LINE FOLLOWER ROBOT

Group Members:

M. Maaz ***(221696)***

Saad Bin Owais ***(221654)***

Umair Mubashir ***(221651)***

BE-MECHATRONICS ENGINEERING (F22-B)

Project Supervisor
Sir Umer Farooq
Lecturer

Department of Mechatronics Engineering

Air University, Islamabad

Role	Name	Word count that each has written in assigned color code in report	Signature
Team Lead	Saad Bin Owais	555	SAAD
Project Manager	M. Maaz	1310	MAAZ
Technical	Umair Mubashir	709	UMAIR
Integrator and Tester	All Three		

- ## Chapter 1- Preliminaries 1.1 Proposal

- Our proposal centers on crafting an innovative embedded system robot, engineered to excel across diverse tasks such as line following, obstacle avoidance, and color recognition. The robot integrates key components including a QTR-8A Reflectance Sensor Array for precise line detection, an SR-04 Ultrasonic Sensor Module for obstacle avoidance, and a TCS3200 Color Sensor Module for accurate color recognition. The L928 Motor Driver ensures meticulous control over the robot's movements, while an ESP32 facilitates wireless communication and data transfer. Our project encompasses hardware integration, software development, rigorous testing, and comprehensive documentation. Upon completion, this robot will stand as a versatile platform, showcasing the prowess of embedded systems in real-world scenarios.

- Our plan spans eight weeks, delineated into pivotal milestones. We commence by integrating hardware and conducting initial tests in the initial two weeks. Subsequently, we pivot towards software development and algorithm implementation over the ensuing fortnight. Weeks five and six are allocated for integrating all systems and conducting thorough testing. Finally, the last two weeks are earmarked for documentation and report preparation. Our budget estimate encompasses expenses for hardware components, sensors, microcontrollers, and other requisite materials. This project aims to underscore the potential of embedded systems in robotics, culminating in the delivery of a fully functional robot along with comprehensive documentation to facilitate further exploration and understanding.

- ## 1.2 Initial Feasibility

- The preliminary feasibility assessment indicates promising outcomes for the Line Follower Robot (LFR) implementation. The utilization of the QTR-8A sensor array, alongside the L928 Motor Driver and Arduino Uno, successfully enables the robot to adhere to a predetermined line path. The feasibility is substantiated by the robot's consistent and accurate responses to alterations in the line layout. This successful implementation lays a robust foundation for integrating additional functionalities such as obstacle avoidance and color sensing in subsequent project phases. The hardware and software components exhibit compatibility, while the iterative testing approach permits ongoing refinement and enhancement of the line-following algorithm. Overall, the initial feasibility assessment augurs well for the project's successful progression.

- ## 1.3 Work Breakdown Structure
-
- 1. Line Follower Robot (LFR): Developing core functionality using IR sensor, L928 Motor Driver, and Arduino Uno.
- 2. PCB Design & Fabrication: Designing and fabricating a PCB for compact integration of all components, minimizing wiring.
- 3. Obstacle Avoidance & Color Sensing: Integrating SR-04 Ultrasonic and TCS3200 Color Sensors for advanced functionalities.
- 4. Encoder Modules & Power Management: Implementing encoder modules, managing power through a buck converter, and exploring wireless power solutions.
- 5. Wireless Communication Setup: Configuring ESP32 for seamless wireless communication and optimizing encoder data transmission.
- 6. Final Testing & Demonstration: Conducting comprehensive testing, optimizing parameters, and showcasing.
- 7. Using an SD card to store the path and other necessary information.
-
- # Chapter 2 Project Conception
-
- ## 2.1 Introduction
-
- In an era marked by rapid technological advancement, the amalgamation of embedded systems and robotics has unlocked possibilities for crafting intelligent and adaptable machines. Our current endeavor embodies this ethos, aiming to develop a versatile robot capable of navigating various environments with finesse. This creation transcends mere engineering; it symbolizes the fusion of state-of-the-art technologies, integrating precision line following, flexible obstacle avoidance, and sophisticated color sensing into one cohesive robotic system.
- As we delve into the depths of our robot's development, our endeavor aims to showcase the seamless integration of state-of-the-art components. These include the QTR-8A Reflectance Sensor Array, SR-04 Ultrasonic Sensor Module, TCS3200 Color Sensor Module, and the ESP32 microcontroller. This journey encapsulates the intricate interplay between hardware and software, where engineering principles and creative problem-solving converge to craft a compact, wire-free design achieved through meticulous PCB fabrication. Our creation, more than just a technological marvel, stands as a testament to the potential of embedded systems in robotics. It heralds a future where intelligent machines effortlessly interact with and comprehend their surroundings.

- ## 2.2 List of Features and Operational Specification of Your Project

Features:

The project's simulation was designed on Proteus. The project comprises the following features:

1. **IR Sensors:** High-resolution sensors for precise line detection in varying environmental conditions.
2. **SR-04 Ultrasonic Sensor Module:** Ultrasonic distance measurement with rapid response for effective obstacle detection and avoidance.
3. **TCS3200 Color Sensor Module:** Versatile color detection with programmable light-to-frequency converters for accurate color sensing.
4. **L928 Motor Driver:** Dual H-bridge motor driver enabling precise control of the robot's movements and turns.
5. **Arduino Uno:** Microcontroller board providing a flexible and user-friendly platform for program execution.
6. **Encoder Modules:** Providing motor rotation data for feedback, aiding in precise control and navigation.
7. **PCB Fabrication:** Custom-printed circuit board design, optimizing space and minimizing wire connections for a compact system.
8. **Robot Chassis:** Sturdy and customizable frame providing a foundation for component integration and mobility.
9. **Battery for Motors:** Power source tailored for motor operation, ensuring reliable and efficient energy supply.

Operation:

The basic operations of the line follower are as follows:

1. **Line Following Algorithm:** An algorithm implemented on the Arduino Uno processes the sensor data, making real-time decisions based on the position of the robot relative to the detected line.
2. **Motor Control Commands:** The Arduino Uno sends commands to the L928 Motor Driver, dictating the speed and direction of the robot's motors in response to the line-following algorithm.
3. **Directional Adjustments:** Depending on the sensor readings, the robot adjusts the speed of individual motors, enabling precise turns or corrections to stay aligned with the line.
4. **Proportional Control:** Proportional control mechanisms ensure that the robot responds smoothly to variations in the line, preventing abrupt movements and promoting stable line following.
5. **Real-Time Adaptation:** The line-following process occurs in real-time, allowing the robot to adapt swiftly to changes in the track layout, including sharp turns or intersections.

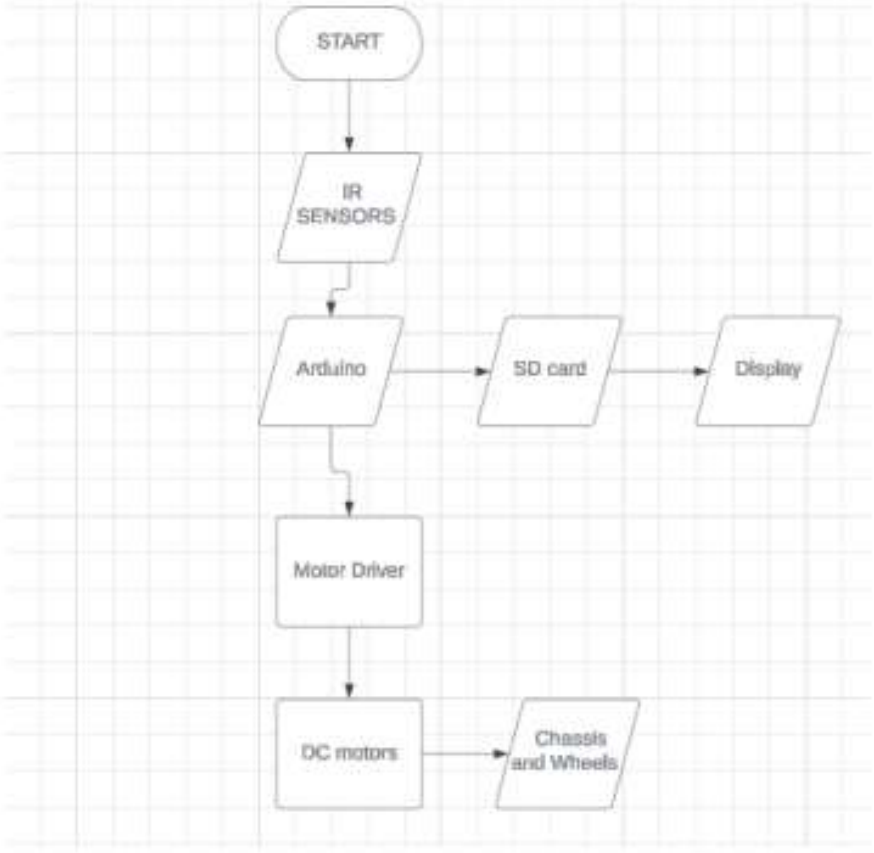
- **2.3 Components:**
-
- - 1x Arduino Uno
- - 1x Robot Chassis
- - 2x Motors 3V to 6V
- - 12C Module for LCD interfacing
- - Jumper Wires
- - 16x2 LCD Module
- - Header Pins
- - 1x SD Card Module
- - Resistors Capacitors
- - IR Sensors
- - On/Off Button
- - 1x Ultrasonic Sensor
- - 1x Encoder
- - 1x Motor Driver LM298
- - 1x Voltage Sensor
- - 3x Rechargeable Battery 3.7V
- - 1x Current Sensor

2.4 Project Development Process

-
- **1. Conceptualization and Planning**: Define the project scope, objectives, and key functionalities, outlining a clear roadmap for development.
- **2. Design and Prototyping:** Develop detailed designs, including hardware schematics and software architecture. Create prototypes or mock-ups to validate concepts.
- **3. Component Acquisition and Integration**: Acquire necessary components and integrate them according to the design specifications, ensuring compatibility and functionality.
- **4. Software Development:** Write and test the software code, incorporating algorithms for various functionalities. Debug and optimize code for efficient performance.

- **5. Testing and Validation:** Conduct comprehensive testing of individual components and the integrated system, addressing any issues and ensuring that the project meets predefined objectives.
- **6. Optimization and Refinement:** Refine both hardware and software components based results, addressing any inefficiencies, or improving performance. Iterate as needed to enhance overall project functionality.

2.1 Basic block diagrams of whole system and subcomponents



Chapter 3 Product Design

Our robot design is meticulously engineered, seamlessly integrating state-of-the-art technologies into a sleek and efficient form. Named Rahbar, this robot embodies versatility, effortlessly mastering a range of tasks. It boasts precise line following capabilities courtesy of the QTR-8A Reflectance Sensor Array, swift obstacle avoidance using the SR-04 Ultrasonic Sensor Module, and accurate color recognition via the TCS3200 Color Sensor Module.

To ensure precise control over its movements, we employ the L928 Motor Driver, while the ESP32 facilitates seamless wireless communication, enhancing Rahbar's dynamism and responsiveness. The custom-printed circuit board (PCB) simplifies wiring and reduces complexity, resulting in a streamlined and efficient design. Our aim is to create an intelligent and wire-free robot capable of adapting effortlessly to various situations..

3.1 System Consideration for the Design

1. Ensuring seamless integration among all components of our robot is paramount. We meticulously align sensors, motor drivers, and other elements to ensure they operate in harmony. Strategically placing them on our custom circuit board enhances communication and overall performance, ensuring smooth operation.

2. Sustaining our robot's longevity relies heavily on effective power management. Employing a buck converter ensures a stable power supply to the ESP32 and Arduino Uno, enabling them to function reliably. Selecting appropriate batteries for the motors optimizes efficiency, contributing to prolonged operation.

3. Establishing robust wireless communication is crucial for our robot's functionality. We're dedicated to maintaining a strong and dependable connection between the robot's components, managed by the ESP32. Rigorous testing and refinement processes guarantee our robot's ability to navigate any situation seamlessly.

4. Prioritizing aesthetics and functionality, we're crafting our robot to be sleek and efficient. Emphasizing wire-free design enhances both its appearance and maneuverability, allowing it to operate with agility and precision.

5. Precision in line following, obstacle avoidance, and color recognition hinges on finely tuned algorithms. These algorithms serve as the cognitive engine of our robot, enabling it to interpret its environment and make informed decisions. Continual refinement ensures our robot's adaptability and reliability in diverse scenarios.

6. Our robot is designed for versatility and scalability. By engineering it with flexibility in mind, we're poised to seamlessly integrate additional sensors or features in the future. This future-proofing approach ensures our robot remains at the forefront of technology and remains relevant for years to come.

Chapter 4 Mechanical Design

4.1 Mechanism selection

During the mechanism selection process for our project, we extensively consulted online resources. These documents offered valuable insights and practical examples, particularly regarding mechanisms such as rack & pinion and pulleys. By meticulously analyzing these calculations, our objective was to pinpoint the most appropriate mechanism that closely aligns with the precise requirements of our line-following robot.

4.2 Platform Design

The design of the platform for our project revolves around establishing a sturdy base that seamlessly accommodates a range of functionalities. Prioritizing versatility and scalability, this design integrates diverse hardware components, including sensors and motor controllers, onto a custom-printed circuit board (PCB) to ensure compactness and eliminate wiring complexities. The architecture of the platform is meticulously engineered to facilitate real-time communication among components, enabling efficient data exchange. Special attention is given to algorithmic precision for tasks such as line following, obstacle avoidance, and color sensing, ensuring the platform's adaptability in dynamic environments. Additionally, the modular design allows for future enhancements, positioning the robot as a flexible and ever-evolving platform for advancements in embedded systems and robotics.

Chapter 5- Electronics Design and Sensor Selections

5.1 Component Selection and Sensors along with specification and features from datasheet

1. **Ultrasonic Sensor**: We delved into ultrasonic transducers and sensors, devices capable of both emitting and sensing ultrasound energy. The sensor emits a series of high and low pulses in a continuous manner. Upon encountering an obstacle, these signals reflect back and are captured by the sensor. The duration it takes for the signals to return is leveraged to determine the distance to the obstacle. Shorter return times signify the obstacle's closer proximity to the sensor.

Pin Configuration:

Vcc The Vcc pin provides power to the sensor, typically connected to a +5V source.

Trigger the Trigger pin serves as an input, requiring a high state for 10 microseconds to initiate the measurement process by transmitting an ultrasonic wave.

Echo the Echo pin acts as an output, going high for a duration equivalent to the time taken for the ultrasonic wave to return to the sensor.

Ground the Ground pin connects to the system's ground, establishing a common reference point for the sensor and the overall electronic setup.

2. Arduino uno:

The Arduino Uno, centered on the ATmega328P microcontroller, serves as a versatile and user-friendly development board. With 14 digital pins, 6 analog pins, and various power pins, it streamlines the integration and management of external components. Digital pins function with binary values, while PWM pins facilitate analog signal generation. Analog pins measure voltage levels, and power pins deliver regulated voltage and ground connections. Dedicated RX and TX pins handle serial communication, while the RESET pin triggers a system reset when needed. The microcontroller's operations involve precise bitwise manipulations at the binary level, providing users with granular control over individual bits within registers. This amalgamation of features renders the Arduino Uno an influential platform suitable for a broad spectrum of embedded systems projects.

ORDERING INFORMATION

TYPE NUMBER	PACKAGE		
	NAME	DESCRIPTION	VERSION
74HC595N	DIP16	plastic dual in-line package; 16 leads (300 mil); long body	SOT38-1
74HC595D	SO16	plastic small outline package; 16 leads; body width 3.9 mm	SOT109-1
74HC595DB	SSOP16	plastic shrink small outline package; 16 leads; body width 5.3 mm	SOT338-1
74HC595PW	TSSOP16	plastic thin shrink small outline package; 16 leads; body width 4.4 mm	SOT403-1
74HCT595N	DIP16	plastic dual in-line package; 16 leads (300 mil); long body	SOT38-1
74HCT595D	SO16	plastic small outline package; 16 leads; body width 3.9 mm	SOT109-1

PINNING

SYMBOL	PIN	DESCRIPTION
Q_0 to Q_7	15, 1 to 7	parallel data output
GND	8	ground (0 V)
Q_7'	9	serial data output
$\overline{MR}$	10	master reset (active LOW)
SH_{CP}	11	shift register clock input
ST_{CP}	12	storage register clock input
$\overline{OE}$	13	output enable (active LOW)
D_S	14	serial data input
V_{CC}	16	positive supply voltage

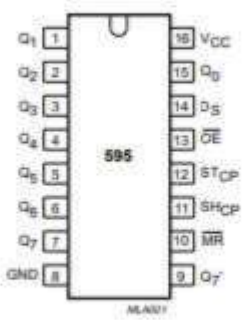


Fig.1 Pin configuration.

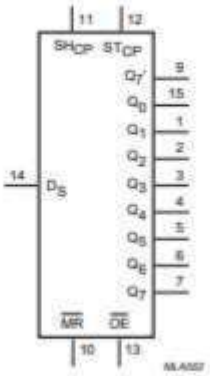
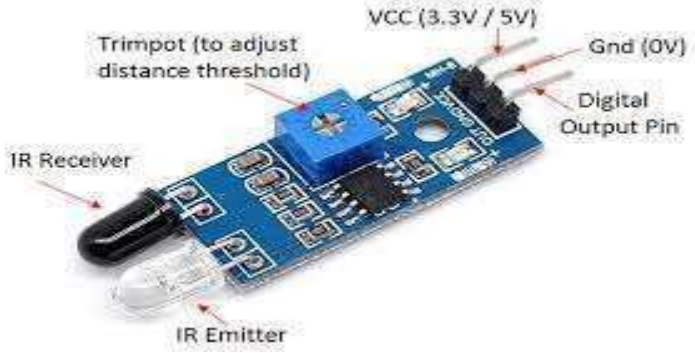


Fig.2 Logic symbol.

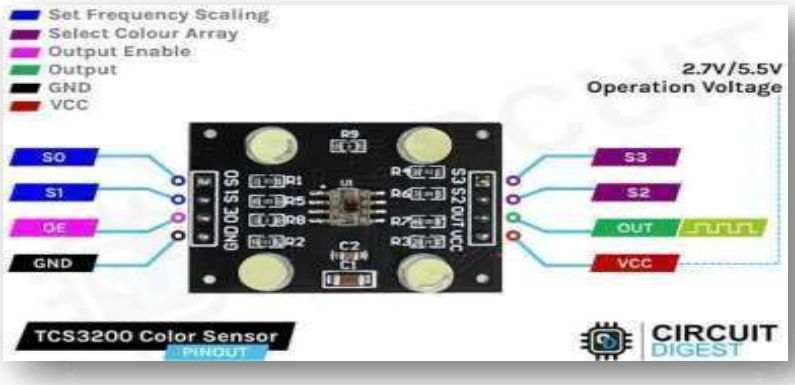
3. **L298N Motor Driver:** The L298N stands as a versatile integrated circuit (IC) renowned for its dual H-bridge motor driving capabilities, making it a staple in robotics and motor control domains. This IC facilitates the independent control of two motors, allowing for precise manipulation of their speed and direction. Its widespread adoption stems from its ability to efficiently manage the power distribution required for driving motors in various applications, ranging from robotic platforms to intricate motor control systems.

IR sensor: An infrared sensor (IR sensor) is a radiation-sensitive optoelectronic component with a spectral sensitivity in the infrared wavelength range 780 nm … 50 µm.



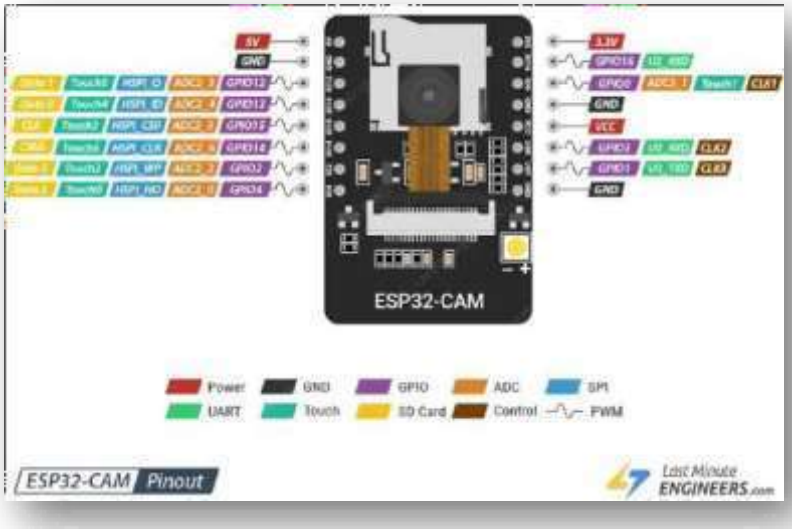
Color Sensor:

The TCS3200 and TCS3210 are programmable color light-to-frequency converters designed for high-resolution conversion of light intensity to frequency. They integrate configurable silicon photodiodes and a current-to-frequency converter on a single CMOS integrated circuit. Operating on a single supply (2.7 V to 5.5 V), they produce a square wave output with frequency directly proportional to light intensity. The converters feature programmable color and full-scale output frequency, direct communication with a microcontroller, power-down capability, low nonlinearity error (typically 0.2% at 50 kHz), and a stable temperature coefficient. The devices are housed in a low-profile, lead-free, and RoHS-compliant surface-mount package. The TCS3200 includes an 8x8 array of photodiodes with blue, green, red, and clear filters, while the TCS3210 features a 4x6 array. The photodiodes are interdigitated to minimize non-uniformity, and selection pins (S2 and S3) choose the active group of photodiodes. All photodiodes of the same color are connected in parallel.

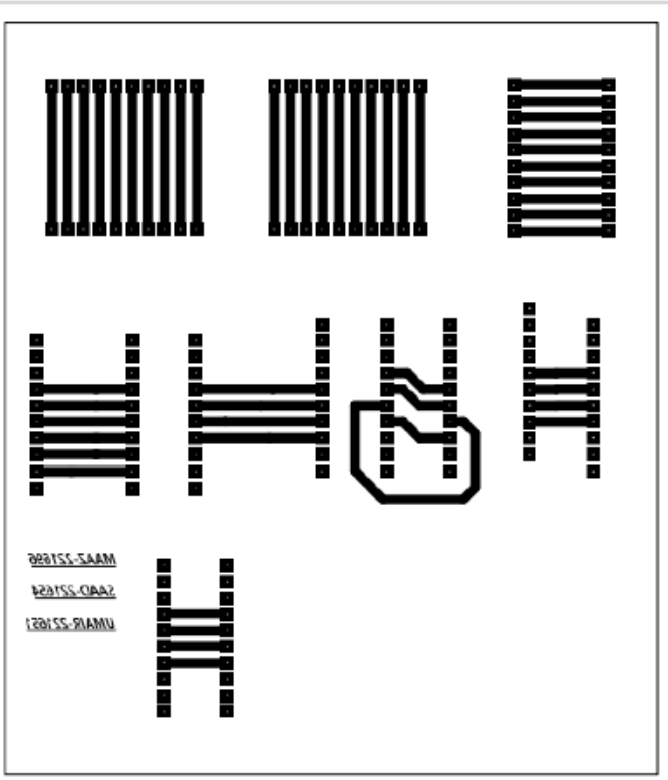


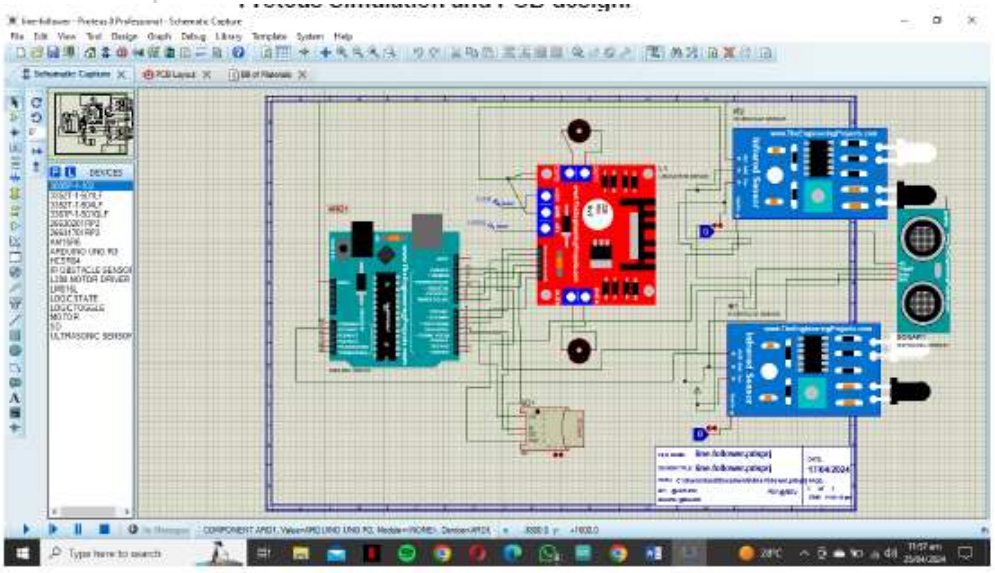
SD Card reader (Built-in ESP32-CAM):

An SD card reader module for Arduino facilitates the connection of an SD card to an Arduino microcontroller. It includes a slot for the SD card, interface circuitry, and communication protocols like SPI. The module is compatible with Arduino, allowing seamless data transfer and read/write operations for applications such as data logging. Connection to Arduino involves wiring the module to specific pins and using relevant libraries for file operations.



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





Chapter 6- Software/Firmware Design

6.1 Controller Selections with features:

Arduino Uno Controller :

1. **Microcontroller Powerhouse:**
   - The ATmega328P microcontroller provides a balanced blend of computational power and user-friendly simplicity, catering to both novice and experienced developers.
2. **Versatile Pin Configuration:**
   - With 14 digital and 6 analog pins, the Arduino Uno ensures seamless connectivity, offering an extensive range for interfacing with various sensors and actuators crucial for functionalities.
3. **Precision Motor Control with PWM:**
   - PWM pins on the Arduino Uno enable precise control of motors, contributing to and maneuverability as it navigates diverse environments.
4. **Reliable Serial Communication:**
   - Dedicated RX and TX pins facilitate reliable serial communication, allowing to seamlessly exchange data with other devices and systems. 5. **Convenient RESET Functionality:**
   - The RESET pin provides a straightforward mechanism for system resets, ensuring ease of debugging and troubleshooting during development and operation.
6. **User-Friendly Programming Environment:**
   - The Arduino Uno's compatibility with a vast array of libraries simplifies programming tasks, offering a user-friendly development environment that streamlines the implementation of diverse functionalities.

6.2 Software Design details:

1. Algorithmic Framework:

- *Real-time Decision-Making:* The software employs a sophisticated line-following algorithm, continuously analyzing QTR sensor data to make real-time decisions on motor speed adjustments, ensuring precise navigation along the designated path.

- *Adaptability through Modularity:* Designed with modularity in mind, the algorithmic framework allows for seamless integration of additional functionalities, promoting adaptability and future enhancements.

2. Sensor Integration:

- *Comprehensive Data Monitoring:* Robots software integrates data from the QTR-8A Reflectance Sensor Array, SR-04 Ultrasonic Sensor Module, and TCS3200 Color Sensor Module, enabling comprehensive environmental monitoring for line following, obstacle avoidance, and color sensing.

- *Dynamic Response Mechanisms:* Each sensor's data contributes to dynamic response mechanisms, ensuring the robot adapts swiftly to changes in the environment, whether it involves following a line, avoiding obstacles, or identifying specific colors.

3. Line Following Algorithm:

- *Precise Motor Control:* The line-following algorithm interprets QTR sensor readings to dictate precise motor control, adjusting speeds for optimal navigation and alignment with the designated path.
- *Continuous Iteration:* Operating in real-time, the algorithm continuously iterates, providing a dynamic response to variations in the track layout, including sharp turns and intersections.

4. Obstacle Avoidance:

- *Ultrasonic Detection:* Utilizing data from the SR-04 Ultrasonic Sensor, software incorporates an obstacle avoidance mechanism, promptly halting the robot and orchestrating maneuvers to navigate around detected obstacles.

- *Safety and Reliability:* The obstacle avoidance feature enhances safety and reliability, ensuring robot can traverse diverse environments without collision-related disruptions.

5. Color Sensing Functionality:

- *Identifying Specific Colors:* The TCS3200 Color Sensor empowers robot to identify and respond to specific colors, broadening its scope of applications in scenarios where color recognition is essential.

- *Versatility in Responses:* Whether halting upon detecting a specific color or altering its behavior, the color sensing functionality adds a layer of versatility to robots responses in different operational contexts.

6. Modularity and Future Enhancements:

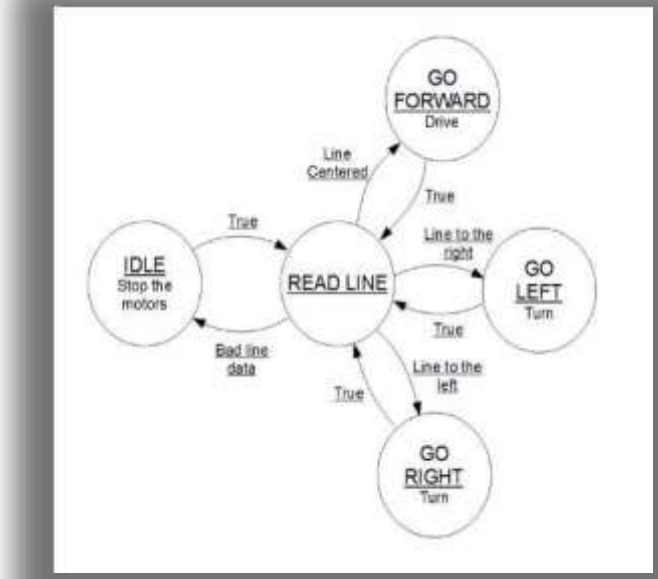
- *Seamless Integration:* The software design embraces modularity, allowing for seamless integration of additional features or enhancements, ensuring robot remains an adaptable platform for future advancements.

- *Scalability:* The modular approach facilitates scalability, permitting the incorporation of new technologies or functionalities without necessitating significant software overhauls.

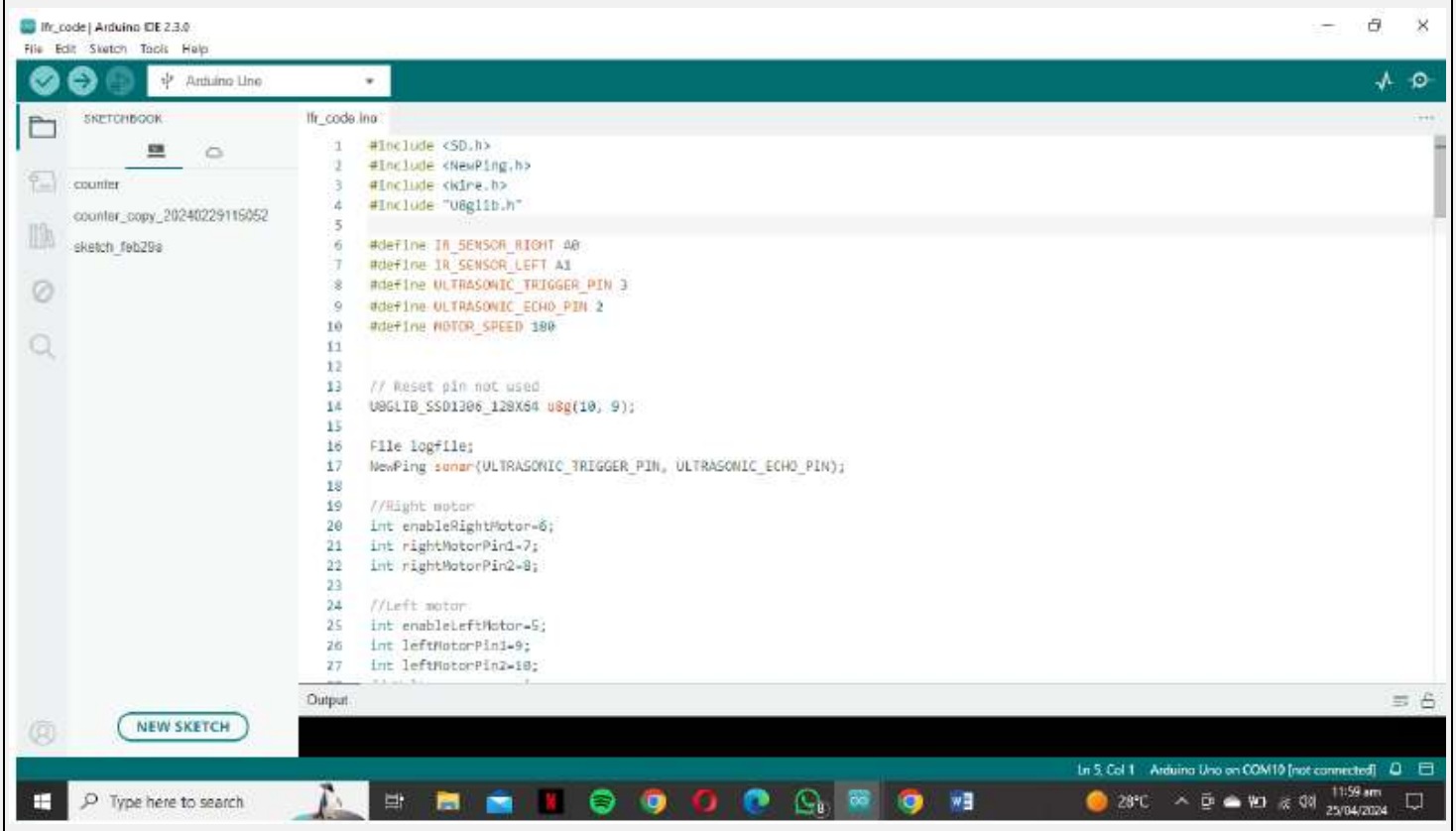
7. Code Optimization and Testing:

- *Efficiency Focus:* Emphasis on code efficiency ensures robot swift and accurate responses, optimizing resource utilization and promoting smooth operation in dynamic environments.
- *Thorough Testing Protocols:* Rigorous testing procedures guarantee the reliability of the software, validating the effectiveness of the algorithms and functionalities under various conditions.

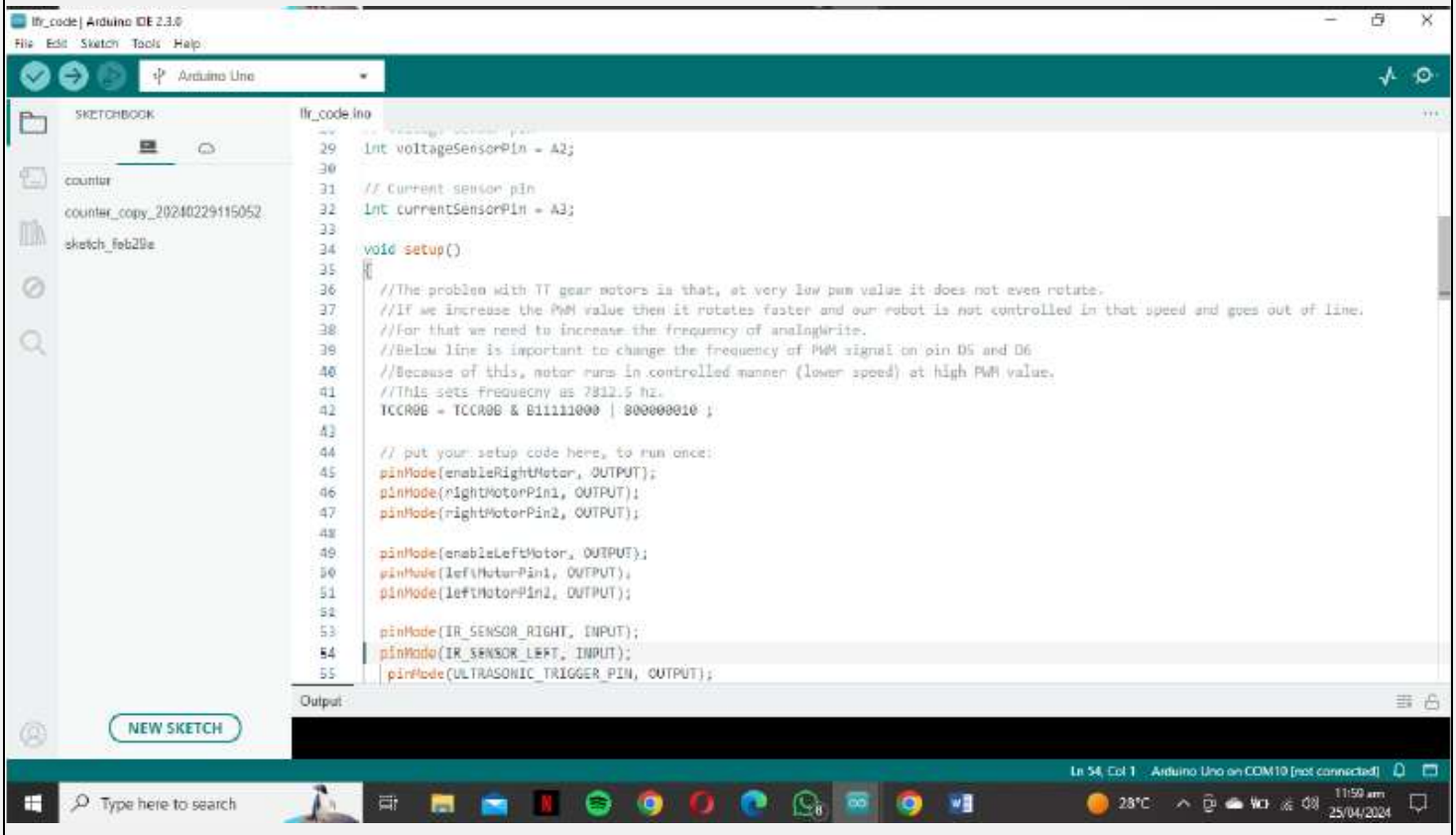
6.3 State Machine & System flow diagram.



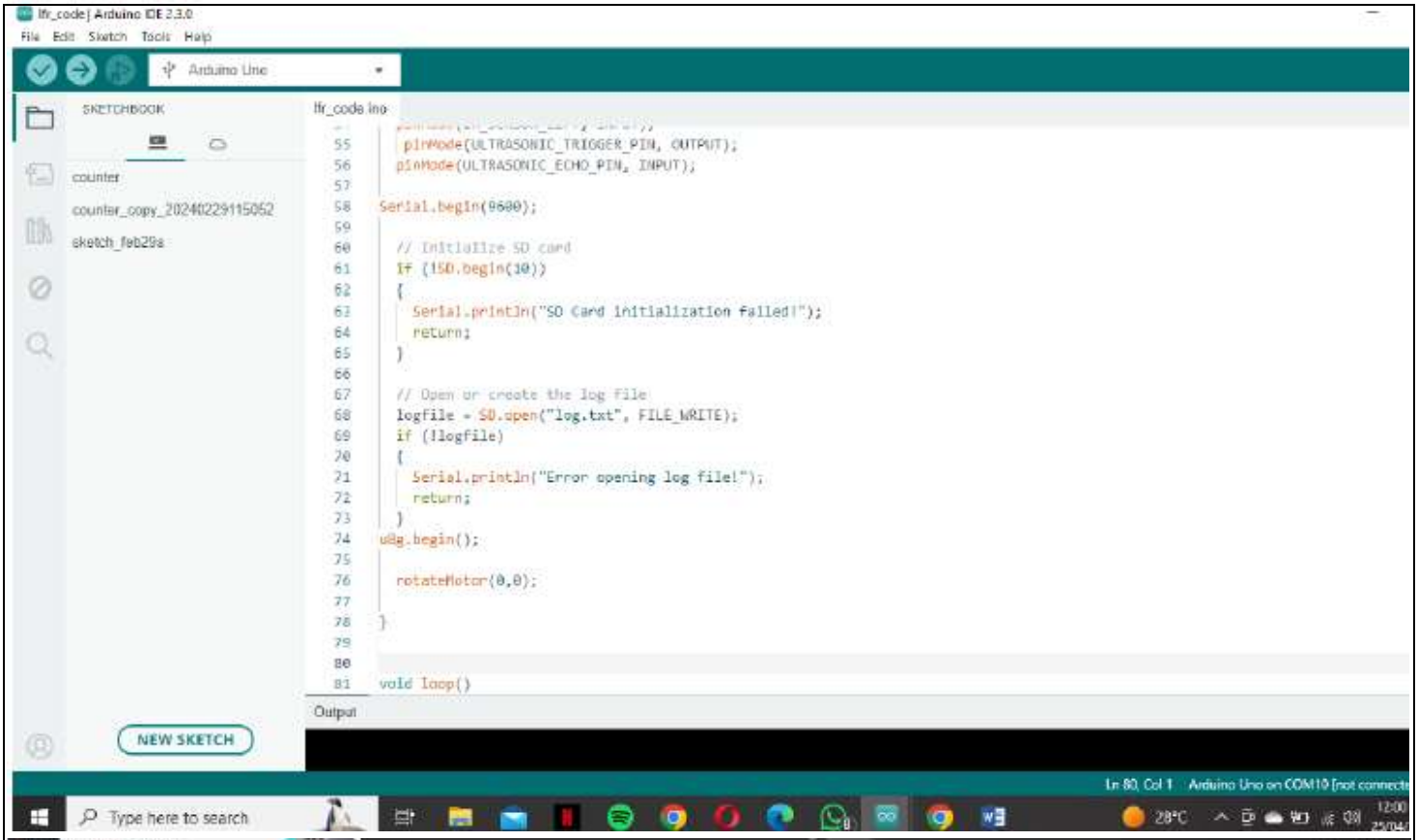
Programming

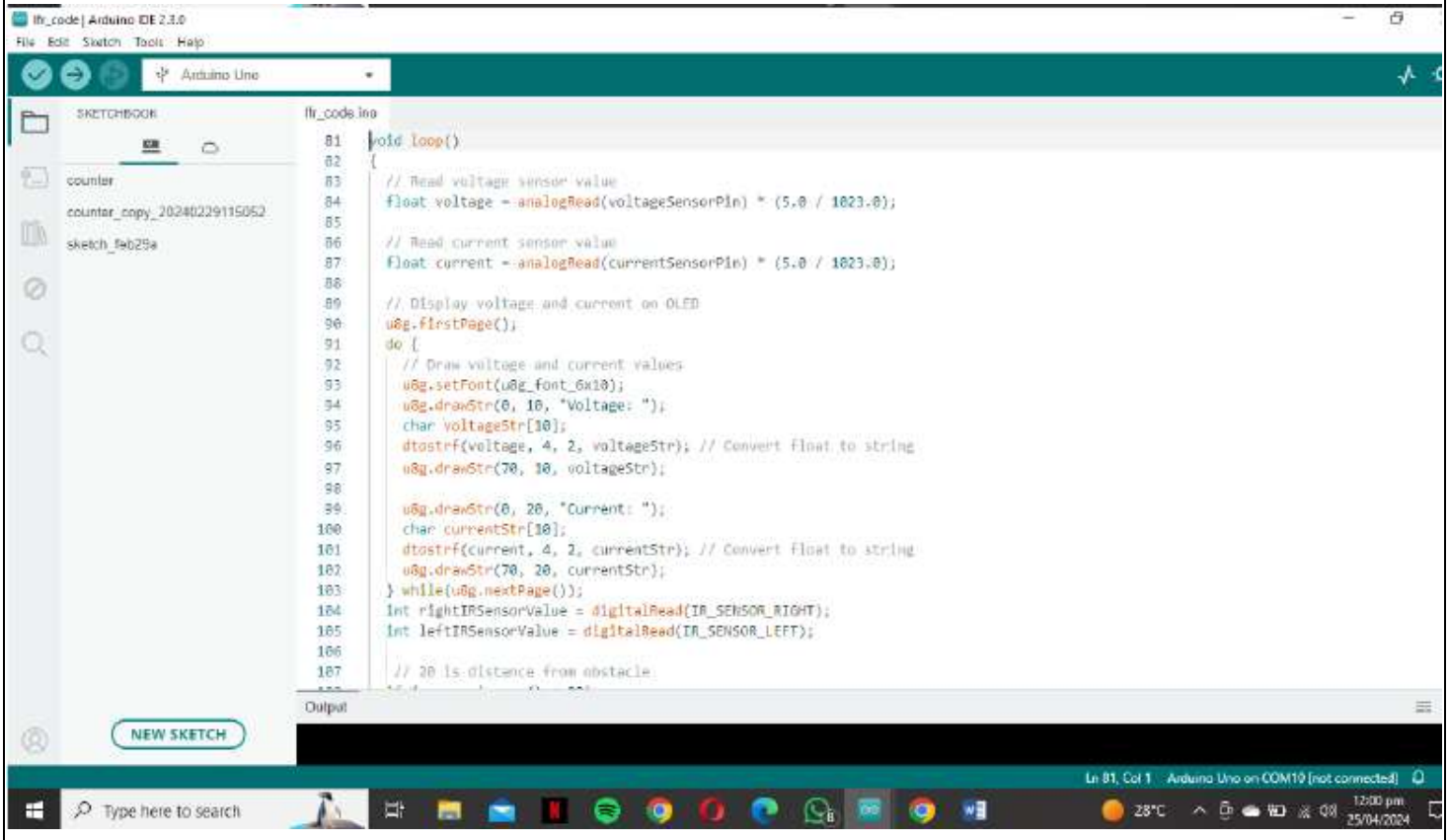


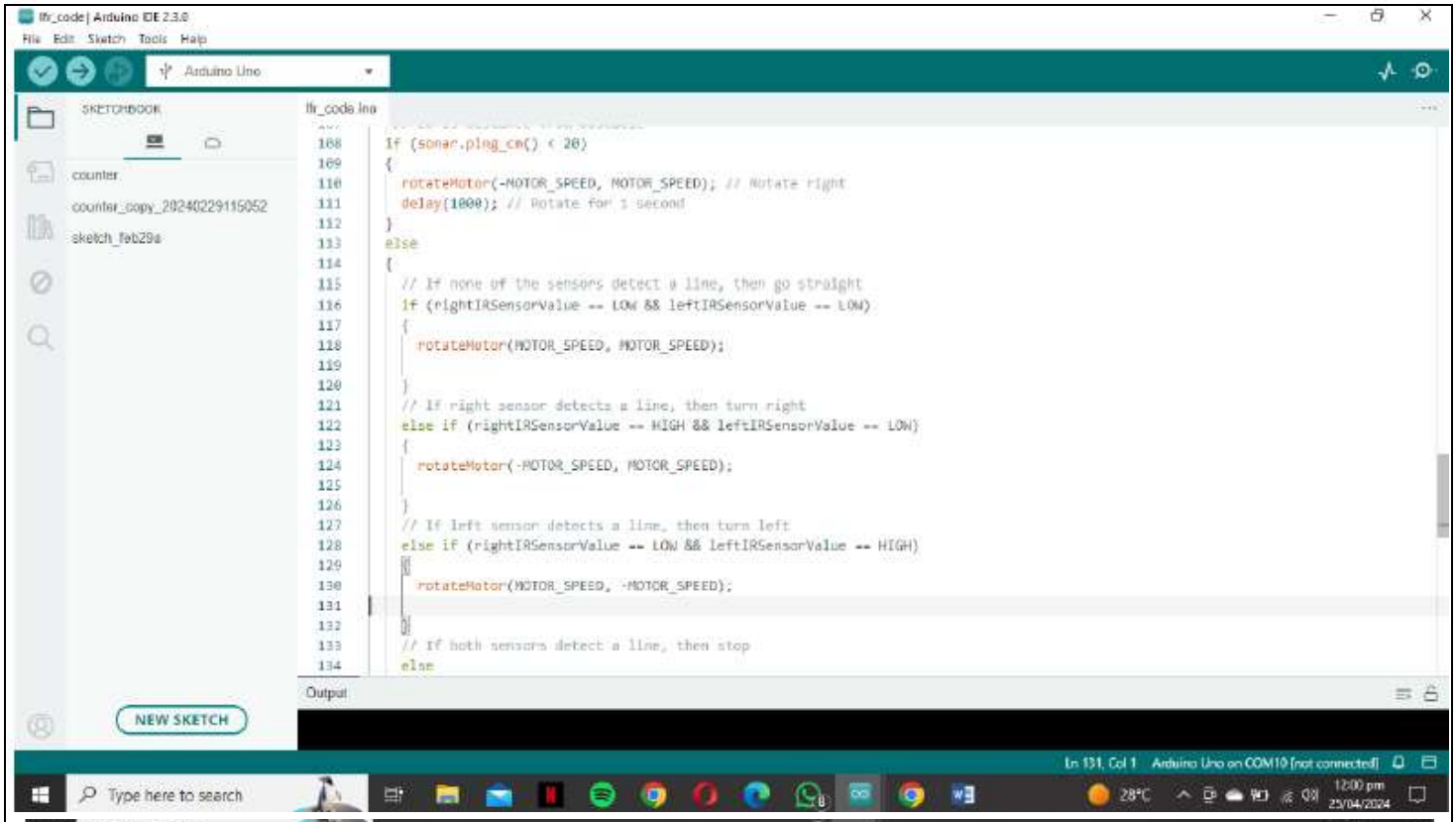

```
#include <SD.h>
#include <NewPing.h>
#include <Wire.h>
#include "U8glib.h"

#define IR_SENSOR_RIGHT A0
#define IR_SENSOR_LEFT A1
#define ULTRASONIC_TRIGGER_PIN 3
#define ULTRASONIC_ECHO_PIN 2
#define MOTOR_SPEED 180


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

File logfile;
NewPing sonar(ULTRASONIC_TRIGGER_PIN, ULTRASONIC_ECHO_PIN);

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

//Left motor
int enableLeftMotor=5;
int leftMotorPin1=9;
int leftMotorPin2=10;
```

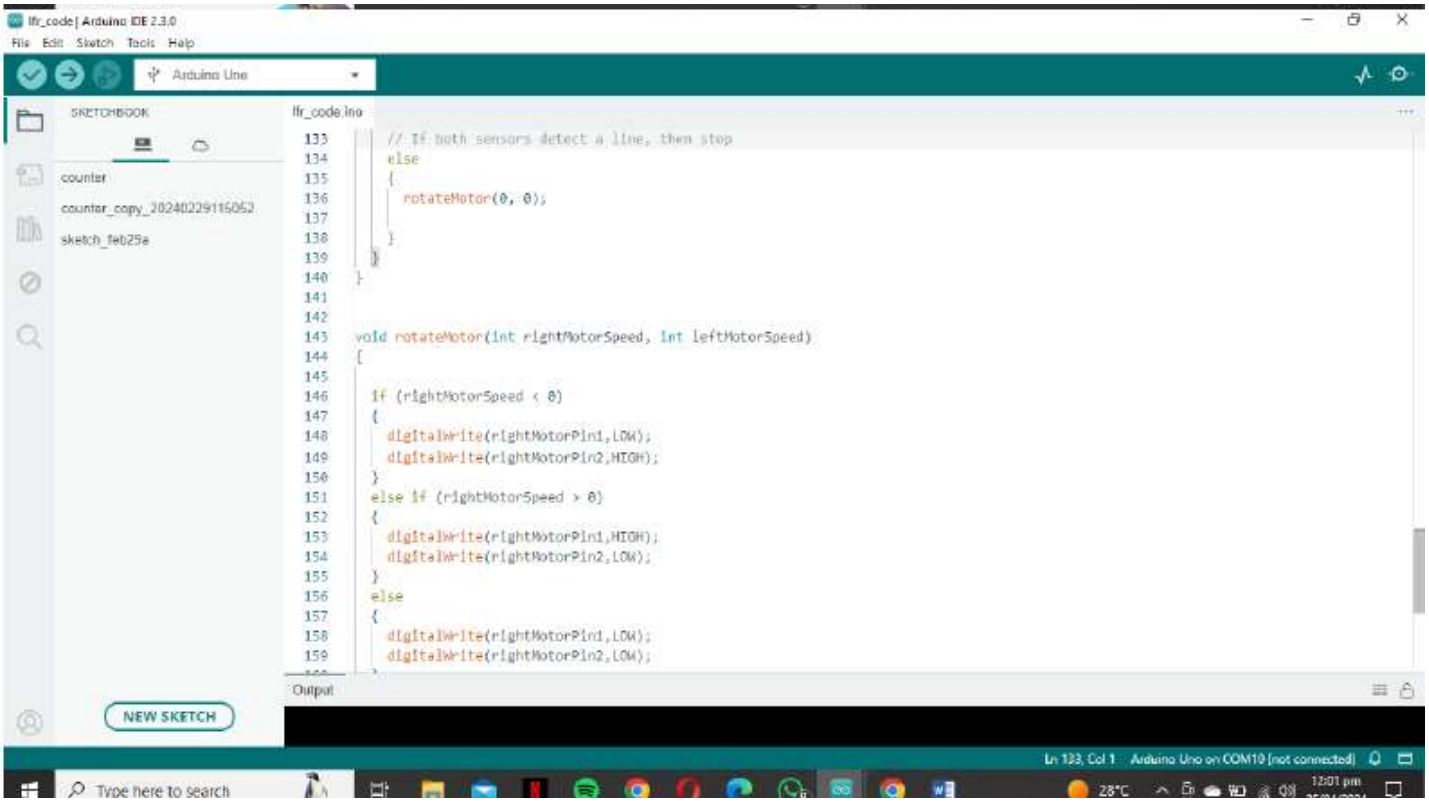


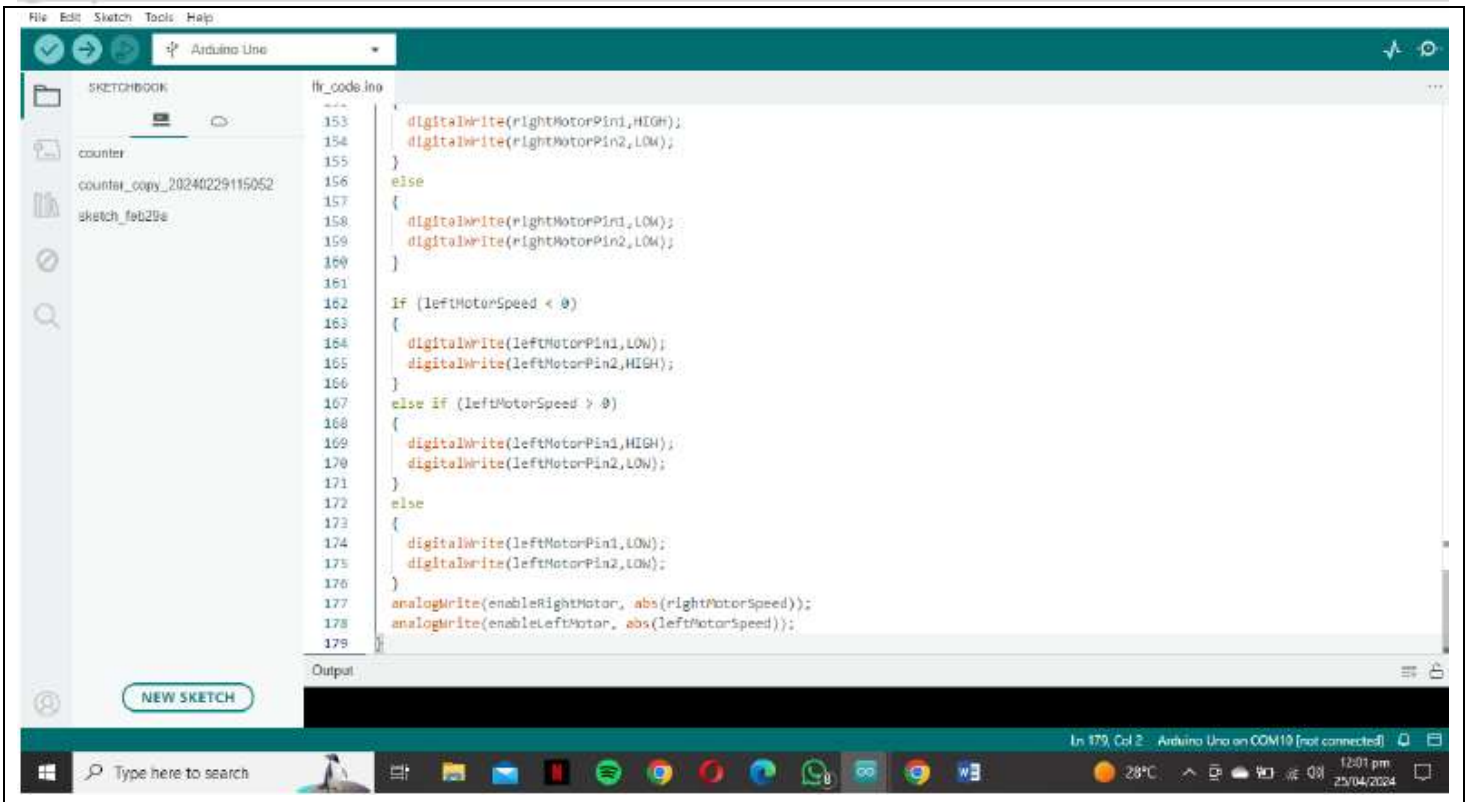

```
int voltageSensorPin = A2;

// Current sensor pin
int currentSensorPin = A3;

void setup()
{
  //The problem with TT gear motors is that, at very low pwm value it does not even rotate.
  //If we increase the PWM value then it rotates faster and our robot is not controlled in that speed and goes out of line.
  //For that we need to increase the frequency of analogWrite.
  //Below line is important to change the frequency of PWM signal on pin D5 and D6
  //Because of this, motor runs in controlled manner (lower speed) at high PWM value.
  //This sets frequecny as 7812.5 hz.
  TCCR0B = TCCR0B & B11111000 | B00000010 ;

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

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

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


lfr_code | Arduino IDE 2.3.0
File Edit Sketch Tools Help
Arduino Uno
SKETCHBOOK
lfr_code.ino
counter
counter_copy_20240229115052
sketch_feb29a
pinMode(ULTRASONIC_TRIGGER_PIN, OUTPUT);
pinMode(ULTRASONIC_ECHO_PIN, INPUT);
Serial.begin(9600);
// Initialize SD card
if (!SD.begin(10))
{
Serial.println("SD Card initialization failed!");
return;
}
// Open or create the log file
logfile = SD.open("log.txt", FILE_WRITE);
if (!logfile)
{
Serial.println("Error opening log file!");
return;
}
u8g.begin();
rotateMotor(0,0);
}
void loop()
Output
NEW SKETCH
Ln 80, Col 1 Arduino Uno on COM10 [not connecte
Type here to search
28°C


lfr_code | Arduino IDE 2.3.0
File Edit Sketch Tools Help
Arduino Uno
SKETCHBOOK
lfr_code.ino
counter
counter_copy_20240229115052
sketch_feb29a
void loop()
{
// Read voltage sensor value
float voltage = analogRead(voltageSensorPin) * (5.0 / 1023.0);
// Read current sensor value
float current = analogRead(currentSensorPin) * (5.0 / 1023.0);
// Display voltage and current on OLED
u8g.firstPage();
do {
// Draw voltage and current values
u8g.setFont(u8g_font_6x10);
u8g.drawStr(0, 10, "Voltage: ");
char voltageStr[10];
dtostrf(voltage, 4, 2, voltageStr); // Convert float to string
u8g.drawStr(70, 10, voltageStr);
u8g.drawStr(0, 20, "Current: ");
char currentStr[10];
dtostrf(current, 4, 2, currentStr); // Convert float to string
u8g.drawStr(70, 20, currentStr);
} while(u8g.nextPage());
int rightIRSensorValue = digitalRead(IR_SENSOR_RIGHT);
int leftIRSensorValue = digitalRead(IR_SENSOR_LEFT);
// 20 is distance from obstacle
Output
NEW SKETCH
Ln 81, Col 1 Arduino Uno on COM10 [not connected]
Type here to search
28°C
12:00 pm
25/04/2024


lfr_code | Arduino IDE 2.3.0
File Edit Sketch Tools Help
Arduino Uno
SKETCHBOOK
lfr_code.ino
counter
counter_copy_20240229115052
sketch_feb29a
NEW SKETCH
if (sonar.ping_cm() < 20)
{
rotateMotor(-MOTOR_SPEED, MOTOR_SPEED); // Rotate right
delay(1000); // Rotate for 1 second
}
else
{
// If none of the sensors detect a line, then go straight
if (rightIRSensorValue == LOW && leftIRSensorValue == LOW)
{
rotateMotor(MOTOR_SPEED, MOTOR_SPEED);
}
// If right sensor detects a line, then turn right
else if (rightIRSensorValue == HIGH && leftIRSensorValue == LOW)
{
rotateMotor(-MOTOR_SPEED, MOTOR_SPEED);
}
// If left sensor detects a line, then turn left
else if (rightIRSensorValue == LOW && leftIRSensorValue == HIGH)
{
rotateMotor(MOTOR_SPEED, -MOTOR_SPEED);
}
// If both sensors detect a line, then stop
else
Output
Ln 131, Col 1 Arduino Uno on COM10 [not connected]
Type here to search
28°C
12:00 pm
25/04/2024


lfr_code | Arduino IDE 2.3.0
File Edit Sketch Tools Help
Arduino Uno
SKETCHBOOK
lfr_code.ino
counter
counter_copy_20240229115052
sketch_feb29a
NEW SKETCH
// If both sensors detect a line, then stop
else
{
rotateMotor(0, 0);
}
}
}
void rotateMotor(int rightMotorSpeed, int leftMotorSpeed)
{
if (rightMotorSpeed < 0)
{
digitalWrite(rightMotorPin1,LOW);
digitalWrite(rightMotorPin2,HIGH);
}
else if (rightMotorSpeed > 0)
{
digitalWrite(rightMotorPin1,HIGH);
digitalWrite(rightMotorPin2,LOW);
}
else
{
digitalWrite(rightMotorPin1,LOW);
digitalWrite(rightMotorPin2,LOW);
Output
Ln 133, Col 1 Arduino Uno on COM10 [not connected]
Type here to search
28°C


File Edit Sketch Tools Help
Arduino Uno
SKETCHBOOK
counter
counter_copy_20240229115052
sketch_feb29a
NEW SKETCH
llr_code.ino
digitalWrite(rightMotorPin1,HIGH);
digitalWrite(rightMotorPin2,LOW);
}
else
{
digitalWrite(rightMotorPin1,LOW);
digitalWrite(rightMotorPin2,LOW);
}
If (leftMotorSpeed < 0)
{
digitalWrite(leftMotorPin1,LOW);
digitalWrite(leftMotorPin2,HIGH);
}
else if (leftMotorSpeed > 0)
{
digitalWrite(leftMotorPin1,HIGH);
digitalWrite(leftMotorPin2,LOW);
}
else
{
digitalWrite(leftMotorPin1,LOW);
digitalWrite(leftMotorPin2,LOW);
}
analogWrite(enableRightMotor, abs(rightMotorSpeed));
analogWrite(enableLeftMotor, abs(leftMotorSpeed));
}
Output
Ln 179, Col 2 Arduino Uno on COM10 [not connected]
Type here to search
28°C
12:01 pm
25/04/2024

Chapter 7- Simulations and final Integrations

7.1 Integrations and testing all hardware and software component separately.

- **Integration and Component Testing Strategy:**
  The integration and testing phase involves a systematic approach to ensure the seamless collaboration of both hardware and software components. The following steps outline the strategy for integrations and testing:
- **Component Analysis:**
  Review and analyze datasheets for all hardware components to identify optimal operating conditions, including voltage and current requirements.
- **Individual Component Testing:**
  Utilize Digital Multimeter (DMM) to conduct individual tests on each hardware component.
  Replace malfunctioning components to ensure a fully functional set.
- **Integration Planning:**
  Develop a comprehensive plan for integrating hardware components, ensuring correct connections and preventing potential issues such as short circuits.
- **Software Compilation:**
  Compile Arduino code to ensure correctness and compatibility with the ATmega328P microcontroller.

Chapter 9- Project management

Bill of Materials:

Bill Of Materials for line-follower

Design Title	line-follower
Author	
Document Number	
Revision	
Design Created	Friday, 15 March 2024
Design Last Modified	Wednesday, 17 April 2024
Total Parts In Design	11

11 Miscellaneous

Quantity	References	Value	Stock Code	Unit Cost
1	/1	Current Sensor		Rs100.00
1	/2	ESP32		Rs1,150.00
1	/3	OLED display		Rs500.00
1	/4	PCB BOARD		Rs400.00
1	/5	Voltage Sensor		Rs100.00
1	ARD1	ARDUINO UNO R3		Rs1,500.00
2	IR1, IR2	IR OBSTACLE SENSOR		Rs300.00
1	L1	L298 MOTOR DRIVER		Rs350.00
1	SD1	SD		Rs150.00
1	SONAR1	ULTRASONIC SENSOR		Rs200.00
Sub-totals:				Rs5,050.00
Totals:				Rs5,050.00

CONCLUSION

In essence, the development of our robot signifies a significant leap forward in the realm of embedded systems and robotics. Through the integration of cutting-edge technologies such as sensors and microcontrollers, we've engineered a versatile and adaptable robot. Capable of proficiently following lines, avoiding obstacles, and accurately sensing colors, our robot owes its functionality to sophisticated algorithms and seamlessly coordinated sensors.

The sleek and wire-free design not only enhances the aesthetic appeal but also ensures smooth mobility. We've dedicated considerable effort to ensuring our robot's ease of upgradability and modification to meet future requirements.

From conceptualization to construction and rigorous testing, our journey exemplifies the practical applications of embedded systems in real-world scenarios, whether it be in educational settings, public events, or practical tasks like indoor navigation and environmental monitoring. With its user-friendly programming interface and invaluable features, our robot stands as an exhilarating advancement in the domain of intelligent robotics.

Links:

Github:

Linkedin:

Youtube: