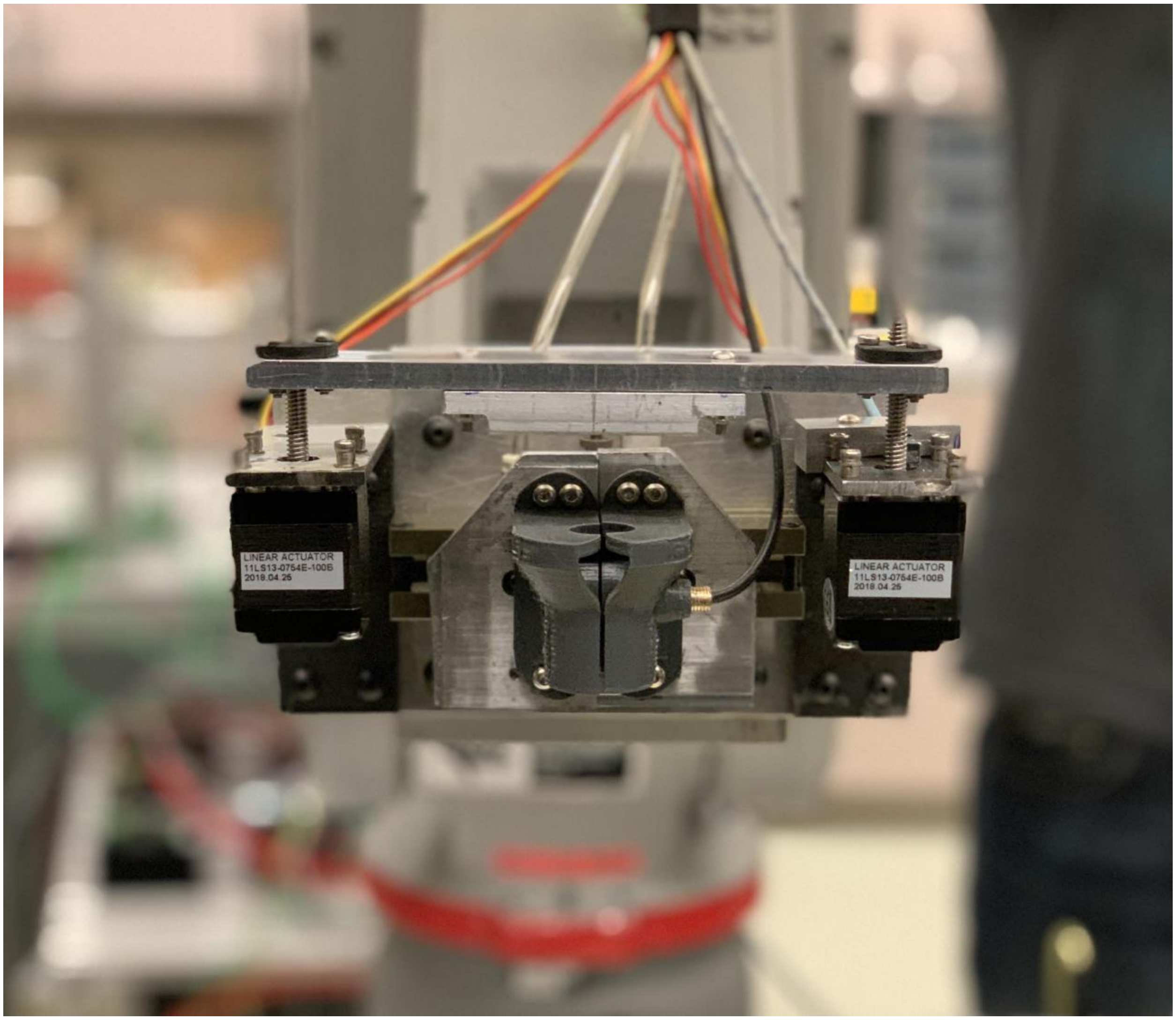
Dual Motor Syringe Dispenser & Changer Manual

ENGR 480




Giovani Ribeiro | Tatum Shobe | Geoffrey Urbin | Donnieval Walker

# Contents

## Overview

Parts:

- **MOTO**MAN **MH5L**
- Custom **MOTO**MAN Mounting Plate
- Custom Assembly Mounting Plate
- Gripper Assembly:
  - Parallel Pneumatic Gripper
  - Mounting Plate for Syringe Holster (x2)
  - Syringe Holster (L)
  - Syringe Holster with Mounting Hole for Optical Sensor ®
- Motor Assembly:
  - NEMA 11 Stepper Motor with Plastic Nut (x2)
  - Shaft Coupler for Depression of Plunger with Spacer
  - Motor Mounting Bracket (x2)
- Syringe Holder Assembly:
  - 3-D Printed Base
  - Syringe Rack Rails
  - Aluminum Coupling Block
  - Extruded Aluminum Supports (x2)
  - Mounting Brackets

Electronics:

- PLC
- Stepper Motor Driver
- 24V Power Supply
- 48V Power Supply
- Optical Sensor
- Inductance Sensor

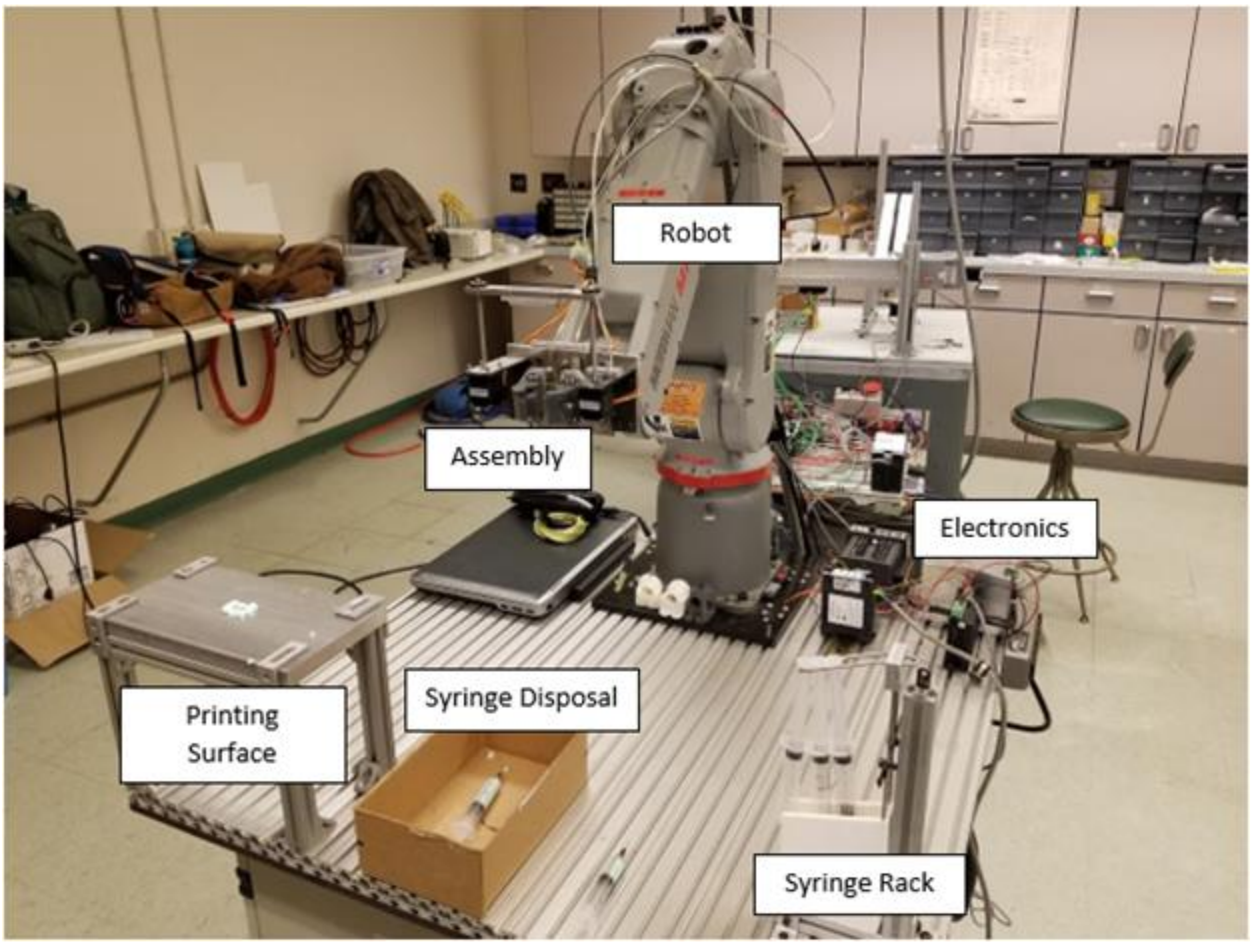
Software:

- PLC Software
- **MOTO**MAN Software (Provided)

Compressed Air

## Description of Function(s)

The Primary function of the syringe dispenser is to dispense [or print] the contents of a syringe onto a designated surface in a predetermined pattern. Secondarily, the syringe dispenser disposes of the empty syringe, and procures a new syringe from the syringe rack. The printing process continues until the desired pattern has been achieved. All operations are editable, including but not limited to: printing pattern, speed of stages, and disposal zone. Non-standard syringes will require a different syringe holster.



## Motoman Steps

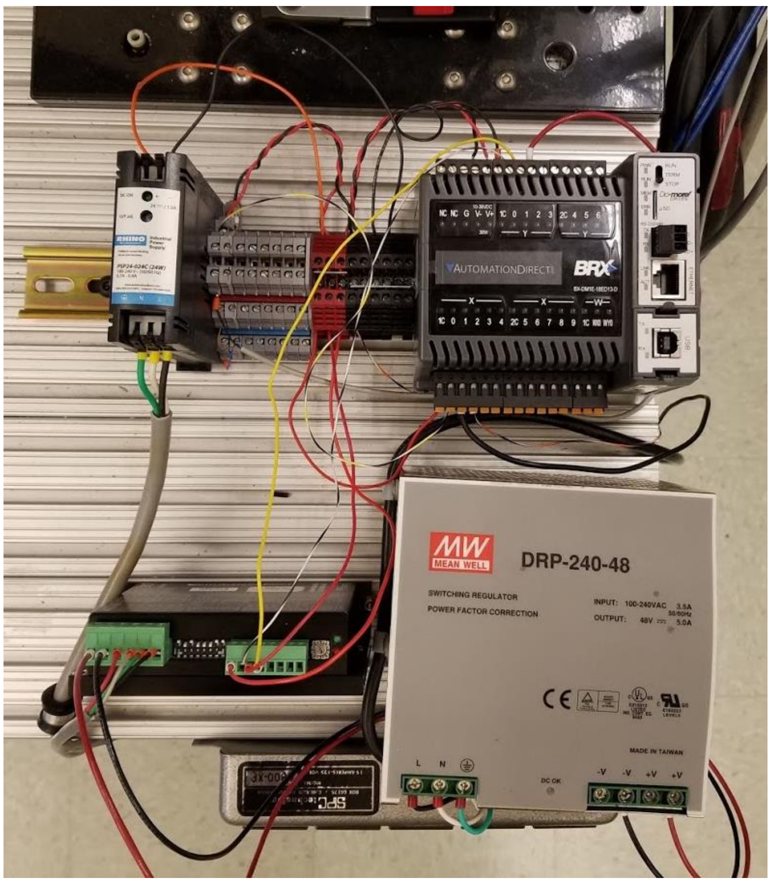
1. Motoman goes to the starting position.
2. Motoman then moves to the front of the syringe dispenser
3. The robot arm slows down until it reaches the point where it can grab the syringe. The robot will remain in this position until the optical sensor on the robot detects a syringe.
4. Once the optical sensor is tripped, the syringe holder on the arm will close and the arm will move onto the next step.
5. The robot arm moves to the extrusion table and pauses at a point to ensure the pressure plate is making contact with the top of the plunger. The stepper motors turn on, bringing the pressure plate down.
6. The stepper motors stop, and the robot moves to the first point of the extrusion pattern.
7. Stepper motors start moving to start the extrusion as the robot arm simultaneously follows the extrusion pattern. The stepper motors stop once the pressure plate makes contact with an inductive proximity sensor (occurs when syringe is empty).
8. The stepper motors then reverse direction to a set distance to reset the pressure plate to the starting position while the motoman moves to the dispenser bin.
9. The syringe clamps release the empty syringe into the bin below, and the process restarts.

## Syringe Rack

The purpose of the syringe rack is to hold full syringes for the robot to grab before it starts extruding. The assembly required a 3-D printed part along with an aluminum block to hold the wire prongs and rails. Extruded aluminum and brackets were used to fasten the printed guides to the table surface and stabilize the rails. When the robot pulls the first syringe off, gravity pulls down on the remaining syringes causing them to slide on the rails to be picked up next.

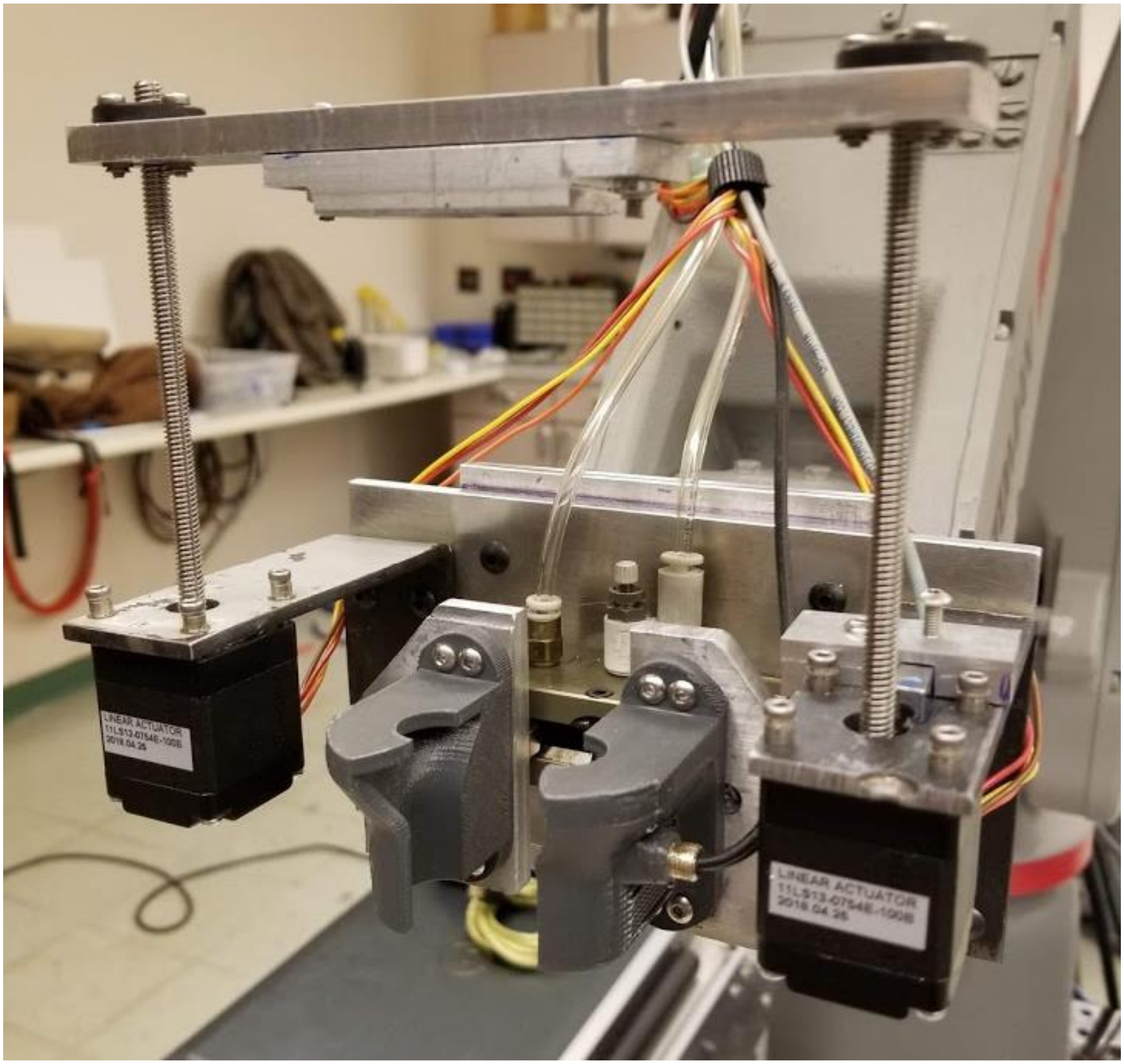
## Electronics and Robot

The two main electronic systems for the project were the PLC and the robot. The electronic sensors and mechanisms were split between the two systems. The pneumatics and optical sensor were run through the robot, while the motors and inductance sensor were run through the PLC. To program the PLC, a state machine was created and programmed on one of the laptops. On the robot side, programming was done using the robot's control interface. Because these two systems were separate, more wiring was needed to interface the PLC and robot to get them talking to each other. The PLC ladder logic, state machine design, robot code, and the electronic schematic are shown in the appendix.




## Assembly

Here in the assembly is where the motors, pneumatics, sensors, grippers, and extrusion bar are placed and set up. A majority of this assembly was machined using the mill, bandsaw, and drill press. Aluminum was used all throughout, except for the steel L brackets and the 3-D printed grippers. All the components were put together through screws and bolts. The pneumatic tubes, motor wires, and sensor wires were organized and guided to their respective systems.




## Operation Instructions

Before the dispensing and changing assembly can be used, the workspace must be set up to reduce the risk of harm to users, equipment, and damage to the surroundings. Following the determination of the tool path, the workspace should be cleared accordingly.

Step 1: Turn on the Motoman Robot and plug in the PLC

Step 2: Turn on the PLC while waiting for the robot to boot up

Step 3: Select AAAFINAL2TATUM program under the job section

Step 4: Ensure that the robot is set at the first location point in the program

Step 5: Ensure all universal outputs are set to off

Step 6: Set the robot into play mode and turn on the server

Step 7: Press the green start button when ready

## Maintenance Instructions

Maintenance instructions for the robot and PLC should be followed (see Motoman and BRX user manuals). Keeping food and liquids (other than the solution being extruded) away from the PLC, robot, and other electronics is vital. Keep area dry and clean as well as clutter free. Before use, ensure that there is nothing the robot will collide with or damage in anyway. To ensure sanitary conditions for biomedical applications, parts should be cleaned and sanitized before each use (some updates to designs need to be implemented to truly be a hygienic design). Be careful to avoid any liquids from getting on or near the electronics.

## Future Work

The current design does work, but there are a few design changes that could be made to optimize form and function.

**Gripper Assembly**

The mounting plate for the syringe holster can be machined to remove excess material. This would create more clearance so the motor brackets can be moved inward toward the center without limiting the gripper jaws' motion. This reduction in material would reduce the footprint as well as the weight of the overall assembly.

**Syringe Holster**

The current syringe holster is the fourth iteration of the design. The CAD was designed to tight tolerances, but the 3-D printer was not able to maintain the tolerances through the print. There was some warping, improperly printed surface finish, and there were inaccurately printed holes. The improperly printed holes caused skewing when attaching the holster to the mounting plate. The warping caused some misalignment of the inside contact surface for the syringe. If the robot attempts to grab a new syringe while the tabs on the body of the second syringe in the rack are perfectly parallel with or perpendicular to the mounting plate, the holster may snag that second syringe. While is naturally unlikely to occur, it is repeatable. Our analysis has confirmed that a simple fillet of the offending contact surface and reprint of the holster will solve the problem.  The warping of the holster hemispheres reduced some of the hole sizes, and as a result, more finishing work was required for the part. The solution to these problems would be to use a 5-axis mill [due to complex geometry] or cast a better part out of metal.

**Assembly Mounting Plate**

The mounting plate on its own was suitable. However, using the CNC mill would have allowed us to make the mounting plate and the brackets for the motor mounts out of one piece, which would mean fewer parts and more even geometry. The mount for the inductance sensor could then have been made from the same piece instead of attaching a separate holder. This would also reduce the weight even further and reduce the complexity of assembly by reducing the number of fasteners and supporting hardware needed.

**Shaft Coupler for Plunger Depression**

The shaft coupler is the long coupler that spans the distance between the two NEMA 11 shafts. The dual motor design was chosen to reduce the vertical profile of the assembly, but without sacrificing performance to do an asymmetrical design. This kept everything compact for more versatile joint articulation. The speeds of the motors are matched to keep the coupler sides moving in sync. This operation is successful when the viscosity of the dispensed material is relatively low, but higher viscosity fluids are more of a challenge. The coupler and the attached spacer are not perfectly centered with the syringe plunger head. This is due to manufacturing defects in the coupler as well as warping of the syringe holster as mentioned before. During operation, the viscosity causes the side with more load to lag the side with less load. Eventually, askew coupler would prevent rotation of the motor shafts. CNC milling this part would reduce the number of parts by combining the geometry of the coupler and the spacer and allow for much tighter tolerances. Making the part perfectly symmetrical will solve the problem.

**Stepper Motor**

The stepper motor driver seems to have a built-in function which causes it to reverse its direction slightly when it can no longer move. This means that we cannot use the persistent signal from the inductance sensor's activation as an input. These required a rewrite of the PLC logic. The shafts on the motor are slightly bent, this may have an effect on how evenly the force of the motors transmits to the syringe plunger.

**Hygiene**

Due to time constraints and limited facilities, a more hygienic design was not feasible. Mentioned before are a few improvements that could contribute to a more hygienic design such as machining several components from a single piece of stock to reduce crevices as well as fasteners. Stainless steel would be our choice for the mounting plate/bracket assembly, and for the holster. The pneumatic gripper could also be encased in a stainless-steel cover, and the exposed parts could have a shield below to prevent any fallout of debris or microscopic particles from getting in contact with the print surface. The entire robot arm could also then be covered with a custom surgical drape.

## Conclusions and Results

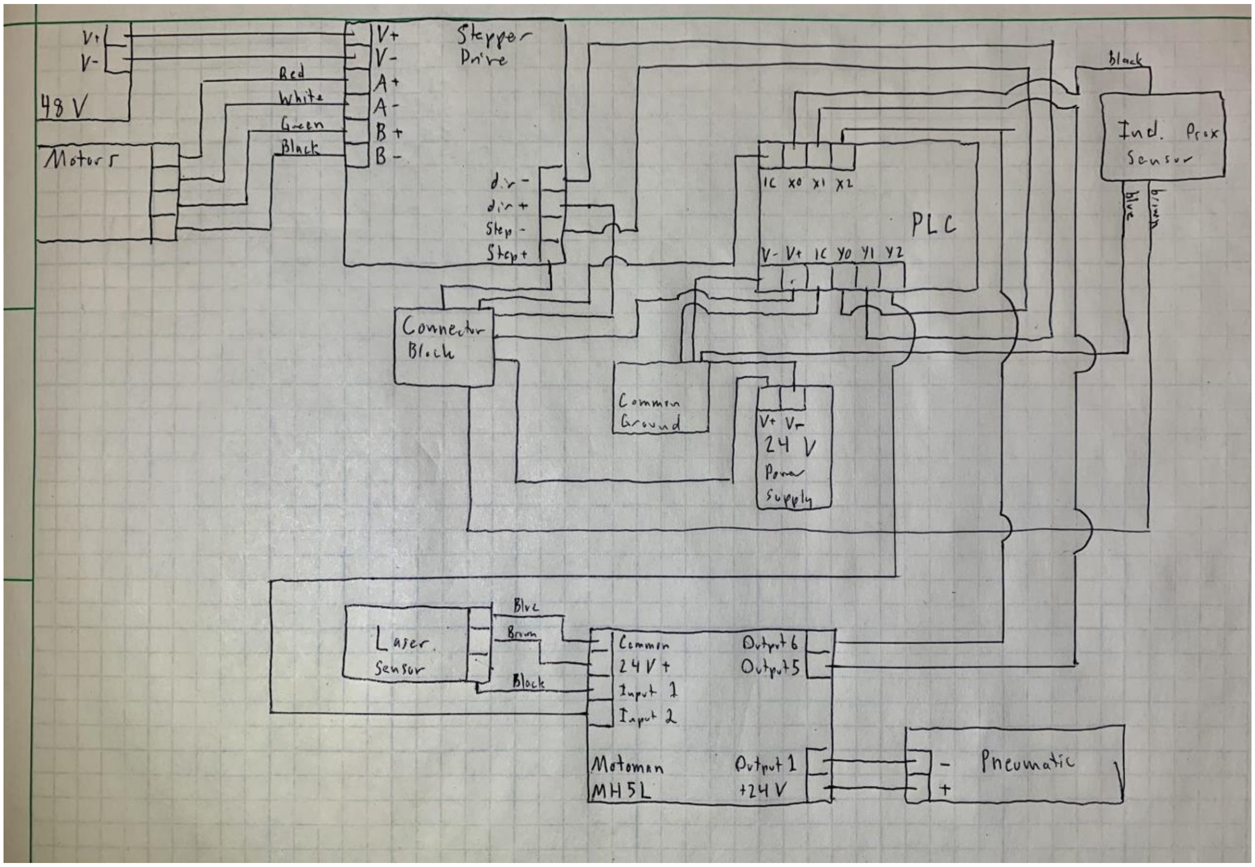
Ultimately, the syringes were successfully changed using the robot. However, there were some problems that occurred which would need to be fixed by implementing the above future work. First, the syringe would sometimes jam in the syringe holster depending on the orientation of the syringes. This could be fixed by reprinting a new iteration of the holster (as discussed above). Second, the stepper motors became jammed when the fluid in the syringe was too viscous. This was primarily because the motors were not perfectly aligned and therefore the shaft coupler would become uneven and cause motors to jam or slow down considerably. This could be fixed by more precise manufacturing of the component parts for better alignment and by slowing the extrusion rate of the fluid. In the end the robot successfully grabbed, dispensed, and changed syringes as long as they did not have fluid in them. The future improvements need to be implemented to have a fully operational syringe changer.

# Appendices

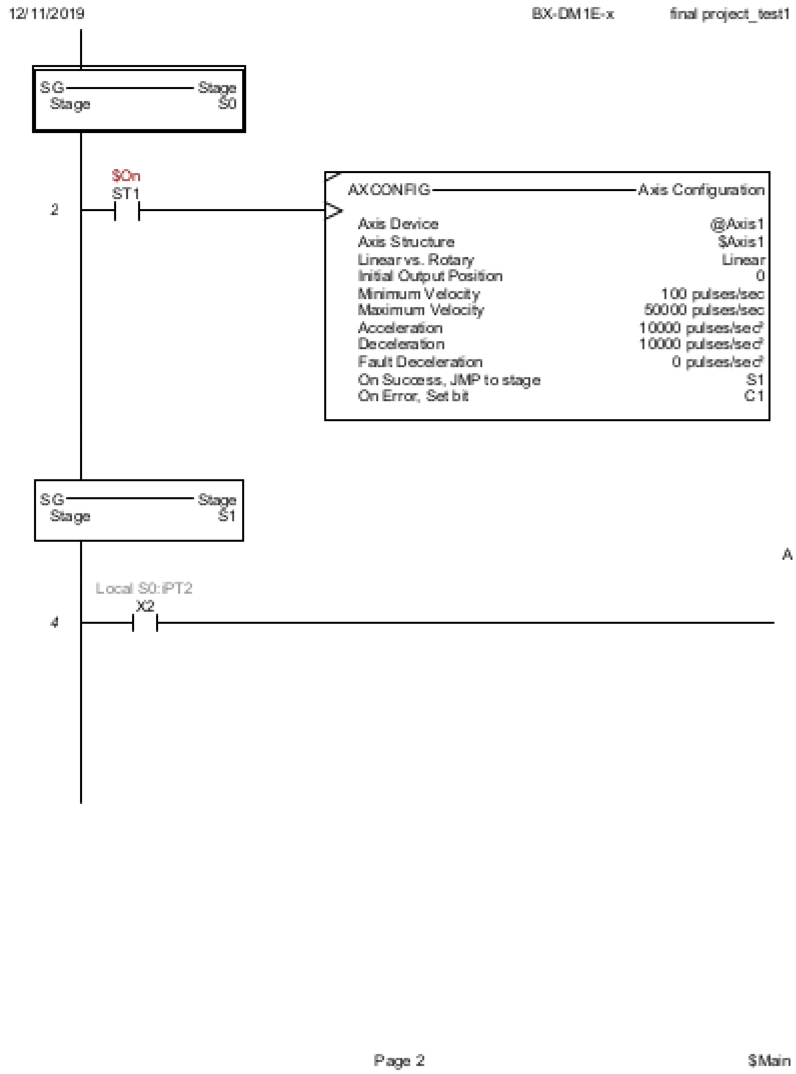
## Mechanical Drawings

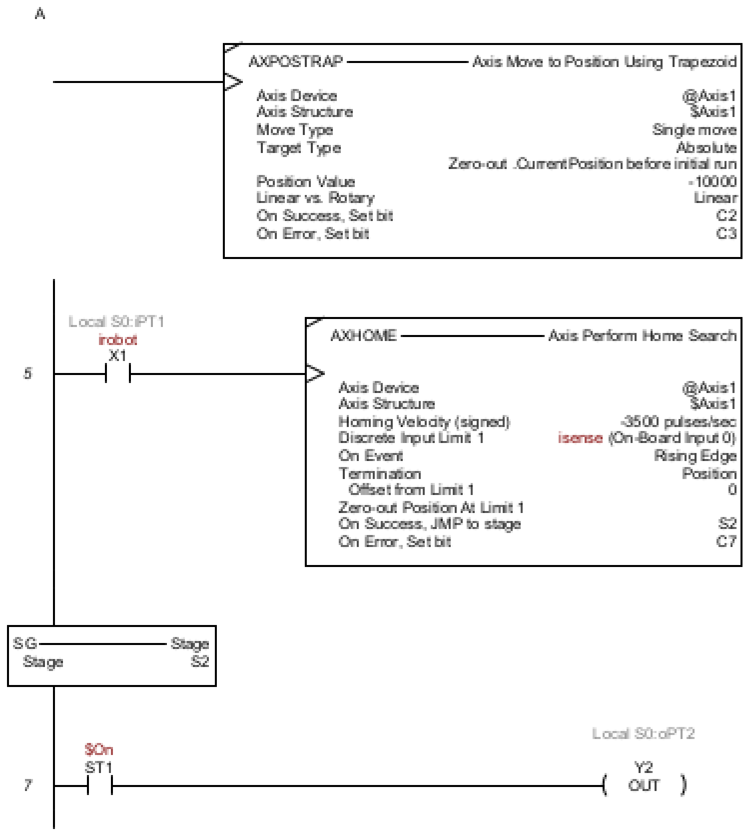
We were not able to access the mechanical drawings due to ransomware.

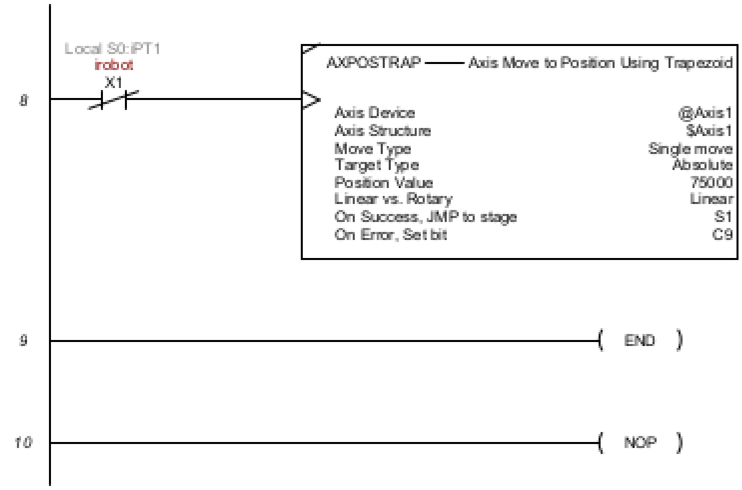
## Electronic Schematic



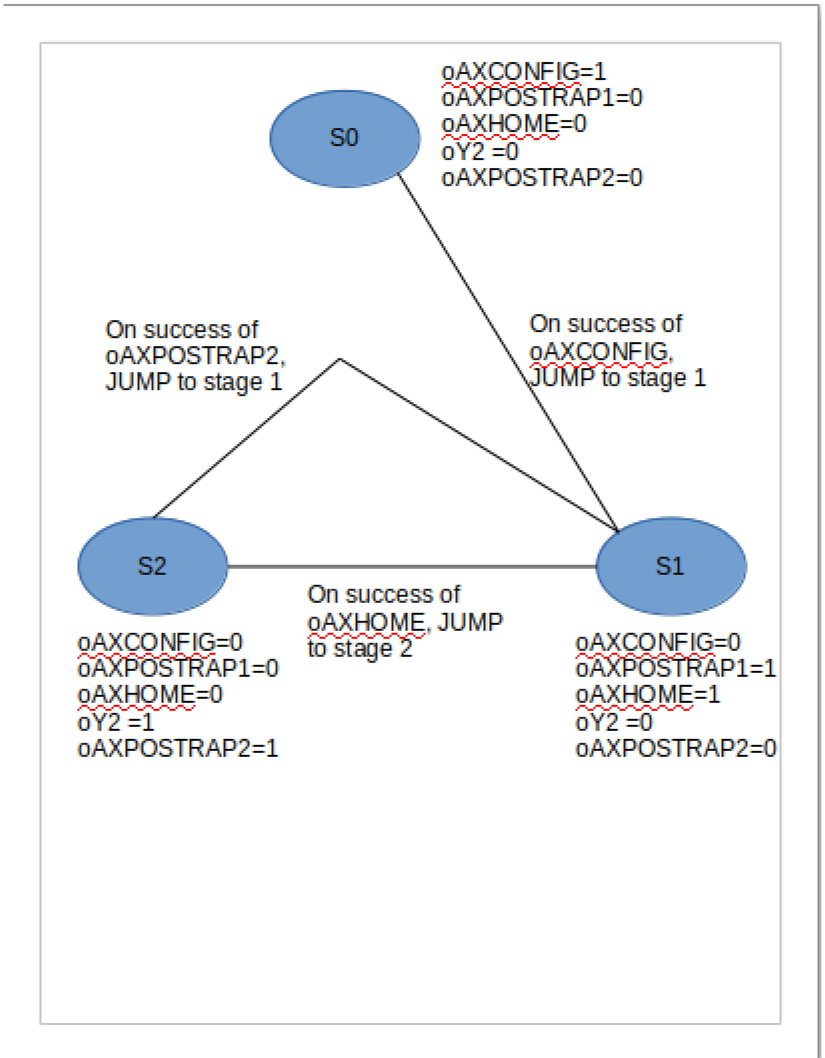
## PLC Ladder Logic




A
AXPOSTRAP
Axis Move to Position Using Trapezoid
Axis Device @Axis1
Axis Structure $Axis1
Move Type Single move
Target Type Absolute
Zero-out .CurrentPosition before initial run
Position Value -10000
Linear vs. Rotary Linear
On Success, Set bit C2
On Error, Set bit C3
Local S0:iPT1
irobot
X1
5
AXHOME
Axis Perform Home Search
Axis Device @Axis1
Axis Structure $Axis1
Homing Velocity (signed) -3500 pulses/sec
Discrete Input Limit 1 isense (On-Board Input 0)
On Event Rising Edge
Termination Position
Offset from Limit 1 0
Zero-out Position At Limit 1
On Success, JMP to stage S2
On Error, Set bit C7
SG
Stage
Stage S2
Local S0:oPT2
$On
ST1
7
Y2
OUT


Local S0:iPT1
irobot
X1
8
AXPOSTRAP —— Axis Move to Position Using Trapezoid
Axis Device @Axis1
Axis Structure $Axis1
Move Type Single move
Target Type Absolute
Position Value 75000
Linear vs. Rotary Linear
On Success, JMP to stage S1
On Error, Set bit C9
9
END
10
NOP

## State Machine Diagrams



## Motoman Job Programs

1. Parent function

```
/JOB
//NAME AAAFINAL2TATUM
//POS
///NPOS 107,0,0,0,0,0
///TOOL 0
///POSTYPE PULSE
///PULSE
C00000=1940,-1373,-63439,-235,45618,-103399
C00001=1960,23337,-47140,-244,48716,-103406
C00002=2366,78286,-43437,-447,79229,-103482
C00003=-7114,89450,-10530,86927,85123,-115870
C00004=14181,91315,-5716,75054,79776,-117276
C00005=21571,92408,-3141,71060,77613,-117811
C00006=10822,91022,-6465,76895,80682,-117112
C00007=10527,49573,-9373,78238,78851,-131431
C00008=2296,38851,-23889,2674,41291,-104688
C00009=-31626,36310,-27666,29631,46948,-116114
C00010=-31713,50957,-30284,26056,56231,-112028
C00011=-36773,43326,-25443,32174,50321,-116360
C00012=-43656,56733,-20457,31683,54269,-114877
C00013=-46997,58770,-17496,34221,54256,-115996
C00014=-46997,58770,-17496,34221,54254,-115996
C00015=-48023,57044,-20256,34556,55241,-115787
C00016=-48023,57044,-20257,34556,55235,-115787
C00017=-44723,54959,-23481,32022,55389,-114640
```

C00018=-44723,54959,-23482,32022,55382,-114641

C00019=-43656,56733,-20457,31683,54269,-114877

C00020=-43655,56734,-20458,31682,54265,-114881

C00021=-46997,58770,-17496,34221,54256,-115996

C00022=-46997,58770,-17496,34221,54254,-115996

C00023=-48023,57044,-20256,34556,55241,-115787

C00024=-48023,57044,-20257,34556,55235,-115787

C00025=-44723,54959,-23481,32022,55389,-114640

C00026=-44723,54959,-23482,32022,55382,-114641

C00027=-43656,56733,-20457,31683,54269,-114877

C00028=-43655,56734,-20458,31682,54265,-114881

C00029=-46997,58770,-17496,34221,54256,-115996

C00030=-46997,58770,-17496,34221,54254,-115996

C00031=-48023,57044,-20256,34556,55241,-115787

C00032=-48023,57044,-20257,34556,55235,-115787

C00033=-44723,54959,-23481,32022,55389,-114640

C00034=-44723,54959,-23482,32022,55382,-114641

C00035=-43656,56733,-20457,31683,54269,-114877

C00036=-43655,56734,-20458,31682,54265,-114881

C00037=-46997,58770,-17496,34221,54256,-115996

C00038=-46997,58770,-17496,34221,54254,-115996

C00039=-48023,57044,-20256,34556,55241,-115787

C00040=-48023,57044,-20257,34556,55235,-115787

C00041=-44723,54959,-23481,32022,55389,-114640

C00042=-44723,54959,-23482,32022,55382,-114641

C00043=-43656,56733,-20457,31683,54269,-114877

C00044=-43655,56734,-20458,31682,54265,-114881

C00045=-46997,58770,-17496,34221,54256,-115996

C00046=-46997,58770,-17496,34221,54254,-115996

C00047=-48023,57044,-20256,34556,55241,-115787

C00048=-48023,57044,-20257,34556,55235,-115787

C00049=-44723,54959,-23481,32022,55389,-114640

C00050=-44723,54959,-23482,32022,55382,-114641

C00051=-43656,56733,-20457,31683,54269,-114877

C00052=-43655,56734,-20458,31682,54265,-114881

C00053=-46997,58770,-17496,34221,54256,-115996

C00054=-46997,58770,-17496,34221,54254,-115996

C00055=-48023,57044,-20256,34556,55241,-115787

C00056=-48023,57044,-20257,34556,55235,-115787

C00057=-44723,54959,-23481,32022,55389,-114640

C00058=-44723,54959,-23482,32022,55382,-114641

C00059=-43656,56733,-20457,31683,54269,-114877

C00060=-43655,56734,-20458,31682,54265,-114881

C00061=-46997,58770,-17496,34221,54256,-115996

C00062=-46997,58770,-17496,34221,54254,-115996

C00063=-48023,57044,-20256,34556,55241,-115787

C00064=-48023,57044,-20257,34556,55235,-115787

C00065=-44723,54959,-23481,32022,55389,-114640

C00066=-44723,54959,-23482,32022,55382,-114641

C00067=-43656,56733,-20457,31683,54269,-114877

C00068=-43655,56734,-20458,31682,54265,-114881

C00069=-46997,58770,-17496,34221,54256,-115996

C00070=-46997,58770,-17496,34221,54254,-115996

C00071=-48023,57044,-20256,34556,55241,-115787

C00072=-48023,57044,-20257,34556,55235,-115787

C00073=-44723,54959,-23481,32022,55389,-114640

C00074=-44723,54959,-23482,32022,55382,-114641

C00075=-43656,56733,-20457,31683,54269,-114877

C00076=-43655,56734,-20458,31682,54265,-114881

C00077=-46997,58770,-17496,34221,54256,-115996

C00078=-46997,58770,-17496,34221,54254,-115996

C00079=-48023,57044,-20256,34556,55241,-115787

C00080=-48023,57044,-20257,34556,55235,-115787

C00081=-44723,54959,-23481,32022,55389,-114640

C00082=-44723,54959,-23482,32022,55382,-114641

C00083=-43656,56733,-20457,31683,54269,-114877

C00084=-43655,56734,-20458,31682,54265,-114881

C00085=-46997,58770,-17496,34221,54256,-115996

C00086=-46997,58770,-17496,34221,54254,-115996

C00087=-48023,57044,-20256,34556,55241,-115787

C00088=-48023,57044,-20257,34556,55235,-115787

C00089=-44723,54959,-23481,32022,55389,-114640

C00090=-44723,54959,-23482,32022,55382,-114641

C00091=-43656,56733,-20457,31683,54269,-114877

C00092=-43655,56734,-20458,31682,54265,-114881

C00093=-46997,58770,-17496,34221,54256,-115996

C00094=-46997,58770,-17496,34221,54254,-115996

C00095=-48023,57044,-20256,34556,55241,-115787

C00096=-48023,57044,-20257,34556,55235,-115787

C00097=-44723,54959,-23481,32022,55389,-114640

C00098=-44723,54959,-23482,32022,55382,-114641

C00099=-43656,56733,-20457,31683,54269,-114877

C00100=-43655,56734,-20458,31682,54265,-114881

C00101=-46997,58770,-17496,34221,54256,-115996

C00102=-46997,58770,-17496,34221,54254,-115996

C00103=-48023,57044,-20256,34556,55241,-115787

C00104=-48023,57044,-20257,34556,55235,-115787

```
C00105=-44723,54959,-23481,32022,55389,-114640
C00106=-44723,54959,-23482,32022,55382,-114641
//INST
///DATE 2019/12/05 18:41
///ATTR SC,RW
///GROUP1 RB1
NOP
DOUT OT#(1) OFF
MOVJ C00000 VJ=10.00
MOVJ C00001 VJ=10.00
MOVJ C00002 VJ=10.00
MOVJ C00003 VJ=10.00
MOVJ C00004 VJ=10.00
MOVJ C00005 VJ=2.00
CALL JOB:AAASENSORTATUM IF IN#(1)=ON
MOVJ C00006 VJ=2.00
MOVJ C00007 VJ=10.00
MOVJ C00008 VJ=10.00
MOVJ C00009 VJ=10.00
MOVJ C00010 VJ=1.00
DOUT OT#(6) ON
TIMER T=3.00
MOVJ C00011 VJ=1.00
DOUT OT#(6) OFF
MOVJ C00012 VJ=0.78
DOUT OT#(5) ON
MOVJ C00013 VJ=0.78
CALL JOB:AAATATUMSWAP IF IN#(2)=ON
MOVJ C00014 VJ=0.78
```

```
DOUT OT#(5) ON
MOVJ C00015 VJ=0.78
CALL JOB:AAATATUMSWAP IF IN#(2)=ON
MOVJ C00016 VJ=0.78
DOUT OT#(5) ON
MOVJ C00017 VJ=0.78
CALL JOB:AAATATUMSWAP IF IN#(2)=ON
MOVJ C00018 VJ=0.78
DOUT OT#(5) ON
MOVJ C00019 VJ=0.78
CALL JOB:AAATATUMSWAP IF IN#(2)=ON
MOVJ C00020 VJ=0.78
DOUT OT#(5) ON
MOVJ C00021 VJ=0.78
CALL JOB:AAATATUMSWAP IF IN#(2)=ON
MOVJ C00022 VJ=0.78
DOUT OT#(5) ON
MOVJ C00023 VJ=0.78
CALL JOB:AAATATUMSWAP IF IN#(2)=ON
MOVJ C00024 VJ=0.78
DOUT OT#(5) ON
MOVJ C00025 VJ=0.78
CALL JOB:AAATATUMSWAP IF IN#(2)=ON
MOVJ C00026 VJ=0.78
DOUT OT#(5) ON
MOVJ C00027 VJ=0.78
CALL JOB:AAATATUMSWAP IF IN#(2)=ON
MOVJ C00028 VJ=0.78
DOUT OT#(5) ON
```

```
MOVJ C00029 VJ=0.78
CALL JOB:AAATATUMSWAP IF IN#(2)=ON
MOVJ C00030 VJ=0.78
DOUT OT#(5) ON
MOVJ C00031 VJ=0.78
CALL JOB:AAATATUMSWAP IF IN#(2)=ON
MOVJ C00032 VJ=0.78
DOUT OT#(5) ON
MOVJ C00033 VJ=0.78
CALL JOB:AAATATUMSWAP IF IN#(2)=ON
MOVJ C00034 VJ=0.78
DOUT OT#(5) ON
MOVJ C00035 VJ=0.78
CALL JOB:AAATATUMSWAP IF IN#(2)=ON
MOVJ C00036 VJ=0.78
DOUT OT#(5) ON
MOVJ C00037 VJ=0.78
CALL JOB:AAATATUMSWAP IF IN#(2)=ON
MOVJ C00038 VJ=0.78
DOUT OT#(5) ON
MOVJ C00039 VJ=0.78
CALL JOB:AAATATUMSWAP IF IN#(2)=ON
MOVJ C00040 VJ=0.78
DOUT OT#(5) ON
MOVJ C00041 VJ=0.78
CALL JOB:AAATATUMSWAP IF IN#(2)=ON
MOVJ C00042 VJ=0.78
DOUT OT#(5) ON
MOVJ C00043 VJ=0.78
```

```
CALL JOB:AAATATUMSWAP IF IN#(2)=ON
MOVJ C00044 VJ=0.78
DOUT OT#(5) ON
MOVJ C00045 VJ=0.78
CALL JOB:AAATATUMSWAP IF IN#(2)=ON
MOVJ C00046 VJ=0.78
DOUT OT#(5) ON
MOVJ C00047 VJ=0.78
CALL JOB:AAATATUMSWAP IF IN#(2)=ON
MOVJ C00048 VJ=0.78
DOUT OT#(5) ON
MOVJ C00049 VJ=0.78
CALL JOB:AAATATUMSWAP IF IN#(2)=ON
MOVJ C00050 VJ=0.78
DOUT OT#(5) ON
MOVJ C00051 VJ=0.78
CALL JOB:AAATATUMSWAP IF IN#(2)=ON
MOVJ C00052 VJ=0.78
DOUT OT#(5) ON
MOVJ C00053 VJ=0.78
CALL JOB:AAATATUMSWAP IF IN#(2)=ON
MOVJ C00054 VJ=0.78
DOUT OT#(5) ON
MOVJ C00055 VJ=0.78
CALL JOB:AAATATUMSWAP IF IN#(2)=ON
MOVJ C00056 VJ=0.78
DOUT OT#(5) ON
MOVJ C00057 VJ=0.78
CALL JOB:AAATATUMSWAP IF IN#(2)=ON
```

```
MOVJ C00058 VJ=0.78
DOUT OT#(5) ON
MOVJ C00059 VJ=0.78
CALL JOB:AAATATUMSWAP IF IN#(2)=ON
MOVJ C00060 VJ=0.78
DOUT OT#(5) ON
MOVJ C00061 VJ=0.78
CALL JOB:AAATATUMSWAP IF IN#(2)=ON
MOVJ C00062 VJ=0.78
DOUT OT#(5) ON
MOVJ C00063 VJ=0.78
CALL JOB:AAATATUMSWAP IF IN#(2)=ON
MOVJ C00064 VJ=0.78
DOUT OT#(5) ON
MOVJ C00065 VJ=0.78
CALL JOB:AAATATUMSWAP IF IN#(2)=ON
MOVJ C00066 VJ=0.78
DOUT OT#(5) ON
MOVJ C00067 VJ=0.78
CALL JOB:AAATATUMSWAP IF IN#(2)=ON
MOVJ C00068 VJ=0.78
DOUT OT#(5) ON
MOVJ C00069 VJ=0.78
CALL JOB:AAATATUMSWAP IF IN#(2)=ON
MOVJ C00070 VJ=0.78
DOUT OT#(5) ON
MOVJ C00071 VJ=0.78
CALL JOB:AAATATUMSWAP IF IN#(2)=ON
MOVJ C00072 VJ=0.78
```

```
DOUT OT#(5) ON
MOVJ C00073 VJ=0.78
CALL JOB:AAATATUMSWAP IF IN#(2)=ON
MOVJ C00074 VJ=0.78
DOUT OT#(5) ON
MOVJ C00075 VJ=0.78
JUMP JOB:AAATATUMHOME IF IN#(2)=ON
MOVJ C00076 VJ=0.78
DOUT OT#(5) ON
MOVJ C00077 VJ=0.78
JUMP JOB:AAATATUMHOME IF IN#(2)=ON
MOVJ C00078 VJ=0.78
DOUT OT#(5) ON
MOVJ C00079 VJ=0.78
JUMP JOB:AAATATUMHOME IF IN#(2)=ON
MOVJ C00080 VJ=0.78
DOUT OT#(5) ON
MOVJ C00081 VJ=0.78
JUMP JOB:AAATATUMHOME IF IN#(2)=ON
MOVJ C00082 VJ=0.78
DOUT OT#(5) ON
MOVJ C00083 VJ=0.78
JUMP JOB:AAATATUMHOME IF IN#(2)=ON
MOVJ C00084 VJ=0.78
DOUT OT#(5) ON
MOVJ C00085 VJ=0.78
JUMP JOB:AAATATUMHOME IF IN#(2)=ON
MOVJ C00086 VJ=0.78
DOUT OT#(5) ON
```

```
MOVJ C00087 VJ=0.78

JUMP JOB:AAATATUMHOME IF IN#(2)=ON

MOVJ C00088 VJ=0.78

DOUT OT#(5) ON

MOVJ C00089 VJ=0.78

JUMP JOB:AAATATUMHOME IF IN#(2)=ON

MOVJ C00090 VJ=0.78

DOUT OT#(5) ON

MOVJ C00091 VJ=0.78

JUMP JOB:AAATATUMHOME IF IN#(2)=ON

MOVJ C00092 VJ=0.78

DOUT OT#(5) ON

MOVJ C00093 VJ=0.78

JUMP JOB:AAATATUMHOME IF IN#(2)=ON

MOVJ C00094 VJ=0.78

DOUT OT#(5) ON

MOVJ C00095 VJ=0.78

JUMP JOB:AAATATUMHOME IF IN#(2)=ON

MOVJ C00096 VJ=0.78

DOUT OT#(5) ON

MOVJ C00097 VJ=0.78

JUMP JOB:AAATATUMHOME IF IN#(2)=ON

MOVJ C00098 VJ=0.78

DOUT OT#(5) ON

MOVJ C00099 VJ=0.78

JUMP JOB:AAATATUMHOME IF IN#(2)=ON

MOVJ C00100 VJ=0.78

DOUT OT#(5) ON

MOVJ C00101 VJ=0.78
```

```
JUMP JOB:AAATATUMHOME IF IN#(2)=ON
MOVJ C00102 VJ=0.78
DOUT OT#(5) ON
MOVJ C00103 VJ=0.78
JUMP JOB:AAATATUMHOME IF IN#(2)=ON
MOVJ C00104 VJ=0.78
DOUT OT#(5) ON
MOVJ C00105 VJ=0.78
JUMP JOB:AAATATUMHOME IF IN#(2)=ON
MOVJ C00106 VJ=0.78
DOUT OT#(5) ON
END
```

2. Swap Function

```
/JOB
//NAME AAATATUMSWAP
//POS
///NPOS 11,0,0,0,0,0
///TOOL 0
///POSTYPE PULSE
///PULSE
C00000=-44836,50372,-15287,34619,48926,-118239
C00001=-5939,23972,-49157,4000,50332,-104966
C00002=-6020,17903,-42916,18788,8473,-113879
C00003=-5146,54882,-34525,9323,12432,-107763
C00004=-5312,65809,-39263,2976,69060,-104016
C00005=16359,89499,-7810,72355,77399,-116576
C00006=22361,90847,-5033,69146,75595,-117103
```

```
C00007=14959,89250,-8333,73117,77790,-116477
C00008=-5941,23971,-49159,3999,50327,-104969
C00009=-28619,49911,-32632,19514,56557,-109796
C00010=-44838,50374,-15287,34621,48921,-118248
//INST
///DATE 2019/12/05 18:42
///ATTR SC,RW
///GROUP1 RB1
NOP
MOVJ C00000 VJ=2.00
DOUT OT#(5) OFF
MOVJ C00001 VJ=10.00
MOVJ C00002 VJ=10.00
MOVJ C00003 VJ=10.00
DOUT OT#(1) OFF
MOVJ C00004 VJ=10.00
MOVJ C00005 VJ=10.00
MOVJ C00006 VJ=2.00
CALL JOB:AAASENSORTATUM IF IN#(1)=ON
MOVJ C00007 VJ=2.00
MOVJ C00008 VJ=10.00
MOVJ C00009 VJ=2.00
DOUT OT#(6) ON
TIMER T=3.00
MOVJ C00010 VJ=2.00
DOUT OT#(6) OFF
END
```

3. Home function

```
/JOB
//NAME AAATATUMHOME
//POS
///NPOS 5,0,0,0,0,0
///TOOL 0
///POSTYPE PULSE
///PULSE
C00000=-44837,50375,-15287,34621,48926,-118247
C00001=-5939,23972,-49157,4000,50332,-104966
C00002=-6020,17903,-42916,18788,8473,-113879
C00003=-5146,54882,-34525,9323,12432,-107763
C00004=1939,-1374,-63442,-236,45614,-103399
//INST
///DATE 2019/12/05 03:07
///ATTR SC,RW
///GROUP1 RB1
NOP
MOVJ C00000 VJ=7.00
DOUT OT#(5) OFF
MOVJ C00001 VJ=7.00
MOVJ C00002 VJ=7.00
MOVJ C00003 VJ=7.00
DOUT OT#(1) OFF
MOVJ C00004 VJ=7.00
END
```

4. Sensor Function

```
/JOB
//NAME AAASENSORTATUM
//POS
///NPOS 0,0,0,0,0,0
//INST
///DATE 2019/12/03 16:07
///ATTR SC,RW
///GROUP1 RB1
NOP
DOUT OT#(1) ON
END
```

## Stepper Motor Drive Configuration

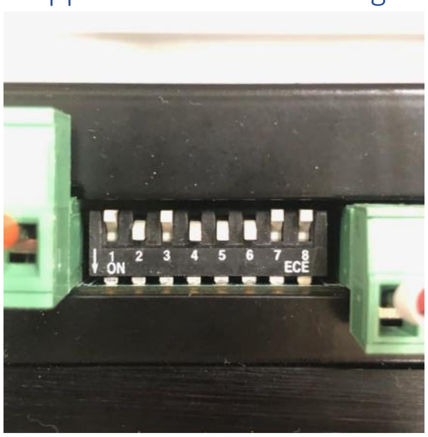



**Figure:** Stepper Motor Switch Configuration

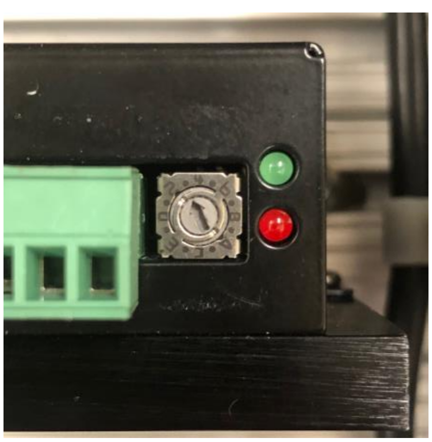


**Figure:** Current Setting

## Miscellaneous Pictures

**Figure:** Team photo


**Figure:** Milling Brackets for the Old NEMA 8 Motors

**Figure:** Group Discussing Robot Logic

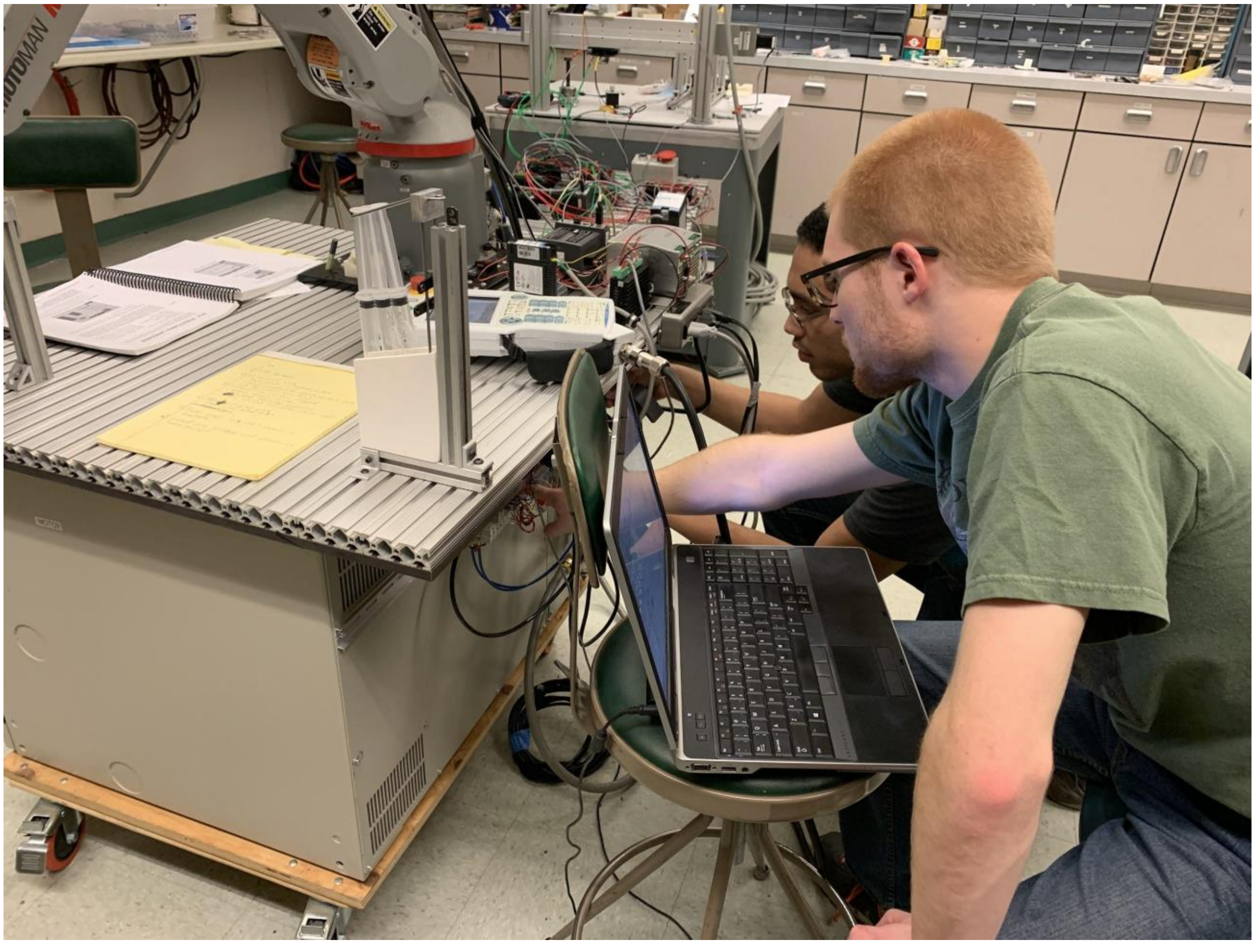



**Figure:** Testing PLC logic with Robot Interface

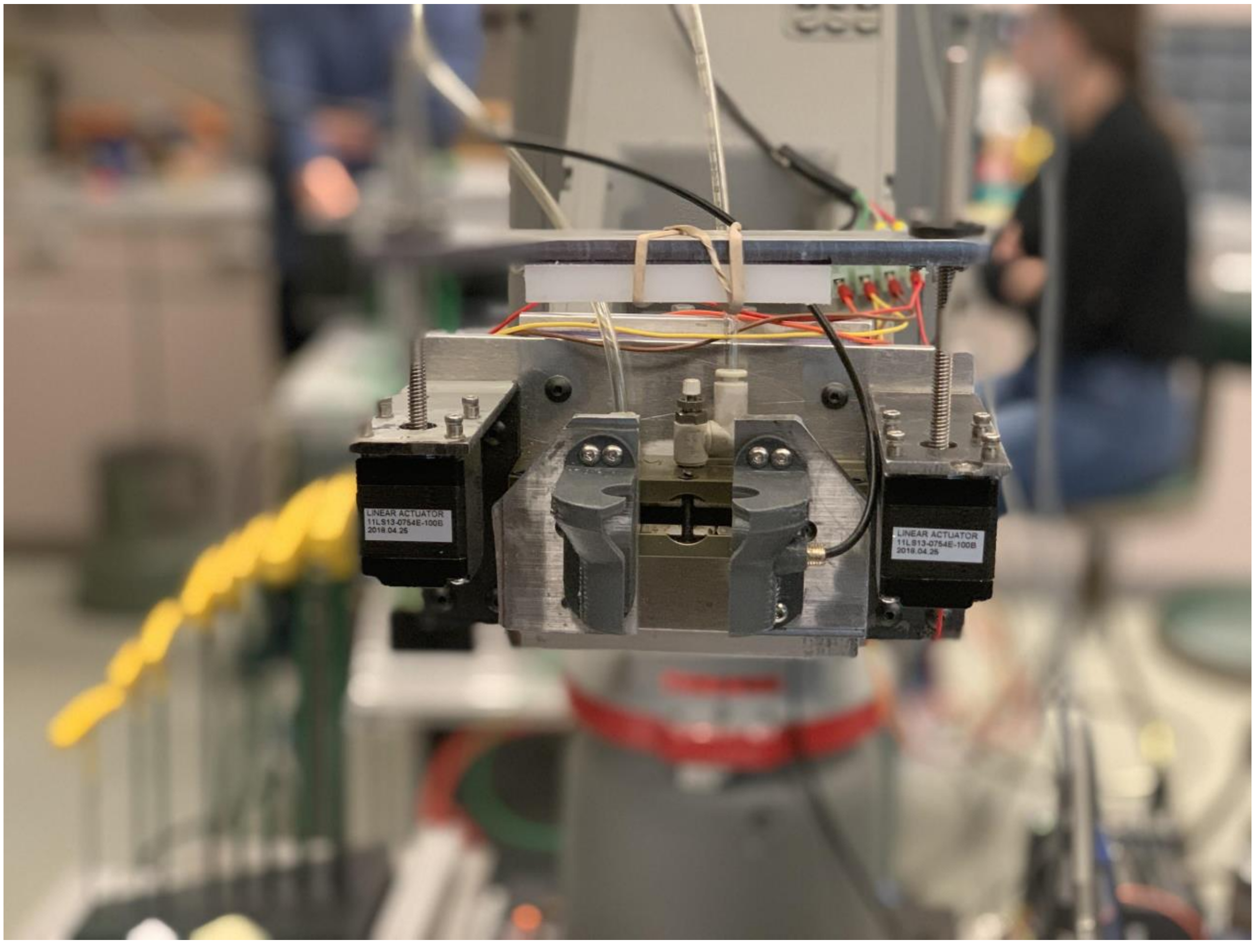



**Figure:** Coupler with Temporary Spacer

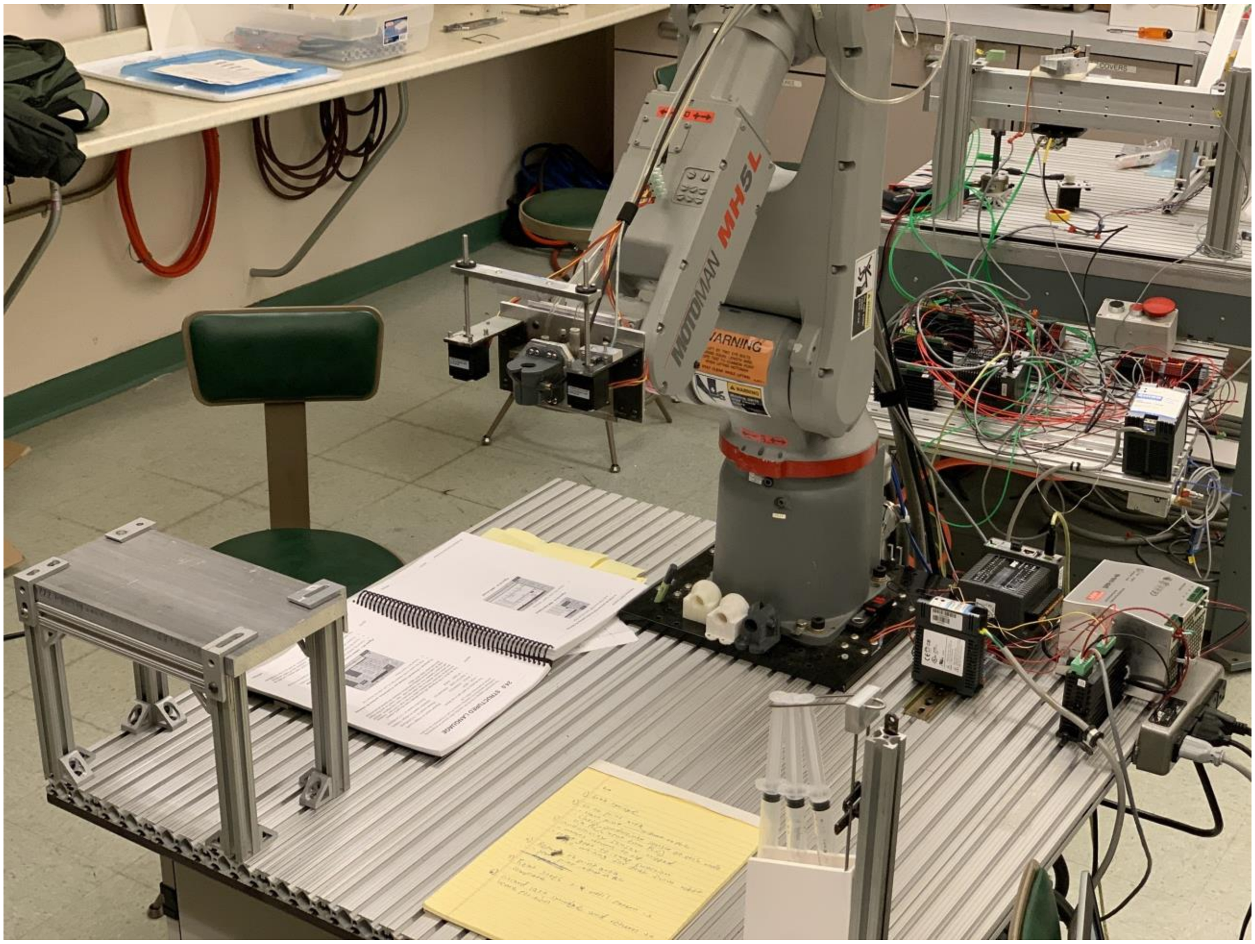



**Figure:** Final Workspace