Microcontroller & Embedded Systems Project

Department Of Mechatronics Engineering

Spring-2024

Name: Syed Suleman Azher, Amir Shahzad, Shahroz Ali Malik

Roll ID: 221720 221747 221754

Subject: Microcontroller & Embedded Systems

Supervised by: Sir Umer Farooq, Ma'am Sadia Saeed

Objectives:

- To understand the mechanism of Line Following Robot.
- To fabricate a PCB to provide all connections for the robot.
- To have a know how of how to interface LCD/OLED, as well as role of Current & Voltage Sensor and SD Card.
- To design an Arduino Code to help our robot function.
- To learn the basic automation robot as a beginner.

Abstract:

Line follower robots are autonomous vehicles designed to navigate predefined paths by tracking lines on the ground. This project aims to develop a line follower robot capable of following a black line on a white surface using infrared sensors for line detection. The robot employs a closed-loop control system, integrating sensor feedback to adjust its movements in real-time. The hardware setup includes an Arduino microcontroller for processing sensor data and controlling motor actions. Through iterative testing and optimization, the robot demonstrates efficient line following behavior, making it suitable for applications in automation, robotics education, and entertainment.

Introduction:

The Line Follower Robot is the most famous robotic design which is being designed in order to follow the visual black line or path in the surface. This type of robot is most commonly used in many universities as the basic introduction to robotics and programming. It is mostly demonstrated using any AVR microcontroller board rather it be Arduino UNO, Arduino Mega 2560, Arduino Nano, and many other microcontroller boards. Just one

click of code uploading and the whole new spectacular sight watching your robot following the black path. This robot can undergo further enhancements just like current sensor, voltage sensor, SD Card module and much more which will be discussed later on. The Ultrasonic sensor play the vital role for robot to detect the black line and follow the path accordingly.

Deliverables:

The following project will demonstrate the first step towards the automation process and obstacle avoidance. And by using the particular sensors, we will be able to display the voltage and current on OLED LCD. Furthermore, it will be able to receive the data from SD Card Module and store it in itself. The results will be shown in the OLED Display. The robot will run on battery for at least 20 minutes using a Li-Po ion battery. This will further follow the interrupt button which will be able to reset the whole mechanism of robot with only one click. The ESP-32 Wifi and Bluetooth Module will be helpful in using wifi and receiving signals for robot to operate. The color sensor will be much more useful in detecting the color in front of it and to display the color being detected by the sensor.

Applications:

Line follower robots are designed to follow a predefined path, typically marked by a line on the ground. They have a wide range of applications across various fields. Here are some notable examples:

- Material Handling: Line follower robots can be used to transport materials within a manufacturing facility, ensuring efficient and reliable delivery of components or finished products.

- Assembly Line: They can follow specific paths to deliver parts and tools to workers on an assembly line, optimizing workflow and reducing downtime.

- Crop Monitoring: Line follower robots can be used to navigate through fields to monitor crops, collect data, or assist in planting and harvesting.
- Precision Farming: They can follow specific routes to apply fertilizers, pesticides, or water to crops, ensuring precise and efficient use of resources.
- Amusement Parks: Line follower robots can be part of interactive exhibits or rides, enhancing the entertainment experience for visitors.
- Thematic Displays: They can be used in thematic displays or parades, following a programmed path to create dynamic and engaging shows.

Further Enhancements:

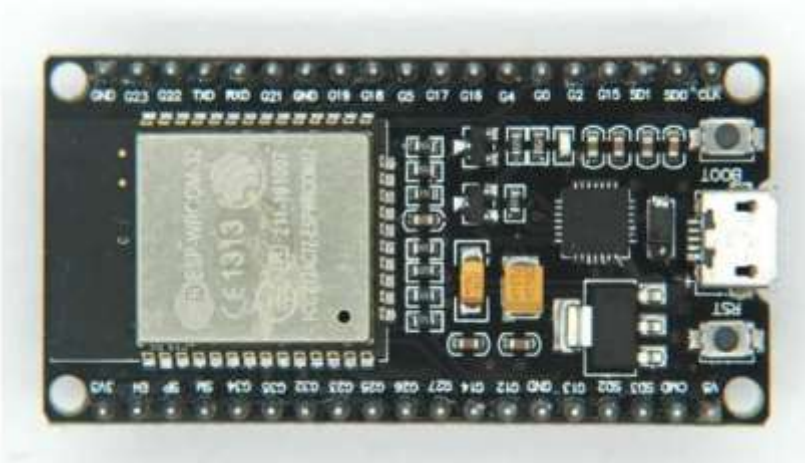
Voltage Sensor:

Current Sensor:

SD Card Module:

Color Sensor:

ESP-32 Wifi Module:



Interrupt Button:

Contributions:

Group Members	Contribution
1. Syed Suleman Azher	PCB designing & Arduino code
2. Amir Shahzad	Ironing, Soldering, Chassis Assembly
3. Shahroz Ali Malik	Etching, Chassis Assembly, Testing

Me & my group members worked tirelessly for the designing of LFR and to enhance it as much as possible. Here are few details for what each group member had contributed with:

Syed Suleman Azher (Me):

My most stronger area is in software side. Means I have expertise in PCB designing, software simulation reports, Arduino coding and much more. The PCB designing took about 4 to 5 days and to think tank on the components to be added and where the output of those components being declared. Then the coding was also a hassle as to ponder upon how to declare the functions and how to make project components work according to the code.

Amir Shahzad:

He has expertise in hard skill and to manage connections of the PCB board as well as assembling the chassis kit. He managed the PCB ironing,

soldering and components fixture in the PCB board. He had the skill to assemble each and every component in any possible way.

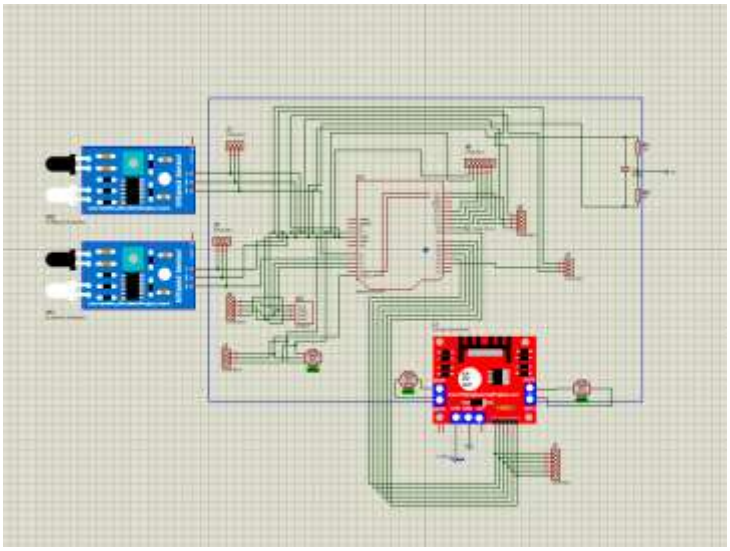
Shahroz Ali Malik:

He also has hard skill expertise and also his testing skills are outstanding. He helped in etching the PCB board as well as he gave assistance to Amir in assembling the chassis. He managed to provide all the connection the components of LFR and to test the robot.

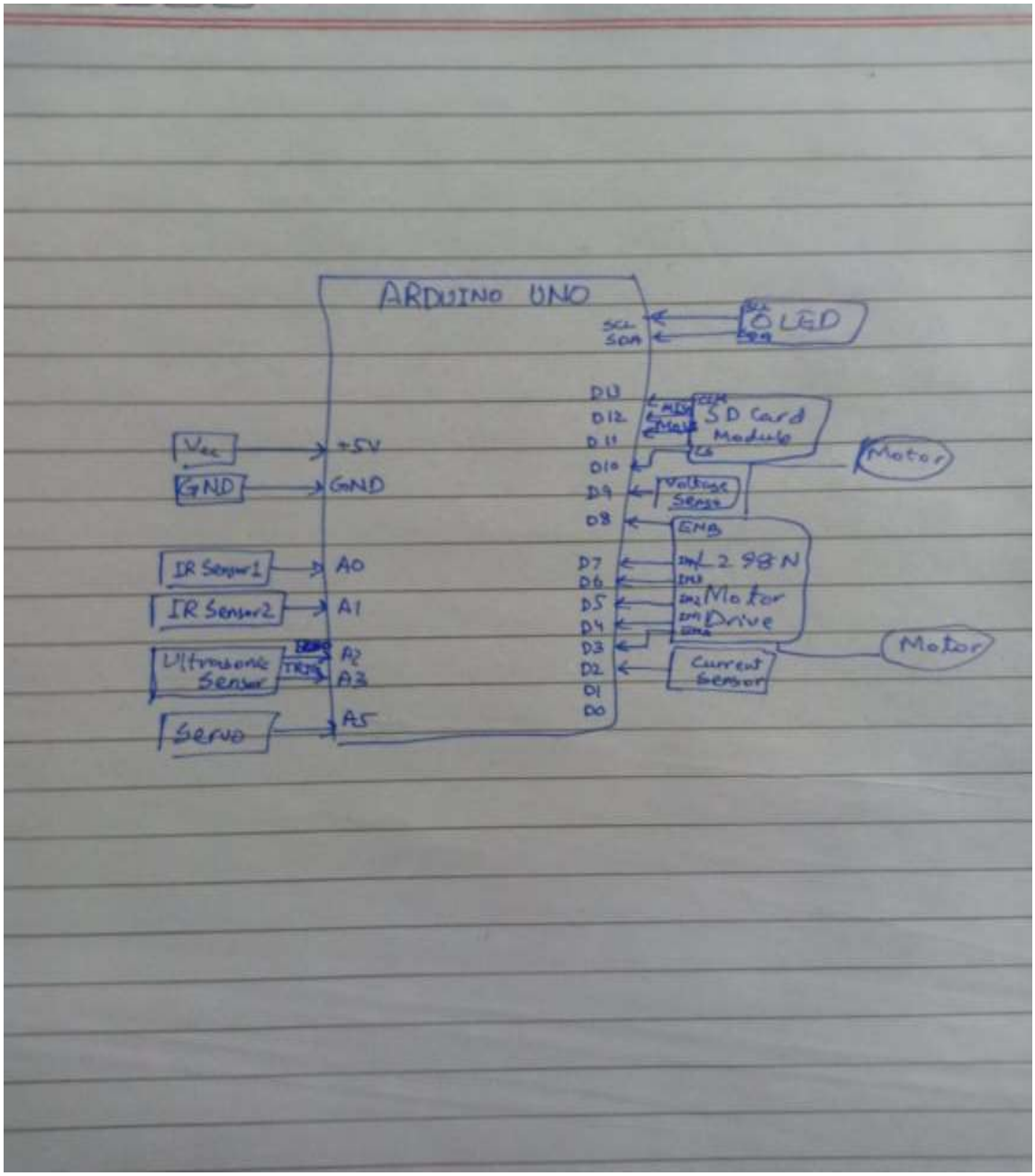
PCB Fabrication Steps:

1. First Design the PCB layout
2. Perform Ironing on the Copper board until all traces are being printed on the copper.
3. Etch the board so that unwanted copper from the board
4. Adjust components according to the size of holes. This depends on the holes sizes from the PCB layout
5. Solder all the components
6. Then test if any issues detected

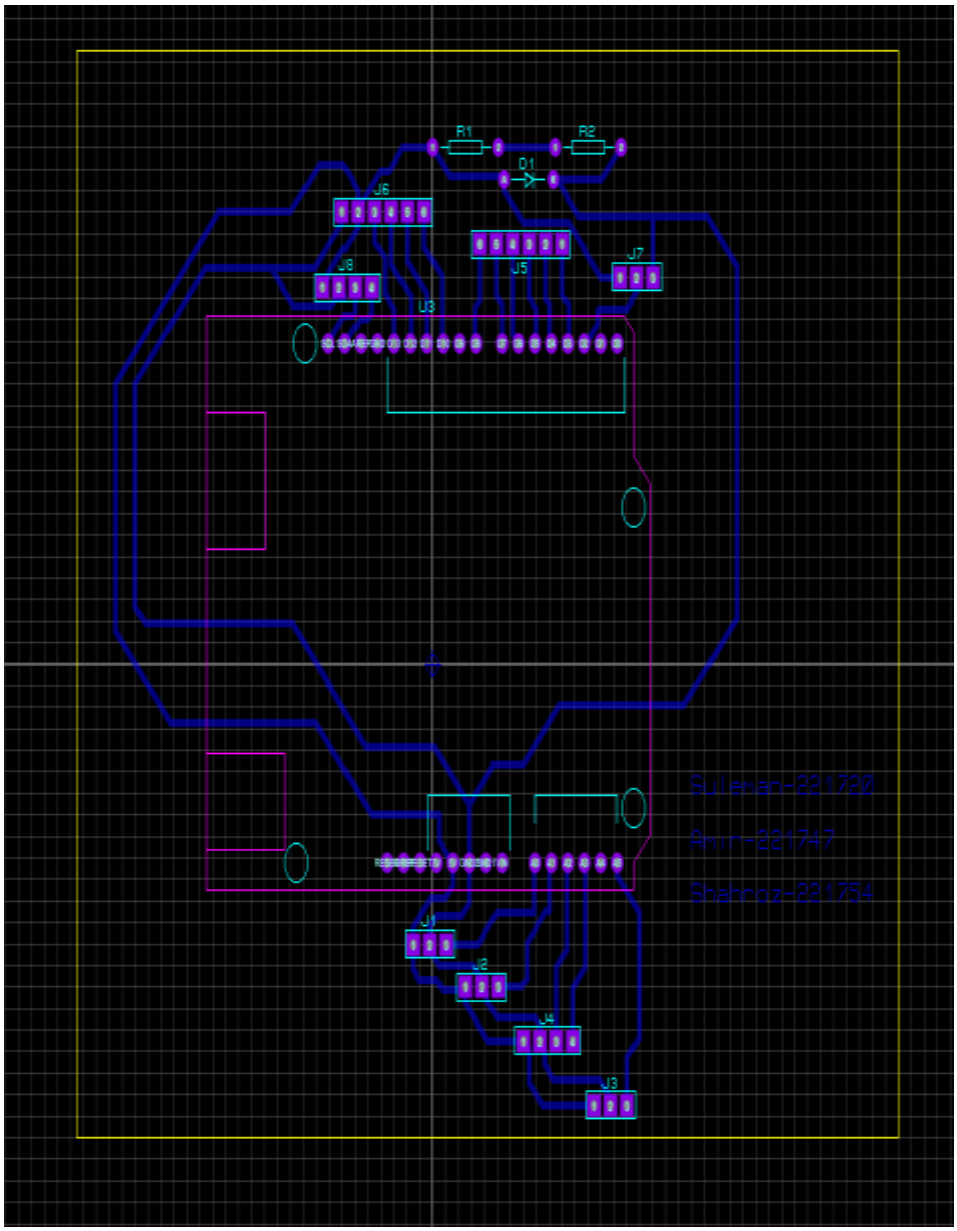
Schematic:



Block Diagram:



PCB Layout:



Some Important Links:

- https://github.com/suli4567-tech/lfr-pro
- https://youtu.be/6wWcvakiAXI

- https://www.linkedin.com/feed/update/urn:li:activity:7206174819313221632/

Project:

Arduino Code:

```
#include <SD.h>
#include <NewPing.h>
#include <Wire.h>
#include "U8glib.h"

#define IR_SENSOR_RIGHT A0
```

```
#define IR_SENSOR_LEFT A1
#define ULTRASONIC_TRIGGER_PIN 3
#define ULTRASONIC_ECHO_PIN 2
#define MOTOR_SPEED 180


// Reset pin not used
U8GLIB_SSD1306_128X64 u8g(10, 9);

File logfile;
NewPing sonar(ULTRASONIC_TRIGGER_PIN,
ULTRASONIC_ECHO_PIN);

//Right motor
int enableRightMotor=6;
int rightMotorPin1=7;
int rightMotorPin2=8;

//Left motor
int enableLeftMotor=5;
int leftMotorPin1=9;
int leftMotorPin2=10;
// Voltage sensor pin
int voltageSensorPin = A2;

// Current sensor pin
int currentSensorPin = A3;

void setup()
{
  //The problem with TT gear motors is that, at very low
pwm value it does not even rotate.
```

```
  //If we increase the PWM value then it rotates faster
and our robot is not controlled in that speed and goes
out of line.
  //For that we need to increase the frequency of
analogWrite.
  //Below line is important to change the frequency of
PWM signal on pin D5 and D6
  //Because of this, motor runs in controlled manner
(lower speed) at high PWM value.
  //This sets frequecny as 7812.5 hz.
  TCCR0B = TCCR0B & B11111000 | B00000010 ;

  // put your setup code here, to run once:
  pinMode(enableRightMotor, OUTPUT);
  pinMode(rightMotorPin1, OUTPUT);
  pinMode(rightMotorPin2, OUTPUT);

  pinMode(enableLeftMotor, OUTPUT);
  pinMode(leftMotorPin1, OUTPUT);
  pinMode(leftMotorPin2, OUTPUT);

  pinMode(IR_SENSOR_RIGHT, INPUT);
  pinMode(IR_SENSOR_LEFT, INPUT);
   pinMode(ULTRASONIC_TRIGGER_PIN, OUTPUT);
  pinMode(ULTRASONIC_ECHO_PIN, INPUT);

Serial.begin(9600);

  // Initialize SD card
  if (!SD.begin(10))
  {
    Serial.println("SD Card initialization failed!");
```

```
    return;
  }

  // Open or create the log file
  logfile = SD.open("log.txt", FILE_WRITE);
  if (!logfile)
  {
    Serial.println("Error opening log file!");
    return;
  }
u8g.begin();

  rotateMotor(0,0);

}


void loop()
{
  // Read voltage sensor value
  float voltage = analogRead(voltageSensorPin) * (5.0 /
1023.0);

  // Read current sensor value
  float current = analogRead(currentSensorPin) * (5.0 /
1023.0);

  // Display voltage and current on OLED
  u8g.firstPage();
  do {
    // Draw voltage and current values
    u8g.setFont(u8g_font_6x10);
```

```
    u8g.drawStr(0, 10, "Voltage: ");
    char voltageStr[10];
    dtostrf(voltage, 4, 2, voltageStr); // Convert float
to string
    u8g.drawStr(70, 10, voltageStr);

    u8g.drawStr(0, 20, "Current: ");
    char currentStr[10];
    dtostrf(current, 4, 2, currentStr); // Convert float
to string
    u8g.drawStr(70, 20, currentStr);
  } while(u8g.nextPage());
  int rightIRSensorValue = digitalRead(IR_SENSOR_RIGHT);
  int leftIRSensorValue = digitalRead(IR_SENSOR_LEFT);

   // 20 is distance from obstacle
  if (sonar.ping_cm() < 20)
  {
    rotateMotor(-MOTOR_SPEED, MOTOR_SPEED); // Rotate
right
    delay(1000); // Rotate for 1 second
  }
  else
  {
    // If none of the sensors detect a line, then go
straight
    if (rightIRSensorValue == LOW && leftIRSensorValue
== LOW)
    {
      rotateMotor(MOTOR_SPEED, MOTOR_SPEED);

    }
```

```
    // If right sensor detects a line, then turn right
    else if (rightIRSensorValue == HIGH &&
leftIRSensorValue == LOW)
    {
      rotateMotor(-MOTOR_SPEED, MOTOR_SPEED);

    }
    // If left sensor detects a line, then turn left
    else if (rightIRSensorValue == LOW &&
leftIRSensorValue == HIGH)
    {
      rotateMotor(MOTOR_SPEED, -MOTOR_SPEED);

    }
    // If both sensors detect a line, then stop
    else
    {
      rotateMotor(0, 0);

    }
  }
}


void rotateMotor(int rightMotorSpeed, int
leftMotorSpeed)
{

  if (rightMotorSpeed < 0)
  {
    digitalWrite(rightMotorPin1,LOW);
    digitalWrite(rightMotorPin2,HIGH);
```

```
  }
  else if (rightMotorSpeed > 0)
  {
    digitalWrite(rightMotorPin1,HIGH);
    digitalWrite(rightMotorPin2,LOW);
  }
  else
  {
    digitalWrite(rightMotorPin1,LOW);
    digitalWrite(rightMotorPin2,LOW);
  }

  if (leftMotorSpeed < 0)
  {
    digitalWrite(leftMotorPin1,LOW);
    digitalWrite(leftMotorPin2,HIGH);
  }
  else if (leftMotorSpeed > 0)
  {
    digitalWrite(leftMotorPin1,HIGH);
    digitalWrite(leftMotorPin2,LOW);
  }
  else
  {
    digitalWrite(leftMotorPin1,LOW);
    digitalWrite(leftMotorPin2,LOW);
  }
  analogWrite(enableRightMotor, abs(rightMotorSpeed));
  analogWrite(enableLeftMotor, abs(leftMotorSpeed));
}
```

Problems Faced:

The PCB of this project involved the jumper wires connections which made the project look messy. And also the wires connections we forgot to connect were being soldered afterwards cautiously, so that the circuit short might be prevented. Also the supply to some of our components was missing which resulted in LFR a bit malfunctioning of LFR. Also the batteries required a lot of charging which was a lot of time waste to charge them and LFR also did not function properly due to them.

Conclusion:

This project helped us to demonstrate the basic automation to follow the line and obstacle avoider. And we also wrote the basic code involving the following of line robot. In addition multiple sensors were used to enhance the functionality of the line follower. Also to display the voltage and current on the OLED display to indicate the voltages and currents the robot was working on.