PLC Controlled Assembly Line Trainer User Manual



Created By: FAMU-FSU COE Senior Design Team 520

Class of 2020

Team 520 Members: Damira Solms, Justin Law, Robert Smith, Ryan Dodson, and Nicholas Salerno

Table of Contents

[**Safety Considerations**](#_fqttglv3jkct) **2**

[**Project Overview**](#_1uiaria50m19) **2**

[**Conveyor Description**](#_x5dfnnm3u83i) **4**

[Conveyor Components](#_moaqx315mguv) 4

[*Flat Belt Conveyor x2*](#_9kvedgjdnbiw) *4*

[*Programmable Logic Controller (PLC) x1*](#_534fgt9z52vb) *4*

[*Inductive Sensor x1*](#_ewhqh88tcxng) *5*

[Photoelectric Sensor x2](#_xwuqnerayl7q) 5

[*Pneumatic Linear Actuator x2*](#_bv8h8qpvuzww) *6*

[Stepper Motor and Driver x1](#_vvctqniufktx) 7

[Rotational Arm x1](#_brhjk4792po1) 7

[Wiring Diagram](#_u1ogsuassfcu) 8

[**Integration**](#_kx641znvb20o) **10**

[**Operation**](#_d6ihhhqx8ffd) **12**

[Connecting Local Computer to PLC:](#_rq11llvflwa4) 12

[Downloading Program to PLC:](#_zcdtdplo8d50) 16

[Uploading Program from PLC:](#_smt9l8770h9k) 17

[Troubleshooting:](#_82e0erv6n83a) 18

[Physical Malfunctions:](#_hwqugd83xj9w) 18

[Software Malfunctions:](#_a3k9ibmeo5vt) 20

[**References**](#_y125f6hm40yv) **22**

# 

# Safety Considerations

* Always disconnect power before troubleshooting or tampering with electrical equipment.
* Always disconnect air supply before troubleshooting or tampering with pneumatics.
* Always keep loose hair,clothing, and other loose objects away from moving conveyors.
* Always stay clear of the direct path of pneumatic actuators.

# Project Overview

The Assembly Line Trainer is an educational tool developed for students at the Tallahassee Community College for the Advanced Manufacturing Training Center. The Assembly Line Trainer uses two conveyors, two pneumatic actuators, two proximity sensors, an inductive sensor, and a diverter arm driven by a stepper motor. The device sorts objects based on material type and size. The students in the manufacturing maintenance course will learn how to troubleshoot hardware and software errors that are integrated into the design. This will familiarize students with industry standard programmable logic controllers and improve their troubleshooting skills for future work in manufacturing plants.

**Software Errors** teach the student how to troubleshoot programming errors. There are a total of five common software errors for the student to troubleshoot. These errors are:

1. Timer/Delays

* Incorrect timing causes the system to energize the actuators at the wrong time thus missing the objects.

1. Sorting Algorithm

* The system sorts inversely. Examples: small objects sort into large object container or metallic objects diverted onto non-metallic conveyor.

1. Stepper Motor Direction

* Arm rotates in the wrong direction and doesn’t divert non-metallic objects.

1. Incorrect Actuator Assignment

* Wrong actuator energized.

1. All Code in a Single Rung

* The system will only operate correctly one time

**Hardware Errors** teach the student how to troubleshoot equipment errors. There are a total of five integrated hardware errors. These errors are:

1. Actuator Placement

* The pneumatic actuators can be moved along the conveyor which causes the actuator to miss the object due to a timing error.

1. Sensor Placement

* Sensor mount can be moved vertically which allows the small 1” cube pass undetected. Sensor mount can also be moved horizontally which causes a timing error and the stepper arm or actuator will not energize at the correct time.

1. Stepper Motor Driver Settings

* Adjustment made to the driver will cause the stepper motor to rotate too much or not enough.

1. Faulty Wiring/Tubing

* Faulty wiring will not provide the power to the electrical components and they cannot activate. Faulty tubing will cause a leak in the pneumatic connections which causes the compressor to continually lose pressure and the actuator will not have enough force to move the cubes.

1. Blown Fuse

* The circuit will be broken, not allowing power to flow to the equipment.

The teachers of the manufacturing maintenance course will be able to integrate any of the 10 errors for the students to solve. Once the students repair the errors, the conveyor will return to working order. Students will then be able to place an object on the conveyor where a proximity sensor will determine the size of the object. Then the inductive sensor mounted on the arm will determine if the object is metallic or non-metallic. If it is metallic, the arm will swing open allowing the object to be sorted into the small metallic or large metallic bin. If the object is non-metallic, the stepper arm will remain closed and the object will be diverted to the next conveyor. Once an object lands on the second conveyor, another proximity sensor allows the PLC to energize the pneumatic actuator at the correct time if needed. Objects recognized as large will fall into a bin located at the end of the conveyor while small objects will be pushed into bins by the pneumatic actuators.

# Conveyor Description

## Conveyor Components

### *Flat Belt Conveyor x2*

The flat belt conveyor is a motor controlled conveyor that is 8 inches wide and 3 feet long. The conveyor transports objects from one place to another and places the object before the sensors which provide the necessary feedback for proper sorting. The conveyors are independently controlled by manual switching. They are powered via 120V AC Power plugs. The conveyor stands and side rails function as excellent mounting areas for the sorting components of the unit.



### *Programmable Logic Controller (PLC) x1*

The PLC is an Allen Bradley Micrologix 1100 model. The software used to program it is RSLogix Micro Starter Lite. The PLC has 10 digital inputs and 6 digital outputs. The inputs allow for sensor communication and the outputs result in the actuator movements needed to direct the test objects into the correct bin.



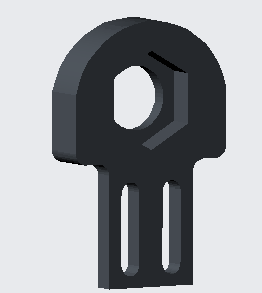
### *Inductive Sensor x1*

The inductive sensor is used as the material type sensor. The inductive sensor functions through induction. When power is supplied to the sensor a magnetic field is created. The sensor provides feedback when the magnetic field is interrupted by a ferrous material. It then triggers a threshold circuit which indicates that a metallic object is present. The sensor will be embedded into the rotational arm.



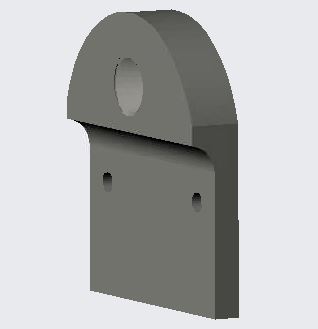
### *Photoelectric Sensor x2*

The photoelectric sensors function as proximity sensors which help with timing and determine whether an object is large or small. The sensor emits a light onto a reflector which creates a continuous loop. When an object is placed between the sensor and reflector then the loop is cut off. The length of time that the feedback loop is cut off for, corresponds to the size of the object. The proximity sensor and mount are shown below:



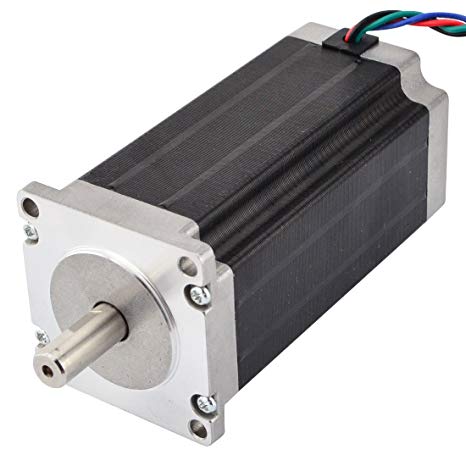
### *Pneumatic Linear Actuator x2*

The pneumatic linear actuator, is an arm that pushes the objects off of the conveyor belt using compressed air. A solenoid valve releases the air after the PLC signals for the actuator to activate. The actuator and mount are shown below:

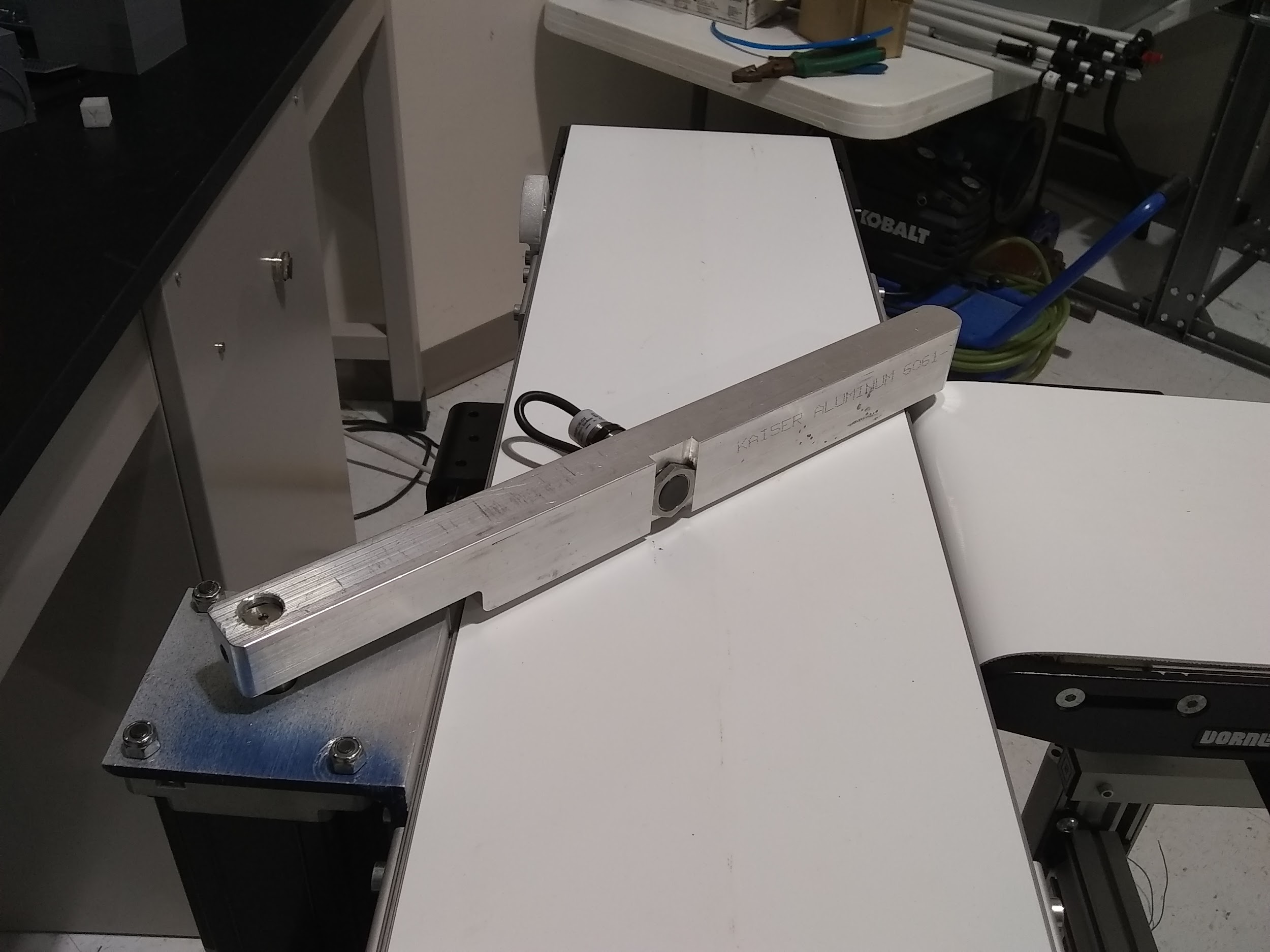
### *Stepper Motor and Driver x1*

The stepper motor and controller will rotate the rotational arm. The motor is a NEMA-34 motor, and the driver is the DM542T. The PLC will communicate with the driver to move and position the motor and the rotational arm to its desired position.

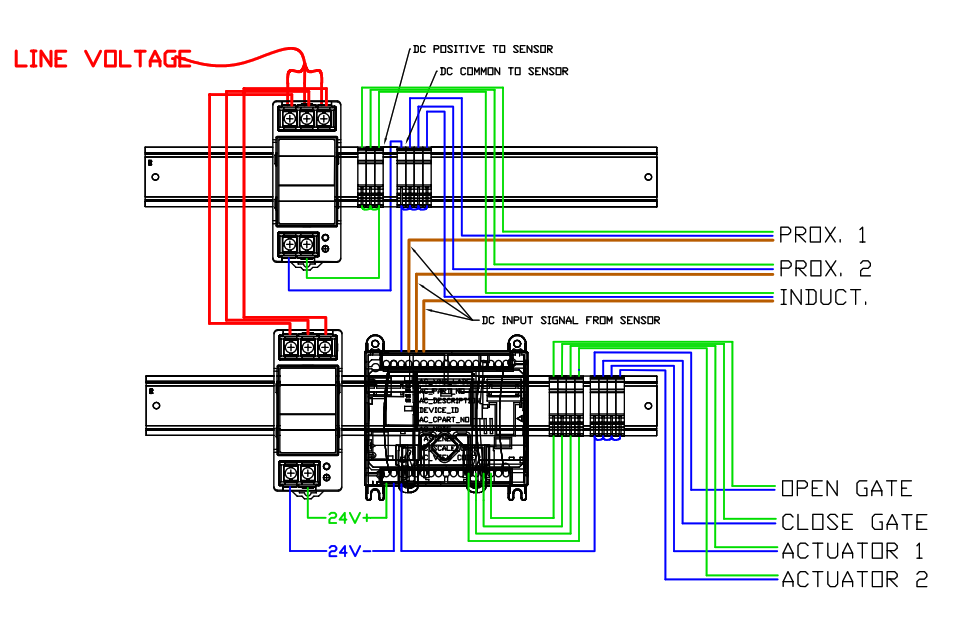
 

### *Rotational Arm x1*

The rotational arm consists of an aluminum rod that directs the object onto the correct conveyor based on material type. The aluminum rod is attached to a stepper motor shaft and has the inductive sensor embedded.



## Wiring Diagram



The wiring diagram above shows how the PLC controlled components of the assembly line trainer will be wired to communicate and receive power from the PLC. The power supplies receive 120V AC and convert the power to 24V DC to energize the PLC and other devices. The stepper motor requires a large current load so to combat this we use two power supplies, one for the sensors and one for the PLC and actuators.

**PLC Logic Pseudo-Code**

\*Start Conveyor, Connect PLC to 24V DC Power Supply

**If** object = metal

Rotate arm off of conveyor

**If** proximity sensor 1 result = small object

Use timer to push off with pneumatic actuator 1

**Else if** proximity sensor 1 result = large object

Do nothing

**end**

**else if** object = nonmetal

Keep arm at default position on belt

**If** proximity sensor 2 result = small object

Use timer to push off with pneumatic actuator 2

**Else if** proximity sensor 2 result = large object

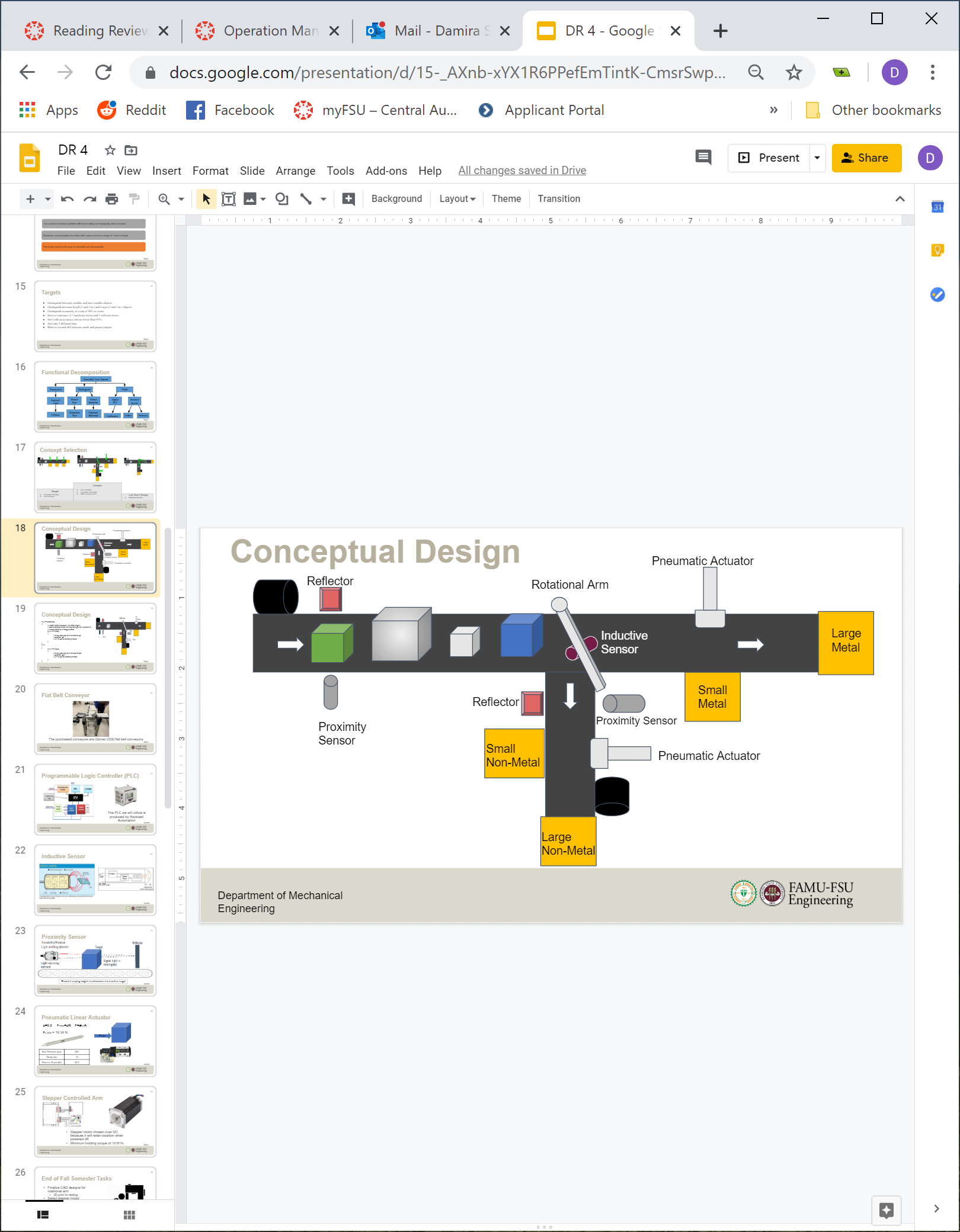
Do nothing

**end**

**End**

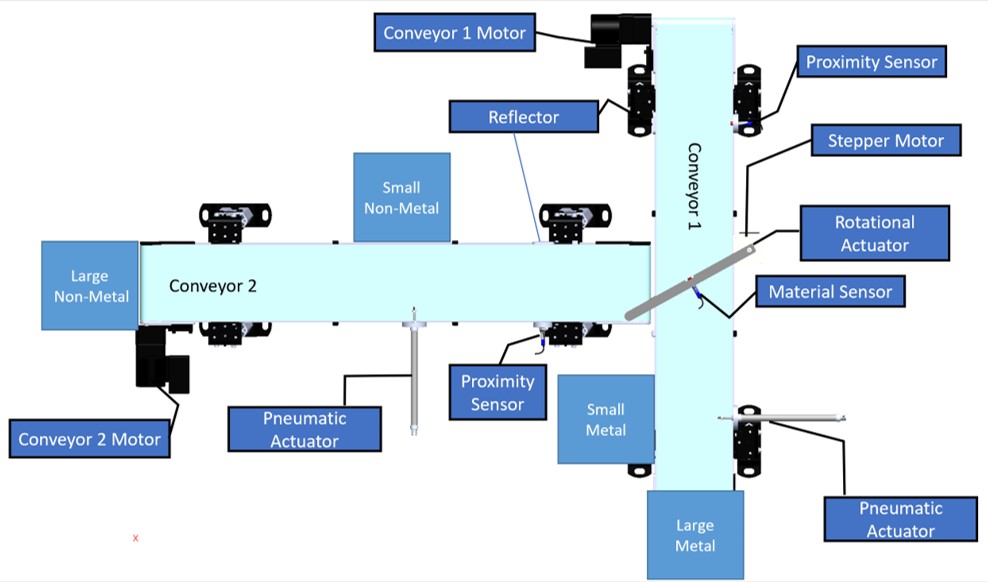
# Integration

The schematic below shows how the conveyor system was designed to function. This provides a top view of the design. The blocks in the schematic are representative of the blocks that the mechanism sorts by size and material type.



Each of the components is highlighted so that their location on the conveyor belts can be seen. The proximity sensor provides information about size (small or large) and location to the PLC. The rotational arm uses the inductive or material sensor to determine material type of the test object. If the object is metal the arm rotates to allow it to pass through and continue along the conveyor. If the material is non-metallic the rotational arm remains in position and diverts the object onto the second conveyor belt. Once the object is past the rotational arm, the pneumatic actuator linearly pushes the small objects into a receptacle and allows the large objects to pass.

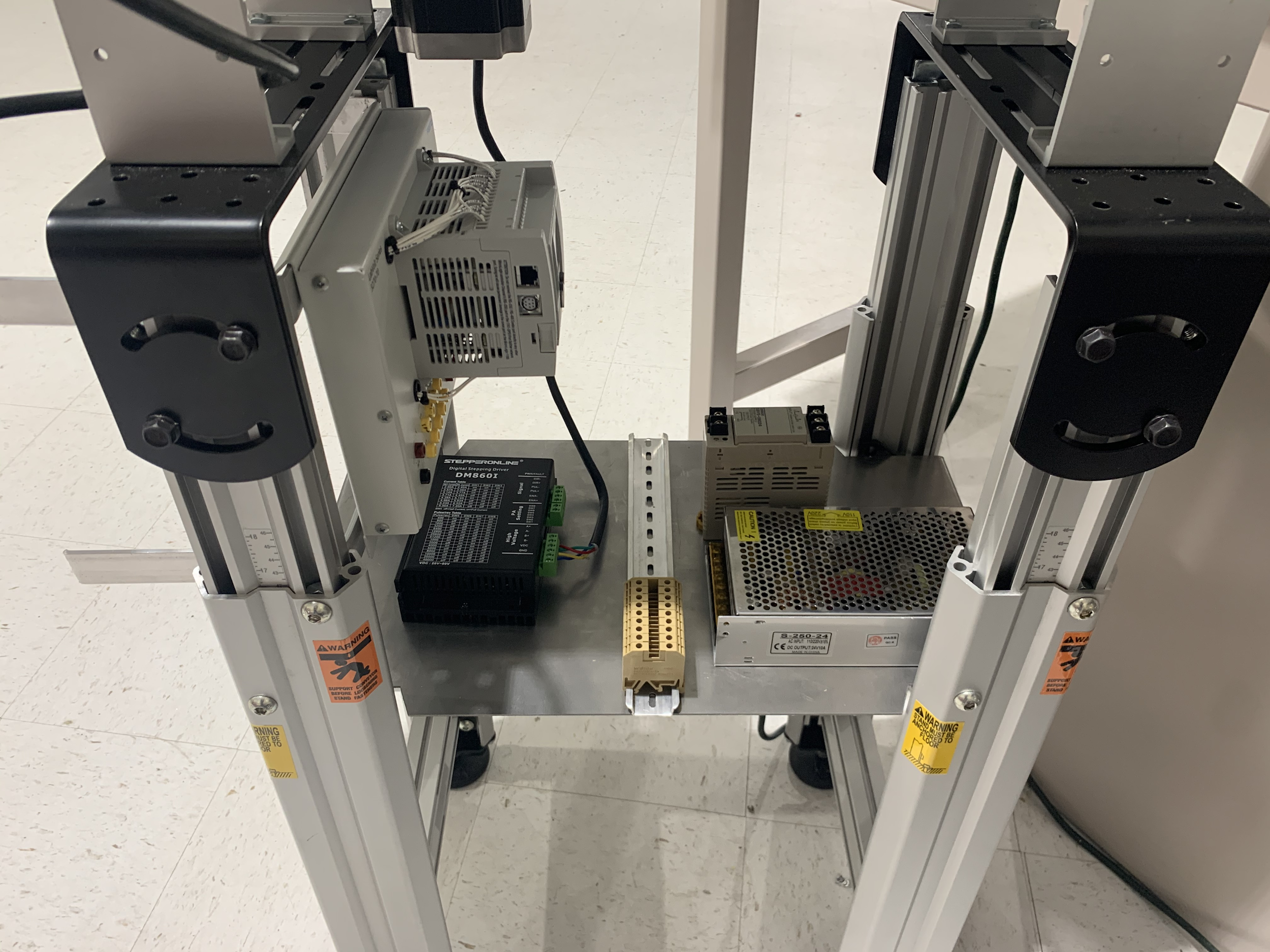
The CAD model below was generated in Solidworks once the unit was constructed to represent the true final design. This representation shows the true size of the components relative to each other.



The CAD model below provides a trimetric view of the constructed conveyor unit. The 8020 Al stands that the conveyors are placed on function as fastening mechanisms for all of the components of the assembly line trainer unit. The blue and gray cubes on the conveyor show where the test objects are to be placed on the conveyor belts.



To maximize safety the electrical components and PLC controller were all placed under the conveyor belts within the confines of the 8020 Al stands. This decreases the likelihood of someone bumping into the electrical components on the unit when walking by. An image of the physical set up is shown below.



The setup above shows how an aluminum sheet fastened to the 8020Al creates a support area for the electrical components of the unit. The PLC is mounted onto the sides and din rail is used to support the fuses. The two 24V power supplies are fastened onto the aluminum sheet for further safety and security.

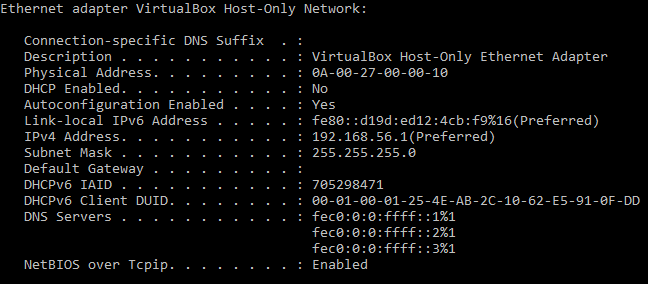
# Operation

**Download Rockwell Software. Link to download is in references.**

## Connecting Local Computer to PLC:

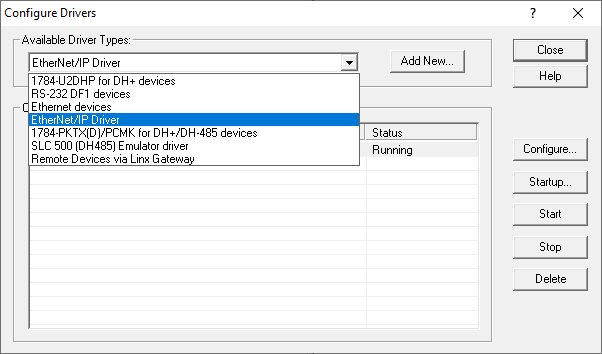
**If running software for first time**

1. Get Ethernet Port data
   1. Start -> Windows System -> Command Prompt
   2. Enter “ipconfig /all” into Command Prompt
   3. Look for “Ethernet Adapter”

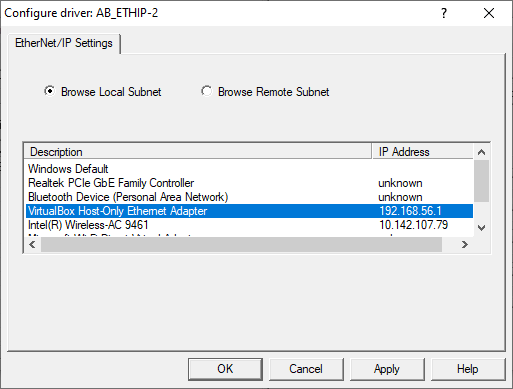


* 1. Record IPv4 Address and Subnet Mask.

1. Set up RSLinx
   1. Start -> All Programs -> Rockwell Software -> RSLinx CLassic
   2. Communications -> Configure Drivers and select EtherNet/IP Drivers

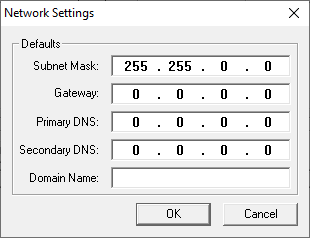


* 1. Once selected EtherNet/IP Driver, click “Add New…” then OK in the pop-up window
  2. When the next window pops up, select the option that has the same IPv4 Address you recorded in step 1.

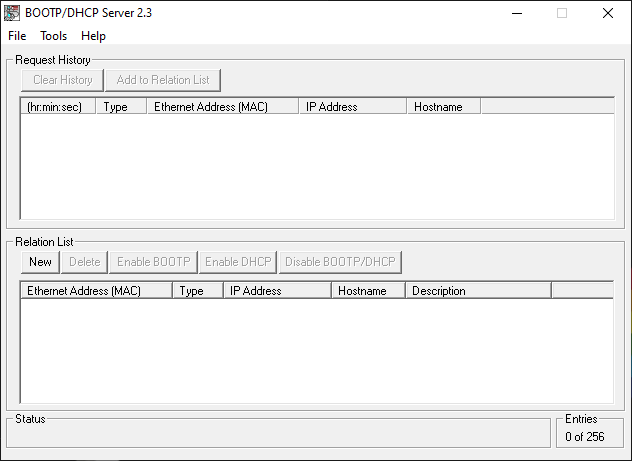


* 1. Click OK

1. Turn the Allen-Bradley MicroLogix 1100 on by connecting a 24V DC power supply, and connect an ethernet cable from the PLC to the computer.
2. Set up BootP/DHCP utility
   1. Start -> All Programs -> Rockwell SOftware -> BootP-DHCP Server
   2. Enter Subnet Mask and click OK



* 1. Wait for PLC to show up in the “Request History” section



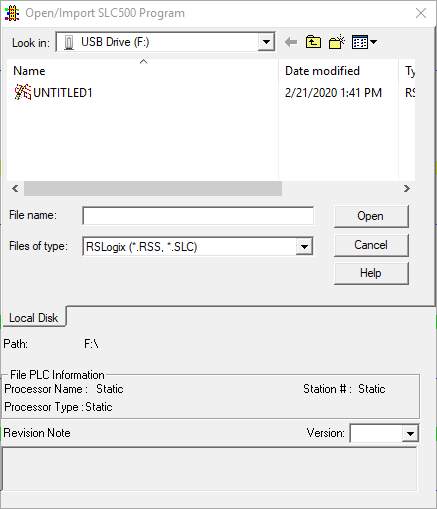
* 1. Select the PLC and click “Add to Relation List”
  2. Enter an IP address in the pop-up window and click OK
     1. IMPORTANT! First three sets of numbers should be the same as the IP address recorded earlier. Fourth number should be different.
  3. Wait until the “Request History” list updates and shows PLC with configured IP address
  4. Select the PLC in the “Relation List” section and press the “Enable DHCP” button
  5. Select File -> Save As… -> FileName
     1. FileName is users choosing
     2. This saves Configuration for future use.

**If already configured**

1. Open BootP-DHCP Server
   1. Start -> All Programs -> Rockwell Software -> BootP-DHCP Server
2. Open configuration file
   1. File -> Open -> FileName
      1. FileName is the name chosen during initial setup
3. Connect PLC to computer
   1. Should see PLC show up in “Request History” with configured IP Address

## Downloading Program to PLC:

1. Open RSLogix Micro Starter
   1. Start -> Rockwell Software -> RSLogix Micro
2. Open Project
   1. File -> Open…
      1. Pop-up window should appear



* + 1. Select desired program and click “Open”

1. Compile Program
   1. In Standard menu bar, select press 
      1. Will be in the bar shown below:

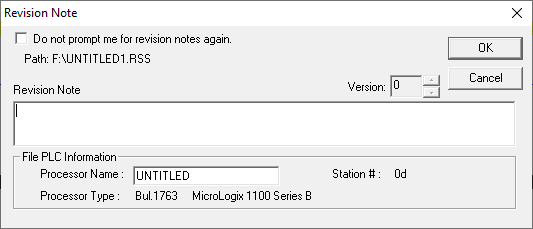


* 1. Check messages at the bottom of the window to ensure the program compiled successfully.
     1. If errors occur, debugging the code will be required to upload to PLC.

1. Upload
   1. Make sure the PLC is connected to the local device, connected to a 24V DC power supply, and communication is configured.
   2. Click drop-down arrow shown below in red



* 1. Select “Download...”
  2. A pop-up window should appear
     1. Insert any revision notes if desired
     2. Select “OK”



* 1. Once the program has been uploaded, a pop-up window may appear asking to go online, select “Yes” to see the logic in real-time.

## Uploading Program from PLC:

1. Connect PLC to local device
   1. Configure PLC Network connection to allow communication between the PLC and the local device.
2. Open RSLogix Micro Starter
   1. Start -> Rockwell Software -> RSLogix Micro
3. Download Program
   1. Click drop-down arrow shown below in red:



* 1. Select “Upload…”
     1. The program installed on PLC will be downloaded to the local device to make changes or watch in real-time.

## Troubleshooting:

### Physical Malfunctions:

**Arm Actuation**

1. Make sure the arm is in the correct starting position when powering on the system. The arm must be in the closed position; pointing towards the secondary conveyor.
2. If the arm is rotating too much or not enough, check that the dip switches located on the side of the stepper driver are set properly.



1. The setting should reflect the maximum current and a pulse/rev of 1000. The switches should be set as follows:  
   SW8: off  
   SW7: on  
   SW6: on  
   SW5: on  
   SW4: on  
   SW3: off  
   SW2: off  
   SW1: off

**Pneumatic Actuation**

1. Check to make sure the air tubing is connected properly. Turn on the air compressor and check for leaks by listening for hissing around connection points and throughout the tubing.
2. If a pneumatic actuator is moving too slow or too fast, the air supply pressure can be adjusted to change the speed and force.



**Proximity Sensors**

1. If the system does not seem to detect objects, ensure your proximity sensors are wired properly.
2. If the system detects only larger objects, ensure that the sensor is adjusted at a height closer to the conveyor.

**Inductive Sensor**

1. If the system is not detecting metal objects, ensure your inductive sensor (embedded in the rotational arm) is wired properly.
2. If objects get caught on the inductive sensor while the conveyor is moving, ensure the sensor is flush with the surface of the rotational arm. If the sensor is embedded too deep in the arm, this may cause the sensor to not detect the objects.

### Software Malfunctions:

**Timing**

1. If the pneumatic actuators are missing the objects:
   1. Ensure the proximity sensors are functioning properly.
   2. Check that the correct actuator is moving. If not, reverse the actuator assignments in the code.
   3. If the pneumatic actuators are moving too soon/late, adjust the delays in the code accordingly.

**Sorting Algorithm**

1. If system does not distinguish object by material:
   1. Ensure the rotational arm and inductive sensor are wired properly.
   2. Check the code to see if the arm is rotating counterclockwise. If the arm is not rotating in the correct direction, it will not sort properly.
2. If the system is not sorting by size correctly:
   1. Check for timing malfunctions.
   2. If objects are being sorted into the wrong bins, ensure the code reflects the corresponding size with the correct actuation.

**Logic Rungs**

1. Ensure that parts of the code are not in separate rungs. The system will only run properly if the code is contained in a single rung.

# 

# 

# 

# References

Andrew Parr MSc, CEng, MIEE, MInstMC, in [Hydraulics and Pneumatics (Third Edition)](https://www.sciencedirect.com/book/9780080966748), 2011

Rotary Actuator:

link:<https://www.sciencedirect.com/topics/engineering/rotary-actuator>

PLC Programming Tutorial:

link:<https://www.youtube.com/watch?v=ArFZD7wEFN4>

RSLogix Micro Starter Download:

http://compatibility.rockwellautomation.com/pages/search.aspx?crumb=117&q=RSLogix%20Micro%20Starter%20Lite%20w/o%20RSLinx%20EN