students to attempt to implement true wireless communication, a way for the wheels to automatically load into the pneumatic lifts, relocate station 1's Emergency-Stop button, and to revive station 4 to a fully functioning station.

SAFETY & OPERATION WARNINGS

Please note the following warnings when operating the FESTO MPS II:

Hazard Awareness:

- Use of compressed air
- Moving components and pinch points

Developer Mode:

When using the developer mode, the user must be aware that the station's workspace should be checked for any objects that could be in the way of the moving parts of the system. For the gantry of each station, it is recommended that the gantry is kept in in between the 3 silos and the wheel pickup location. **IF THIS WARNING IS DISREGARDED, UNDESIRED BEHAVIOUR WITHIN THE SYSTEM WILL OCCUR.** To FIX this issue of the gantry being stuck outside of the perimeter mentioned above the following steps must be taken:

1. Open the ACD file on studio 5000 and go to the tags of the corresponding station
2. Find the X and Y limit switch tags for the limit switches
3. Enter the desired number (1 or 0) of the forced mask needed to move the gantry in the opposite direction.
4. Enable I/O forces for the controller.
5. Move the gantry back within the perimeter mentioned above. When finished, disable **ALL** I/O forces.

Emergency Stop:

In the event of an emergency the E-Stop can be pressed to halt the station completely. To start it back up again, release the E-Stop and press the start button again.

TABLE OF CONTENTS

List of Figures

List of Tables

1 INTRODUCTION

This report has been written to outline the objective of the project, the different subsystems needed to make the project function, how functionality was obtained, and how to operate it. The report covers details regarding the use of a programmable logic controller (PLC) to sort different colored objects, and how the program was designed to make similar workspaces work asynchronously while also being dynamic as one system. This report will also periodically refer to students who previously worked on the project, but not the specific details thereof.

2 SYSTEM OVERVIEW

The Festo MPS's job is to sort three different types of wheels into three separate silos. The types of wheels consist of black plastic wheels, orange plastic wheels, and metal wheels. All wheels must be the same height. The system is composed of three stations that carry out the same process. To achieve the process, the Festo MPS must complete the following tasks in the corresponding order:

1. Determine the wheel's material via capacitive, inductive, and optical sensors
2. Move the wheel up to the conveyor's level via a pneumatically controlled lift
3. Determine the wheel's height via an analog distance sensor

* If the wheel is the correct height, the process will continue as:

4. Push the wheel onto the conveyor via a pneumatic cylinder
5. Move the wheel to a pick-up zone via conveyor
6. Sense that the wheel is in the pick-up zone via a capacitive sensor
7. Move the gantry arm to the pick-up location
8. Pick up the wheel via pneumatics and the arm's suction cup
9. Move the wheel to its corresponding silo
10. Drop the wheel into the silo
11. Restart the Process

* If the wheel is not the correct height, the process will continue as:

4. Move the wheel back down to the base level via the lift
5. Discard the wheel by pushing it off the lift via a pneumatic cylinder
6. Restart the Process

2.1 SYSTEM REQUIREMENTS

Initially when handed the project, most of the individual components for each station were already setup for 1 station and partially setup for the next 2 stations; so, most of the requirements would be with regards to software implementation. The system will require a single controller that will have stable control over 3 stations individually. In order for the PLC to know when to dynamically change its behaviour, faults must be checked and acted upon immediately after detection. Each station will be sorting out different objects and a single interface will be needed to communicate with the Controller, observe visually what is occurring on each station, and check if anything is broken on each station. The system requirements are as follows:

Table 1: Full System Requirements

Requirement	**Detail**
Station Individuality	Each station must run autonomously
Fault Detection	Faults must be detected to change system behaviour
Object Categorization	Objects need to be sorted
User Interface (HMI)	User needs to be able to interact with Controller (Beijer Electronics)
PLC	Allen Bradley CompactLogix L30ER
Point I/O	Allen Bradley 1734-AENTR Series-B

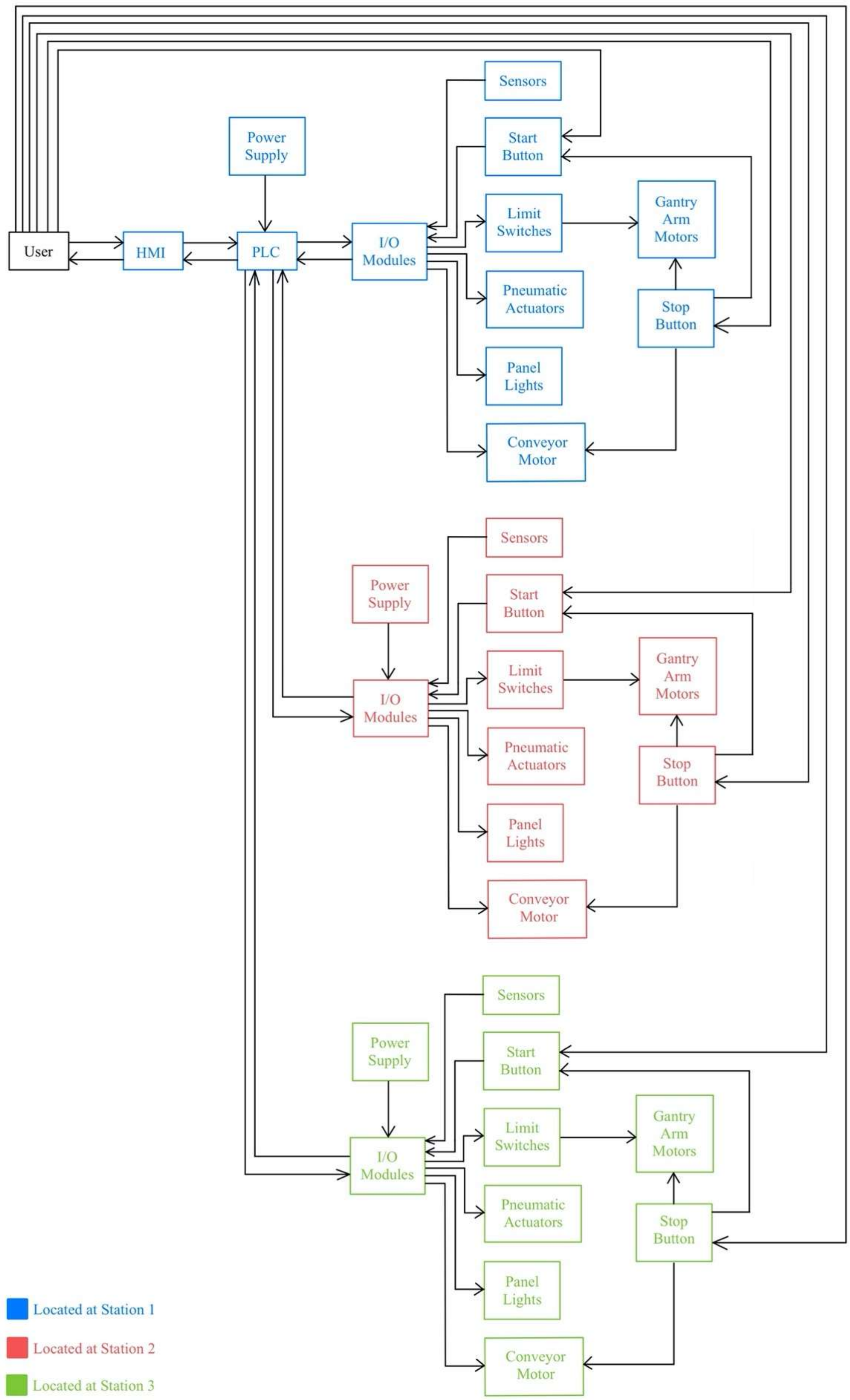

Figure 1: System Block Diagram

2.2 PLC

The PLC is the brain of the Festo MPS. The PLC uses ladder logic to control various outputs based off its incoming inputs. The PLC outputs are used to control pneumatic actuators, motors, the HMI, and lights. The PLC inputs come from pushbuttons, the HMI, and digital and analog sensors.

2.3 HMI

The HMI carries out all the necessary communication between the operator and PLC, which controls the three stations. The HMI provides the operator with virtual versions of buttons, information of each station, and quick access control over the PLC's outputs and certain inputs.

2.4 Power Supply

On each Festo MPS station, there are 24V power supplies. Each supply provides the necessary voltage to the station's relays, motors, pushbuttons, inductive sensors, capacitive sensors, optical sensors, and proximity sensors.

2.5 I/O Modules

Each station contains point I/O modules by Allen Bradley. Connecting inputs such as push buttons or outputs such as a motor, enable the PLC to properly interface with them when programming on Studio 5000.

2.6 Sensors

The sensors are connected to the PLC's inputs and are used to determine height, colour, and position. Sensor input is crucial for the PLC to determine its next decision. All the sensors connected to a station are digital, except for one analog senser, which is used to determine a wheel's acceptable height. A schematic of the sensors is shown in section 3.2.1, Figure 3.

2.7 Start Button

After putting the PLC in run mode, the start button is what commences the operation of the system. A virtual option of the start button can be found on the home page of the PLC's HMI.

2.8 Limit Switches

On each Festo MPS station, there are 4 limit switches. 2 of these switches control the x-axis motor of the gantry arm and the other 2 control the y-axis motor of the gantry arm. The limit switches are outputs of the PLC that are used to control the position of the gantry arm. They do this by energizing a solenoid that controls a relay, which allows power to be supplied to the motor. A schematic of the limit switches is shown in section 3.2.1, Figure 3.

2.9 Gantry Arm Motors

There are two gantry arm motors per station. One motor is used to control the x-axis and the other controls the y-axis. These motors are controlled by relays, which are controlled by solenoids that are energized by the station's limit switches. A schematic of the gantry motors can be found in Appendix A, Figure 5.

2.10 Pneumatic Actuators

The pneumatic actuators are controlled by the outputs of the PLC. They use the air supplied to the station to make components move. The pneumatically controlled lift is used to move the wheel up and down levels. The pneumatic cylinder is used to push the wheel off the lift. The gantry arm's pneumatic cylinder is used to bring the arm closer to the wheel. Lastly, the gantry arm's pneumatic vacuum is used to pick up the wheel via suction.

2.11 Panel Lights

The panel lights are used to communicate to an operator the states of an MPS station. The green light tells the operator when the corresponding station is running (Ex. If station 1 was running, the green light on station 1 would be on). The orange light is used to tell the operator that the stations have met the maximum wheel count for a colour. Lastly, the red light is used to tell the operator when the emergency stop button has been hit or when a fault has occurred on the corresponding station.

2.12 Conveyor Motor

The conveyor motor is used to move the conveyor belt so that a wheel can move from the pneumatic lift to the gantry arm. The motor is controlled by a relay. This relay is controlled by a solenoid that is energized by a PLC output. A schematic of the conveyor motor can be found in Appendix A, Figure 5.

2.13 Stop Button

The emergency stop button is a normally closed pushbutton that controls the corresponding station's conveyor motor, gantry arm motors, start button, and green panel light. It does this by disabling the functionality of the start button and disabling the outputs of the gantry arm's limit switches, the conveyor motor, and the green light.

3 PLC

The PLC is the controller of the Festo MPS. It controls the three stations by connecting to each station's Point I/O adapter module via EtherNet/IP. Based off the ladder logic written to the PLC, the PLC uses the sensor input from each remote I/O module to decide the current state of each station's outputs. The PLC communicates to each remote I/O module to turn on and off certain outputs.

3.1 PLC Requirements

Table 2: PLC subsystem requirements

Requirement	**Detail**
PLC	Allen Bradley PLC
Point I/O adapter module	Each station needs its own I/O module
Digital input cards	Cards to read inputs
Digital output cards	Cards to read write to outputs

Remote I/O communication	Communication must be established from each station back to the PLC
Analog input card	Cards to read analog inputs
PLC software	Programmed with ladder logic
PLC software structure	Programmed with state method 3
Project solution implementation	Colour sorting algorithm
Implementation of fault checking	To determine when a system is malfunctioning
Implementation of an emergency state	To make everything cease from working

3.2 PLC Design

The PLC is a product that comes from Allen Bradley, the design of the PLC with the rest of the systems had been implemented by previous students. To be safe, the PLC and its connections were checked beforehand to make sure things would operate as expected.

3.2.1 Hardware Design

The PLC connects to a special Network Switch that uses star topology to connect to the 3 Point I/O module adapters. The diagram below shows how the PLC is connected to each station via EtherNet/IP (Figure 2).

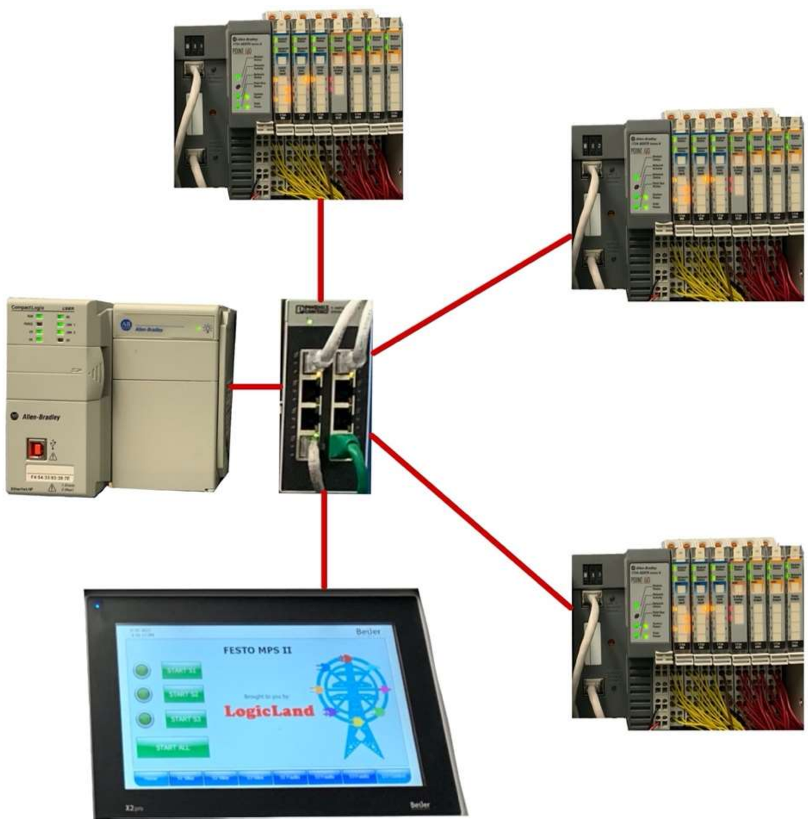

Figure 2: Diagram of the PLC's Network Connections to the Remote I/O Modules and HMI

The PLC I/O module connections are the same for each station. Below is a schematic diagram (Figure 3) of the I/O connections for a station. Note that "S#" stands for station #, where # ranges from 1 to 3 since there are 3 stations.

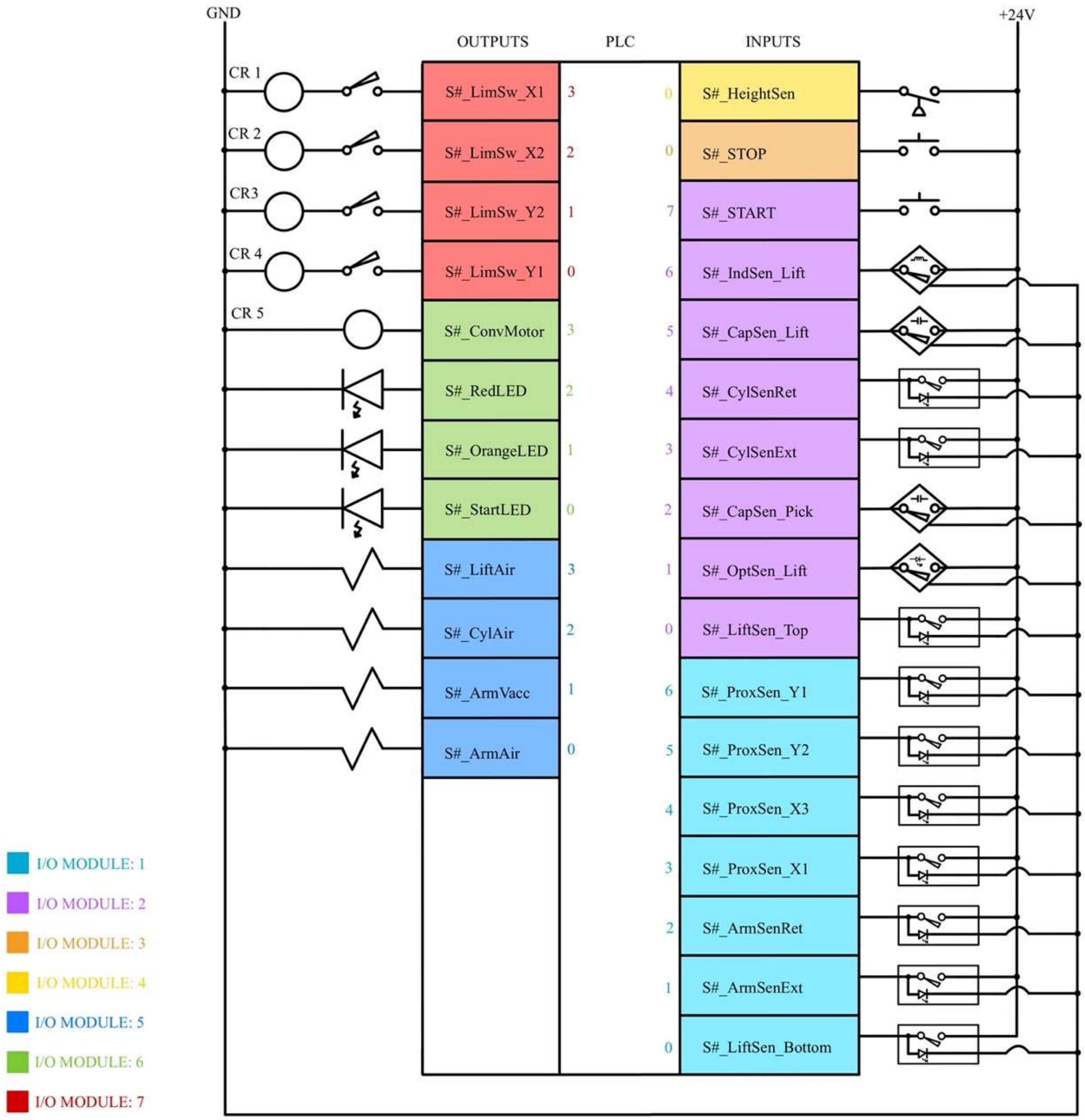

Figure 3: PLC I/O Connections

3.2.2 SOFTWARE DESIGN

The PLC's program was written with ladder logic using state method 3 in Studio 5000. The main operation describing how a wheel is determined acceptable and placed into its correct silo is shown in black in the state diagram below (Figure 4). The diagram also includes the event of entering developer mode in blue, as well as for faults and the E-stop in red. More information regarding the state diagram's entry conditions, actions, and exit conditions for each state can be found in Appendix B.

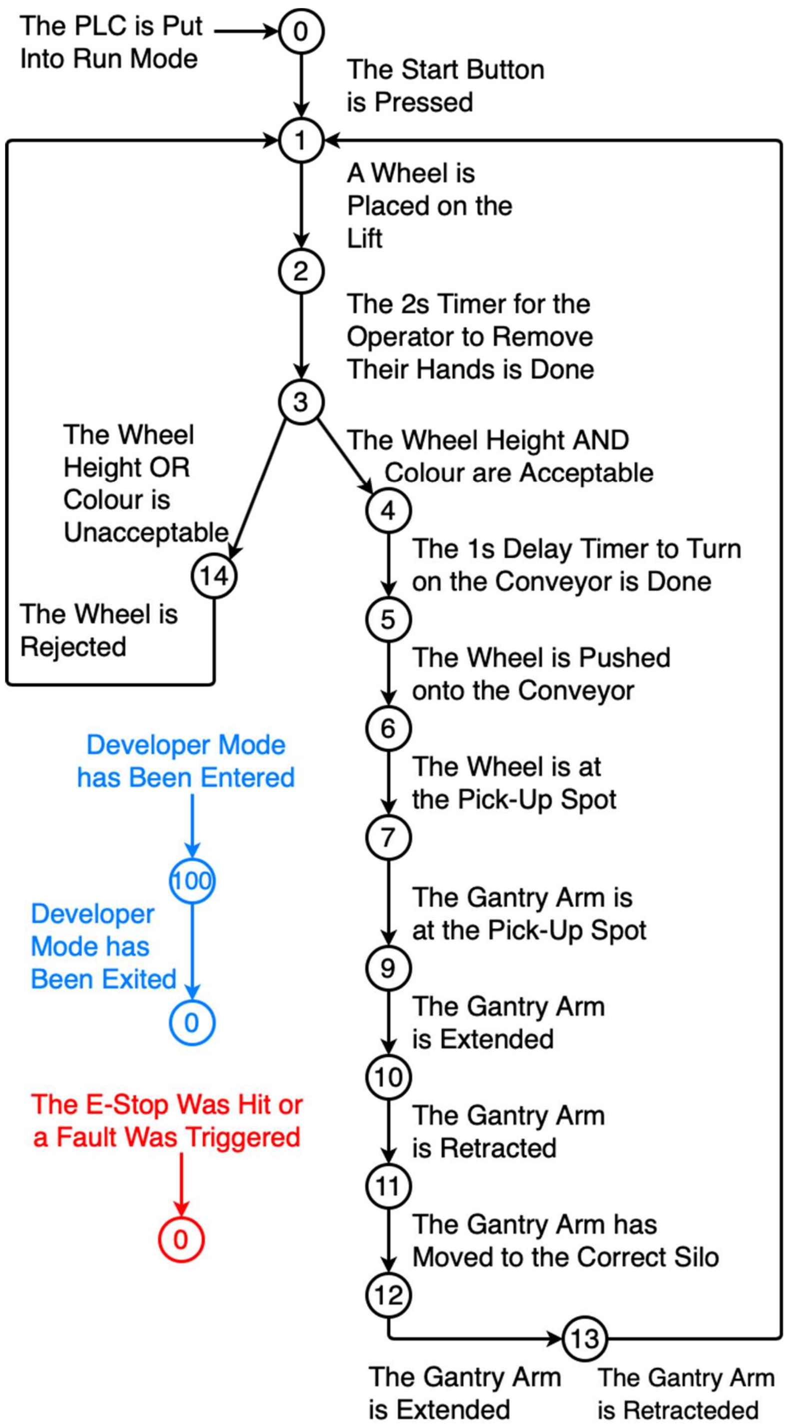

Figure 4: State Diagram of the Festo MPS II

3.3 PLC IMPLEMENTATION

The objective of implementing the project solution of the Festo MPS with the PLC was mostly done with software, but some components on station 3 required our attention in order to look and operate the same as the other two stations.

3.3.1 SOFTWARE IMPLEMENTATION

The full Festo MPS II code for the PLC can be found on LogicLand's repository on GitHub. The link to the source code is pasted below. The file called "MultiStationMaster_Version12.ACD" is the final version of the Festo MPS II code. There are also PDFs of the code on the repository in a folder called "Routine PDFs May 2-2022".

https://github.com/BCITMechatronics/LogicLand

To briefly summarize how its running, the PLC is running a 2 specific tasks, one continuous main task and one periodic fault task. The fault task runs every 75 milliseconds and once it has been scanned through, it goes back to the main task. The main task contains all the core ladder logic, it is separated into 4 programs, a program for HMI I/O tags and the other 3 programs represent the 3 stations that are being controlled. Each station's program consists of subroutines for inputs, outputs, logic, and various others to help with the sorting logic. Most of the stations' tags are contained locally in their respective programs, but tags regarding faults have been relocated to the Controller Scope level for the HMI to be able to access.

3.3.2 HARDWARE IMPLEMENTATION

Two key hardware implementations that were developed during the project were the N.C Stop Push Button, and the Capacitive Sensor.

The stop button for station 1 and station 2 were implemented by previous students, but station 3 was still using is original stop button, which presented a hazard to anyone using the station because of all the exposed wires. To mitigate this risk, an electrical junction box was purchased and installed. This enabled a much more proper mounting of the display lights to the station. The E-Stop was then separated from the lights just like the other stations. Based off some observations, the stop buttons on the other stations had a hardware implementation and an indicator in an Input module so that it could be implemented in software as well. When the stop button was hit the gantry motors and the green light on the station would cease to work; the same wiring logic was placed into station 3.

The capacitive sensor on station 1 right away looked different from all the other stations and the PLC was sensing the 3 colours of wheels in a different way compared to the other 2 stations. To make everything the same across all the stations, the sensor was replaced with one from station 4. This gave the exact expected results when debugging the code.

Any other hardware implementation (with regards to the PLC) was directed at station 3 and these changes are listed below:

- Replacement of analog distance sensor
- Replacement of inductive sensor
- Replacement of pneumatic tubing
- Replacement of conveyor rubber grips
- Replacement of gantry limit switches
- Replacement of stop button and lights (as mentioned above)

4 SUBSYSTEM HMI

The Human Machine Interface (HMI) is an interface that enables the user to interact with the PLC without the need of looking into its source code. The goal was to create an interface that would display the number of objects that had been sorted by the system, display the issues of each individual station, to be able to clear the issues, and alert the PLC that the user had emptied out the silos (the containers holding the sorted objects). Also, the want for a developer page for the user to test different components on the stations was desired.

4.1 HMI REQUIREMENTS

Table 33: HMI subsystem requirements

Requirement	Detail
Manufacturer	Beijer Electronics X2 Pro 10
Station Display	Each station needs its own visual display
Fault Display	A section to check for any faults
Developer Display	A section to test components

4.2 HMI IMPLEMENTATION

Normally for Allen Bradley products they have an application called Factory Talk to design HMIs developed by Allen Bradley themselves and even some other foreign products. The software implementation of our Beijer Electronics HMI required a separate application called IX Developer to design and link the PLC tags. The hardware implementation was previously installed from previous students.

4.2.1 SOFTWARE IMPLEMENTATION

A total of 11 main screens were designed to fulfill the vision for an interactive HMI between the operator and the PLC. As seen in Appendix C, Figure 6, the home screen enables the operator to start each station individually or start all stations with the press of one button. The next 3 screens are to display the number of wheels in each corresponding silo. The fault screens display indicators as to which station has a fault and the specific location of where the fault is located in the station's workspace.

The last menu implemented is the Developer Mode. This password restricted menu has been created for the purpose of testing out or manually actuating components in each of the 3 stations. Since the stations have been programmed to work independently from each other; entering the developer mode of a single station will not impact the processes of the other 2 stations. This excludes the editing of the number of wheels in the stations, as that could make the other stations start rejecting or accepting the different colored wheels.

(Please note the warning in beginning of report about Developer Mode)

4.2.2 HARDWARE IMPLEMENTATION

The HMI had already been mounted onto Station 1 by the previous group, but to understand how it was powered and how to make it communicate with the PLC was still needed.

The HMI is powered by a separate power supply mounted on the rear of Station 1's component rack. It supplies the HMI with 24VDC and contains a RS232 port and 2 ethernet ports. For the project, only one ethernet port which is connected to a Network Switch, to communicate effectively with our PLC was required. The HMI's connections to the Network Switch and PLC can be seen in section 3.2.1, Figure 2.

5 SYSTEM VERIFICATION

After successfully putting together this project, the process was initiated over multiple iterations to determine if the system was working as intended. As mentioned in the Request for Proposal, the PLC was desired to behave dynamically, wherein it would change how it processed wheels if one of the stations were to fail. To verify this specific functionality, one or more stations were placed into a faulted state; this would render the entire station inoperable so that the other station(s) could be observed.

6 PROJECT SUMMARY

The goal of the Festo MPS was to sort 3 different colours of acceptable sized wheels into their corresponding silo. Originally the project objective was to get 2 stations to achieve the goal by working asynchronously to sort a shared total count, but the addition of the 3rd station was achieved.

6.1 SYSTEM SPECIFICATIONS

Based off the work done to the Festo MPS, the project goal and objective were achieved with the following specifications:

- 3 Stations with their own remote I/O adapter connected to one PLC via EtherNet
- Each station works asynchronously to sort a shared maximum count of 6 wheels per colour between all the stations
- Wheels come in 3 different sizes: small, medium, and large
- Only medium sized wheels will be sorted, the rest are rejected
- Wheels come in 3 different colours: black, orange, and silver
- An HMI located on station 1 can start any station, view any station's silo count, display any station's faults, and manually control certain I/O for any station

6.2 RECOMMENDATIONS/FUTURE WORK

Although the project requirements were met, there is still room for improvement on the Festo MPS. Since this system is expanded upon every year, the following suggested items can be implemented:

The Repositioning of Station 1's E-Stop

In the event of an emergency, the E-Stop should be pressed quickly and easily to prevent further damage or danger. Currently, station 1's E-Stop is located behind the HMI. This makes accessing the E-Stop hard to do. By relocating the E-Stop to a much more open area, it would provide a much safer environment.

Automatic Wheel Loading

After pressing the start button, an operator needs to manually place a wheel onto a station's pneumatic lift to continue the process. By implementing a system to automatically load wheels into the station's lift, it can greatly increase the production time of the Festo MPS.

The Revival of Station 4

In order to get station 3 up and running, station 4 was used to provide station 3 with spare parts. This left station 4 in a bit of a mess. By fixing up station 4, the Festo MPS's throughput will massively increase.

True Wireless Communication

Currently the Festo MPS stations are connected via ethernet to the special Network Switch. With true wireless connection, operators would be able to move stations more conveniently and easily since they won't be tied together by EtherNet cables.

7 Conclusion

The Festo MPS II was an excellent project to work on. The current state of the project has no major issues, but there is always room for improvement! The major issues that were faced with this project mainly occurred on station 3. Some of those issues include faulty/broken sensors, a broken limit switch, missing parts. To fix these issues, parts were either machined or scrapped from station 4. More information and images of the machined parts can be found in Appendix D. With the structure of each station, alignment was a key thing to watch out for as well. When the project was moved to the Expo for demonstration, the vibrations from moving the project changed the structural alignment as well as the analog sensor's values; of which both were re-calibrated. The project overall was successful, the project solution was implemented and executed as was visioned.

8 Operation of System

A few steps can be taken to start the system:

1. Ensure the PLC, HMI, and each station's point I/O is supplied power.
2. Ensure each station is supplied with air by checking their corresponding gauges; after this verification, the shut-off valve can be opened by pulling it straight downward.
3. When no objects or hazards are in the workspace of each station, the green pushbutton can be pressed to turn on the station or the HMI's start buttons can be used as well.
4. Place a wheel at the lift where the 3 sensors are located, and the station will carry out its task.

In the event of an emergency the E-Stop can be pressed to halt the station completely. To start it back up again, release the E-Stop and press the start button again.

APPENDIX A: MOTOR SCHEMATIC

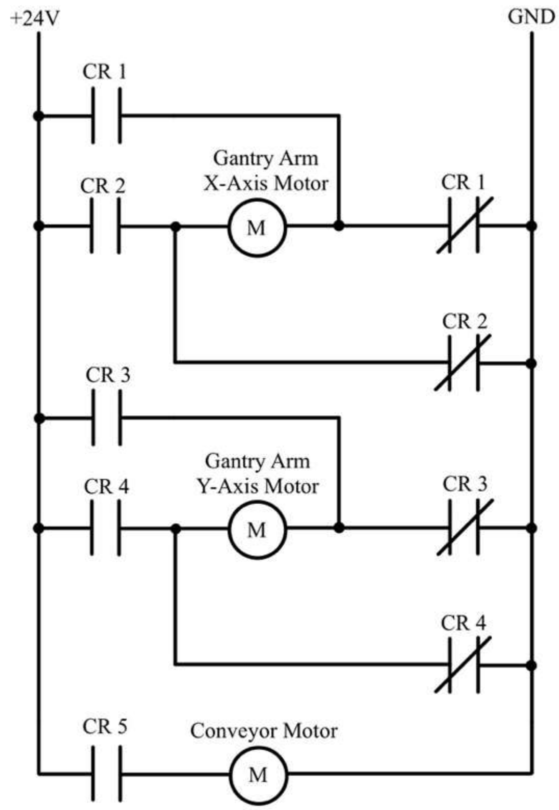

Figure 5: Schematic Diagram of Gantry Arm Motors and the Conveyor Motor

Appendix B: State Machine Details

State 0: Idle Until Started

Entry Conditions:

- The PLC is put into RUN mode
- From state 100 (Developer Mode), the ‘Back’ button or ‘Logout’ button on the HMI has been pressed
- From any state, the E-Stop was pushed
- From any state, a fault has been triggered

Actions:

- Move the gantry arm to the ‘Pick Spot’
- If the E-Stop was pushed or a fault has been triggered, turn on the red LED
- If the E-Stop was pushed or a fault has been triggered, disable all motors

Exit Conditions:

- To state 1, the start button has been pushed, the station’s corresponding virtual start button was pushed on the HMI, or the virtual start all button was pushed on the HMI
- To state 100, the corresponding station’s control button has been pressed after logging in for developer mode on the HMI

State 1: Idle Until a Part is Placed on the Lift

Entry Conditions:

- From state 0, the start button has been pushed, the station’s corresponding virtual start button was pushed on the HMI, or the virtual start all button was pushed on the HMI
- From state 13, The gantry arm’s retraction proximity sensor is activated
- From state 14, The lift’s bottom proximity sensor is activated, and the lift cylinder’s extension proximity sensor is activated

Actions:

- Turn on the green LED
- Move the gantry arm to the ‘Pick Spot’

Exit Conditions:

- To state 0, the E-Stop was pushed, or a fault has been triggered
- To state 100, the corresponding station’s control button has been pressed after logging in for developer mode on the HMI
- To state 2, the capacitive sensor on the lift is activated

State 2: Determine the Wheel Colour

Entry Conditions:

- From state 1, the capacitive sensor on the lift is activated

Actions:

- Turn on the green LED
- Move the gantry arm to the ‘Pick Spot’
- Start the 2 second elevator delay timer

- If the lift's inductive, optical, and capacitive sensors are all activated, move 3 into wheel_colour
- If only the lift's optical and capacitive sensors are activated, move 2 into wheel_colour
- If only the lift's capacitive sensor is activated, move 1 into wheel_colour

Exit Conditions:

- To state 0, the E-Stop was pushed, or a fault has been triggered
- To state 100, the corresponding station's control button has been pressed after logging in for developer mode on the HMI
- To state 3, the 2 second elevator delay timer is done

State 3: Determine if the Wheel Height is Acceptable or not

Entry Conditions:

- From state 2, the 2 second elevator delay timer is done

Actions:

- Turn on the green LED
- Move the gantry arm to the 'Pick Spot'
- Start the 2 second lift rise timer
- Jump to the HeightCheck subroutine
- Raise the lift to the conveyor's level if the total colour count for the wheel placed on the lift is less than 6

Exit Conditions:

- To state 0, the E-Stop was pushed, or a fault has been triggered
- To state 100, the corresponding station's control button has been pressed after logging in for developer mode on the HMI
- To state 4, the lift's top proximity sensor is activated, the 2 second lift rise timer is done, and the Height_accepted_bit is activated
- To state 14, the total colour count for the wheel placed on the lift is equal to or greater than 6
- To state 14, the lift's top proximity sensor is activated, the 2 second lift rise timer is done, the Height_accepted_bit is not activated, and the Height_rejected_bit is activated

State 4: Turn on the Conveyor

Entry Conditions:

- From state 3, the lift's top proximity sensor is activated, the 2 second lift rise timer is done, and the Height_accepted_bit is activated

Actions:

- Turn on the green LED
- Move the gantry arm to the 'Pick Spot'
- Raise the lift to the conveyor's level if the total colour count for the wheel placed on the lift is less than 6
- Start the 1 second conveyor delay timer
- Turn on the conveyor motor

Exit Conditions:

- To state 0, the E-Stop was pushed, or a fault has been triggered
- To state 100, the corresponding station's control button has been pressed after logging in for developer mode on the HMI

- To state 5, the 1 second conveyor delay timer is done

State 5: Push the Wheel onto the Conveyor

Entry Conditions:

- From state 4, the 1 second conveyor delay timer is done

Actions:

- Turn on the green LED
- Move the gantry arm to the 'Pick Spot'
- Raise the lift to the conveyor's level if the total colour count for the wheel placed on the lift is less than 6
- Turn on the conveyor motor
- Extend the lift's cylinder

Exit Conditions:

- To state 0, the E-Stop was pushed, or a fault has been triggered
- To state 100, the corresponding station's control button has been pressed after logging in for developer mode on the HMI
- To state 6, the lift cylinder's extension proximity sensor is activated

State 6: Move the Wheel to the Pick Spot

Entry Conditions:

- From state 5, the lift cylinder's extension proximity sensor is activated

Actions:

- Turn on the green LED
- Move the gantry arm to the 'Pick Spot'
- Turn on the conveyor motor

Exit Conditions:

- To state 0, the E-Stop was pushed, or a fault has been triggered
- To state 100, the corresponding station's control button has been pressed after logging in for developer mode on the HMI
- To state 7, the capacitor sensor at the 'Pick Spot' is activated

State 7: Make Sure the Gantry Arm is at the Pick Spot

Entry Conditions:

- From state 6, the capacitor sensor at the 'Pick Spot' is activated

Actions:

- Turn on the green LED
- Move the gantry arm to the 'Pick Spot'

Exit Conditions:

- To state 0, the E-Stop was pushed, or a fault has been triggered
- To state 100, the corresponding station's control button has been pressed after logging in for developer mode on the HMI
- To state 9, the gantry arm's X3 and Y2 proximity sensors are activated, indicating that the arm is at the 'Pick Spot'

State 9: Extend the Gantry Arm to Grab the Wheel

Entry Conditions:

- From state 7, the gantry arm's X3 and Y2 proximity sensors are activated, indicating that the arm is at the 'Pick Spot'

Actions:

- Turn on the green LED
- Extend the gantry arm
- Turn on the gantry arm's vacuum

Exit Conditions:

- To state 0, the E-Stop was pushed, or a fault has been triggered
- To state 100, the corresponding station's control button has been pressed after logging in for developer mode on the HMI
- To state 10, the gantry arm's extension proximity sensor is activated

State 10: Retract the Gantry Arm While Gripping the Wheel

Entry Conditions:

- From state 9, the gantry arm's extension proximity sensor is activated

Actions:

- Turn on the green LED
- Turn on the gantry arm's vacuum

Exit Conditions:

- To state 0, the E-Stop was pushed, or a fault has been triggered
- To state 100, the corresponding station's control button has been pressed after logging in for developer mode on the HMI
- To state 11, the gantry arm's retraction proximity sensor is activated

State 11: Move the Gantry Arm with the Wheel to its Correct Silo

Entry Conditions:

- From state 10, the gantry arm's retraction proximity sensor is activated

Actions:

- Turn on the green LED
- Turn on the gantry arm's vacuum
- If wheel_colour = 1, jump to the Black_Sorting subroutine to move the arm to the black silo
- If wheel_colour = 2, jump to the Red_Sorting subroutine to move the arm to the orange silo
- If wheel_colour = 3, jump to the Metal_Sorting subroutine to move the arm to the blue silo

Exit Conditions:

- To state 0, the E-Stop was pushed, or a fault has been triggered
- To state 100, the corresponding station's control button has been pressed after logging in for developer mode on the HMI
- To state 12, the arm has finished moving to the correct silo

State 12: Extend the Gantry Arm to Place the Wheel and Increment the Wheel Colour Count

Entry Conditions:

- From state 11, the arm has finished moving to the correct silo

Actions:

- Turn on the green LED
- Turn on the gantry arm's vacuum
- If wheel_colour = 1, increment the station's black wheel count by 1
- If wheel_colour = 2, increment the station's orange wheel count by 1
- If wheel_colour = 3, increment the station's metal wheel count by 1
- Extend the gantry arm
- Turn on the gantry arm's vacuum

Exit Conditions:

- To state 0, the E-Stop was pushed, or a fault has been triggered
- To state 100, the corresponding station's control button has been pressed after logging in for developer mode on the HMI
- To state 13, the gantry arm's extension proximity sensor is activated

State 13: Retract the Gantry Arm After Placing the Wheel

Entry Conditions:

- From state 12, the gantry arm's extension proximity sensor is activated

Actions:

- Turn on the green LED

Exit Conditions:

- To state 0, the E-Stop was pushed, or a fault has been triggered
- To state 100, the corresponding station's control button has been pressed after logging in for developer mode on the HMI
- To state 1, the gantry arm's retraction proximity sensor is activated

State 14: Reject Unacceptable Heights or Colours

Entry Conditions:

- From state 3, the lift's top proximity sensor is activated, the 2 second lift rise timer is done, the Height_accepted_bit is not activated, and the Height_rejected_bit is activated

Actions:

- Turn on the green LED
- Start the 1 second lift drop timer
- Extend the left cylinder after the 1 second lift drop timer is done

Exit Conditions:

- To state 0, the E-Stop was pushed, or a fault has been triggered
- To state 100, the corresponding station's control button has been pressed after logging in for developer mode on the HMI
- To state 1, the lift's bottom proximity sensor is activated, and the lift cylinder's extension proximity sensor is activated

State 100: Developer Mode

Entry Conditions:

- From any state, the corresponding station's control button has been pressed after logging in for developer mode on the HMI

Actions:

- If the gantry arm cylinder extend/retract button is pressed on the HMI, extend or retract the gantry arm cylinder accordingly
- If the gantry arm vacuum on/off button is pressed on the HMI, turn on or off the gantry arm vacuum accordingly
- If the conveyor motor on/off button is pressed on the HMI, turn on or off the conveyor motor accordingly
- If the wheel lift position up/down button is pressed on the HMI, move the lift up or down accordingly
- If the wheel lift cylinder extend/retract button is pressed on the HMI, extend or retract the wheel lift cylinder accordingly
- While the +X button is pressed on the HMI, move the gantry arm in the +X direction
- While the -X button is pressed on the HMI, move the gantry arm in the -X direction
- While the +Y button is pressed on the HMI, move the gantry arm in the +Y direction
- While the -Y button is pressed on the HMI, move the gantry arm in the -Y direction
- If a number is entered in the black wheel count per silo, update the station's black wheel count accordingly
- If a number is entered in the orange wheel count per silo, update the station's orange wheel count accordingly
- If a number is entered in the metal wheel count per silo, update the station's metal wheel count accordingly

Exit Conditions:

- To state 0, the E-Stop was pushed, a fault has been triggered, the 'Back' button or 'Logout' button on the HMI has been pressed

Appendix C: HMI Screens

Refer to section 4.2.1 for a brief description of each of the HMI pages.

HMI Home Page

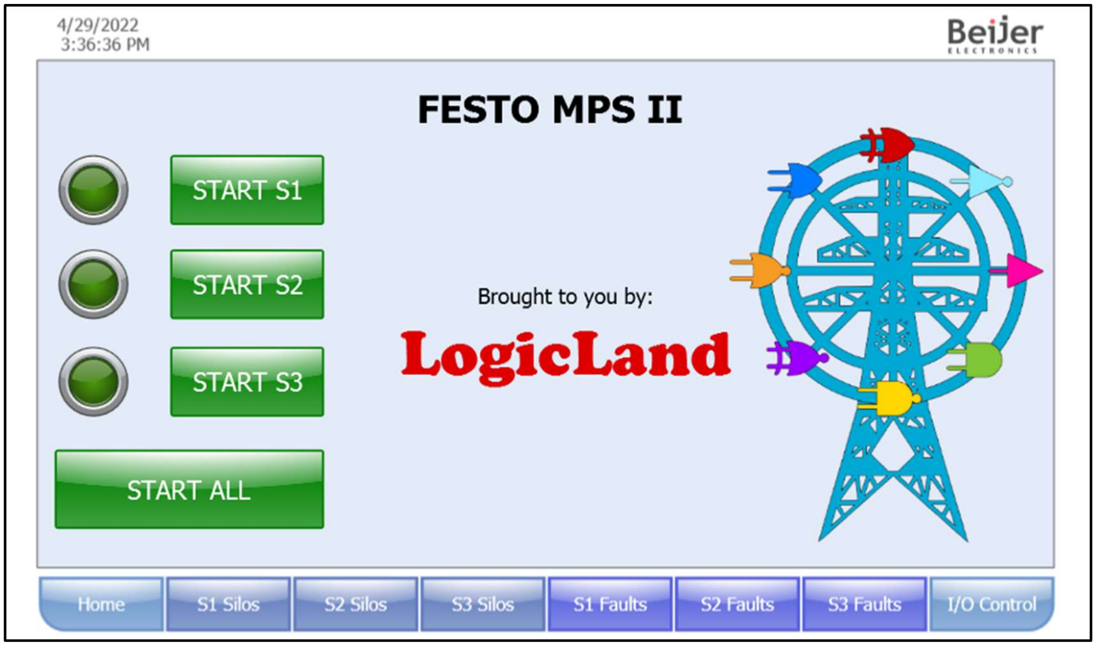

Figure 6: Home Page of the HMI

HMI Silo View Page

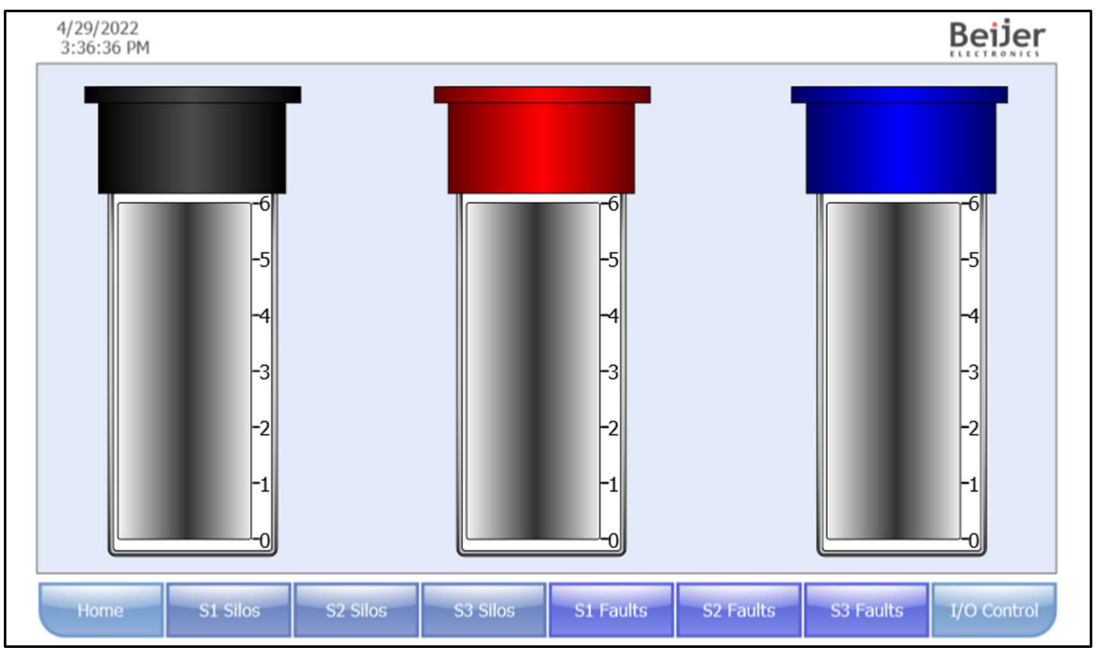

Figure 7: A Station's Silo View Page of the HMI

HMI STATION FAULTS PAGE

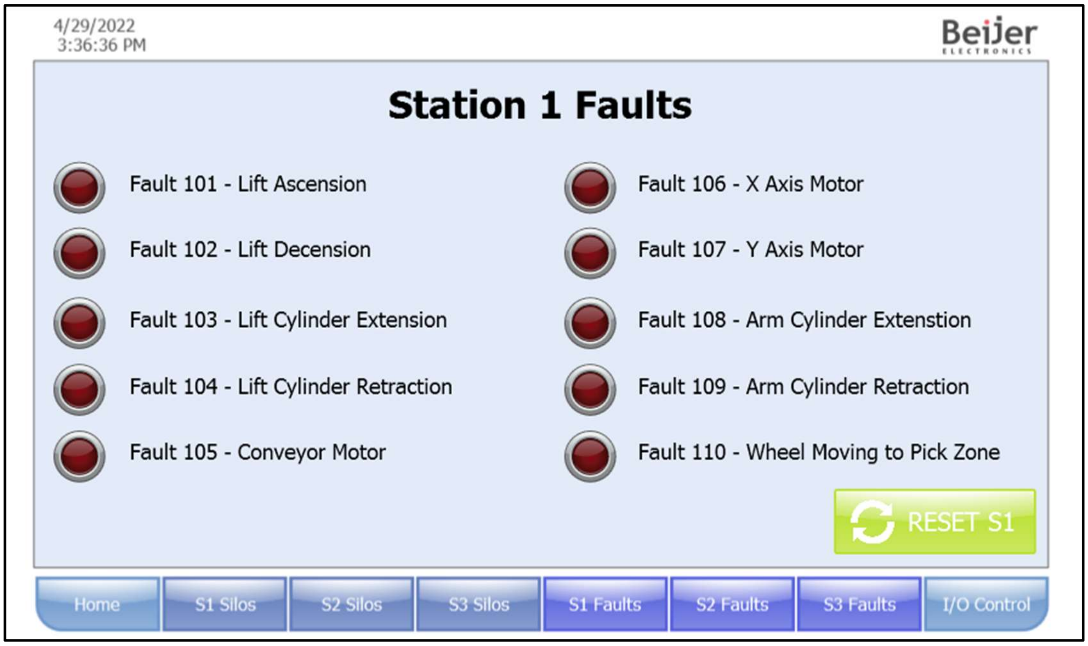

Figure 8: A Station's Fault Page of the HMI

Note: Although Figure 8 shows Station 1's fault page, Station 2 and 3 look similar to Figure 8.

HMI DEVELOPER MODE PAGES

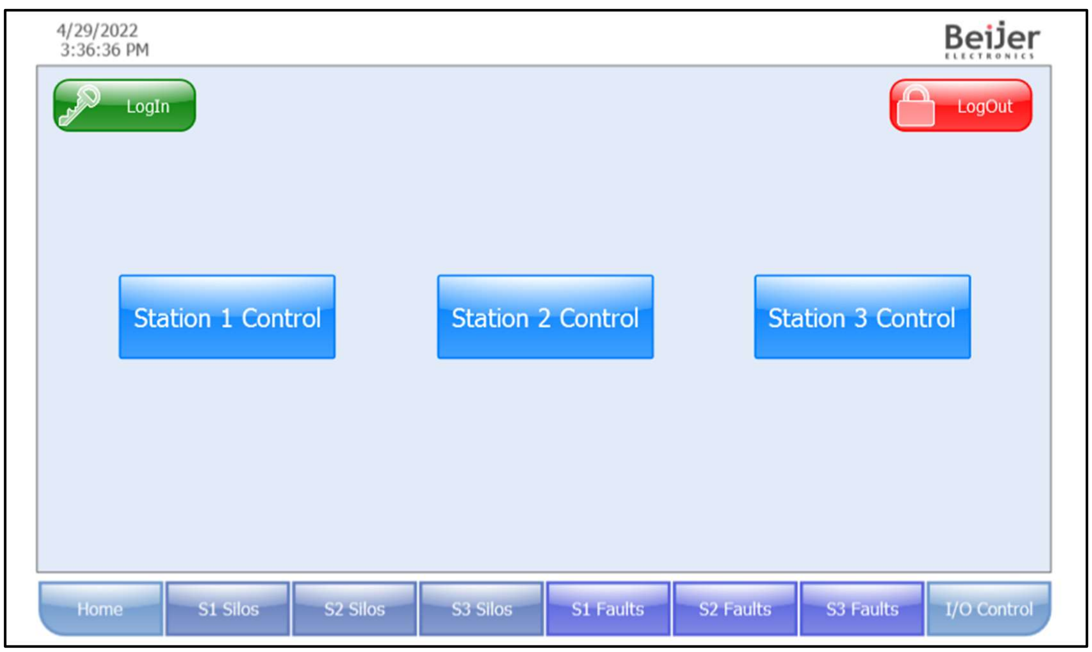

Figure 9: Developer Mode Login in / Station Select Page of the HMI

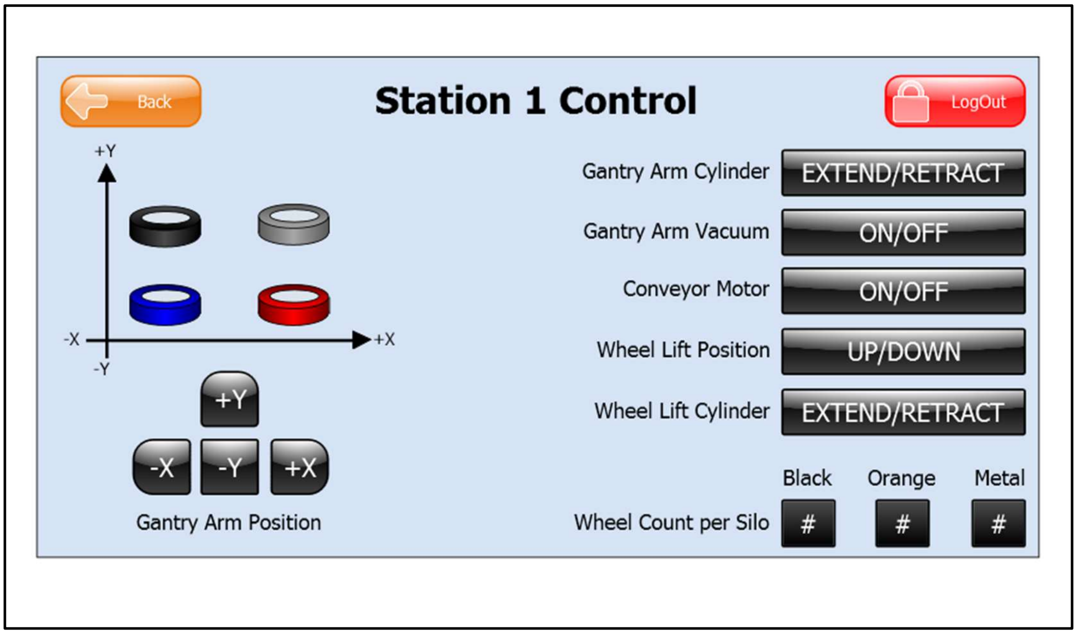

Figure 10: A Station's Developer Mode Controls Page of the HMI

Note: Although Figure 10 shows Station 1's control page, Station 2 and 3 look similar to Figure 10.

Appendix D: Manufactured / Created Parts

Figure 11: Manufactured Silo Slip

Figure 12: Manufactured Rejection Slide for Station 3

Figure 13: Waterjet Cut Rejection Slide Mount for Station 3

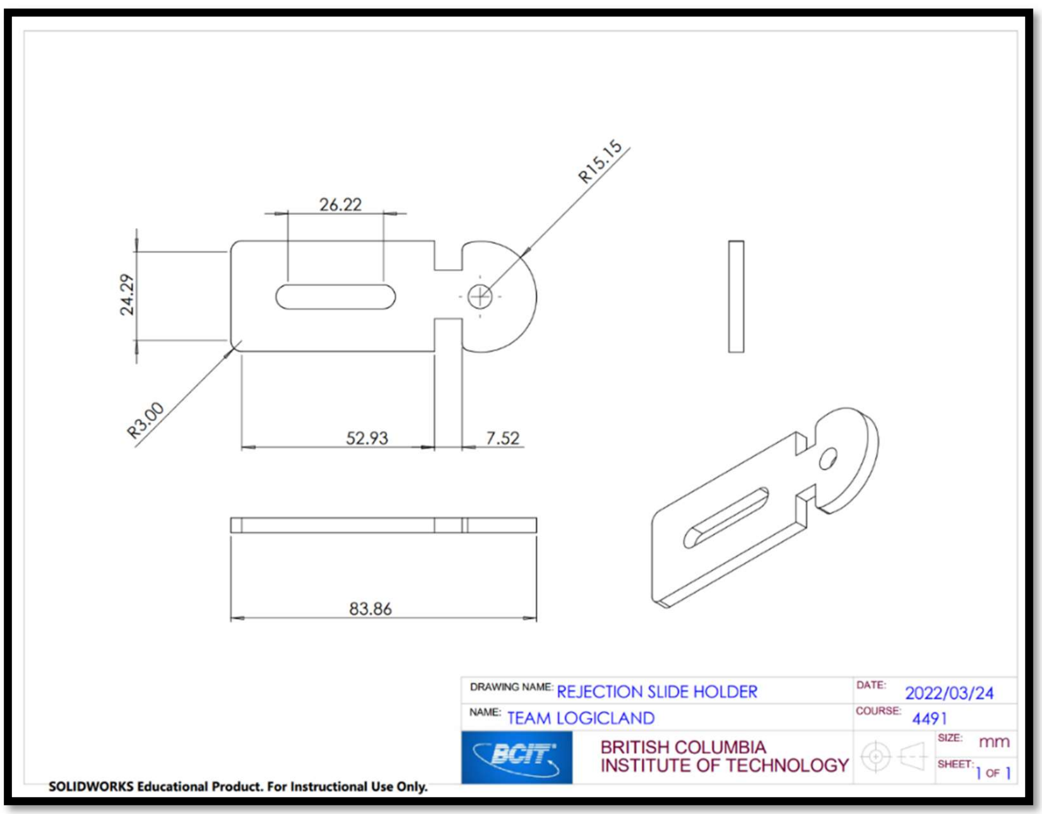

Figure 14: CAD Drawing of Rejection Slide Mount

Figure 15: 3D Printed Silo Cap

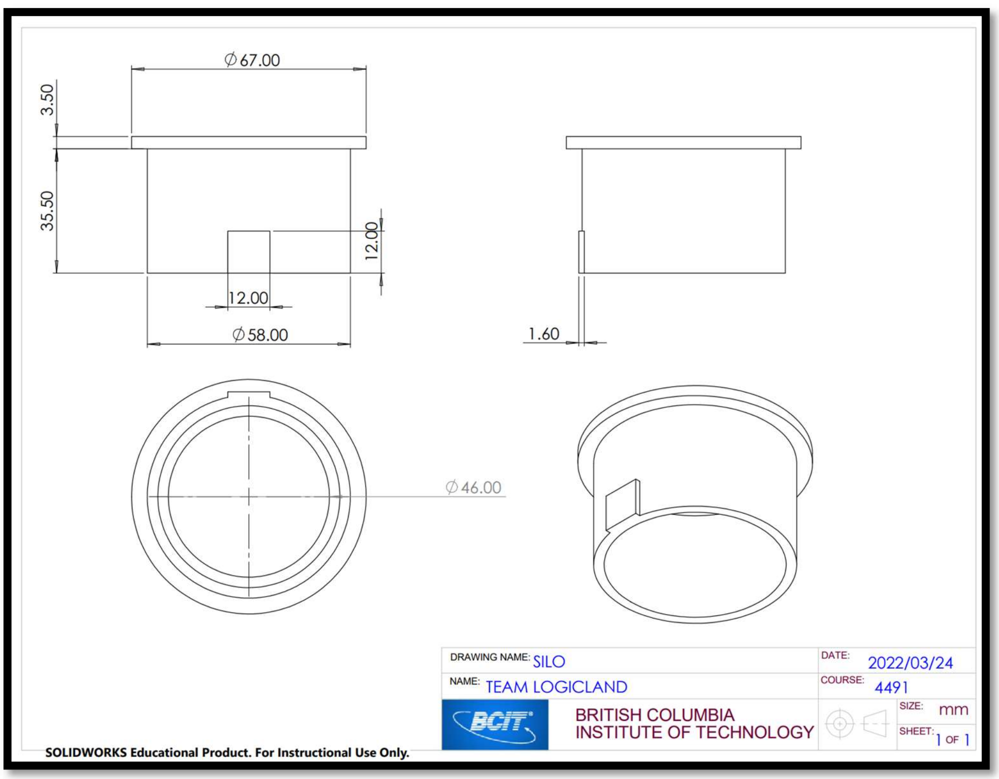

Figure 16: CAD Drawing of the Silo Cap