MCT 331 Design of Mechatronic Systems (1)

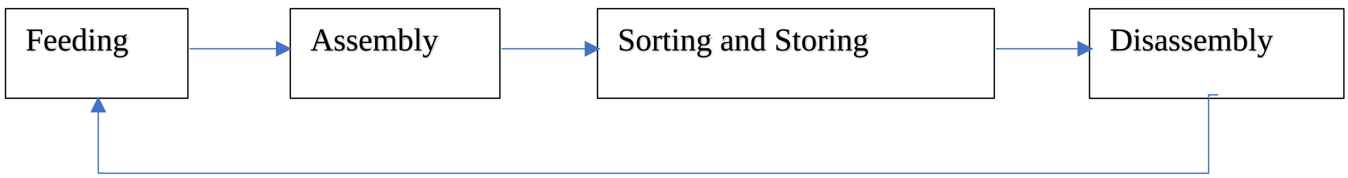
Individual Report
(Handling Subsystem)

Name: Mostafa Ashraf Ebrahim Ali ELFEEL
ID: 18P7611
Group: 2
Section: 1

Introduction

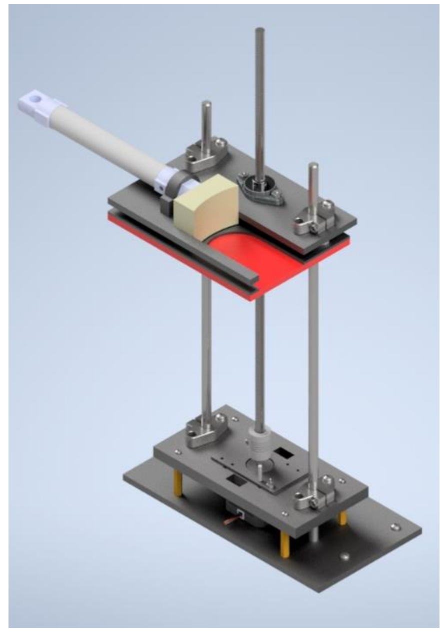
This subsystem Responsibility is to connect all 4 stations together. The Handling subsystem consists of three parts. First part is the assembly line (lower conveyor), second part is the elevator, and the third part is the disassembly line (upper conveyor).

The main part in handling subsystem is the elevator which focuses on lifting the product for the storing station to the disassembly line.
The lower conveyor moves the product from feeding station to assembly station then to storing station and the upper conveyor moves the part from elevator to disassembly station then back to feeding station connection all 4 stations together.



Design & Planning

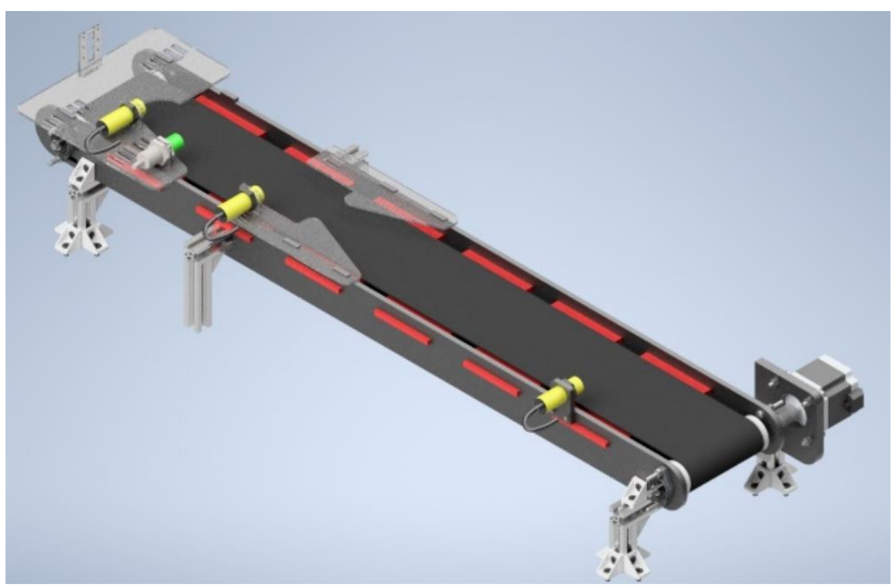
Elevator + pneumatic piston



2 DOF:
UP and Down(elevator).
Push and return(penumatic piston).
Actuation:
The movement is controlled by a stepper motor with a lead screw and a nut to turn the rotational motion into a linear one.
A pneumatic cylinder pushes the product to the conveyor.
Materials:
The materials selected for the elevator mainly laser cut wood (MDF) and the rod supports are aluminum and fixed by end rod supports.
The rest of the parts are standard
(Lead screw, Nut, rigid coupler, linear bearings, bearings, and a pneumatic cylinder).

Operation: There are 2 limit switches in the elevator to implement the homing and top positions after that a signal from the sorting is sent to elevator to start moving it to top until the second limit switch is triggered then that triggers the pneumatic piston. After that the pneumatic piston retracts and conveyor start moving.
Final motion is that the elevator returns to its homing position waiting for another signal from the sorting.

Assembly conveyor



The uses of IR sensors are preferable in these places to do the next.

First IR:

To stop feeding

Start conveyor

Second IR:

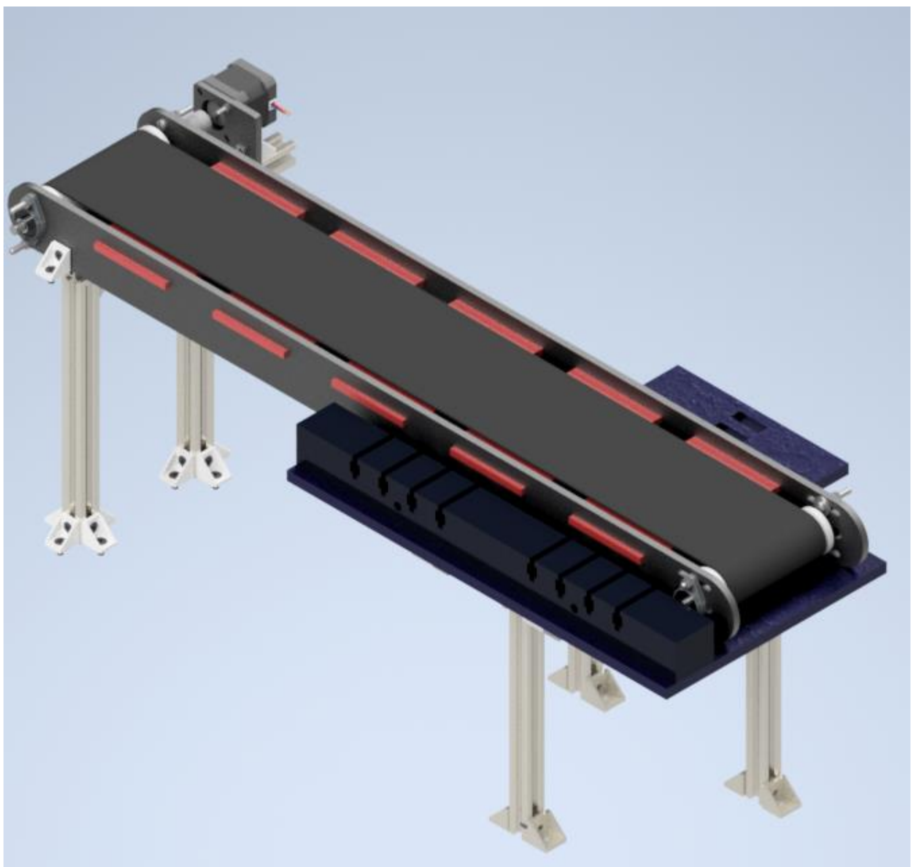
Starts assembly

Third IR:

Stops conveyor
Starts sorting

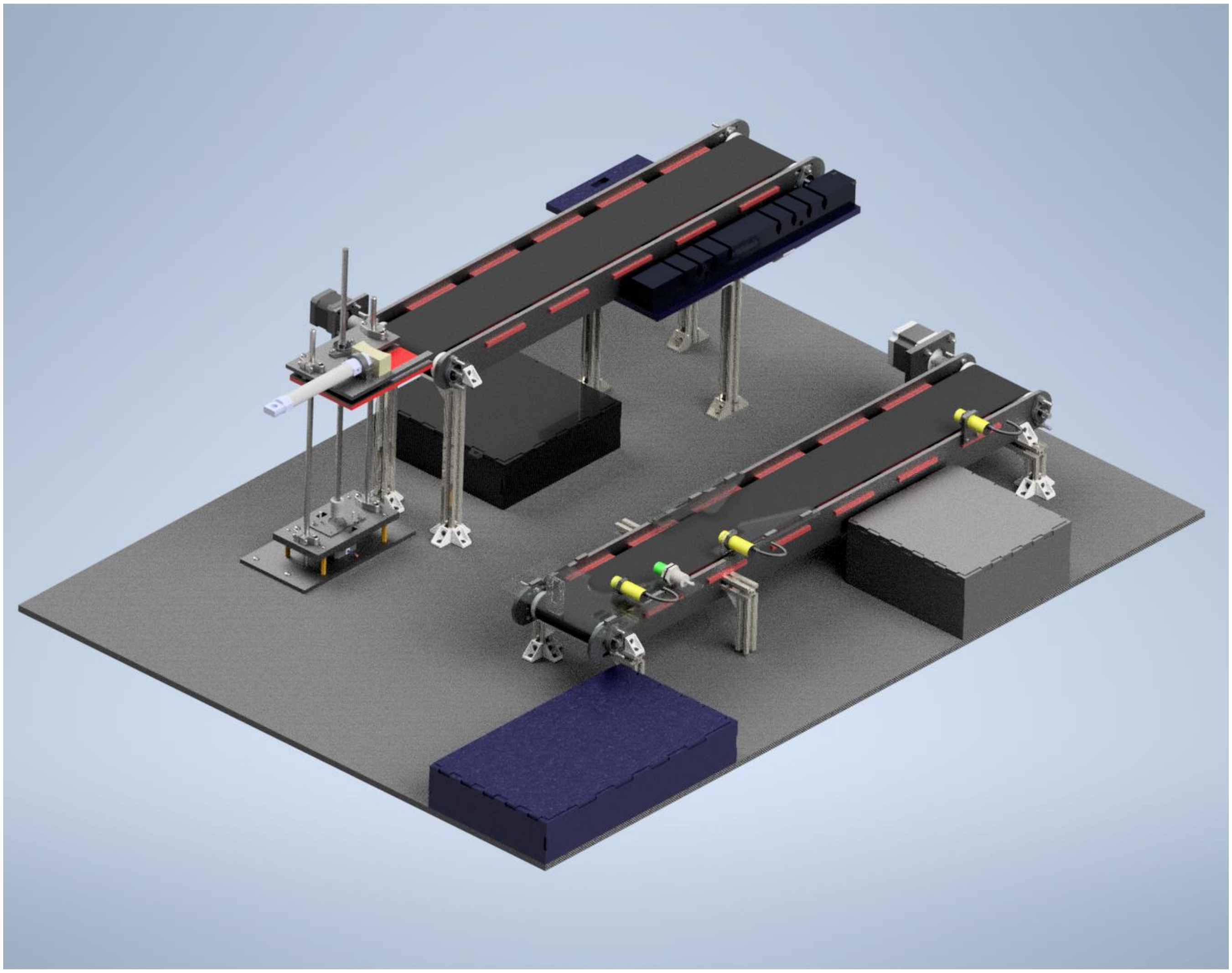
Disassembly conveyor

1 linear DOF.
Actuation:
A motor connected to a pully with a coupler, the pully rotates a meter long belt by friction between them, under the belt are wooden planks to support the weights on the belt, these planks are connected to two side walls to hold them in place, all of that is supported by 4 or more supports (if needed) and 4 bearings to Eliminate difference in speeds between static and rotating parts.
Material:
Most of conveyor supports are made of laser cut wood and the pully is made of plastic polymer and the rest are standard parts.
Note: initially the motors will be supported by v-slots but we're planning to change that.
I removed it from Figure 2 to eliminate confusion as there would be two different supports behind each other.

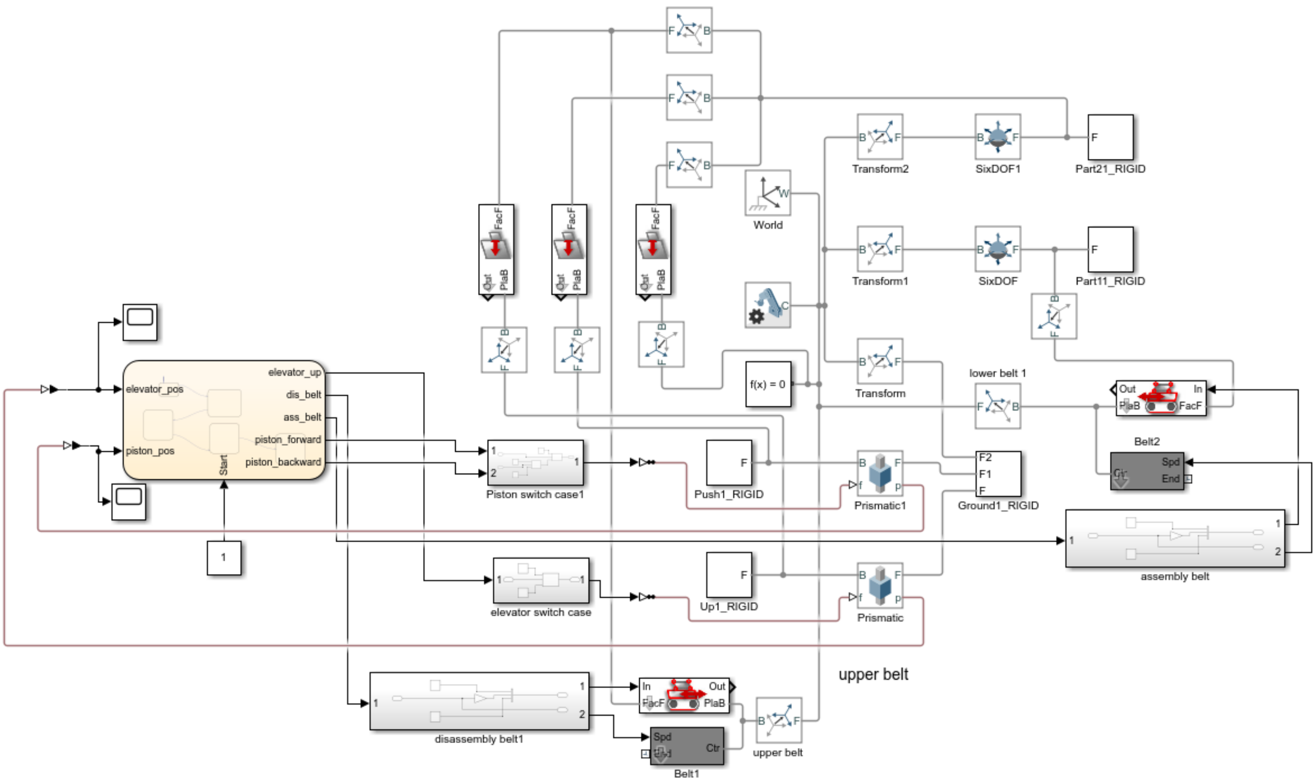


1. Inventor (CAD)

The full implementation of our Handling subsystem.



2. MATLAB and Implementations

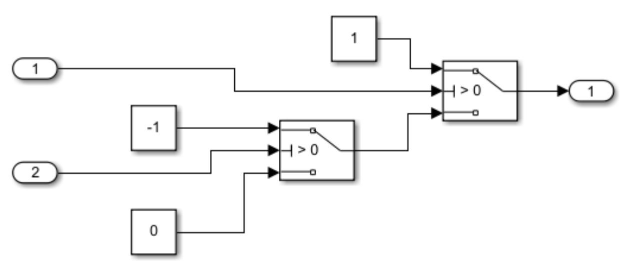


In the MATLAB simulation we used a controller on each joint to supply torque or force. The controller consisted of 2 switches. This type of controller controls positive direction, negative direction, and a normal 0 Force supply. The signal given to the switches is “2 bit binary like signal” where:

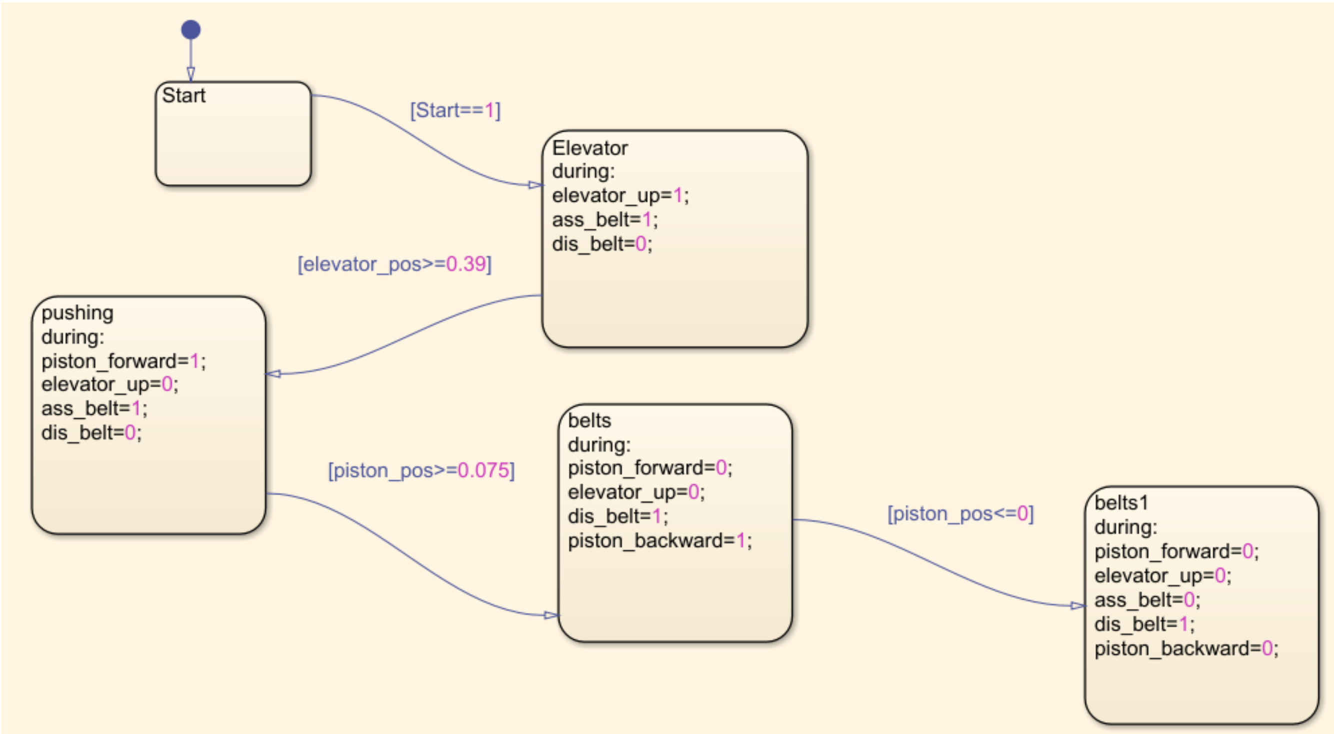
00 means stop.

10 means positive.

01 means negative.

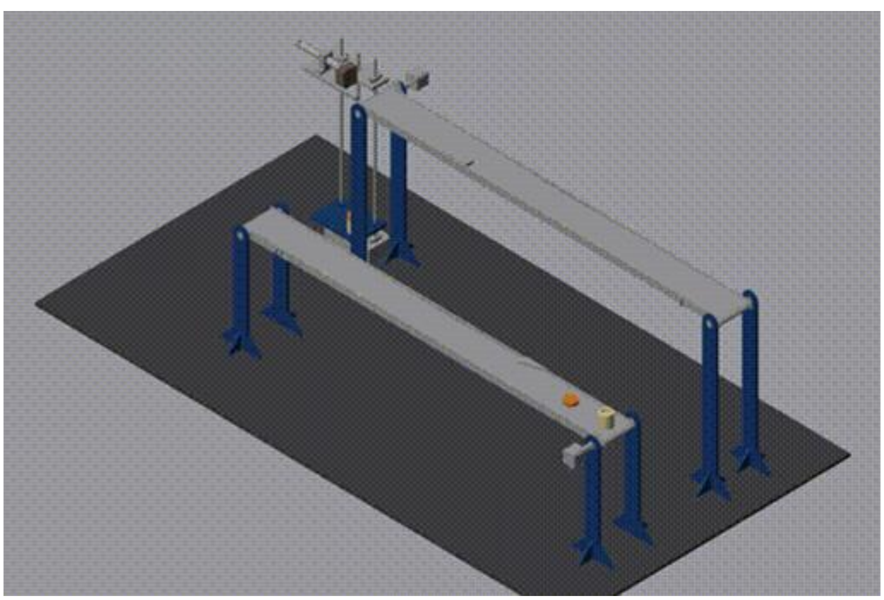


State-flow



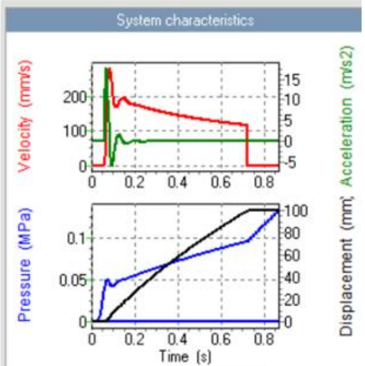
Our state-flow is a simple sequential flow that takes positions as input and outputs a high/low 0/1 signal to each controller depending on the state. Allowing for easier tracing and troubleshooting for the subsystem.

MATLAB simulation

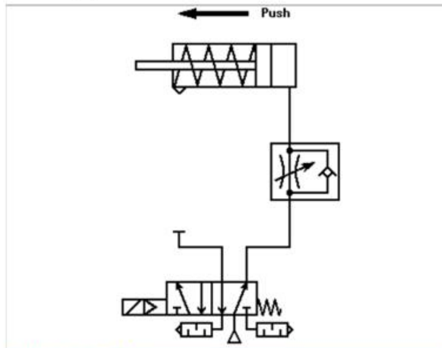


Motor sizing

Actuator Sizing (Pneumatic Sizing)



Full stroke time	0.72	s
Start up time	0.06	s
90% Force time	1.81	s
Mean velocity	139	mm/s
Max. velocity	283	mm/s
Stroke end velocity	157	mm/s
Max. acceleration	17.7	m/s2
Air consumption/cycle	0.431	dm3(ANR)
Required air flow	35.9	dm3/min(ANR)



Model changing >>> Results of model selection

Model	Part number	Quan.	Fitting 1	Quan.
Cylinder	C76E[]32-100S-[]	1		
Solenoid valve	NVFS11[]-[][][][]-01T	1	KQ2L04-U01	1
Manifold				
Silencer				
Quick exhaust valve				
Speed controller R	AS1000-M3-[]	1	KJL04-M3	1
Speed controller L				
Piping R	T0425[]-[]	1		
Piping L	T0425[]-[]			
Shock absorber				

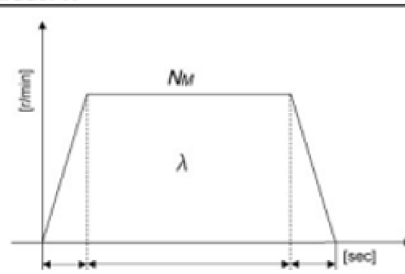
Input values

Stroke	100	mm
Full stroke time	1.00	s
Moving direction	Push (L)	
Supply pressure	0.35	MPa
Ambient temperature	20.0	degC
Total length (L)		m
Total length (R)	4	m
Speed controller position (L)		
		m
Opening		%
Speed controller position (R)	On cylinder	
	0	m
Opening	100	%
Load mass	0.20	kg
Resistance force		N
Mounting angle	0	deg
Application/Load rate	Transfer	
	0.7	
Friction factor	Sliding friction	
	0.30	
Selection method	Optimal size priority	

Motor elevator

Actuator Sizing (Elevator sizing)

■Operating Pattern



Travel Amount	λ =	203	[mm (deg)
Rotation Speed	N_M =	217.5	[r/min]
Acceleration Time	t_1 =	1	[sec]
Constant Speed Operation Time	t_2 =	6	[sec]
Deceleration Time	t_3 =	1	[sec]
Positioning Time	$t0$ =	8	[sec]

■Motor Selection Calculations

Screw Mechanism (Stepping Motor)

◆ Calculation of Inertia

● Screw Load Inertia J_B

$$J_B = \frac{\pi}{32}\rho L D^4 = \quad 0.013 \times 10^{-4}\ [\mathrm{kg \cdot m^2}]$$

● Load Inertia of Table and Work J_m

$$J_m = m\left(\frac{P}{2\pi}\right)^2 = \quad 0.002 \times 10^{-4}\ [\mathrm{kg \cdot m^2}]$$

● Moment of inertia of the transfer mechanism J_g

$$J_g = \frac{1}{8} m_{g1} D_{g1}^2 + \frac{1}{8} m_{g2} \frac{D_{g2}^2}{i_g^2} + \frac{1}{4} m_{gb} D_{g1}^2 = \quad - \times 10^{-4}\ [\mathrm{kg \cdot m^2}]$$

● Total Load Inertia J_L

$$J_L = \frac{(J_B + J_m)}{i_g^2} + J_c + J_g = \quad 0.015 \times 10^{-4}\ [\mathrm{kg \cdot m^2}]$$

◆ Calculation of Required Torque

● Load in Axial Direction of Table and Work F

$$F = F_A + mg(\sin\alpha + \mu\cos\alpha) = \quad 1.471\ [\mathrm{N}]$$

● Load Torque T_L

$$T_L = \left(\frac{FP}{2\pi\eta} + \frac{\mu_0 F_0 P}{2\pi}\right) \times \frac{1}{i_g \eta_g} = \quad 0.006\ [\mathrm{N \cdot m}]$$

● Number of Operating Pulses A — Step angle θs [deg]

$$A = \frac{\lambda}{P} \times \frac{360°}{\theta s} \times i_g = \quad 25375\ [\mathrm{Pulse}]$$

* If it is not an integer, an error will occur in the actual traveling amount.

● Operating Pulse Speed f

$$f = \frac{A}{t_0 - t_1} \text{ or } \frac{A}{t_0} = \quad 3625\ [\mathrm{Hz}]$$

● Acceleration Torque T_a Rotor inertia J_0 $n : (3.6° / \theta s) \cdot i$ — Gear Head Gear Ratio i

$$Ta = (J_0 \cdot i^2 + J_L)\frac{\pi \cdot \theta_s \cdot f}{180° \cdot t_1} \text{ or } (J_0 \cdot i^2 + J_L)\frac{\pi \cdot \theta_s \cdot f^2}{180° \cdot n} = \quad 0\ [\mathrm{N \cdot m}]$$

● Required Torque T

$$T = T_a + T_L = \quad 0.006\ [\mathrm{N \cdot m}]$$

◆ Calculation of the selected motor judgment item (Safety Factor S_f = 2)

● Safety factor S Motor torque of the rotation speed Tr 0.022 [N·m]

$$S = \frac{Tr}{T} = \quad 3.45$$

● Inertia ratio β

$$\beta = \frac{J_L}{J_0 \times i^2} = \quad 5.61$$

● Acceleration/Deceleration Rate T_R

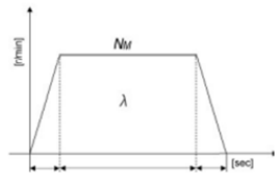
$$T_R = \frac{t_1}{f} = \quad 275.86$$

(Confirmation of acceleration/deceleration rate is not necessary for this motor.)

Conveyor Belt assembly:

Actuator Sizing (Conveyor assembly)

■Operating Pattern



Travel Amount	λ =	500	[mm (deg)
Rotation Speed	N_M =	79.58	[r/min]
Acceleration Time	t_1 =	1	[sec]
Constant Speed Operation Time	t_2 =	3	[sec]
Deceleration Time	t_3 =	1	[sec]
Positioning Time	t_0 =	5	[sec]

■Motor Selection Calculations

Belt Mechanism (Stepping Motor)

◆ Calculation of Inertia

- ●Pulley Load Inertia J_P

  $J_P = \frac{\pi}{32}\rho w D^4 = \frac{1}{8} m_P D^2 =$ 0.137 $\times 10^{-4}$ [kg·m^2]

- ●Load Inertia of Work and Belt J_m

  $J_m = \frac{1}{4} m D^2 =$ 1.125 $\times 10^{-4}$ [kg·m^2]

- ●Moment of inertia of the transfer mechanism J_g

  $J_g = \frac{1}{8} m_{g1} D_{g1}^2 + \frac{1}{8} m_{g2} \frac{D_{g2}^2}{i_g^2} + \frac{1}{4} m_{gb} D_{g1}^2 =$ – $\times 10^{-4}$ [kg·m^2]

- ●Total Load Inertia J_L

  $J_L = \frac{(J_P n + J_m)}{i_g^2} + J_s + J_g =$ 1.399 $\times 10^{-4}$ [kg·m^2]

◆ Calculation of Required Torque

- ●Frictional Force of Sliding Portion F

  $F = F_A + mg(\sin\alpha + \mu\cos\alpha) =$ 0.49 [N]

- ●Load Torque T_L

  $T_L = \frac{FD}{2\eta} \times \frac{1}{i_g \eta_g} =$ 0.008 [N·m]

- ●Number of Operating Pulses A — Step angle θs [deg]

  $A = \frac{\lambda}{\pi D} \times \frac{360°}{\theta s} \times i_g =$ 190986 [Pulse]

  * If it is not an integer, an error will occur in the actual traveling amount.

- ●Operating Pulse Speed f

  $f = \frac{A}{t_0 - t_1}$ or $\frac{A}{t_0} =$ 47746 [Hz]

- ●Acceleration Torque T_a Rotor inertia J_0 $n:(3.6°/\theta s) i$ — Gear Head Gear Ratio i

  $T_a = (J_0 i^2 + J_L)\frac{\pi \cdot \theta s \cdot f}{180° \cdot t_1}$ or $(J_0 i^2 + J_L)\frac{\pi \cdot \theta s \cdot f^2}{180° \cdot n} =$ 0.401 [N·m]

- ●Required Torque T

  $T = T_a + T_L =$ 0.409 [N·m]

◆ Calculation of the selected motor judgment item (Safety Factor S_f = 2)

- ●Safety factor S Motor torque of the rotation speed Tr 4.129 [N·m]

  $S = \frac{Tr}{T} =$ 10.1

- ● Inertia ratio β

  $\beta = \frac{J_L}{J_0 \times i^2} =$ 0

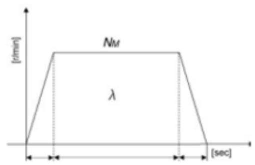
- ● Acceleration/Deceleration Rate T_R

  $T_R = \frac{t_1}{f} =$ 20.94 (Confirmation of acceleration/deceleration rate is not necessary for this motor.)

Conveyor belt Disassembly

Actuator Sizing (Conveyor Disassembly)

■Operating Pattern



Travel Amount	λ =	530	[mm (deg)
Rotation Speed	N_M =	78.47	[r/min]
Acceleration Time	t_1 =	1	[sec]
Constant Speed Operation Time	t_2 =	3.3	[sec]
Deceleration Time	t_3 =	1	[sec]
Positioning Time	t_0 =	5.3	[sec]

■Motor Selection Calculations

Belt Mechanism (Stepping Motor)

◆ Calculation of Inertia

●Pulley Load Inertia J_P

$J_P = \frac{\pi}{32}\rho w D^4 = \frac{1}{8} m_P D^2 =$ 0.137 $\times 10^{-4}$ [$kg \cdot m^2$]

●Load Inertia of Work and Belt J_m

$J_m = \frac{1}{4} m D^2 =$ 0.675 $\times 10^{-4}$ [$kg \cdot m^2$]

●Moment of inertia of the transfer mechanism J_g

$J_g = \frac{1}{8} m_{g1} D_{g1}^2 + \frac{1}{8} m_{g2} \frac{D_{g2}^2}{i_g^2} + \frac{1}{4} m_{gb} D_{g1}^2 =$ – $\times 10^{-4}$ [$kg \cdot m^2$]

●Total Load Inertia J_L

$J_L = \frac{(J_P n + J_m)}{i_g^2} + J_e + J_g =$ 0.949 $\times 10^{-4}$ [$kg \cdot m^2$]

◆ Calculation of Required Torque

●Frictional Force of Sliding Portion F

$F = F_A + mg(\sin\alpha + \mu\cos\alpha) =$ 0.294 [N]

●Load Torque T_L

$T_L = \frac{FD}{2\eta} \times \frac{1}{i_g \eta_g} =$ 0.005 [N·m]

●Number of Operating Pulses A Step angle θs [deg]

$A = \frac{\lambda}{\pi D} \times \frac{360°}{\theta s} \times i_g =$ 101222.6 [Pulse]

* If it is not an integer, an error will occur in the actual traveling amount.

●Operating Pulse Speed f

$f = \frac{A}{t_0 - t_1}$ or $\frac{A}{t_0} =$ 23540 [Hz]

●Acceleration Torque T_a Rotor inertia J_0 $n:(3.6°/\theta s)i$ Gear Head Gear Ratio i

$T_a = (J_0 i^2 + J_L)\frac{\pi \cdot \theta s \cdot f}{180° \cdot t_1}$ or $(J_0 i^2 + J_L)\frac{\pi \cdot \theta s \cdot f^2}{180° \cdot n} =$ 0.187 [N·m]

●Required Torque T

$T = T_a + T_L =$ 0.192 [N·m]

◆ Calculation of the selected motor judgment item (Safety Factor S_f = 2)

●Safety factor S Motor torque of the rotation speed Tr 1.357 [N·m]

$S = \frac{Tr}{T} =$ 7.06

● Inertia ratio β

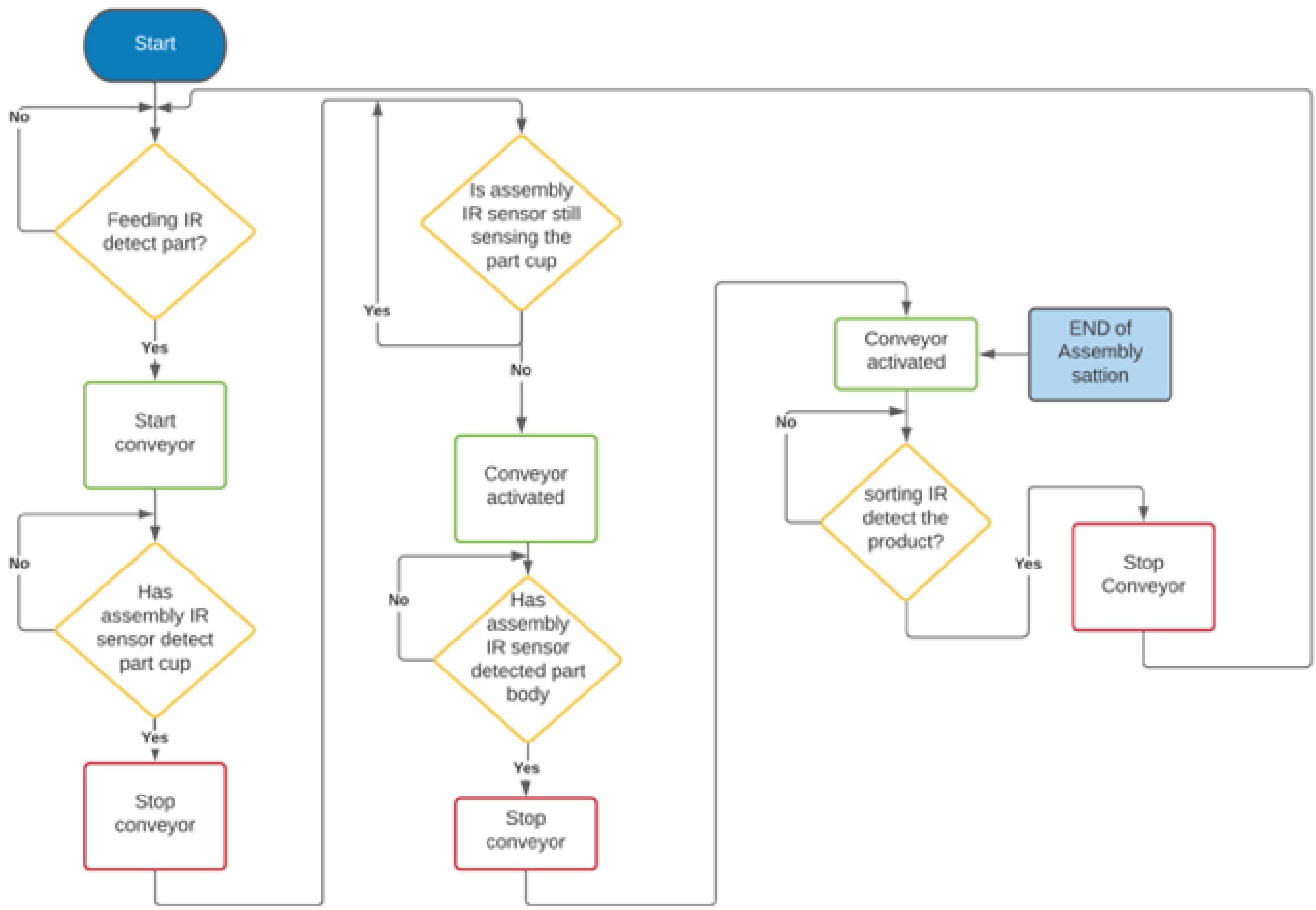
$\beta = \frac{J_L}{J_0 \times i^2} =$ 0

● Acceleration/Deceleration Rate T_R

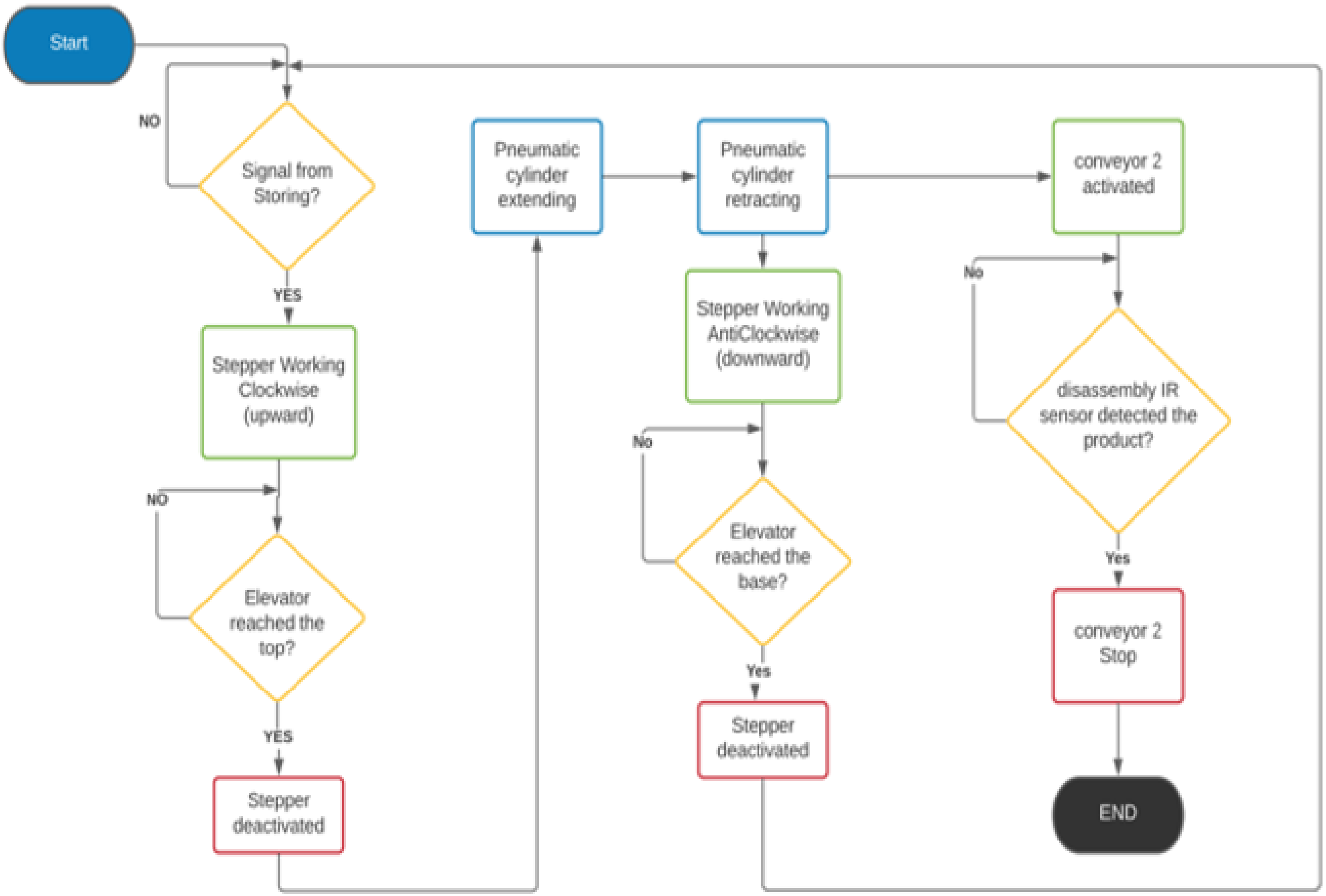
$T_R = \frac{t_1}{f} =$ 42.48 [ms / kHz]

Flow Chart

Assembly conveyor



Elevator + Disassembly conveyor



Conclusion

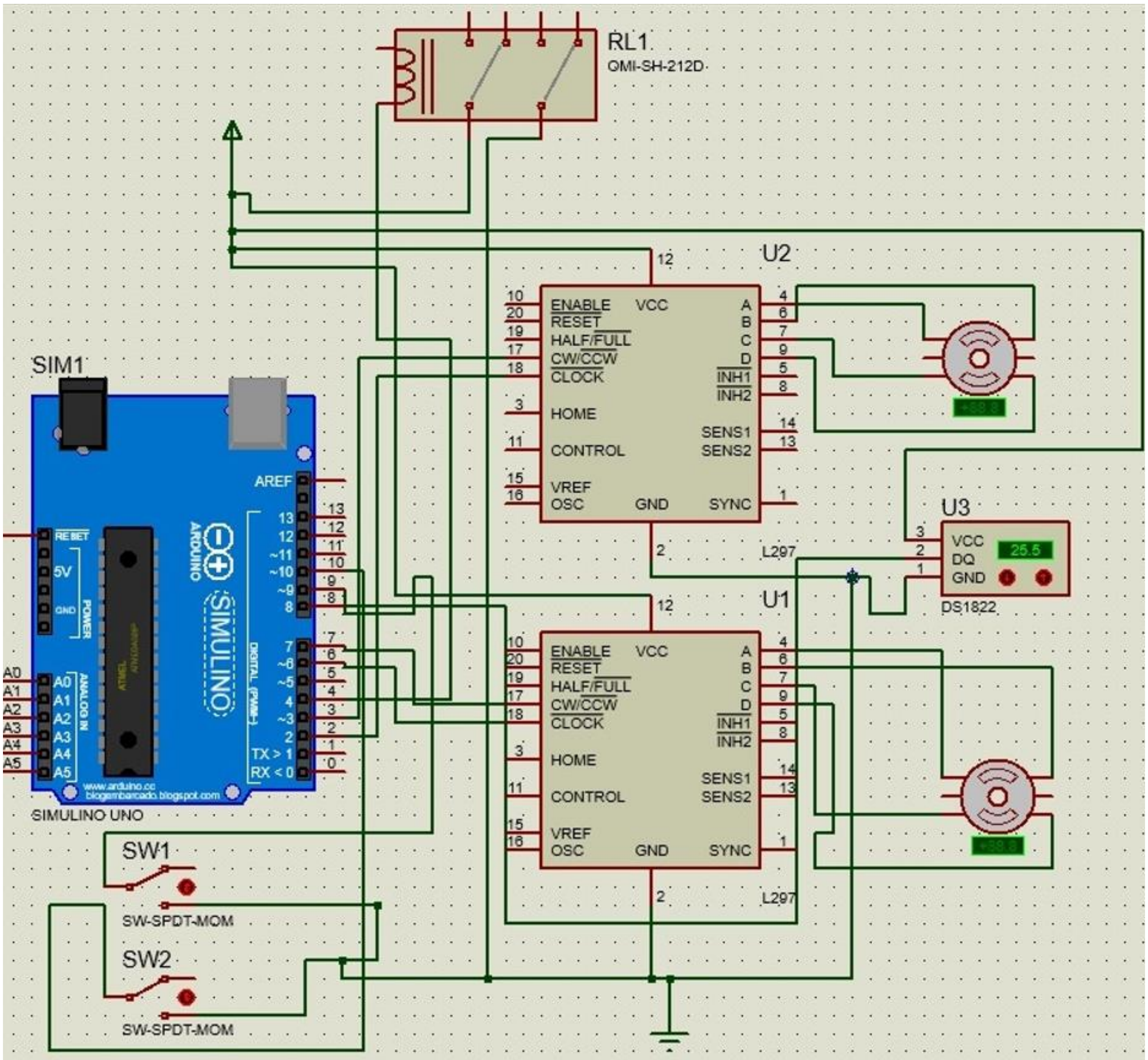
The design strategies we used provided us with a wide range of motion to add more than 4 subsystems if wanted. Unfortunately, this slows down the whole system which decreases efficiency but not using an elevator was not avoidable. Furthermore, after making the simulation we took notes for manufacturing consideration to ease the process of building such system and implementing it with the rest of the project.

[Handling - Google Drive](https://drive.google.com/drive/folders/1s8xer9g8bCx68oPaZ18lFTx-yH3PkwZm)

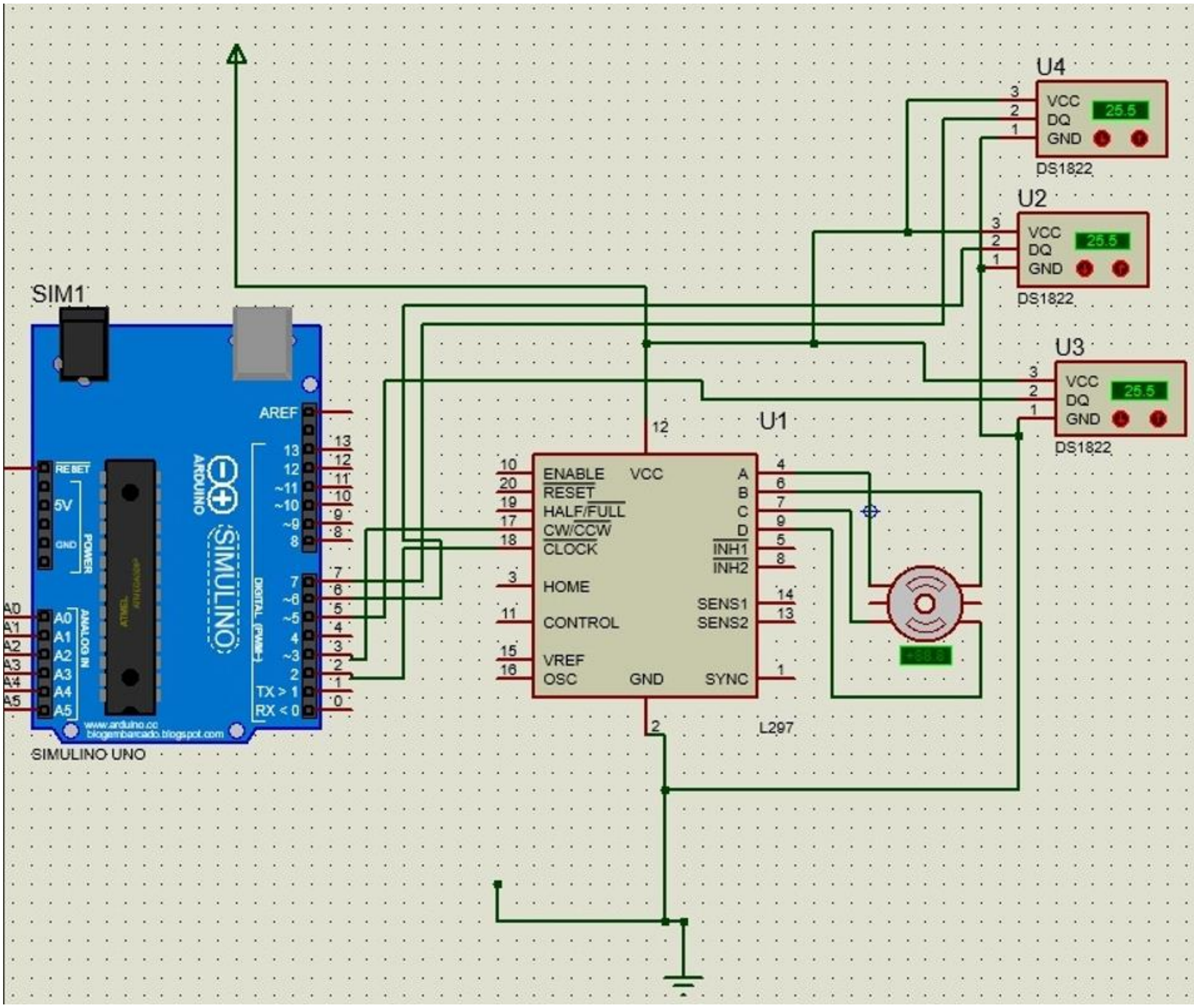
https://drive.google.com/drive/folders/1s8xer9g8bCx68oPaZ18lFTx-yH3PkwZm

Schematics

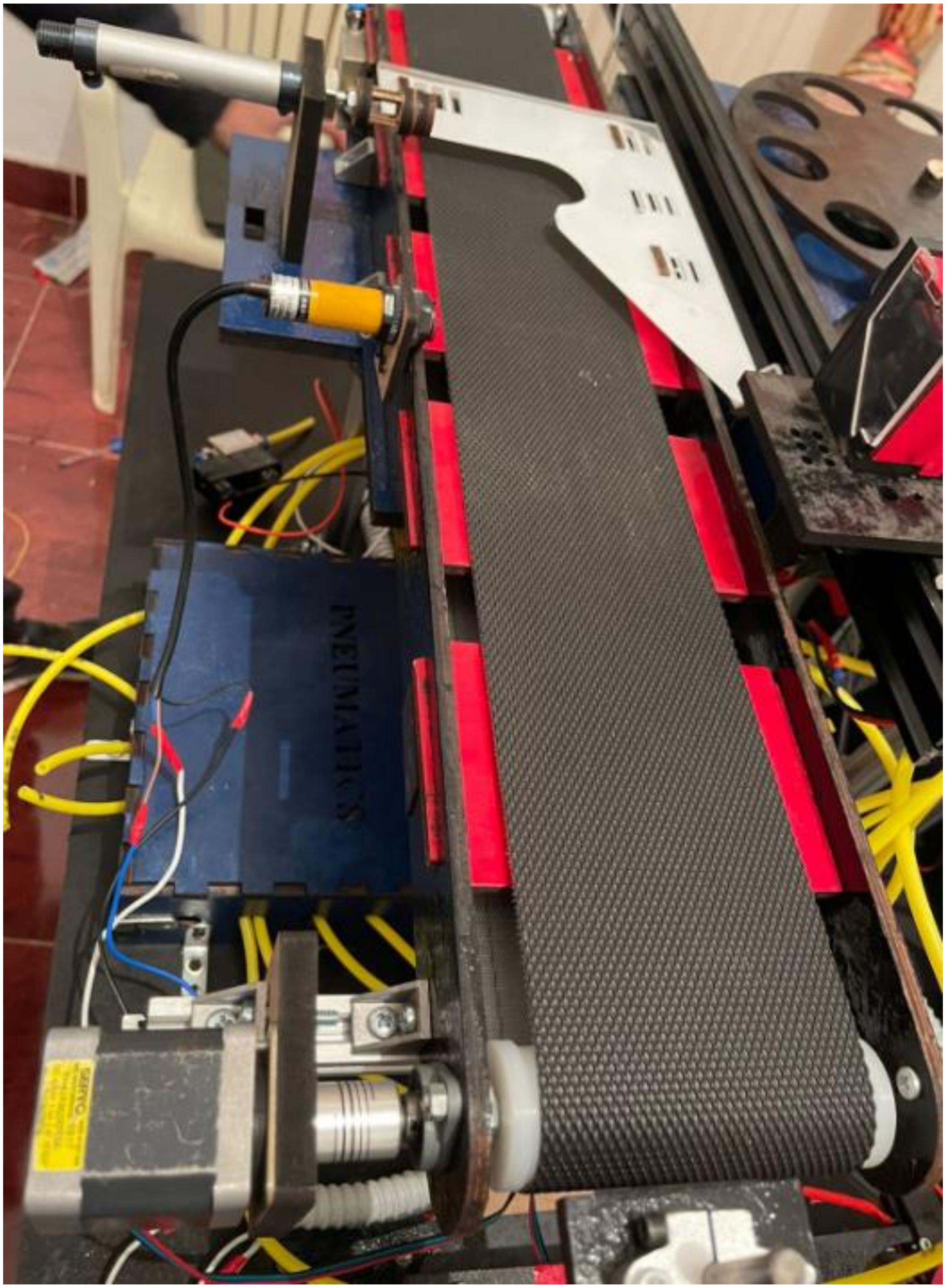
Elevator + disassembly conv.

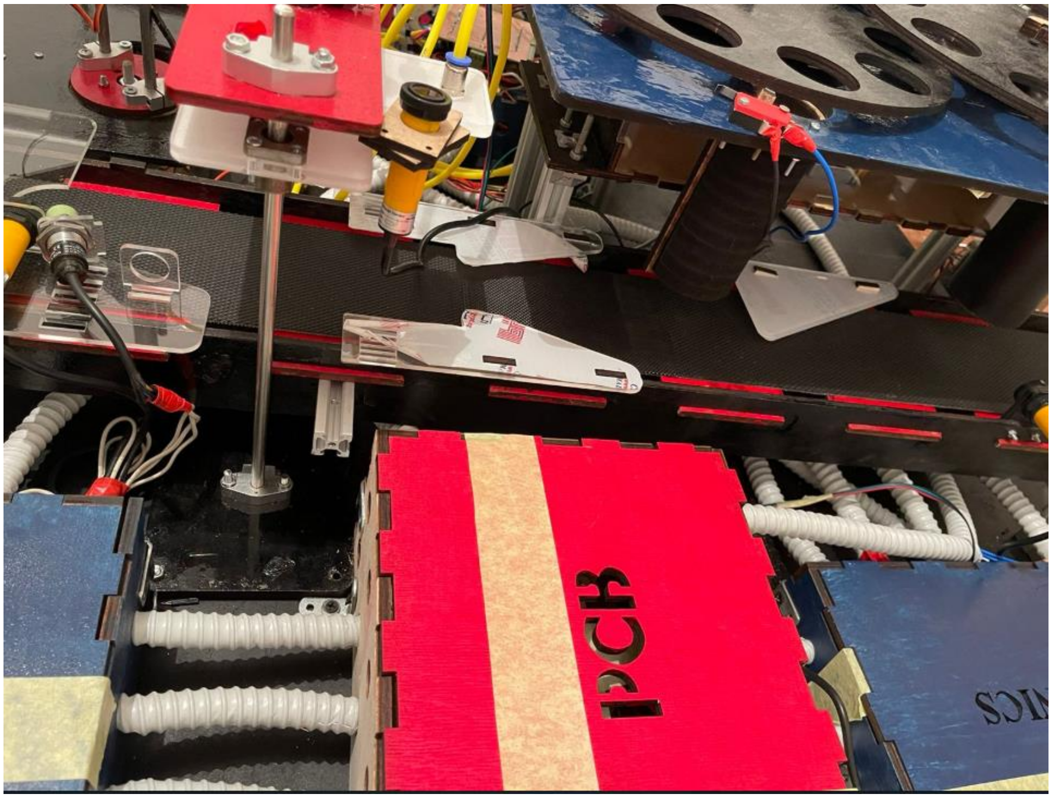


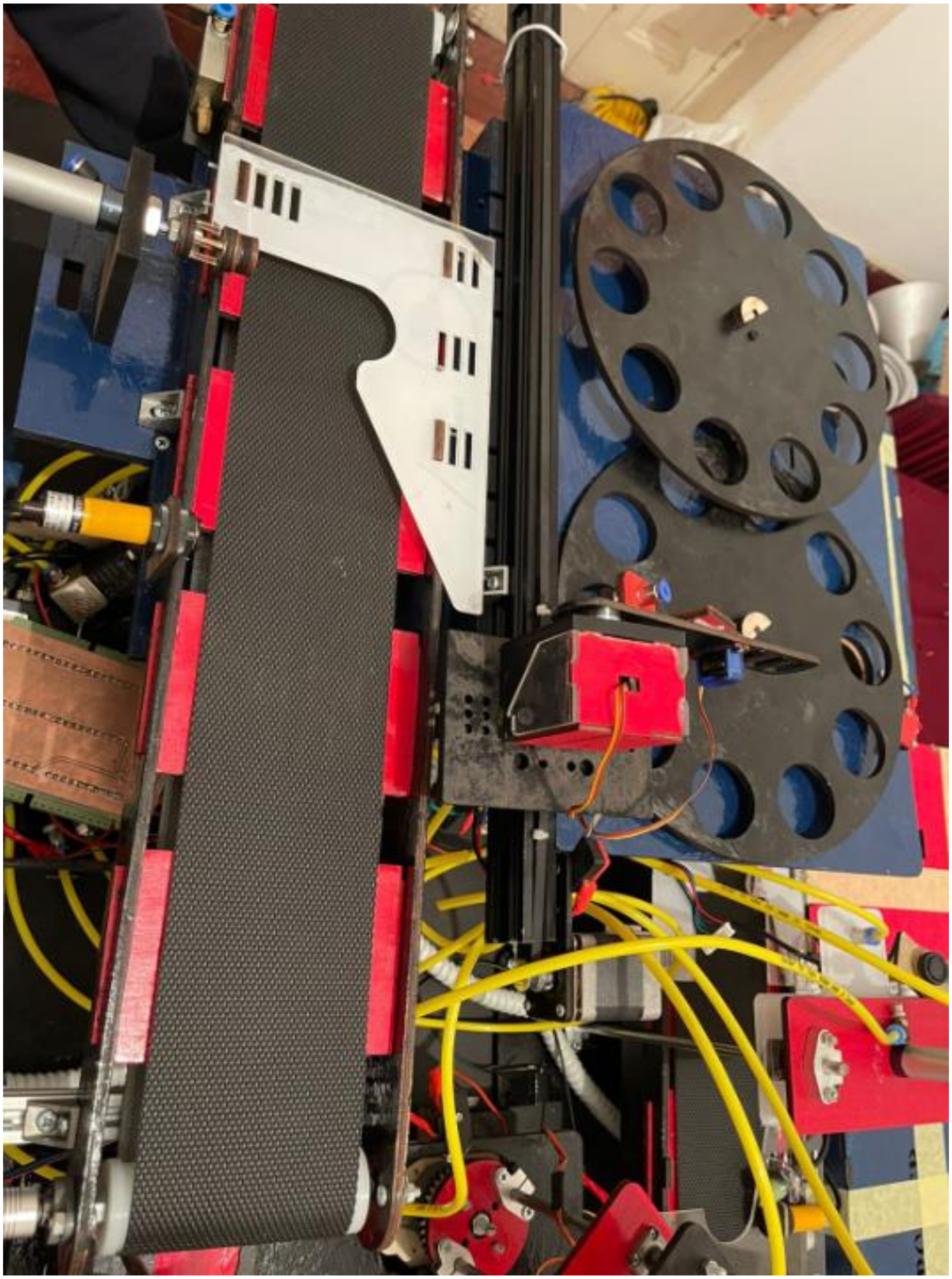
Assembly conv.

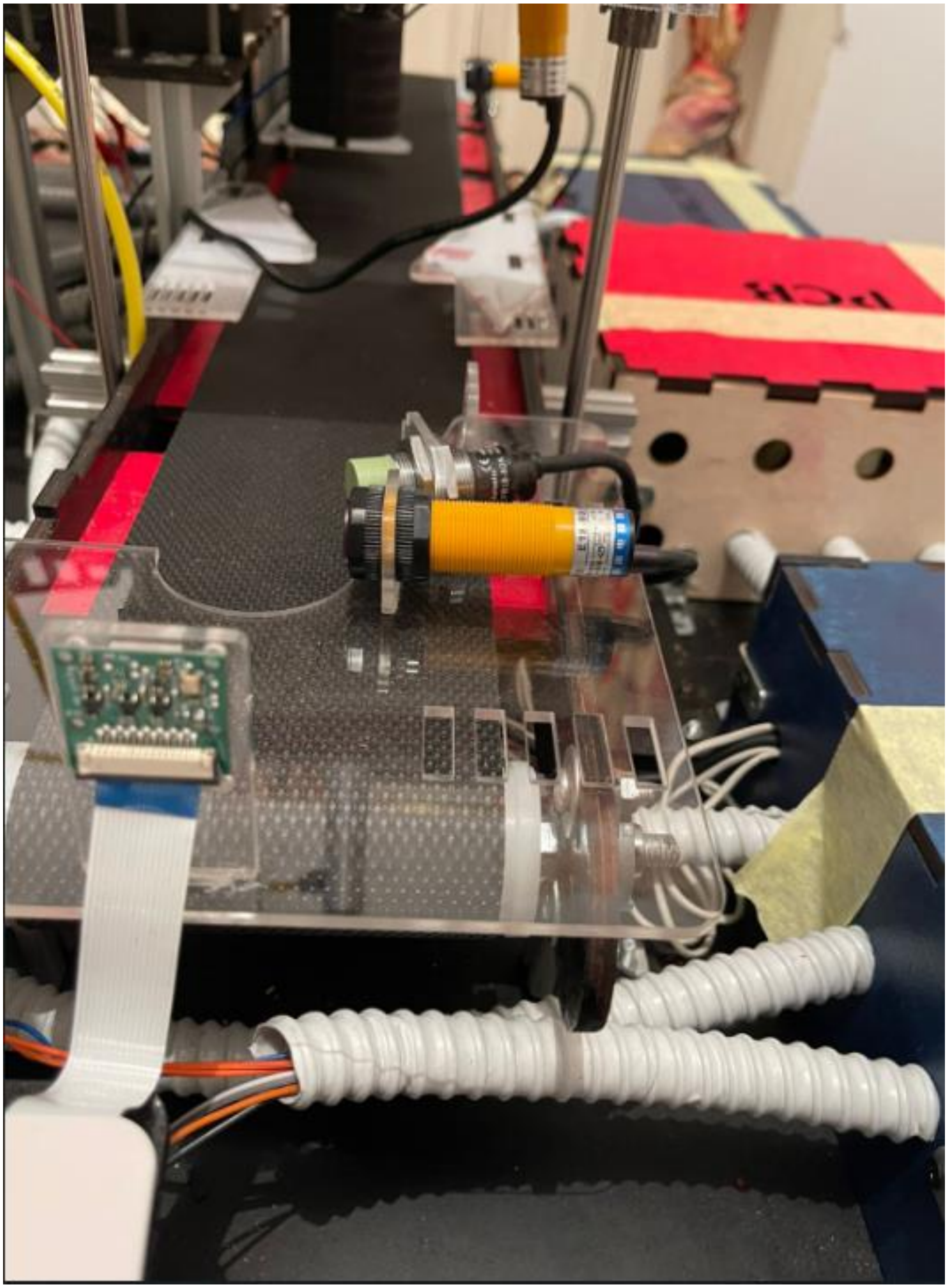


Handling System




PCB





Appendix 1

Assembly Conv.

```
#include "HCMotor.h"
#define CLK_PIN 2 //Connect to drive modules 'step' or 'CLK' input.
#define DIR_PIN 3 //Connect to drive modules 'direction' input.
#define IRfeeding 5
#define IRasmbly 6
#define IRstoring 7
#define cylinder 8
#define suction 9
bool feed = 0;
bool assup = 0;
bool ass1 = 0;
bool ass2 = 0;
bool stor = 0;

HCMotor HCMotor;

void setup()
{
  pinMode(suction, OUTPUT);
  pinMode(cylinder, OUTPUT);
  pinMode(IRfeeding, INPUT);
  pinMode(IRasmbly, INPUT);
  pinMode(IRstoring, INPUT);
  HCMotor.Init();
  HCMotor.attach(0, STEPPER, CLK_PIN, DIR_PIN);
  HCMotor.DutyCycle(0, 25);
  HCMotor.Direction(0, FORWARD);
  Serial.begin(9600);

  digitalWrite(cylinder, LOW);
  delay(1000);
  digitalWrite(suction, HIGH);
}

void loop()
{
  if (digitalRead(IRfeeding) == LOW) {
    feed = 1;
  }
```

```
while (feed == 1) {
  HCMotor.Steps(0, 1);
  if (digitalRead(IRasmbly) == LOW) {
        HCMotor.Steps(0, 250);
    feed = 0;
    ass1 = 1;

  }
}



if (ass1 == 1) {
  digitalWrite(cylinder, LOW);
  delay(1000);
  digitalWrite(cylinder, HIGH);
  delay(3000);
  digitalWrite(suction, LOW);
  delay(1000);
  digitalWrite(cylinder, LOW);
  delay(1000);
  ass1=0;
  assup = 1;

}



while (assup==1) {
  HCMotor.Steps(0, 1);
  if (digitalRead(IRasmbly) == LOW) {
      HCMotor.Steps(0, 250);
    assup = 0;
    ass2 = 1;
  }
}

if (ass2 == 1) {
  digitalWrite(cylinder, HIGH);
  delay(3000);
  digitalWrite(suction, HIGH);
  delay(1000);
  ass2 = 0;
  stor = 1;
}
```

```
  while (stor==1) {
    HCMotor.Steps(0, 1);
    if (digitalRead(IRstoring) == HIGH) {
       HCMotor.Steps(0, 200);
      stor = 0;
    }

  }

}
```

Elevator + disassembly conv.

```
#include <SpeedyStepper.h>

#define MOTOR_STEP_PIN  2
#define MOTOR_DIRECTION_PIN  3
#define MOTORCONV_STEP_PIN  6
#define MOTORCONV_DIRECTION_PIN  7

#define IRdis 8
#define LIMIT_SWITCH_PIN_TOP 9
#define LIMIT_SWITCH_PIN_HOME 10

#define Pnuematic 4

#define homingSpeedInMMPerSec  40
#define maxHomingDistanceInMM 400
#define directionTowardHomeDown  1
#define directionTowardHomeUp  -1

#define STORING_SIGNAL 12
bool intiate = 0;
bool ReachTop = 0;
bool disConv = 0;
bool END = 0;

SpeedyStepper stepper;
SpeedyStepper stepperconv;

void setup()
{
  pinMode(STORING_SIGNAL,INPUT);
```

```
  pinMode(LIMIT_SWITCH_PIN_HOME, INPUT_PULLUP);
  pinMode(LIMIT_SWITCH_PIN_TOP, INPUT_PULLUP);
  pinMode( IRdis,INPUT);
  pinMode(Pnuematic,OUTPUT);


  Serial.begin(9600);
  stepper.connectToPins(MOTOR_STEP_PIN, MOTOR_DIRECTION_PIN);

  stepperconv.connectToPins(MOTORCONV_STEP_PIN, MOTORCONV_DIRECTION_PIN);

  stepper.setStepsPerMillimeter(25 * 8);
  stepper.setSpeedInMillimetersPerSecond(1000.0);
  stepper.setAccelerationInMillimetersPerSecondPerSecond(1000.0);


  stepperconv.setStepsPerMillimeter((200 / (2 * 3.14 * 30)) * 8);
  stepperconv.setSpeedInMillimetersPerSecond(2.0);
  stepperconv.setAccelerationInMillimetersPerSecondPerSecond(50.0);


  if (stepper.moveToHomeInMillimeters(directionTowardHomeDown, homingSpeedInMMPerSec,
maxHomingDistanceInMM, LIMIT_SWITCH_PIN_HOME) != true) {}
digitalWrite(Pnuematic,LOW);
intiate=1;
}



void loop()
{

if(digitalRead(STORING_SIGNAL)==HIGH){
  intiate=1;
}

  if (intiate == 1) {
    if (stepper.moveToHomeInMillimeters(directionTowardHomeUp, homingSpeedInMMPerSec,
maxHomingDistanceInMM, LIMIT_SWITCH_PIN_TOP) != true) {}
    intiate = 0;
    ReachTop = 1;
    Serial.println("Phase one");
  }
  if (ReachTop == 1) {
    //stepper.moveRelativeInSteps(-200);
```

```
    //pneumatic extend and retract
    delay(1000);
    digitalWrite(Pnuematic,HIGH);
    delay(3000);
    digitalWrite(Pnuematic,LOW);
    delay(1000);
    ReachTop = 0;
    disConv = 1;
    END = 1;

     Serial.println("Phase 2");

  }

  if (disConv == 1) {
    if (stepperconv.moveToHomeInMillimeters(1, homingSpeedInMMPerSec, 10000, IRdis) != true) {}
    stepperconv.setSpeedInStepsPerSecond(100);
    stepperconv.setAccelerationInStepsPerSecondPerSecond(100);
    stepperconv.moveRelativeInSteps(100);
    disConv = 0;
    Serial.println("Phase 3");
  }

  if (END == 1) {
    if (stepper.moveToHomeInMillimeters(directionTowardHomeDown, homingSpeedInMMPerSec,
maxHomingDistanceInMM, LIMIT_SWITCH_PIN_HOME ) != true) {}
    END = 0;
    Serial.println("Phase 4");
  }
}
```