FACULTY OF MANAGEMENT

ERR3036
MOBILE ROBOTS AND DRONES

Final Project Report

T1 2023/2024

Lecturer - Prof Fabian kung wai Lee

Name	ABDELKERIM ALI HASSAN
ID	1211302792
Major	I.R

TABLE OF CONTENT:

1) Introduction

The goal of creating autonomous systems that can navigate and interact with their environment has been a long-standing pursuit in the field of robotics. This report focuses on the design and development of one such system - a robotic platform equipped with sensors for obstacle avoidance and decision-making capabilities.

The primary objective of this project is to engineer a robotic system that can autonomously move through various environments while intelligently avoiding obstacles. This is achieved through the integration of mechanical, electrical, and software components to achieve seamless functionality.

An essential part of a robotic system is mechatronics, a combination of mechanical engineering, electronic engineering, and computer science. Propulsion is provided by DC motors, environmental perception is provided by sensors, and decision-making is done by microcontrollers. Essentially, it emulates some of the rudimentary features of autonomous systems.

2) I) Planning

* Run Simple Code to Operate the LED Light

Object: To establish the LED light

Tasks:

- Write a C/C++ code to control the LED light, ON/OFF accordingly behaved based on your code.

* Run Simple Code to Operate Two Motors

- Objective: To establish basic motor control functionality.

Tasks:

- Research and select appropriate motor control libraries or code examples.
- Write and test code to operate two DC motors in a simple forward and backward motion.
- Write and test a code to operate The speed of the two DC motor in a forward and backward.

* Add Sensor for Testing

- Objective: Integrate sensors to enable obstacle detection.

Tasks:

- Research and select suitable sensors for obstacle detection (e.g., ultrasonic, infrared, IR sensor).
- Connect sensors to the Arduino and verify functionality.
- Modify motor control code to incorporate sensor readings for obstacle avoidance.

* Prototype Development and Testing

- Objective: Build a prototype of the robotic system for initial testing.

Tasks:

- Design and assemble the mechanical structure of the robot, including mounting points for motors and sensors.
- Install electronic components (Arduino, motor drivers, sensors) and wire them according to the electrical wiring diagram.
- Upload motor control and sensor integration code to the Arduino.
- Conduct initial testing to ensure proper functionality of both motors and sensors.

* Run and Troubleshoot Prototype.

- Objective: Identify and resolve any issues with the prototype.

Tasks:

- Test the prototype in various environments to assess its obstacle avoidance capability.
- Monitor sensor readings and motor responses to detect any inconsistencies or errors.
- Troubleshoot and debug code or hardware problems as they arise.

* Finalize Design and Implementation.

- Objective: Refine the prototype into the final version of the robotic system.

Tasks:

- Incorporate any necessary design changes or improvements based on feedback from testing.
- Ensure all components are securely mounted and connections are stable.
- Conduct final testing to validate the functionality and performance of the robotic system.
- Document the design process, including any modifications made during prototyping and finalization.

* Review and Documentation.

- Objective: Prepare the project documentation for submission.

Tasks:

- Compile all documentation, including design diagrams, code snippets, and testing results.
- Review the report for completeness and accuracy.
- Make any necessary revisions or additions.
- Finalize the report and ensure it is ready for submission by the deadline.

II) **Gantt Chart:**

Task	Duration	Start Date	End Date
Research and buy things	**2 to 3 weeks**	**6/11/2023-**	**17/11/2023**
Run Basic with Arduino uno board, Dc motor drive, and dc motor.	**3 weeks**	**20/11/2023**	**5/12/2023**
Build with Prototype based	**1 weeks**	**6/12/2023**	**12/12/2023**
Component procurement	**3 days**	**13/12/2023**	**15/12/2023**
Mechanical Assembly	**3 days**	**15/12/2023**	**18/12/2023**
Electrical Wiring	**3 days**	**20/12/2023**	**23/12/2023**
Programming	**2 weeks**	**30/12/2023**	**8/1/2024**
Testing and Debugging	**3 weeks**	**9/1/2024**	**29/1/2024**
Finalize the design	**5 days**	**29/1/2024**	**4/2/2024**
Demonstration	**1 day**	**5/2/2024**	**5/22/2024**

III) **Budget Information:**

Component	Quantity	Unit Price (RM)	Total Cost (Rm)
Arduino Uno	1	29.00	29.00
DC Motor driver (L293D Motor Driver)	1	45.00	45.00
L298N DC Motor Driver	1	5.97	5.97
LiPo Rechargeable battery 3s 11.1V 30C 2200mAH	1	98.00	98.00
Ultrasonic Sensor	1	4.90	4.90
Infrared Sensor	2	1.90	3.80
push button switch	1	1.15	1.15
Male to Female Jumper Wire	1	4.60	4.60
Male to Male Jumper Wire	1	4.60	4.60
Breadboard 8.5x5.5cm (400 holes)	1	9.49	9.49
Mini DC Motor + wheel kit	2	2.90	5.80
Total Budget	-	-	**212,31**

3) Methodology:

I) Research and Planning:

- Conducted extensive research on existing robotic systems, sensor technologies, motor control methods, and wireless communication protocols.
- Defined project objectives, scope, and requirements based on research findings and project specifications.
- Established a timeline and allocated tasks to team members, ensuring efficient project management.

II) Component Selection and Acquisition:

- Identified and selected appropriate components based on project requirements, including DC motors, sensors (such as ultrasonic or infrared sensors for obstacle detection), motor drivers, microcontrollers (e.g., Arduino)
- Procured necessary components within the allocated budget, considering factors such as quality, compatibility, and availability.

III) Initial Experiments:

- Conducted preliminary experiments to validate the functionality and performance of individual components.
- Tested DC motors to assess speed control, torque, and directional movement.
- Experimented with various sensor types to evaluate their accuracy, range, and responsiveness in detecting obstacles.
- Tested motor drivers to ensure compatibility with the selected motors and microcontroller.

IV) Integration and Prototyping:

- Integrated components into a prototype robotic platform, considering mechanical, electrical, and software integration aspects.
- Designed and assembled the mechanical structure of the robot, ensuring stability, maneuverability, and space for component placement.
- Wired electrical connections according to the schematic diagram, ensuring proper power distribution and signal transmission.
- Programmed the microcontroller (e.g., Arduino) to interface with sensors, control motors, and implement obstacle avoidance algorithms.

V) Testing and Debugging:

- Conducted comprehensive testing of the robotic system to validate its functionality and performance.
- Tested the robot's ability to navigate autonomously in various environments, including obstacle detection and avoidance.
- Identified and addressed any issues or malfunctions through iterative testing and debugging processes.
- Fine-tuned parameters and algorithms to optimize the robot's behavior and responsiveness.

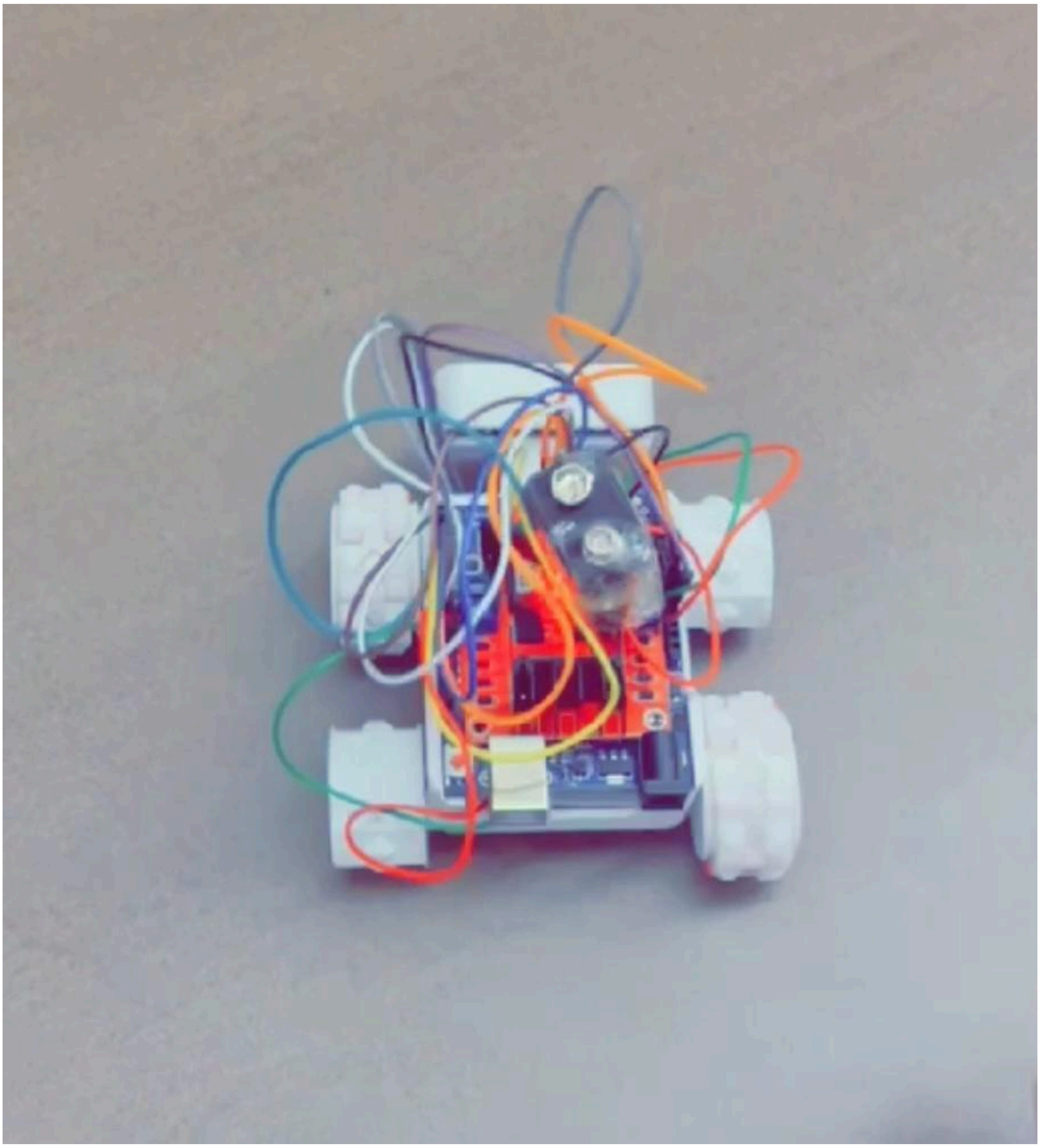
VI) Documentation and Reporting:

- Documented the design process, methodology, implementation details, and test results.
- Prepared technical documentation including schematics, wiring diagrams, code documentation, and design specifications.
- Compiled the individual report according to the provided structure, ensuring clarity, coherence, and adherence to the specified guidelines.

4) Detailed design:

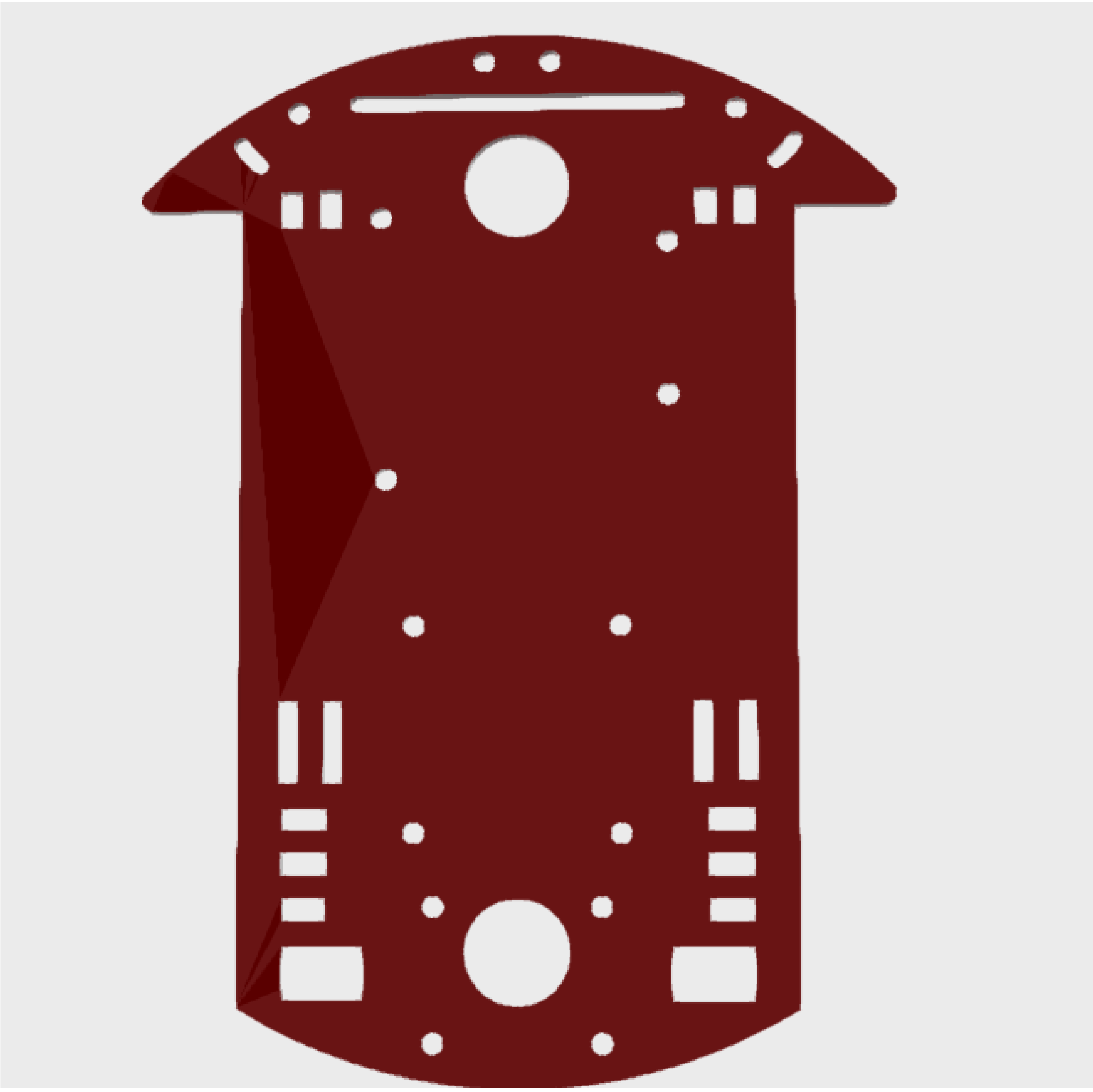
I)Mechanical Design

a)First Prototype:

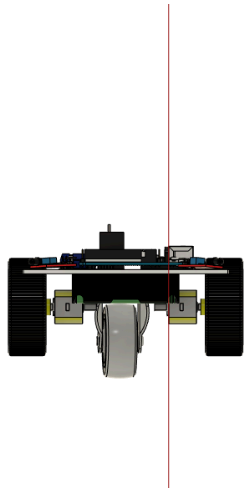


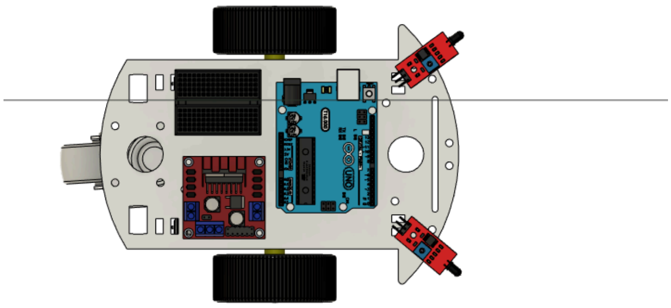
b)Final Prototype:

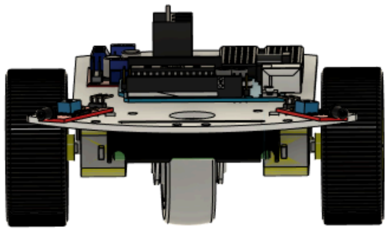
1.Robot based:

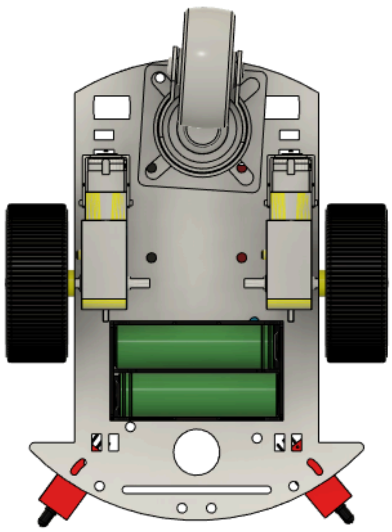


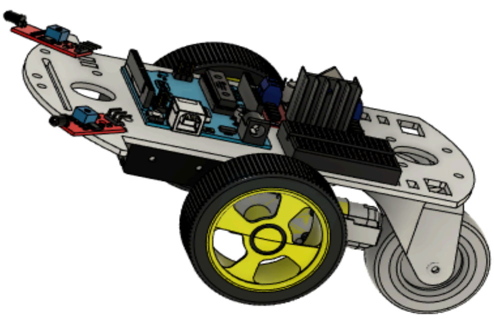
2.Full version of robot design:



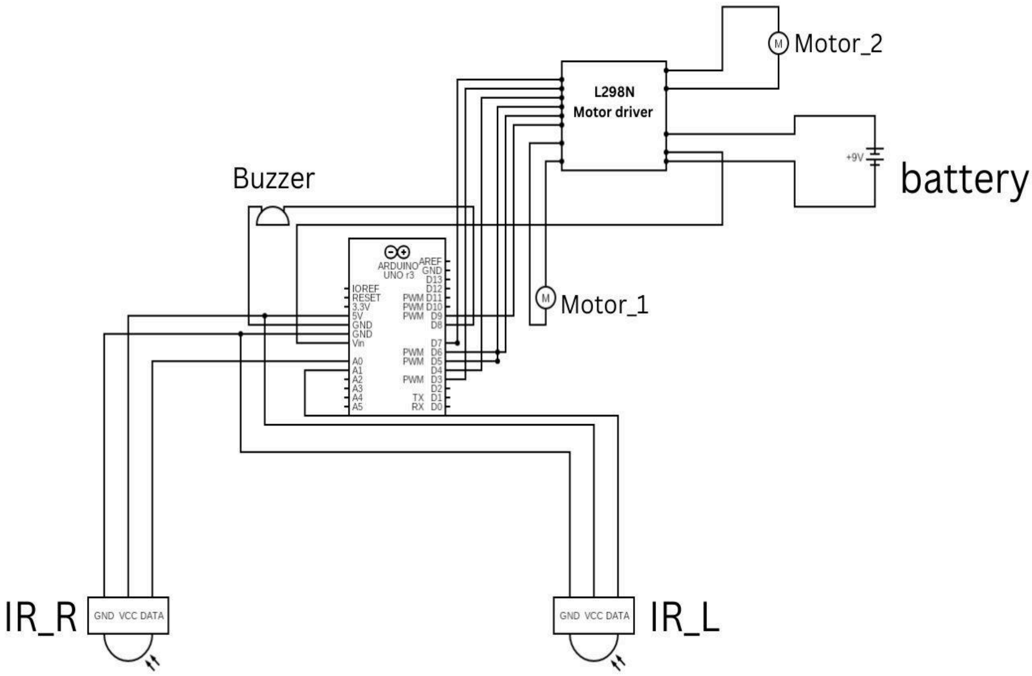








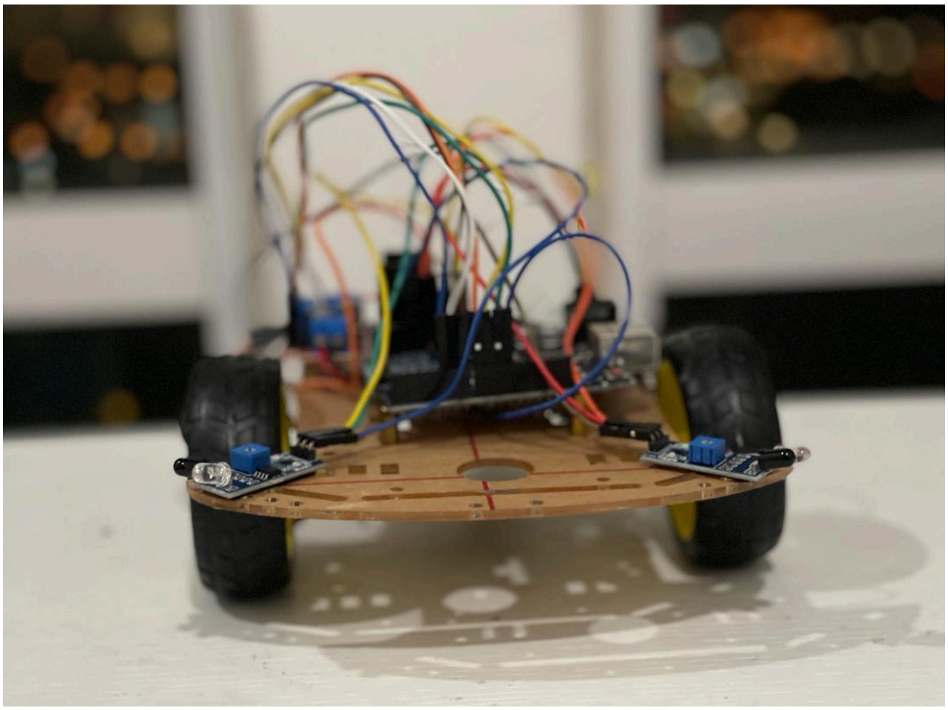
II) Electrical wiring diagram:



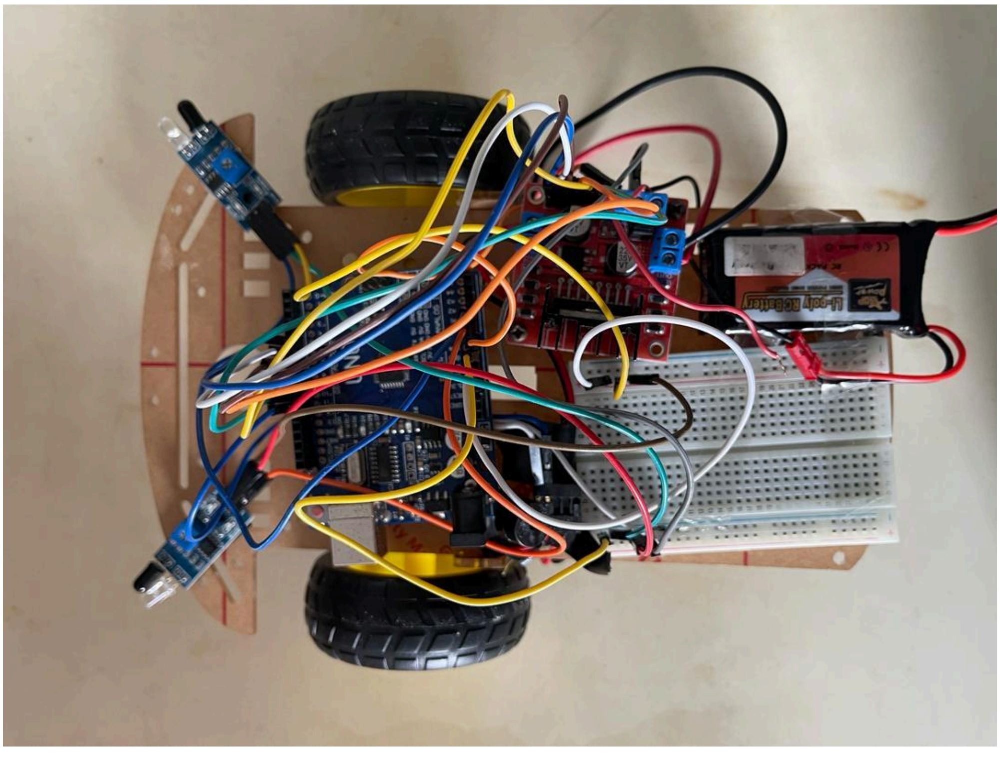
III) Flow-chat:

5) Result:

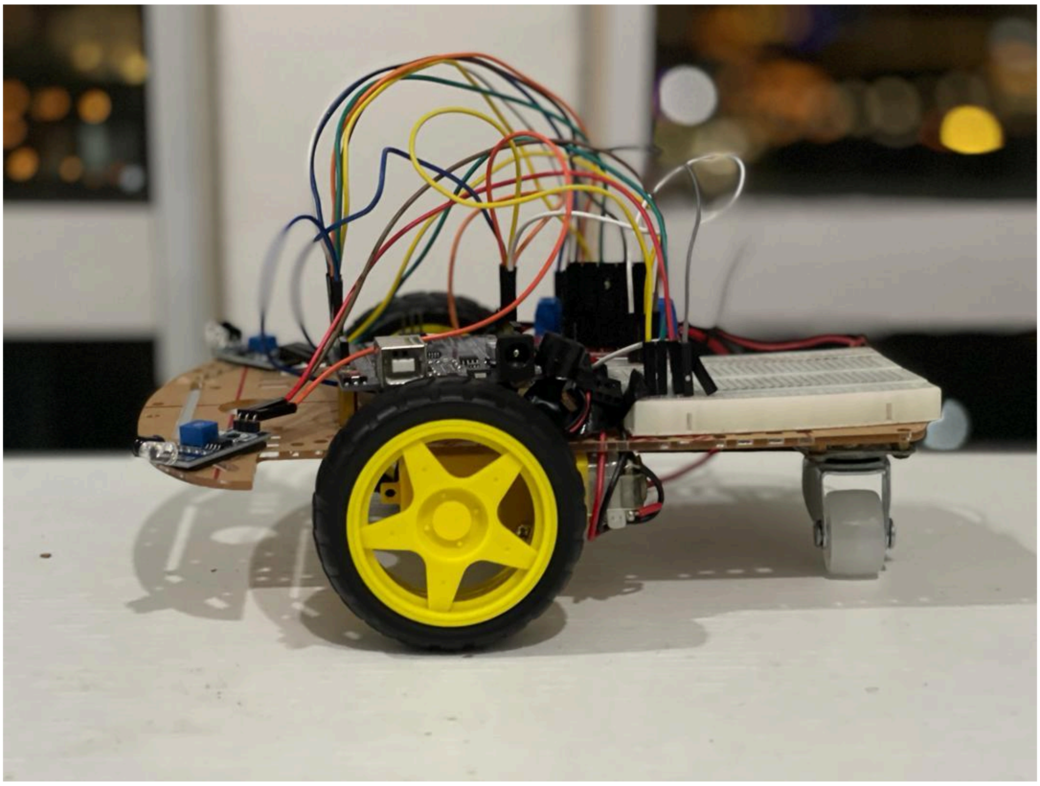
Front view:



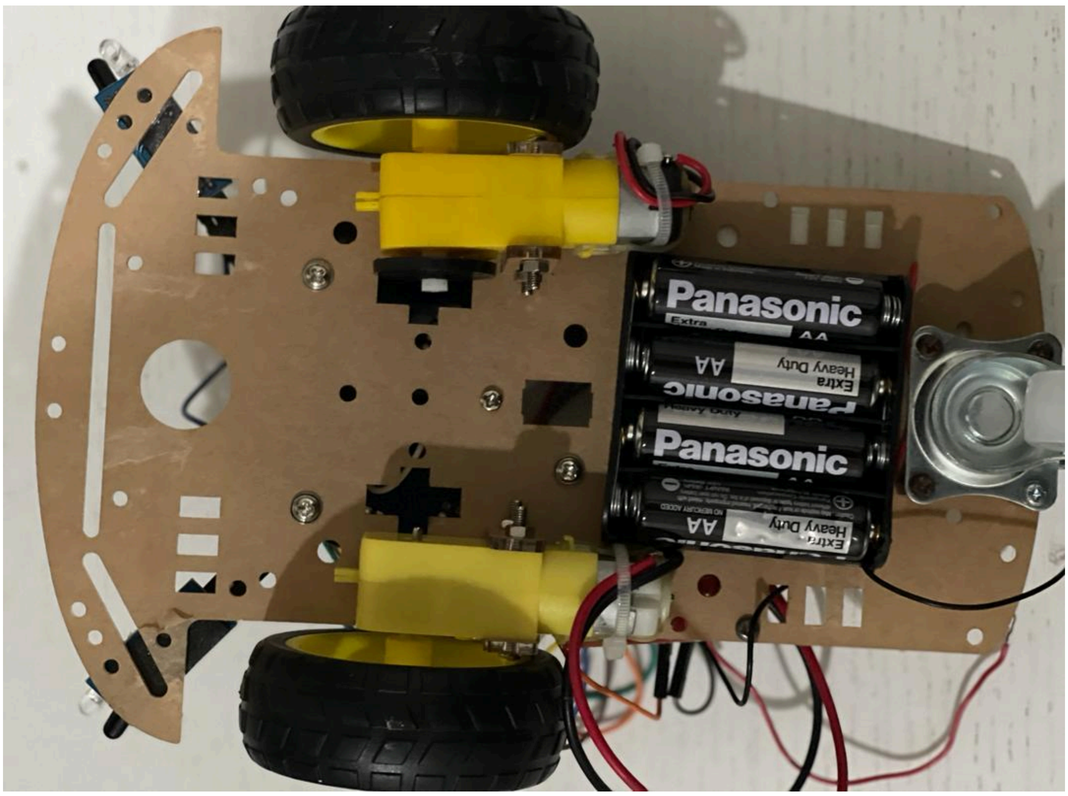
TOP VIEW:



SIDE VIEW:



Down view:



5.A breaf explanation about this robot:

The robot is designed to avoid obstacles using infrared sensors placed on its left (IR_L) and right (IR_R) sides. The code controls the robot's motors by defining input and output pins, which allows it to move forward, turn left, turn right, or stop depending on the sensor readings. The robot is equipped with a buzzer to indicate startup. During its operation, the robot continuously checks for obstacles using the infrared sensors. If there are no obstacles, the robot moves forward slowly. If there are obstacles, it adjusts its movement independently by turning left or right. The code includes functions for accurate motor speed control, resulting in smooth and controlled movements. Although the robot faced challenges during its development, it successfully achieved its goal of navigating its environment independently, avoiding obstacles, and demonstrating its capabilities through a final demonstration.

6)Discussion

Throughout the development of the robotic system, we encountered many obstacles that we were able to overcome effectively. One of the major challenges was integrating different components like DC motors, sensors, motor drivers, and wireless communication modules into a cohesive and functional system. This required careful planning and testing to ensure that each component was compatible and functioned properly.

The main issue that we faced in our first Prototype was making the machine move smoothly and continuously, However, we overcame this issue in our final design model

One of the most significant difficulties was the implementation of the obstacle avoidance algorithm. It required accurate sensor readings, robust decision-making logic, and efficient motor control to ensure the robot could navigate safely through its environment. We had to test and refine the algorithm repeatedly to optimize its performance and ensure reliable obstacle avoidance.

One that I overcome again was that any time you want to run your machine, you need to make sure that all components are functioning correctly otherwhis you would faced some unexpected behavoir from your machine e.g: an ultrasonic sensor is reading the distance of not longer than 5 cm in that case your machine will keep flipping.

We also paid meticulous attention to the mechanical design of the robot. We made sure that its structure was sound and that it could move efficiently. Technical drawings and Fusion 360 models were critical in visualizing and refining the physical layout, ensuring proper positioning and securing of components.

Despite these challenges, we were able to achieve our objectives and deliver a fully functional robotic system capable of autonomously navigating and avoiding obstacles. During the final demonstration, the robot showcased its adept maneuvering through obstacles and responsive behavior to changes in its surroundings.

Conclusion

In conclusion, the development of the robotic system was a rewarding and educational experience for us. Such as developing a real machine, It provided valuable insights into robotics, electronics, and programming, while also fostering teamwork and problem-solving skills. Looking ahead, we can further refine and expand the robot's capabilities, such as incorporating additional sensors for enhanced perception or integrating advanced control algorithms for more precise navigation. Overall, the project demonstrated the feasibility and potential applications of autonomous robotic systems in real-world scenarios.

Appendix:

Arduino source code:

```
#define IN1 3
#define IN2 4
#define IN3 5
#define IN4 6
#define ENA 7
#define ENB 9
#define IR_L A0
#define IR_R A1
#define BUZZ 8

// Motor 1 pins configuration
#define m1sp ENA
#define m1p1 IN1
#define m1p2 IN2

// Motor 2 pins configuration
#define m2sp ENB
#define m2p1 IN3
#define m2p2 IN4

void setup()
{
  Serial.begin(9600);
  pinMode(IN1, OUTPUT);
  pinMode(IN2, OUTPUT);
  pinMode(IN3, OUTPUT);
  pinMode(IN4, OUTPUT);
```

```
  pinMode(ENA, OUTPUT);
  pinMode(ENB, OUTPUT);
  pinMode(IR_L, INPUT);
  pinMode(IR_R, INPUT);
  digitalWrite(BUZZ, HIGH);
  delay (1000);
  digitalWrite(BUZZ, LOW);


}

void loop()
{
  // Check if obstacles are detected by the infrared sensors
  bool noObstacleRight = digitalRead(IR_R) == HIGH;
  bool noObstacleLeft = digitalRead(IR_L) == HIGH;

  Serial.print("IR Right: ");
  Serial.print(noObstacleRight);
  Serial.print(" IR Left: ");
  Serial.println(noObstacleLeft);

  // If both sensors don't detect an obstacle, move
  if (noObstacleRight && noObstacleLeft) {
    forwardSlow();
  }
  // If only the right sensor detects an obstacle, turn left
  else if (noObstacleRight) {
    moveLeft();
    delay(100);
  }
  // If only the left sensor detects an obstacle, turn right
  else if (noObstacleLeft) {
    moveRight();
    delay(100);
  }
  else {
    stopMoving();
    delay(500);
  }
}

// Function to control motor speed
void controlMotorSpeed(int nSpeed)
```

```
{
  // Motor 1 control
  if (nSpeed > 0)
  {
    analogWrite(m1sp, nSpeed);
    digitalWrite(m1p1, HIGH);
    digitalWrite(m1p2, LOW);
  }
  else if (nSpeed < 0)
  {
    analogWrite(m1sp, -nSpeed);
    digitalWrite(m1p1, LOW);
    digitalWrite(m1p2, HIGH);
  }
  else
  {
    analogWrite(m1sp, 0);
    digitalWrite(m1p1, LOW);
    digitalWrite(m1p2, LOW);
  }

  // Motor 2 control
  if (nSpeed > 0)
  {
    analogWrite(m2sp, nSpeed);
    digitalWrite(m2p1, HIGH);
    digitalWrite(m2p2, LOW);
  }
  else if (nSpeed < 0)
  {
    analogWrite(m2sp, -nSpeed);
    digitalWrite(m2p1, LOW);
    digitalWrite(m2p2, HIGH);
  }
  else
  {
    analogWrite(m2sp, 0);
    digitalWrite(m2p1, LOW);
    digitalWrite(m2p2, LOW);
  }
}

void stopMoving() {
  digitalWrite(m1p1, LOW);
```

```
  digitalWrite(m1p2, LOW);
  digitalWrite(m2p1, LOW);
  digitalWrite(m2p2, LOW);
  analogWrite(m1sp, 50);
  analogWrite(m2sp, 50);
}

void forwardSlow() {
  digitalWrite(m1p1, LOW);
  digitalWrite(m1p2, HIGH);
  digitalWrite(m2p1, LOW);
  digitalWrite(m2p2, HIGH);
  analogWrite(m1sp, 100);
  analogWrite(m2sp, 100);
}

void moveRight() {
  digitalWrite(m1p1, HIGH);
  digitalWrite(m1p2, LOW);
  digitalWrite(m2p1, LOW);
  digitalWrite(m2p2, HIGH);
  analogWrite(m1sp, 100);
  analogWrite(m2sp, 100);
}

void stopMotors()
{
  // Stop both motors
  controlMotorSpeed(0);
  delay(500);
   controlMotorSpeed(-100);
  delay(500);
  controlMotorSpeed(0);
}

void moveLeft() {
  digitalWrite(m1p1, LOW);
  digitalWrite(m1p2, HIGH);
  digitalWrite(m2p1, HIGH);
  digitalWrite(m2p2, LOW);
  analogWrite(m1sp, 100);
  analogWrite(m2sp, 100);
}
```