Development of an Arduino-Based Line Following Robot

RECIPIENT
ENGR. KEVIN NACINO
Computer Engineering

GROUP 5

ALVARADO, IAN JOSEPH

VISPERAS, JHON MICHAEL

SANTOS, MARC ROMAN

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

INTRODUCTION

A line following robot is an autonomous vehicle engineered to detect and follow a predefined path or line marked on the floor, typically using high contrast colors such as black on white. These robots are popular in robotics education and competitions due to their straightforward yet educational design. Utilizing an Arduino as the control system simplifies the design process, making it accessible for beginners and educational environments, while still offering ample scope for enhancement and complexity in more advanced applications.

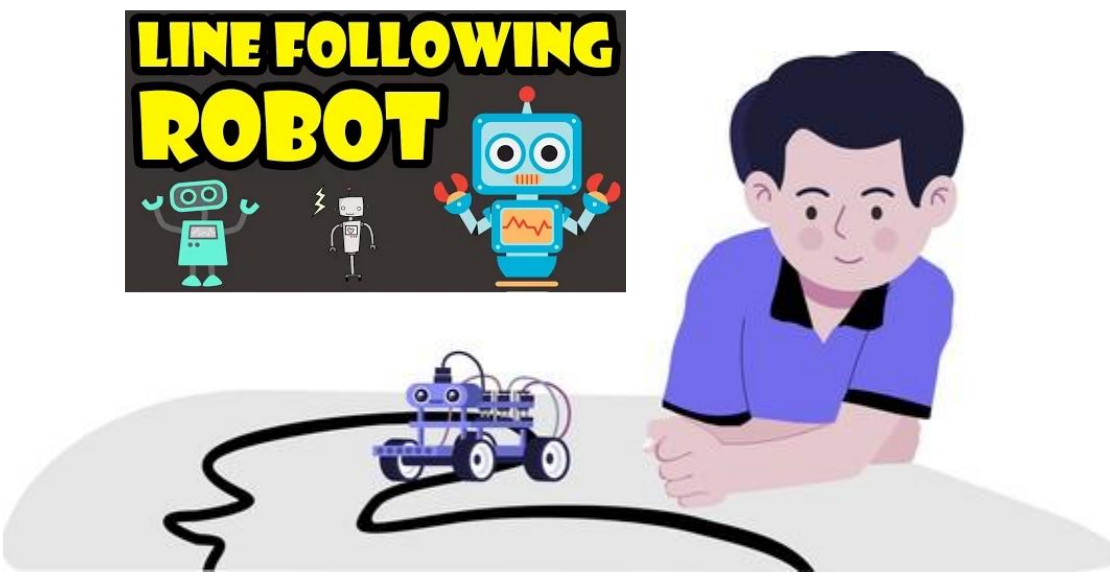


Figure 1: INTRODUCTION

ABSTRACT

This paper details the development of a line following robot controlled by an Arduino microcontroller, designed to autonomously navigate by following a black line on a white surface. The robot uses 5 IR sensors to detect the line and employs a proportional-derivative (PD) control strategy to adjust motor commands, maintaining its course. Construction integrates an Arduino Uno, sensors, DC Motors, and a motor driver. Tests demonstrate the robot's capability to adapt to varying path curvatures and environmental contions. This project serves as our project in CPE 047 - Microprocessor.

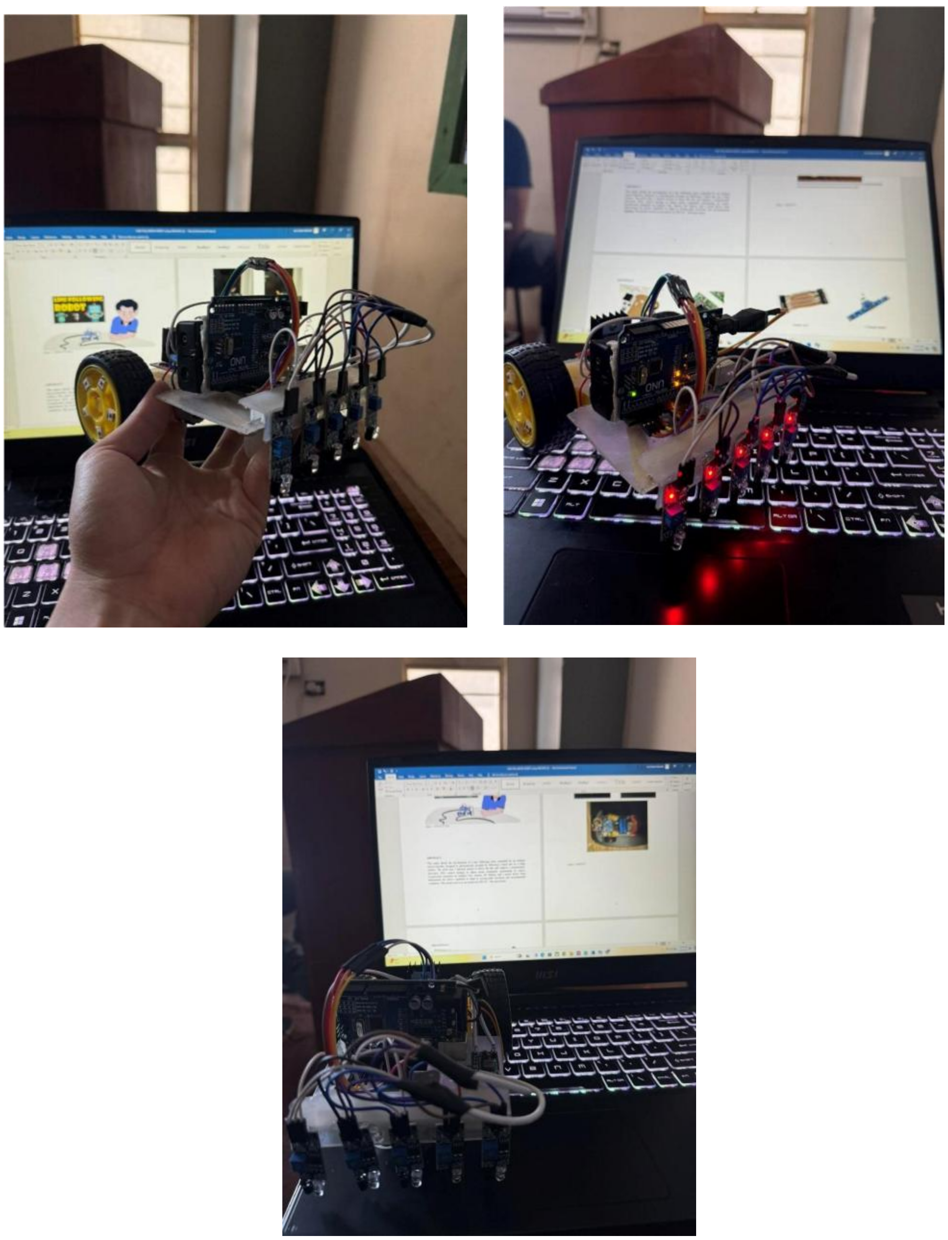
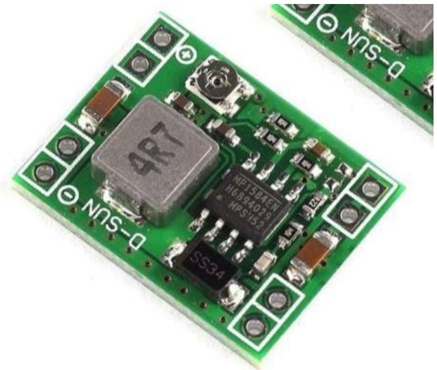


Figure 2: ABSTRACT

MATERIALS

2WD Car Chassis Kit

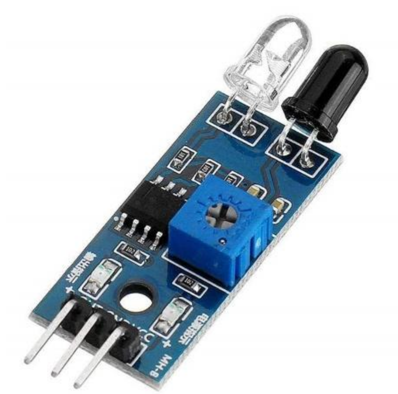


dc buck converter 4r7

18650 battery

Arduino UNO Shield

Jumper wire



5 Channel sensor

USB Cable Arduino Uno R3

Arduino UNO

Solid wire

Figure 3: MATERIALS

Table 1: SPECIFICATIONS & DEFINITIONS

COMPONENTS	SPECIFICATIONS	DEFINITIONS
2WD Car Chassis Kit		A mobile robot chassis is the physical framework or body of a mobile robot that provides it with mobility. It typically includes the wheels or tracks, motors, sensors, and other components necessary for movement.
dc buck converter 4r7	4.5V-28V to 0.8V-20V Adjustable	Steps down voltage (while stepping up current) from its input (supply) to its output (load).
18650 battery	3.7V (aprox. 5.10V) To 14.8V (aprox. 20.4V)	To power the bot, Lithium-Ion batteries (like 18650) along with a boost converter (to convert 3.7V to 5V) are preferred since a simple 9V battery can't run the bot.
IR sensor	3.3V to 5V supply input	For line detection logic, we used two IR Sensors, which consists of IR LED and Photodiode.
USB Cable Arduino Uno R3		The USB cable used for the Arduino Uno R3 is a standard USB 2.0 A to B cable, typically used for connecting printers and other peripherals to computers. This cable provides both power and data transfer capabilities, allowing for programming and serial communication between the Arduino and the computer.
Arduino UNO		The Arduino Uno is a versatile microcontroller board based on the ATmega328P, featuring 14 digital I/O pins and 6 analog inputs. It is programmable via the Arduino IDE, widely used for educational projects, DIY

		electronics, and prototypes due to its ease of use and robust community support.
Arduino UNO Shield	L293D currents of up to 600-mA at voltages from 4.5 V to 36 V	The L293D is a motor driver integrated circuit that can control up to two DC or stepper motors, supporting motors with operating voltages from 4.5 V to 36 V and currents up to 600 mA per channel. It features two H-bridge configurations, enabling bidirectional control of motors and includes built-in diodes for back-EMF protection.
Solid wire	Black and Red	It is ideal for outdoor use where more durability and higher currents are required. The red wire carries the positive voltage, and the black is circuit ground.
Jumper wire	Male to male	To interconnect the components of a breadboard or other prototype or test circuit internally or with other equipment or components without soldering.
	Female to male	The male and female jumper wires are used to connect batteries to an electrical outlet or to connect batteries to each other for charging.

PROCEDURES

Design and Components Selection:

- **Microcontroller:** Select Arduino Uno for its simplicity and extensive support.
- **Sensors:** Choose IR sensors to detect the line on the surface.
- **Motors and Driver:** Opt for L298D driver shield for motor control.
- **Power Supply:** Use batteries suitable for the Arduino and motors.
- **Chassis:** Design or select a chassis that can hold all components including the Arduino, sensors, motors, and batteries.

Assembly:

- **Mounting the Arduino:** Secure the Arduino Uno on the chassis.
- **Attaching Sensors:** Install the IR sensors at the front of the chassis, ensuring they are positioned close enough to the ground to detect the line.
- **Motor Setup:** Attach the motors to the chassis and connect them to the L293D motor driver.
- **Wiring:** Connect all components (sensors, motors, motor driver) to the Arduino according to the circuit diagram.

Wiring and Connections:

- **Power Connections:** Ensure the power supply is correctly connected to both Arduino and the motor driver.
- **Sensor Connections:** Connect the IR sensors to the analog pins on the Arduino if they are analog; use digital pins if they are digital.
- **Arduino UNO Shield:** Connecting the IC's 5V and ground pins to the Arduino's 5V and ground pins respectively.

Programming:

- **Code Basics:** Write or modify existing Arduino code to read sensor inputs, process them, and control motor outputs based on sensor data.
- **Algorithm Implementation:** Implement a simple control algorithm, focusing primarily on the proportional part for line following.
- **Testing and Calibration:** Upload the code to Arduino, test the robot, and adjust sensor positions or code parameters as needed.

Testing and Troubleshooting:

- **Initial Test:** Test the robot on a track with a clearly marked line under different light conditions.
- **Troubleshoot Issues:** Identify and fix issues related to sensor sensitivity, motor speed, or programming errors.
- **Fine-tuning:** Adjust the PID parameters and sensor placement until the robot follows the line smoothly and reliably.

Documentation and Reporting:

- **Record Findings:** Document the design choices, implementation details, and test results.
- **Challenges and Solutions:** Discuss any challenges encountered and how they were resolved.
- **Future Improvements:** Propose future enhancements like adding more sensors for better accuracy or integrating wireless control.reless control.

Setting up an Arduino-based line following robot, from the initial design and assembly to programming and final testing, ensuring a functional and effective robotic system.

SCHEMATIC DIAGRIAM

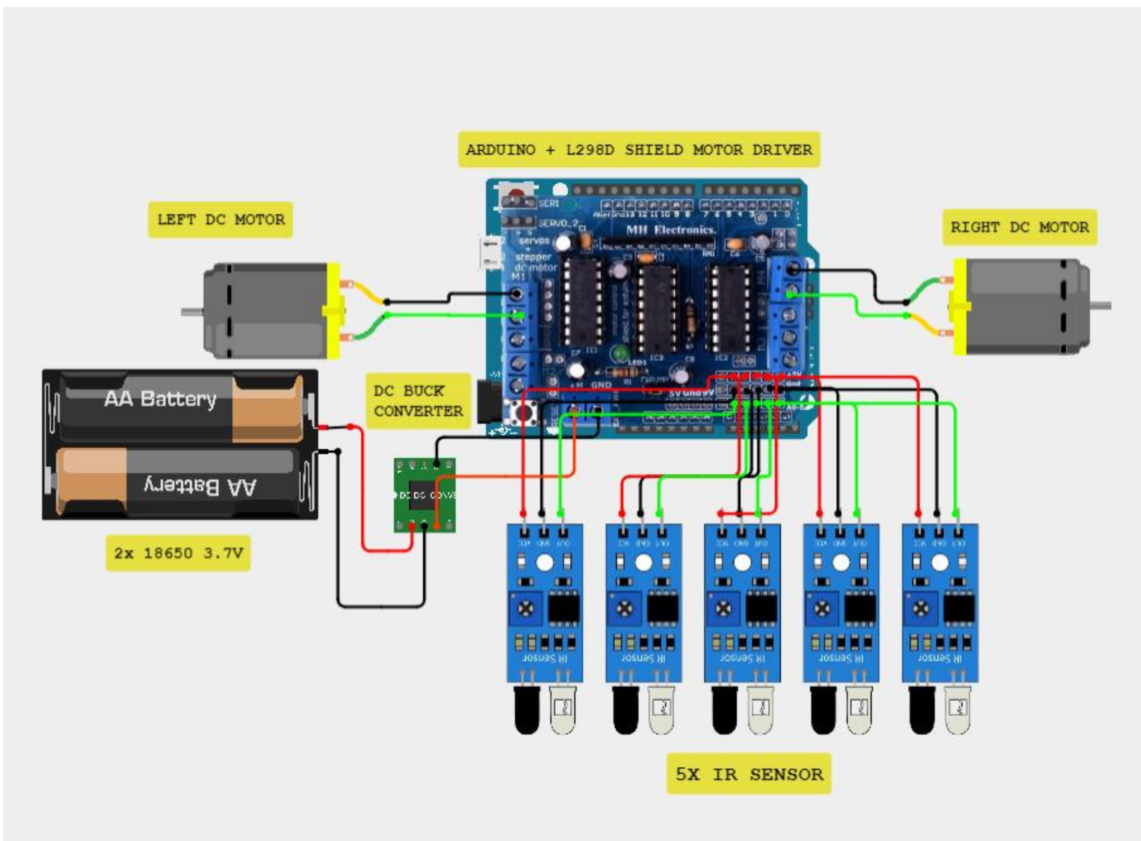
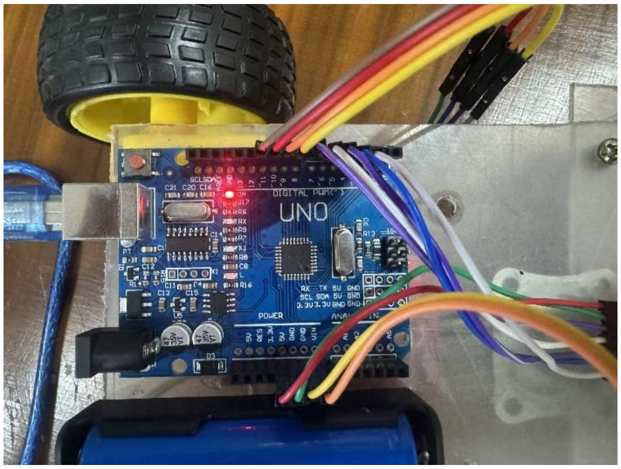


Figure 4: SCHEMATIC DIAGRAM

Arduino Image



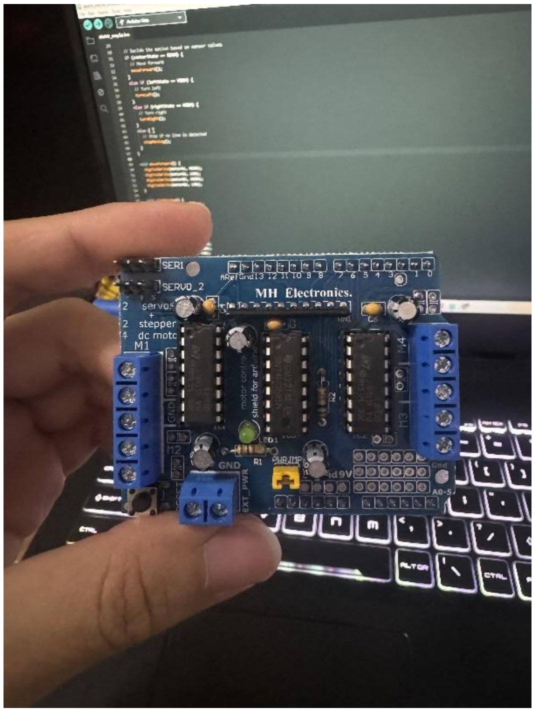
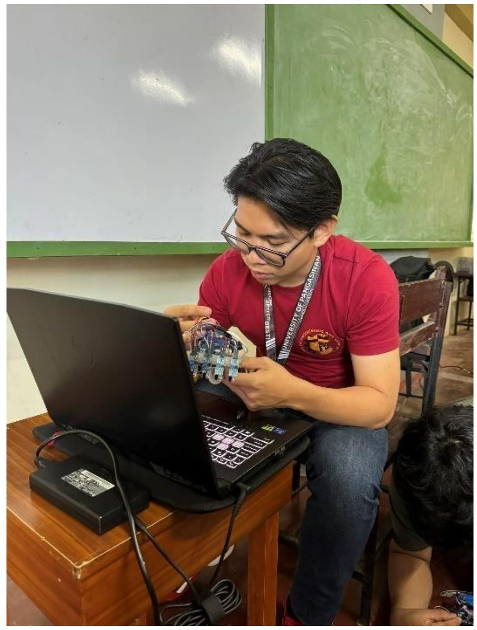


Figure 5: ARDUINO UNO AND ARDUINO UNO SHIELD/ L293D ACTUAL IMAGE

After we activated and combined the components on the breadboard, calibrating and placing the components on the protoboard and built a 3 wheels smart robot car on the chassis.



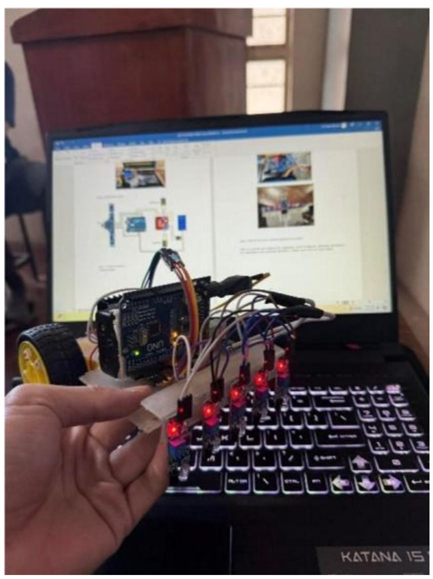


Figure 6: CALIBRATING ARDUINO ROBOT

Connecting components and checkings codes is part of the preparation processing to avoid error codes and making connections every components.

Figure 7: CONNECTING COMPONENTS AND CHECKING CODES

After connectings components you will be able to connect there wirings and soldering to tap all those components, and next after checking codes is the next step is to test and run input codes into output arduino robot.

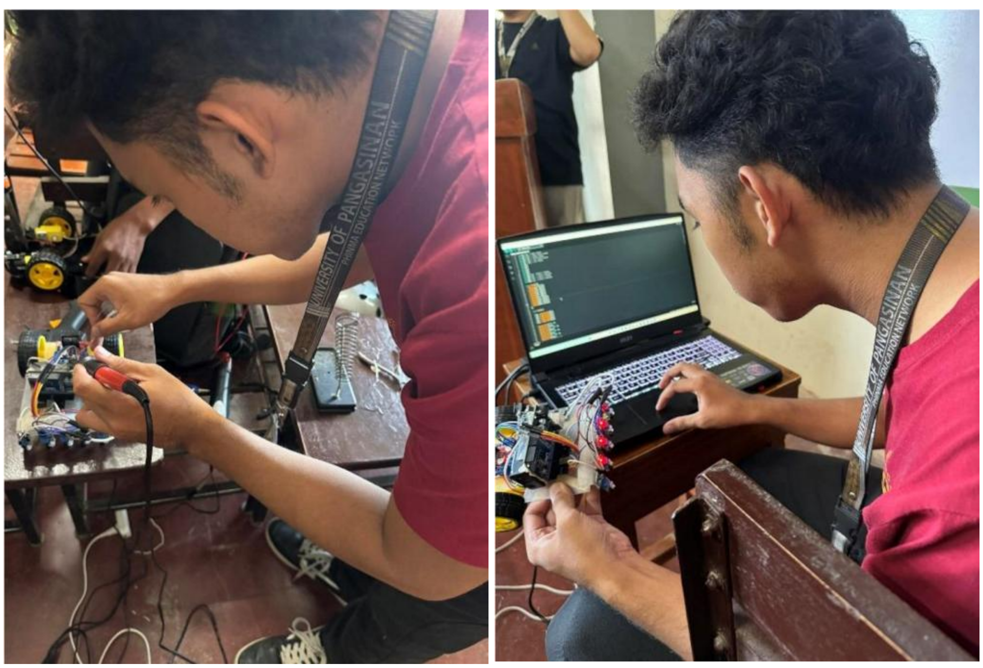


Figure 8: SOLDERING WIRES AND RUN TEST THE CODE

Then in the final output we tried to operate it on the temporary line we made.

Figure 9: FOLLOWING LINE TESTING

SPECIFICATION

Table 2: 2WD SMART ROBOT CAR CHASSIS KIT

ITEM	QUANTITY
2WD Car Chassis Kit	
DC Geared Motor	3~6V
Motor Gear Ratio	1:48
Wheel Diameter	66mm
Wheel Width	26mm
Size	approx. 20 x 14cm(L x W)
Weight	280grams
EMC testing geared motors	2x
Rubber Wheel	2x
Speed encoder disc	2x
Caster wheel	1x
Two battery holder	1x
M3x30 Long screw	4x
M3x8 brass standoff spacer	4x
M3x8 Philips Screws	10x
M3 Hex Nut	10x
Motor fastener T-plate	4x
On/Off switch	1x
15mm connecting cable	4x
Installation manual	1x

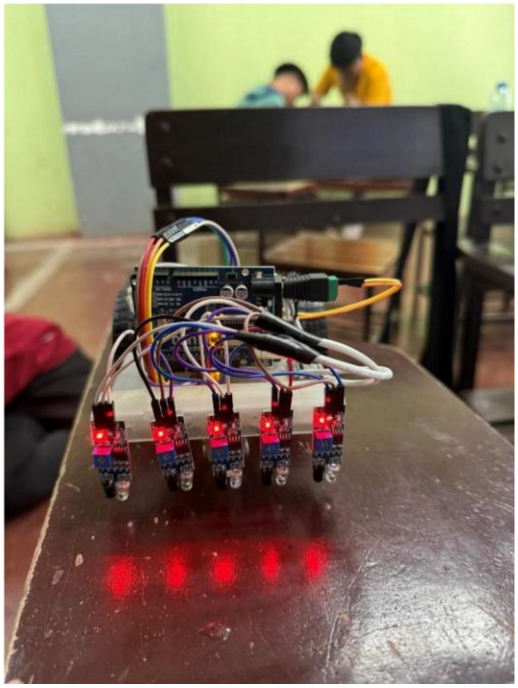
DESCRIPTIONS & UNITS

Table 3: DESCRIPTON & UNITS OF COMPONENTS

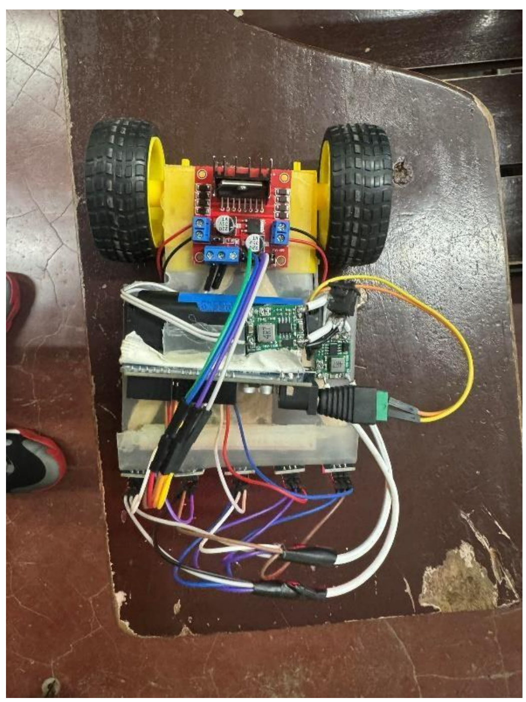
ITEMS	DESCRIPTION/ UNITS	Quantity
dc buck converter 4r7	4.5V-28V to 0.8V-20V Adjustable	1
18650 battery	3.7V (aprox. 5.10V)	1
	6.4V (aprox. 10.20V)	2
DC Motor L298N	5V and 35V, with a peak current up to 2A	1
5 channel sensor	3.3V to 5V supply input	1
Arduino UNO		1
R3 USB Cable		1
Solid wire	Black and Red	2
Jumper wire	Male to male	
	Female to male	

ACTUAL PROTOTYPE

FRONT



BACK



TOP

Figure 10: ACTUAL PROTOTYPE

Core Components

Arduino Microcontroller: The brain of the robot, typically an Arduino Uno, which processes inputs from line sensors to determine the robot's position relative to the line and drives the motors accordingly.

Sensors: These sensors detect the reflection of 5 channel sensor, which varies between different surface colors (e.g., black absorbs light, while white reflects it).

Motors and Wheels: Motors, often DC types, are controlled by the Arduino to adjust the robot's speed and direction. The wheel and motor setup can vary, but typically two driven wheels with one or more free-rotating wheels (casters) for balance are used.

Motor Driver: An intermediary device like an H-Bridge or a motor driver shield, compatible with the Arduino, to control the motor's direction and speed based on the signals from the Arduino.

Power Supply: Batteries provide power to the Arduino and the motors. It's important that the power supply is sufficient to handle the load and duration of operation without excessive weight.

Chassis: The physical structure that holds all components together. This can be custom-made from materials like acrylic, or sourced from kits.

How It Works

Setup and Calibration

- Line Detection: The 5 Channel sensors continuously scan the surface beneath them. By calibrating the sensors, the robot can distinguish between the contrasting colors on the path.
- Sensor Placement: The sensors are typically placed at the front of the robot, spaced such that they can detect the width of the line and any deviations.

Programming

- Programming the Arduino: The Arduino is programmed using the Arduino IDE, where the code is written to process the input from the line sensors. The logic implemented allows the robot to make decisions about movements (forward, left, right) based on the line position.
- Control Algorithm: Simple robots might use basic threshold-based algorithms, where the motor's speed is adjusted if one sensor detects the line while the other does not. More complex algorithms, can be used for smoother and more responsive control.

Operation

- The robot starts following the line upon activation. It adjusts its path by altering the wheel speeds: slowing down the wheel on the side where the sensor detects the line to make turns or to realign itself.

Applications

Line following robots are not just educational tools but also serve practical applications in industries such as automated warehouses, where they can follow routes marked on the floor to move items from one place to another efficiently.

Conclusion

Building a line following robot using an Arduino is an excellent project for learning about robotics, sensor integration, motor control, and programming. It serves as a foundational project that encapsulates fundamental robotics principles and can be a stepping stone to more complex robotic systems and applications. C++ or Arduino IDE used by the programmer to create codes and execute.

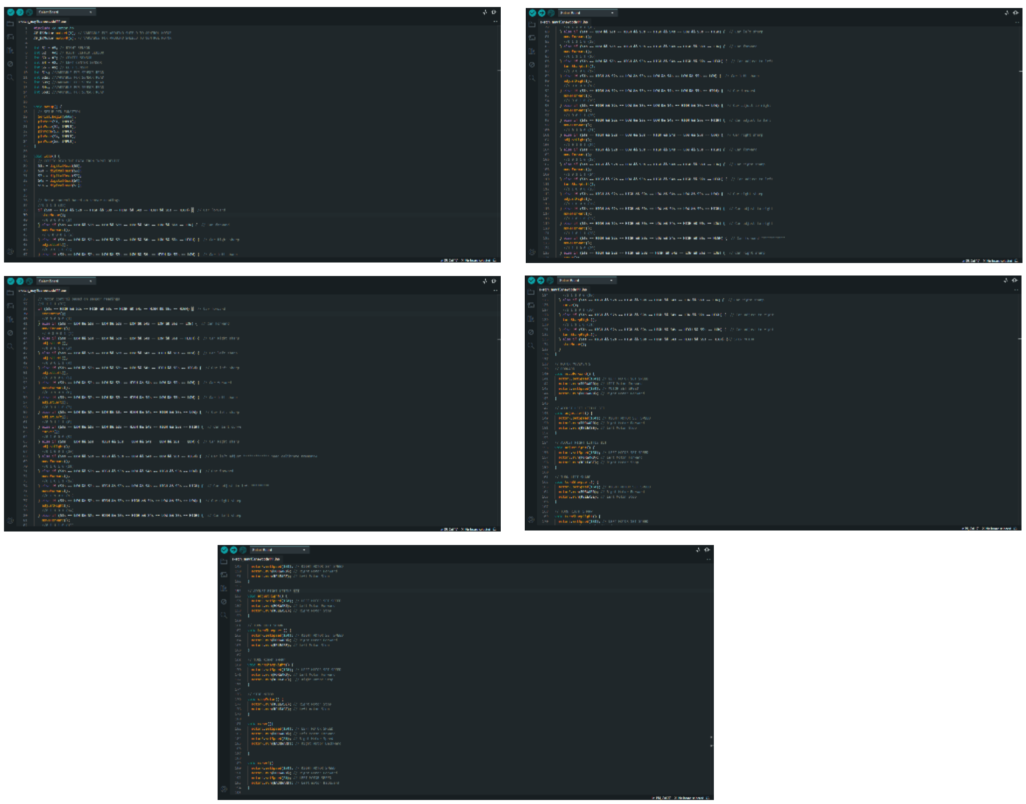


Figure 11: USED CODES BY THE PROGRAMMER

L	LC	CS	RC	R
0	0	0	0	0
0	0	0	0	1
0	0	0	1	0
0	0	0	1	1
0	0	1	0	0
0	0	1	0	1
0	0	1	1	0
0	0	1	1	1
0	1	0	0	0
0	1	0	0	1
0	1	0	1	0
0	1	0	1	1
0	1	1	0	0
0	1	1	0	1
0	1	1	1	0
0	1	1	1	1
1	0	0	0	0
1	0	0	0	1
1	0	0	1	0
1	0	0	1	1
1	0	1	0	0
1	0	1	0	1
1	0	1	1	0
1	0	1	1	1
1	1	0	0	0
1	1	0	0	1
1	1	0	1	0
1	1	0	1	1
1	1	1	0	0
1	1	1	0	1
1	1	1	1	0
1	1	1	1	1

Figure 12: TRUTH TABLE