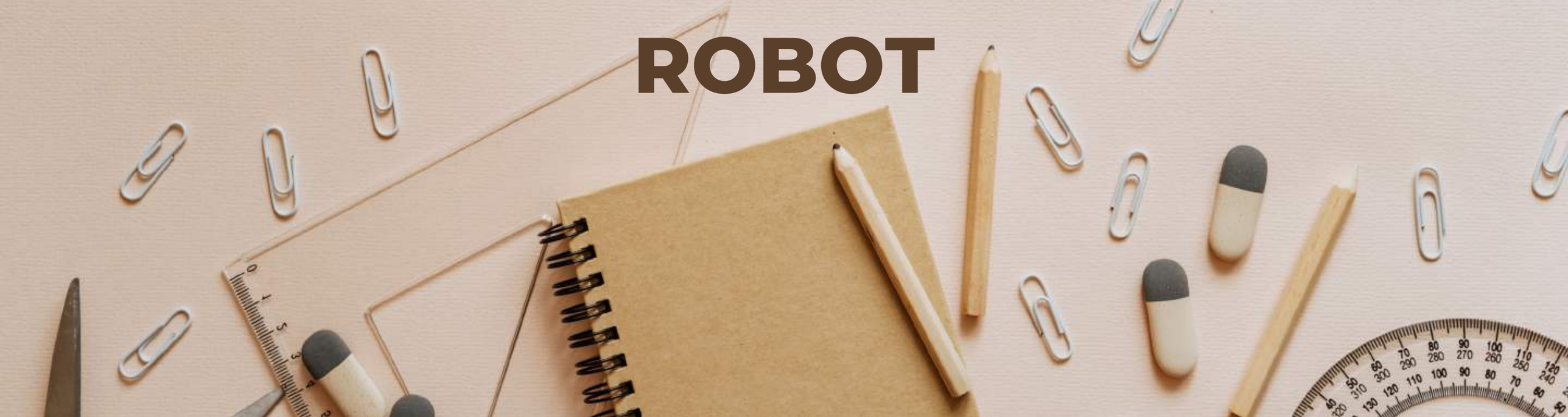
TEAM ID: 12

ARDUINO OBSTACLE AVOIDING LINE FOLLOWER ROBOT



TEAM MEMBERS:

ANKIT KUMAR PATEL

22ME10008

Team leader and procurement of components, Making electrical connections, coding, debugging the code.

SAMPRIT SAHA

22CE10066

Assembling the components, testing the working of components and preparing the presentation.

DIDWM MOCHAHARY

22ME10027

Designing circuit in TinkerCAD, testing the working of components. coding, debugging the code.

SUBHANSHU KUMAR

22MA10060

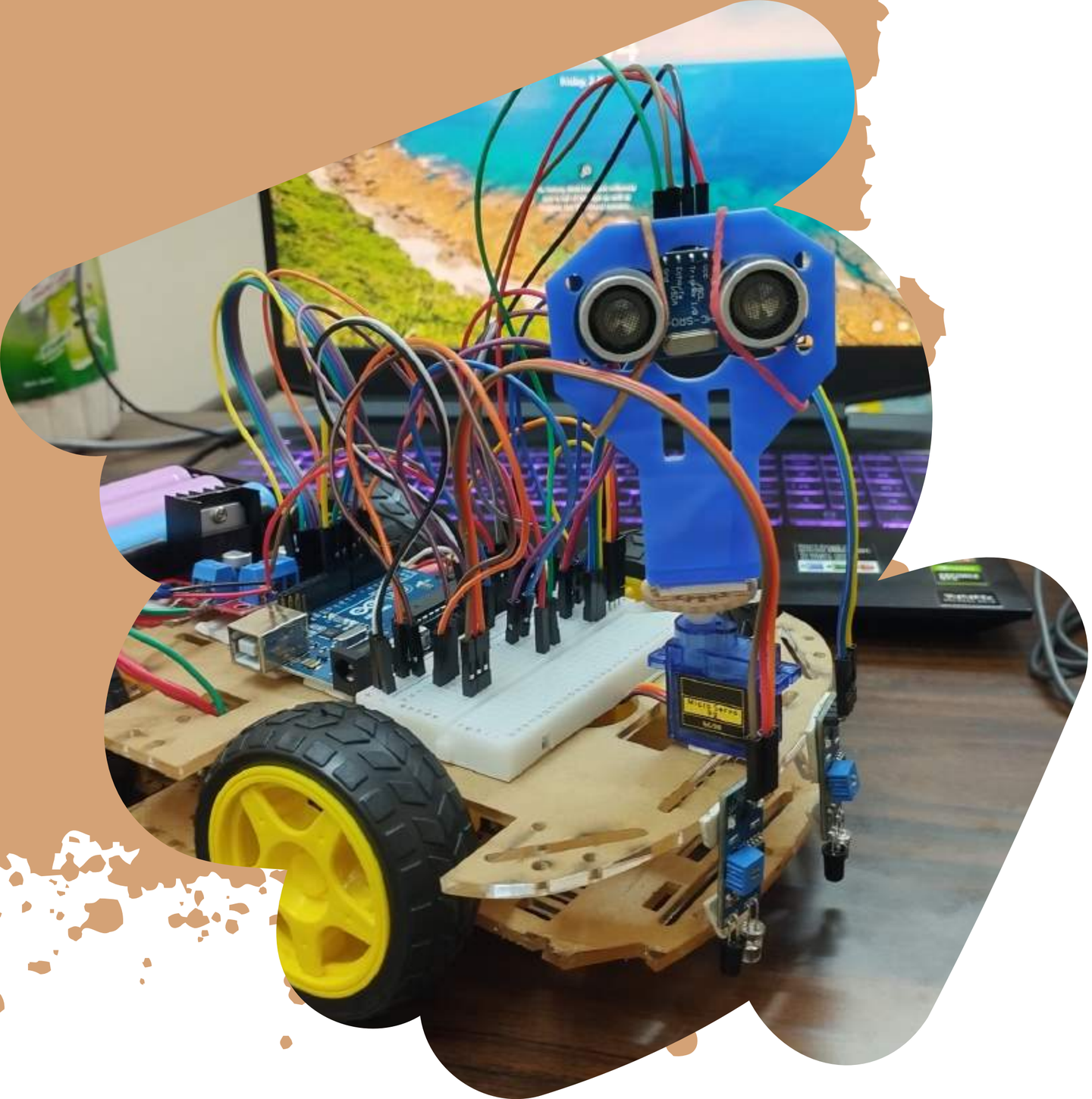
Designing circuit in TinkerCAD, coding, debugging the code.

BACKGROUND AND MOTIVATION:

Employee safety is a prime concern in various industries, including logistics and warehousing, manufacturing, metals & heavy machinery, and automotive, as workers are required to accomplish potentially dangerous activities and work in environments that pose a threat to personal safety. In most of these industries, tasks such as loading, unloading, storage, and movement of goods are carried out manually, putting human workers at risk

AGVS ensure reduced instances of accidents and greater safety of workers.

Industries are focusing on improving workplace safety to eliminate accidents, reduce downtime, and increase productivity. There is growing awareness about AGVS, and an increasingly large number of companies are implementing AGVS in their warehouses. These vehicles are capable of lifting heavier weights and can transport materials without delay. Intelligent unmanned vehicles are less prone to errors and help companies reduce accidents and ensure safety at the workplace.



OBJECTIVE:

TO DESIGN A LINE FOLLOWER ROBOT THAT CAN AUTONOMOUSLY FOLLOW A SPECIFIC PATH MARKED BY A LINE, ALONG WITH THE FUNCTIONALITY OF AVOIDING OBSTACLES.

METHODOLOGY:

THE IR SENSORS TRACK THE PATH AND ULTRASONIC SENSOR SENSE THE OBSTACLE ON THE LINE PATH. IF THERE IS ANY OBSTACLE, THE CAR AUTOMATICALLY DIVERTS ITS POSITION TO LEFT OR RIGHT. AVOIDES THE OBSTACLE AND AGAIN RETURNS BACK TO IT'S ORIGINAL PATH.

Items used :

1. Arduino Uno R3 Development Board with ATMEGA328
2. L298N Motor Driver/Servo Shield for Arduino
3. Digital IR Sensors Obstacle Avoidance Module(x2)
4. HC-SR04 Ultrasonic Sensor
5. Mini Servo Motor SG - 90 180° Rotation
6. 100 RPM BO Motor Double Shaft I Shape connected with wheels. (x4)
7. 20cm Female-Female, Male-Female,Male-Male
8. Ultrasonic Sensor Holder
9. Acrylic Chassis
10. 12V HW Battery

Items used and their technical specifications

- **Arduino Uno R3 Development Board with ATMEGA328**
- It is based on the Atmel's ATmega328
- It has 14 digital input output pins of which 6 can be used as PWM outputs and 6 analog inputs

- **L298N Motor Driver/Servo Shield for Arduino**
- Driver chip: L298 dual H-bridge driver chip.
- Operates up to 35V DC
- Drive part of the peak current Io: 2A / Bridge
- Logical part of the terminal power supply range VSS: 4.5V-5.5V
- Maximum power consumption: 20W

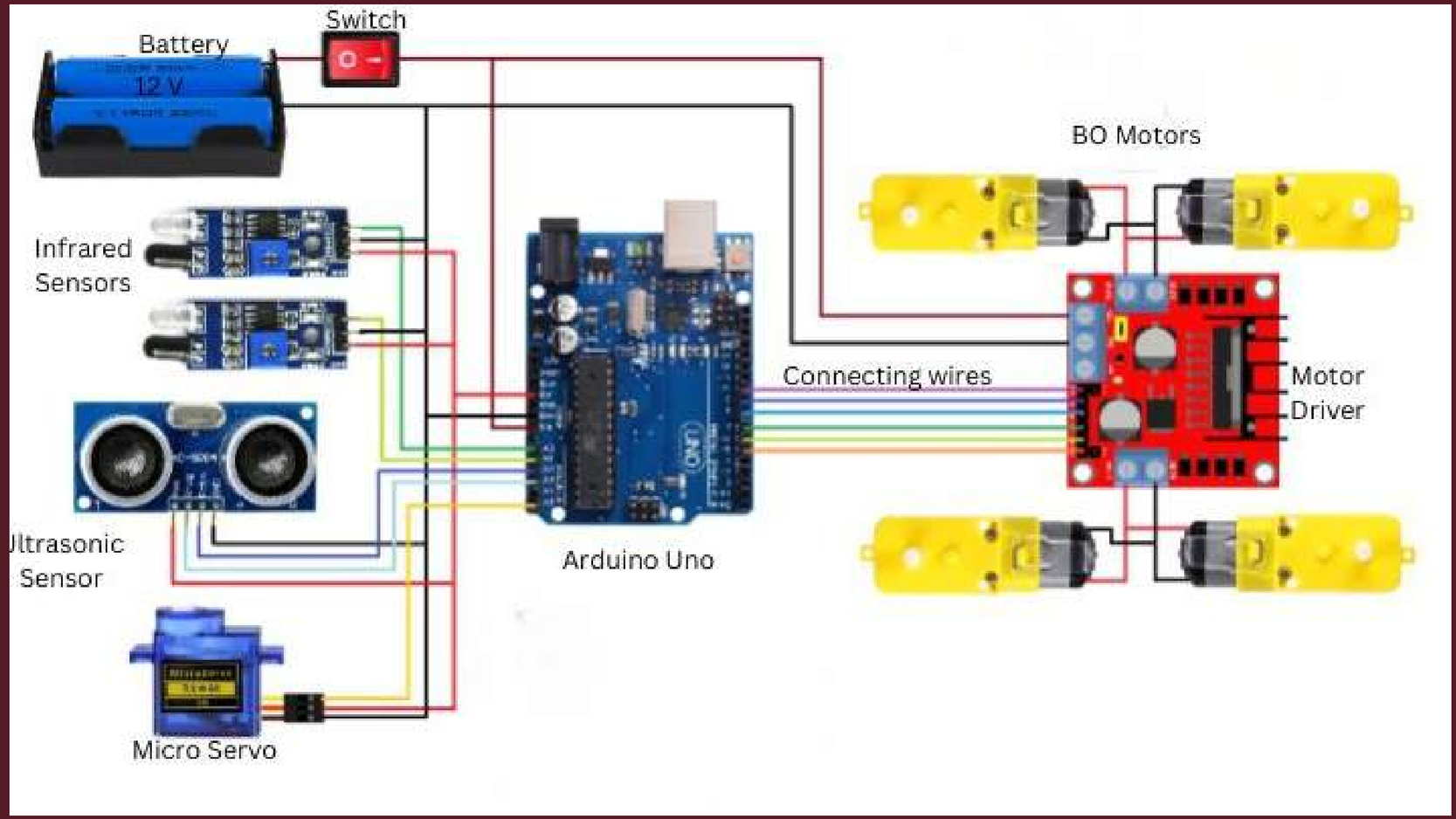
- **2 Digital IR Sensors Obstacle Avoidance Module (3 Pins)**
- Operating Voltage: 3.3-5V DC
- Digital Output: logic one (+3.5V DC) logic zero (0V DC)
- Analog Output: 0 to 1023 respective to 5v & 8-bit ADC.

- **HC-SR04 Ultrasonic Sensor**
- Working Voltage:5V(DC)
- Detection Distance:2cm-450cm
- Echo Signal :Output TTL PWL signal

- **Mini Servo Motor SG - 90 180° Rotation**
- Operating Voltage (VDC):3.0 ~ 7.2
- Operating Speed @6.6V:0.1sec/60°
- Stall Torque @ 4.8V (Kg-Cm):1.2

- **4*100 RPM BO Motor Double Shaft I Shape connected with wheels.**
- Shaft length:7 mm Motor
- Design: L-Shaped Shaft Diameter:5.5 mm
- Size:55x48x23 mm
- Diameter of wheel:65 mm
- Operating Voltage:3 to 12V
- **20cm Female-Female, Male-Female, Male-Male**
- **Ultrasonic Sensor Holder**
- **Acrylic Chassis**
- **9V HW Battery**

Circuit diagram of the Project



CODE

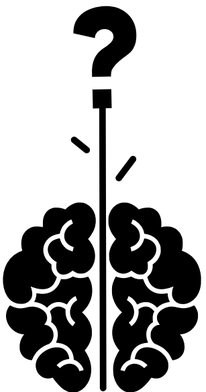
```
#define enA 10//Enable1 L298 Pin enA
#define in1 9 //Motor1  L298 Pin in1
#define in2 8 //Motor1  L298 Pin in1
#define in3 7 //Motor2  L298 Pin in1
#define in4 6 //Motor2  L298 Pin in1
#define enB 5 //Enable2 L298 Pin enB

#define L_S A0 //ir sensor Left
#define R_S A1 //ir sensor Right

#define echo A2    //Echo pin
#define trigger A3 //Trigger pin

#define servo A5

int Set=15;
int distance_L, distance_F, distance_R;

void setup(){

Serial.begin(9600); // start serial communication at 9600bps

pinMode(R_S, INPUT); // declare if sensor as input
pinMode(L_S, INPUT); // declare ir sensor as input

pinMode(echo, INPUT );// declare ultrasonic sensor Echo pin as input
pinMode(trigger, OUTPUT); // declare ultrasonic sensor Trigger pin as Output

pinMode(enA, OUTPUT); // declare as output for L298 Pin enA
pinMode(in1, OUTPUT); // declare as output for L298 Pin in1
pinMode(in2, OUTPUT); // declare as output for L298 Pin in2
pinMode(in3, OUTPUT); // declare as output for L298 Pin in3
pinMode(in4, OUTPUT); // declare as output for L298 Pin in4
pinMode(enB, OUTPUT); // declare as output for L298 Pin enB

analogWrite(enA, 100); // Write The Duty Cycle 0 to 255 Enable Pin A for Motor1 Speed
analogWrite(enB, 100); // Write The Duty Cycle 0 to 255 Enable Pin B for Motor2 Speed
void loop(){
//==========================================
//     Line Follower and Obstacle Avoiding
//==========================================

distance_F = Ultrasonic_read();
Serial.print("D F=");Serial.println(distance_F);


//if Right Sensor and Left Sensor are at White color then it will call forward function
 if((digitalRead(R_S) == 0)&&(digitalRead(L_S) == 0)){
  if(distance_F > Set){forward();}
  else{check_side();}
 }

//if Right Sensor is Black and Left Sensor is White then it will call turn Right function
else if((digitalRead(R_S) == 1)&&(digitalRead(L_S) == 0)){turnRight();}

//if Right Sensor is White and Left Sensor is Black then it will call turn Left function
else if((digitalRead(R_S) == 0)&&(digitalRead(L_S) == 1)){turnLeft();}
```

```
//*********************Ultrasonic_read*
long Ultrasonic_read(){
  digitalWrite(trigger, LOW);
  delayMicroseconds(2);
  digitalWrite(trigger, HIGH);
  delayMicroseconds(10);
  long time = pulseIn (echo, HIGH);
  return time / 29 / 2;
}

void compareDistance(){
    if(distance_L > distance_R){
  turnLeft();
  delay(500);
  forward();
  delay(600);
  turnRight();
  delay(500);
  forward();
  delay(600);
  turnRight();
  delay(400);
  }
  else{
  turnRight();
  delay(500);
  forward();
  delay(600);
  turnLeft();
  delay(500);
  forward();
  delay(600);
  turnLeft();
  delay(400);
  }
}
```

```
void Check_side(){
    Stop();
    delay(100);
 for (int angle = 70; angle <= 140; angle += 5)  {
    servoPulse(servo, angle);  }
    delay(300);
    distance_R = Ultrasonic_read();
    Serial.print("D R=");Serial.println(distance_R);
    delay(100);
  for (int angle = 140; angle >= 0; angle -= 5)  {
    servoPulse(servo, angle);  }
    delay(500);
    distance_L = Ultrasonic_read();
    Serial.print("D L=");Serial.println(distance_L);
    delay(100);
 for (int angle = 0; angle <= 70; angle += 5)  {
    servoPulse(servo, angle);  }
    delay(300);
    compareDistance();
}
void forward(){  //forward
digitalWrite(in1, LOW); //Left Motor backward Pin
digitalWrite(in2, HIGH); //Left Motor forward Pin
digitalWrite(in3, HIGH); //Right Motor forward Pin
digitalWrite(in4, LOW); //Right Motor backward Pin
}

void backward(){ //backward
digitalWrite(in1, HIGH); //Left Motor backward Pin
digitalWrite(in2, LOW); //Left Motor forward Pin
digitalWrite(in3, LOW); //Right Motor forward Pin
digitalWrite(in4, HIGH); //Right Motor backward Pin
}

void turnRight(){ //turnRight
digitalWrite(in1, LOW); //Left Motor backward Pin
digitalWrite(in2, HIGH); //Left Motor forward Pin
digitalWrite(in3, LOW); //Right Motor forward Pin
digitalWrite(in4, HIGH); //Right Motor backward Pin
}

void turnLeft(){ //turnLeft
digitalWrite(in1, HIGH); //Left Motor backward Pin
digitalWrite(in2, LOW); //Left Motor forward Pin
digitalWrite(in3, HIGH); //Right Motor forward Pin
digitalWrite(in4, LOW); //Right Motor backward Pin
}

void Stop(){ //stop
digitalWrite(in1, LOW); //Left Motor backward Pin
digitalWrite(in2, LOW); //Left Motor forward Pin
digitalWrite(in3, LOW); //Right Motor forward Pin
digitalWrite(in4, LOW); //Right Motor backward Pin
}
```

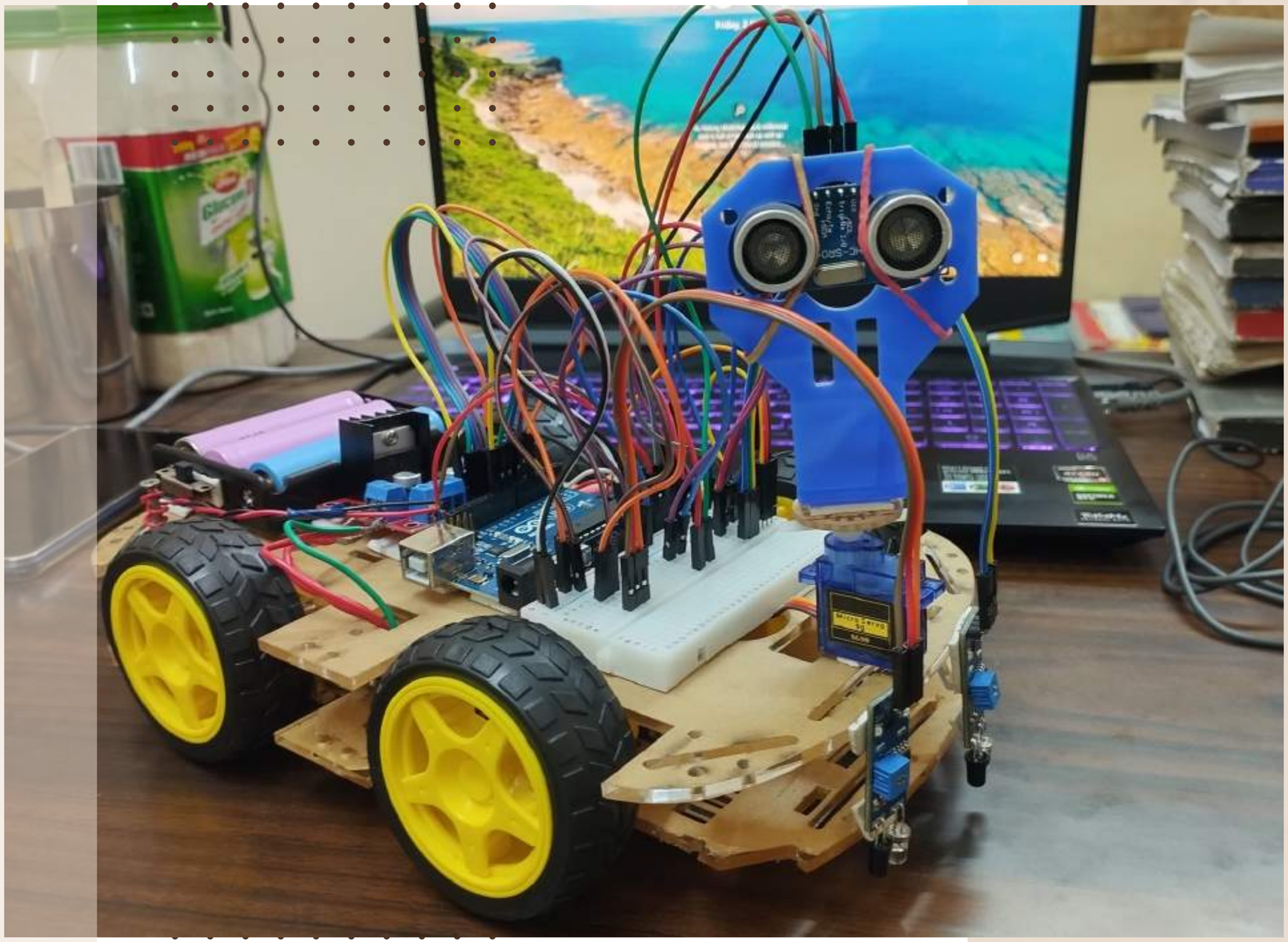
CHALLENGES FACED AND SOLUTIONS FOUND

DURING THE DIY PROJECT OF THE ARDUINO OBSTACLE AVOIDANCE LINE FOLLOWER ROBOT, WE FACED CHALLENGES IN SENSOR INTEGRATION AND MOTOR CONTROL. THROUGH METICULOUS TESTING AND CALIBRATION, WE ACHIEVED ACCURATE SENSOR READINGS. ITERATIVE ADJUSTMENTS AND RE-TESTING THE CODE WITH DIFFERENT VALUES ENABLED US TO OPTIMIZE MOTOR CONTROL FOR SMOOTHER NAVIGATION. WE FACED THE PROBLEM OF POWERSUPPLY FROM 9V BATTERY WHICH WAS SOLVED BY USING 12V BATTERY.



TEAM:12

PHOTO OF FINAL SYSTEM



LEARNING'S FROM THE PROJECT

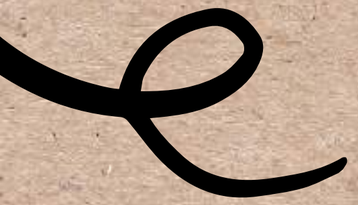
THROUGHOUT THE PROCESS OF CREATING THE ARDUINO OBSTACLE AVOIDANCE LINE FOLLOWER ROBOT, OUR TEAM LEARNED INVALUABLE LESSONS. WE GAINED A DEEP UNDERSTANDING OF CODING, ELECTRONICS, AND ROBOTICS PRINCIPLES. WE ENCOUNTERED AND OVERCAME NUMEROUS CHALLENGES, ENHANCING OUR PROBLEM-SOLVING AND TROUBLESHOOTING ABILITIES. COLLABORATION WAS KEY AS WE WORKED TOGETHER, POOLING OUR STRENGTHS AND IDEAS TO ACHIEVE SUCCESS. PATIENCE AND PERSISTENCE BECAME VIRTUES AS WE FACED SETBACKS AND SETBACKS, CONTINUALLY ITERATING AND IMPROVING OUR DESIGN. MOREOVER, THIS PROJECT FOSTERED A PASSION FOR INNOVATION AND SPARKED OUR CURIOSITY FOR FURTHER EXPLORATION IN THE REALM OF ROBOTICS. OVERALL, THE EXPERIENCE ENRICHED OUR KNOWLEDGE, SKILLS, AND PASSION FOR DIY PROJECTS AND TECHNOLOGY.

CONCLUSION

In conclusion, the journey of our team in creating the Arduino Obstacle Avoiding Line Follower Robothas been both challenging and rewarding. Through countless hours of research, coding, and debugging, we have successfully built a functional and versatile robot that can navigate its surroundings and follow designated paths. This DIY project has not only honed our technical skills in robotics and programming but has also fostered teamwork, problem-solving, and innovation. As we reflect on this project, we are proud of the knowledge gained, the obstacles overcome, and the excitement of witnessing our creation in action. This experience has truly sparked our passion for robotics and opened doors to endless possibilities for future projects.

References:

- Guide to Make Arduino Obstacle Avoidance Line Follower Robot
- Scratch Code



YouTube link for video demonstration of the project

THANK YOU

TEAM:12

Any questions?

yes!

Let's work together