REHBER

Group Members:

Muhammad Hamid (**221750**)

Raja Muhammad Anas (**221705**)

Aman Johar (**221675**)

BE MECHATRONICS (Session F22-B)

Project Supervisor

Engr. Umer Farooq

Assistant Professor

Engr. Sadia Saeed

Lab Engineer

DEPARTMENT OF MECHATRONICS ENGINEERING

FACULTY OF ENGINEERING

AIR UNIVERSITY, ISLAMABAD

Policy on Professional Ethics & Plagiarism

You are free to consult any book and online resources during the design and analysis phase, but you could not copy from them. Your design and implementation must be your original effort and the same applies to the source code. Remember that if anyone is found to copy from the Internet or other group members, the group shall face severe penalty. You are not allowed to copy any material or code directly from the web or elsewhere. Note that if you are found to violate this policy or it becomes obvious that the work you have submitted is not your own or has been taken from some other source you shall be facing severe

Plagiarism will be tested through software and it must be less than 20 %

consequences

Team Member Name	Role	Word count that each has written in assigned color code in report	Signature
Team Lead	221750		
Project Manager	221705		
Technical	221750		
Integrator and Tester	221675		

Chapter 1- Preliminaries

1.1 Proposal

Abstract

Our project aims to design and build a smart robot that combines various sensors and microcontroller technologies. The robot will be capable of following lines, detecting obstacles using ultrasonic sensors, communicating with an online server, monitoring voltage levels, identifying colors, and storing data on an SD card. This proposal outlines the key components, objectives, and implementation plan for our project.

Objectives

1. **Line Following**: Develop algorithms and control mechanisms to enable the robot to follow predefined paths or lines.
2. **Ultrasonic Obstacle Detection**: Implement ultrasonic sensors to detect obstacles and adjust the robot's path accordingly.
3. **ESP32 Communication**: Establish communication between the robot and an online server using the ESP32 module.
4. **Voltage Monitoring**: Integrate a voltage sensor to monitor the robot's power supply.
5. **Color Sensing**: Utilize a color sensor to identify and respond to different colors.
6. **Data Logging**: Store relevant data (such as sensor readings, path information, and obstacle encounters) on an SD card.

1.2 Initial feasibility

1 **Hardware Components and Availability**:

Assess the availability and cost of the required components (ESP32, ultrasonic sensors, color sensor, motor drivers, wheels, etc.).

Ensure that these components can be easily sourced or purchased within our budget.

2 **Technical Challenges**:

Evaluate the complexity of implementing line-following algorithms, obstacle detection, and color sensing.

Consider the processing power and memory limitations of the ESP32 for handling multiple tasks simultaneously.

3 **Power Supply and Battery Life**:

Calculate the power requirements for all components (motors, sensors, ESP32, etc.).

Choose an appropriate power supply (battery or external source) and estimate the robot's runtime.

4 **Communication and Networking**:

Investigate the feasibility of establishing Wi-Fi communication with an online server using the ESP32.

Ensure that the server infrastructure (hosting, security, etc.) aligns with our project goals.

5 **Sensor Integration**:

Verify compatibility between the sensors (ultrasonic, color, voltage) and the ESP32.

Consider any potential interference or limitations when using multiple sensors simultaneously.

6 **Software Development**:

Plan the software architecture, including real-time control loops, sensor data processing, and decision-making logic.

Explore existing libraries or frameworks that can simplify development.

7 **Testing and Iteration**:

Allocate time for testing and debugging.

Be prepared for iterative improvements based on testing results.

7.1 Technical Standards

Speed: 1.5m/s

Safety: Robot is harmless as per mechanical standards but battery can be explosive if handled without care. Electrical voltages and current is pretty low so hazard of shock is low.

7.2 Team Roles & Details

Muhammad Hamid:
Report, Research, Circuit Design.

Raja Muhammad Anas:
Etching, Soldering, Code

Amna Johar:

Testing, Tweaking, Simulation, Code, Error Correcting in Schematic.

7.3

Gantt Chart

Task	Start Date	Duration	Dependencies
Project Kickoff	April 25, 2024	1 day	None
Research	April 26, 2024	5 days	None
Component Procure	May 2, 2024	2 days	Research
Hardware Assembly	May 4, 2024	3 days	Component Procure
Line Following Alg	May 7, 2024	5 days	Hardware Assembly
Ultrasonic Setup	May 12, 2024	4 days	Hardware Assembly
ESP32 Integration	May 16, 2024	3 days	Hardware Assembly
Voltage Monitoring	May 19, 2024	2 days	Hardware Assembly
Color Sensing	May 20, 2024	1 day	Voltage Monitoring
SD Card Logging	May 20, 2024	1 day	Color Sensing
Testing & Debug	May 20, 2024	1 day	SD Card Logging
Documentation	May 20, 2024	1 day	Testing & Debug
Final Presentation	May 20, 2024	1 day	Documentation

7.4 Estimated budgets

Estimated Budget was 7k, including everything.

Chapter 2-Project Conception

2.1 List of features and operational specification of your project (Preliminary Product Specification)

- **Line Following**
- **Obstacle Avoiding**
- **Color Sensing**
- **Uploading Values to Server**
- **Path memory**

2.2 Basic block diagrams of whole system

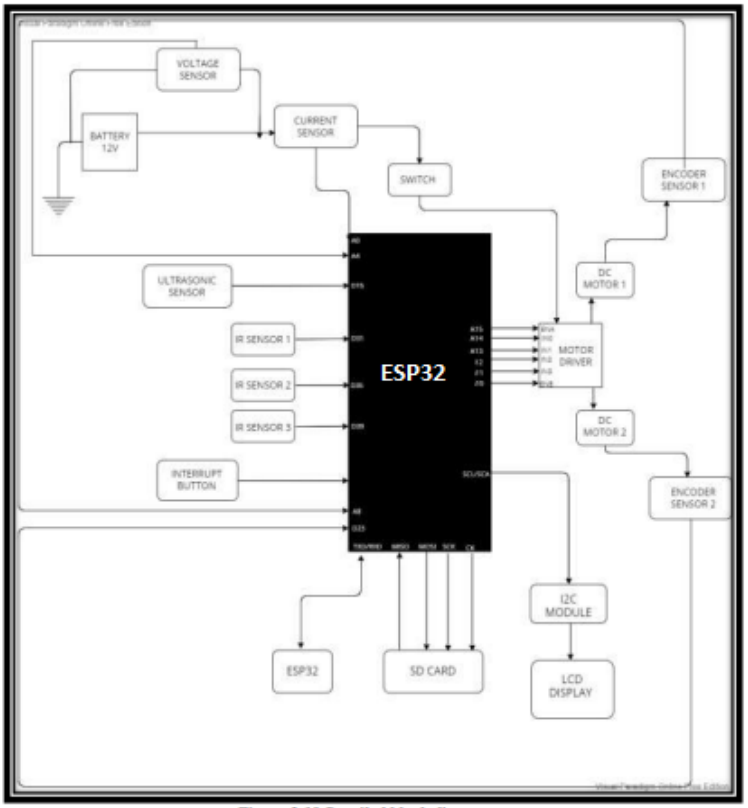
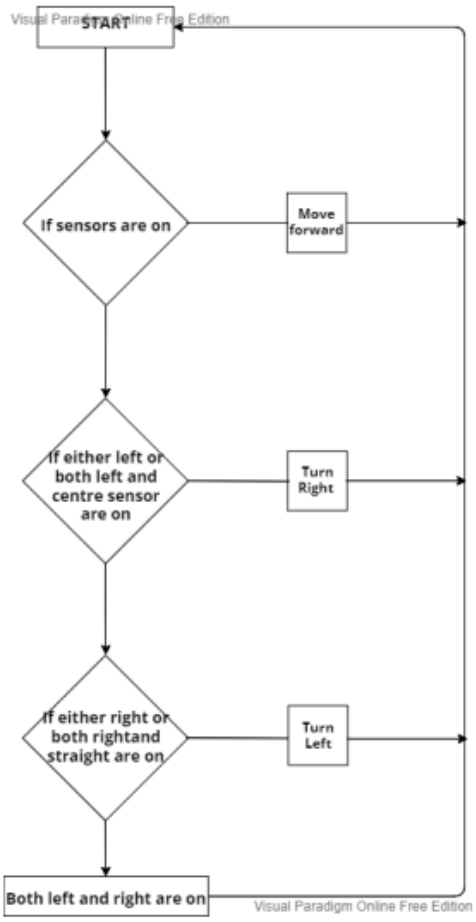
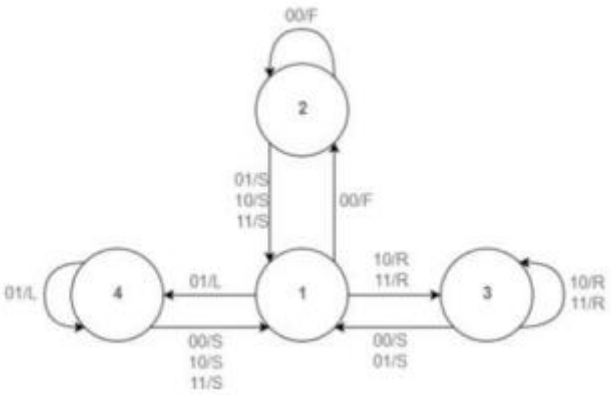


Figure 2.18 Detailed block diagram





Chapter 3-Product Design(Not required for ME&S add only relevant parts)

3.1 System Consideration for the Design

3.2 Criteria for Component Selection (from final approved project specification)

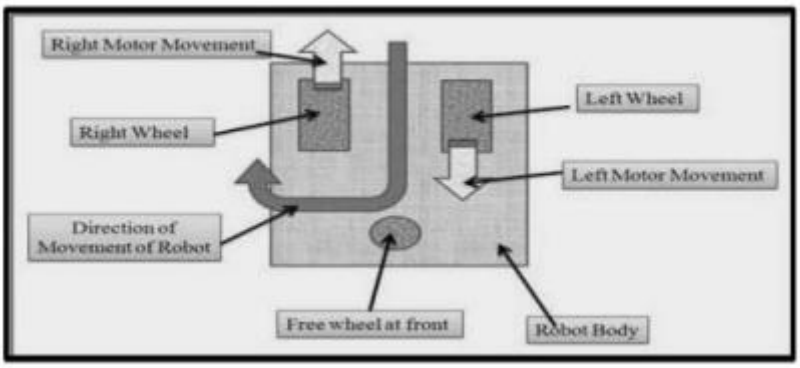
Why DC motors?

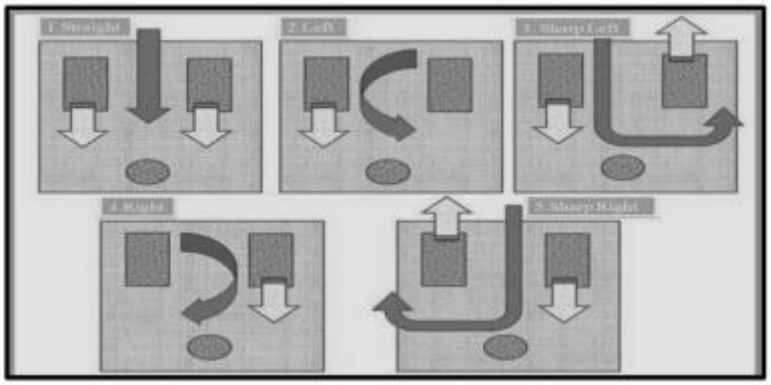
Dc motors are most easy to control. One dc motor requires only 2 signals for its operation. If we want to change its direction just reverse the polarity of the power supply across it, we can vary speed by varying the voltage across

motor. Mathematical interpretation: Rotation power is given by: T=$Pr/\ w$ Rotational power is constant for DC motor for constant input electrical power. Thus, torque is inversely proportional speed.

Why 2 Motors?

By using 2 motors we can move our robot in any direction. This steering mechanism of the robot is called differential drive. This means that in order to increase the power output of the motor, you can increase the voltage rating or increase the current. For example, a 12 volt DC motor can supply the same power as a six volt DC motor, but at 1/2 the current. This is important because most components are limited by the amount of current they can carry. If your robot will be extremely heavy, you may even want to look at 24 volt DC or even 90 volt DC motors. One of the trade offs for the higher voltage is safety. It is hard to shock yourself at 12 or 24 volts, but 90 volts can cause shock and possible injury. Another key property of DC motors is that the speed is controlled by changing the voltage





3.3 Committed accuracy and resolution system

3.4 Deliverable with calculated values of forces for actuator section and discussion of tradeoff for your final choice of actuator selection

Chapter 4- Mechanical Design

Mechanical design play very important rule in any project of engineering. Our line following robot is an engineering project

that can be affected by a lame or bad model of mechanical design. We can say that the mechanical design is responsible for the sustainability, weight lifting, long lasting life of the robot. So, designing the robot in a most sufficient way so that external condition or internal condition on that material used and that design are minimum

We adopted for our line follower robot is a Base (Figure 1) that is the main body over which the PCB (Figure 2) is mounted by some Connecter (Figure 3) of some height. So, there is some space created between the PCB and Base. This Space is filled by placing the power supply/ Lithium Rechargeable Cells (Figure 4). The base's lower side is connected to 2 Motors (Figure 5) on each back side (Left and Right) by a rectangular shape connecter (Figure 6). These Motors Then connected to the Tires (Figure 7) which are responsible for moment (Front, Left or right). Now the front side of the line follower robot Base is directly connected to a freely moving wheel (Figure 8) in any direction according to the moment of backward tires. Over the PCB, there will be the Circuit that is MUXs, Gates, Sil Headers, Diodes, Motor Driver. Bolts and Nuts (Figure 9) are used in different places for tightness. The 5 IR Sensors (Figure 10) are mounted on the front end of the Base according to the truth table condition. Round 2 wheel Robot Car Chassis Mini Round Double-Deck kit comes with Round shape chassis, Motors, wheels and all required assembly parts to build a complete robot car. Unleash your inner Engineer and make your vehicle dreams a reality with the Mini Round Robot Chassis Kit. Round 2 wheel Robot Car Chassis is a transparent platform for building a robot. The set includes two DC motors with wheels with a diameter of 65 mm. At the front and back supported by a metal rotating wheel. The chassis components are made of acrylic, have mounting holes allowing you to install all the sensors, controllers and others. Transparent, universal platform, allowing you to build a robot, e.g. line follower or fighting robot (sumo). For control, you can use any controller, including Arduino with Motor Shield. Round 2 wheel Robot Car Chassis gives you everything you need to build the shell of a 2- wheel-drive Mobile Platform Robot! You get the metal plates that make up the chassis, two DC drive

motors with matching wheels, and a caster ball for balance. You'll fill in the rest with a power supply, microcontroller board, and motor controller. The differential drive (two separately driven wheels) allows for a near zero turning radius while the high-strength aluminum alloy body plus high-quality high-speed motors make it suitable for flat indoor surfaces. Also it's adorably small

Features:

1. Mechanical structure is simple,
2. 2. it is easy to install
3. . 3. This car is the tachometer encoder.
4. 4. With 2 AA battery box.
5. 5. Can be used for distance measurement, velocity.
6. 6. Can use with other devices to realize function of tracing, obstacle avoidance, distance testing, speed testing, wireless remote control

Chapter 5- Electronics Design and Sensor Selections

Ultrasonic transducers and ultrasonic sensors are devices that generate or sense ultrasound energy. An ultrasonic sensor sends a high pulse (signal) and then a low pulse (signal) in a continuous manner. Once these signals hit an obstacle, the signals reflect back and are received by ultrasonic sensor. The time taken by the signals to return is used to calculate the distance between the sensor and the obstacle. The closer the obstacle is to the sensor, the quicker these signals return.

Figure 2.1 ULTRASONIC SENSOR

Pin Configuration:

Pin Number	Pin Name	Description
1	Vcc	The Vcc pin powers the sensor, typically with +5V
2	Trigger	Trigger pin is an Input pin. This pin has to be kept high for 10us to initialize measurement by sending US wave.
3	Echo	Echo pin is an Output pin. This pin goes high for a period of time which will be equal to the time taken for the US wave to return back to the sensor.
4	Ground	This pin is connected to the Ground of the system.

Table 2.1 Pin Configuration

HC-SR04 Ultrasonic (US) sensor is a 4 pin module, whose pin names are Vcc, Trigger, Echoand Ground respectively. This sensor is a very popular sensor used in many applications

where measuring distance or sensing objects are required. The module has two eyes like projects in the front which forms the Ultrasonic transmitter and Receiver. The sensor works with the simple high school formula that

Distance = Speed × Time

The Ultrasonic transmitter transmits an ultrasonic wave, this wave travels in air and when it gets objected by any material it gets reflected back toward the sensor this reflected wave is observed by the Ultrasonic receiver module.

ARDUINO MEGA 2560:

ESP32 WROOM32-38pins

The ESP32 is a versatile Wi-Fi and Bluetooth module widely used in IoT projects and applications, renowned for its low power consumption, dual-core processing, and extensive connectivity options, making it ideal for a rangeof wireless projects from simple sensor networks to complex IoT solutions

Figure 2.4 ARDUINO AtMega-2560

Pin Configuration:

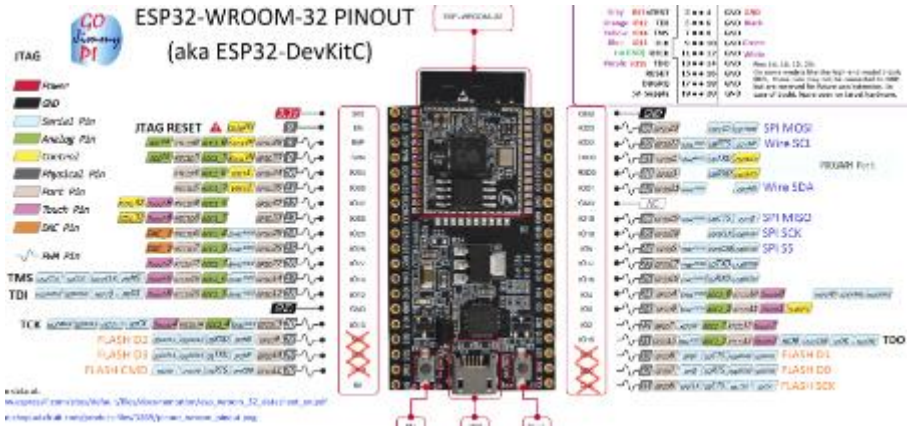


Figure 2.5 Pin Configuration

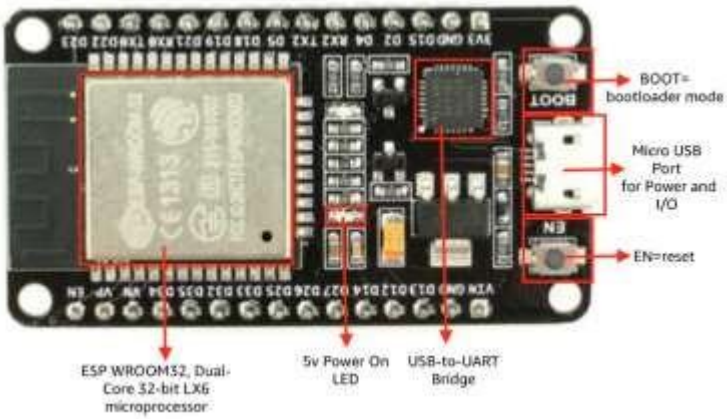
2.2.1. Robotic Chassis (2 Wheel with DC Motor):

This robotic chassis kit contains of an acrylic base with two gear motors, two compatiblewheels, a ball caster, and other accessories.

Figure 2.6 ROBOTIC CHASIS

2.2.2. ESP32

ESP32 is the name of the chip that was developed by Espressif Systems. This provides Wi-Fi (and in some models) dual-mode Bluetooth connectivity to embedded devices. While ESP32 is technically just the chip, modules and development boards that contain this chip are often also referred to as "ESP32" by the manufacturer. ESP32 can perform as a complete standalone system or as a slave device to a host MCU, reducing communication stack overhead on the main application processor. ESP32 can interface with other systems to provide Wi-Fi and Bluetooth

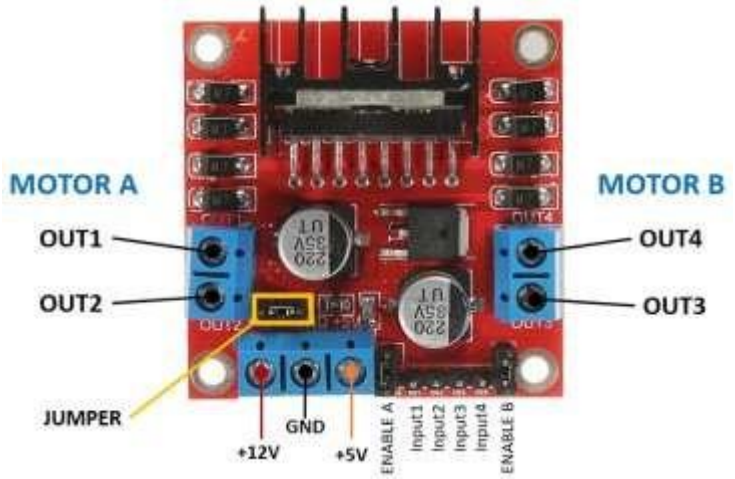


functionality through its SPI / SDIO or I2C / UART interface.

Figure 2.7 ESP 32

2.2.3. L298N Motor Driver

This **L298N Motor Driver Module** is a high power motor driver module for driving DC andStepper Motors. This module consists of an L298 motor driver IC and a 78M05 5V regulator. **L298N Module** can control up to 4 DC motors, or 2 DC motors with directional and speed control.



2.2.4. TT Gear Motor:

The controller i.e., L298N is basically used to control this DC motor. Its speed and direction are controlled by the l298N. D.C. motors are built to operate in the steady state in a speed range close to their no-load speed this speed is generally too high for applications like the robot we are making so that is why we merge the DC motor with gearbox to reduce its speed. The basic working principle of the DC motor is that whenever a current carrying conductor places in the magnetic field, it experiences a mechanical force. The principle remains the same but using them both as a single component is very helpful in performing variety of functions.

Figure 2.9 TT Gear Motor

2.2.5. TCRT 5000 IR SENSOR

The TCRT5000 are reflective sensors which include an infrared emitter and phototransistor ina leaded package which blocks visible light. The package includes two mounting clips. The TCRT5000 are reflective sensors which include an infrared emitter and phototransistor in a leaded package which blocks visible light.

2.2.6. SD Card Module

SD cards or Micro SD cards are widely used in various applications, such as data logging, data visualization, and many more. Micro SD Card Adapter modules make it easier for us toaccess these SD cards with ease. The Micro SD Card Adapter module is an easy-to-use module with an SPI interface and an on-board 3.3V voltage regulator to provide proper supply to the SD card.

Figure 2.12 SD CARD MODULE

2.2.7. 16 x 4 LCD Display

16X4 CHARACTER LCD 1604 GREEN LCD DISPLAY is a dot-matrix liquid crystal display module specially used for displaying letters, numbers, symbols, etc. Divided into 4- bit and 8-bit data transmission methods. 1604 Green Character LCD provides rich command settings: clear display; cursor return to origin; display on/o; cursor on/o; display character ashes; cursor shift; display shift, etc. It can be used in any embedded systems, industrial device, security, medical and hand-held equipment.

Pin Configuration:

Pin No.	Symbol	Description
1	V_{SS}	Ground
2	V_{DD}	Power supply for logic
3	V_O	Contrast Adjustment
4	RS	Data/ Instruction select signal
5	R/W	Read/Write select signal
6	E	Enable signal
7~14	DB0~DB7	Data bus line
15	A	Power supply for B/L +
16	K	Power supply for B/L -

Table 2.3 Pin Configuration

2.2.8. ACS712 CURRENT SENSOR

The **ACS712 Module** uses the famous **ACS712 IC** to **measure current** using the Hall Effectprinciple. The module gets its name from the IC (ACS712) used in the module, so for you final products use the IC directly instead of the module.

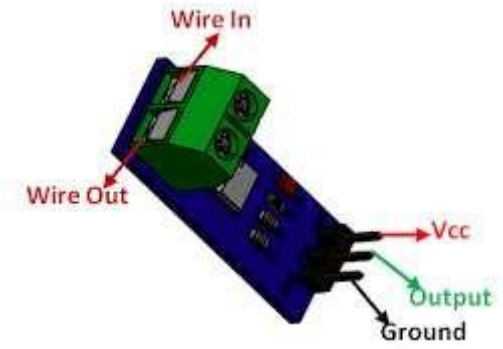


Figure 2.14 Current Sensor

2.2.9. VOLTAGE SENSOR

Voltage Sensor is a precise low-cost sensor for measuring voltage. It is based on the principle of resistive voltage divider design. It can make the red terminal connector input voltage to 5 times smaller.

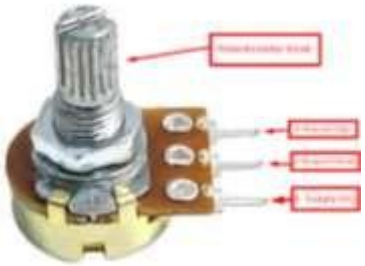
Figure 2.15 Voltage Sensor

2.2.10. *Potentiometer*

A potentiometer is a three-terminal resistor with a sliding or rotating contact that forms an adjustable voltage divider. If only two terminals are used, one end and the wiper, it acts asa variable resistor or rheostat.

The measuring instrument called a potentiometer is essentially a voltage divider used formeasuring electric potential (voltage); the component is an implementation of the same principle, hence its name.

Potentiometers are commonly used to control electrical devices such as volume controls on audio equipment. Potentiometers operated by a mechanism can be used as position transducers, for example, in a joystick. Potentiometers are rarely used to directly control significant power (more than a watt), since the power dissipated in the potentiometer would be comparable to the power in the controlled load.



CHAPTER 5- SOFTWARE/FIRMWARE DESIGN/THEORETICALDESIGN

DESIGN

5.1 Explanation

Following are the inputs and outputs of the circuit.

Inputs

1. Left Sensor (LS)
2. Right Sensor (RS)
3. Right Centre Sensor (RC)
4. Left Centre Sensor (LS)
5. Centre Sensor (CS)
6. Ultrasonic Sensor
7. Voltage sensor
8. Current sensor
9. Rotary encoder (Tachometer)

Outputs

1. Left Motor
2. Right Motor
3. SD card
4. 16 X 4 LCD

5.2 SCHEMATIC COMPONENTS

1. Ultrasonic sensor
2. SD card module
3. Servo motor
4. TCRT5000 IR sensors
5. L298 motor driver
6. ESP 32
7. 16x4 LCD display
8. Motors

9. Current sensor
10. Voltage sensor
11. Potentiometer

IR1	IR2	IR3	L	R
1	1	1	0	0
1	1	1	0	1
1	1	1	0	1
1	1	1	0	1
1	1	0	1	0
1	1	0	0	1
1	1	0	0	1
1	1	0	0	1
1	0	1	1	0
1	0	1	1	1
1	0	1	1	1
1	0	1	1	1
1	0	0	1	0
1	0	0	1	0
1	0	0	1	0
1	0	0	0	1
0	1	1	1	0
0	1	1	1	1
0	1	1	1	1
0	1	1	1	1
0	1	0	1	0
0	1	0	1	0
0	1	0	0	1
0	1	0	0	1
0	0	1	1	0
0	0	1	1	1

Chapter 6- Programming And Sensor Integration

6.1 Integration of LCD

RS pin of the LCD module is connected to digital pin 12 of the Arduino. R/W pin of the LCDis grounded. Enable pin of the LCD module is connected to digital pin 11 of the Arduino. In this project, the **LCD module and Arduino are interfaced in the 4-bit mode**. This means only four of the digital input lines (DB4 to DB7) of the LCD are used. This method is very simple, requires less connections and you can almost utilize the full potential of the LCD module. Digital lines DB4, DB5, DB6 and DB7 are interfaced to digital pins 5, 4, 3 and 2 of the Arduino. The 10K potentiometer is used for adjusting the contrast of the display. 560 ohmresistor R1 limits the current through the back light LED. The Arduino can be powered through the external power jack provided on the board. +5V required in some other parts of the circuit can be tapped from the 5V source on the Arduino board. The Arduino can be also powered from the PC through the USB port. The full program for interfacing LCD to Arduino is

shown below.

CODE

```
/* This is a sketch to test 16x4 LCD:
#include LiquidCrystal lcd(8,9,4,5,6,7);
void setup() {
lcd.begin(16,4);
lcd.setCursor(0,0);
lcd.print("VOLTAGE");
lcd.setCursor(0,1);
lcd.print("CURRENT!");
lcd.setCursor(0,2);
lcd.print("P/N: ");
lcd.setCursor(0,3);
lcd.print("SD CARD");
}
void loop() {
}
```

6.2 Integration of Sd Card Module

Description	Command
Initializes the SD library and card. Enter the pin connected to the SS pin as a function's argument.	SD.begin(#sspin)
Tests whether a file or directory exists on the SD card.	SD.exists(filename)
Opens a file on the SD card in reading or writing mode. (If you leave the mode section blank, the file will open in reading mode by default) If the file is opened for writing, it will be created a file with this name if it doesn't already exist.	SD.open(filepath, mode)
Close the file and ensure that any data written to it is physically saved to the SD card.	file.close()*
Remove a file from the SD card.	SD.remove(filename)
Create a directory on the SD card	SD.mkdir(filename)
Remove a directory from the SD card.	SD.rmdir(filename
Returns the file name	file.name()*
Print data to the file	file.print(data)
Print data, followed by a carriage return and newline	file.println(data
Read from the file.	file.read()
Check if there are any bytes available for reading from the file.	file.available()

Storing data is one of the most important parts of every project. There are several ways to store data according to the data type and size. SD and micro SD cards are one of the most practical ones among the storage devices, which are used in devices such as mobile phones, minicomputers and etc.

6.3 Final complete code of project

Microcontroller Code

```
#include <LiquidCrystal.h> //library for LCD

// initialize the library with the numbers of the interface pins
 LiquidCrystal lcd(8, 9, 10, 11, 12, 13);

// Voltage Sensor
int analogInput = A1;      //voltage sensor
float vout = 0.0;

float vin = 0.0;
```

```
int value = 0;
int sensor = A2; //u sensor unsigned
long start_time = 0;unsigned long
end_time = 0; int steps=0;

float steps_old=0;float
temp=0; float rps=0;

//Measuring Current Using ACS712
const int currentPin = A0; // current sensor is attachedint sensitivity =
66;   // 66-185 mv/A

int adcValue= 0; //
int offsetVoltage = 2500; // 0.25 mv 0v ki value ko remove karne k liyedouble
adcVoltage = 0;

double currentValue = 0;
//variables for battery level indicatorint f=2,
e=3,d=4, c=5, b=6,a=7;

void setup() {
   //baud rate
   Serial.begin(9600);//baud rate at which arduino communicates withLaptop/PC
   // set up the LCD's number of columns and rows
   lcd.begin(16, 4);   //LCD order

pinMode(sensor,INPUT_PULLUP);
//   lcd.setCursor(0,0);
//   lcd.print(" STEPS - 0");
//   lcd.setCursor(0,1);
//   lcd.print(" RPS                - 0.00");
pinMode(a,OUTPUT);
pinMode(b,OUTPUT);
pinMode(c,OUTPUT);
pinMode(d,OUTPUT);
pinMode(e,OUTPUT);
```

```
digitalWrite(d,HIGH);
delay(500);
digitalWrite(c,HIGH);
delay(500);
digitalWrite(b,HIGH);
delay(500);
digitalWrite(a,HIGH);
delay(500);
digitalWrite(a,LOW);
delay(500);
digitalWrite(b,LOW);
delay(500);
digitalWrite(c,LOW);
delay(500);
digitalWrite(d,LOW);
delay(500);
digitalWrite(e,LOW);
delay(500);
digitalWrite(f,LOW);
delay(500);

  lcd.setCursor(0,0);//Setting cursor on LCD
  lcd.print("CHEAP THRILLS");//Prints on the LCDdelay(1000);

  lcd.setCursor(1,1); lcd.print("~HASSAN
  190979"); lcd.setCursor(1,2);
  lcd.print("~MARIUM 190967");
  lcd.setCursor(1,3); lcd.print("~UZAIR
  190970"); delay(2000);//time delay for 3
  sec

  lcd.clear();//clearing the LCD display
  lcd.display();//Turning on the display again
  lcd.setCursor(1,0);//setting LCD cursor lcd.print("Values
  from");//prints on LCD lcd.setCursor(1,1);
```

```
    lcd.print("ACS712 and");
    lcd.setCursor(1,4);
    lcd.print("Voltage sensor");
    lcd.clear(); delay(500);//delay for 1
    sec}

void loop() //method to run the source code repeatedly
{start_time=millis();
  end_time=start_time+1000;
  while(millis()<end_time)

  {if(digitalRead(sensor))
      { steps=steps+1; while(digitalRead(sensor));}

      lcd.setCursor(9,2);
      lcd.print(steps);
      lcd.print("              ");}

// voltage
      value = analogRead(analogInput);vout
      = (value * 5.0) / 1024.0; vin = vout /
      (R2/(R1+R2));

if(vin>11.88)
{ digitalWrite(a,HIGH);
      digitalWrite(b,HIGH);

        digitalWrite(c,HIGH);
          digitalWrite(d,HIGH);

            digitalWrite(e,HIGH);
              digitalWrite(f,HIGH);}

if (vin<=11.46 && vin>11.28)
{ digitalWrite(a,LOW);
      digitalWrite(b,HIGH);

        digitalWrite(c,HIGH);
          digitalWrite(d,HIGH);
```

```
        digitalWrite(d,HIGH);
          digitalWrite(e,HIGH);

            digitalWrite(f,HIGH);}if
(vin<=10.90 && vin>10.79)

{ digitalWrite(a,LOW);
     digitalWrite(b,LOW);

       digitalWrite(c,LOW);
        digitalWrite(d,HIGH);

          digitalWrite(e,HIGH);
            digitalWrite(f,HIGH);}

if (vin<=10.60 && vin>10.53)
{digitalWrite(a,LOW);
     digitalWrite(b,LOW);

       digitalWrite(c,LOW);
        digitalWrite(d,LOW);

          digitalWrite(e,HIGH);
            digitalWrite(f,HIGH);}

if   (vin<=10.53)
{ digitalWrite(a,LOW);
     digitalWrite(b,LOW);

       digitalWrite(c,LOW);
        digitalWrite(d,LOW);

          digitalWrite(e,LOW);
            digitalWrite(f,HIGH);}

// current
   adcValue = analogRead(currentPin); adcVoltage =
   (adcValue / 1024.0) * 5000;

   currentValue = ((adcVoltage - offsetVoltage) / sensitivity);lcd.clear();
```

```
 lcd.setCursor(8,1);
 lcd.print(vin,2);
 lcd.setCursor(12,1);
 lcd.print("V");

lcd.setCursor(0,2);
  lcd.print(" STEPS = 0");
  lcd.setCursor(0,3);
  lcd.print(" RPS   = 0.00");

      temp=steps-steps_old;
    steps_old=steps;
    rps=(temp/20);
    lcd.setCursor(9,3);
    lcd.print(rps);
```

CODE:

```
#include <SPI.h>
#include <SD.h>

#include <Servo.h> //Servo motor library. This is standard library
#include <NewPing.h> //Ultrasonic sensor function library. You must install this library
File myFile;
//sensor pins
#define trig_pin A4 //analog output 1
#define echo_pin A5 //analog input 2
#define maximum_distance 200

boolean goesForward = false;
int distance = 100;

NewPing sonar(trig_pin, echo_pin, maximum_distance); //sensor function
Servo servo_motor; //our servo name

void setup()
{
Serial.begin(9600);
SD.begin(10);
```

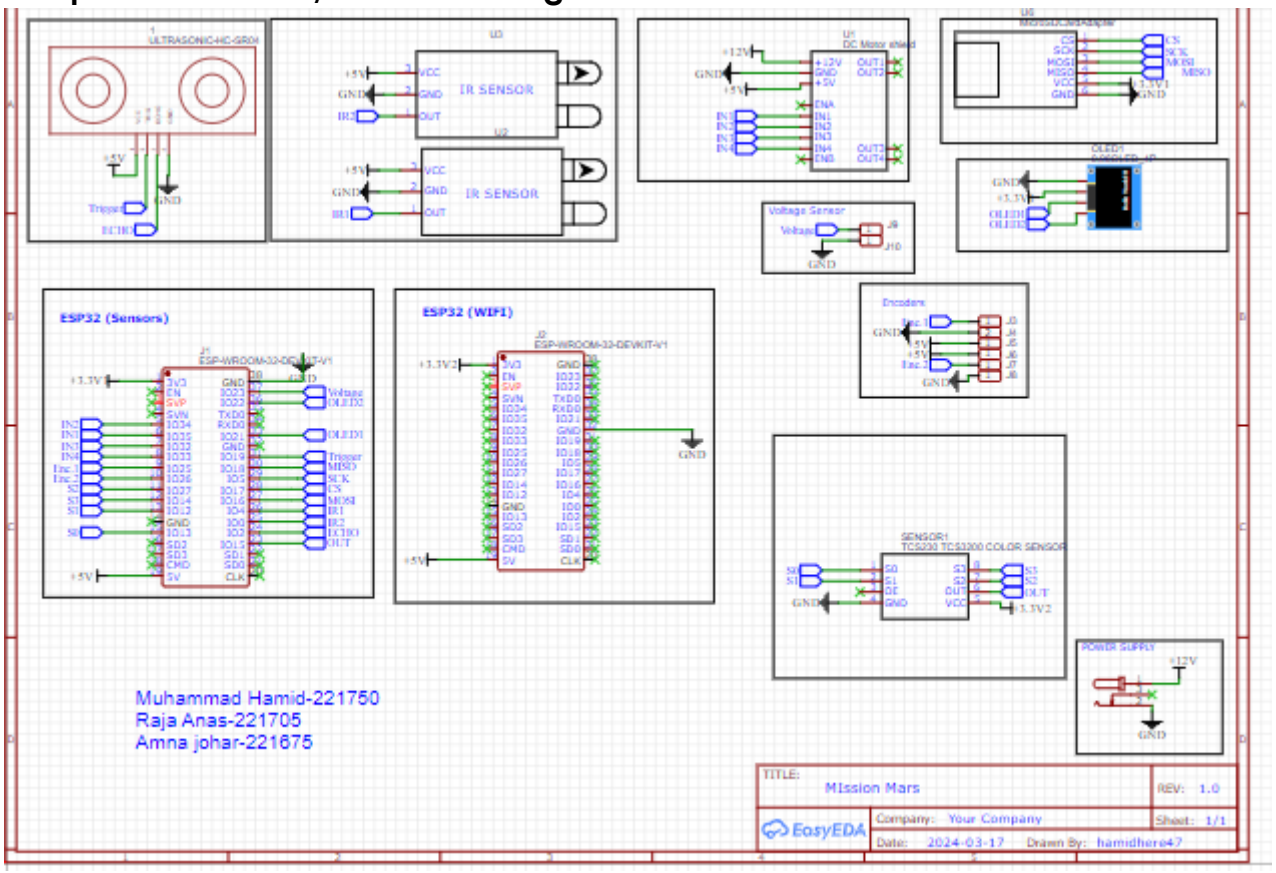
```
  distance   =   readPing(); pinMode(2,
  INPUT);//left sensor 1

  pinMode(4, INPUT); //left sensor 2
  pinMode(5, INPUT); //middle sensor3
  pinMode(6, INPUT); //right sensor 4
  pinMode(7, INPUT); //right sensor5
  pinMode(3, OUTPUT);   // pwm

    pinMode(8, OUTPUT);
  pinMode(A0, OUTPUT); //LEFT MOTOR.
  pinMode(A1, OUTPUT);
  pinMode(A2, OUTPUT); //RIGHT MOTOR.
  pinMode(A3, OUTPUT);}

  // put your setup code here, to run oncevoid
loop(){

  int distanceRight = 0;int
  distanceLeft = 0; delay(50);

  if (distance <= 25){ distanceRight =
  lookRight();delay(0);

  distanceLeft = lookLeft();
  delay(0);}

  distance = readPing();if
(distance<=25){ STOP();}

else{
  int a = digitalRead(2);int b =
  digitalRead(4);

int c = digitalRead(5); int d =
  digitalRead(6);int e =
  digitalRead(7);
```

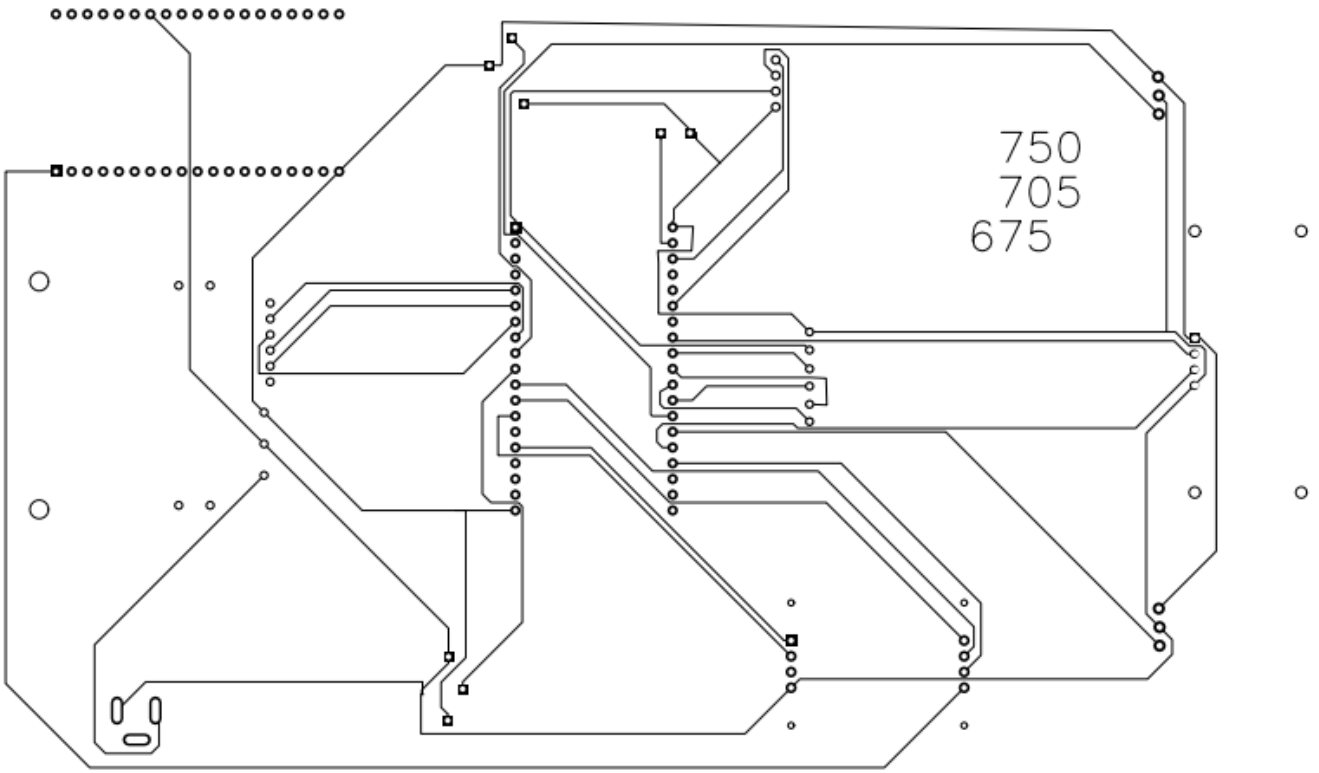
```
  analogWrite(A3, 255);
  analogWrite(3, 100);f(); }

if ((a == LOW && b == HIGH && c == LOW && d==LOW && e==LOW) | | (a==HIGH &&b==
HIGH && c==LOW &&d==
LOW &&e==LOW) | | (a==HIGH && b== HIGH && c==HIGH &&d==
LOW&&e==LOW) | | (a==HIGH && b== LOW &&
c==LOW &&d== LOW &&e==LOW) | | (a==LOW && b== HIGH && c==HIGH &&d==
LOW&&e==LOW) | | (a==HIGH &&
b== HIGH && c==HIGH && d==HIGH && e==LOW)) //LEFT TURN
  {
  analogWrite(A0, 0);
  analogWrite(A1, 255);
  analogWrite(A2, 0);
  analogWrite(A3, 0);
  analogWrite(3, 100);l(); }

if ((a == LOW && b == LOW && c == LOW && d==HIGH && e==LOW) | | (a==LOW && b==LOW
&& c==LOW &&d==
HIGH &&e==HIGH) | | (a==LOW && b== LOW && c==HIGH &&d== HIGH
&&e==HIGH) | | (a==LOW && b== LOW &&
c==LOW &&d== LOW &&e==HIGH) | | (a==LOW && b== HIGH && c==HIGH &&d==
HIGH&&e==HIGH) | | (a==LOW &&
b== LOW && c==HIGH &&d== HIGH &&e==LOW) ) //RIGHT TURN
  {
  analogWrite(A0, 0);
  analogWrite(A1, 0);
  analogWrite(A2, 0);
  analogWrite(A3, 255);
  analogWrite(3, 100);r(); }

if (a == LOW && b == LOW && c == LOW && d==LOW && e== LOW) // BACK {
  analogWrite(A0, 0);
  analogWrite(A1, 255);
```

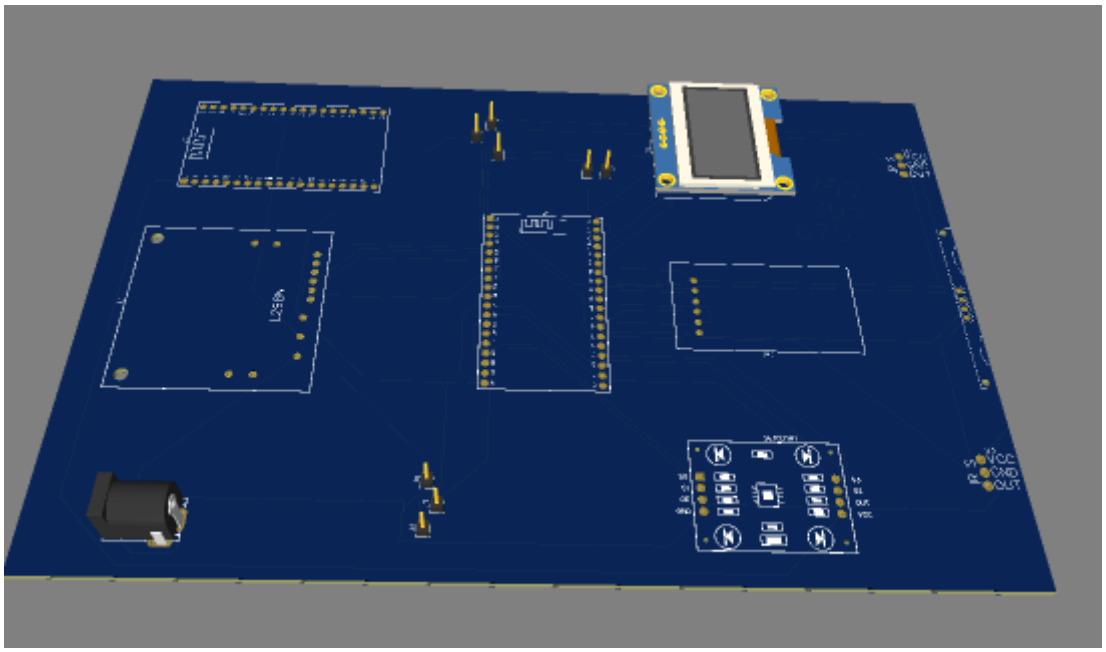
```
  delay(100);
  int distance = readPing();
  delay(100);
  servo_motor.write(115); return
  distance;}

int lookLeft(){
  servo_motor.write(170);
  delay(100);

  int distance = readPing();
  delay(100);
  servo_motor.write(115); return
  distance; delay(100);}

int readPing(){
  delay(70);

  int cm = sonar.ping_cm();if
  (cm==0)

  { cm=250;   }
  return cm; }
void STOP(){

  analogWrite(A0, 0);
  analogWrite(A1, 0);
  analogWrite(A2, 0);
  analogWrite(A3, 0);
  analogWrite(3, 100);
  obs();

  }
  void f(){
    // open the file. note that only one file can be open at a time,
    // so you have to close this one before opening another.myFile =
    SD.open("test.txt", FILE_WRITE);
```

```
  // open the file. note that only one file can be open at a time,
  // so you have to close this one before opening another.myFile =
  SD.open("test.txt", FILE_WRITE);

  // re-open the file for reading:
  myFile = SD.open("test.txt");
  Serial.println("right.txt:");

     Serial.println("right.txt:"); myFile.close(); }

  void l(){
  // open the file. note that only one file can be open at a time,
  // so you have to close this one before opening another.myFile =
  SD.open("test.txt", FILE_WRITE);

  // re-open the file for reading:
  myFile = SD.open("test.txt");
  Serial.println("left.txt:");

     Serial.println("left.txt:"); myFile.close();   }

  void bs(){
  // open the file. note that only one file can be open at a time,
  // so you have to close this one before opening another.myFile =
  SD.open("test.txt", FILE_WRITE)

  // re-open the file for reading:
  myFile = SD.open("test.txt");
  Serial.println("back.txt:");

     Serial.println("back.txt:"); myFile.close(); }

void obs(){
  // open the file. note that only one file can be open at a time,myFile =
  SD.open("test.txt", FILE_WRITE)
```

Chapter 6- Software/Firmware Design




750
705
675



COMPONENT	PRICE
1 chassis (2 tires, 2 motors, bolts, nuts, and base)	1200-1500
Wires (Male to female, male to male, female tofemale)	400
Soldering Wire	400
Sticker Sheet	150-210
Ultrasonic Sensor	250
ESP 32	950
Sil headers (male & female)	180-200
Voltage Sensor	150
Current Sensor	230-250
Double sided tape	100-150
Ferric chloride	250
5 TCRT5000 IR sensors	5x180
L298N motor driver	350-370
ARDUINO MEGA 2650	4000
PCB	800
Bread board	140-200
12 battery cells	210
LCD Display	650
SD Card module	200
Ultrasonic sensor Holder	70-100
	Total= 8850

TABLE 10.1 TABLE OF CONTENT

Number of Sensors

Digital IR sensor works if we have only black or white surface area. The simplest casewill be if we use 2 IRsensor and can work if we have black path with white surface. But toget accuracy in robot, we are advised to used minimum 5 or 6 sensors, that will increase

the complexity of the circuit. So, making conditions, truth tables, simulation on proteus,PCB Layout etc. tickling of these isn't an easy job. So that part is also challenging.

Possibilities in Arena Track

The tract, we will use in project is called arena. The arena is of different type, sometimes we have white line with black surface area, sometimes the opposite. Let consider the black one with white surface area. Here the track may be trajectory, may include u turn, may have left and right turn atsingle place. That's challenging, and we will try to see if what can we do with these conditions.

Components Cost

The Component cost is something that may be so challenging for some students. As Pakistan is apoor country and we a lot of friends, class fellows for which it is not easy to buy some expensive component. We can identify the quality of component with less price available. Also, if somewhere a somebodyfinds the old used component that can used again and have less price, can also be bought.

Design of Robot

How to design our robot in a most sufficient way so that external condition or internal condition on that robot is minimum. There are thousands of designs available in daily life. Adding 4 wheels or 2 wheels to robot, where to place the sensors for best configuration, where to place components and batteries etc. they all should be kept in mind for good design.

Circuit Patching on bread board

For this regard, we first use proteus software, to check if our circuit is working or not. So, making the circuit on proteus may be challenging. Now if we have done it then, implementing circuit on bread board is according to proteus work. But here we got the problem. On proteus, conditions are allideal. And it's easy to simulate, patch things. But it's when it comes to real life.

Interfacing different sensors on Microcontrollers

Interfacing different sensors on microcontrollers may cause problem so tickling them wasone of the good things we did during the project.

Desired Output

Now if all the things are done quite good. So, the robot got to be work according to desiredoutput required. But in practical world, we can't say anything. Maybe the output is different from what we expected. That may be the challenging part.

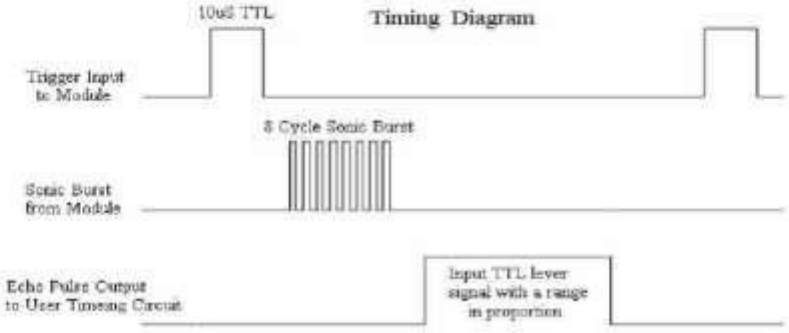
10.3 Learning Outcomes

1. In the subject, we study about microcontroller and embedded systems.This project will help us more about understanding and tickling them.
2. This project will let us learn, the implementation of these microcontrollers in reallife.
3. If we have given any simple real-life problem, then how to tickle them withmicrocontrollers and sensors
4. It will enhance our proteus schematic and PCB layout skills.
5. Making logic diagrams, truth table etc. skills will be enhanced.
6. We learn about sensors (IR, ultrasonic, current, voltage sensor), and how it worksand how to use and implement them inreal time life. We will learn, how quantityof sensor effects the robot.
7. How to store data in SD card.
8. How to display data on LCD display.
9. How to build something that can work by itself without human interference(Automation).
10. We will understand how to collect data from a given problem and do processesto make therequired and desires product.

Vcc Trig Echo GND

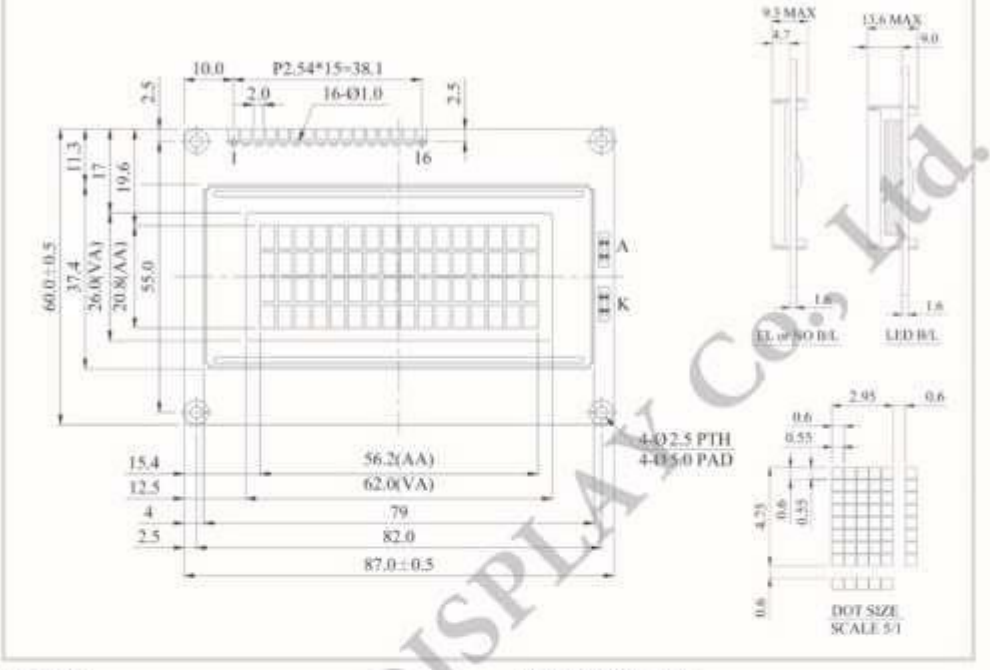
Timing diagram

The Timing diagram is shown below. You only need to supply a short 10uS pulse to the trigger input to start the ranging, and then the module will send out an 8 cycle burst of ultrasound at 40 kHz and raise its echo. The Echo is a distance object that is pulse width and the range in proportion .You can calculate the range through the time interval between sending trigger signal and receiving echo signal. Formula: uS / 58 = centimeters or uS / 148 =inch; or: the range = high level time * velocity (340M/S) / 2; we suggest to use over 60ms measurement cycle, in order to prevent trigger signal to the echo signal.



16X4 LCD:

WH1604A Character 16x4



Feature

1.5x8 dots includes cursor
2.Built-in controller (ST7066 or Equivalent)
3.5V power supply (Also available for 3V)
4.N.V. optional for 3V power supply
5.1/16 duty cycle
6.LED can be driven by PIN1, PIN2, PIN15, PIN16 or A and K
7.Interface : 6800, option SPI/I2C (RW1083 IC)

Pin No.	Symbol	Description
1	V_{SS}	Ground
2	V_{DD}	Power supply for logic
3	V_o	Contrast Adjustment
4	RS	Data/ Instruction select signal
5	R/W	Read/Write select signal
6	E	Enable signal
7	DB0	Data bus line
8	DB1	Data bus line
9	DB2	Data bus line
10	DB3	Data bus line
11	DB4	Data bus line
12	DB5	Data bus line
13	DB6	Data bus line
14	DB7	Data bus line
15	A	Power supply for B/L +
16	K	Power supply for B/L -

Mechanical Data

Item	Standard Value	Unit
Module Dimension	87.0 x 60.0	mm
Viewing Area	62.0 x 26.0	mm
Mounting Hole	82.0 x 55.0	mm
Character Size	2.95 x 4.75	mm

Electrical Characteristics

Item	Symbol	Standard Value typ.	Unit
Input Voltage	VDD	3/5	V
Recommended LCD Driving Voltage for Normal Temp. Version module @25°C	VDD-VO	4.35	V

Display Character Address Code

Display position	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
DD RAM Address	00	01														0F
DD RAM Address	40	41														4F
DD RAM Address	10	11														1F
DD RAM Address	50	51														5F

Conclusion

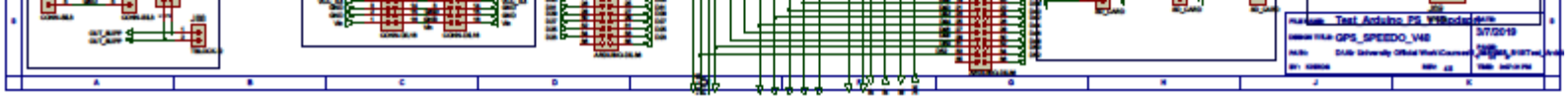
Although the outcome was simple, as mentioned earlier, the project makes us familiar with Arduino, the working mechanism of it and future aspects of it in a simple and understandable way.

✓ Although the thesis project is very little about the robot's use in real world, with the help ofguidelines and the abundance of resources the outcome, it could be very beneficial for many people and different sectors of the world depending on the sensors and features required as per necessity.

- *The goal of this project is to get students interested in and excited about the fields of engineering, mechatronics, and software development as they design, construct, and program an autonomous*

robot. The guidelines provided are very simple to use and understand thus, making it very easy for thenew students to build a foundation in their Robotics learning. Even being a powerful little device, microcontroller would be nothing without its hardware

x----------x


Test Arduino PS V4Prodelec
GPS_SPEEDO_V48
5/7/2019