LINE FOLLOWING ROBOT

Muhammad Nouman (221729)
Mirza Faizaan Baig (221708)

BE MECHATRONICS (Session 2022-2026)

Project Supervisor

ENGR. UMER FAROOQ

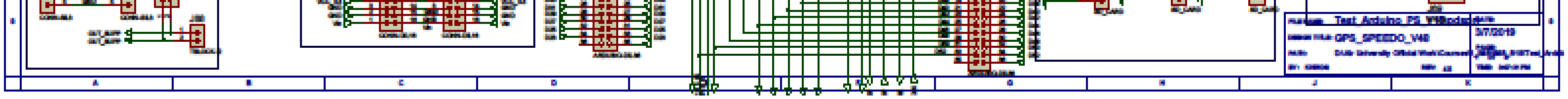
Designation:

Lecturer

DEPARTMENT OF MECHATRONICS ENGINEERING

FACULTY OF ENGINEERING

AIR UNIVERSITY, ISLAMABAD



Chapter 1- Preliminaries

1.1 Proposal

1.2 Initial feasibility

1.3 Comparison table of at least 2 similar products

1.4 Technical Standards (for example Speed of a standard automatic sliding door)

1.5 Team Roles & Details

1.6 Work Break down structure

1.7 Gantt Chart

1.8 Estimated budgets

Chapter 2-Project Conception

2.1 Introduction

2.2 Literature Review (At least five research paper and three commercial products, two patents (WPO))

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

2.4 Project Development Process (Design Process of Mechatronics System)

2.5 Basic block diagrams of whole system and subcomponents in Draw.io or similar tool

2.6 Deliverable with complete Speciation sheet with discussion

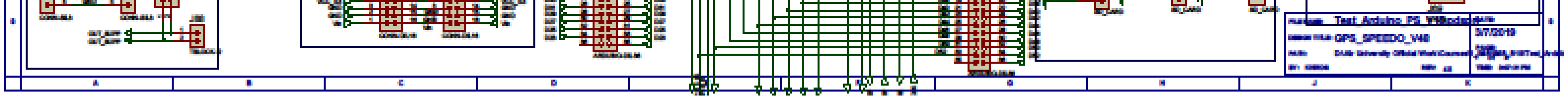
Chapter 3-Product Design(Not required for ME&S add only relevant parts)

3.1 System Consideration for the Design

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

3.3 Free body Diagrams (with Clearly mark Center of Gravity)

3.4 Design Calculation (Dynamic & Static Calculation)



3.5 Compute forces and torques in SI unit and highlight them with proper equation number

3.6 Committed accuracy and resolution system

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

Chapter 4- Mechanical Design (Show picture of BASE)

4.1 Mechanism selection (Use the document usa_tech_calculation.pdf to refer your selection and its calculation example rack & pinion, pulley etc.)

4.2 Platform Design

4.3 Material Selection and choices

4.4 3D CAD design and Analysis Screen shots and explanation

4.5 Factor of safety and maximum stress analysis(simulation)

4.6 Actuators with speciation and datasheet

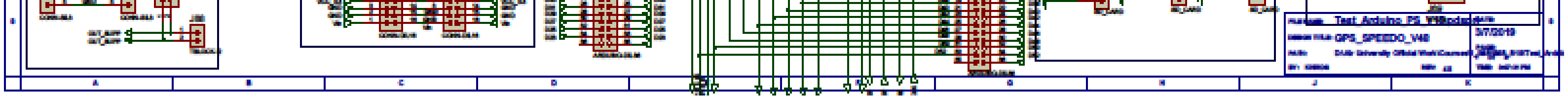
4.7 Deliverable of complete CAD with discussion

Chapter 5- Electronics Design and Sensor Selections

5.1 Component Selection

5.2 Sensors along with specification and features from datasheet

5.3 Power requirements and power supply design



5.4 Motor Selection Current/Voltage/Speed/Torque

5.5 Feedback Mechanism positional sensors

5.6 Deliverable of complete electronics design with A4 size Schematic and PCB with discussion

Chapter 6- Software/Firmware Design

6.1 Input Output pin outs

6.2 Controller Selections with features

6.3 Software Design details & user Requirements

6.4 State Machine & System flow diagram

6.5 Flowchart for each circle of State machine

6.6 User cases for running your system (Test Cases)

6.7 Deliverable of complete commented code as per state machine and discussion

Chapter 7- Simulations and final Integrations (Test definition phase)

7.1 Integrations and testing all hardware and software component separately

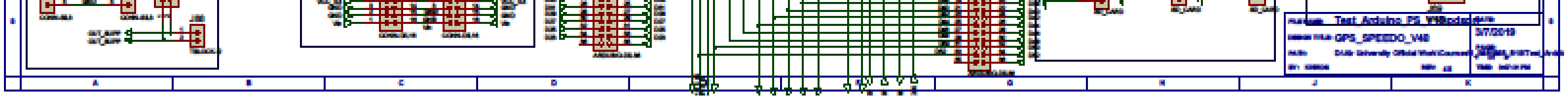
7.2 Simulations (Proteus with state machines)

7.3 Simulation PCB (3D diagram with connector and power interface)

7.4 Simulation Cad etc.

7.5 Actual Wiring plan with color codes (Wires external to system from sensor, power supply, actuator etc)

7.6 Discussion on simulations with possible challenges



Chapter 8- System Test phase

8.1 Final testing

8.2 Things that are questionable and get burned again and again

8.3 Project actual Pictures

Chapter 9- Project management

9.1 Everyone must write one page about how he executed his role in project

9.2 Comment individually about success /failure of your project

9.3 A paragraph about other team member positive and negative aspects highlighting how they could improve rather blaming them

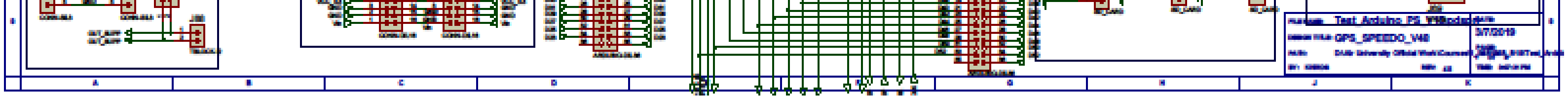
9.4 Final Bill of material list and paragraph about project budget allocation

9.5 Give a word count how many words each member write in final report in their allocated color as assigned

9.6 Risk management that you learned

Appendix

References-



CHAPTER 1- PROJECT PRELIMINARIES

1.1 PROPOSAL

Our proposed was to design a line following robot that can cover any given path in minimum possible time while being cost effective also. Microcontroller System for a line following robot that can go on binary path from Point A to Point B, upon reaching Point B which is a T junction with all black line and stop, which it uses IR sensors which detects the line and H bridge which controls the working of the wheels.

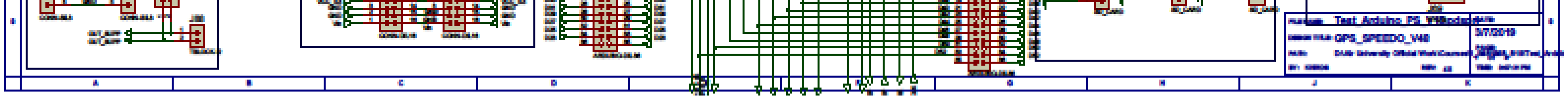
1.2 CONCEPT

Generally, the path is predefined and can be either visible like a black line on a white surface with a high contrasted color or it can be invisible like a magnetic field. Definitely, 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, a small line follower robot designed for the line follower robot's competition. Basic attributes of our project considered by us were:

1. Reliable construction.

2. Cost effective.

3. Collection of data of various types.



1.3 DESIGN

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

<u>**Reliable construction:**</u>

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

2. Implementing circuit on breadboard.

<u>**Cost Effective:**</u>

1. Purchasing the components from whole seller.

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

<u>**Collection of data of various types:**</u>

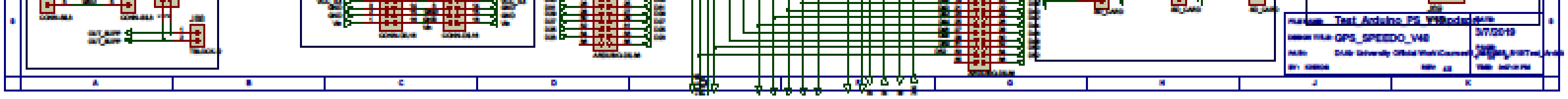
1. Current required by motors.

2. Required speed.

3. Voltage required.

4. Weight of the system with batteries.

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



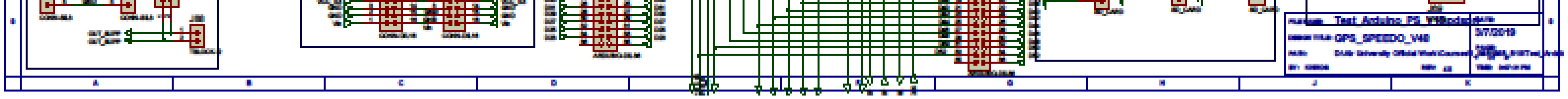
1.4 WORK BREAK DOWN STRUCTURE

PHASES	WORK DISTRIBUTION
PHASE I	Simulation on Proteus, Programming
PHASE II	PCB designing and Making
PHASE III	Ordering components
PHASE IV	Implementation on breadboard and then on PCB
PHASE V	Complete Project and final testing

Table 1.4 Work distribution

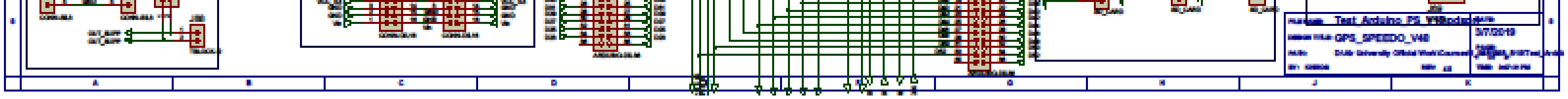
1.5 ESTIMATED BUDGETS

COMPONENT	PRICE
1 chassis (2 tires, 2 motors, bolts, nuts, and base)	1000
Wires (Male to female, male to male, female to female)	250
Soldering Wire	300
Photo Sheet	30
Ultrasonic Sensor	290
Headers (male & female)	150
PCB board	400
Voltage Sensor	150
Current Sensor	300



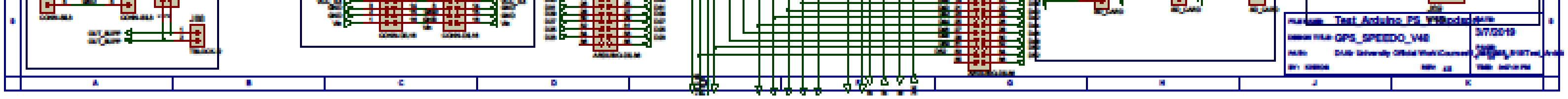
Double sided tape	120
Ferric chloride	250
3 IR sensors	360
L298N motor driver	350
ARDUINO MEGA	3000
Bread board	140
4 battery cells	1000
LCD Display	400
SD Card module	200
Servo motor	200
Ultrasonic sensor Holder	70
	Total = 9,500-

Table 1.5 Estimated Budgets



1.5 TEAM MEMBERS ROLES & DETAIL

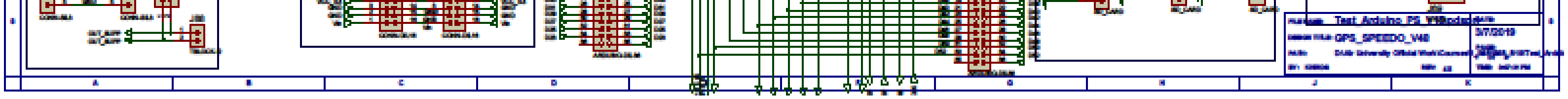
Team Members	Roles
MIRZA FAIZAN BAIG (RED)	**Buying components, writing code, Schematic simulation on Proteus, Report Making**
MUHAMMAD NOUMAN (GREEN)	**Writing code, Schematic Simulation, hardware testing, updating code and Report making.**



1.6Work Break down structure

Phase	Work distribution
Phase I	**Simulation on Proteus**
Phase II	**Buying Components**
Phase III	**PCB Making**
Phase IV	**Implementation on Breadboard**
Phase VI	**PCB Implementation**
Phase V	**Complete Project and final Testing**
Phase VII	**Report Making**

Table 1.6 Work breakdown



CHAPTER # 2-PROJECT CONCEPTION

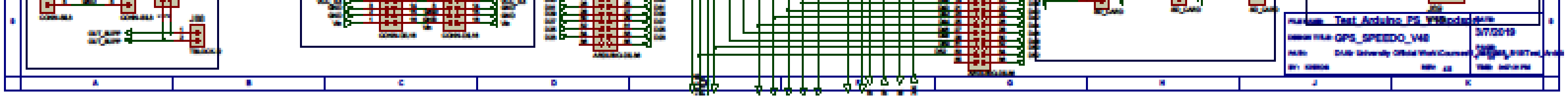
2.1 INTRODUCTION

Robots are fully automated machines that initiate action upon sensing something, make decisions based on conditions, and stop autonomously in response to other conditions or sensory inputs. They can be considered as replicas of human beings, functioning based on similar principles. The purpose of robots is to facilitate human tasks, and they can be either stationary or mobile. A line follower robot is an advanced type of mobile robot, designed with a movable base enabling it to move from one location to another. This robot is programmed to follow a specific path or trajectory, making decisions based on its encounters with obstacles. Typically, it tracks a black line on a white floor, although in some cases, it may navigate using a magnetic field that is not visible to the naked eye. Line follower robots find applications in various industries and domestic settings, especially for tasks like transporting parcels or materials.

2.2 LITERATURE REVIEW:

2.2.1. ULTRASONIC SENSOR:

Ultrasonic transducers and sensors are devices capable of generating or detecting ultrasound energy. An ultrasonic sensor operates by emitting a high pulse followed by a low pulse continuously. When these pulses encounter an obstacle, they bounce back and are detected by the ultrasonic sensor. The time taken for these signals to return is used to calculate the distance between the sensor and the obstacle. Shorter return times indicate closer obstacles, while longer times suggest greater distances between the sensor and the obstacle.



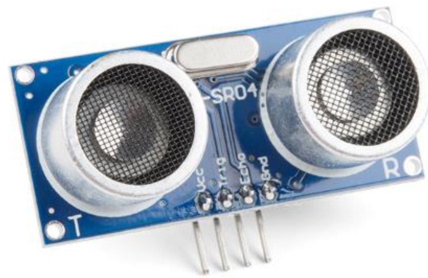


Figure 2.2 ULTRASONIC SENSOR

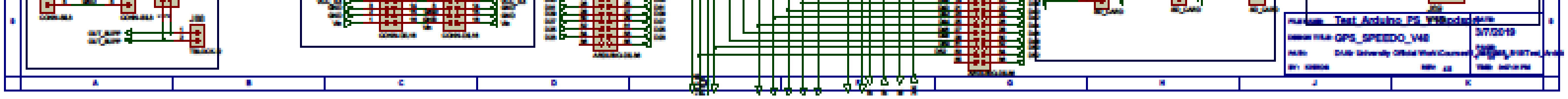
The sensor works with the simple high school formula that

$$Distance = Speed \times Time$$

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.

2.2.2 ARDUINO MEGA 2560:

The Arduino Mega 2560 is a microcontroller board centered around the ATmega2560 chip. It boasts 54 digital input/output pins, with 15 capable of functioning as PWM outputs, along with 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. This board includes all the necessary components to support the microcontroller; you can simply connect it to a computer using a USB cable or power it with an AC-to-DC adapter or battery to begin using it. Additionally, the Mega 2560 board is compatible with most shields designed for the Uno model, enhancing its versatility and expandability.



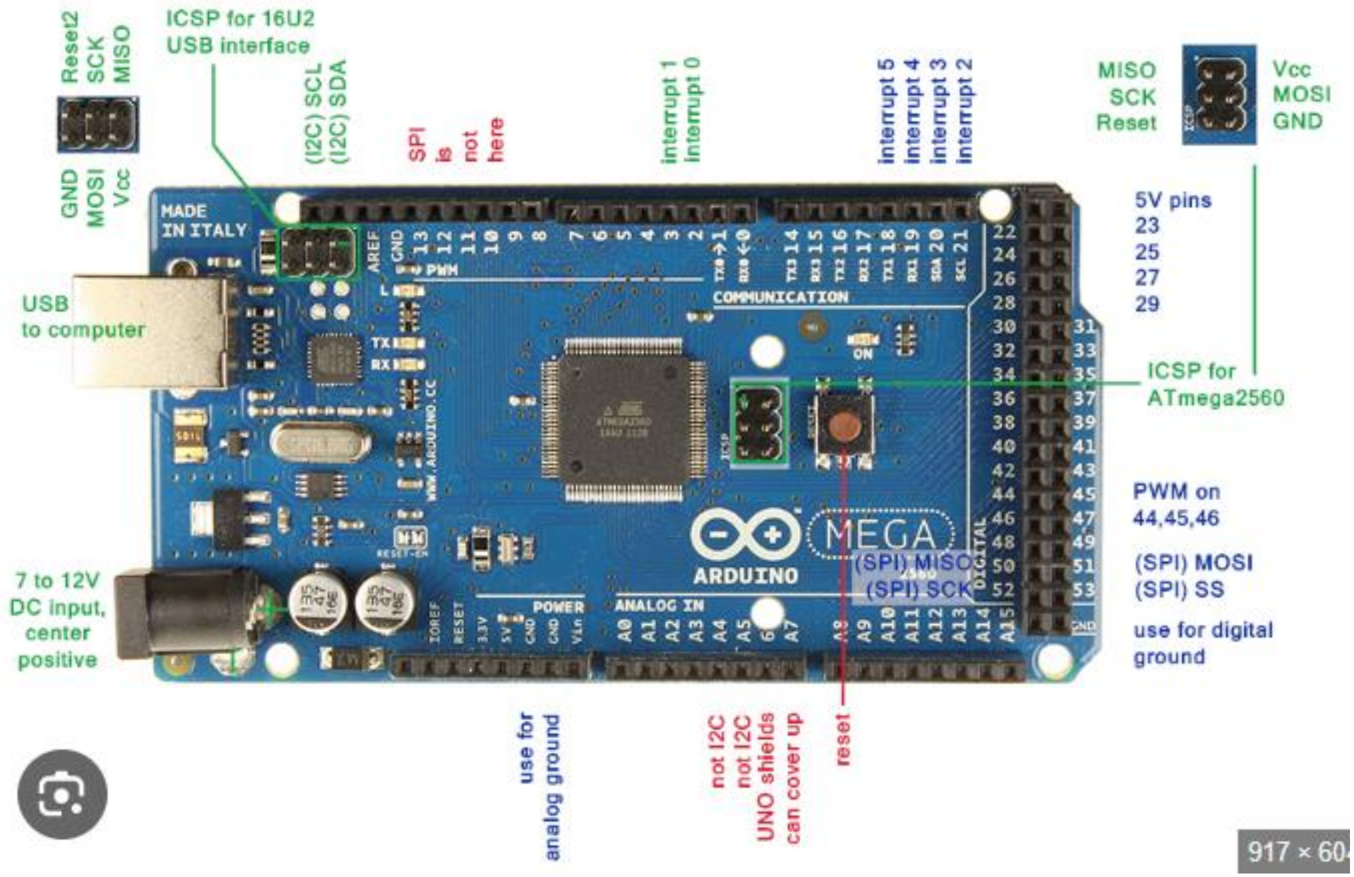
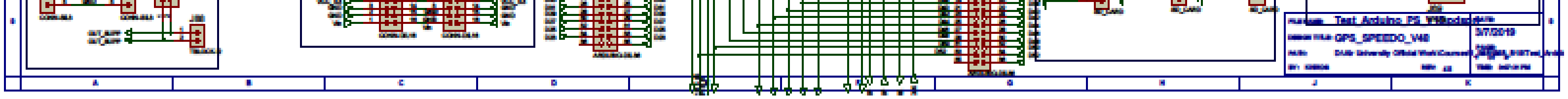
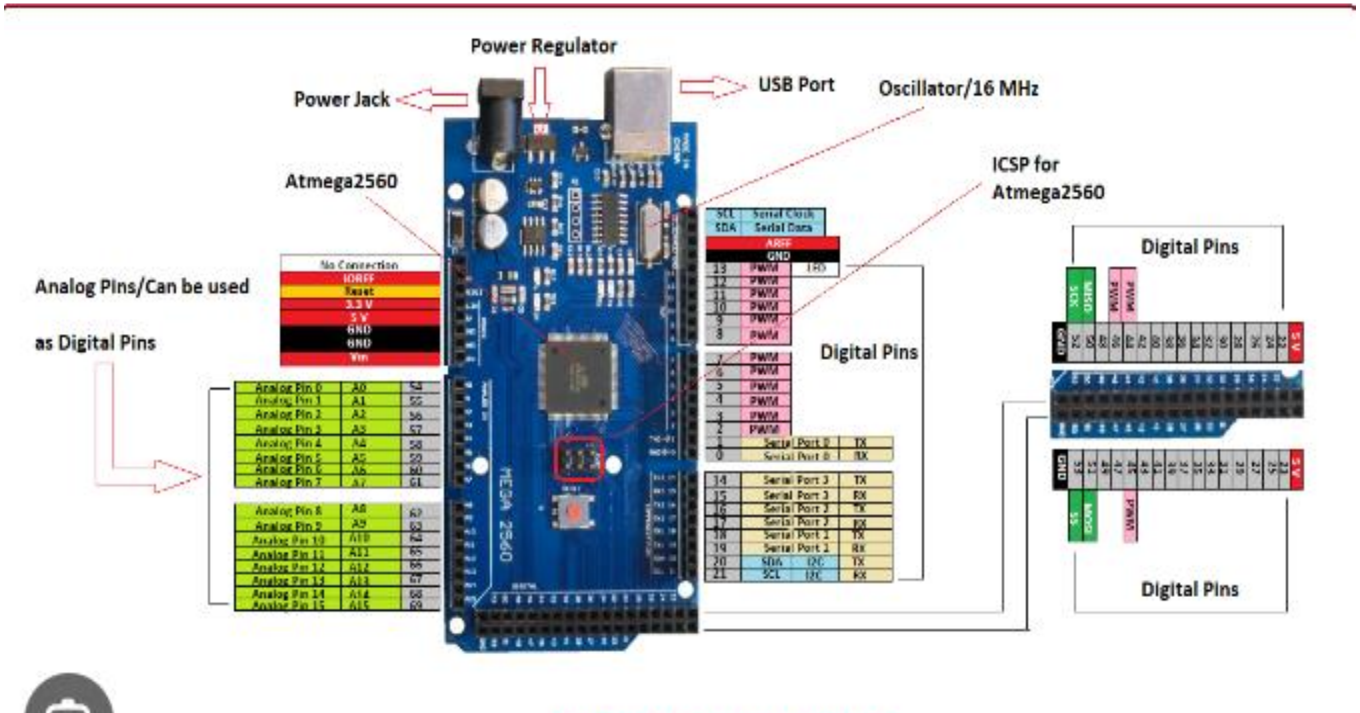


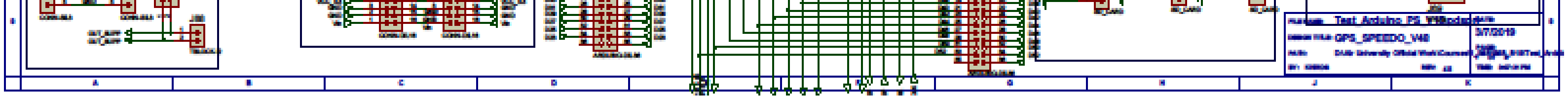
Figure 2.3 ARDUINO AtMega-2560



Pin Configuration:



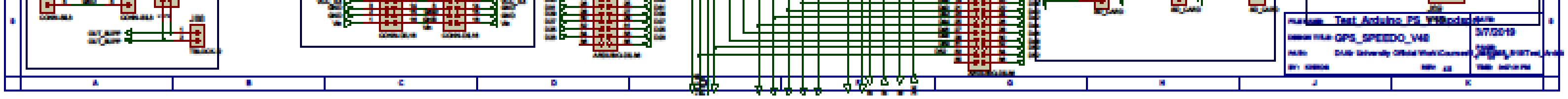
Arduino Mega 2560 Pinout



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

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

Figure 2.5 ROBOTIC CHASIS



2.2.1. L298N Motor Driver

This **L298N Motor Driver Module** is a high power motor driver module for driving DC and Stepper 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.

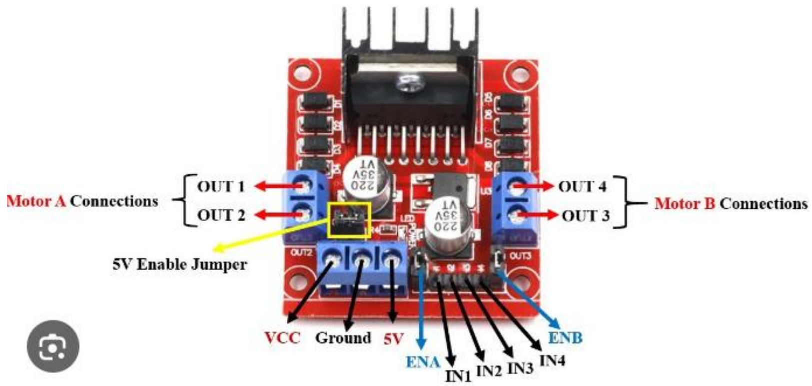


Figure 2.6 L298N

2.2.1. TCRT 5000 IR SENSOR

The TCRT5000 are reflective sensors which include an infrared emitter and phototransistor in a 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.

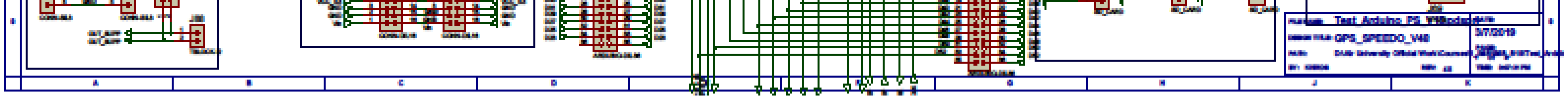


Figure 2.7 TCRT5000 IR Sensor

2.2.1. 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.

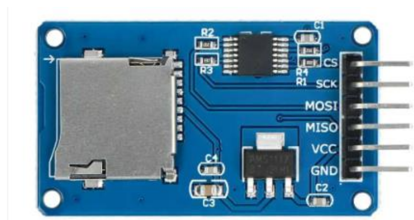
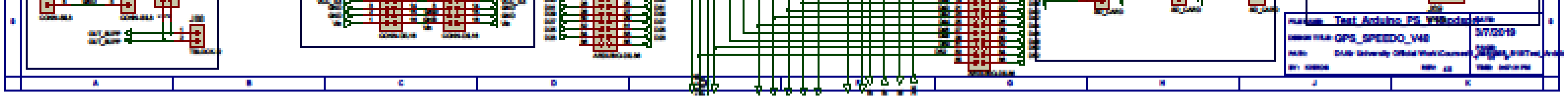


Figure 2.8 SD CARD MODULE



2.2.1. 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.

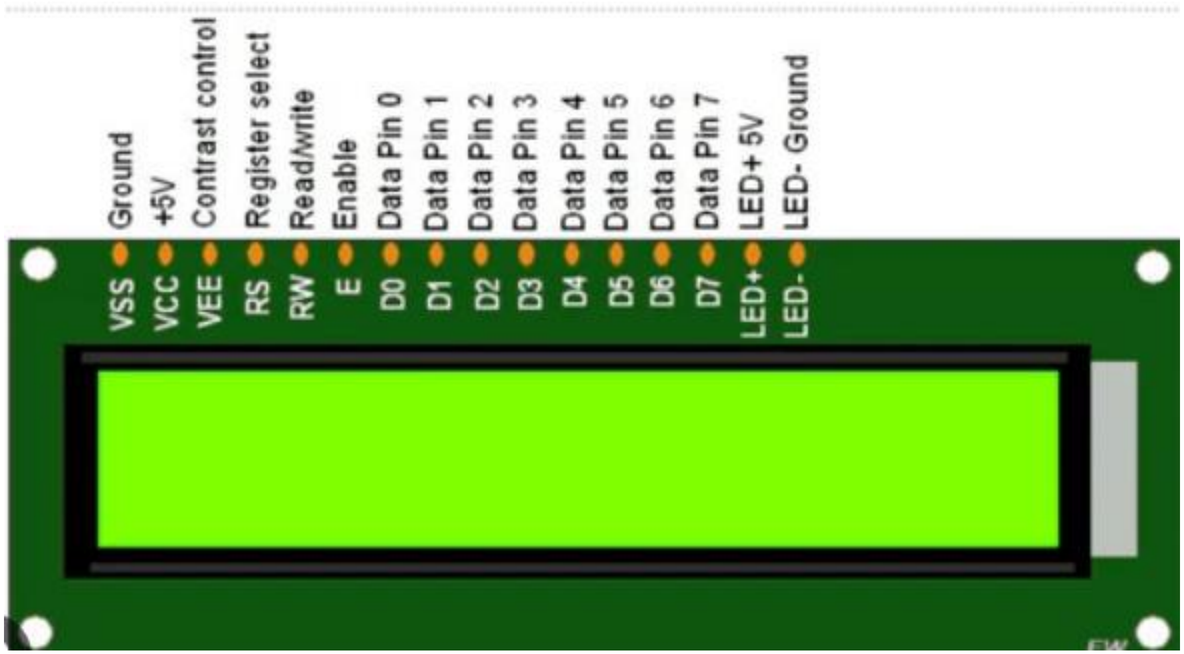
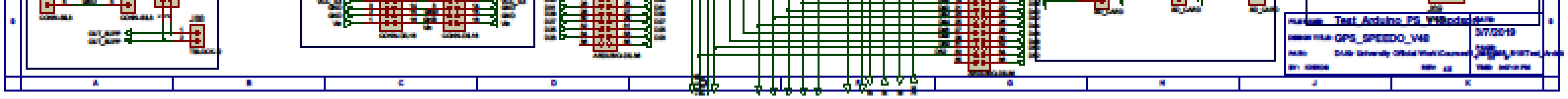


Figure 2.9 LCD DISPLAY



2.2.1. 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.

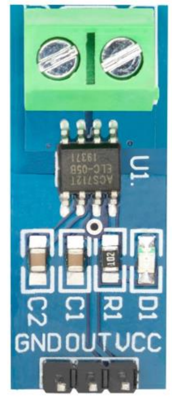
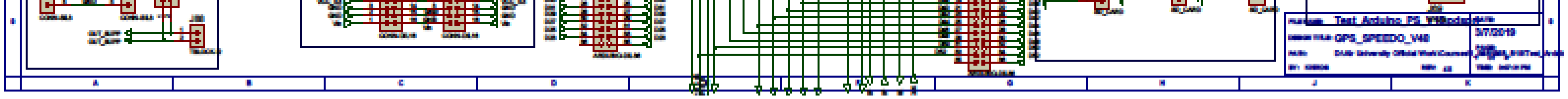


Figure 2.10 Current Sensor

2.2.1. 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 2.11 Voltage Sensor



2.2.1. Potentiometer

A potentiometer is a type of resistor that has three terminals, one of which is a sliding or rotating contact that creates an adjustable voltage divider. When only two terminals are utilized, namely one end and the wiper, the potentiometer functions as a variable resistor or rheostat. In terms of instrumentation, a potentiometer is essentially a voltage divider used to measure electric potential (voltage). The component itself embodies this principle, thus earning its name. Potentiometers are frequently employed to regulate electrical devices like volume controls on audio equipment. Mechanically operated potentiometers can also serve as position transducers, such as in a joystick. However, they are seldom used to directly control substantial power (over a watt) due to the power dissipation within the potentiometer being comparable to the power in the load being controlled.

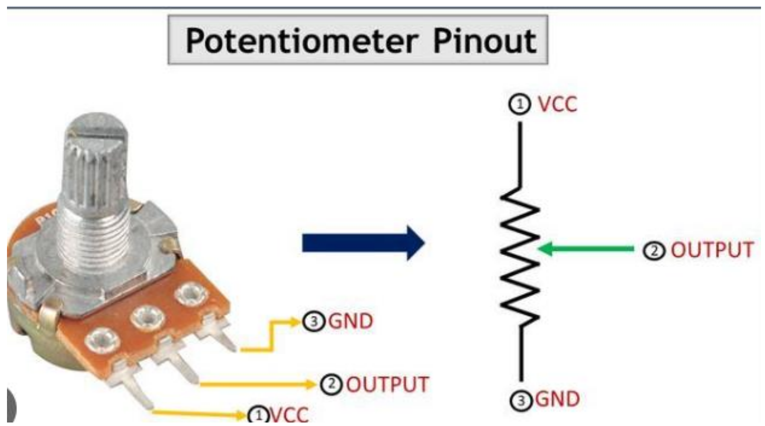
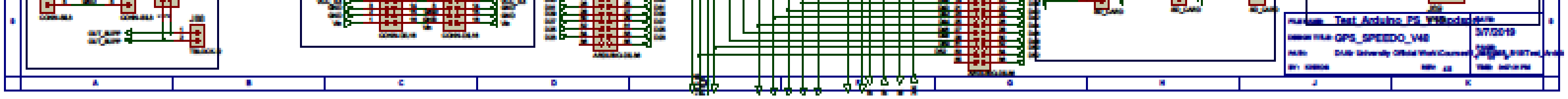
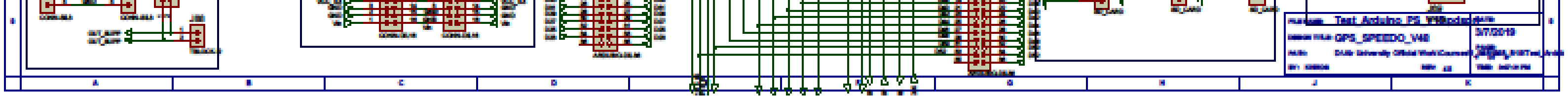


Figure 2.12 Potentiometer



2.3 FLOW CHART

Figure 2.17 Flow chart



2.4 DETAILED BLOCK DIAGRAM:

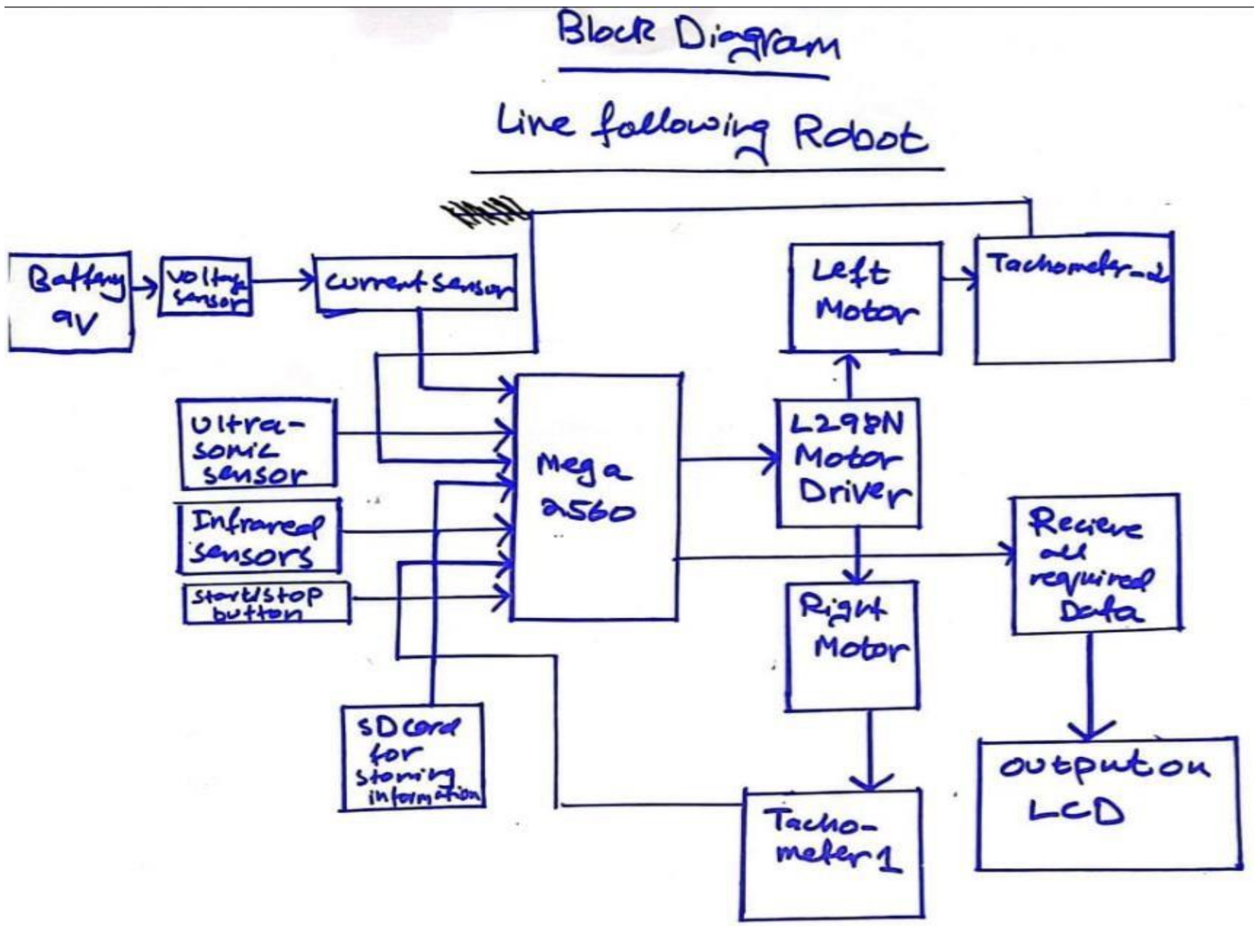
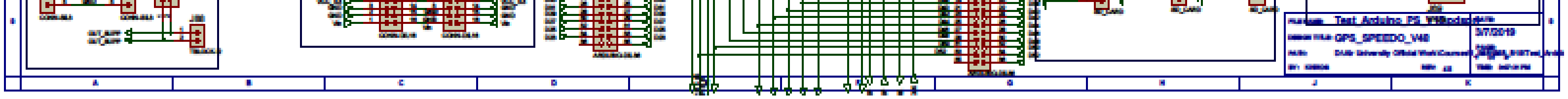


Figure 2.13 Detailed block diagram



CHAPTER # 3 PRODUCT DESIGN

3.1 SYSTEM CONSIDERATION FOR THE DESIGN

Our project utilized microcontrollers and logic-based sensors, with logic implementation based on truth tables. We aimed to simplify logic to reduce costs by minimizing the number of components. This involved achieving the lowest cost form, defined as the lowest number of gates with the fewest inputs per gate.

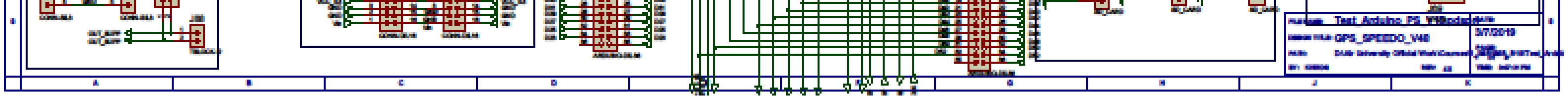
3.2 Criteria for Component Selection:

<u>**DC motors**</u>

DC motors were chosen due to their ease of control, requiring only two signals for operation. Their direction can be changed by reversing the polarity of the power supply, and speed can be varied by adjusting the voltage across the motor. Using two motors enables movement in any direction, known as a differential drive system. The power output of a motor can be increased by raising the voltage rating or current, with higher voltage options providing more power but also posing safety concerns due to potential shock hazards. Additionally, DC motors allow for speed control through voltage adjustments.

Sensors Selection

Given the requirement to build a line follower robot without a microcontroller, IR sensors are the most suitable choice. In our arena setup, these sensors will detect black lines as paths and white areas as obstacles, necessitating circuit configuration accordingly. While a minimum of two IR sensors could suffice, using five sensors, as advised, will enhance the robot's accuracy despite increasing circuit complexity.

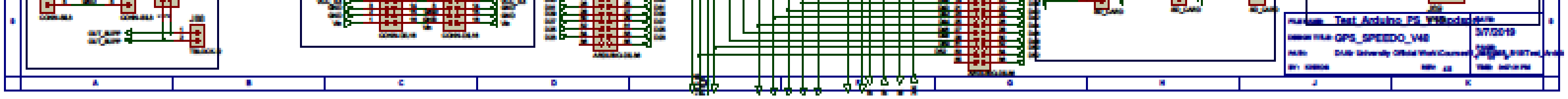


Working principle of IR sensor

The working principle of an IR sensor in a line follower robot is based on light behavior. When IR rays or light falls on a white surface, it reflects significantly, whereas on a black surface, it gets absorbed. This light behavior forms the basis for the robot's navigation.

16x4 LCD DISPLAY

1. Slim profile
2. Better under brighter conditions because of anti-glare technology
3. Lighter in weight with respect to screen size
4. Energy efficient because of lower power consumption

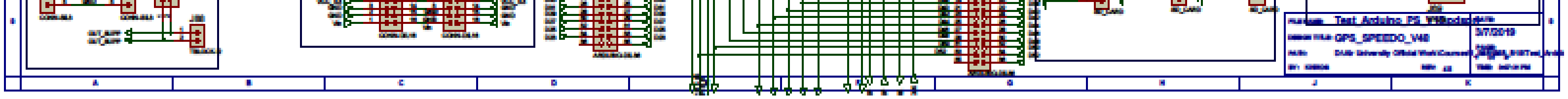


CHAPTER # 4 Mechanical Design

Figure 4.1 Base

Features:

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



CHAPTER 5- SOFTWARE/FIRMWARE DESIGN/THEORETICAL DESIGN

5.1 Explanation

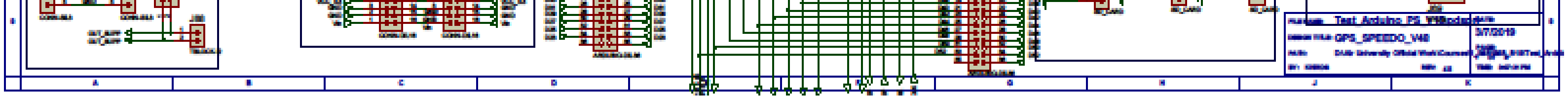
Following are the inputs and outputs of the circuit.

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. SD card

Outputs

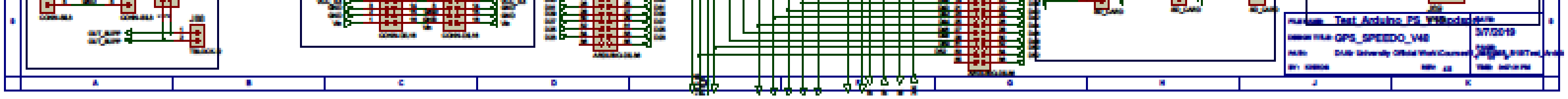
1. Left Motor
2. Right Motor



3. 16 X 4 LCD

5.2 SCHEMATIC COMPONENTS

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

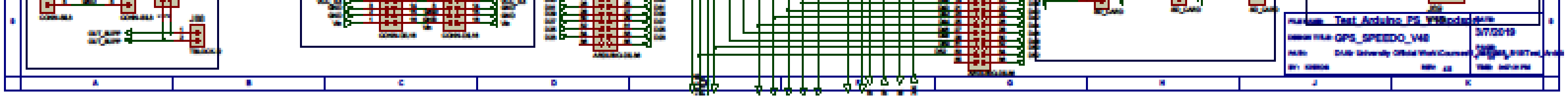


Chapter 6- Programming And Sensor Integration

6.1Integration of LCD

CODE

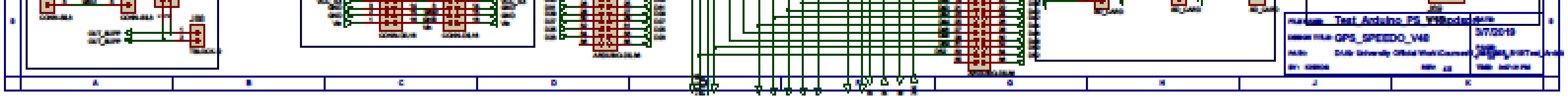
```
/* This is a sketch to test 16x4 LCD:
#include LiquidCrystal lcd(8,9,4,5,6,7);
void setup() {
lcd.begin(16,4);
lcd.setCursor(0,0);
lcd.print("VOLTAGE");
lcd.setCursor(0,1);
lcd.print("CURRENT!");
lcd.setCursor(0,2);
lcd.print("P/N: ");
lcd.setCursor(0,3);
lcd.print("SD CARD");
}
void loop() {
}
```



6.2Integration of SD Card Module

CODE

```
#include <SPI.h>
#include <SD.h>
File myFile;
void setup() {
// Open serial communications and wait for port to open:
Serial.begin(9600);
while (!Serial) {
; // wait for serial port to connect. Needed for native USB port only
}
Serial.print("Initializing SD card...");
if (!SD.begin(10)) {
Serial.println("initialization failed!");
while (1);
}
Serial.println("initialization done.");
// 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);
// if the file opened okay, write to it:
if (myFile) {
Serial.print("Writing to test.txt...");
myFile.println("This is a test file :)");
myFile.println("testing 1, 2, 3.");
for (int i = 0; i < 20; i++) {
myFile.println(i);
}
// close the file:
myFile.close();
Serial.println("done.");
} else {
```

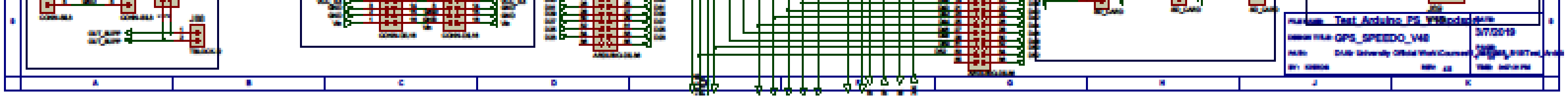


```
// if the file didn't open, print an error:
Serial.println("error opening test.txt");
}
}
void loop() {
// nothing happens after setup
}
```

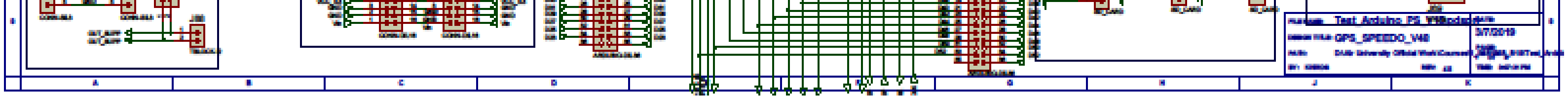
6.1Integration of Ultrasonic Sensor

CODE

```
/*

* Ultrasonic Sensor HC-SR04 interfacing with Arduino.

*/

// defining the pins

const int trigPin = 9;

const int echoPin = 10;

// defining variables

long duration;

int distance;

void setup() {

pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output
```



```
pinMode(echoPin, INPUT); // Sets the echoPin as an Input
Serial.begin(9600); // Starts the serial communication
}
void loop() {
// Clears the trigPin
digitalWrite(trigPin, LOW);
delayMicroseconds(2);
// Sets the trigPin on HIGH state for 10 micro seconds
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);
// Reads the echoPin, returns the sound wave travel time in microseconds
duration = pulseIn(echoPin, HIGH);
// Calculating the distance
distance= duration*0.034/2;
// Prints the distance on the Serial Monitor
Serial.print("Distance: ");
Serial.println(distance);
}
```

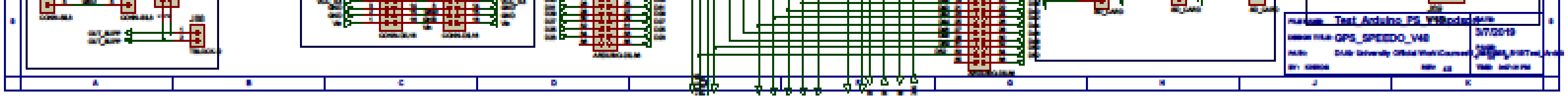


6.1 Integration of Optical Encoder

The optical encoder used, its purpose here is to measure the speed of the robot.

CODE

```
int encoder_pin = 2;
unsigned int rpm = 0;
volatile byte pulses = 0;
unsigned long timeold = 0;
unsigned int pulsesperturn = 20;
static volatile unsigned long debounce = 0;
void setup(){
   Serial.begin(9600);
   pinMode(encoder_pin, INPUT);
   attachInterrupt(0, counter, RISING);
   pulses = 0;
   rpm = 0;
   timeold = 0;
  Serial.print("Seconds ");
  Serial.print("RPM ");
}
 void loop(){
   if (millis() - timeold >= 1000){
      noInterrupts();
      rpm = (60 * 1000 / pulsesperturn )/ (millis() - timeold)*
pulses;
      timeold = millis();
      Serial.print(millis()/1000); Serial.print("       ");
      Serial.print(rpm,DEC); Serial.print("   ");
      pulses = 0;
      interrupts(); // Restart the interrupt processing
   }
  }
```



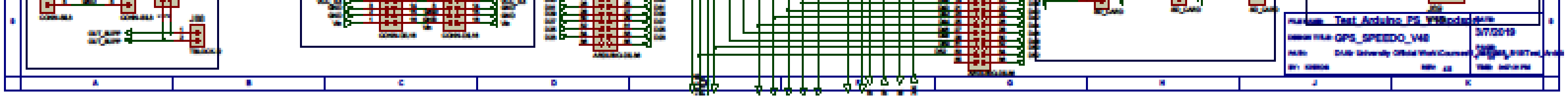
```
 void counter(){
  if(  digitalRead (encoder_pin) && (micros()-debounce > 500) &&
digitalRead (encoder_pin) ) {
                             //Check again that the encoder sends
a good signal and then check that the time is greater than 1000
microseconds and check again that the signal is correct.
        debounce = micros(); // Stores the time to check that we
do not count the bounce in the signal.
        pulses++;}
        else ;
```

6.2Integration of IR Sensor

The connections for the IR sensor with the Arduino are as follows: Connect the negative wire on the IR sensor to GND on the Arduino. Connect the middle of the IR sensor which is the VCC to 5V on the Arduino. Connect the signal pin on the IR sensor to pin 8 on the Arduino.

CODE

```
void setup() {
  // put your setup code here, to run once:
pinMode(4,INPUT);
pinMode(12,OUTPUT);//LED
}

void loop() {
  // put your main code here, to run repeatedly:
if(digitalRead(4)==LOW){
  digitalWrite(12,HIGH);
}
else{
```



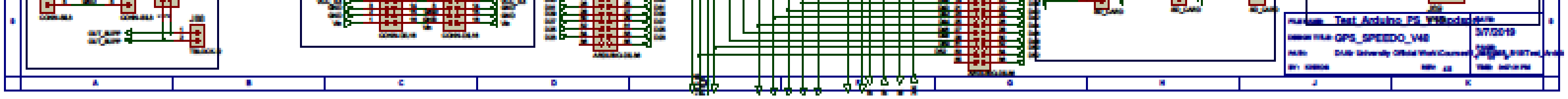
```
  digitalWrite(12,LOW);
}
}
```

6.3Integration of Current Sensor

First, the load. I have used a 12V DC Motor along with a 12V power supply. The screw terminals of the ASC712 Current Sensor Module board are connected in series with the motor and power supply. Then connect the VCC, GND and OUT of the ASC712 board to +5V, GND and A0 of Arduino. Now, in order to view the results, a 16×2 LCD is connected to Arduino. Its RS, E, D4-D7 pins are connected to Digital I/O Pins 7 through 2 of Arduino. A 10KΩ POT is connected to Pin 3 of LCD and its VCC and GND are connected to +5V and GND.

CODE

```
void setup() {
  Serial.begin(9600); //Start Serial Monitor to display current
read value on Serial monitor
}

void loop() {
unsigned int x=0;
float AcsValue=0.0,Samples=0.0,AvgAcs=0.0,AcsValueF=0.0;

  for (int x = 0; x < 150; x++){ //Get 150 samples
  AcsValue = analogRead(A0);     //Read current sensor values
  Samples = Samples + AcsValue;  //Add samples together
  delay (3); // let ADC settle before next sample 3ms
}
AvgAcs=Samples/150.0;//Taking Average of Samples

AcsValueF = (2.5 - (AvgAcs * (5.0 / 1024.0)) )/0.185;

Serial.print(AcsValueF);//Print the read current on Serial monitor
```



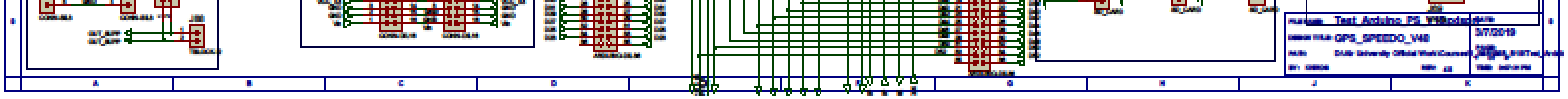
```
delay(50);
}
```

6.4Integration of Voltage Sensor

The Arduino Voltage Sensor Interface is pretty straight forward. Connect the voltage to be measured to the screw terminal of the Voltage Sensor, connected the output of the voltage divider to the Arduino. That’s it.

<u>CODE</u>

```
const float voltage_sensor= A2;
float voltage ;
float value;
float r2 = 39000.00 ;
float r1 = 10000.00 ;
void setup()
{
  // put your setup code here, to run once:
  Serial.begin(9600);
  pinMode(voltage_sensor, INPUT);
}
void loop() {
  // put your main code here, to run repeatedly
  value= analogRead(voltage_sensor);
  voltage= (value*(5.0/682)*(r1 + r2))/r2 ;
  Serial.print("Voltage Value :  ");
Serial.println(voltage);
}
```



CHAPTER 7 SIMULATIONS AND FINAL INTEGRATIONS

7.1SCHEMATIC CIRCUIT

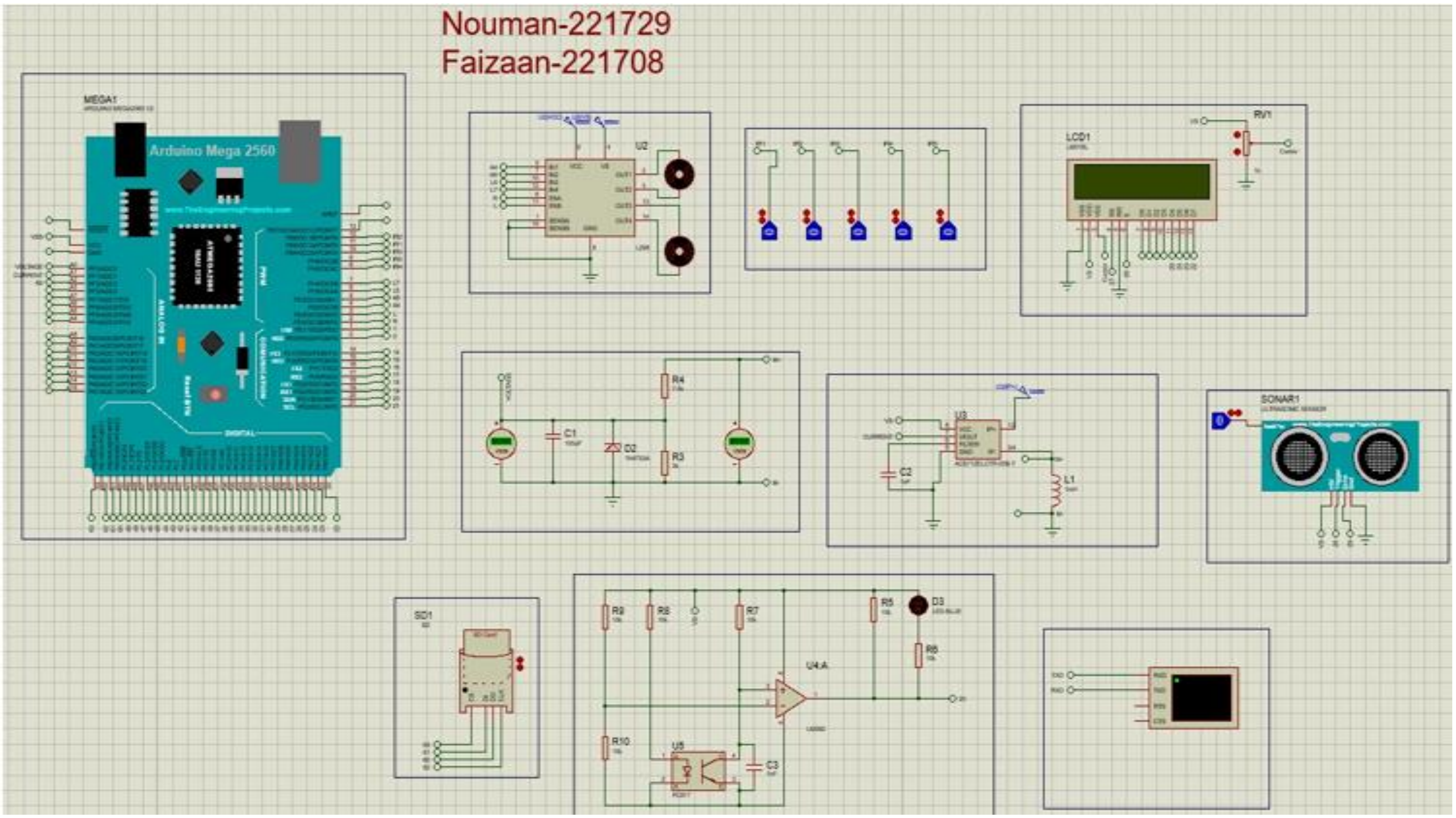
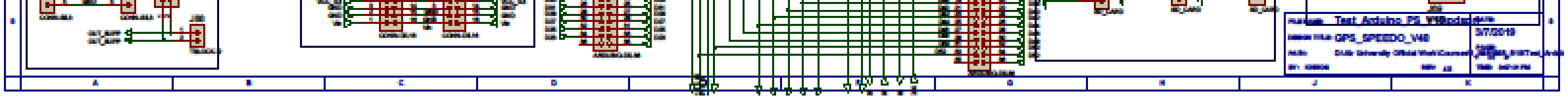


Figure 7.1 SCHEMATIC CIRCUIT



7.2PCB ON PROTEUS

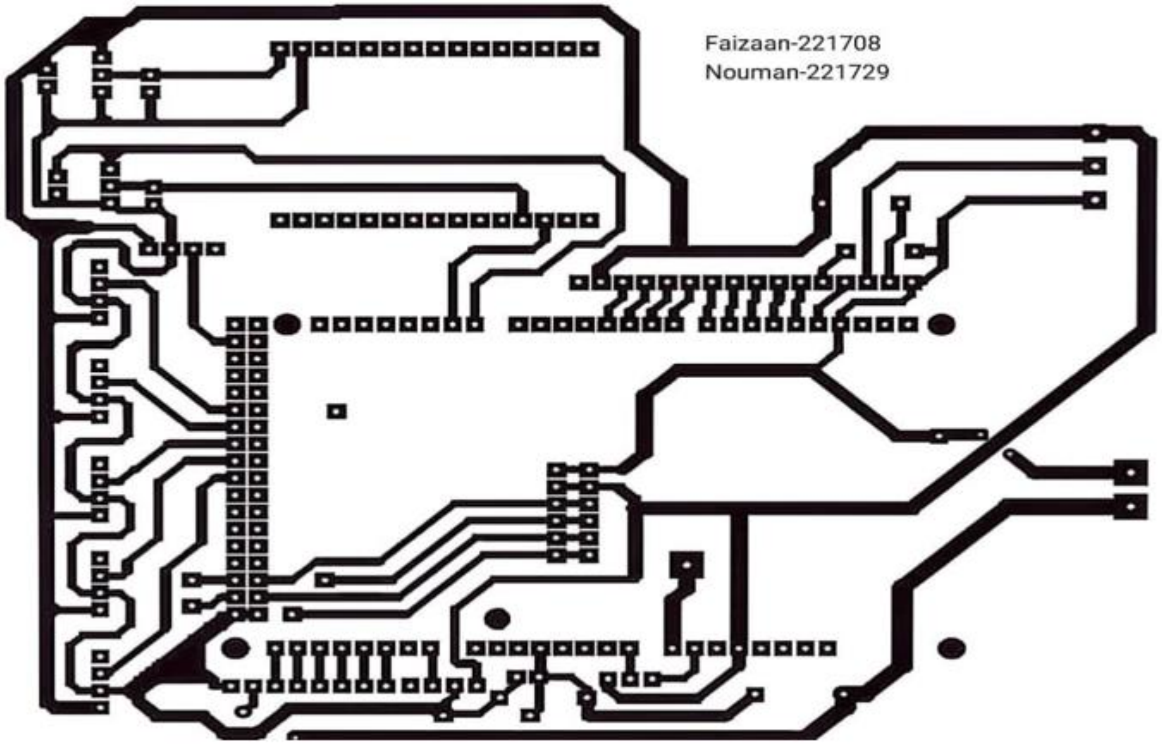
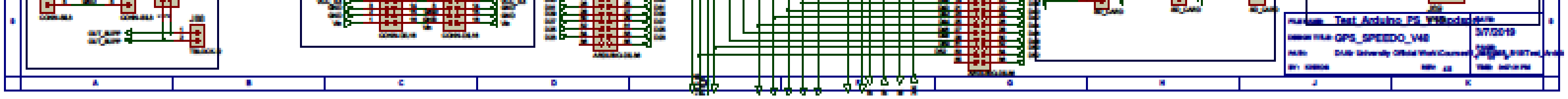
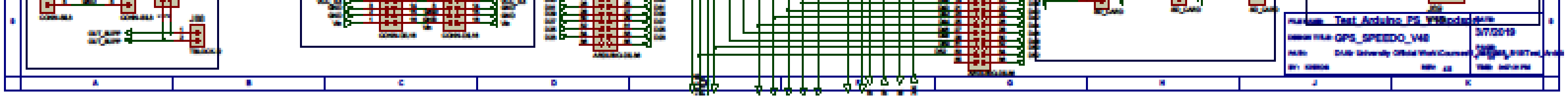


Figure 7.2 PCB LAYOUT



7.2 PCB TO BE IMPLEMENTED ON BOARD

Figure 7.3 PCB TO BE IMPLEMENTED



CHAPTER 8 HARDWARE IMPLEMENTATION

8.1 TESTING

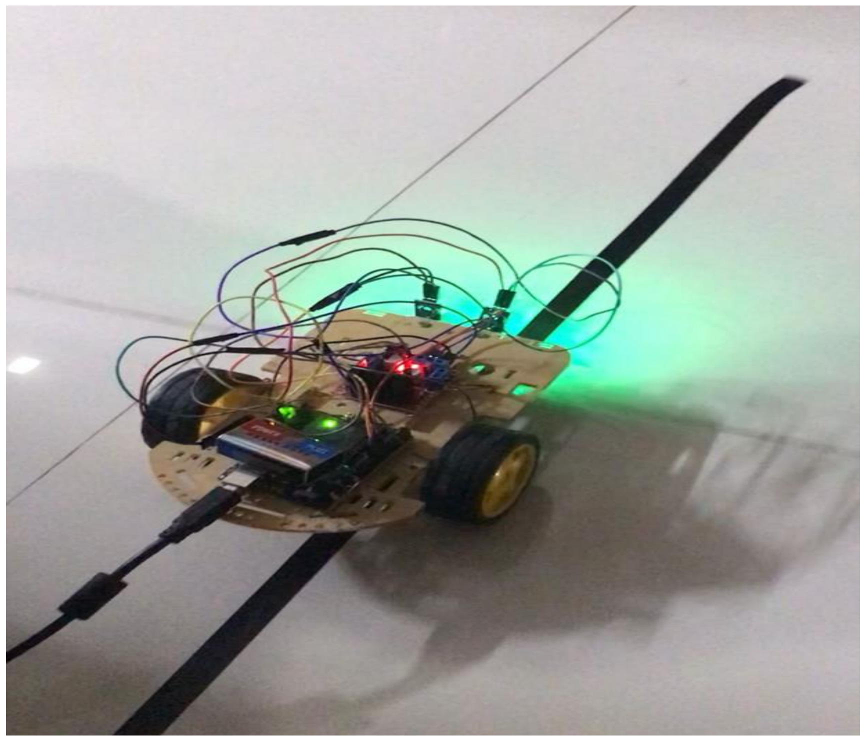
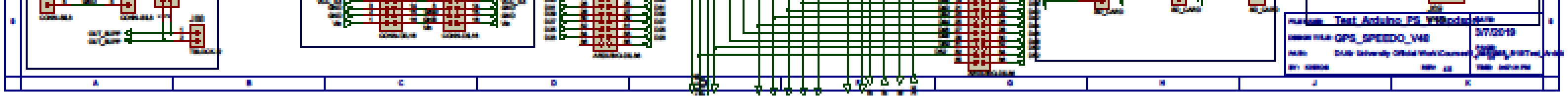


Figure 8.1 TESTING



8.2 PCB Board

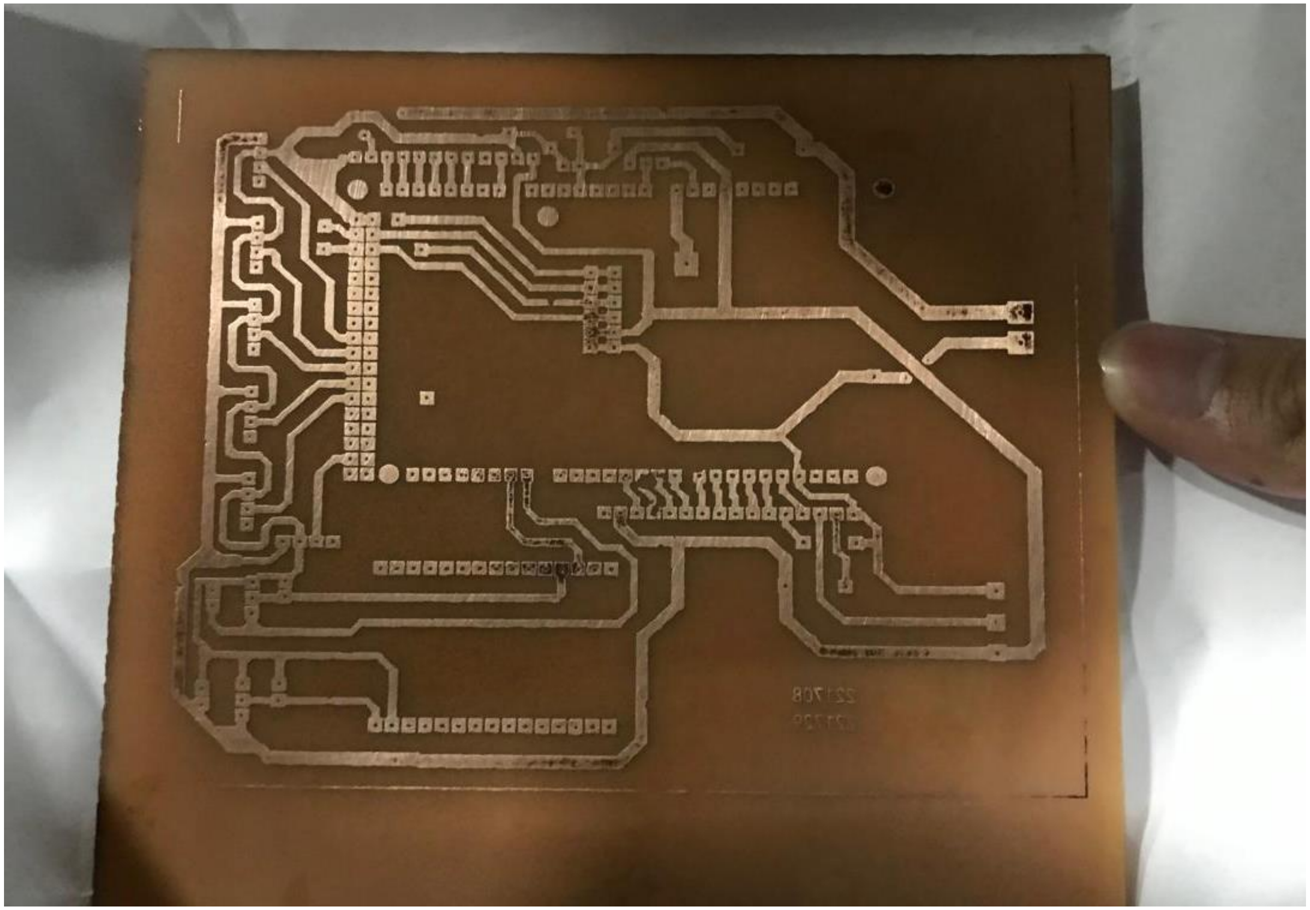
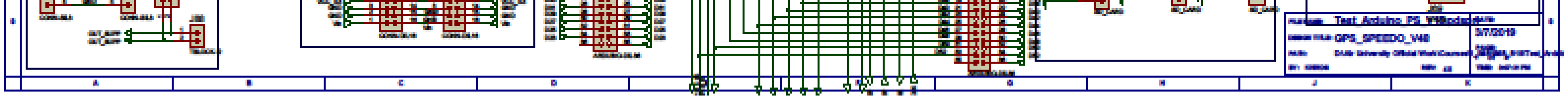


Figure 8.2 PCB Board



8.3 Placing Components on PCB Board

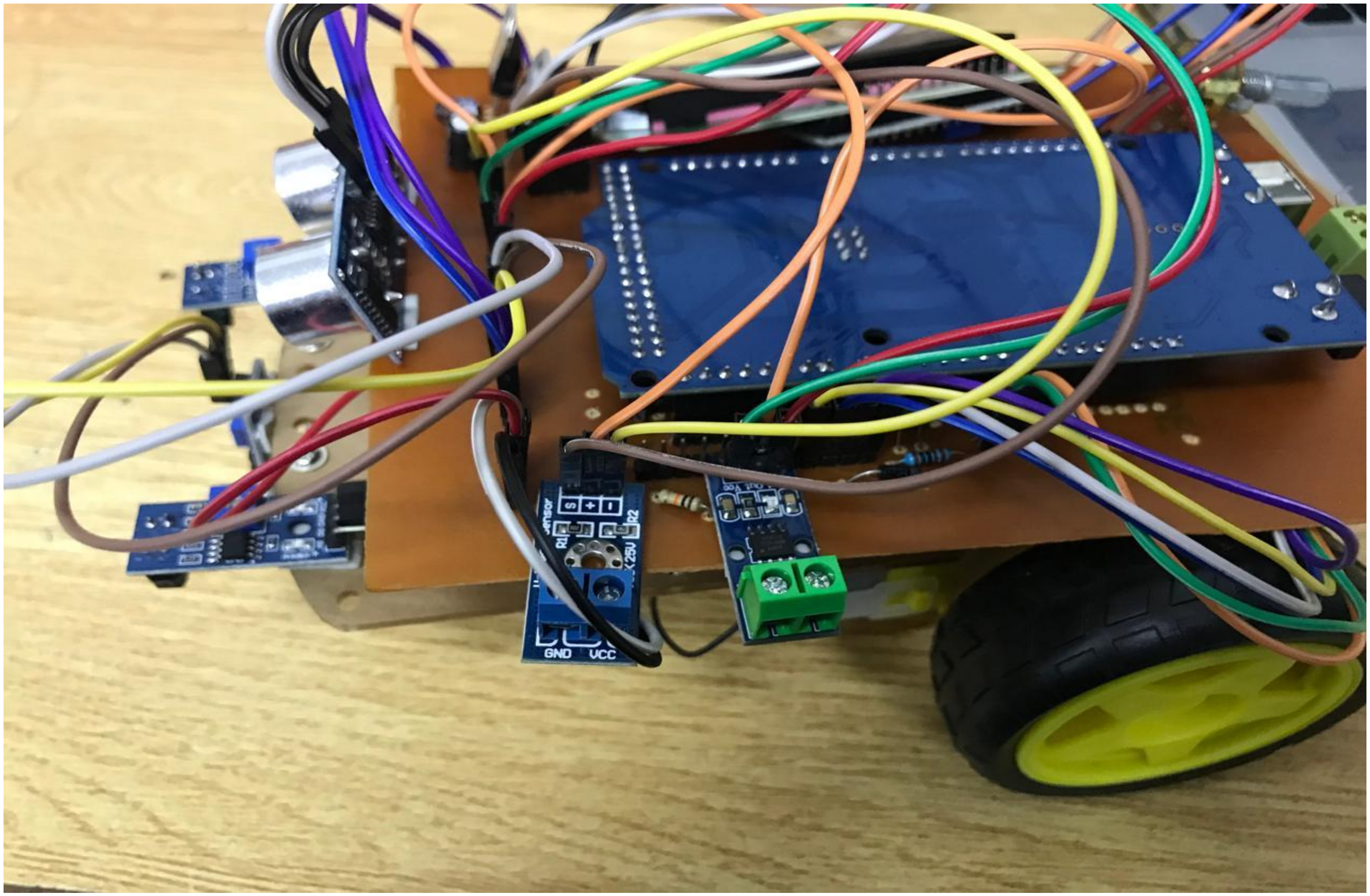
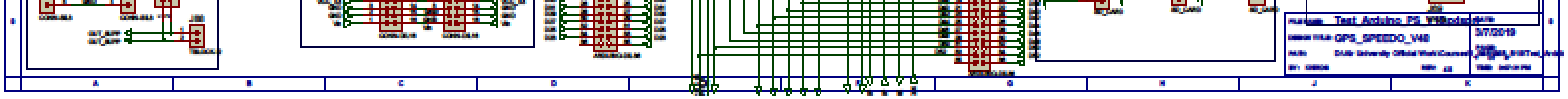


Figure 8.3 PCB Board After Placing Components



CHAPTER 9 SYSTEM TEST PHASE

9.1 FINAL TESTING

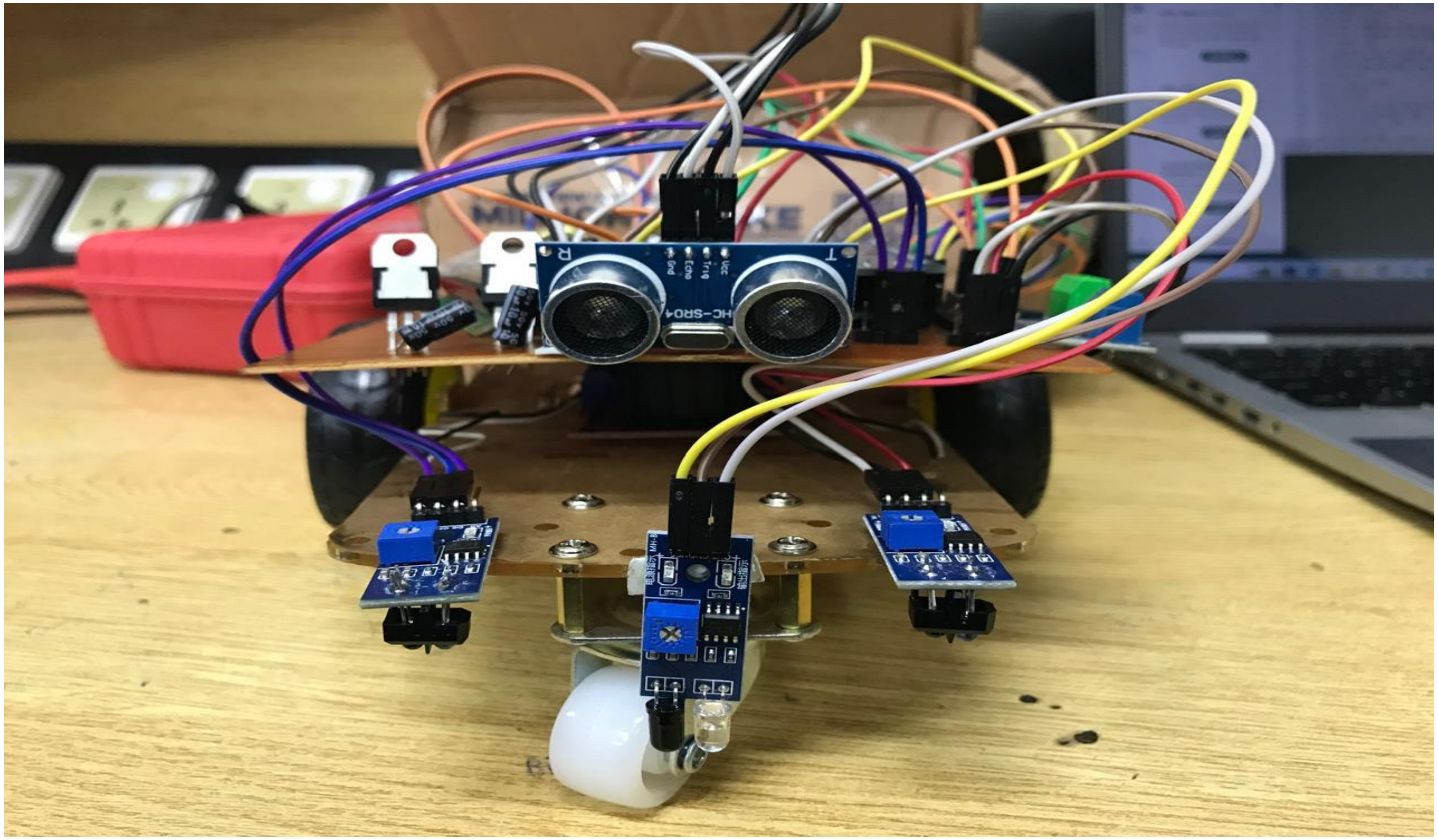
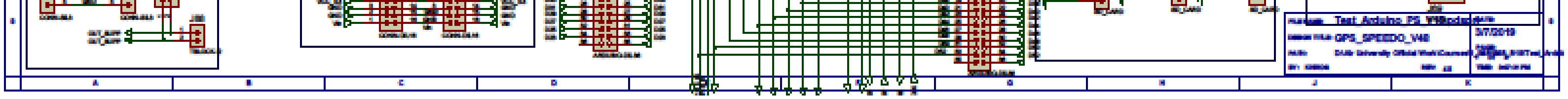


Figure 9.1 ARENA



References:

https://components101.com/sensors/ultrasonic-sensor-working-pinout-datasheet

https://components101.com/sites/default/files/component_datasheet/L298N-Motor-Driver-Datasheet.pdf

https://components101.com/sensors/ir-sensor-module

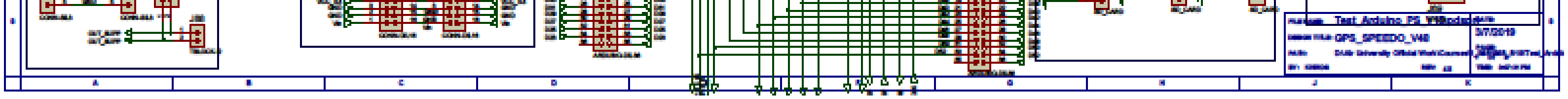
Arduino RPM Sensor (RPM Meter/Counter With Encoder) (deepbluembedded.com)

LCD 16X2 : Pin Configuration, Commands, Interfacing & Its Applications (watelectronics.com)

(PDF) Line Following Robot Using Arduino (researchgate.net)

Line Following Robot | Arduino Project Hub

THE END



GPS_SPEEDO_V40
3/7/2019
A
B
C
D
G
H
J
K