Faculty of Engineering

Ain Shams University

MCT 332

Design of Mechatronics (2)

Final Submission

Submitted to:

Dr. Shady Maged

Team: 17

Youssef Ashraf Ahmed	18P6536
Mohamed Sherif Mohamed Shaaban	18P5784
Nadin Ahmed Ali	18P6472

Table of Contents

Faculty of Engineering

Ain Shams University

Price guide guide Robot

1. Abstract

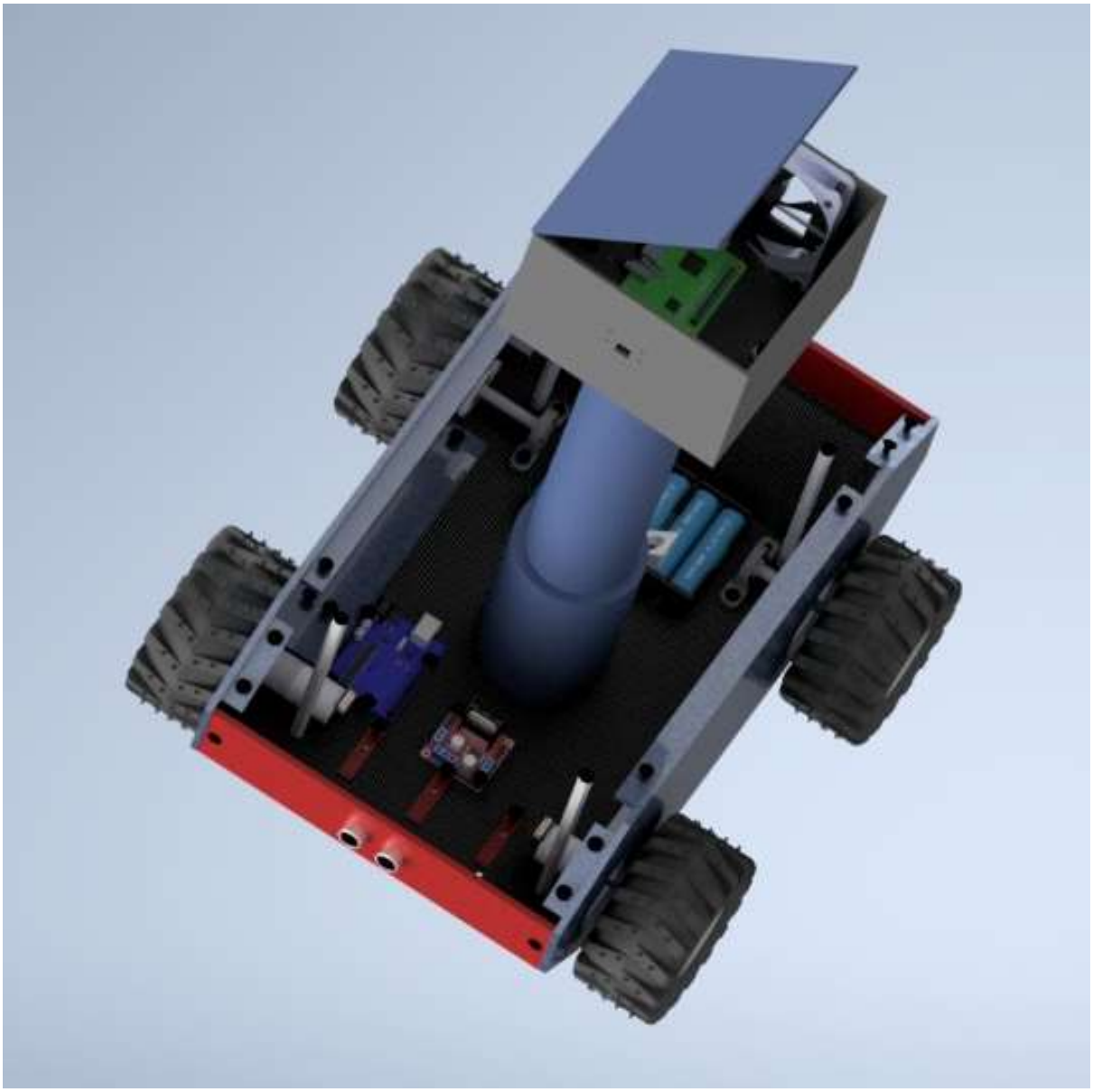
Supermarket numbers are countless all over the world. However, going to a supermarket without knowing the price of each item would feel curious and confused. In this work, we introduce the implementation of a price-guide robot using machine vision and sensors to ease the process of price guides. The robot will replace the current screens. The robot will follow black lines wherever they go, and provide information about the item once the clients have given it the command to do so. The robot detects moving bodies, thus providing information about items, using bar code.

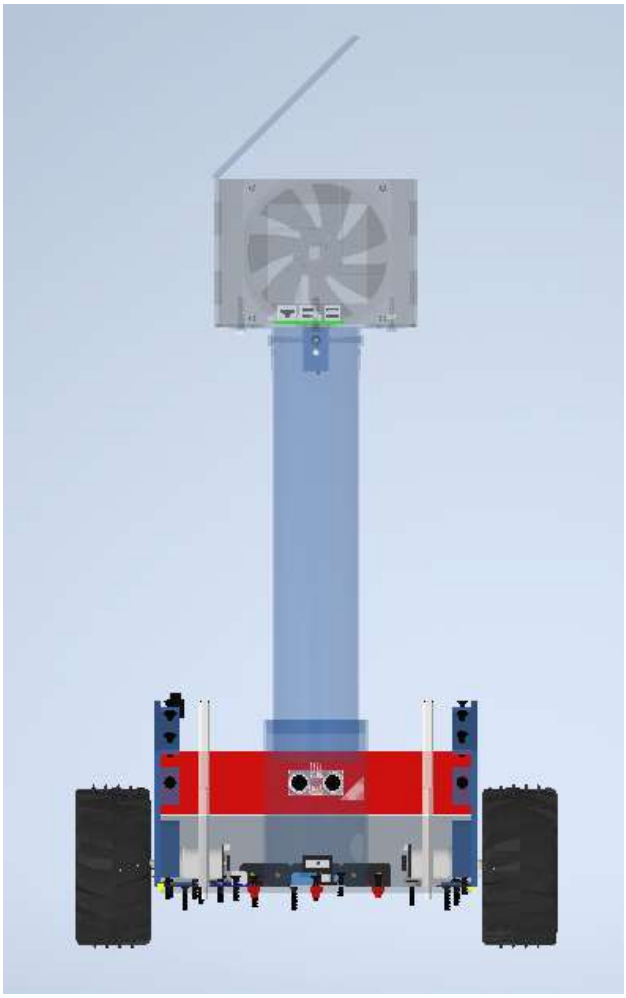
2. Mechanical Design

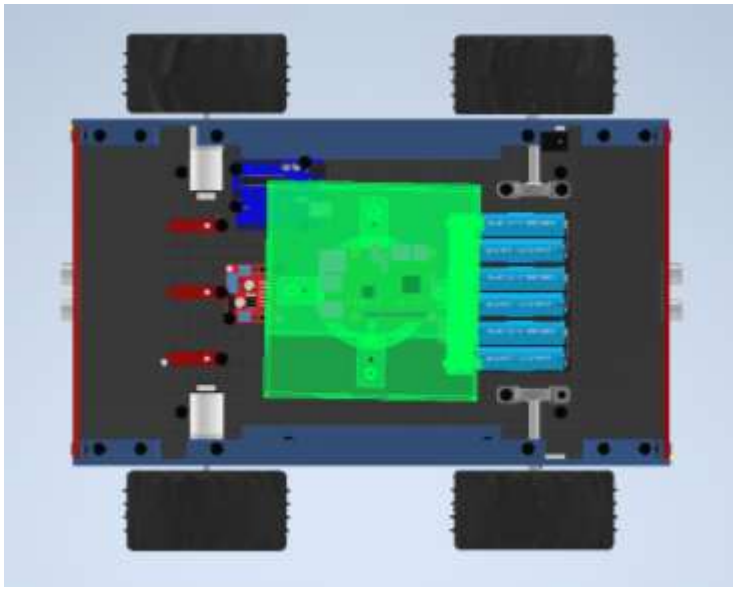
1. Overview

Faculty of Engineering

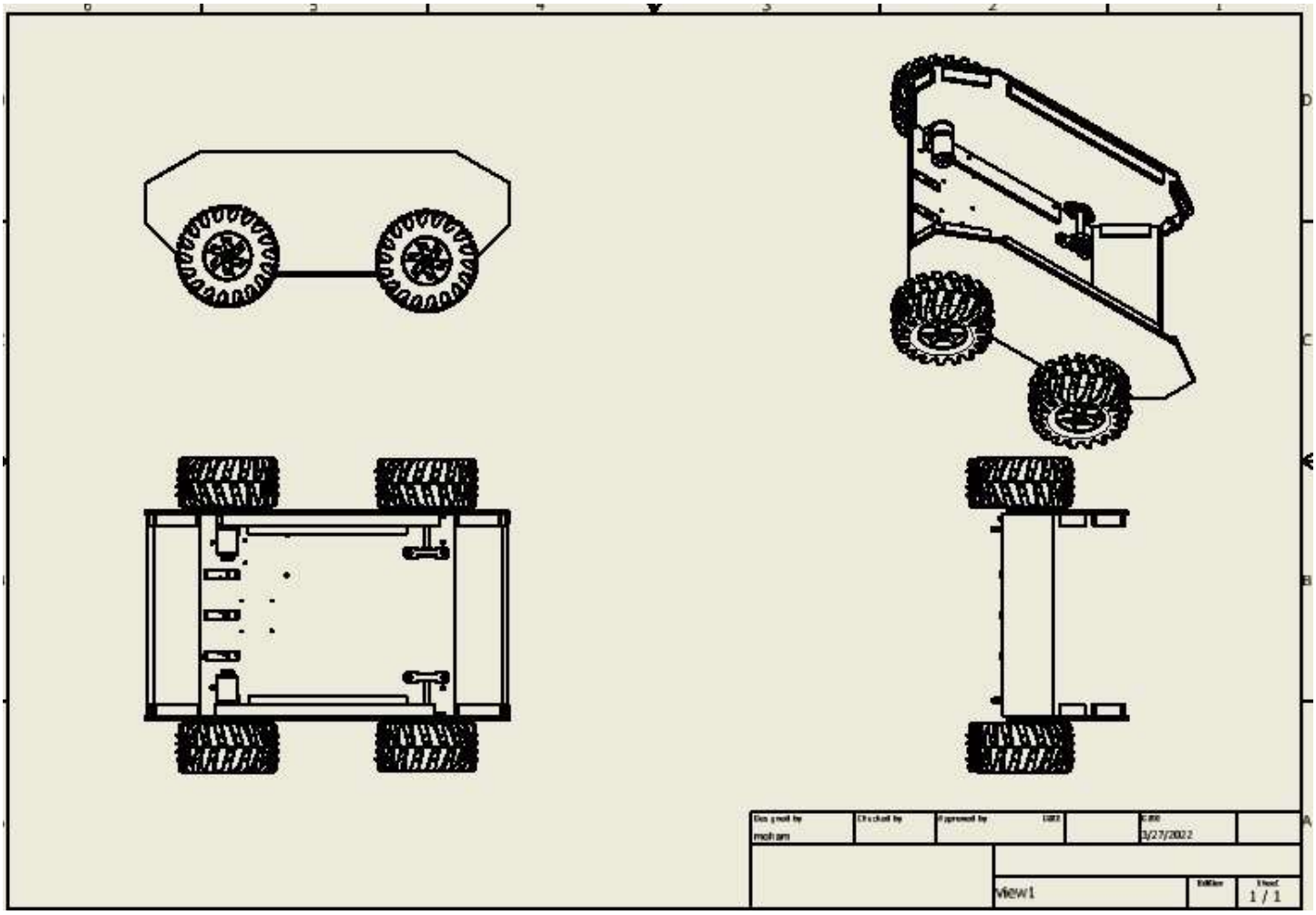
Ain Shams University

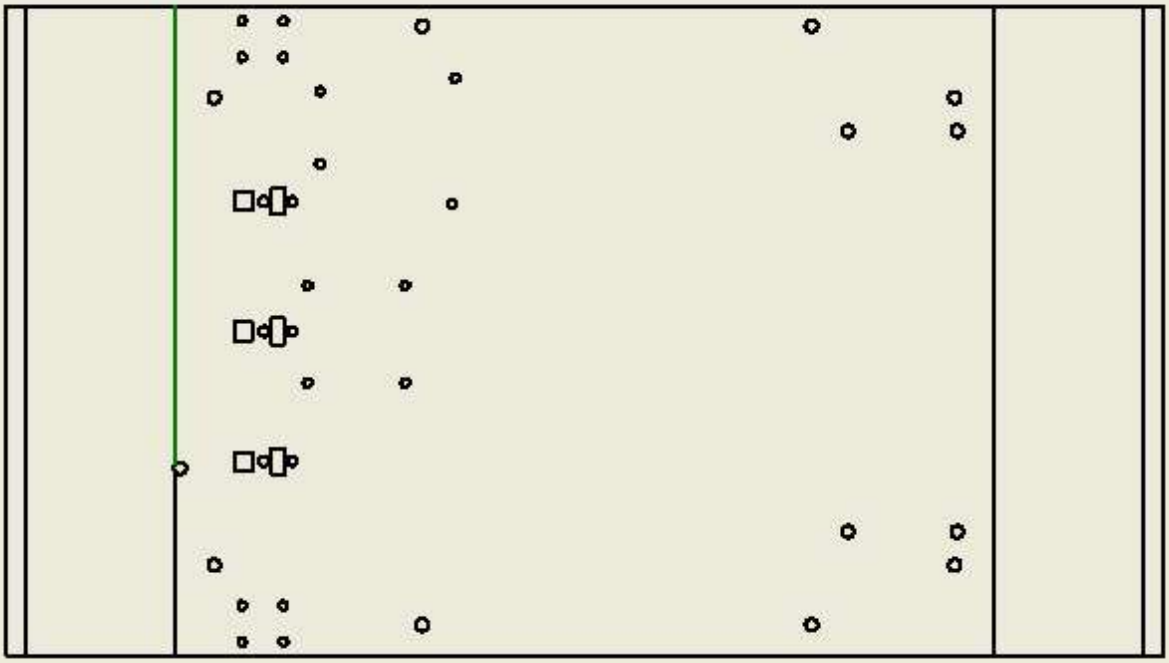


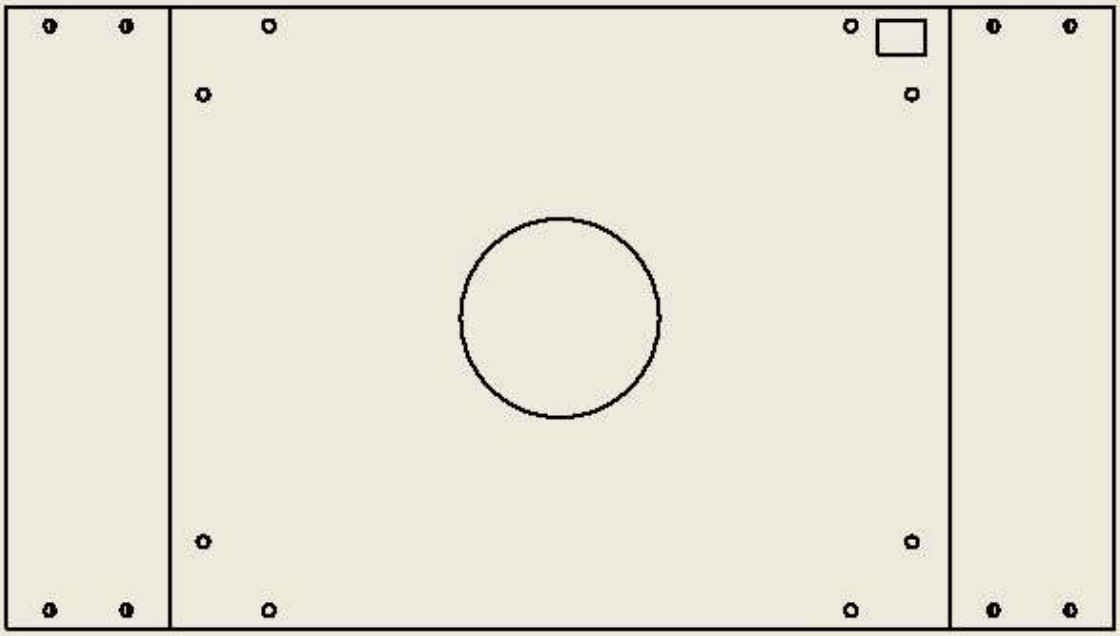




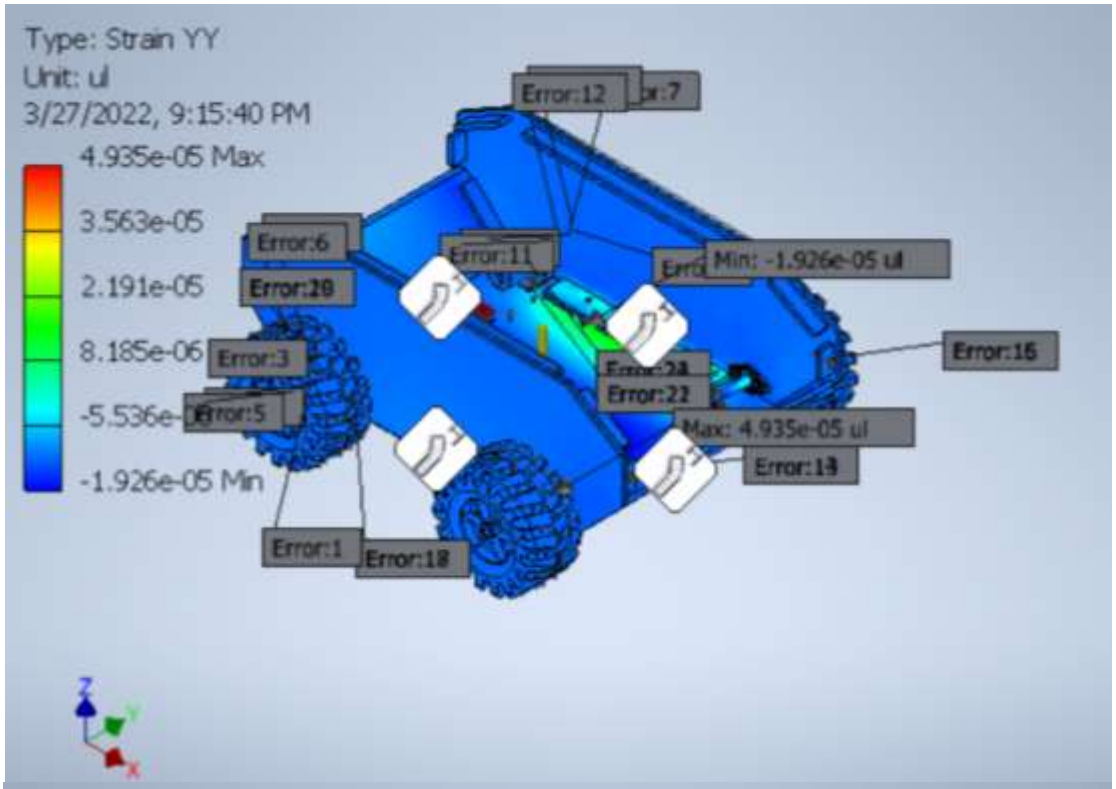
2. Working Drawings

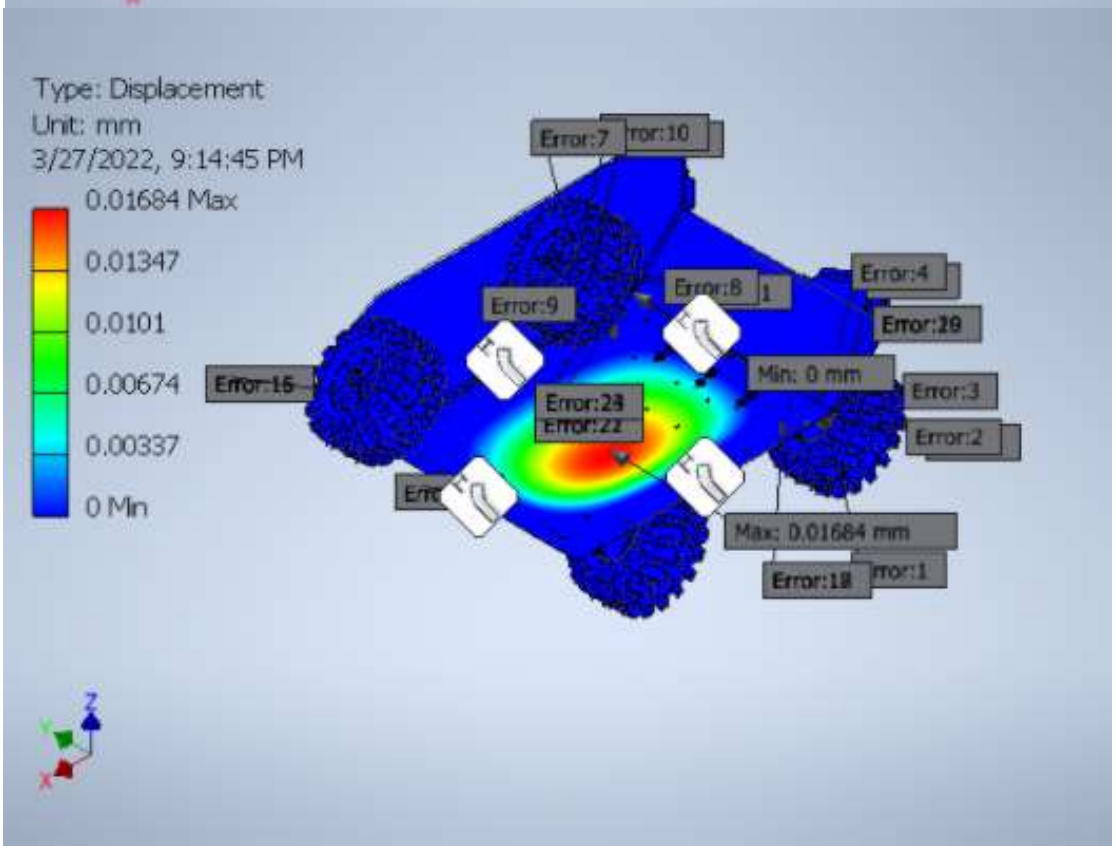






3. Stress Analysis





⊟ Reaction Force and Moment on Constraints

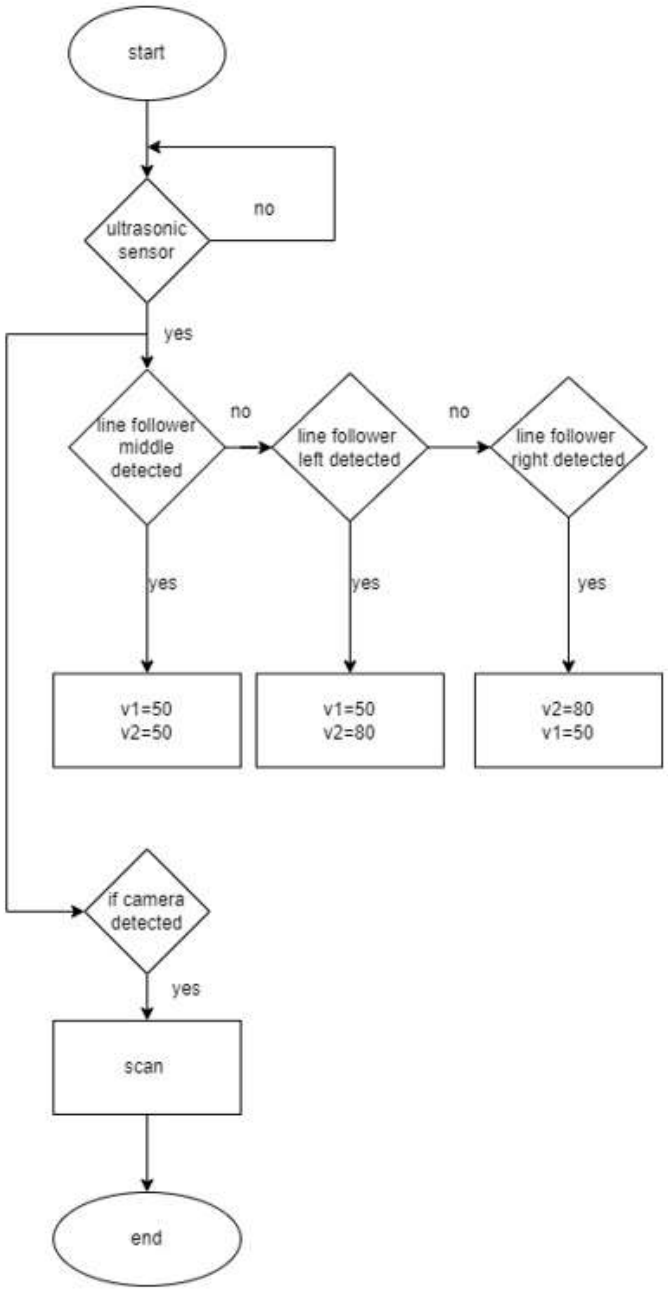
Constraint Name	Reaction Force		Reaction Moment	
	Magnitude	Component (X,Y,Z)	Magnitude	Component (X,Y,Z)
Fixed Constraint:1	200 N	0 N	3.16052 N m	-1.59083 N m
		0 N		2.73096 N m
		200 N		0 N m

⊟ Result Summary

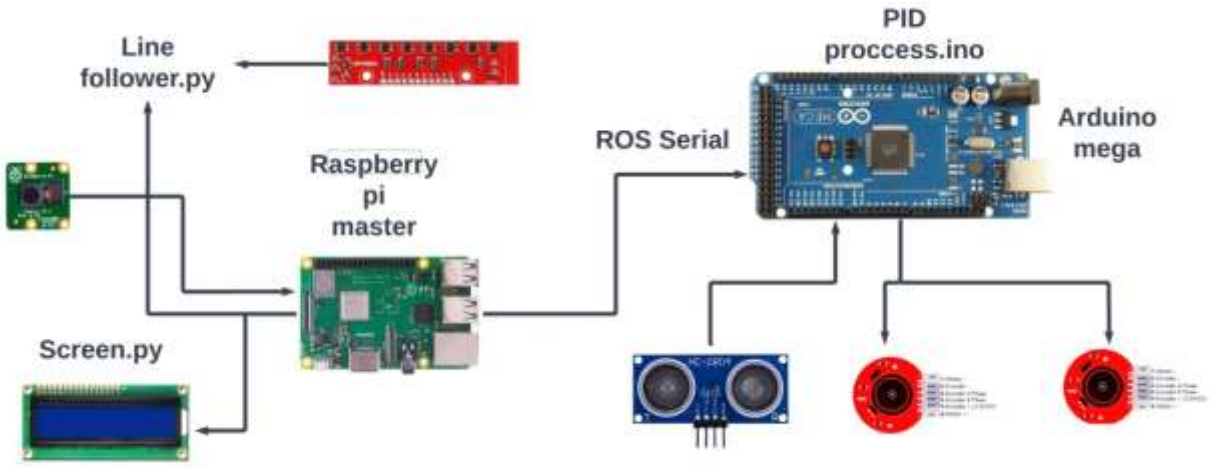
Name	Minimum	Maximum
Volume	3317940 mm^3	
Mass	8.95844 kg	
Von Mises Stress	0 MPa	10.1438 MPa
1st Principal Stress	-1.74112 MPa	7.6402 MPa
3rd Principal Stress	-10.876 MPa	1.31242 MPa

3. Software Architecture

1. Flowchart



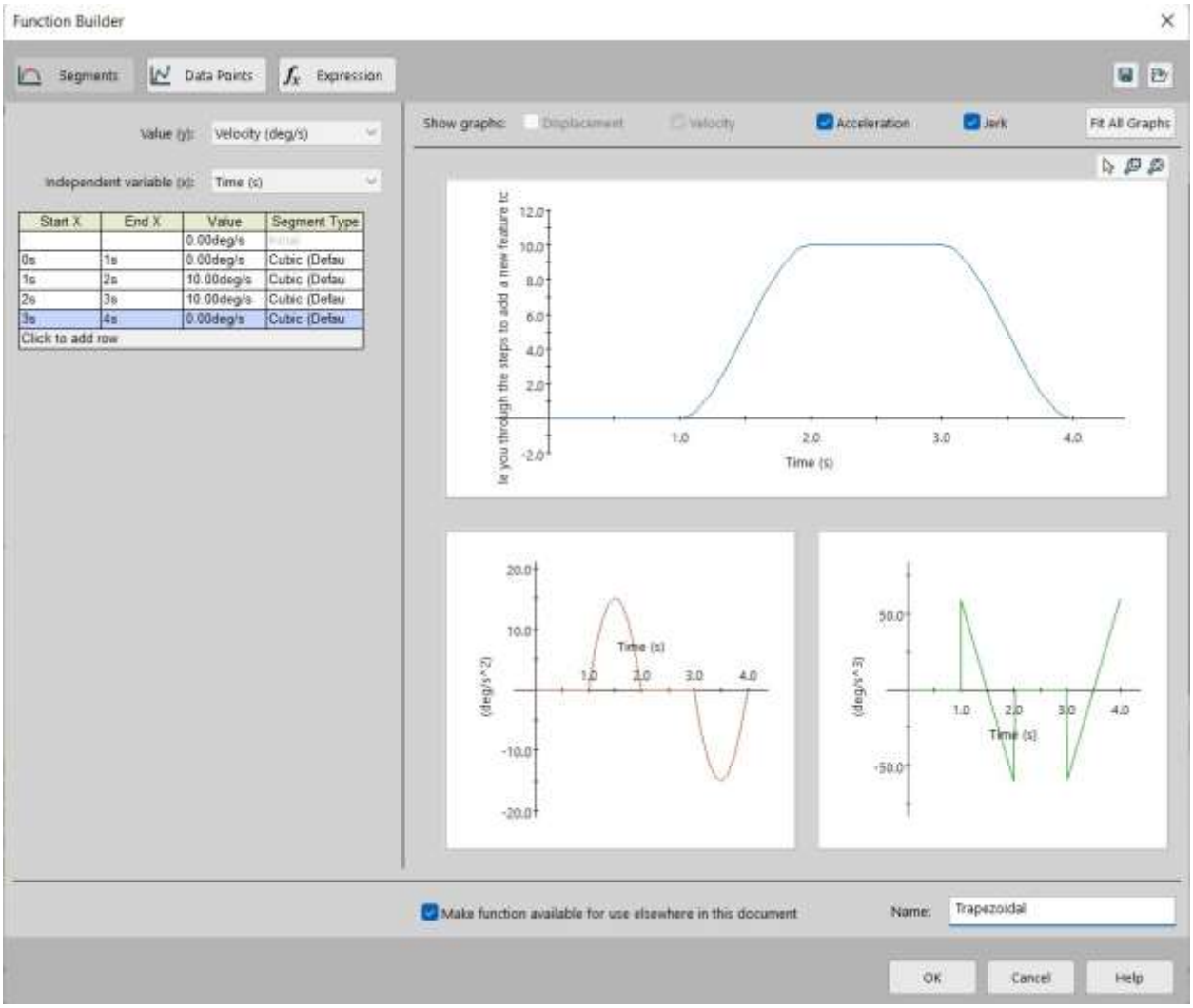
2. Communication



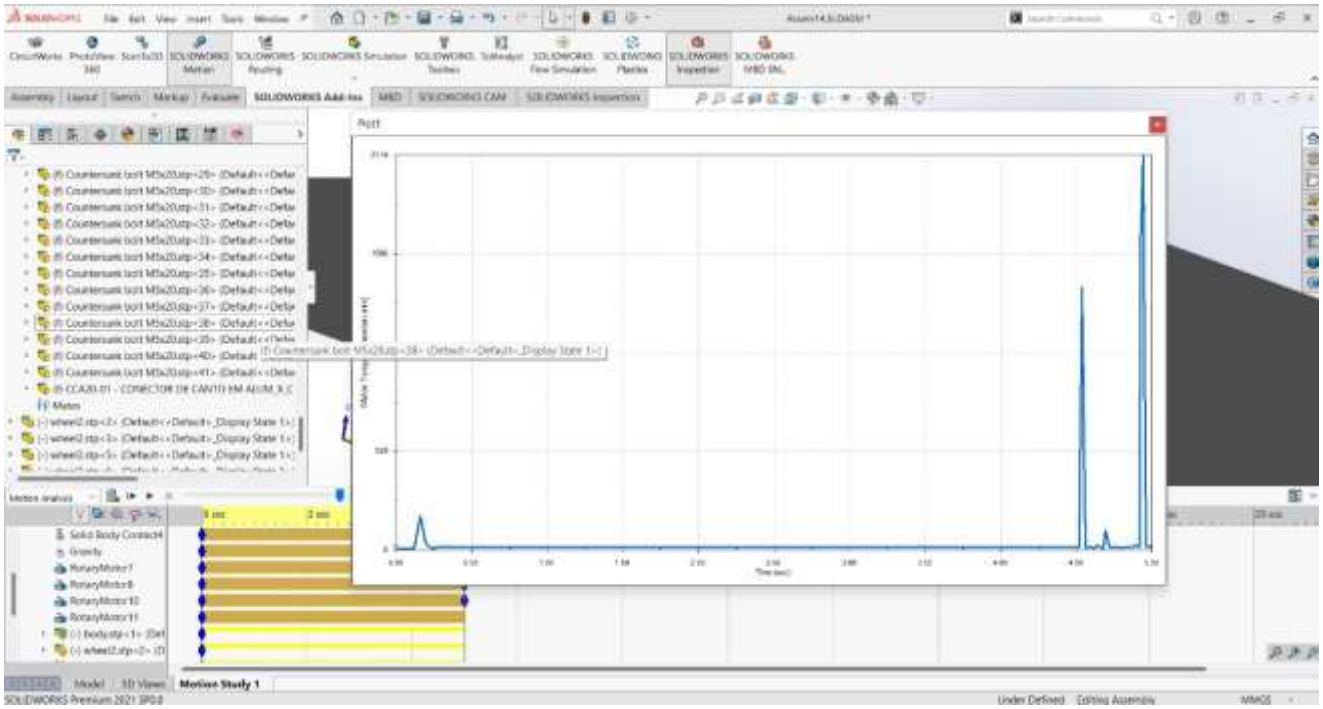
4. Actuator Sizing

1. Software

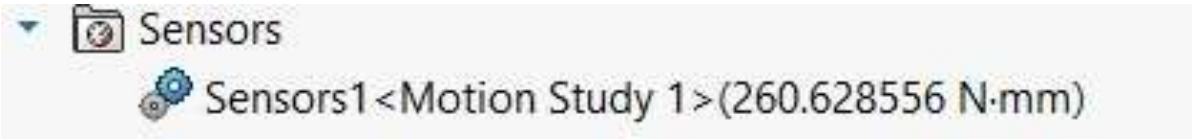
Velocity motion profile



Torque graph



Torque RMS

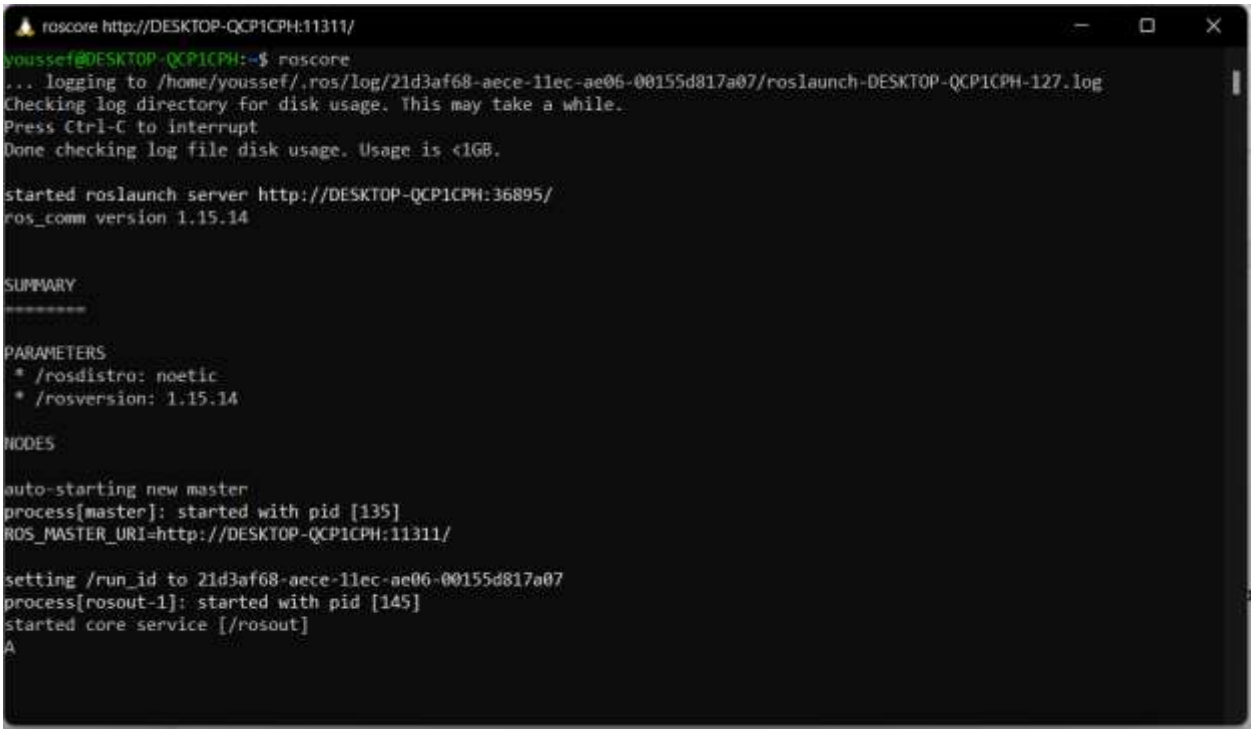


Will be multiplied by 2, so it will be 520 N.mm

5. Simulation

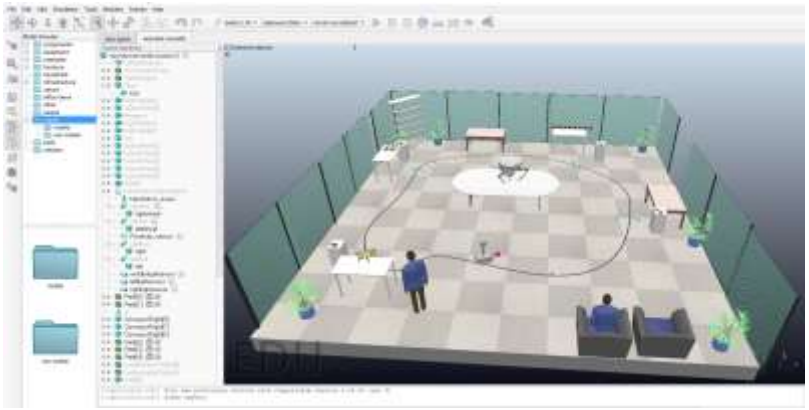
So, in order to simulate the robot in Coppeliasim environment we went through some steps

1- Open roscore



2- Open coppeliasim (ubuntu version)

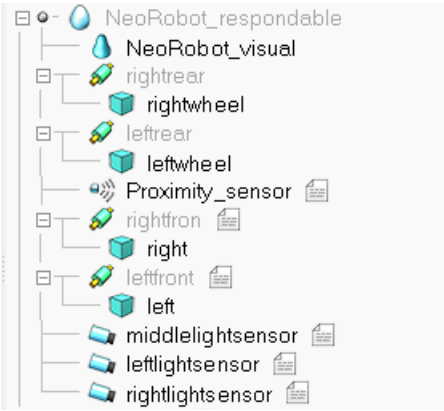
3- Open the scene that we have created and placed our robot in it



4- Open WSL new terminal and open the python file which contain the line following code

5- Run the simulation from coppeliasim

CoppliaSim scripts is as follows:



for the Right front motor, the child script is:

```
function speed_callback(msg)
    commandedVelocity = msg.data
end

function sysCall_init()
    commandedVelocity = 0.0
    jointHandle = sim.getObject('.')
    publisher = simROS.advertise('right_motor/actual_speed', 'std_msgs/Float32')
    subscriber = simROS.subscribe('right_motor/setpoint_speed', 'std_msgs/Float32', 'speed_callback')
end

function sysCall_actuation()
    sim.setJointTargetVelocity(jointHandle, commandedVelocity)
end

function sysCall_sensing()
   actualJointSpeed = sim.getJointVelocity(jointHandle)
   simROS.publish(publisher, {data=actualJointSpeed})
```

```
end

function sysCall_cleanup()
   simROS.shutdownPublisher(publisher)
   simROS.shutdownSubscriber(subscriber)
end
```

for the Left front motor, the child script is:

```
function speed_callback(msg)
   commandedVelocity = msg.data
end

function sysCall_init()
   commandedVelocity = 0.0
   jointHandle = sim.getObject('.')
   publisher = simROS.advertise('left_motor/actual_speed', 'std_msgs/Float32')
   subscriber = simROS.subscribe('left_motor/setpoint_speed', 'std_msgs/Float32',
'speed_callback')
end

function sysCall_actuation()
   sim.setJointTargetVelocity(jointHandle, commandedVelocity)
end

function sysCall_sensing()
  actualJointSpeed = sim.getJointVelocity(jointHandle)
  simROS.publish(publisher, {data=actualJointSpeed})
```

```
end


function sysCall_cleanup()
    simROS.shutdownPublisher(publisher)
    simROS.shutdownSubscriber(subscriber)
end
```

for the proximity sensor (ultrasonic sensor), the child script is:

```
function sysCall_init()
    sensorHandle = sim.getObject('.')
    publisher = simROS.advertise('proximity_sensor/state', 'std_msgs/Int32')
end
function sysCall_actuation()
 -- put your actuation code here
end
function sysCall_sensing()
    proximitySensorValue = sim.readProximitySensor(sensorHandle)
    simROS.publish(publisher, {data=proximitySensorValue})
end
function sysCall_cleanup()
    simROS.shutdownPublisher(publisher)
end
```

The code to publish and subscribe from and to coppeliasim

```
#!/usr/bin/env pyhton3
import rospy
import math
import time
from std_msgs.msg import Float32, Float32MultiArray, Int32
def pose_clbk(msg):
    global x, y, theta
    x = msg.data[0]
    y = msg.data[1]
    theta = msg.data[2]


if __name__ == "__main__":
    rightV = 0
    stopsign = 0
    leftV = 0
    speed = 3
    rospy.init_node("navigator_node")
    # Subscriber to the velocity commanding topic
    rospy.Subscriber("/wheel_odometry_localizer/pose", Float32MultiArray, pose_clbk)


    # Setup wheel speed publishers
    LWS_pub = rospy.Publisher("/sim_ros_interface/left_motor/setpoint_speed", Float32,
queue_size=10)
    RWS_pub = rospy.Publisher("/sim_ros_interface/right_motor/setpoint_speed", Float32,
queue_size=10)


    # Setup ROS rate
```

```
  rate = rospy.Rate(20)  # 10 Hz
  rospy.loginfo("Line follower node worked successfully")

  while not rospy.is_shutdown():
     rightV = -1
     leftV = -1
     flagstart = 0
     left_siganl = rospy.wait_for_message("/sim_ros_interface/leftvision/state", Float32)

     right_siganl = rospy.wait_for_message("/sim_ros_interface/rightvision/state", Float32)

     mid_siganl = rospy.wait_for_message("/sim_ros_interface/middlevision/state", Float32)

     proximity_msg = rospy.wait_for_message("/sim_ros_interface/proximity_sensor/state",Int32)  #
byceck 3la el sensor

     if (proximity_msg.data) != 0:  # proximity sees something
        rospy.loginfo("i will stop -- obstacle ")
        rightV = 0
        leftV = 0
     elif left_siganl.data == 1:
        rightV = - (speed + (0.5 * speed))
        leftV = - (speed - (0.5 * speed))
      #   rospy.loginfo("im here at left joint ")

     elif right_siganl.data == 1:
        rightV = - (speed - (0.5 * speed))
        leftV = - (speed + (0.5 * speed))
```

```
 #   rospy.loginfo("im here at right joint ")


elif mid_siganl.data == 1:
    leftV = -speed
    rightV = -speed
    flagstart = 1
 #   rospy.loginfo("im here  at middle ")


if (mid_siganl.data or right_siganl.data or left_siganl.data) == 0:
   # rospy.loginfo("im not in the right place ")
    leftV = speed
    rightV = speed


if  (mid_siganl.data and right_siganl.data and left_siganl.data) ==  1 :


        rospy.loginfo("i will stop now")
        stopsign = stopsign +1
        leftV=0
        rightV = 0
        LWS_pub.publish(Float32(leftV))
        RWS_pub.publish(Float32(rightV))
        time.sleep(4)
        rightV = -speed
        rightV = -speed
        if(stopsign == 1):
          rospy.loginfo("this is the first stop , welcome to the first table  ")


        if(stopsign == 2):
```

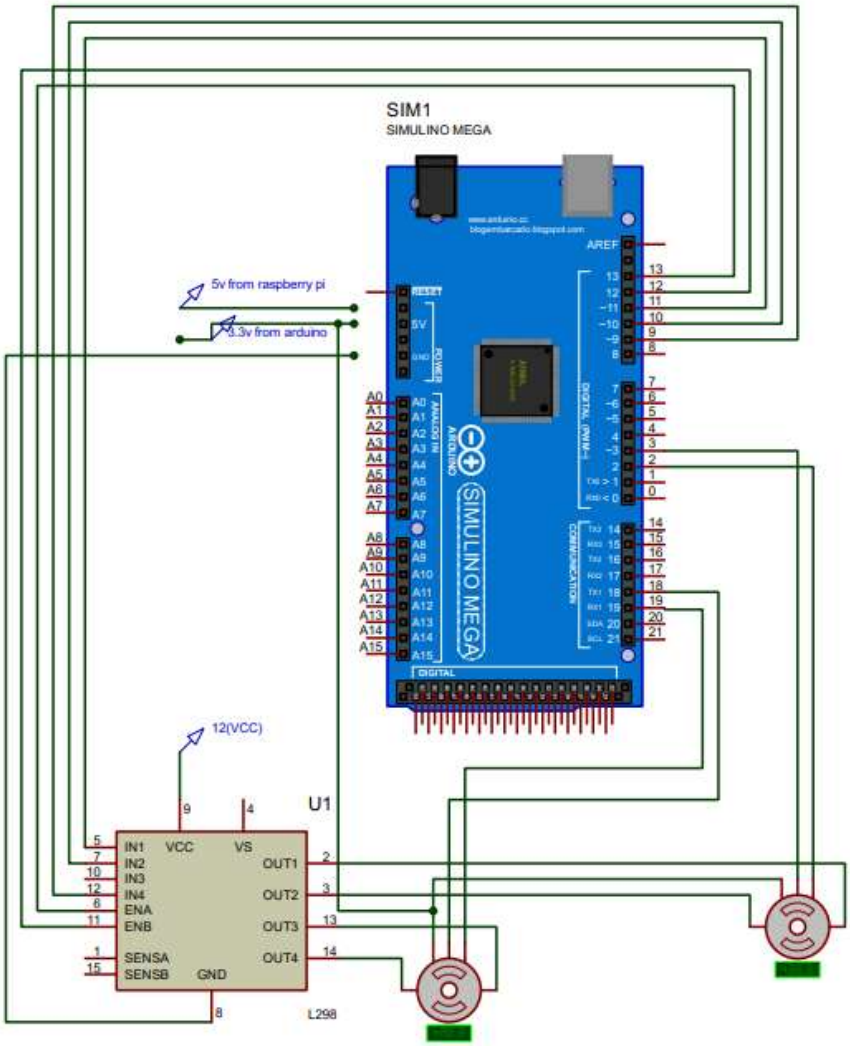
```
            rospy.loginfo("this is the second stop , welcome to the second table ")

        if(stopsign == 3):
            rospy.loginfo("this is the third stop , welcome to the third table ")

        if(stopsign == 4):
            rospy.loginfo("this is the fourth stop , welcome to the fourth table ")
        LWS_pub.publish(Float32(leftV))
        RWS_pub.publish(Float32(rightV))
        time.sleep(2)


    LWS_pub.publish(Float32(leftV))
    RWS_pub.publish(Float32(rightV))
    rate.sleep()
```

for the vision sensor (line follower sensor), the child script is:

```
function sysCall_init()
  middlevision = sim.getObject('.')
  publisher = simROS.advertise('middlevision/state', 'std_msgs/Float32')
end
function sysCall_actuation()
   -- put your actuation code here
end
```


```
function sysCall_sensing()
   result,data=sim.readVisionSensor(middlevision)
    if (result>=0) then
      if(data[11]<0.4) then
        comand = 1
      end
      if(data[11]>0.4) then
        comand = 0
      end
    end
   simROS.publish(publisher,  {data=comand})
end
function sysCall_cleanup()
   simROS.shutdownPublisher(publisher)
end
```

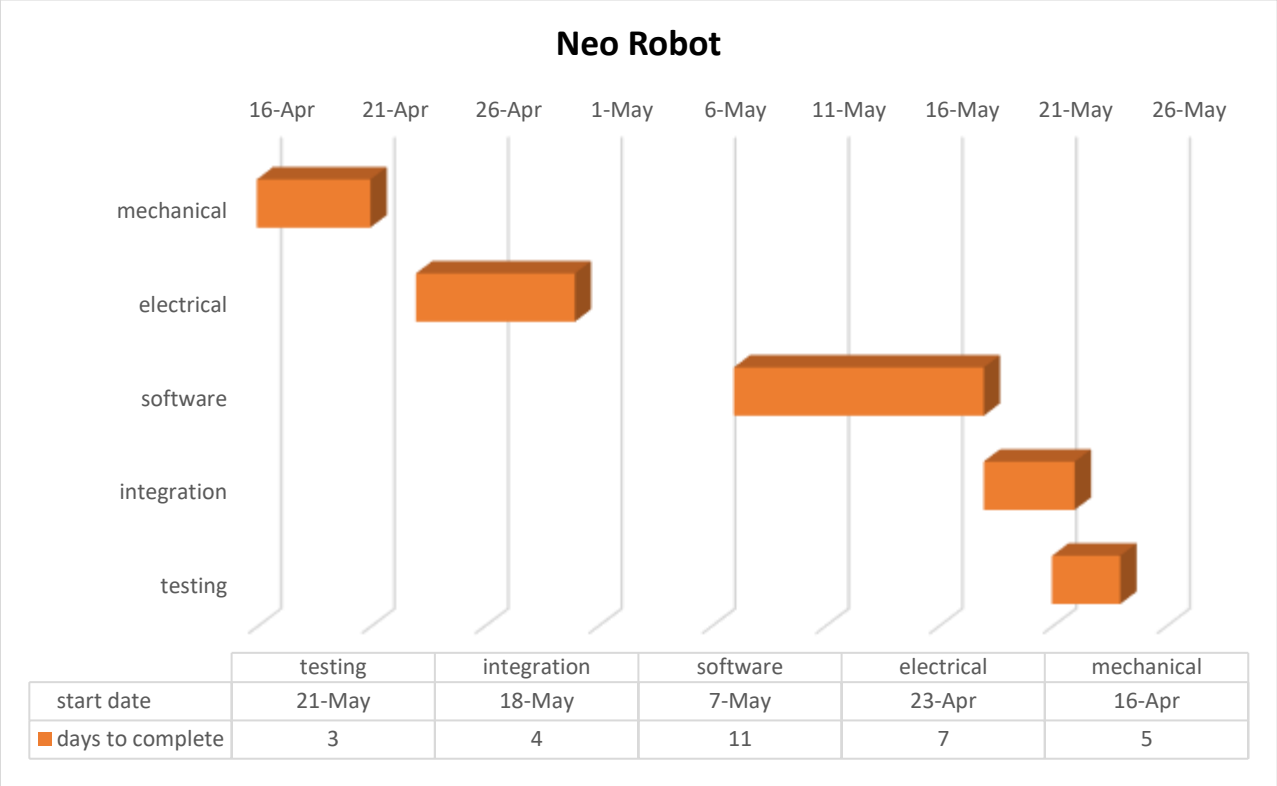
6. Wire drawings



7. Cost

item	cost
Arduino mega	350
Raspberry pi 4b	1100
camera	250
2 motors	460
battery	300
Sheet metal	490
Wood laser cut	35
Motor driver	50
4 wheels	300
Ultrasonic sensor	40
2 motor brackets	50
6 Vertical flange bearings	180
4 horizontal flange bearing	120
4 pulleys and 2 belts	120
Power bank	200
PVC	50
LCD screen	40
I2C module	60
total	4195

8. Gantt chart

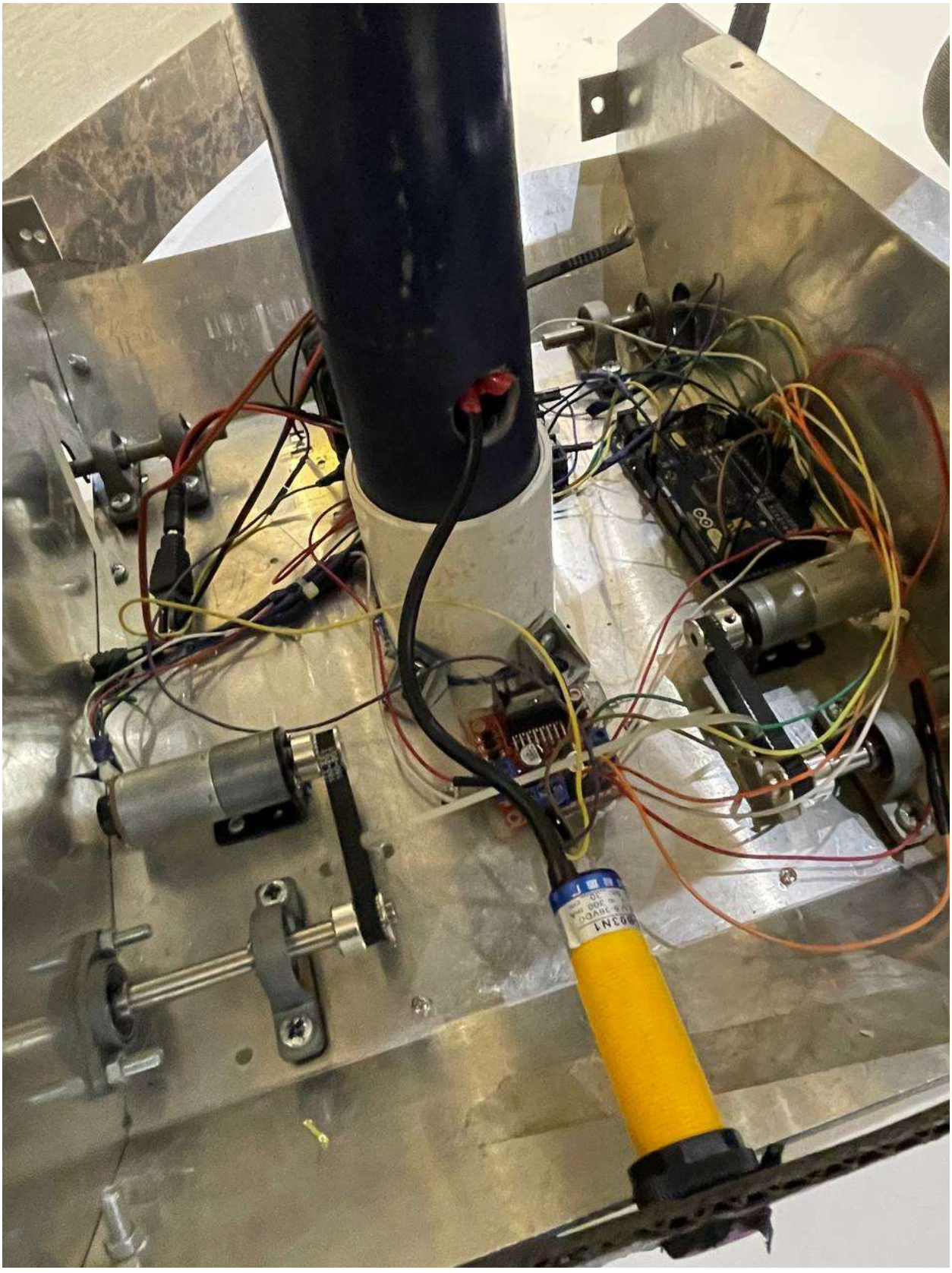


	testing	integration	software	electrical	mechanical
start date	21-May	18-May	7-May	23-Apr	16-Apr
days to complete	3	4	11	7	5

9. Fixation

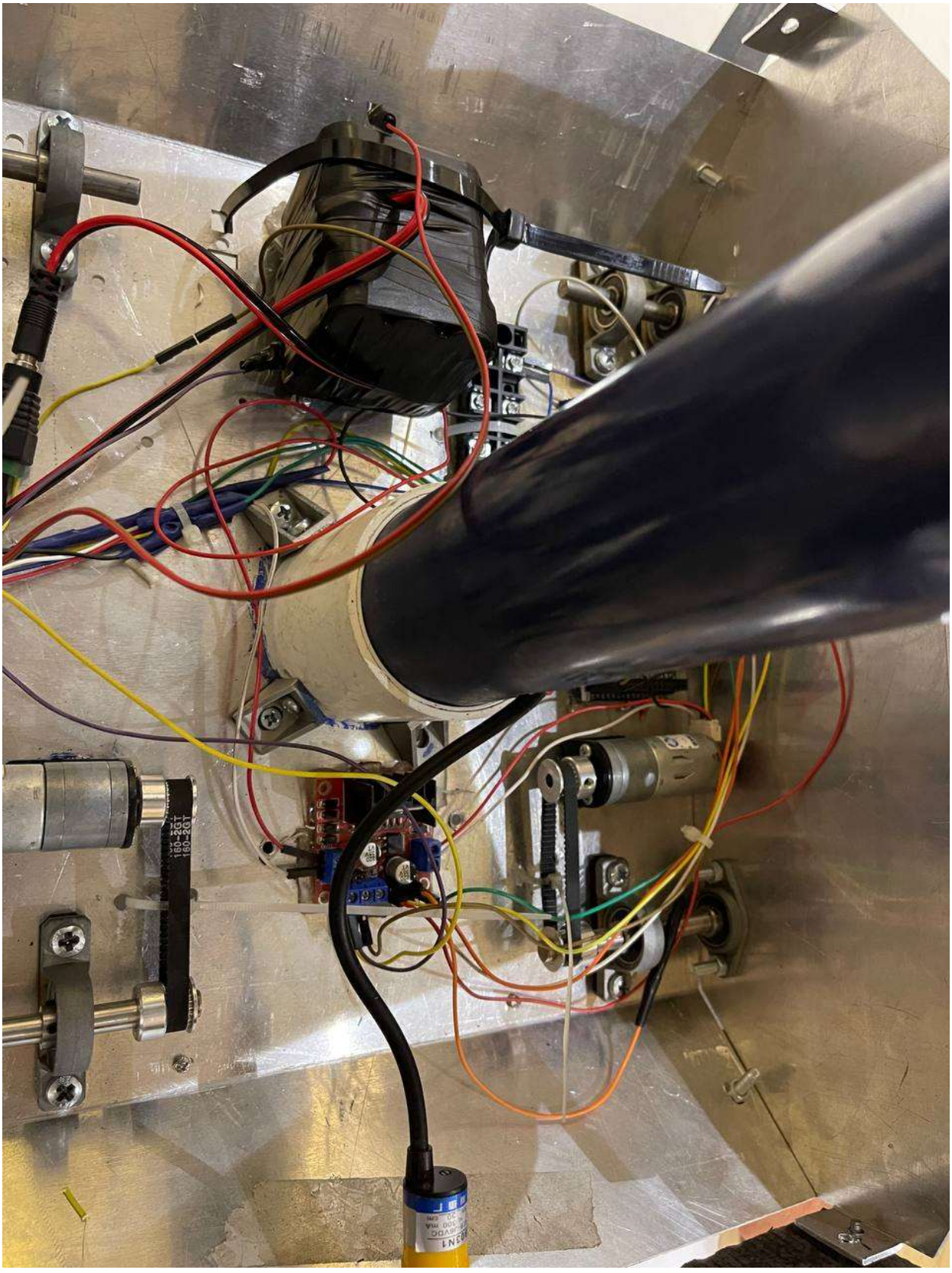
Faculty of Engineering

Ain Shams University



Faculty of Engineering

Ain Shams University



10. Working environment

Faculty of Engineering

Ain Shams University

Faculty of Engineering

Ain Shams University

Faculty of Engineering

Ain Shams University

11. Integration

```
#define kp_100 0.02

#define kp2_100 0.05

#define ki_100 0.00015

#define ki2_100 0.00015

#define kd_100 0.5

#define kd2_100 0.5


#define kp_80 0.05

#define kp2_80 0.05

#define ki_80 0.0002

#define ki2_80 0.0002

#define kd_80 0.5

#define kd2_80 0.5

#define prox_pin 15


#include <ros.h>

#include <std_msgs/Float32.h>


ros::NodeHandle nh;


std_msgs::Float32 str_msg;

ros::Publisher chatter("chatter", &str_msg);


float var;


float camera_flag ;
```

```
float prox_flag ;



////////////////////////////////////////////////// communication
//////////////////////////////////////////


float kp = 0.08;
float ki = 0.0002;
float kd = 0.5;
float kp2 = 0.05;
float ki2 = 0.0002;
float kd2 = 0.5;
unsigned long t;
unsigned long t_prev = 0;
////////////////////////////////
const byte interruptPinA = 2;
const byte interruptPinB = 3;
////////////////////////////////////////////////////////////////////////////*MOHEEEEEEEEEEEMMMM
//////////
const byte interruptPinC = 18;
const byte interruptPinD = 19;
volatile long EncoderCount1 = 0;
volatile long EncoderCount2 = 0;

const byte PWMPin1 = 8;
const byte PWMPin2 = 9;
const byte DirPin1 = 11;
```

```
const byte DirPin2 = 12;
const byte DirPin3 = 5;
const byte DirPin4 = 6;
////////////////////////////////
volatile unsigned long count = 0;
unsigned long count_prev = 0;

float Theta, RPM, RPM_d;
float Theta2, RPM2, RPM_d2;
float Theta_prev = 0;
float Theta_prev2 = 0 ;
int dt;
float RPM_max = 230;

#define pi 3.1416
float Vmax = 12;
float Vmin = -12;
float V = 0.1;
float V2 = 0.1;

float e, e_prev = 0, inte, inte_prev = 0;
float e2, e_prev2 = 0, inte2, inte_prev2 = 0;


 void ISR_EncoderA(){
  int b = digitalRead(interruptPinB);
  if(b>0){
     EncoderCount1++;
```

```
  }
else{
 EncoderCount1--;
 }
}

void ISR_EncoderC(){
 int d = digitalRead(interruptPinD);
 if(d>0){
   EncoderCount2++;
  }
else{
  EncoderCount2--;
 }
}


float sign(float x) {
 if (x > 0) {
  return 1;
 } else if (x < 0) {
  return -1;
 } else {
  return 0;
 }
}
```

```
//**Motor Driver Functions**
int PWMval = 0;
void WriteDriverVoltage1(float V, float Vmax) {    /// eh heya el v??   el vmax = 6 volt
  PWMval = int(255 * abs(V) / Vmax);
  if (PWMval > 255) {
    PWMval = 255;
  }
  if (V > 0) {
    digitalWrite(DirPin1, HIGH);
    digitalWrite(DirPin2, LOW);

  }
  else if (V < 0) {
    digitalWrite(DirPin1, LOW);

    digitalWrite(DirPin2, HIGH);

  }
  else {
    digitalWrite(DirPin1, LOW);
    digitalWrite(DirPin2, LOW);
  }
  analogWrite(PWMPin1, PWMval);
}


void WriteDriverVoltage2(float V, float Vmax) {    /// eh heya el v??   el vmax = 6 volt
  PWMval = int(255 * abs(V) / Vmax);
```

```
 if (PWMval > 255) {
   PWMval = 255;
 }
 if (V > 0) {
   digitalWrite(DirPin3, HIGH);
   digitalWrite(DirPin4, LOW);

 }
 else if (V < 0) {
   digitalWrite(DirPin3, LOW);

   digitalWrite(DirPin4, HIGH);
 }

 else {
   digitalWrite(DirPin3, LOW);
   digitalWrite(DirPin4, LOW);
 }
 analogWrite(PWMPin2, PWMval);
}
```

```
void messageCb(const std_msgs::Float32 &msg)
{
  var=msg.data;
```

```
if(var == 1){
 RPM_d = 80;
 RPM_d2= -50;
 kp = kp_80;
 ki = ki_80;
 kd = kd_80;
 kp2 = kp2_80;
 ki2 = ki2_80;
 kd2= kd2_80;
 str_msg.data = 1;
chatter.publish( &str_msg );
}
if(var == 2){
 RPM_d = 80;
 RPM_d2= -50;
 kp = kp_80;
 ki = ki_80;
 kd = kd_80;
 kp2 = kp2_80;
 ki2 = ki2_80;
 kd2= kd2_80;
 str_msg.data = 2;
chatter.publish( &str_msg );
}
if(var == 3){

 RPM_d = 50;
 RPM_d2= 50;
```

```
 kp = kp_80;
 ki = ki_80;
 kd = kd_80;
 kp2 = kp2_80;
 ki2 = ki2_80;
 kd2= kd2_80;
  str_msg.data = 3;
chatter.publish( &str_msg );
}
if(var == 4){
 RPM_d = -50;
 RPM_d2= 80;
 kp = kp_80;
 ki = ki_80;
 kd = kd_80;
 kp2 = kp2_80;
 ki2 = ki2_80;
 kd2= kd2_80;
 kd2= kd2_100;
str_msg.data = 4;
chatter.publish( &str_msg );

}
if(var == 5 ){

 RPM_d = -50;
```

```
  RPM_d2= 80;
  kp = kp_80;
  ki = ki_80;
  kd = kd_80;
  kp2 = kp2_80;
  ki2 = ki2_80;
  kd2= kd2_80;
 str_msg.data = 5;
 chatter.publish( &str_msg );
 }

 if (var== 0){

  RPM_d = 50;
  RPM_d2= 50;

  kp = kp_80;
  ki = ki_80;
  kd = kd_80;
  kp2 = kp2_80;
  ki2 = ki2_80;
  kd2= kd2_80;
 str_msg.data = 0;
 chatter.publish( &str_msg );
}
}
void messageCb2(const std_msgs::Float32 &msg1){
 camera_flag=msg1.data;
```

```
 }

ros::Subscriber<std_msgs::Float32> sub("line_follower", &messageCb);
ros::Subscriber<std_msgs::Float32> sub2("camera_vision", &messageCb2);




void setup() {
  nh.initNode();
  nh.subscribe(sub);
  nh.advertise(chatter);
  nh.subscribe(sub2);

 pinMode(interruptPinA, INPUT);
 pinMode(interruptPinB, INPUT);

   attachInterrupt(digitalPinToInterrupt(interruptPinA), ISR_EncoderA,  RISING);
   attachInterrupt(digitalPinToInterrupt(interruptPinC), ISR_EncoderC, RISING);


  pinMode(DirPin1, OUTPUT);
  pinMode(DirPin2, OUTPUT);
  // for second driver
  pinMode(interruptPinC, INPUT);
  pinMode(interruptPinD, INPUT);

  pinMode(DirPin3, OUTPUT);
```

```
 pinMode(DirPin4, OUTPUT);
 pinMode(prox_pin, INPUT);
 cli();
 TCCR1A = 0;
 TCCR1B = 0;
 TCNT1 = 0;
 OCR1A = 12499; //Prescaler = 64
 TCCR1B |= (1 << WGM12);
 TCCR1B |= (1 << CS11 | 1 << CS10);
 TIMSK1 |= (1 << OCIE1A);
 sei();
}

void loop() {


 if(camera_flag == 10 or camera_flag == 11 or prox_flag== 1){
 analogWrite(PWMPin2, 0);
 analogWrite(PWMPin1, 0);
  digitalWrite(DirPin3, HIGH);
  digitalWrite(DirPin4, LOW);
  digitalWrite(DirPin1, HIGH);
  digitalWrite(DirPin2, LOW);
 delay(3000);
 analogWrite(PWMPin2, 100);
 analogWrite(PWMPin1, 100);
 delay(1500);
 }
```

```
/* if(prox_flag == 1){
 analogWrite(PWMPin2, 0);
 analogWrite(PWMPin1, 0);
  digitalWrite(DirPin3, HIGH);
  digitalWrite(DirPin4, LOW);
  digitalWrite(DirPin1, HIGH);
  digitalWrite(DirPin2, LOW);
  delay(5000);

  }*/


   if(prox_flag !=1 ){
  if(camera_flag == 12){
 if (count > count_prev) {
   t = millis();    // b7sb el wa2t mn bdayt el loop
   Theta = EncoderCount1 / 300.0;    // el one revolution feeha 900 count yb2a a2dr ageeb el theta
   Theta2 = EncoderCount2 / 300.0;
   dt = (t - t_prev);  // theta 3ady
   //RPM_d = -80  ;
   //RPM_d2 = -80;

   RPM = (Theta - Theta_prev) / (dt / 1000.0) * 60;  // b2sm 3la alf 34an a7wl milly to seconds and then *
60 to get per menuite

   RPM2 = (Theta2 - Theta_prev2) / (dt / 1000.0) * 60;
   e = RPM_d - RPM;
```

```
e2 = RPM_d2 - RPM2;


inte = inte_prev + (dt * (e + e_prev) / 2);
inte2 = inte_prev2 + (dt * (e2 + e_prev2) / 2);



V = kp * e + ki * inte + (kd * (e - e_prev) / dt) ;

V2 = kp2 * e2 + ki2 * inte2 + (kd2 * (e2 - e_prev2) / dt) ;

if (V > Vmax) {
  V = Vmax;
  inte = inte_prev;
}
if (V < Vmin) {
  V = Vmin;
  inte = inte_prev;

}
if (V2 > Vmax) {
  V2 = Vmax;

  inte2 = inte_prev2;
}
if (V2 < Vmin) {
  V2 = Vmin;
  inte2 = inte_prev2;
}
```

```
 WriteDriverVoltage1(V, Vmax);
 WriteDriverVoltage2(V2, Vmax);



  //Serial.print(RPM_d2); Serial.print(" \t");
  //Serial.print(RPM2); Serial.print(" \t ");
  //Serial.print(V2); Serial.println("\t  ");

    Serial.print(RPM_d2); Serial.print(" \t");
    Serial.print(RPM2); Serial.print(" \t ");
    Serial.print(V2); Serial.println("\t  ");
    //Serial.print(e); Serial.println("  ");

  Theta_prev = Theta;
  count_prev = count;
  t_prev = t;
  inte_prev = inte;
  e_prev = e;



  Theta_prev2 = Theta2;
  inte_prev2 = inte2;
  e_prev2 = e2;

}}}
```

```
nh.spinOnce();
prox_flag =digitalRead(prox_pin);
 delay(1);
}



ISR(TIMER1_COMPA_vect) {
 count++;
 Serial.print(count * 0.05); Serial.print(" \t");
}
```

12. Link of working video

https://drive.google.com/drive/folders/1XhsmOpy6A1cEQqL19sPOZ6LgDBN4PRdy?usp=sharing