

PATH OR LINE FOLLOWER ROBOT

Chanda Akshay Kumar

2024102014

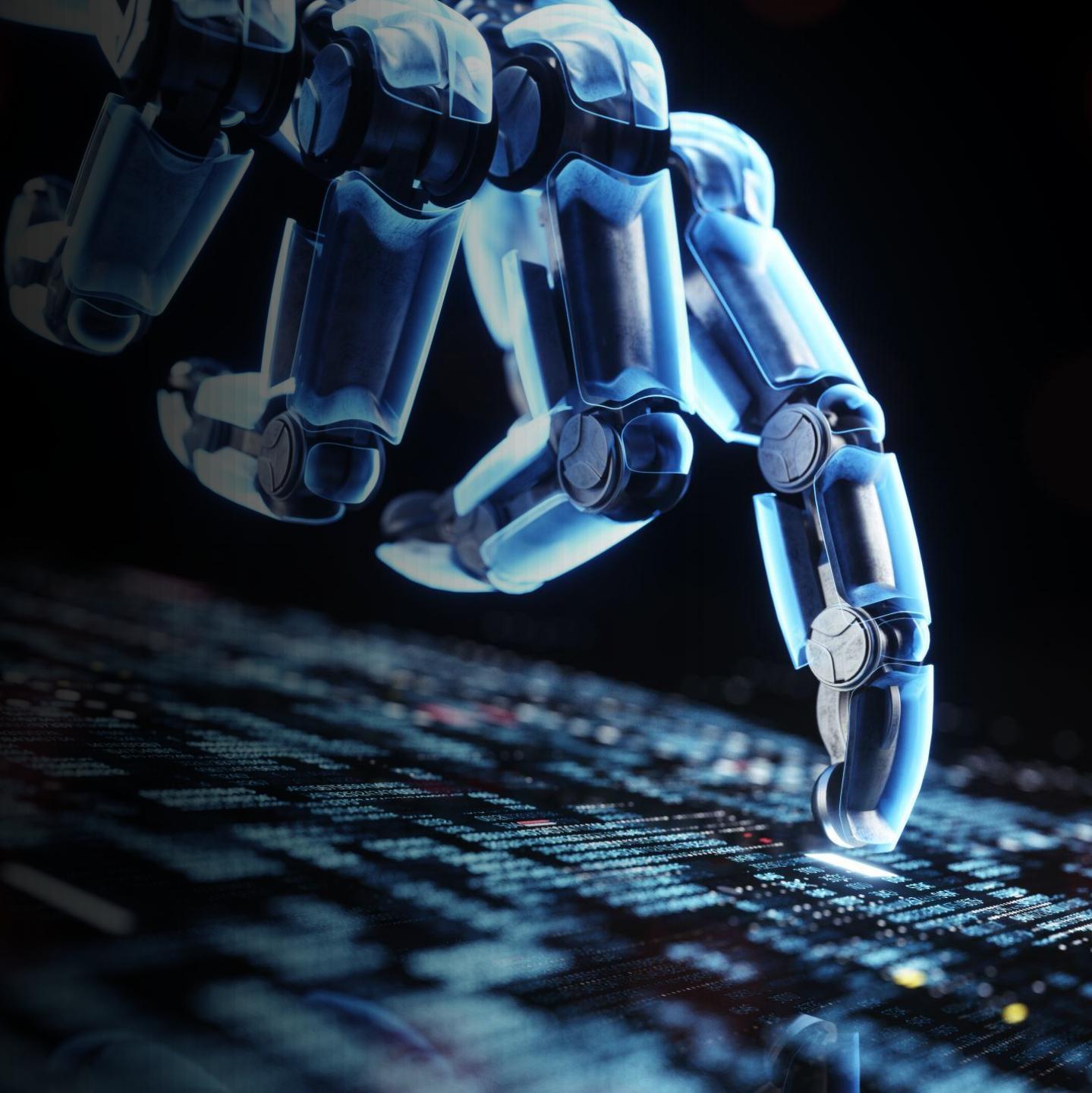
Samadhi Santhosh Kumar

2024102054

Problem Statement

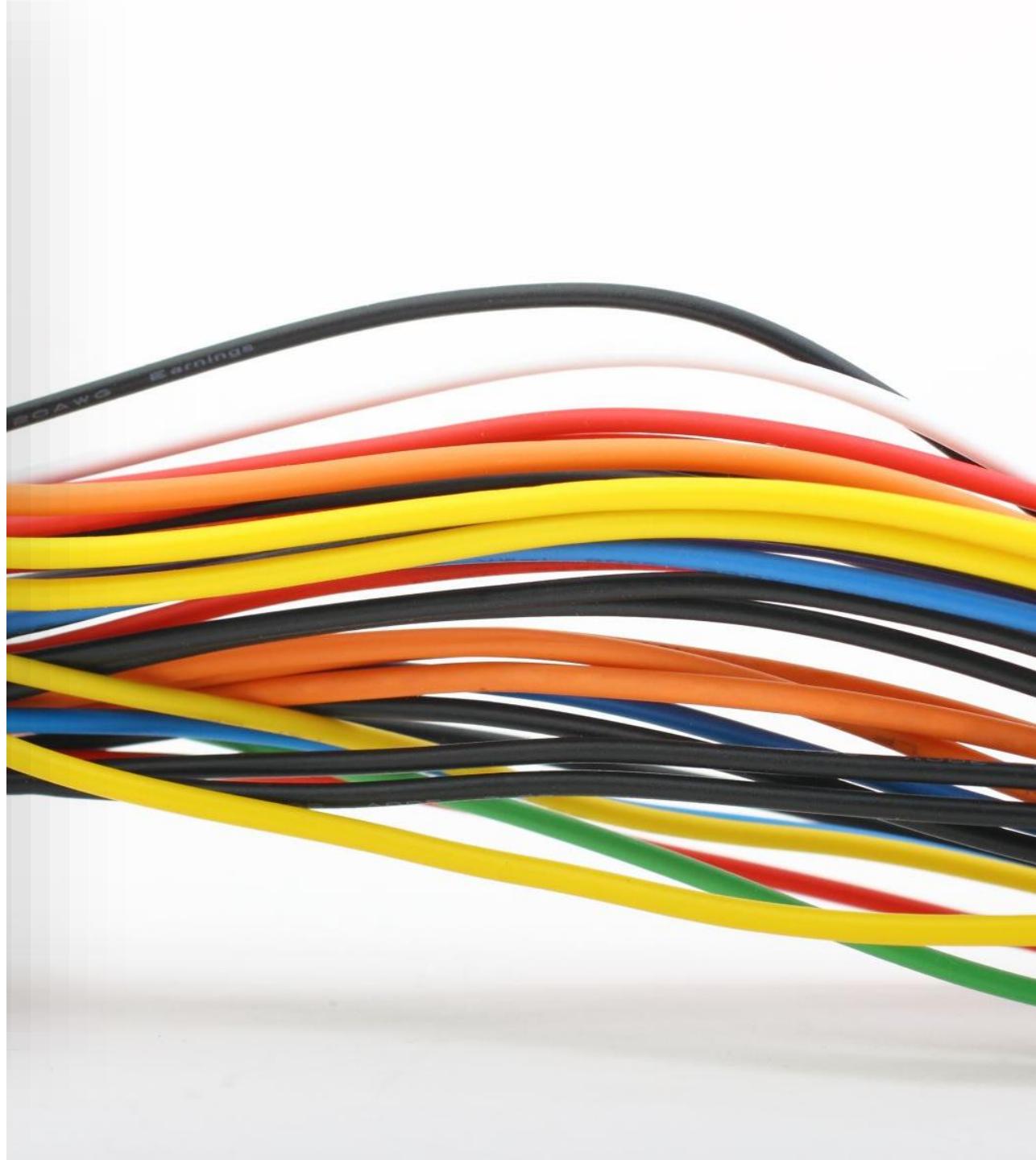
Explanation robot car

- In this project, we build a simple Line Follower Robot that can follow a line and avoid obstacles, demonstrating basic features of autonomous navigation. This adaptable robot can handle paths and obstacles.
- making it useful for tasks in warehouses or participating in robotics competitions.
- Autonomous vehicles like Tesla use sensors to navigate paths and avoid obstacles.



COMPONENTS REQUIRED

- **1)Acrylic chassis**
- **2)bread board and PCB**
- **3)IR sensors(2)**
- **4)Arduino UNO**
- **5>wheels(2)**
- **6)N20 motors (100 rpm)**
- **7)ball caster wheel**
- **8)jumper wires**
- **9)L293D(motor driver)**
- **10)ultrasonic sensor**
- **11)ultrasonic holder**
- **12)slide switch**



PROPOSED SOLUTION



Approach: The robot follows a black line on the ground using IR sensors placed on the left and right. These sensors help track the edges of the black line, ensuring accurate following. When an obstacle is detected, the ultrasonic sensor measures its proximity, and the robot makes necessary turns using a servo motor. The robot will then resume following the line once the obstacle is cleared.



Testing: The robot was tested upside down, which confirmed that the motors were functioning as expected. The code integrated with the L293D motor driver was verified by checking motor rotations and sensor readings. Although the black line test hasn't been performed yet due to lack of tape, the basic movement mechanics have been confirmed.

ELECTRONIC PROPOSED SOLUTION



Arduino: The core controller of the system, processing inputs from sensors and controlling motor movements.



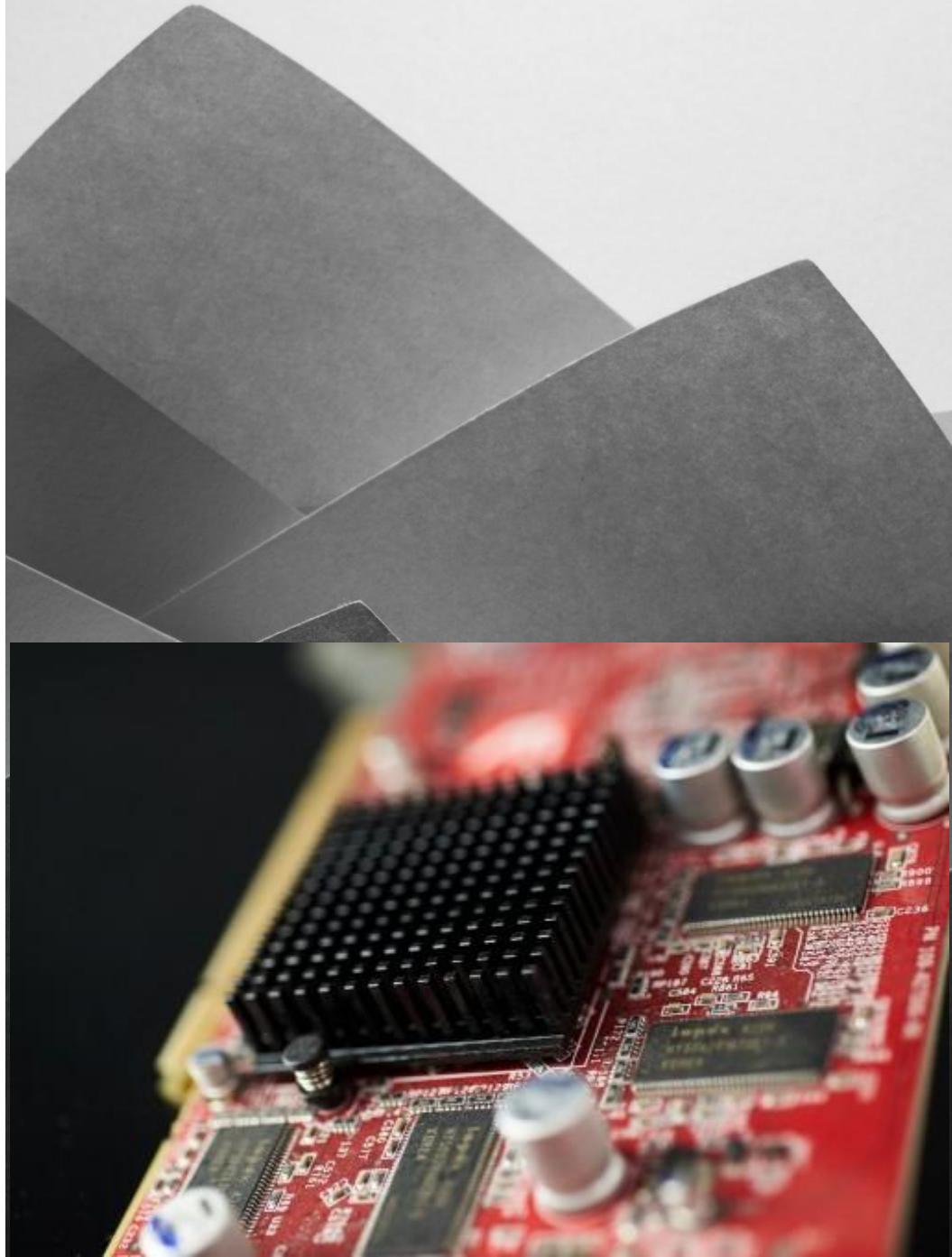
L293D/L298 Motor Driver: This motor driver controls the speed and direction of the robot's wheels, enabling both clockwise and counterclockwise rotation.



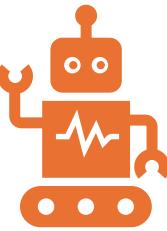
IR Sensors: Used to track the black line, ensuring accurate navigation.



Ultrasonic Sensor: Detects obstacles, providing distance information to help the robot avoid collisions.



PROPOSED SOLUTION



Servo Motor: Used to turn the robot when obstacles are detected.



Testing: The robot was tested upside down, which confirmed that the motors were functioning as expected. The code integrated with the L293D motor driver was verified by checking motor rotations and sensor readings. Although the black line test hasn't been performed yet due to lack of tape, the basic movement mechanics have been confirmed.

- **WORK DONE**

- The Line Follower Robot has been built, and the motors are moving as expected. However, the robot has not yet been tested on an actual black line due to a lack of tape.
- Our path follower robot detects the white path which was surrounded by the Black path and tries to move on the black tape

- Testing Results: The robot successfully moved when checked upside down, confirming that the motor control and movement functions are operational.
- Acknowledge that the line-following robot has been built and the motors are moving as expected, though the black line test hasn't been conducted yet due to a lack of tape.
- Testing Results: Mention that the robot works when tested upside down, confirming that motor control and movement are functional.

- **FUTURE WORKS**

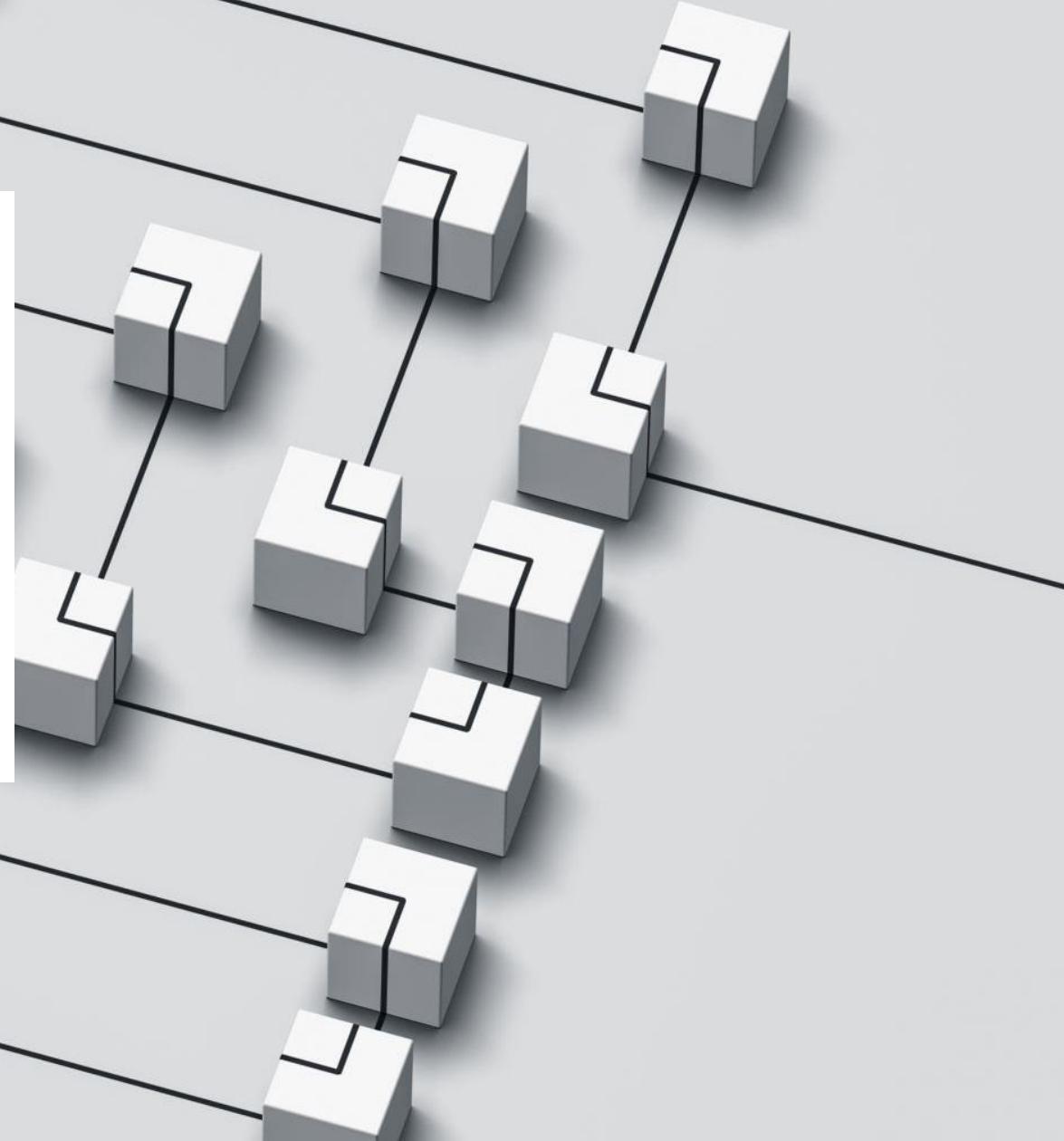
- Plans for adding obstacle detection, code optimization for smoother turns, and handling more complex navigation with the L298 motor driver and optimized power sources (e.g., batteries)

- **MOTIVATIONS**

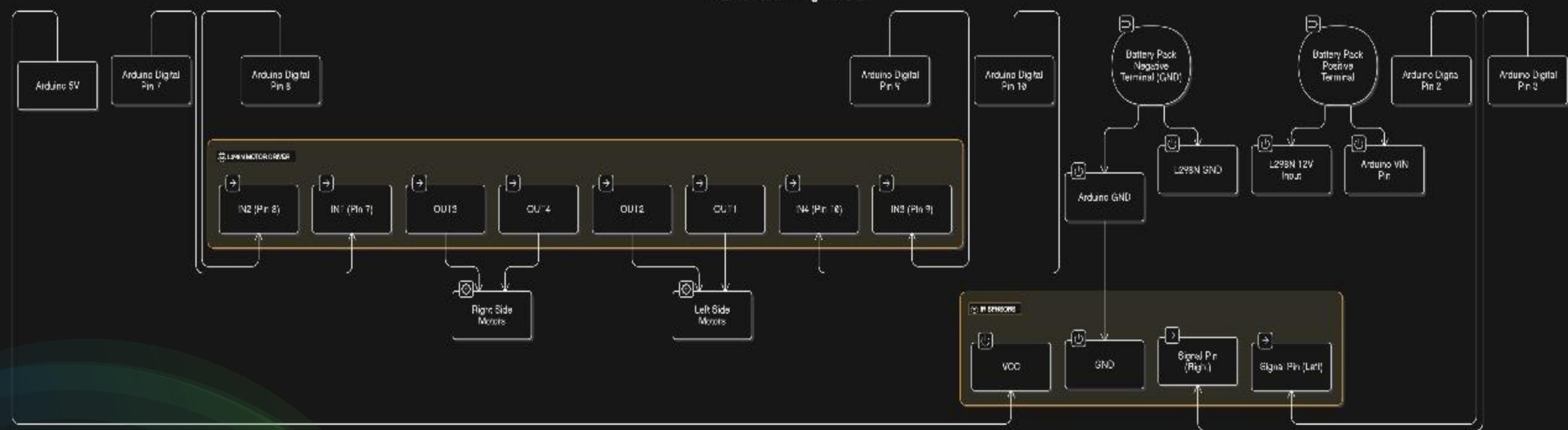
- The thought process behind this project started with creating a simple line-following algorithm, then adding obstacle detection to make the robot more adaptable. We used an Arduino for its ease of programming and flexibility. The L298 motor driver controls motor movement, while IR sensors detect the black line for path-following. An ultrasonic sensor detects obstacles, allowing the robot to adjust its direction accordingly. Our aim is to create a reliable and efficient robot capable of autonomous navigation with minimal human intervention.



CIRCUIT DIAGRAM OF THE PATH FOLLOWER ROBOT



Arduino Robot Wiring Flowchart



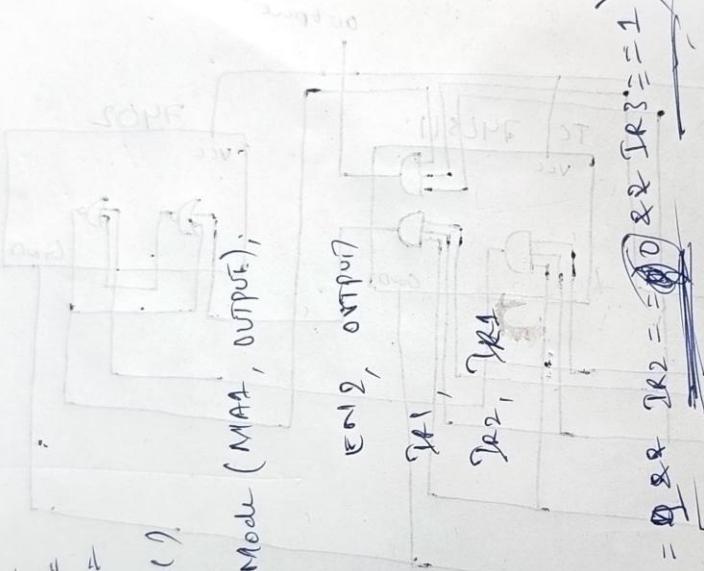
```
#include <avr/delay.h>
#include <avr/io.h>

#define MIA1 = 1
#define MIA2 = 2
#define MIB1 = 3
#define MIB2 = 4
#define EN1 = 5
#define EN2 = 6
#define IR1 = 7
#define IR2 = 8
#define IR3 = 9

void setup()
{
    pinMode(MIAA, OUTPUT);
}

void loop()

```



```
if (IR1 == 0 && IR2 == 0 && IR3 == 1)
{
    forward();
}
else
{
    reverse();
}

void forward()
{
    PORTB = 0x00;
    PORTC = 0x00;
    PORTD = 0x00;
    DDRB = 0x00;
    DDRC = 0x00;
    DDRD = 0x00;
    PORTB |= 0x01;
    PORTC |= 0x02;
    PORTD |= 0x04;
    DDRB |= 0x01;
    DDRC |= 0x02;
    DDRD |= 0x04;
    delay();
}

void reverse()
{
    PORTB = 0x00;
    PORTC = 0x00;
    PORTD = 0x00;
    DDRB = 0x00;
    DDRC = 0x00;
    DDRD = 0x00;
    PORTB |= 0x04;
    PORTC |= 0x02;
    PORTD |= 0x01;
    DDRB |= 0x04;
    DDRC |= 0x02;
    DDRD |= 0x01;
    delay();
}
```

```
{ digitalWrite(EN1, 255);
  (EN2, 255);
  (MIA1, HIGH);
}
```

Thank You