INFORMATION TECHNOLOGY
UNIVERSITY OF THE PUNJAB

DEPARTMENT OF COMPUTER ENGINEERING

ELECTRONICS WORK BENCH LAB

LAB PROJECT

LINE FOLLOWING AND OBSTACLE AVOIDING ROBOT

Submitted by:

Tayyab Akram	BSCE-23055
Ashba Khaliq	BSCE-23027
Minahil Imran	BSCE-23030

Hardware Components:

- **Microcontroller/Processor:**

  Arduino UNO3 microcontroller processor.

- **Motor Drivers:**

  Motor drivers to control the speed and direction of the

  motors.

- **DC Motors:**

  Geared DC motors for the wheels, allowing precise control of movement.

- **Chassis:**

  The physical structure of the robot that holds all components together.

- **Wheels:**

  Suitable wheels for smooth movement on the surface.

- **Line Sensor :**

  Infrared (IR) sensors or other types of line sensors to detect the line on the ground.

- **Obstacle Detection Sensors:**

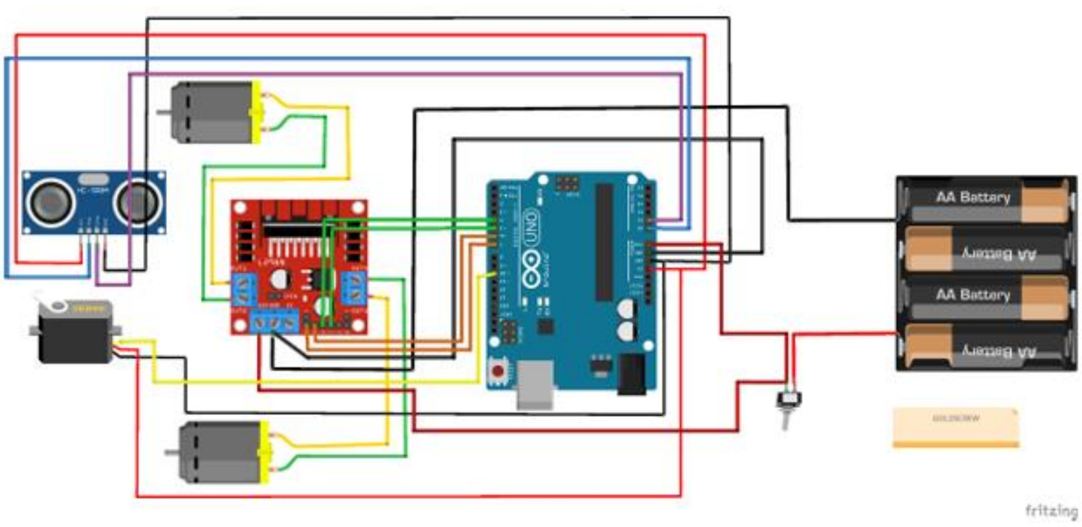
  Ultrasonic sensors or infrared sensors to detect

  obstacles in the robot's path .

- **Chassis:**
  The physical structure of the robot that holds all components together.

- **Wires:**
  - Wires to establish electrical connections between components.

Schematic Diagram:



Schematic Diagram

Program:

```
// Define the pins for the motor controller module

int in1 = 4;

int in2 = 3;

int in3 = 2;

int in4 = 1;

int ENA = 6;

int ENB = 5;

int right=A0;

int left=A1;

int button=12;

int led=13;
int trig = 8;
int echo = 9;
long distance;
void setup()
{
    pinMode(right, INPUT );
    pinMode(left, INPUT );
    pinMode(button, INPUT_PULLUP);
    pinMode(led, OUTPUT);
    pinMode(in1, OUTPUT);
    pinMode(in2, OUTPUT);
    pinMode(in3, OUTPUT);
    pinMode(in4, OUTPUT);
    pinMode(ENA, OUTPUT);
    pinMode(ENB, OUTPUT);
    pinMode(echo,INPUT);
    pinMode(trig, OUTPUT);
    Serial.begin(115200);
}
void loop()
{
        int state=digitalRead(button);
        if(state==0){
            digitalWrite(led, HIGH);
            if(digitalRead(right)==1 && digitalRead(left)==1)
    {
        digitalWrite(in1, LOW);
        digitalWrite(in2, LOW);
        digitalWrite(in3, LOW);
        digitalWrite(in4, LOW);
        digitalWrite(ENA, LOW);
        digitalWrite(ENB, LOW);
        delay(70);
    }
    else if(digitalRead(right)==1 && digitalRead(left)==0)
    {
        digitalWrite(in1, LOW);
        digitalWrite(in2, LOW);
        digitalWrite(in3, LOW);
        digitalWrite(in4, LOW);
        turn_left(80);
        delay(70);
    }
    else if(digitalRead(left)==1 && digitalRead(right)==0)
    {
        digitalWrite(in1, LOW);
        digitalWrite(in2, LOW);
        digitalWrite(in3, LOW);
        digitalWrite(in4, LOW);
        turn_right(80);
        delay(70);
    }
    else if(digitalRead(right)==0 && digitalRead(left)==0)
    {
        digitalWrite(in1, LOW);
        digitalWrite(in2, LOW);
        digitalWrite(in3, LOW);
        digitalWrite(in4, LOW);
        run_forward(80);
        delay(70);
          }      }
    else{
            digitalWrite(led, LOW);
            digitalWrite(ENA, LOW);
            digitalWrite(ENB, LOW);
            distance = measure();
  Serial.println(distance);
  if(distance<30){
    delay(1000);
    turn_right(100);
    delay(500);
  }
```

```
  else{
    run_forward(80);
    delay(70);
  }
}
}
long measure(){
  digitalWrite(trig, LOW);
  delay(10);
  digitalWrite(trig, HIGH);
  delay(10);
  digitalWrite(trig, LOW);
  long duration = pulseIn (echo, HIGH);
  return duration / 29 / 2;
}
void run_forward(int fwd_speed)
{
  digitalWrite(in1, LOW);
  digitalWrite(in2, HIGH);
  digitalWrite(in3, LOW);
  digitalWrite(in4, HIGH);
  analogWrite(ENA, fwd_speed);
  analogWrite(ENB, fwd_speed*2);
}
void run_back(int back_speed)
{
  digitalWrite(in1, HIGH);
  digitalWrite(in2, LOW);
  digitalWrite(in3, HIGH);
  digitalWrite(in4, LOW);
  analogWrite(ENB, back_speed/4);
  analogWrite(ENA, back_speed*2);
}
void turn_left(int left_speed)
{
  digitalWrite(in1, LOW);
  digitalWrite(in2, HIGH);
  digitalWrite(in3, LOW);
  digitalWrite(in4, HIGH);
  analogWrite(ENA, left_speed/4);
  analogWrite(ENB, 0);
}
void turn_right(int right_speed)
{
  digitalWrite(in1, LOW);
  digitalWrite(in2, HIGH);
  digitalWrite(in3, LOW);
  digitalWrite(in4, HIGH);
  analogWrite(ENA, 0);
  analogWrite(ENB, right_speed*2);
}
```

Report Outline:

1. **Functionality:**

    - Line Following

- The robot is equipped with sensors, typically infrared (IR) sensors, positioned on the upside of the robot. These sensors detect the contrast between the line and the surrounding surface. The robot's microcontroller receives input from the line sensors, determining whether the robot is centered on the line, to the left, or to the right. Based on the feedback from the line sensors, the microcontroller adjusts the speed and direction of the motors. For example, if the robot detects the line to the right, it may increase the speed of the left motor to steer the robot back to the center of the line.

    - Obstacle Avoiding

The robot is equipped with obstacle detection sensors, such as ultrasonic sensors or infrared sensors, to identify obstacles in its path. The sensors measure the distance between the robot and the obstacle. The microcontroller processes the distance data and decides whether the robot needs to take evasive action to avoid the obstacle. The microcontroller processes the distance data and decides whether the robot needs to take evasive action to avoid the obstacle.

- Over All Functionality

In this project, when we turn on the button the robot works as line following robot and when we turn off the button the robot works as obstacle avoiding robot.