MCT 381 Design of Mechatronics Systems 1

Final term project

Disassembly Module

Submitted By:

Name : Mohamed Yaser Mohamed

ID: 17P8213

Submitted to:

Dr.Shady Maged

Dr. Omar Shehata

Contents

Abstract

This report is a sub report which discusses the step by step procedure of creating the sorting and storage mechanisms used in a fully automated production line which consists of 5 modules.The assembly,disassembly,handling,sorting and storage and finally,feeding module.The production line starts with two parts fed on a conveyor and then assembled to be one product then stored in a shelf for a specific time then fed on another conveyor to be disassembled back then stored again in a magazine.Every detail of the disassembly module will be explained thoroughly starting from the basic idea to the mechanical CAD design moving to the electronic design as well as the software design.In addition,motor sizing and a description to the different mechanical fixations and considerations made for this module to do its function accurately and efficiently.

1.0 Introduction

The sorting and storage stations is considered as the bridge that connects the two halves of the conveyors with each other.as it takes the assembled product from one half of the project (Conveyor 1) and then store it within the shelves for a certain time ,then it descends to ground level (Conveyor 2) and then hands the product to it. In this report ,we will be discussing in detail the step by step manufacturing and implementation of the station including materials used, components used and the integration all the mechanical ,electrical and software blocks together.

2.0 System Requirements

The requirements of the sorting and storage station during the design phase were to:

build a system that is mechanically easy and fast to assemble

In addition, to be electrically/electronically organized so as to be able to diagnose any errors quickly.

Finally, to have a closed loop system that is fully automated.

3.0 Project Plan

Weeks 1-2 Brainstorming ideas for the system and learning the basics to build an accurate model and CAD design

Weeks 3-6 CAD Design as well as modeling and simulation of the system

Weeks 8-9 Listing and purchasing the necessary components for the system

Weeks 10-11 Mechanical assembly and setting up the electrical and pneumatic circuits

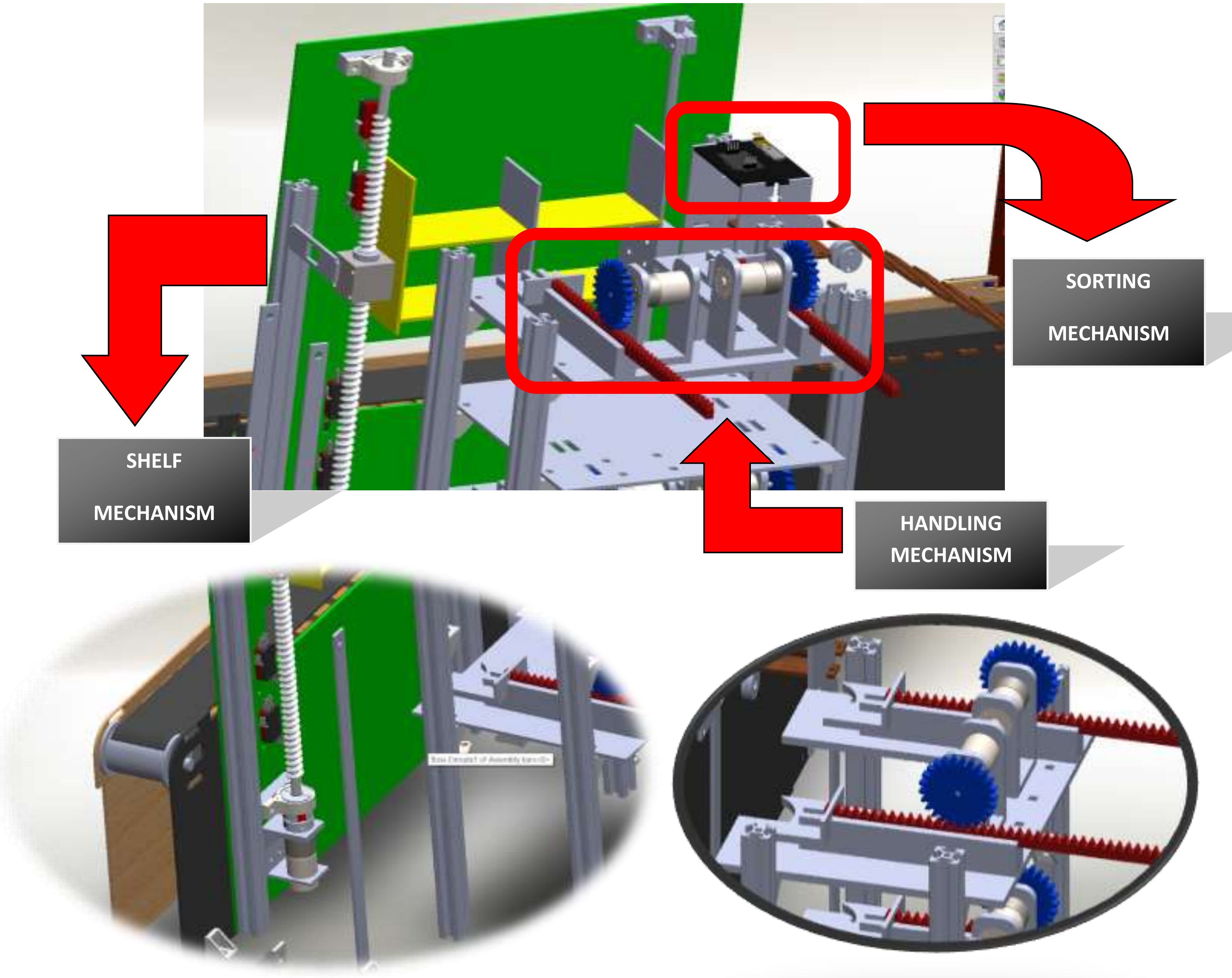
Weeks 12-13 Integrating the electrical,mechanical and pneumatic systems together by building a software program run by an ARDUINO UNO

Weeks 14-15 Testing the subsystem individually and integrating it with the other subsystems (Handling)

4.0 Mechanical Design

The mechanism is separated in to three different parts yet associated and dependable on each other which are:

1. SORTING PART
2. STORAGE PART
3. HANDLING PART



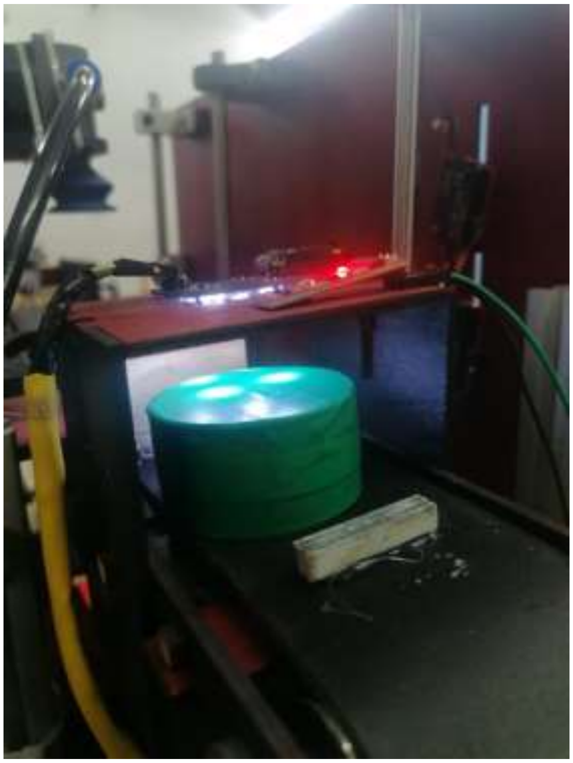
4.1 SORTING PART

This part is mainly electrical from my point of view as it's mechanical part is mostly executed by the conveyor .

I decided to go for a color sensor to differentiate between the parts (GREEN & RED) so I mounted it on variables:

1. saves the color detected in a variable called color
2. checks if floor no.1 in the shelf is full or not

   then the parts goes for the assembly module to get assembled, then when it's done the conveyor runs upward or backward according to the color saved before until it reaches the Rack and Pinion (HANDLING MECHANISM) ,it stops in front of it using the distance sensor in conveyor 1 ,then conveyor 1 sends a signal to the shelf controller with the variable saved before and accordingly it decides which rack and pinion should work and whether the first floor is full or not so that the shelf ascends to put the part in the second floor.



4.1.2 Components Used:

Mechanical System Components

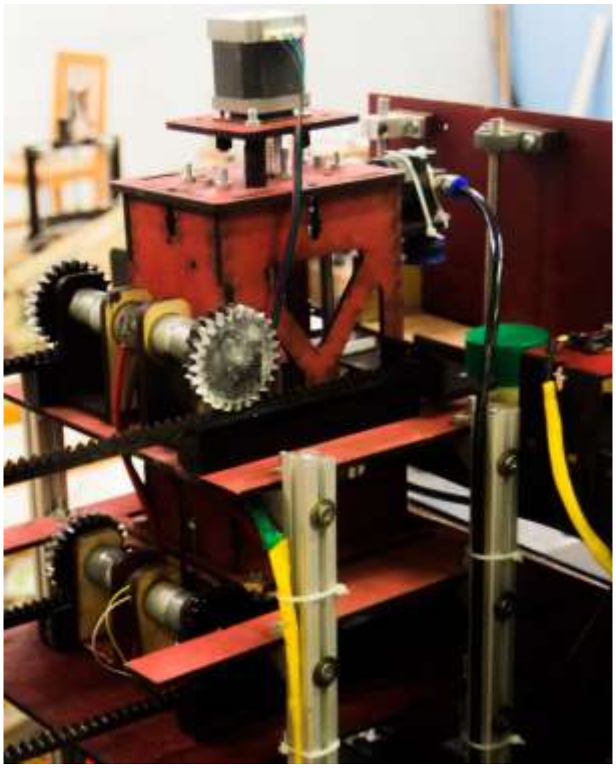
This part is Mainly executed mechanically with the help of the conveyor.

Electrical System Components

1 color sensor

1 distance sensor

4.2 HANDLING MECHANISM



It's mainly consisted of a laser cut acrylic rack and pinion with a head made of a 3d printed plastic.

It works when it gets signal from the conveyor that the parts are ready for storing.

4.2.1 Components Used:

Mechanical System Components

4 laser cut acrylic rack and pinion

4 3d printed heads

Electrical System Components

4 Limit switch

4 DC motor driver

4 DC motors 50 rpm

1 Power supply

Cables/jumpers

1 Bread Board

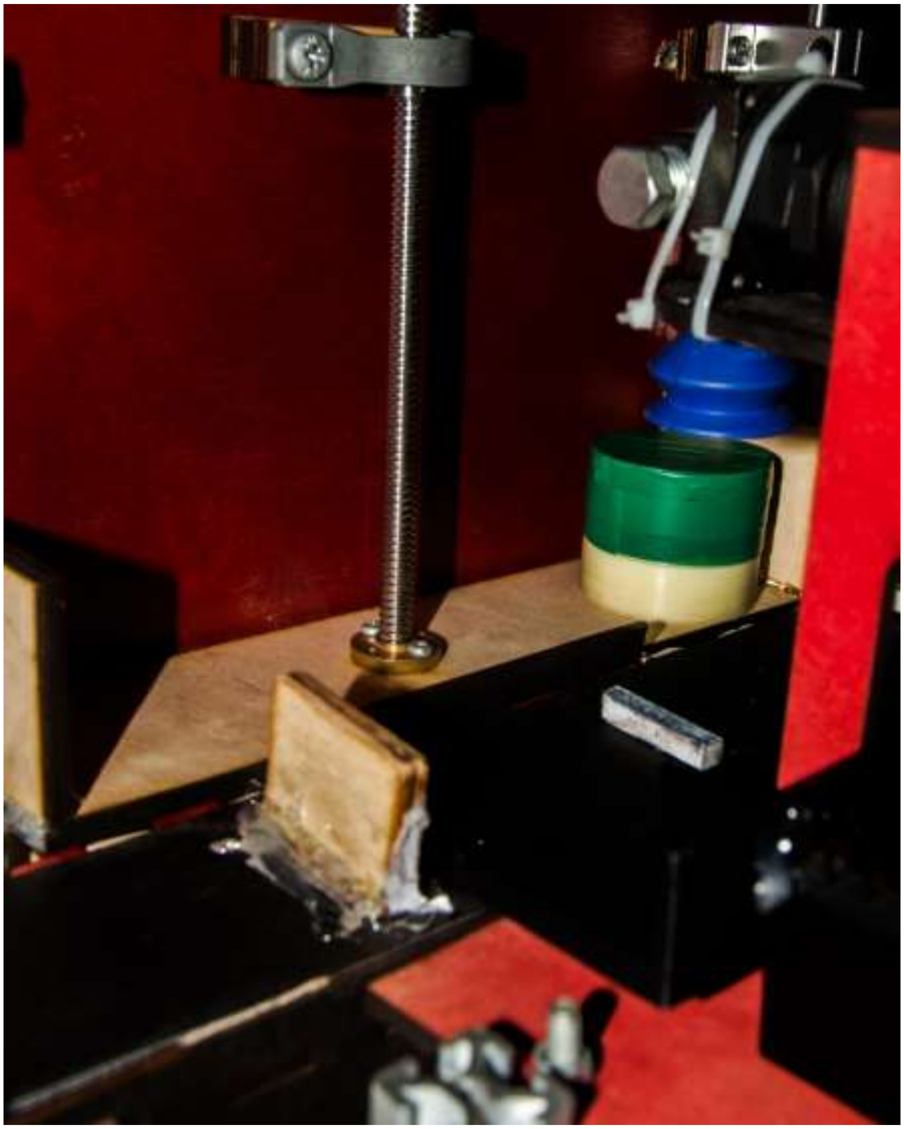
4.3 SHELF MECHANISM

It consists of two parts the fixed part and the movable part.

The fixed part is big rectangle which hold all the components of the shelf and has a hole which lead to the second conveyor.it holds the motor the power screw and the supporting rod all fixed using vertical supports.

The movable part was laser cut and assembled as two floor shelf with a guiding nut on one side and linear bearing on the other.

4.3.1 Components Used:



Mechanical System Components

1 Lead Screw

1 Linear Bearings

4 Vertical Support

28 Bolts & Nuts

Washers

2 Supporting rod

4 20x20 square cross section supports

1 5 mm - 8 mm coupler

Electrical System Components

4 Limit switch

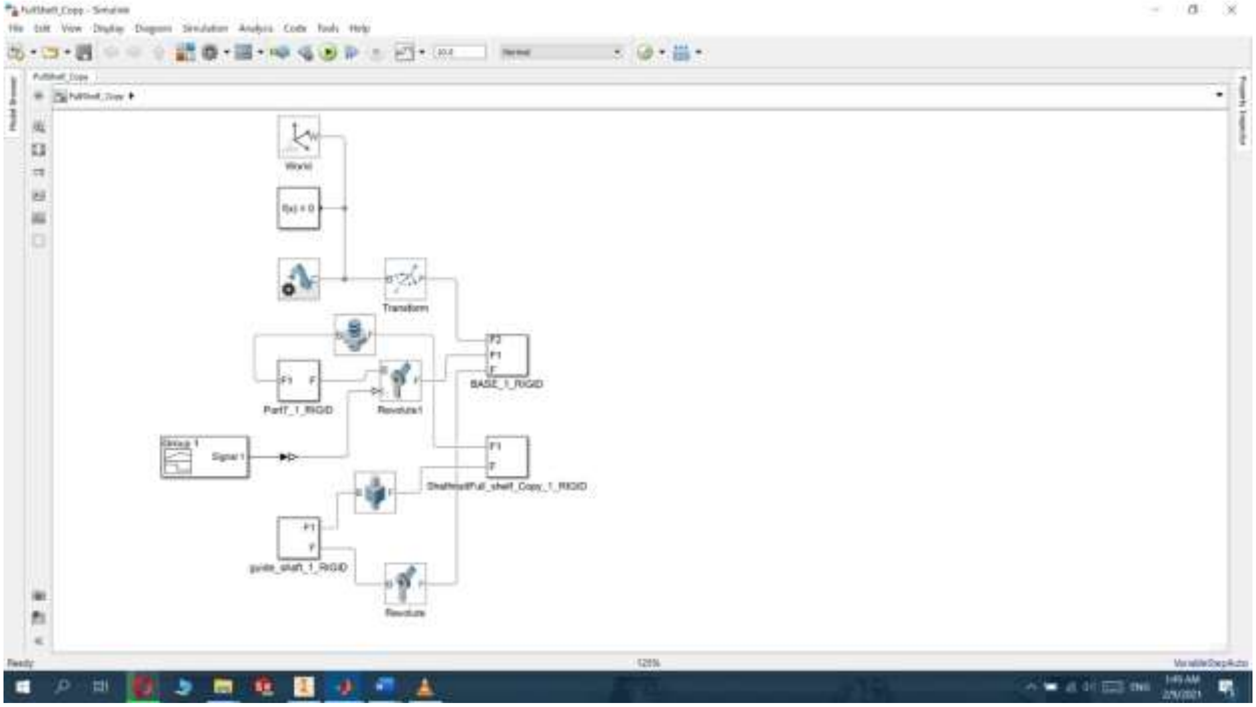
1 DC motor driver

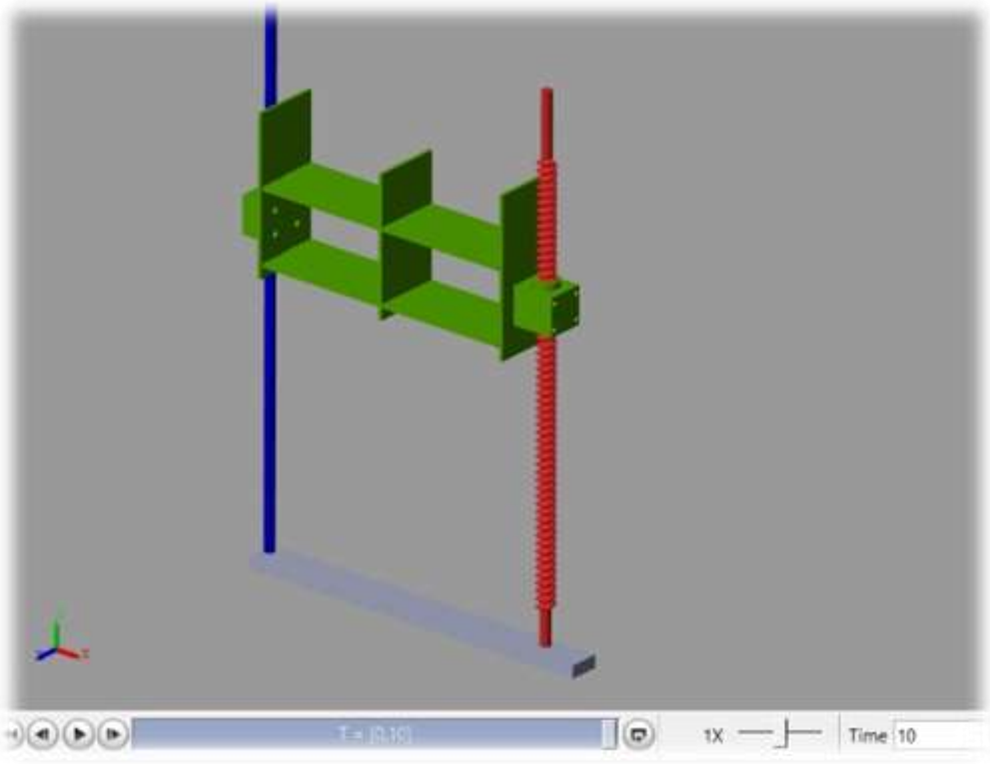
1 DC motor 250 rpm

1 Power supply

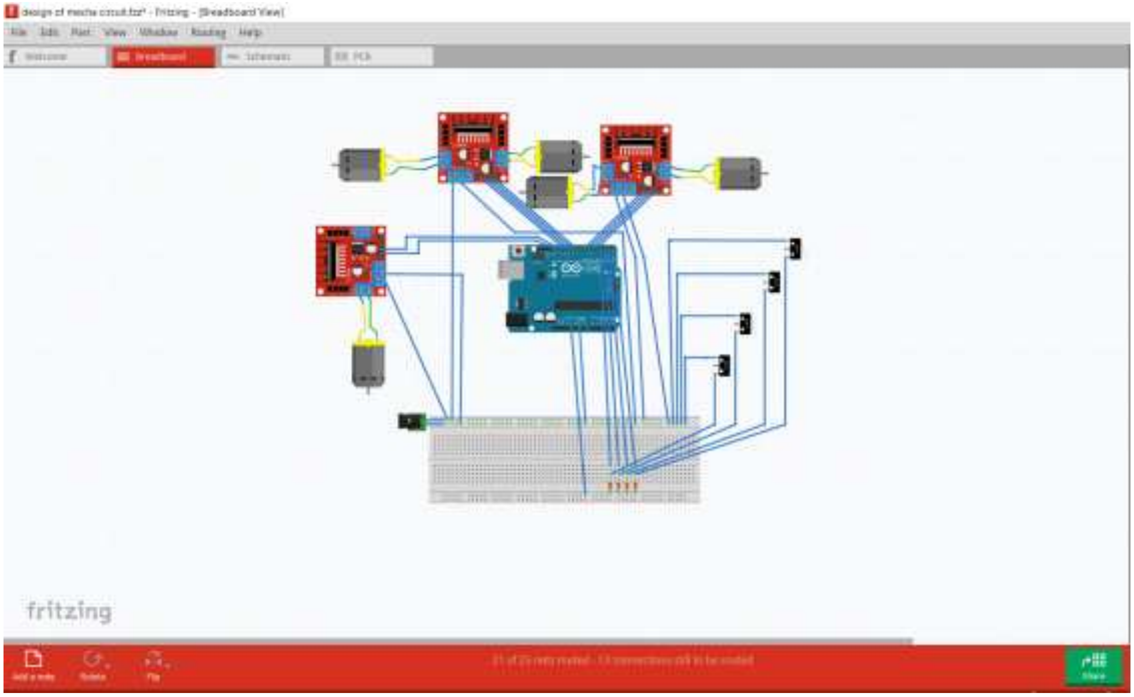
Cables/jumpers

4.3.2 MATLAB MECHANICAL MODEL FOR SHELF:





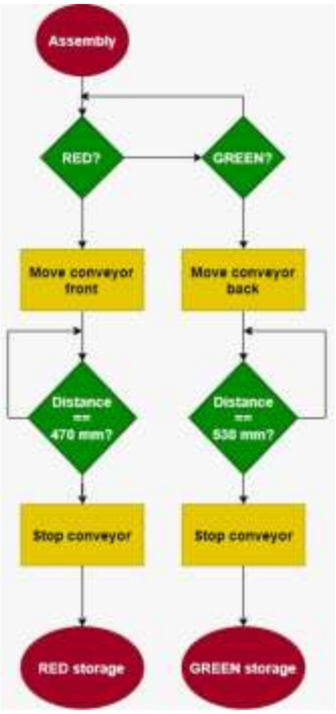
5.0 Electrical Design



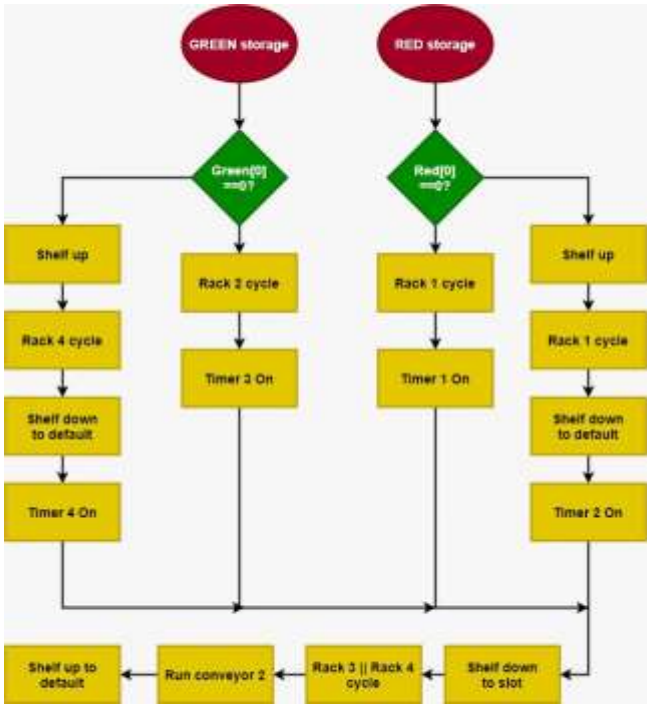
21 of 25 nets routed - 13 connections still to be routed

6.0 Software flowchart

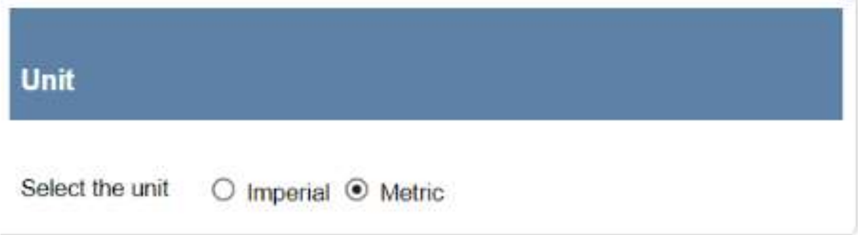
FIRST STAGE:



SECOND STAGE:

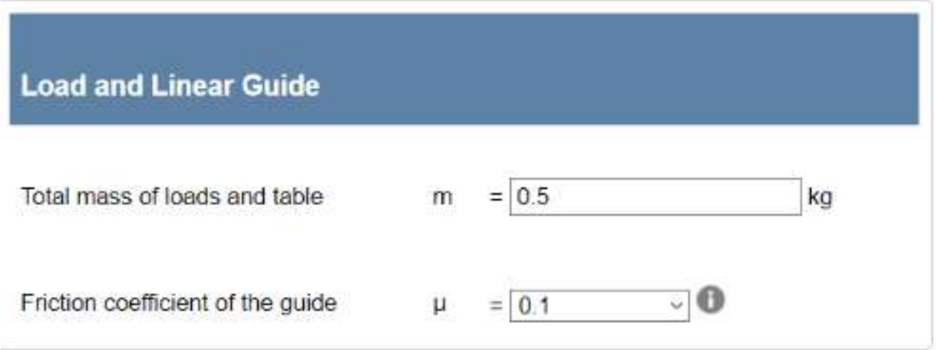


7.0 MOTOR SIZING



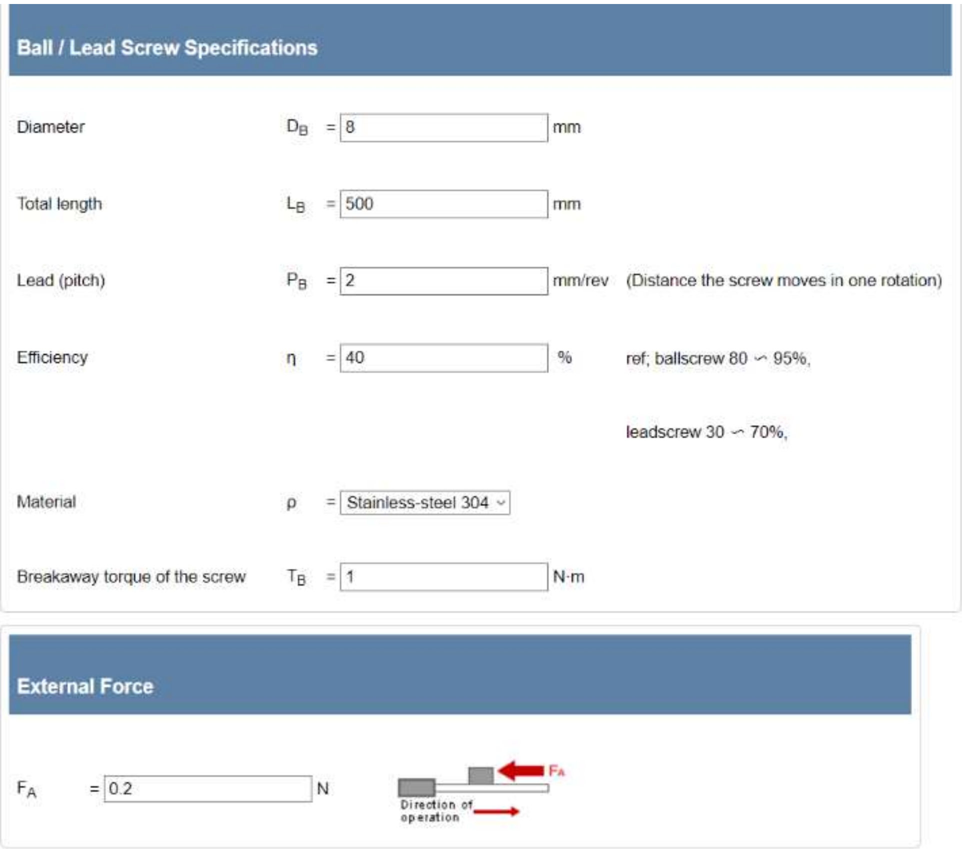
Unit

Select the unit ○ Imperial ● Metric



Load and Linear Guide

Total mass of loads and table	m	= 0.5	kg
Friction coefficient of the guide	μ	= 0.1	



Ball / Lead Screw Specifications

Diameter D_B = 8 mm

Total length L_B = 500 mm

Lead (pitch) P_B = 2 mm/rev (Distance the screw moves in one rotation)

Efficiency η = 40 % ref; ballscrew 80 ∽ 95%,

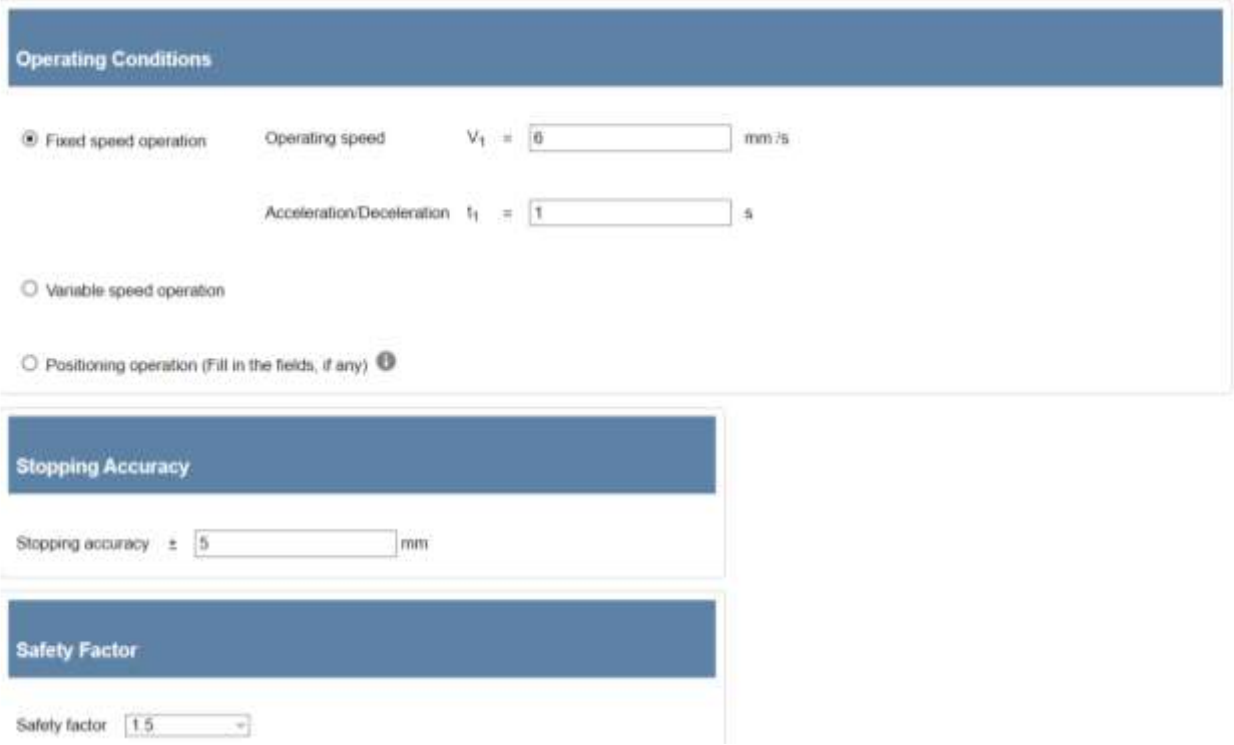
leadscrew 30 ∽ 70%,

Material ρ = Stainless-steel 304

Breakaway torque of the screw T_B = 1 N·m

External Force

F_A = 0.2 N



Operating Conditions

● Fixed speed operation Operating speed V_1 = 6 mm /s

Acceleration/Deceleration t_1 = 1 s

○ Variable speed operation

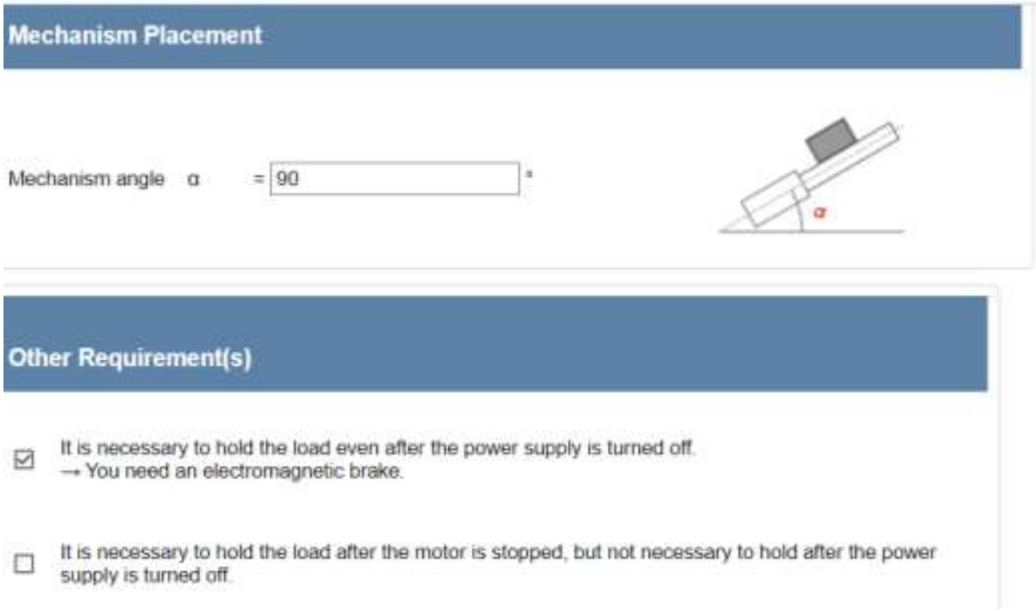
○ Positioning operation (Fill in the fields, if any)

Stopping Accuracy

Stopping accuracy ± 5 mm

Safety Factor

Safety factor 1.5



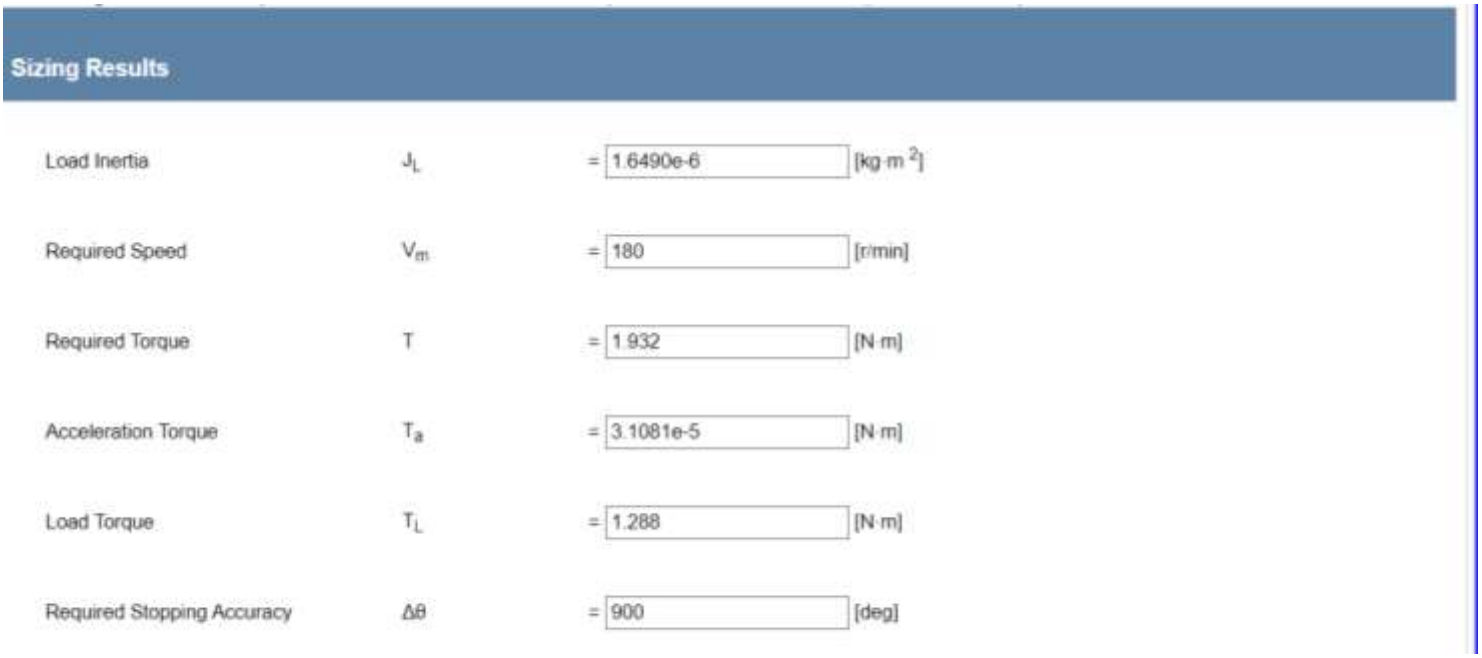
Mechanism Placement

Mechanism angle α = 90 °

Other Requirement(s)

☑ It is necessary to hold the load even after the power supply is turned off.
→ You need an electromagnetic brake.

□ It is necessary to hold the load after the motor is stopped, but not necessary to hold after the power supply is turned off.



Sizing Results

Load Inertia	J_L	= 1.6490e-6	[kg·m ^{2}]
Required Speed	V_m	= 180	[r/min]
Required Torque	T	= 1.932	[N·m]
Acceleration Torque	T_a	= 3.1081e-5	[N·m]
Load Torque	T_L	= 1.288	[N·m]
Required Stopping Accuracy	$\Delta\theta$	= 900	[deg]

8.0 Manufacturing Process

The manufacturing process in the sorting module included the usage of the laser printing technique to print everything from the walls that make up the shelf floor and the shelf back to the rack and pinions .MDF wood 6 mm and acrylic 6mm were used to assemble the station.

We also used 3d printing in making the piston heads and the dc motor holder.

After all the components were printed, the assembly process began.

8.1 Assembly of the module description

SHELF ASSEMBLY

First the wall of the shelf was fixed using the steel sections,bolts and nuts in the base of the project. With that step done it made a good base for me to fix the rest of the components including the flanged bearings for the guidways and the power screw .

The walls of the shelf was assembled alone then fixed with power screw in the wall.a coupling was placed at the end of the power screw in which at the second end of it the dc motor-the main actuator- will be fixed.

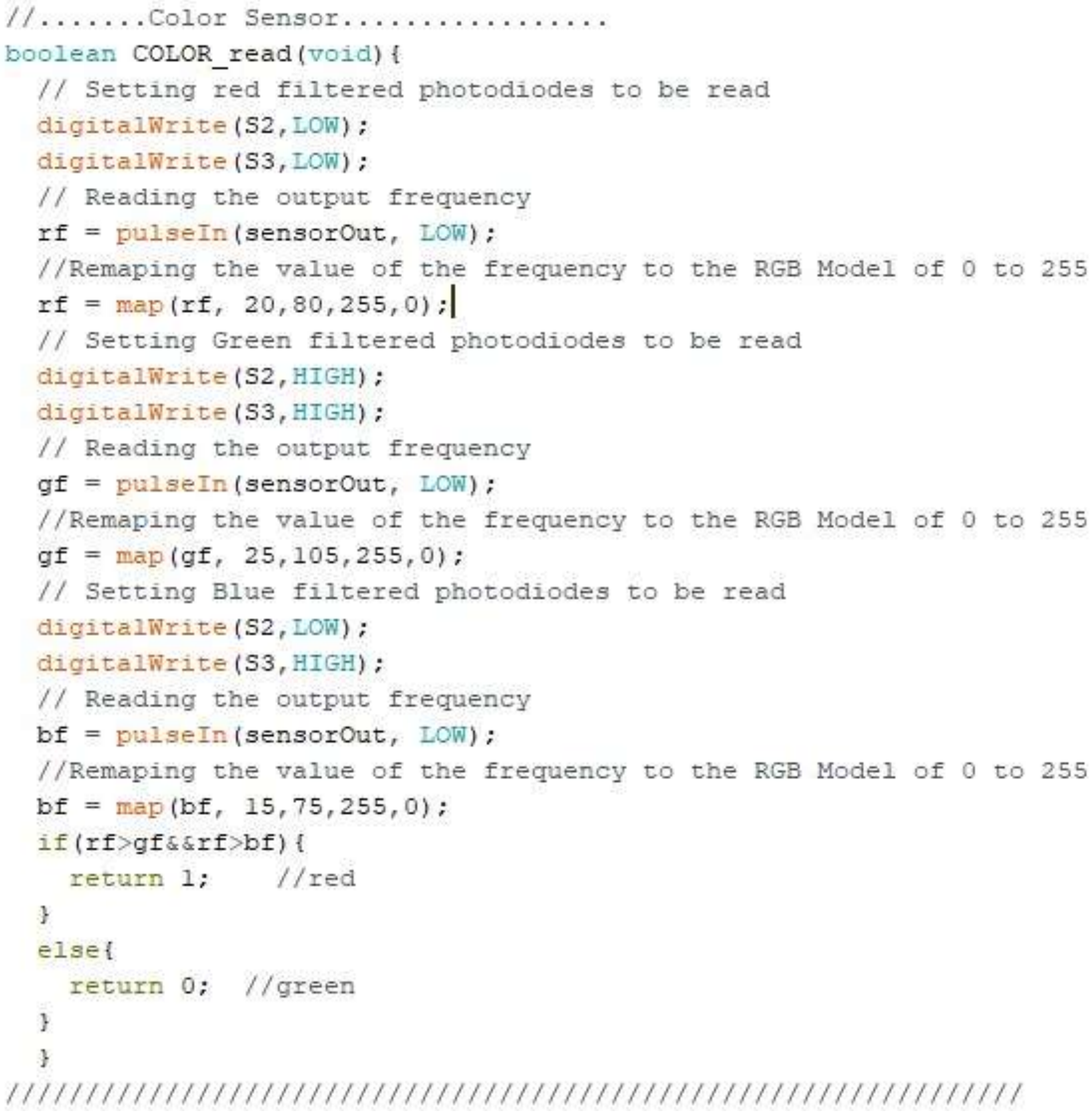
Finally the limit switches was fixed using two bolts and nuts.

RACK AND PINIONS ASSEMBLY

First 4 steel sections are fixed to the base then the tables were oriented on the required levels for the rack and pinions to operate. The rack and pinions guideways was fixed using super glue. The motors were placed in their pre-made holes in the tables.

9.0 SOFTWARE FLOW

First the color sensor senses the color of the part and saves it in a variable called color.This is done by calibration of the sensor and mapping the results. all is done in this function.



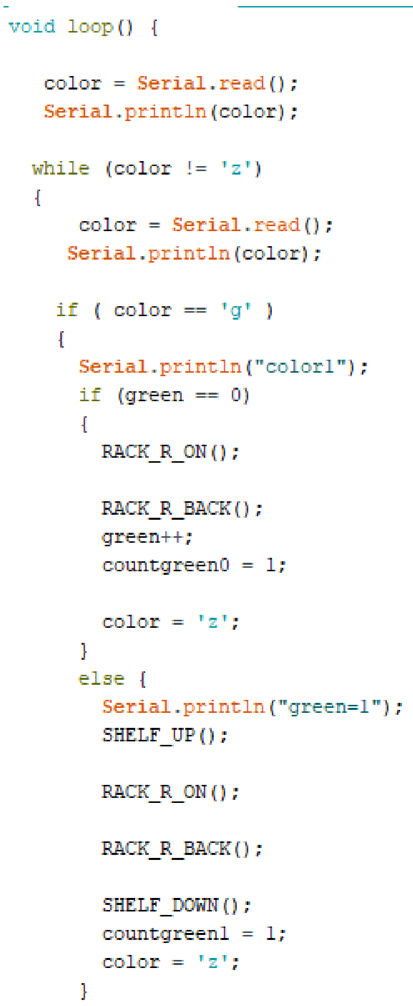
```
//.......Color Sensor................
boolean COLOR_read(void){
  // Setting red filtered photodiodes to be read
  digitalWrite(S2,LOW);
  digitalWrite(S3,LOW);
  // Reading the output frequency
  rf = pulseIn(sensorOut, LOW);
  //Remaping the value of the frequency to the RGB Model of 0 to 255
  rf = map(rf, 20,80,255,0);
  // Setting Green filtered photodiodes to be read
  digitalWrite(S2,HIGH);
  digitalWrite(S3,HIGH);
  // Reading the output frequency
  gf = pulseIn(sensorOut, LOW);
  //Remaping the value of the frequency to the RGB Model of 0 to 255
  gf = map(gf, 25,105,255,0);
  // Setting Blue filtered photodiodes to be read
  digitalWrite(S2,LOW);
  digitalWrite(S3,HIGH);
  // Reading the output frequency
  bf = pulseIn(sensorOut, LOW);
  //Remaping the value of the frequency to the RGB Model of 0 to 255
  bf = map(bf, 15,75,255,0);
  if(rf>gf&&rf>bf){
    return 1;    //red
  }
  else{
    return 0;  //green
  }
  }
////////////////////////////////////////////////////////////////
```

Second stage happens after the assembly finishes it's work. The conveyor checks the color saved from the previous station and moves forward or backward according to the color of the product. It keeps moving until the distance sensor stops it by pre-determined distance to stop exactly in front of the rack & pinion.

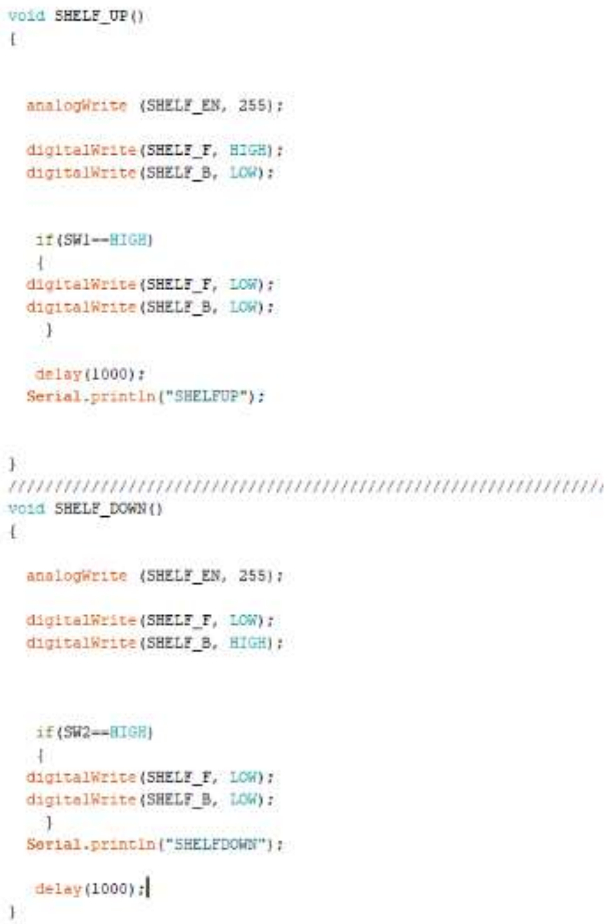
Third stage the microcontroller checks if the first floor of the shelf is full .

if No it moves the rack and pinion to insert the product in the first floor and opens the timer related to this product and raises a flag means that the first floor is full.

If yes the shelf moves upward until it reaches the limit switch which says that the second floor in front of conveyor 1 .the rack and pinion moves to insert the product in the second floor then the shelf returns to it's default position and enables the timer interrupt .



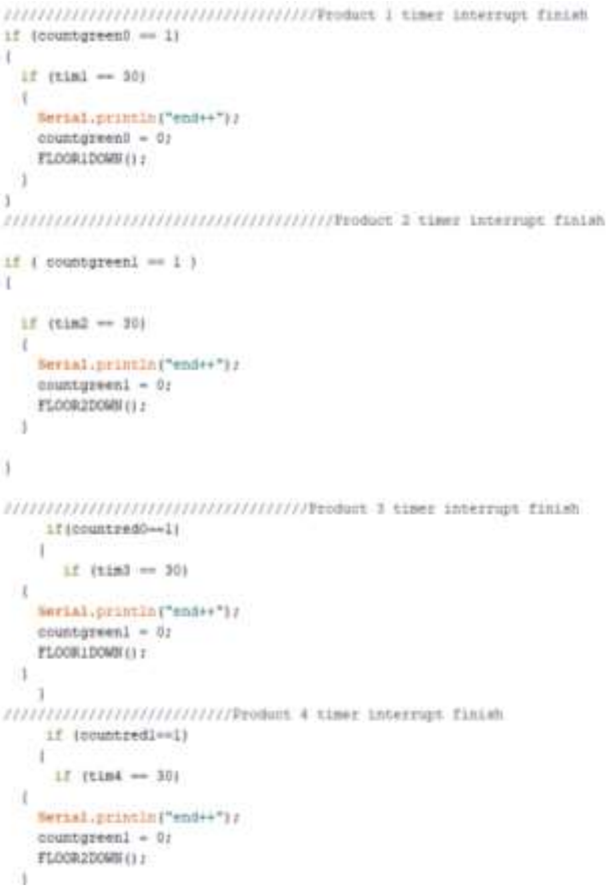
```
void loop() {

   color = Serial.read();
   Serial.println(color);

  while (color != 'z')
  {
      color = Serial.read();
     Serial.println(color);

    if ( color == 'g' )
    {
      Serial.println("color1");
      if (green == 0)
      {
        RACK_R_ON();

        RACK_R_BACK();
        green++;
        countgreen0 = 1;

        color = 'z';
      }
      else {
        Serial.println("green=1");
        SHELF_UP();

        RACK_R_ON();

        RACK_R_BACK();

        SHELF_DOWN();
        countgreen1 = 1;
        color = 'z';
      }
```



```
void SHELF_UP()
{


  analogWrite (SHELF_EN, 255);

  digitalWrite(SHELF_F, HIGH);
  digitalWrite(SHELF_B, LOW);


   if(SW1==HIGH)
   {
  digitalWrite(SHELF_F, LOW);
  digitalWrite(SHELF_B, LOW);
    }

   delay(1000);
  Serial.println("SHELFUP");


}
///////////////////////////////////////////////////////////////////////
void SHELF_DOWN()
{

  analogWrite (SHELF_EN, 255);

  digitalWrite(SHELF_F, LOW);
  digitalWrite(SHELF_B, HIGH);



   if(SW2==HIGH)
   {
  digitalWrite(SHELF_F, LOW);
  digitalWrite(SHELF_B, LOW);
    }
  Serial.println("SHELFDOWN");

   delay(1000);
}
```

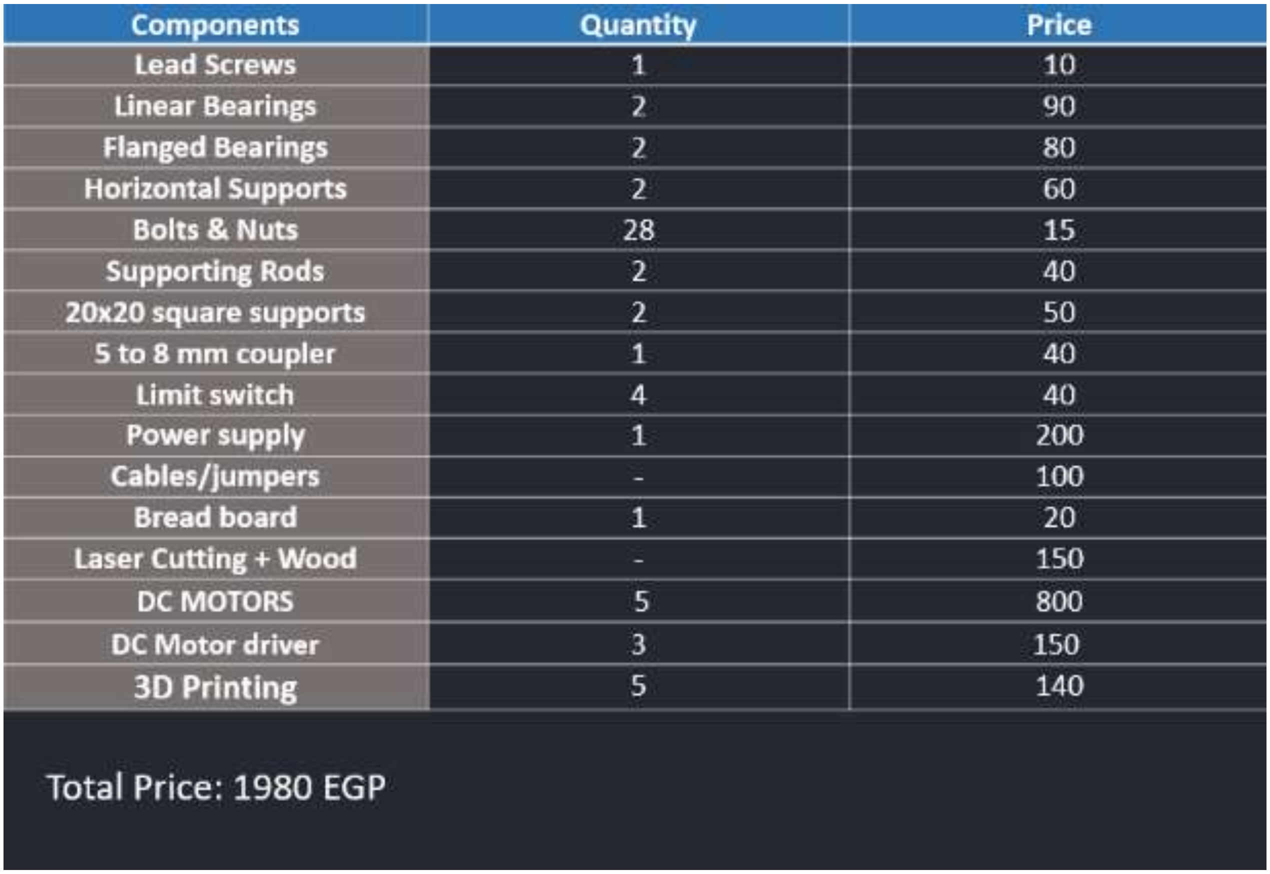
Forth stage the timer interrupt counts to 30sec then it automatically descends down to conveyor 2 and the lower rack and pinion operates according to which product is meant to exit.

```
ISR(TIMER1_COMPA_vect)

{
  //interrupt commands here
  //according to the part stored a Timer is set specifically for it
  //Every Tick perfomed by the controller it adds 1 to the count variable

  if ( countgreen0 == 1 )
  {
    tim1++;
    if (tim1 == 30)
    {

      TIMSK1 &= ~(1 << OCIE1A);
      TCNT1  = 0;//initialize counter value to 0
      cli(); //stop interrupts
      Serial.println("Tim1OUTTTTTTTTTTTTTTTTTT ");


    }
    Serial.println("Tim1 = ");
    Serial.print(tim1);
  }


  if ( countgreen1 == 1 )
  {
    tim2++;
    if (tim2 == 30)
    {

      TIMSK1 &= ~(1 << OCIE1A);
      TCNT1  = 0;//initialize counter value to 0
      cli(); //stop interrupts
      Serial.println("Tim1OUTTTTTTTTTTTTTTTTTT ");


    }
    Serial.println("Tim2 = ");
    Serial.print(tim2);
  }
```



```
////////////////////////////////////////Product 1 timer interrupt finish
if (countgreen0 == 1)
{
  if (tim1 == 30)
  {
    Serial.println("end++");
    countgreen0 = 0;
    FLOOR1DOWN();
  }
}
//////////////////////////////////////////Product 2 timer interrupt finish

if ( countgreen1 == 1 )
{

  if (tim2 == 30)
  {
    Serial.println("end++");
    countgreen1 = 0;
    FLOOR2DOWN();
  }


}

//////////////////////////////////////Product 3 timer interrupt finish
    if(countred0==1)
    {
      if (tim3 == 30)
  {
    Serial.println("end++");
    countgreen1 = 0;
    FLOOR1DOWN();
  }
    }
/////////////////////////////Product 4 timer interrupt finish
    if (countred1==1)
    {
     if (tim4 == 30)
  {
    Serial.println("end++");
    countgreen1 = 0;
    FLOOR2DOWN();
  }
```

10.0 Bills of materials used



Components	Quantity	Price
Lead Screws	1	10
Linear Bearings	2	90
Flanged Bearings	2	80
Horizontal Supports	2	60
Bolts & Nuts	28	15
Supporting Rods	2	40
20x20 square supports	2	50
5 to 8 mm coupler	1	40
Limit switch	4	40
Power supply	1	200
Cables/jumpers	-	100
Bread board	1	20
Laser Cutting + Wood	-	150
DC MOTORS	5	800
DC Motor driver	3	150
3D Printing	5	140

Total Price: 1980 EGP