LINE FOLLOWING ROBOT

Group Members:

Abdullah Ilyas	**(211212)**
Rabiah	**(211254)**
Salman Ali	**(211269)**
Izza Nadeem	**(211284)**
Muhammad Sohaib	**(212002)**

BE MECHATRONICS (Session 2021-2025)

Project Supervisor

Engr. Umer Farooq

Lecturer

DEPARTMENT OF MECHATRONICS ENGINEERING

FACULTY OF ENGINEERING

AIR UNIVERSITY, ISLAMABAD

ABSTRACT

The line follower robot is a mobile machine that can detect and follow the line drawn on the floor. Generally, the path is predefined and can be either visible like a black line on a white surface with a high contrasted color. This kind of robot should sense the line with its infrared ray (IR) sensors that installed under the robot. After that, the data is transmitted to the processor by specific transition buses. Hence, the processor is going to decide the proper commands and then it sends them to the driver and thus the path will be followed by the line follower robot.

Table of Contents

Contents

CHAPTER # 1 – PROJECT PRELAMINARIES

1.1 PROPOSAL

Our plan is to create a cost-efficient line-following robot that can navigate a given path in the shortest time possible. The robot would utilize a microcontroller system, infrared sensors to detect the line and to control the movement of the wheels. The robot would follow a binary path from Point A to Point B, and upon reaching Point B, a T-junction with a black line, it would come to a stop.

1.2 CONCEPT

Generally, the path is predefined and can be either visible like a black line on a white surface with a high contrasted colour or it can be invisible like a magnetic field. This kind of Robot should sense the line with its Infrared Ray (IR) sensors that installed under the robot. After that, the data is transmitted to Logic circuit. Hence, The Logic circuit is going to decide the proper commands and then it sends them to the driver and thus the path will be followed by the line follower robot. In this Report, we have illustrated the process of design, implementation and testing of a small line follower robot. Basic attributes of our project considered by us were reliable construction, cost effective design and data collection.

1.3 DESIGN

We have done our designing by working on the attributes of this project in detail.

1.4 CONSTRUCTION

1. Using button to turn ON or OFF the robot.

2. Implementing circuit on breadboard.

1.5 COST

1. Purchasing the components from whole seller.

2. Using the minimally optimum components to complete the project.

1.6 DATA COLLECTION

1. Current required by motors.

2. Required speed.

3. Voltage required.

4. Weight of the system with batteries.

5. Lighting check for the working of IR-sensors.

1.7 ESTIMATED BUDGETS

COMPONENT	PRICE
1 chassis (2 tires, 2 motors, bolts, nuts, and base)	1200-1500
Wires (Male to female, male to male, female to female)	400
Soldering Wire	200
Sticker Sheet	120-150
Ultrasonic Sensor	250
Sil headers (male & female)	180-200
Vero board	0
Voltage Sensor	150
Current Sensor	230-250
Double sided tape	100-150
Ferric chloride	250
5 IR sensors	5x180
Adhesives	0
motor driver	350-370
2 ARDUINO MEGA	2200
PCB	800
Bread board	140-200
12 battery cells	210
LCD Display	650
SD Card module	200
Servo motor	280
Ultrasonic sensor Holder	70-100

Total = 8,000-9,000

CHAPTER # 2-PROJECT CONCEPTIPON

2.1 INTRODUCTION

Robots are basically complete automatic machines. It starts working when sense something, decides to work according to the conditions and stop by its own due to other condition or senses. So, we can say robot are the replica of human being, as they work on phenomena as human being do. Robots are always made for the ease of human being. Now the robot may be **fixed or mobile robot**.

The **line follower robot** is one of the advance mobile robots, as we make its base moveable and it can move from one place to another. Line follower robot follows a particular path or trajectory and decides it decides its choices to interacts with obstacle. Usually the path is a black line on the white floor, but in other cases it may be a magnetic field which can't be seen by naked eyes. They can be mostly use in industry and in domestic chores as they can carry the parcels or materials from one place to another place.

2.2 LITRATURE REVIEW

2.2.1 Historical Background: The historical perspective of line follower robots encompasses the foundational work that laid the groundwork for the development of these autonomous systems. In the early days of robotics, the focus was primarily on creating rudimentary machines capable of following a predefined path. One of the earliest instances can be traced back to the mid-20th century when researchers began experimenting with basic sensors and control mechanisms.

Early line follower robots utilized simple light sensors that could detect the contrast between the line and its surroundings. These sensors provided the essential input for steering the robot along the path. The initial designs were often limited in their capabilities, relying on basic algorithms to maintain a steady course.

As technological advancements progressed, so did the sophistication of line follower robots. Researchers started incorporating more advanced sensors, such as infrared sensors, which offered improved accuracy in detecting the line.

Line follower robots are autonomous systems designed to navigate a predefined path by following a visible line on the ground. These robots have gained significant attention due to their applications in various fields, including industrial automation, logistics, and education. This literature review aims to provide an overview of the key concepts, methodologies, and advancements in the field of line follower robots.

2.2.2 Sensor Technologies: One of the critical components of a line follower robot is the sensor system. This section reviews the various sensor technologies employed in line follower robots. In our particular line follower robot we are using an 5 IR sensor array that can detect line strip of any size and covers larger area so the robot does not miss the line. Other than IR we are using an ultrasonic sensor for obstacle avoidance and other small sensors like current, voltage etc. these are some of the basic compulsory sensors required for the LFR, we have also added a display to show the readings those sensors take, also we are using a SD card reader with an interpreter to store the path followed by the robot so we can latter analyze the behavior of motors and to help choosing better paths.

2.2.3 Hardware Platforms: Line follower robots come in various shapes and sizes, utilizing different hardware platforms. This section explores the types of motors, wheels, and chassis commonly used in these robots. Additionally, it discusses the integration of microcontrollers, such as Arduino and Raspberry Pi, and their role in enhancing the overall functionality and versatility of line follower robots. In our project we have decided to use Arduino and particularly arduio mega because of its efficiency with this kind of smaller projects, more over it is relatively easier to code since we have learned to code in IDE in our labs as well. Another reason for choosing arduio mega is that arduino systems are less expensive and are widely available in market. After microcontroller is decided other main hardware component is the chases for the robot and we are using an acrylic board because it is much durable and will be able to withstand the wait of carbon fiber PCB easily.

2.2.4 Applications and Case Studies: The applications and case studies section delves into the real-world utilization of line follower robots across diverse industries and scenarios. It highlights the practical implications of these robots and provides insights into how they address specific challenges in various fields.

1. **Warehouse Automation:** Line follower robots play a pivotal role in warehouse automation, where efficient material handling is essential. These robots are employed to transport goods within warehouses, following designated paths marked on the floor. By automating the movement of items, line follower robots contribute to increased efficiency, reduced labor costs, and minimized errors in inventory management.

2. **Material Handling in Manufacturing:** Within manufacturing environments, line follower robots are deployed to automate the movement of materials along production lines. This application streamlines the manufacturing process, ensuring a smooth and continuous flow of components between different stages of production. This not only accelerates production but also enhances precision and reduces the risk of human error.

3. **Educational Settings:** Line follower robots serve as valuable educational tools in teaching robotics, programming, and engineering concepts. They provide a hands-on learning experience for students at various academic levels, from primary schools to universities. Through the construction and programming of line follower robots, students gain practical insights into robotics principles, enhancing their problem-solving and critical-thinking skills.

2.2.5 Applications and Case Studies: The future directions in line follower robots present exciting possibilities for further advancements and expanded applications. Several trends and emerging areas of interest are shaping the trajectory of research and development in this field:

1. **Advanced Sensor Integration:** Future line follower robots are likely to benefit from the integration of more advanced sensor technologies. Incorporating lidar, multi-spectral imaging, or computer vision capabilities can enhance the robot's perception and decision-making abilities, allowing it to navigate more complex and dynamic environments with improved accuracy.

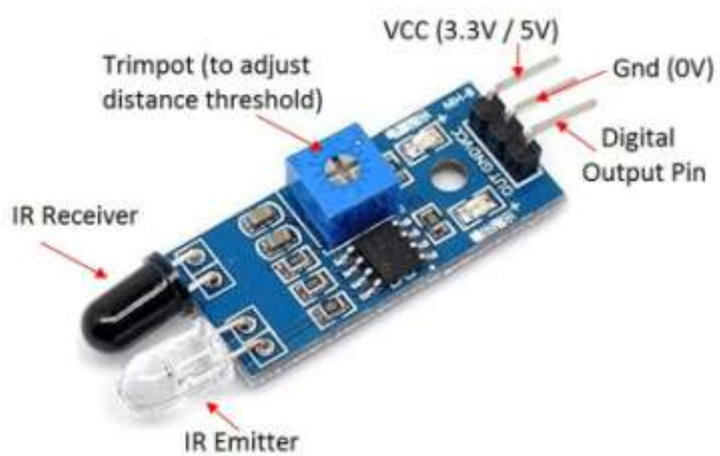
2. **Machine Learning and AI Algorithms:** The integration of machine learning and artificial intelligence algorithms holds promise for enhancing the intelligence of line follower robots. These algorithms can enable the robots to adapt to changing conditions, learn from experience, and optimize their navigation strategies over time. This could lead to improved performance in diverse and unpredictable scenarios.

3. **Multi-Robot Collaboration:** Future developments may focus on enabling collaboration among multiple line follower robots to accomplish complex tasks collectively. Coordinated efforts and communication between robots could enhance efficiency and scalability in applications such as warehouse management or search and rescue missions.

4. **Robustness in Challenging Environments:** Research efforts may concentrate on developing line follower robots that are more robust in challenging environments. This includes terrains with irregular lines, dynamic obstacles, or adverse weather conditions. Enhancements in robustness will broaden the range of applications and make these robots more adaptable to real-world scenarios.

2.3 LM393 IR SENSOR

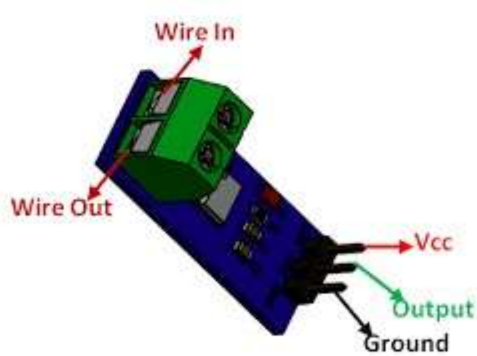
IR sensors use an IR LED to emit infrared signals and an IR photodiode to detect them. The photodiode's resistance and output voltage change based on the amount of infrared light it receives. These changes can be used to detect obstacles or measure distance. IR sensors are



commonly used for these purposes.

2.4 ACS712 CURRENT SENSOR

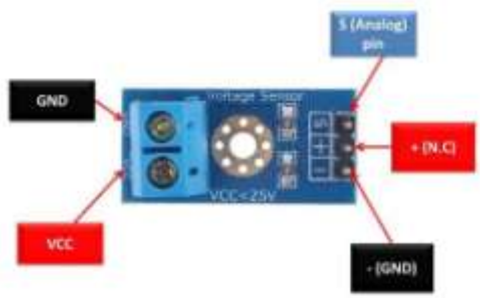
The **ACS712 Module** uses the famous **ACS712 IC** to **measure current** using the Hall Effect



principle. 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.

2.5 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.

2.6 ULTRASONIC SENSOR

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.

2.7 SERVO MOTOR

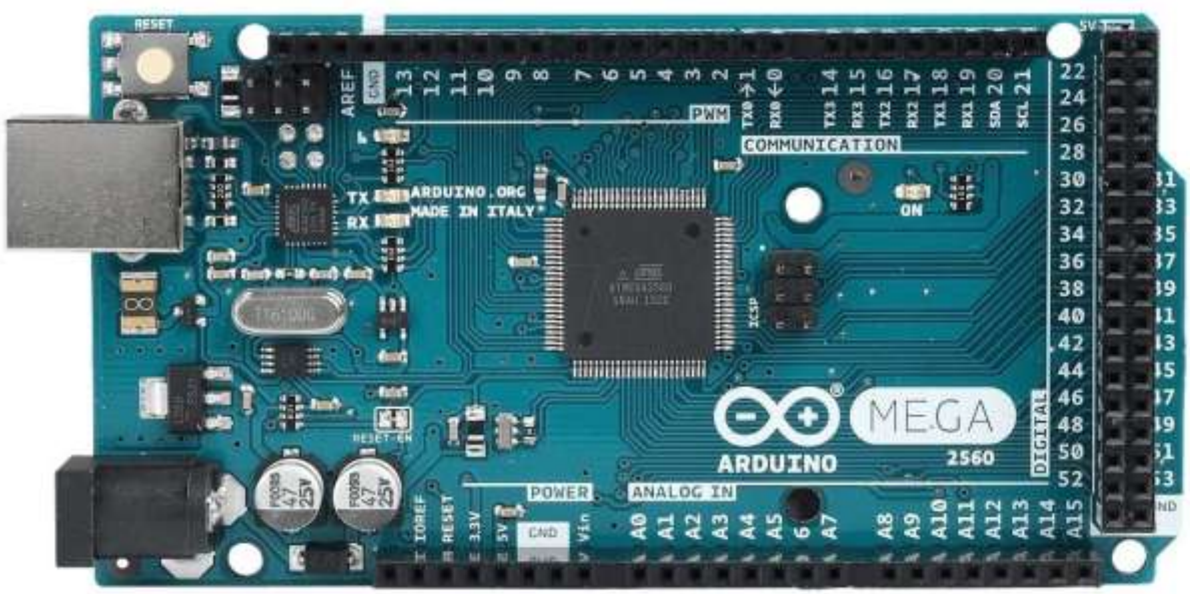
A **servo motor** is a type of motor that can rotate with great precision. Normally this type of motor consists of a control circuit that provides feedback on the current position of the motor shaft, this feedback allows the servo motors to rotate with great precision. If you want to rotate an object at some specific angles or distance, then you use a servo motor. It is just made up of a simple motor which runs through a **servo mechanism**. If motor is powered by a DC power supply then it is called DC servo motor, and if it is AC-powered motor then it is called AC

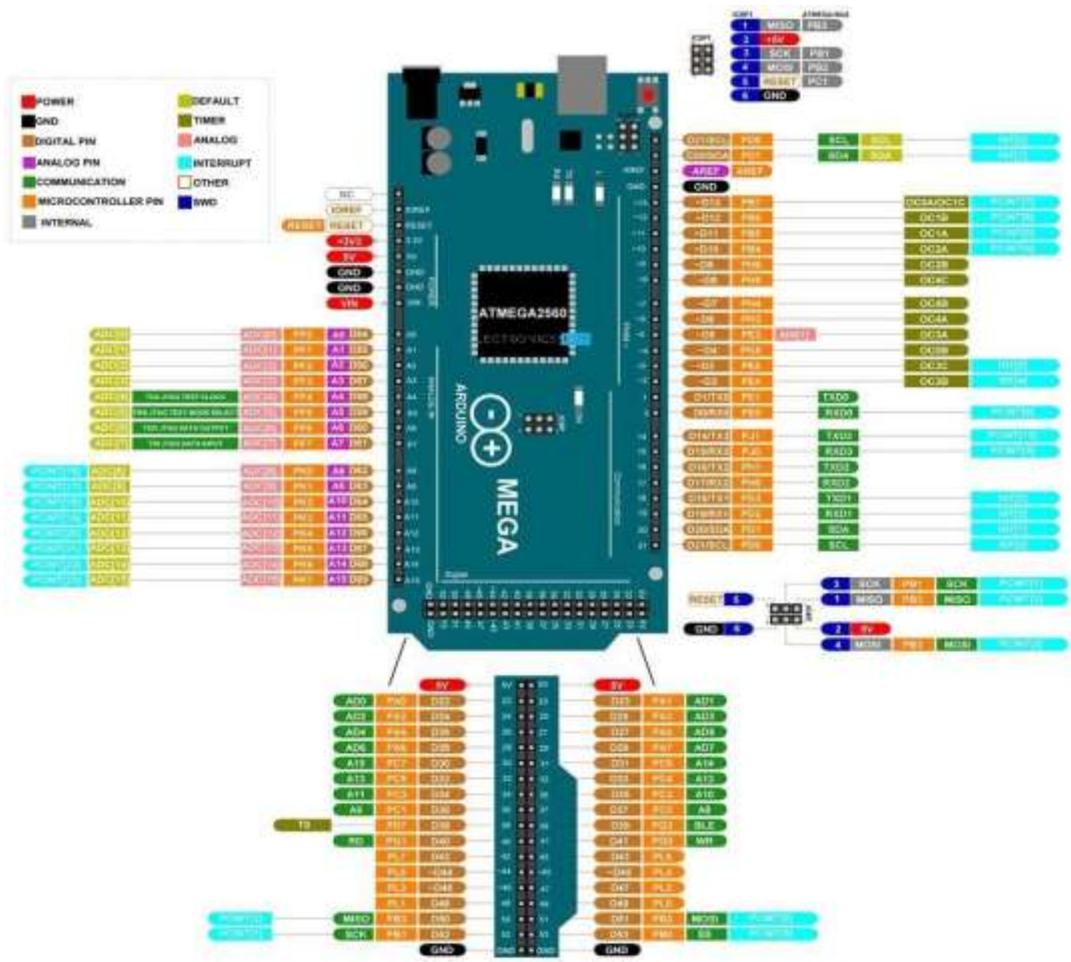
servo motor. For this tutorial, we will be discussing only about the **DC servo motor working.**

Servo motor can be rotated from 0 to 180 degrees, but it can go up to 210 degrees, depending on the manufacturing. This degree of rotation can be controlled by applying the **Electrical Pulse** of proper width, to its Control pin. Servo checks the pulse in every 20 milliseconds. The pulse of 1 ms (1 millisecond) width can rotate the servo to 0 degrees, 1.5ms can rotate to 90 degrees (neutral position) and 2 ms pulse can rotate it to 180 degree.

2.8 ARDUINO MEGA 2560

The **Arduino Mega 2560** is a microcontroller board based on the ATmega2560. It has 54 digital input/output pins (of which 15 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.





2.9 ROBOT CHASSIS

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

2.10 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 to access 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.

2.11 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 8bit 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 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 -

2.12 POTENTIOMETER

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 as a variable resistor or rheostat.

The measuring instrument called a potentiometer is essentially a voltage divider used for measuring 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

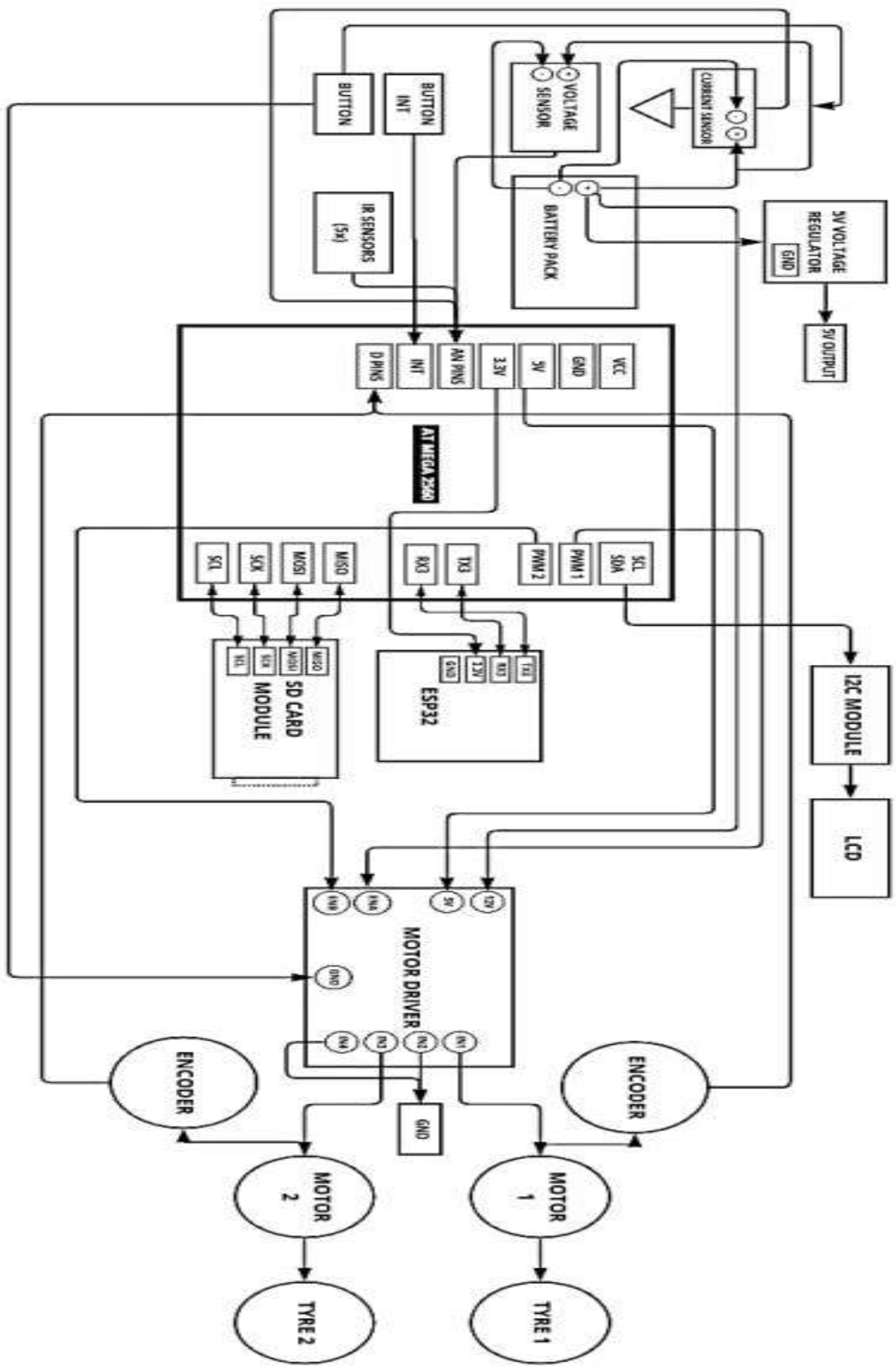
in the controlled load

2.13 BLOCK DIAGRAM

CHA

3.1 CC

Our p based



on lo ogic to

a low being

the lc

3.2 COMPONENTS SELECTION

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:

$$\mathbf{T}=Pr/\,w$$

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.

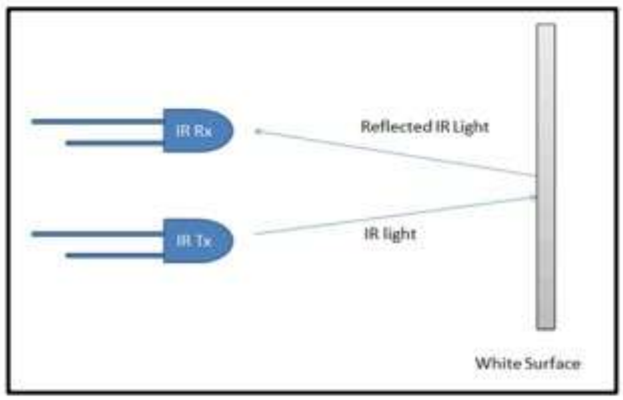
3.3 SENSORS SELECTION

As we are asked to make the line follower robot without using any microcontroller, so the best selection of the sensor in this regard is IR sensors. In our case of arena, the sensor will detect black line as path while white area as obstacle, so we have to configure our circuit accordingly. The simplest case will be if we use 2 IR sensor and can work, even 1 IR sensor works if we have only black or white surface area. But as we are advised to used minimum 6 or 5 sensors, that will increase the complexity of the circuit and robot, but it will improve the accuracy of our robot. That is why we will use 5 sensors in our robot.

Working principle of IR sensor

Let we have a black line path and white surface area. So, the basic working principle of line follower robot is related to light. We will use behavior of light at the black and white surfaces.

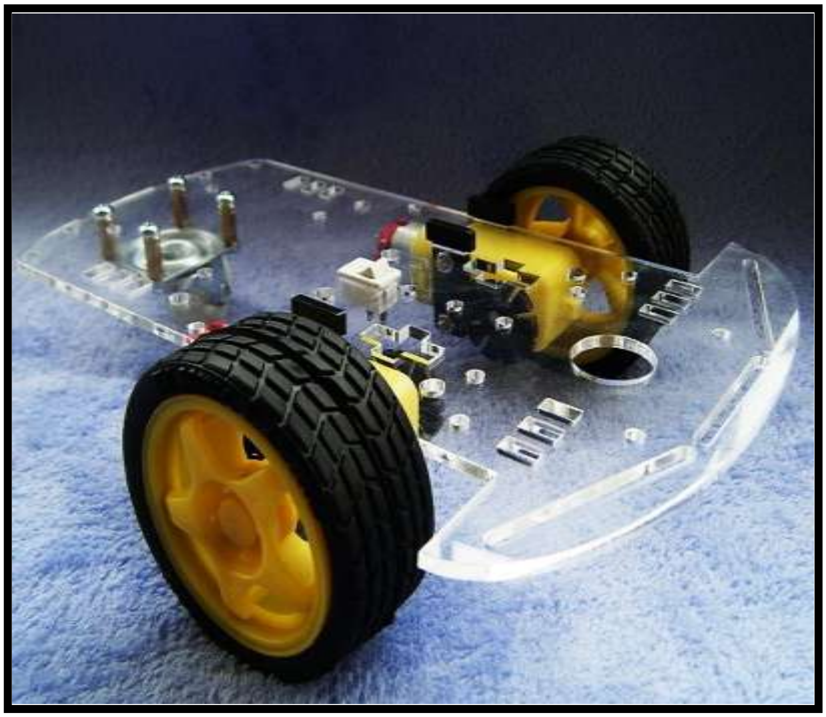
When light or IR rays falls on a white surface it is almost reflected. Now when light or IR rays falls on black surface, it is completely absorbed. This behaviour of light is used in building a line follower robot, and can be seen below in figure.



3.4 WHY 16 X 4 LCD DISPLAY

1. Slim profile
2. No radiation emission from the screen
3. Better under brighter conditions because of anti-glare technology
4. Lighter in weight with respect to screen size
5. Low power consumption

CHAPTER # 4 MECHANICAL DESIGN



4.1 MECHANISM AND FEATURES

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.

4.2 MATERIAL SELECTION

Mechanism, 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. Bults 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

Driver. Bults 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

4.3 PLATFORM DESIGN

So, there are so many designs of which an individual can think of for a line follower robot. Some of them may be available in the market in easy cast or we can design some by yourself. The design maybe in rectangular form, maybe a triangular shape, maybe pentagonal shape or maybe a circular shape. So, it’s just depends upon our feasibility. That is because the design wouldn’t affect our project, this is because our project contains very simple configuration above the base. We just the Lithium Battery cells, a PCB and Motor Driver.

CHAPTER # 5 FIRMWARE DESIGN

5.1 SCHEMATIC COMPONENTS

1. Ultrasonic sensor
2. SD card module
3. Servo motor
4. IR sensors
5. L298 motor driver
6. Tachometer
7. 16x4 LCD display
8. LED bargraph
9. Motors
10. Current sensor
11. Voltage sensor

Potentiometer

5.2 PCB MAKING PROCEDURE

1. To design PCB in proteus there is an ARES designing tool is exist that we will use to design PCB.

2. After that we will create our design through the usage of 2D graphics box Mode, 3. Press at the select layer and now choose board edge.

4. After that now construct the box shape in the working area. After that press mouse button then the color of green line will turn in yellow.

5. ow we can construct our circuit in that rectangle box.

6. After that press at the element and change its location if needed through the use of the rotation tab after that put it in the working area.
7. When you put all elements in the working area make a configuration of all these elements.
8. the location of any element can be varied through pressing at the selection mode and after that chose the element and move it to the new location.
9. After that link, all the elements of circuitry, Choose trach option and now we can vary the track width through choosing C or E option.
10. Choose the width of the screen according to layout of PCB circuitry.
11. After that link all elements used to press at the elements at endpoint there will be green colored line that will show move this line to another component where you have to link.
12. After making link green line will automatically diminish.
13. For a single layer, PCB designing put all elements of circuit at one side and make their links at the other side.
14. For two-layer PCB connection and interlinking of all components can be made at both sides.
15. If there is any fault exist in our design will show in red color circles.
16. To avoiding any fault, we can vary the track path. In the case of 2 layer PCB, we can vary track amongst layers through double-pressing the left button.
17. Red color track seen shown at upper portion and the track of blue-colored shown at lower portion among layers. It is 2 layer PCB.
18. Start tracking on the connections and use T50 for the track size. Also use square connectors of 5mm.When tracking is completed save the project in the similar file that we saved in proteus.
19. We can also select a three-dimensional display for a final look of our project.
20. We can analyze all joints of elements angles placements and other parameters.

5.3 SELECTION OF CONTROLLER

Line following robot detects and follows a line. The line is black line on a white surface and vice versa. The line is sensed by sensor, proximity sensor and IR sensor. The proximity sensor used for path detection and IR sensor used for obstacle detection. These sensors mounted at front end give input to microcontroller (Arduino uno) which decides whether right motor or left motor will move to turn right, left, forward etc

CHAPTER # 6 SIMULATIONS AND PROGRAMMING

We firstly read all the data sheets of our components and noted down the optimal voltage and current foreach of our components. Then tested all the components with DMM separately. Most of the componentswere working properly. The components which were not in working condition were replaced. Then after checking the components we integrated them together. While integration we took a good care that no components should be short or not plugged in wrong pin as it could lead to burring of that component.

6.1 AURDINO COMPLETE CODE:

```
#include <SPI.h>
#include <SD.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SH110X.h>

#define LMS  A4
#define LS   A3
#define CS   A2
#define RS   A1
#define RMS  A0
#define enA  10
#define enB  11
#define MotorAip1  4
#define MotorAip2  5
#define MotorBip1  3
#define MotorBip2  17

#define i2c_Address  0x3c
#define screen_width  128
#define screen_height  64
#define OLED_RESET     -1

#define Current_sensor A6
#define Voltage_sensor A7

#define right_Encoder  27
#define left_Encoder   26
```

```
#define Ultra_Trigger  7
#define Ultra_Echo     6

const int chipSelect = 53;
float current = 0.00;
float voltage = 0.00;
int Obstacle_distance = 0;
int rpm_right = 0;
int rpm_left = 0;
int RPM = 0;

Adafruit_SH1106G display = Adafruit_SH1106G(screen_width,screen_height,&Wire,OLED_RESET);
void setup()
{
  display.begin(i2c_Address,true);
  display.clearDisplay();
  display.setTextSize(1);
  display.setTextColor(SH110X_WHITE);
  display.setCursor(0, 10);
  display.println("Initializing SD card.");
  display.display();
  delay(1000);
  if (!SD.begin(chipSelect))
  {
    display.println("Card failed, or not present");
    display.display();
    delay(2000);
  }
  else
  {
    display.println("Card initialized.");
    display.display();
    delay(3000);
  }

  pinMode(enA, OUTPUT);
  pinMode(enB, OUTPUT);
  pinMode(LMS, INPUT);
  pinMode(LS,  INPUT);
  pinMode(CS,  INPUT);
  pinMode(RS,  INPUT);
  pinMode(RMS, INPUT);
  pinMode(MotorAip1,OUTPUT);
  pinMode(MotorAip2,OUTPUT);
  pinMode(MotorBip1,OUTPUT);
  pinMode(MotorBip2,OUTPUT);
  analogWrite(enA, 100);
```

```
  analogWrite(enB, 100);

  pinMode(Current_sensor, INPUT);
  pinMode(Voltage_sensor, INPUT);
  pinMode(Ultra_Echo, INPUT);
  pinMode(Ultra_Trigger, OUTPUT);
  pinMode(right_Encoder, INPUT_PULLUP);
  pinMode(left_Encoder, INPUT_PULLUP);

  display.clearDisplay();
  display.setTextSize(1.8);
  display.setTextColor(SH110X_WHITE);
  display.setCursor(20, 15);
  display.println("Ready to Go..!!!");
  display.display();
}

void loop()
{
   if(digitalRead(LMS)==1 && digitalRead(LS)==1 && digitalRead(CS)==0 && digitalRead(RS)==1 &&
digitalRead(RMS)==1)
   forward();
   else if(digitalRead(LMS)==1 && digitalRead(LS)==0 && digitalRead(CS)==0 && digitalRead(RS)==0
&& digitalRead(RMS)==1)
   forward();
   else if(digitalRead(LMS)==1 && digitalRead(LS)==0 && digitalRead(CS)==1 && digitalRead(RS)==1
&& digitalRead(RMS)==1)
   left();
   else if(digitalRead(LMS)==0 && digitalRead(LS)==1 && digitalRead(CS)==1 && digitalRead(RS)==1
&& digitalRead(RMS)==1)
   left();
   else if(digitalRead(LMS)==0 && digitalRead(LS)==0 && digitalRead(CS)==1 && digitalRead(RS)==1
&& digitalRead(RMS)==1)
   left();
   else if(digitalRead(LMS)==1 && digitalRead(LS)==0 && digitalRead(CS)==0 && digitalRead(RS)==1
&& digitalRead(RMS)==1)
   left();
   else if(digitalRead(LMS)==0 && digitalRead(LS)==0 && digitalRead(CS)==0 && digitalRead(RS)==1
&& digitalRead(RMS)==1)
   left();
   else if(digitalRead(LMS)==1 && digitalRead(LS)==1 && digitalRead(CS)==1 && digitalRead(RS)==0
&& digitalRead(RMS)==1)
   right();
   else if(digitalRead(LMS)==1 && digitalRead(LS)==1 && digitalRead(CS)==1 && digitalRead(RS)==1
&& digitalRead(RMS)==0)
   right();
   else if(digitalRead(LMS)==1 && digitalRead(LS)==1 && digitalRead(CS)==1 && digitalRead(RS)==0
&& digitalRead(RMS)==0)
```

```
   right();
   else if(digitalRead(LMS)==1 && digitalRead(LS)==1 && digitalRead(CS)==0 && digitalRead(RS)==0
&& digitalRead(RMS)==1)
   right();
   else if(digitalRead(LMS)==1 && digitalRead(LS)==1 && digitalRead(CS)==0 && digitalRead(RS)==0
&& digitalRead(RMS)==0)
   right();
   else
   stop();
}

void  Line_follower()
{
   if(digitalRead(LMS)==1 && digitalRead(LS)==1 && digitalRead(CS)==0 && digitalRead(RS)==1 &&
digitalRead(RMS)==1)
   forward();
   else if(digitalRead(LMS)==1 && digitalRead(LS)==0 && digitalRead(CS)==0 && digitalRead(RS)==0
&& digitalRead(RMS)==1)
   forward();
   else if(digitalRead(LMS)==1 && digitalRead(LS)==0 && digitalRead(CS)==1 && digitalRead(RS)==1
&& digitalRead(RMS)==1)
   left();
   else if(digitalRead(LMS)==0 && digitalRead(LS)==1 && digitalRead(CS)==1 && digitalRead(RS)==1
&& digitalRead(RMS)==1)
   left();
   else if(digitalRead(LMS)==0 && digitalRead(LS)==0 && digitalRead(CS)==1 && digitalRead(RS)==1
&& digitalRead(RMS)==1)
   left();
   else if(digitalRead(LMS)==1 && digitalRead(LS)==0 && digitalRead(CS)==0 && digitalRead(RS)==1
&& digitalRead(RMS)==1)
   left();
   else if(digitalRead(LMS)==0 && digitalRead(LS)==0 && digitalRead(CS)==0 && digitalRead(RS)==1
&& digitalRead(RMS)==1)
   left();
   else if(digitalRead(LMS)==1 && digitalRead(LS)==1 && digitalRead(CS)==1 && digitalRead(RS)==0
&& digitalRead(RMS)==1)
   right();
   else if(digitalRead(LMS)==1 && digitalRead(LS)==1 && digitalRead(CS)==1 && digitalRead(RS)==1
&& digitalRead(RMS)==0)
   right();
   else if(digitalRead(LMS)==1 && digitalRead(LS)==1 && digitalRead(CS)==1 && digitalRead(RS)==0
&& digitalRead(RMS)==0)
   right();
   else if(digitalRead(LMS)==1 && digitalRead(LS)==1 && digitalRead(CS)==0 && digitalRead(RS)==0
&& digitalRead(RMS)==1)
   right();
   else if(digitalRead(LMS)==1 && digitalRead(LS)==1 && digitalRead(CS)==0 && digitalRead(RS)==0
&& digitalRead(RMS)==0)
```

```
  right();
  else
  stop();
}

void forward()
{

  digitalWrite(MotorAip1, HIGH);
  digitalWrite(MotorAip2, LOW);
  digitalWrite(MotorBip1, LOW);
  digitalWrite(MotorBip2, HIGH);
}

void left()
{


  digitalWrite(MotorAip1, HIGH);
  digitalWrite(MotorAip2, LOW);
  digitalWrite(MotorBip1, LOW);
  digitalWrite(MotorBip2, LOW);
}

void right()
{


  digitalWrite(MotorAip1, LOW);
  digitalWrite(MotorAip2, LOW);
  digitalWrite(MotorBip1, LOW);
  digitalWrite(MotorBip2, HIGH);
}

void reverse()
{
  digitalWrite(MotorAip1, LOW);
  digitalWrite(MotorAip2, HIGH);
  digitalWrite(MotorBip1, HIGH);
  digitalWrite(MotorBip2, LOW);
}

void stop()
{
  digitalWrite(MotorAip1, LOW);
  digitalWrite(MotorAip2, LOW);
  digitalWrite(MotorBip1, LOW);
  digitalWrite(MotorBip2, LOW);
```

```
    voltage = Voltage();
    current = Current();
    Obstacle_distance = Ultrasonic();
    display_data();

}

float Current()
{
  unsigned int x=0;
  float AcsValue=0.0,Samples=0.0,AvgAcs=0.0,AcsValueF=0.0;
    for (int x = 0; x < 150; x++)
    {
      AcsValue = analogRead(Current_sensor);
      Samples = Samples + AcsValue;
      delay (3);
    }
      AvgAcs=Samples/150.0;
      AcsValueF = (-2.5 + (AvgAcs * (5.0 / 1024.0)) )/0.185;
      delay(50);
      return AcsValueF;
}

float Voltage()
{
    float adc_voltage = 0.0;
    float in_voltage = 0.0;
    float R1 = 30000.0;
    float R2 = 7500.0;
    float ref_voltage = 5.0;
    int adc_value = 0;

   adc_value = analogRead(Voltage_sensor);
   adc_voltage  = (adc_value * ref_voltage) / 1024.0;
   in_voltage = (adc_voltage / (R2/(R1+R2)))-3.23 ;
   return in_voltage;
}

int Ultrasonic()
{
  long Duration;
  int Distance;
  digitalWrite(Ultra_Trigger, LOW);
  delayMicroseconds(2);
  digitalWrite(Ultra_Trigger, HIGH);
  delayMicroseconds(10);
  digitalWrite(Ultra_Trigger, LOW);
  Duration = pulseIn(Ultra_Echo, HIGH);
```

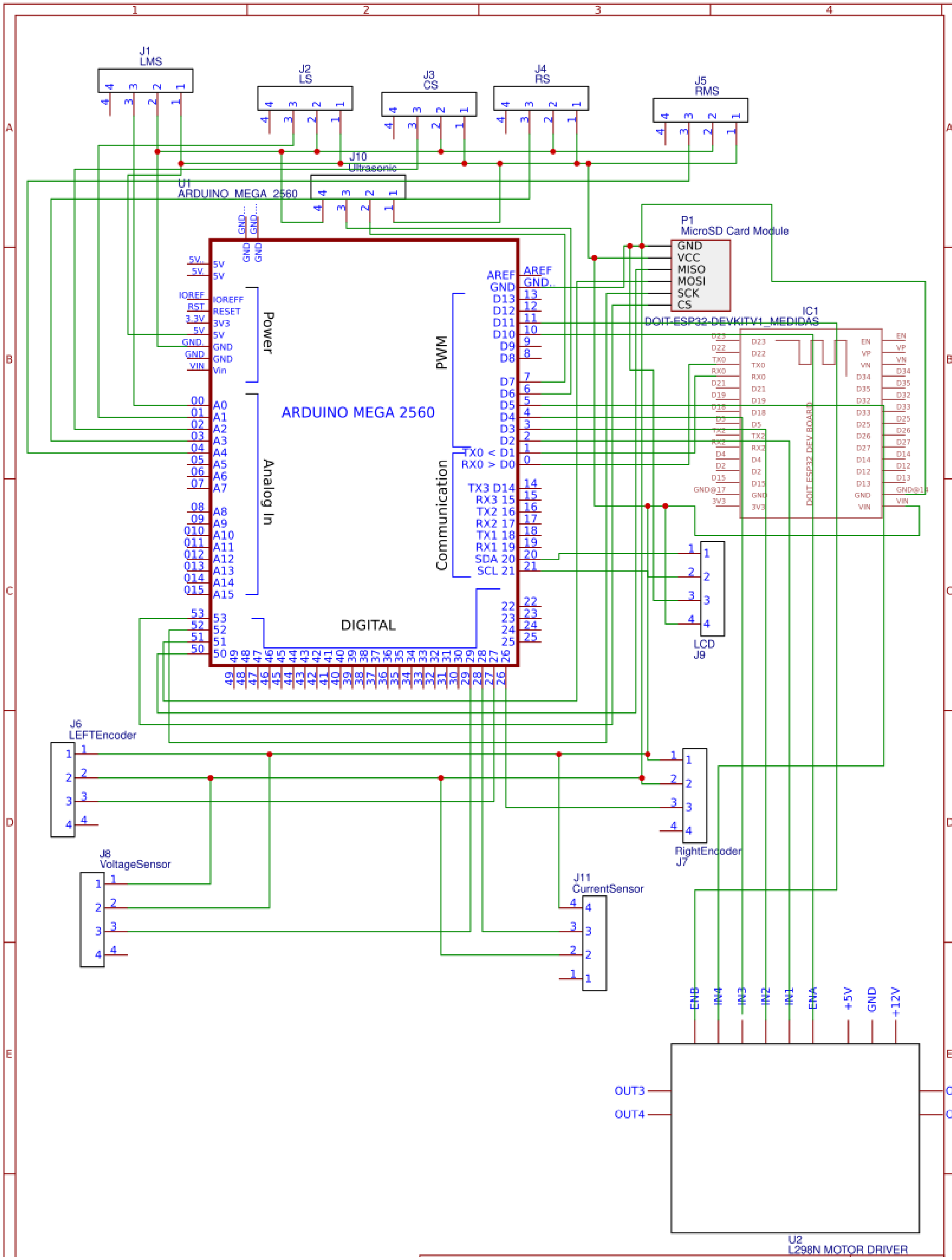
```
 Distance = Duration * 0.034 / 2;
 return Distance;
}

float Encoder_right()
{
   unsigned long start_time = 0;
   unsigned long end_time = 0;
   int steps=0;
   float steps_old=0;
   float temp=0;
   float rpm=0;
   start_time=millis();
   end_time=start_time+100;
 while(millis()<end_time)
 {
  if(digitalRead(right_Encoder))
  {
   steps=steps+1;
   while(digitalRead(right_Encoder));
  }
 }
   temp=steps-steps_old;
   steps_old=steps;
   rpm=(temp/20)*10*60;
   return rpm;
}

float Encoder_left()
{
   unsigned long start_time = 0;
   unsigned long end_time = 0;
   int steps=0;
   float steps_old=0;
   float temp=0;
   float rpm=0;
   start_time=millis();
   end_time=start_time+100;
 while(millis()<end_time)
 {
  if(digitalRead(left_Encoder))
  {
   steps=steps+1;
   while(digitalRead(left_Encoder));
  }
 }
   temp=steps-steps_old;
   steps_old=steps;
```

```
  rpm=(temp/20)*10*60;
  return rpm;
}
int average_rpm(int rpm1, int rpm2)
{
 return (rpm1+rpm2)/2;
}

void SD_card()
{
 // make a string for assembling the data to log:
 String dataString = "";

 // read three sensors and append to the string:
 for (int analogPin = 0; analogPin < 3; analogPin++) {
  int sensor = analogRead(analogPin);
  dataString += String(sensor);
  if (analogPin < 2) {
   dataString += ",";
  }
 }

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

 // if the file is available, write to it:
 if (dataFile) {
  dataFile.println(dataString);
  dataFile.close();
  // print to the serial port too:
  Serial.println(dataString);
 }
 // if the file isn't open, pop up an error:
 else {
  Serial.println("error opening datalog.txt");
 }
}
void forward_indication()
{
 display.clearDisplay();
 display.setCursor(0, 10);
 display.writeLine(64,5,54,20,SH110X_WHITE);
 display.writeLine(64,5,74,20,SH110X_WHITE);
 display.writeLine(64,10,54,25,SH110X_WHITE);
 display.writeLine(64,10,74,25,SH110X_WHITE);
 display.writeLine(64,15,54,30,SH110X_WHITE);
 display.writeLine(64,15,74,30,SH110X_WHITE);
```

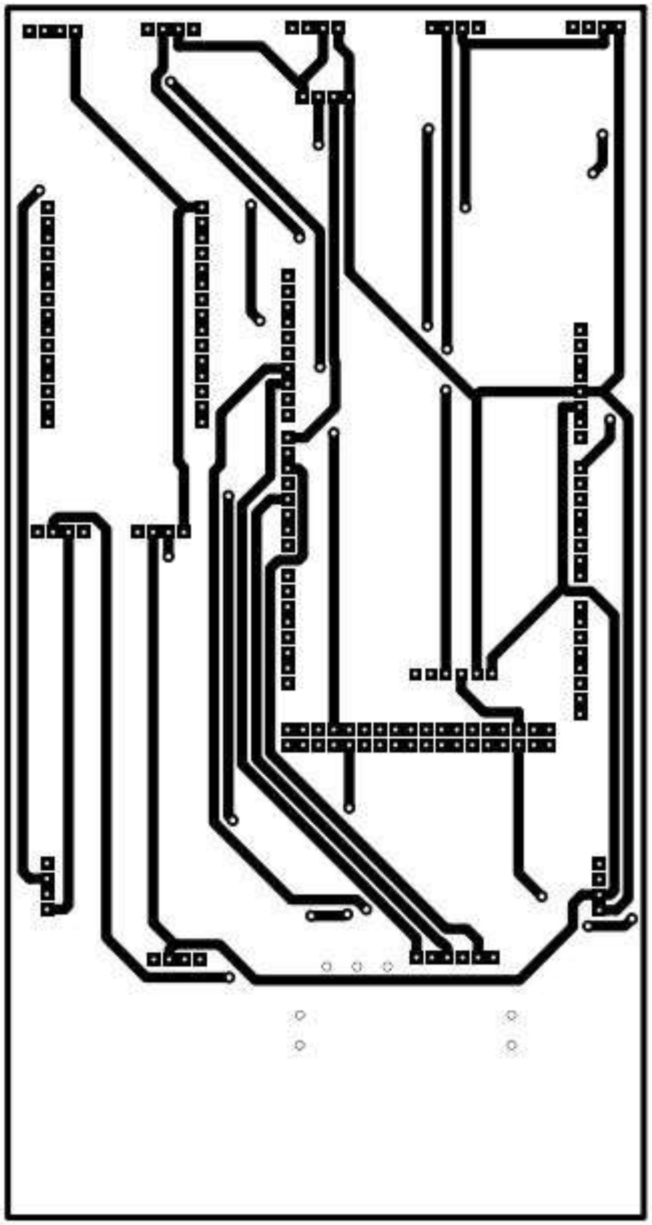
```
  display.display();
}
void left_indication()
{
  display.clearDisplay();
  display.setCursor(0, 10);
  display.writeLine(64,5,49,15,SH110X_WHITE);
  display.writeLine(64,25,49,15,SH110X_WHITE);
  display.writeLine(59,5,44,15,SH110X_WHITE);
  display.writeLine(59,25,44,15,SH110X_WHITE);
  display.writeLine(54,5,39,15,SH110X_WHITE);
  display.writeLine(54,25,39,15,SH110X_WHITE);
  display.display();
}
void right_indication()
{
  display.clearDisplay();
  display.setCursor(0, 10);
  display.writeLine(64,5,79,15,SH110X_WHITE);
  display.writeLine(64,25,79,15,SH110X_WHITE);
  display.writeLine(69,5,84,15,SH110X_WHITE);
  display.writeLine(69,25,84,15,SH110X_WHITE);
  display.writeLine(74,5,89,15,SH110X_WHITE);
  display.writeLine(74,25,89,15,SH110X_WHITE);
  display.display();
}
void display_RPM(int value)
{
  display.setTextSize(2.2);
  display.setTextColor(SH110X_WHITE);
  display.setCursor(35, 40);
  display.print(value);
  display.setTextSize(1);
  display.setCursor(100, 40);
  display.print("rpm");
  display.display();
}

void display_data()
{
  display.clearDisplay();
  display.setCursor(10, 20);
  display.setTextSize(0.5);
  display.setTextColor(SH110X_WHITE);
  display.print("Voltage: ");
  display.print(voltage, 2);
  display.print(" V");
  display.setCursor(10, 30);
```

```
  display.print("Current: ");
  display.print(current, 2);
  display.print(" mA");
  display.setCursor(10, 40);
  display.print("Obstacle: ");
  display.print(Obstacle_distance, 10);
  display.print(" cm");
  display.display();
}
```

6.2 PCB SCHEMATIC

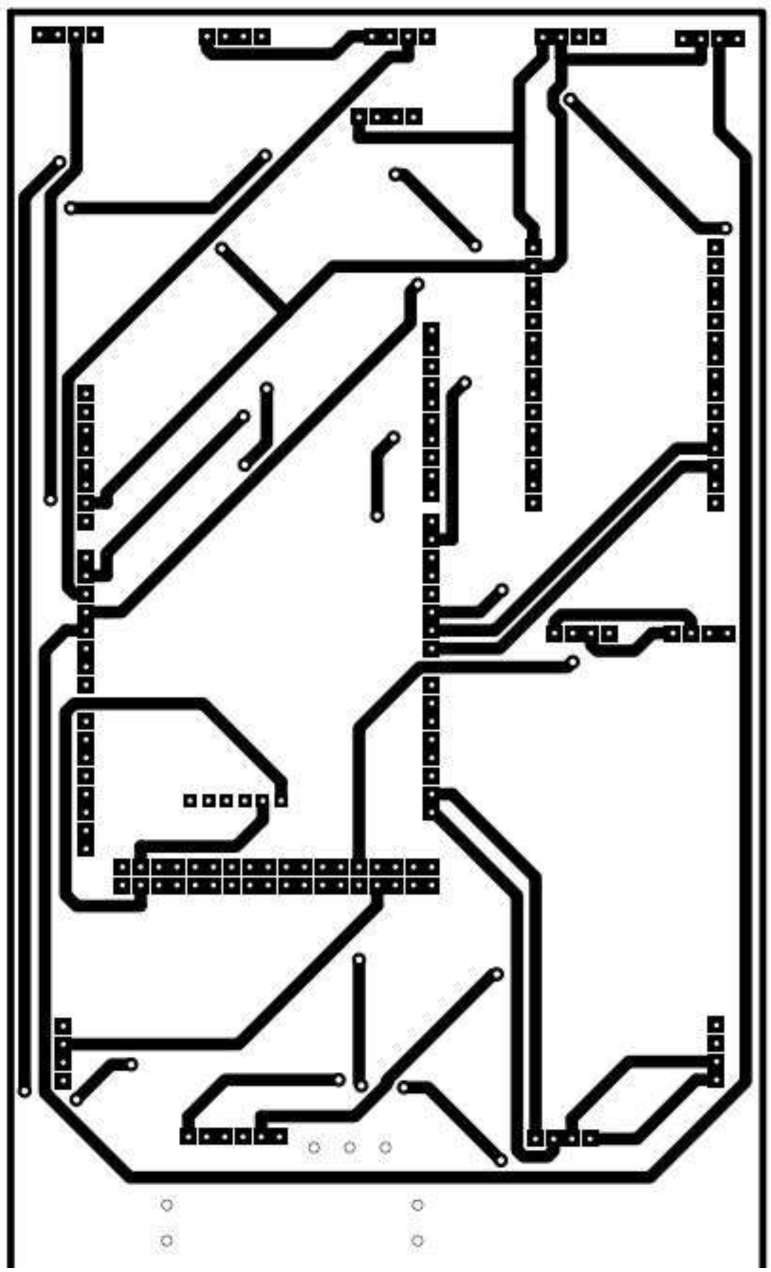


6.3 PCB LAYOUTS:

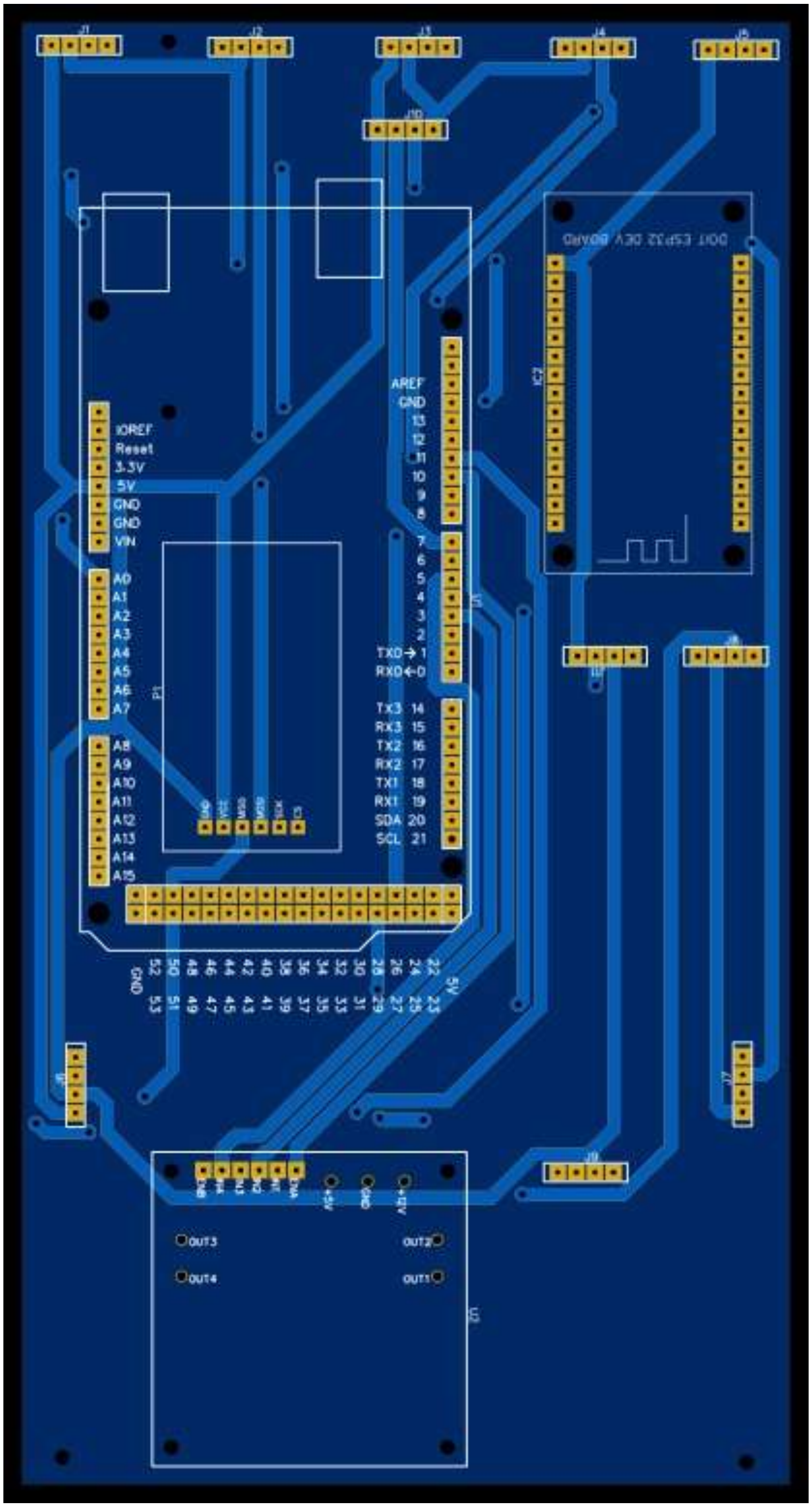


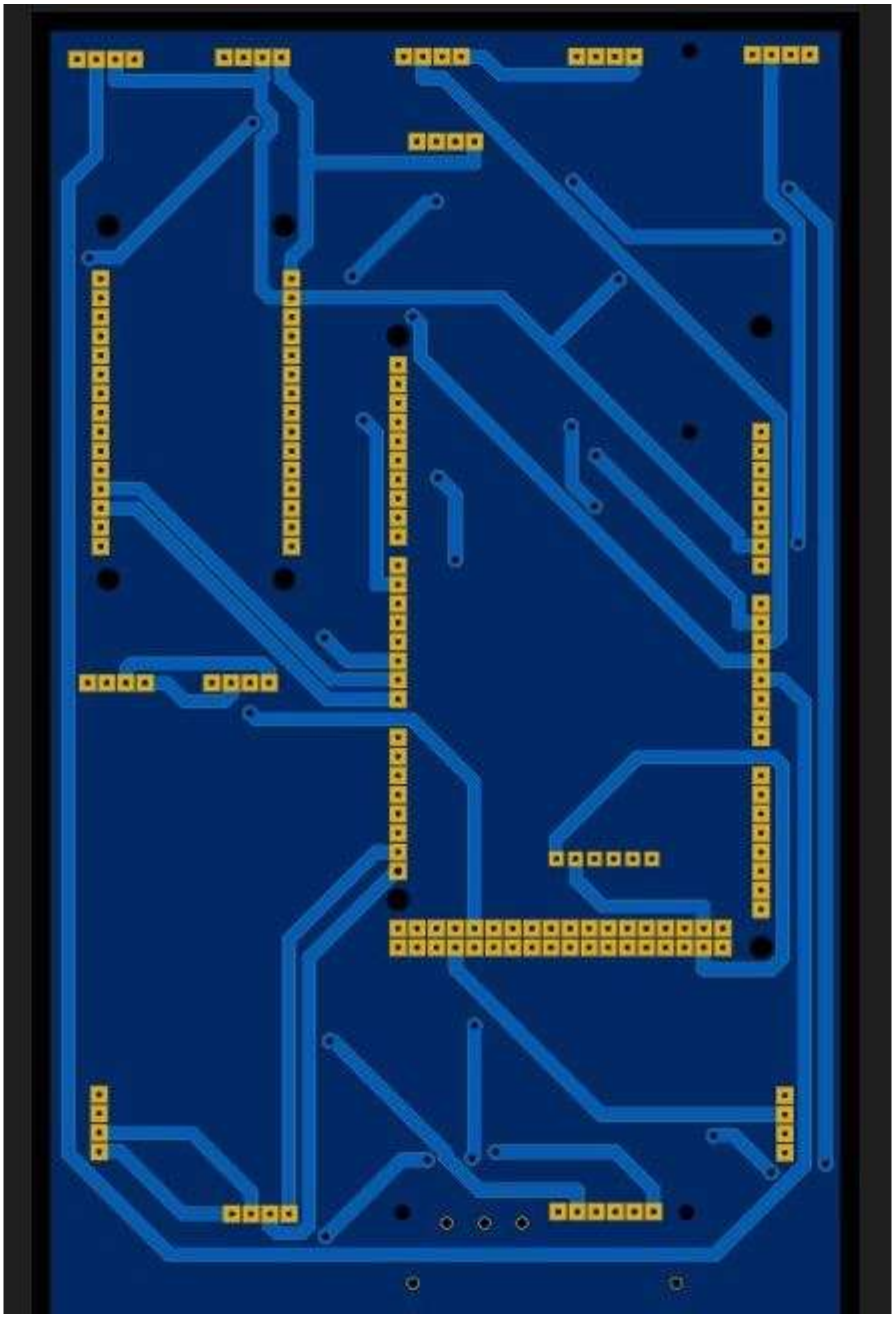
TOP:

BOTTOM:



VISULAIZERS:


J1
J2
J3
J4
J5
J10
DOIT ESP32 DEV BOARD
IC2
IOREF
Reset
3.3V
5V
GND
GND
VIN
A0
A1
A2
A3
A4
A5
A6
A7
A8
A9
A10
A11
A12
A13
A14
A15
P1
AREF
GND
13
12
11
10
9
8
7
6
5
4
3
2
TX0→1
RX0←0
TX3 14
RX3 15
TX2 16
RX2 17
TX1 18
RX1 19
SDA 20
SCL 21
OUT3
OUT4
OUT2
OUT1
+5V
GND
+12V



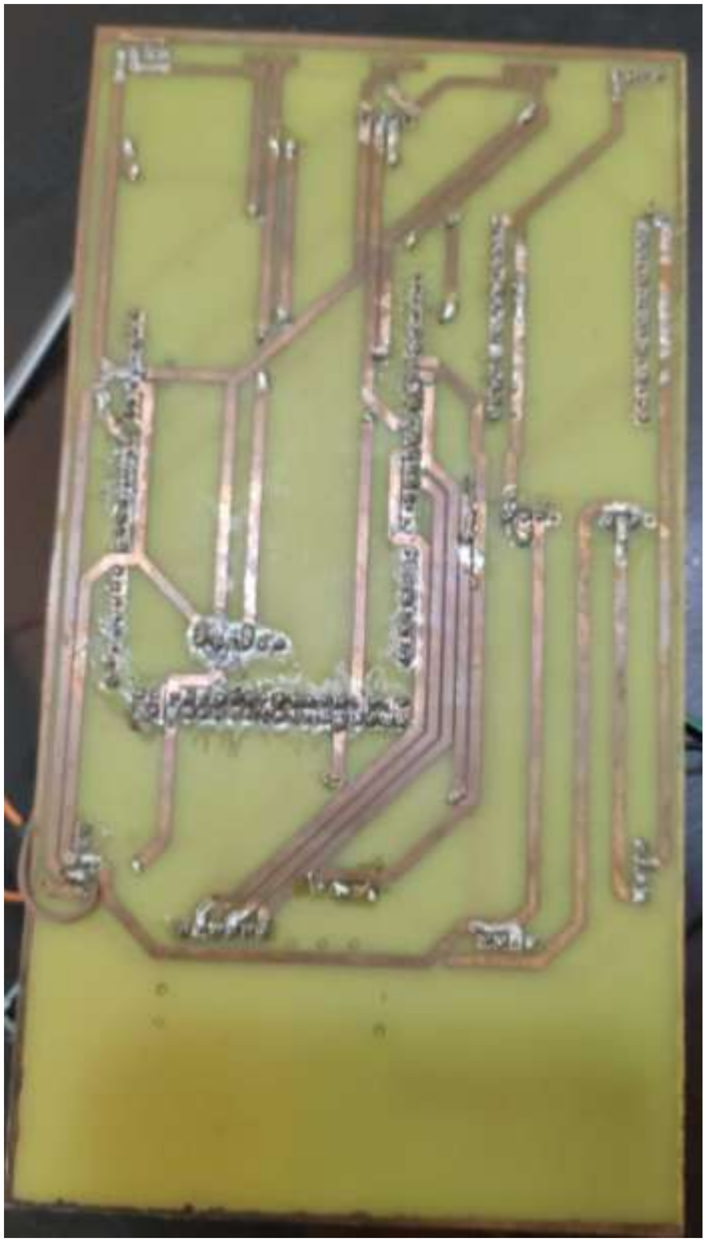
CHAPTER # 7 HARDWARE IMPLIMENTATION

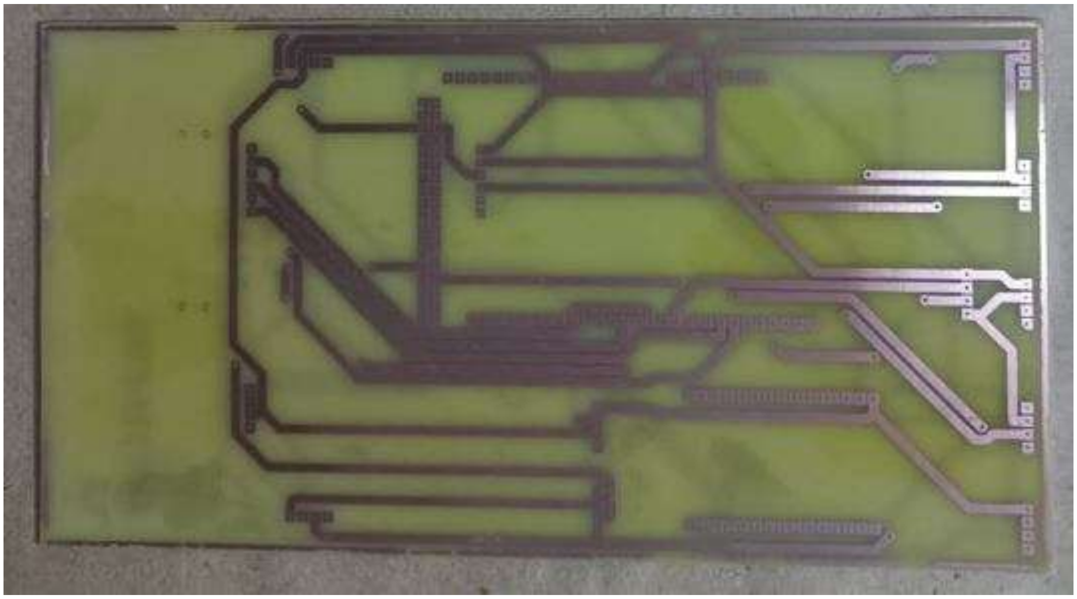
7.1 PROCESS:

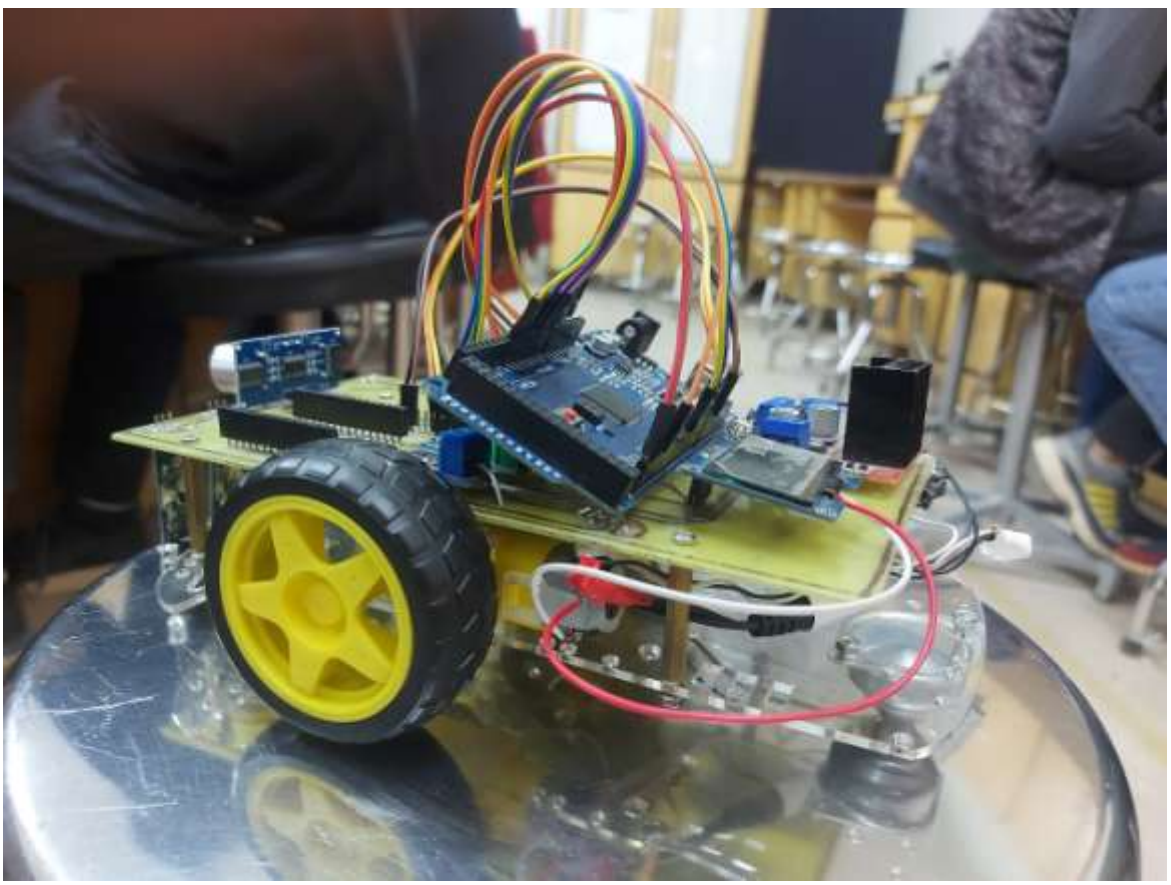
1. For making PCB we have to do print our final PCB layout on sticky paper.

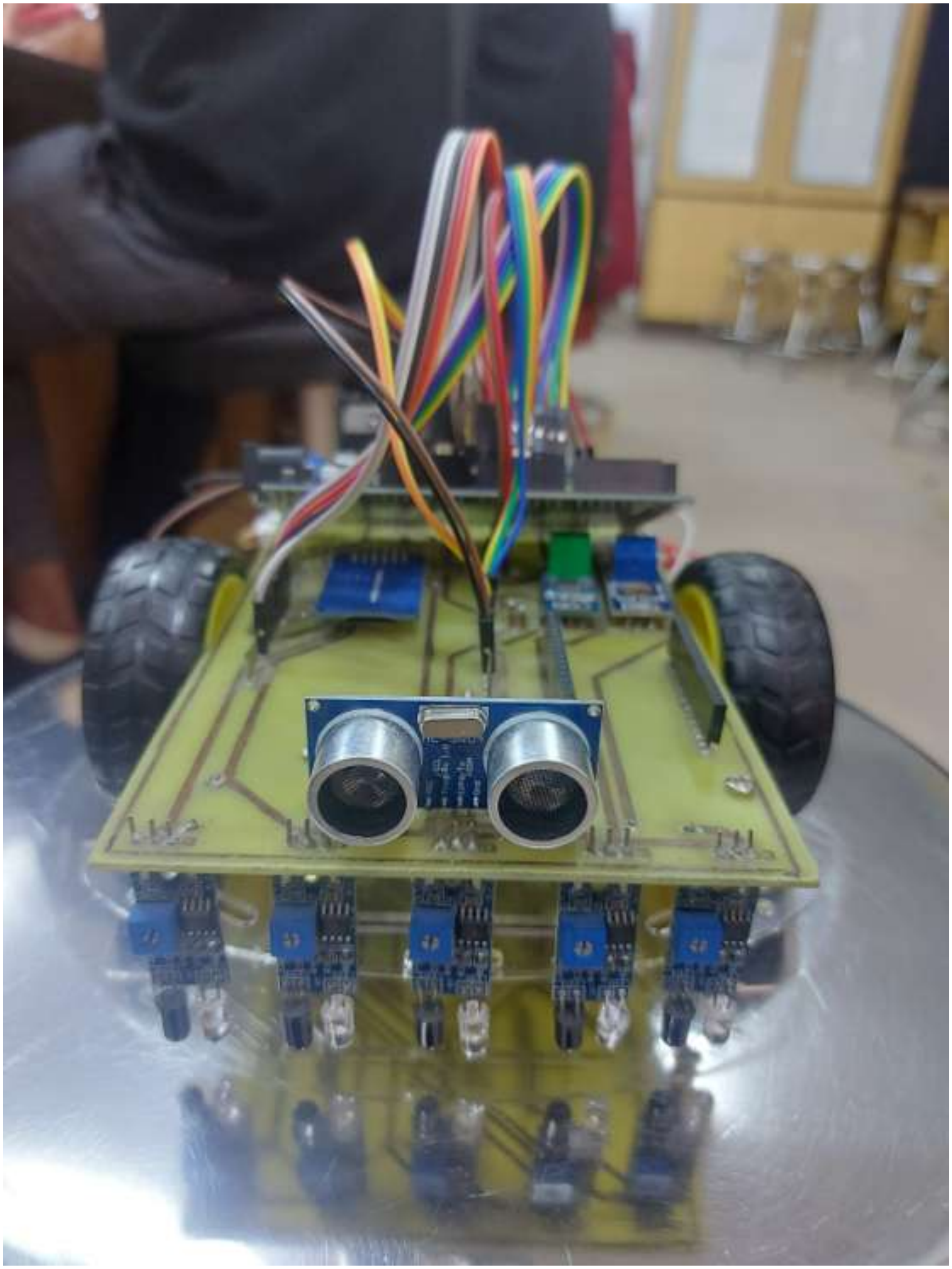
2. You should take the mirror print out.
3. Select the output in black both from the PCB design software and the printer driver settings.
4. Cut the copper board according to the size of the layout using a hacksaw or a cutter.
5. Next, rub the copper side of the PCB using steel wool or abrasive sponge scrubs. This removes the top oxide layer of copper as well as the photoresist layer. Sanded surfaces also allow the image from the paper to stick better.
6. Transfer the printed image (taken from a laser printer) from the photo paper to the board. Make sure to flip top layer horizontally. Put the copper surface of the board on the printed layout. Ensure that the board is aligned correctly along the borders of the printed layout and use tape to hold the board and the printed paper in the correct position.
7. Using the circuit as a reference, draw a basic sketch on the copper plate with a pencil. Once your sketch looks good, trace over it with a permanent black marker.
8. After printing on glossy paper, we iron it image side down to the copper side, then heat up the electric iron to the maximum temperature.
9. Put the board and photo paper arrangement on a clean wooden table (covered with a table cloth) with the back of the photo paper facing you.
10. Using pliers or a spatula, hold one end and keep it steady. Then put the hot iron on the other end for about 10 seconds. Now, iron the photo paper all along using the tip while applying a little pressure for about 5 to 15 mins.
11. Pay attention to the edges of the board – you need to apply pressure and do the ironing slowly.
12. Doing a long hard press seems to work better than moving the iron around.
13. The heat from the iron transfers the ink printed on the glossy paper to the copper plate.

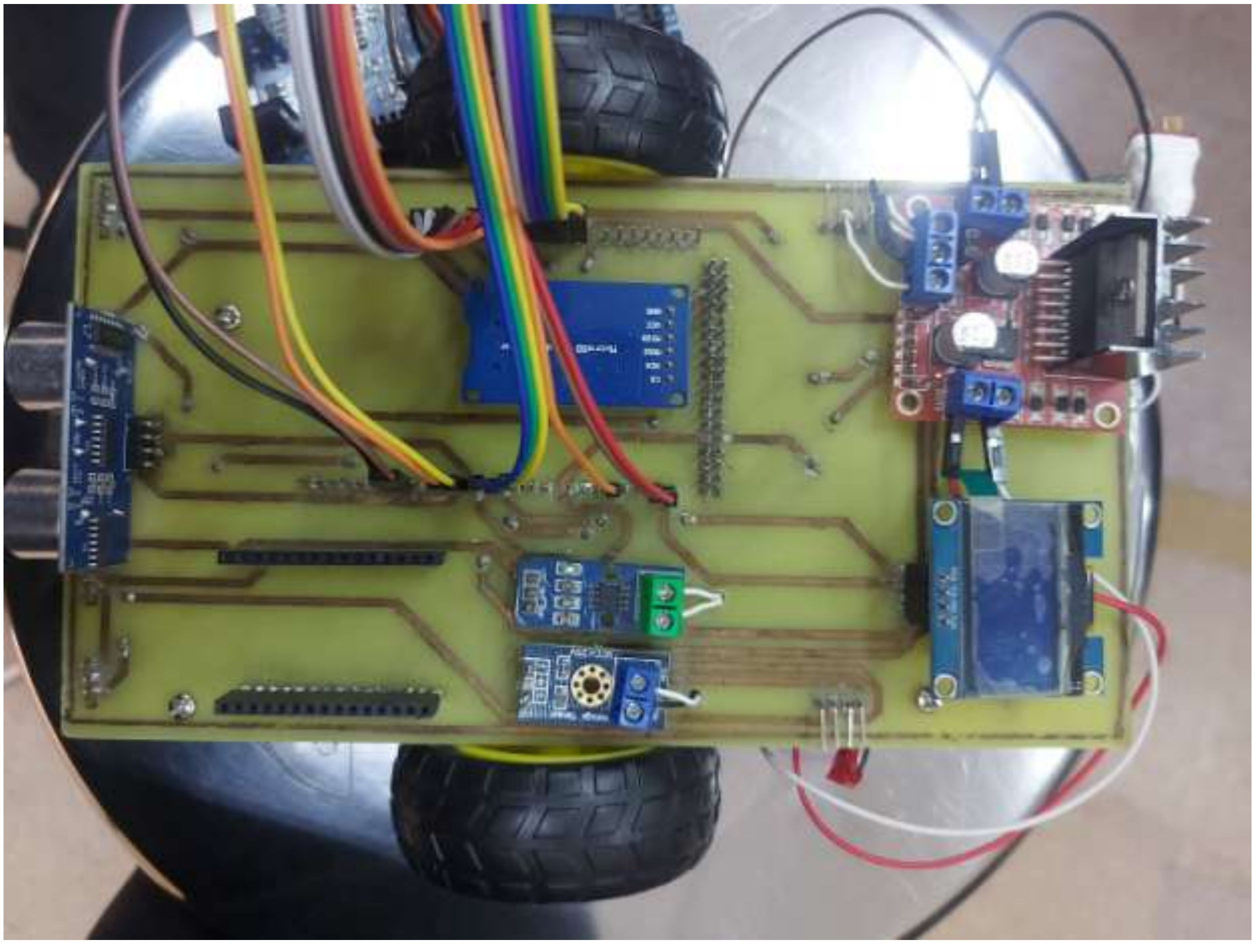
7.2 HARDWARE IMAGES











CHAPTER # 8 PROJECT MANAGEMENT

8.1 FINAL COST

The Component cost is something that may be so challenging for some students. As Pakistan is a poor 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 somebody finds 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 all ideal. And it's easy to simulate, patch things. But it's when it comes to real life.

8.2 RISK MANAGEMENT

In project management, it is the practice of identifying, evaluating, and preventing or risks to a project that have the potential to impact the desired outcomes. Project managers are typically responsible for overseeing the risk management process throughout the duration of a given project.

During this project, we have gone through several risks like PCB etching was a huge risk in which if etching is not done correctly, we would have to get a new one from store which was a big trouble. Another one is drilling of the PCB holes, soldering.

As our PCB was double layer so PCB alignment was a big game for us. Many components were not available in proteus, so we made them for ourselves.

But in all these risks we have gone through, by the grace of Allah almighty, we have got success in such pressurized environment.

8.3 CHALLENGES

Digital IR sensor works if we have only black or white surface area. The simplest case will be if we use 2 IR sensor and can work if we have black path with white surface. But to get 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.

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 at single place. That's challenging, and we will try to see if what can we do with these conditions.

8.4 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 real life.
3. If we have given any simple real-life problem, then how to tickle them with microcontrollers 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 works and how to use and implement them in real time life. We will learn, how quantity of 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 processes to make the required and desires product.

CHAPTER # 9 INDIVIDUAL ROLES

Team Member Name	Role
Izza nadeem	**Pcb Design and Proteus**
Salman Ali	**Trouble Shooting**
Rabia	**Coding and Report**
Muhammad Sohaib	**Literature Review**
Abdullah Ilyas	**Hardware Patching**

9.1 INDIVIDUAL ROLES IN PROJECT AND TASK PERFORMANCE

1. Creating truth table on excel file
2. Review and revision of truth table with further iterations and improvement
3. Software programming contribution on ide
4. Proteus file placement contribution
5. Hardware and software interfacing of module
6. PCB trouble shooting
7. Bread board wiring
8. Rawalpindi tour for component
9. Karachi company tour for remaining component
10. G-11 tour for remaining component
11. PCB printing contribution
12. Robot testing on arena

CONCLUSION

The overall expected results were to make the robot follow the line by using the sensors attached to it and to display voltage and current and tachometer sensors readings on LCD. 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 of guidelines 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.

APPENDIX

L298

DUAL FULL-BRIDGE DRIVER

- OPERATING SUPPLY VOLTAGE UP TO 46 V
- TOTAL DC CURRENT UP TO 4 A
- LOW SATURATION VOLTAGE
- OVERTEMPERATURE PROTECTION
- LOGICAL "0" INPUT VOLTAGE UP TO 1.5 V (HIGH NOISE IMMUNITY)

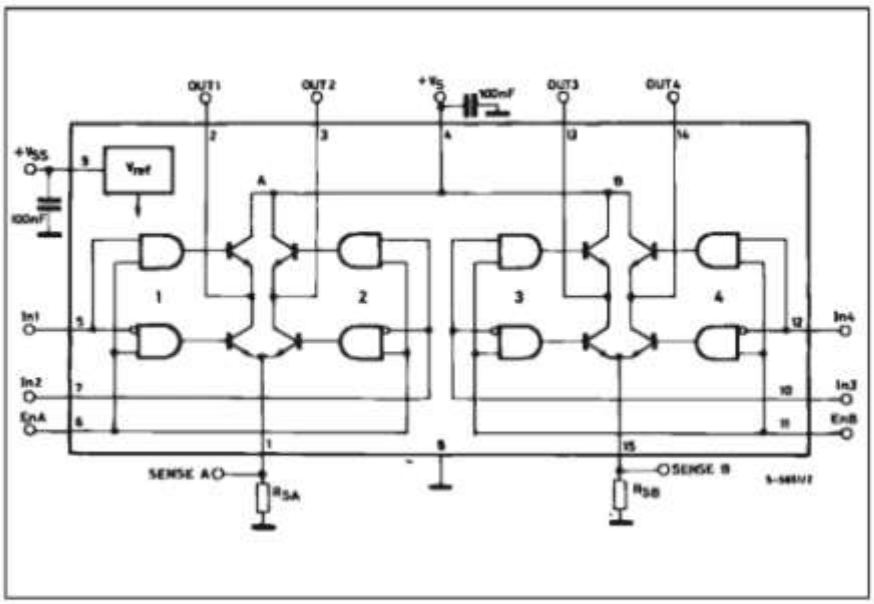
Multiwatt15 PowerSO20

ORDERING NUMBERS : L298N (Multiwatt Vert.)
L298HN (Multiwatt Horiz.)
L298P (PowerSO20)

DESCRIPTION

The L298 is an integrated monolithic circuit in a 15-lead Multiwatt and PowerSO20 packages. It is a high voltage, high current dual full-bridge driver designed to accept standard TTL logic levels and drive inductive loads such as relays, solenoids, DC and stepping motors. Two enable inputs are provided to enable or disable the device independently of the input signals. The emitters of the lower transistors of each bridge are connected together and the corresponding external terminal can be used for the connection of an external sensing resistor. An additional supply input is provided so that the logic works at a lower voltage.

BLOCK DIAGRAM



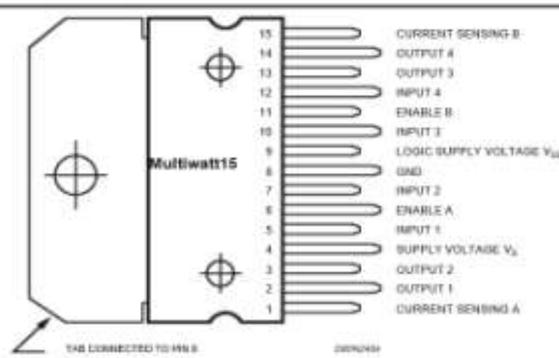
January 2000 1/13

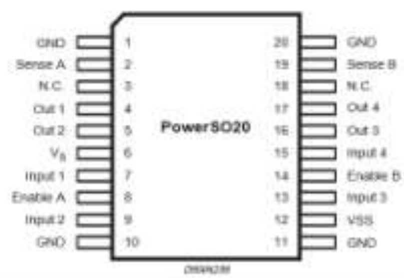
L298

ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
V_S	Power Supply	50	V
V_{SS}	Logic Supply Voltage	7	V
V_I,V_{en}	Input and Enable Voltage	–0.3 to 7	V
I_O	Peak Output Current (each Channel) – Non Repetitive (t = 100μs) –Repetitive (80% on –20% off; t_{on} = 10ms) –DC Operation	 3 2.5 2	 A A A
V_{sens}	Sensing Voltage	–1 to 2.3	V
P_{tot}	Total Power Dissipation (T_{case} = 75°C)	25	W
T_{op}	Junction Operating Temperature	–25 to 130	°C
T_{stg}, T_j	Storage and Junction Temperature	–40 to 150	°C

PIN CONNECTIONS (top view)





THERMAL DATA

Symbol	Parameter		PowerSO20	Multiwatt15	Unit
$R_{th\ j-case}$	Thermal Resistance Junction-case	Max.	–	3	°C/W
$R_{th\ j-amb}$	Thermal Resistance Junction-ambient	Max.	13 (*)	35	°C/W

(*) Mounted on aluminum substrate

2/13

Description (continued)

loss. The thickness of the copper conductor allows survival of the device at up to 5× overcurrent conditions. The terminals of the conductive path are electrically isolated from the sensor leads (pins 5 through 8). This allows the ACS712 current sensor to be used in applications requiring electrical isolation without the use of opto-isolators or other costly isolation techniques.

The ACS712 is provided in a small, surface mount SOIC8 package. The leadframe is plated with 100% matte tin, which is compatible with standard lead (Pb) free printed circuit board assembly processes. Internally, the device is Pb-free, except for flip-chip high-temperature Pb-based solder balls, currently exempt from RoHS. The device is fully calibrated prior to shipment from the factory.

Selection Guide

Part Number	Packing*	T_{OP} (°C)	Optimized Range, I_P (A)	Sensitivity, Sens (Typ) (mV/A)
ACS712ELCTR-05B-T	Tape and reel, 3000 pieces/reel	–40 to 85	±5	185
ACS712ELCTR-20A-T	Tape and reel, 3000 pieces/reel	–40 to 85	±20	100
ACS712ELCTR-30A-T	Tape and reel, 3000 pieces/reel	–40 to 85	±30	66

*Contact Allegro for additional packing options.

Absolute Maximum Ratings

Characteristic	Symbol	Notes	Rating	Units
Supply Voltage	V_{CC}		8	V
Reverse Supply Voltage	V_{RCC}		–0.1	V
Output Voltage	V_{IOUT}		8	V
Reverse Output Voltage	V_{RIOUT}		–0.1	V
Output Current Source	$I_{IOUT(Source)}$		3	mA
Output Current Sink	$I_{IOUT(Sink)}$		10	mA
Overcurrent Transient Tolerance	I_P	100 total pulses, 250 ms duration each, applied at a rate of 1 pulse every 100 seconds.	60	A
Maximum Transient Sensed Current	I_R(max)	Junction Temperature, $T_J < T_J$(max)	60	A
Nominal Operating Ambient Temperature	T_A	Range E	–40 to 85	°C
Maximum Junction	T_J(max)		165	°C
Storage Temperature	T_{stg}		–65 to 170	°C

TÜV America
Certificate Number:
U8V 06 05 54214 010

Parameter	Specification
Fire and Electric Shock	CAN/CSA-C22.2 No. 60950-1-03 UL 60950-1:2003 EN 60950-1:2001

Arduino IR Infrared Obstacle Avoidance Sensor Module

The sensor module adaptable to ambient light, having a pair of infrared emitting and receiving tubes, transmitting tubes emit infrared certain frequency, when the direction of an obstacle is detected (reflection surface), the infrared reflected is received by the reception tube, After a comparator circuit processing, the green light is on, but the signal output interface output digital signal (a low-level signal), you can adjust the detection distance knob potentiometer, the effective distance range of 2 ~ 30cm, the working voltage of 3.3V- 5V. Detection range of the sensor can be obtained by adjusting potentiometer, with little interference, easy to assemble, easy to use features, can be widely used in robot obstacle avoidance, avoidance car, line count, and black and white line tracking and many other occasions.

Specification

1. When the module detects an obstacle in front of the signal, the green indicator lights on the board level, while the OUT port sustained low signal output, the module detects the distance 2 ~ 30cm, detection angle 35 °, the distance can detect potential is adjusted clockwise adjustment potentiometer, detects the distance increases; counter clockwise adjustment potentiometer, reducing detection distance.
2. The sensor active infrared reflection detection, target reflectivity and therefore the shape is critical detection distance. Where the minimum detection distance black, white, maximum; small objects away from a small area, a large area from the Grand.
3. The sensor module output port OUT port can be directly connected to the microcontroller IO can also be directly drive a 5V relay; Connection: VCC-VCC; GND-GND; OUT-IO
4. Comparators LM393, stable;

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.