TECHFEST, IIT Bombay

Meshmerize

Documentation by
IIT Kanpur

Date of Submission:

November 14, 2021

Contents

1 Team Members

Name	**Roll No.**
Aman Agarwal	200097
Arushi Garg	200193
Navneet Singh	200625
Suhani	201011

2 Introduction

2.1 Aim

The goal of this project is to build a highly optimized line following robot, in which a dry run and actual run algorithms will be provided to achieve the task within the stipulated time.

2.1.1 Problem Statement

Teams have to build an autonomous robot that can follow a white line and keep track of directions while going through the maze. The bot has to analyze the path in the Dry Run and use this information in the Actual Run to go through the maze from the starting point to the ending point in the minimum possible time.

3 Brief Approach

The bot was first designed with basic components - motors, wheels, batteries, casters etc. Then the LFS sensors were attached to it, along with a motor driver and the Arduino MEGA. Then an arena was designed and printed on a flex sheet. After this, we started working on the implementation of the line following algorithms in a module-based manner.

4 Components

This section gives a detailed description of all the components used in this project.

4.1 Mechanical Parts

This section describes all the mechanical components used.

4.1.1 Chassis

An acrylic sheet of thickness 5mm was used to make the chassis of the robot. The width and length of the sheet were decided according to the constraints specified in the problem statement.

Figure 1: Acrylic Sheet

Features:

- Can be folded easily by applying heat
- Lightweight and rigid
- Inexpensive

4.1.2 Wheels

For the wheels of the robot we used the 3PI miniQ Car wheel Tyre N20 DC Gear Motor Wheel. These are customized high-quality rubber wheels of diameter 42mm.

Figure 2: Wheels

Features:

- Coder Accuracy: 12 pulses per revolution
- Heavy Duty Material
- Durable and specially designed

Loading Capacity (Kg)	3
Weight (gm)	43
Wheel Diameter(mm)	42
Color	Black (Tyre) White (Rim)
Wheel Width	18
Body Material	Plastic
Grip Material	Rubber
Center Shaft Hole Diameter	3 mm D-type

Table 1: Wheel Specifications

4.1.3 Caster

Heavy duty metal steel ball encased in plastic casing. The caster allows smooth omni-directional movements.

Features:

- Three equally spaced mounting holes to fix it tightly with chassis
- Easy to apply lubricant
- Lightweight
- Inexpensive

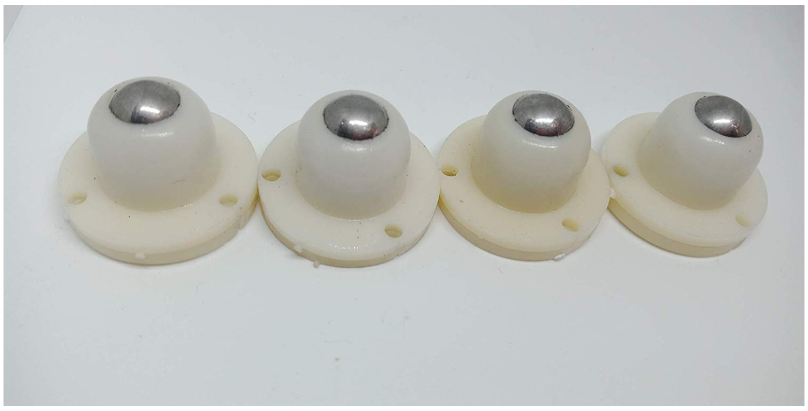


Figure 3: Caster Wheel

Wheel Total Height	23 mm
Mounting Hub Diameter	33.5 mm
Mounting Hole Diameter	4 mm
Mounting Hole PCD	27.5 mm

Table 2: Caster Specifications

4.1.4 Mounting Bracket

The bracket is used to clamp the motors onto the chassis.

Features:

- Durable
- Lightweight
- Perfectly fits to micro motors
- Inexpensive

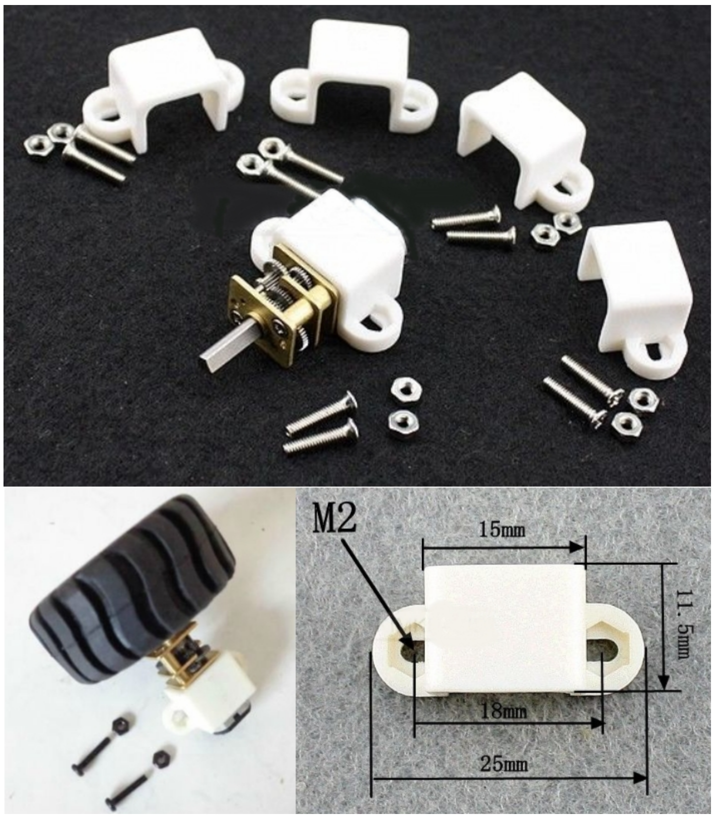


Figure 4: Mounting Bracket

Height	11 mm
Width	25 mm
Material	High Quality Plastic

Table 3: Bracket Specifications

4.2 Electronic Components

This section describes all the electronic components used.

4.2.1 Batteries

For powering the robot, we have used Lithium Polymer batteries. These batteries, also known as Li-Po batteries are widely used in GPS, DVD, i-pod, Tablet PC, MP4 Player, Power Bank, Mobile Backup Power Supply, Bluetooth Speaker, and Industrial applications.

`Features:`

- Lightweight
- Rechargeable
- Small size
- Easy and safe to use

Figure 5: Batteries

Voltage	3.7V
Capacity	1000 mAh
Dimensions	50mm X 30mm X 4mm

Table 4: Battery Specifications

4.2.2 Controller

Arduino Mega 2560 follows Open-source principles, which means you can freely download the development environment and many associated resources. Ideal for project work and often used in schools, teaching, motor control, robotics and similar applications; the Arduino platform allows for rapid application development using either Java (cross-platform) or Processing (a C derivative used to write sketches using Arduino IDE downloadable from Arduino.cc website.)

Input voltage should be limited to 6   12V, but if the voltage supplied is less than 6V, I / O port may not be supplied to a voltage of 5V, and therefore readings may become unstable. If the voltage is greater than 12V, the regulator device may overheat causing damage to the control board. It is therefore recommended that you supply between 6.5 and 12V, utilizing typical power supplies of 7.5V or 9V.

Mega 2560 is fully compatible with the Arduino software and has complete hardware compatibility with the Mega 2560 Arduino Shield high-performance processor and enough memory to handle even complex solutions.

Features:

- ATmega2560 Microcontroller. Boot-loader to allow downloading of ‘sketches’ via USB without having to go through other external writers.
- Powered from USB or via external PSU (not supplied). The device will automatically switch between power inputs.
- Heavy gold plate construction.

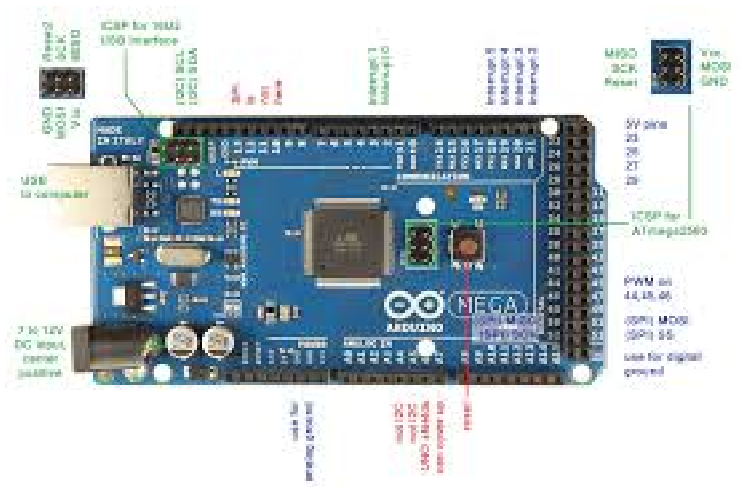


Figure 6: Arduino mega labelled diagram

Microcontroller	ATmega2560
EEPROM	4KB
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8KB
PWM Output Pins	15
Analog I/O Pins	16
Digital I/O Pins	54 digital input / output terminals (14 of which have programmable PWM outputs)
DC Current per I/O Pin	DC Current per I/O Pin: 40 mA ; DC Current for 3.3V Pin: 50 mA
Architecture	Advanced RISC Architecture
Clock Speed	16 MHz
Input Voltage(Recommended)	7-12V
Operating Voltage	5
Weight (gm)	35g
Color	Blue
Dimensions (mm) LxWxH	110 x 53 x 15

Table 5: Arduino MEGA specifications

4.2.3 Sensor

The Line Follower Sensor (LFS) gives the robot the ability to sense the lines. It consists of reflectance sensors to differentiate between white and black or dark and light lines by detecting the reflected light coming from its own infrared LED. A single board has an array of six sensors precisely designed for smooth line following. It provides analog outputs and can be interfaced with any microcontroller.

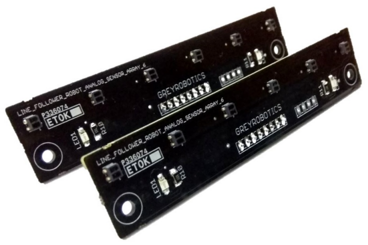


Figure 7: LFS Sensor

Features:

- Uses Reflectance Sensors for high precision
- Easy to use analog output pins
- TTL (Transistor Transistor Logic) compatible outputs
- Two power LED indicators
- Six sensors in an array
- Black and white line following
- 3mm diameter hole for easy mounting
- Compatible with Arduino, AVR, PIC, ARM, etc

Operating Voltage	5V
Current Requirement	210 mA
Output Voltage Range	0-5V
Recommended Sensing Distance	5 mm
Maximum Sensing Distance	12 mm
Operating Wavelength	920 mm
Distance between two reflectance sensors	18 mm
Dimensions	100 mm X 20 mm
Weight	less than 10 grams

Table 6: Sensor Specifications

4.2.4 Motors

The Micro Metal Geared encoder motor comes with a built-in encoder which measures the motor's speed in real time. The average output number of pulses can reach up to 420 pulses per revolution. The motor has a long (0.354"/9.0 mm) D profile metal output shaft. The brass faceplate has two mounting holes threaded for M1.6 screws (1.6mm diameter, 0.35mm thread pitch). The hall sensor has 7 pole pairs, so the encoder resolution will be increased by 7-fold.

Figure 8: Motor with Encoder

Rated Voltage	6V
Motor Speed	15000 rpm
Gear Reduction Ratio	30:1
Reducer Length	9.0 mm
No-Load Speed	530 rpm at 6V
No-Load Current	60 mA
Rated Torque	0.2 kgcm
Rated Speed	300 rpm at 6V
Current Rating	170 mA
Instant Torque	0.45 kgcm
Hall Feedback Resolution	420
Weight	18g

Table 7: Motor Specifications

4.2.5 Motor Driver

The L298N Motor Driver is a high power motor driver perfect for driving DC Motors and stepper motors. It uses the popular L298 motor driver IC and has an onboard 5V regulator which it can supply to an external circuit. It can control up to 2 DC motors with directional and speed control. This motor driver is perfect for controlling motors using microcontrollers, switches, relays, etc.

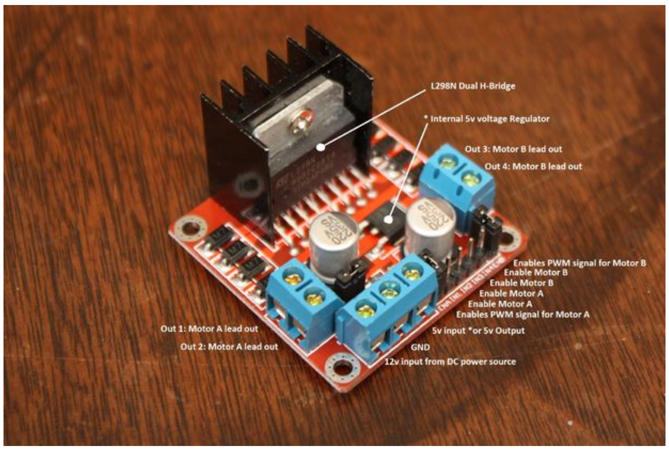


Figure 9: L298N Motor Driver

`Features:`

- Maximum motor supply current: 2A per motor.
- Current Sense for each motor.
- Heat sink for better performance.
- Power-On LED indicator.
- Double H bridge Drive Chip: L298N.

5 Design

This section covers all the design aspects of the project.

5.1 Bot

Here we describe the process followed in designing the bot.

5.1.1 Ideation

The starting phase of the project involved reading various research papers and looking at already built line follower robots to get design ideas. After watching numerous Youtube videos on line follower bots and comparing their performances, we concluded that the design of the bot should be as simple as possible. We also concluded that the bot should have at least the following features:

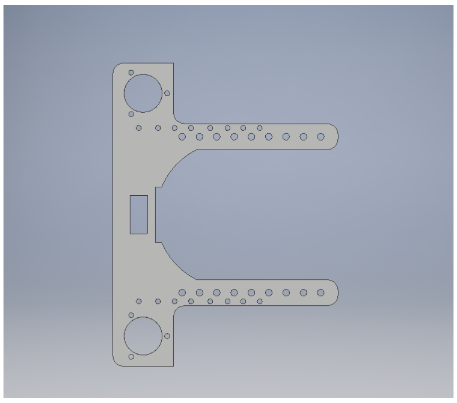
- **Ground Clearance:** We observed that the robots which had very low ground clearance always seemed to perform better. This was due to the fact that a low ground clearance implied a low of centre of gravity, which increased the stability of the bot and allowed it to accurately take sharp turns at high speeds. However the clearance would also have to be greater than a minimum threshold so that the sensor could sense the line properly.
- **Cylindrical Wheels:** We observed that robots that had small wheels that were not wide enough tended to skid a lot, which would take it off track at a turn and then the

controller would be unable to find the line again. Hence we decided to use cylindrical roller-like wheels so that there would be less slipping with greater area of contact.

- **Multiple Sensors:** Most robots were implemented using only one simple array of IR sensors. We observed that the robots in such a case were unable to distinguish between a '+' junction and a 'T' junction.Three additional sensors were placed in front of the bot in order to be able to determine the type of intersection. The output of these sensors is taken only at the time of intersections and does not interfere with simple line following algorithm. Hence we decided to use 2 LFS sensors, one at the front of the bot which would be used for distinguishing purpose, and another LFS at the back which would be used for basic line following purpose.

5.1.2 Implementation

The two parts of the chassis were first designed using AutoCAD Inventor LT 2021. Slots were made in the frame according to the measured sizes of each component. Additionally, some holes were made in order to have adjustable placement of both the sensors. The front part was made curved in order to decrease the total weight of the robot. The back part of the chassis is rectangular since it would carry most of the components. The robot was then first assembled in AutoCAD after importing CAD diagrams of the major components from the internet. Once all the discrepancies had been removed from the assembly, the top faces of the chassis parts were exported into .dxf files. These files were then used to laser cut the acrylic sheet to specifications.


(a) Front Part

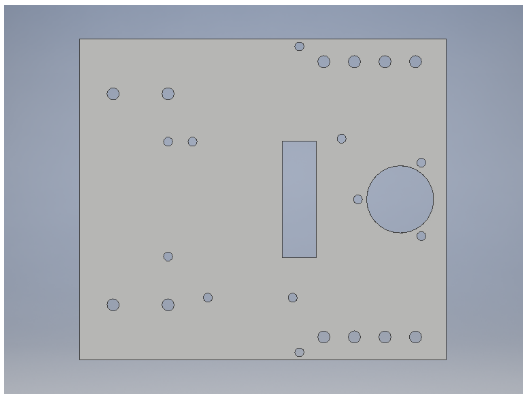

(b) Back Part

Figure 10: Chassis Frame

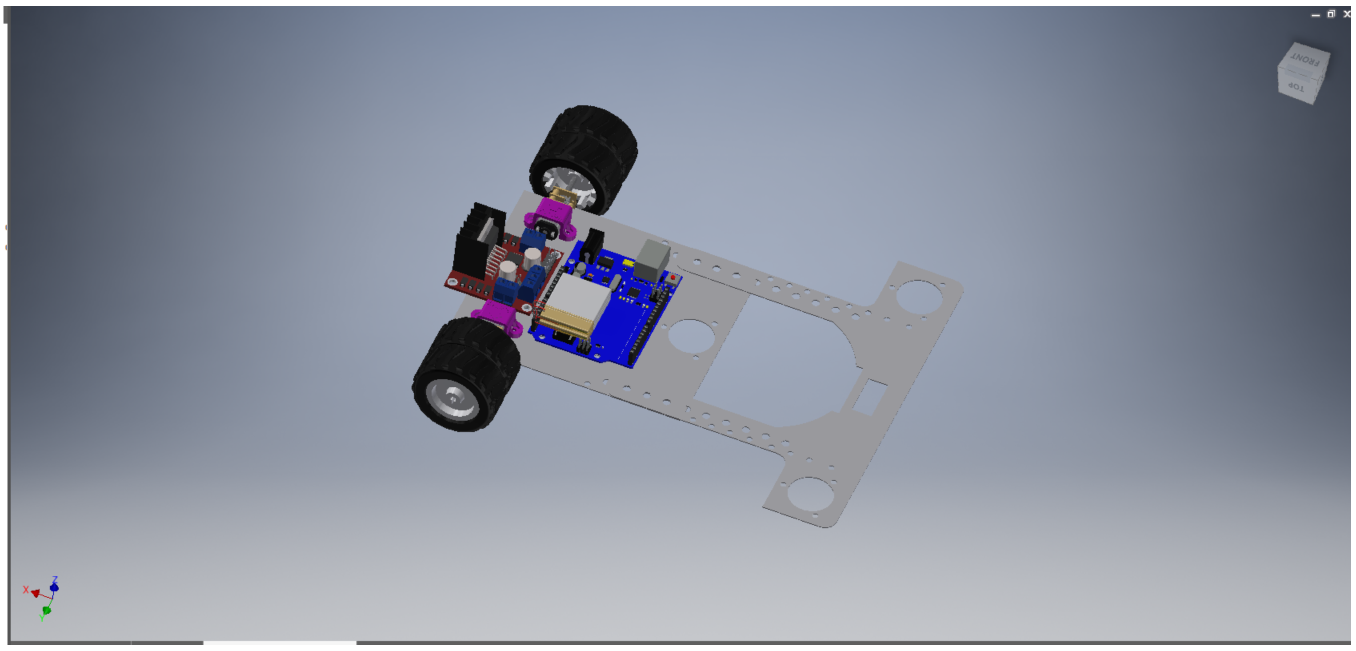


Figure 11: Assembled CAD with Electronics

5.2 Arena

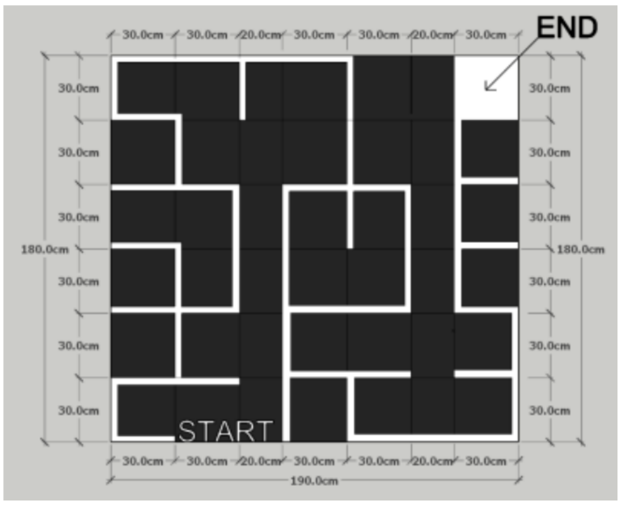


Figure 12: Sample Arena provided for the competition

5.3 Algorithms

This section describes all the algorithms designed for the functioning of the robot.

5.3.1 Sensor Input

The LFS sensor has an array of 6 IR sensors, each of which output an integer in the range 0-1000 which is proportional to the ratio of reflected and transmitted light. Hence values towards the higher end correspond to black, while values on the lower end belong to white. A threshold value was hence set to determine whether to classify the sensor reading as black or white. This threshold value is affected by the height of the sensor, external lighting conditions and the surface's reflectivity.

5.3.2 Line Position

A 2D coordinate system is set up to pinpoint the position of the line with respect to the central axis of the end. The positive axis of this system points towards the right when the bot is viewed from above, the sensors are read in this direction. 10cm of the actual path corresponds to 1 unit in the matrix.

5.3.3 Dry run Algorithm

`Depth First Search`

In order to map the intersections, loops and dead-ends of the arena, we decided to apply a simple depth-first-search algorithm giving right as first priority, followed by straight and left respectively. Whenever the bot detects an intersection, it turns right. If it detects no line in front of it, it turns anticlockwise until it has found the line again. These two actions constitute the right-hand algorithm and allow the bot to map the arena while staying on course to the finish.

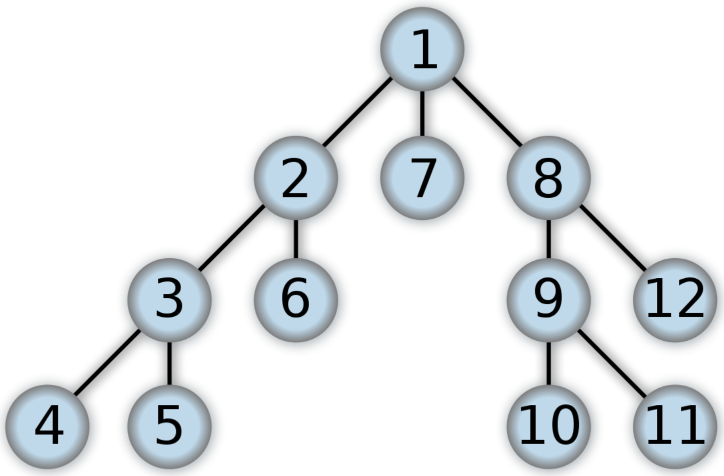


Figure 13: Depth first search

5.3.4 Actual Run Algorithm

`Dijkstra's Algorithm`

Using the path traced by the bot in the dry run, the bot will now find the shortest path using the Dijkstra's algorithm and try to reach the end in minimum time. We generate a SPT (shortest path tree) with a given source as a root. We maintain two sets, one set contains vertices included in the shortest-path tree, other set includes vertices not yet included in the shortest-path tree. At every step of the algorithm, we find a vertex that is in the other set (set of not yet included) and has a minimum distance from the source.

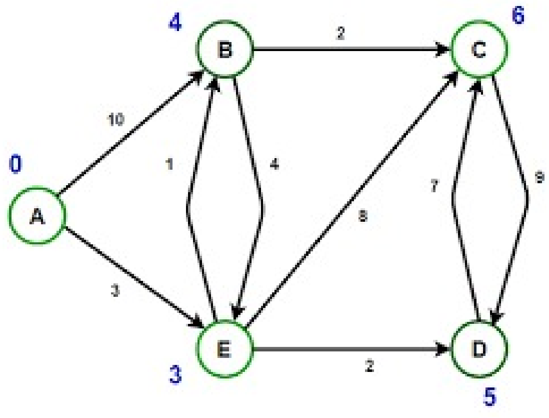


Figure 14: Dijkstra's algorithm

6 Assembly

This section describes the assembly of the robot.

6.1 Bot

Once the chassis parts had been manufactured, they were joined together using nuts and screws. Then the casters, motors, mounting brackets and wheels were screwed onto the frame. The sensors, motor driver and Arduino MEGA were also fixed according to the designated slots on the frame.

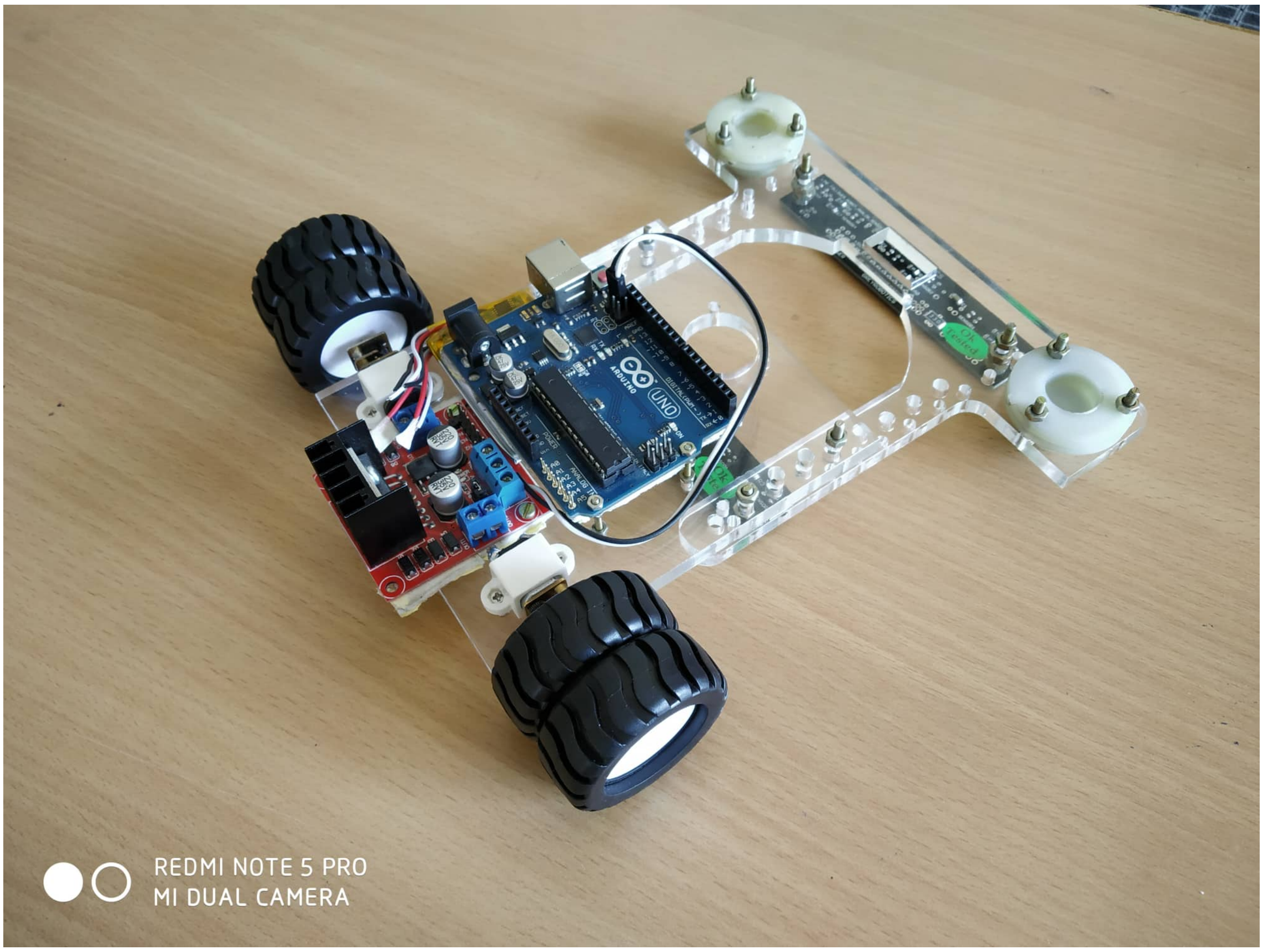


Figure 15: Assembled bot

6.2 Electronics

This section covers the assembly of all the electronic components, detailing the pin connections of each individual component.

6.2.1 Motor Driver

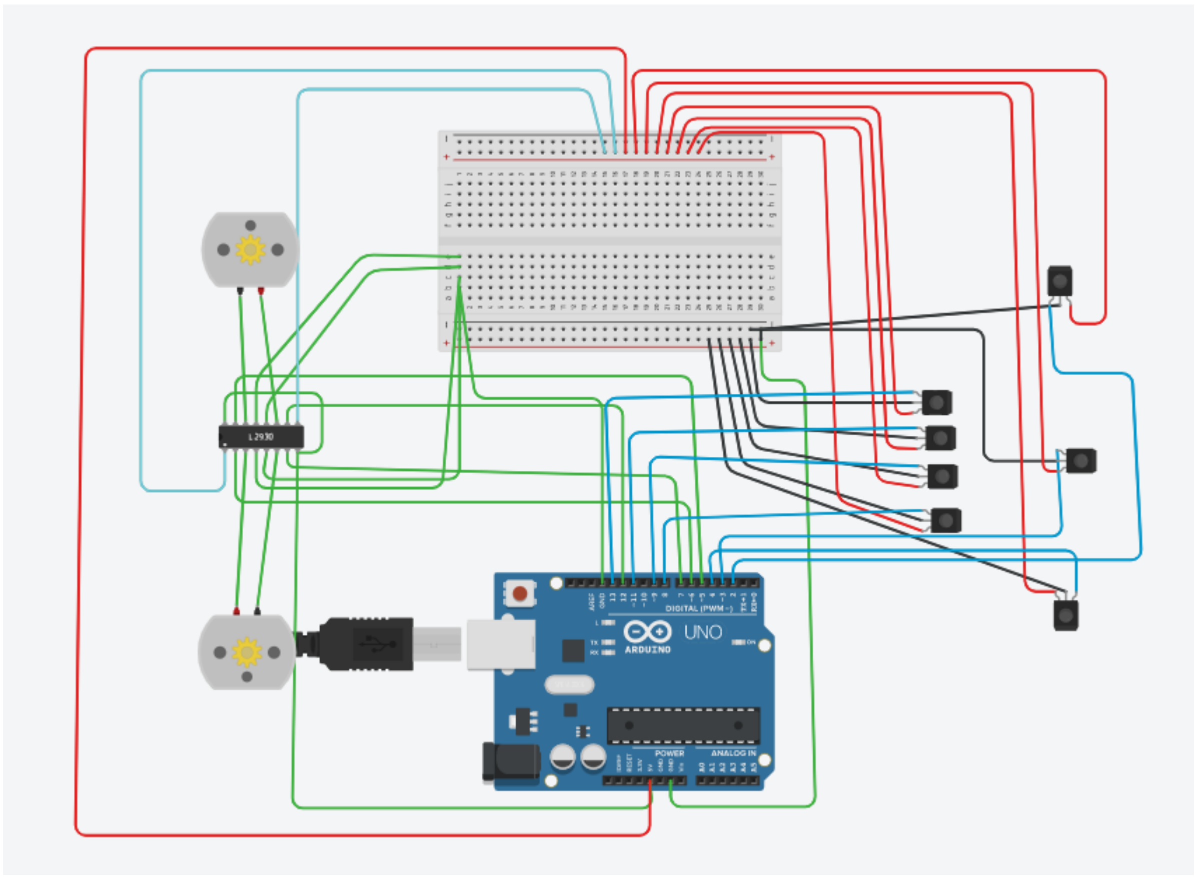


Figure 16: Connection Diagram

`NOTE:` The actual bot will use Arduino MEGA but the above circuit shows Arduino UNO due to unavailability of Arduino MEGA component in TinkerCAD.

`Connections:`

- Out 1: Motor A lead out
- Out 2: Motor A lead out
- Out 3: Motor B lead out
- Out 4: Motor B lead out
- GND: Ground
- 5v: 5v Output
- 12v: 7-35V Input

- EnA: Enables PWM signal for Motor A
- In1: Input Motor A
- In2: Input Motor A
- In3: Input Motor B
- In4: Input Motor B
- EnB: Enables PWM signal for Motor B

`Important Points:`

- All the grounds (Arduino, Power source, and the Motor Controller) must be tied together.
- Since PWM features are being used, connect the motor driver pins to corresponding PWM-enabled pins on the controller.

`Arduino Sketch Considerations:`

Since there isn't a library for the L298N Dual H-Bridge Motor Controller, only declaring which pins the controller is hooked to is sufficient. The motor driver thus becomes very versatile in this project as a lot of the Arduino's pins are being used.
PWM pins that can be used on the MEGA are 2 to 13 and 44 to 46. These provide 8-bit PWM output with the analogWrite() function.

6.2.2 LFS Sensors

`Connections:`

- V_{cc} - Supply voltage
- GND - Common circuit ground
- A2,A3,A4,A8,A9,A11,A13 - Analog output pins

6.2.3 Encoder Motor

`Connections:`

- V_{cc} - Supply voltage to encoder with Arduino's 5v pin
- c1 - encoder digital output1 pin with interrupt pin
- c2 - encoder digital output2 pin with interrupt pin
- m1 - motor input pin with motor driver's output pin1
- m2 - motor input pin with motor driver's output pin2

6.2.4 Battery

Two 3.7V batteries are connected in series in order to get desired voltage (7.4 V).

`Connections:`

- GND : with motor driver's ground
- Positive : with motor driver's 12v pin

`Connections:`

- VCC: Connect to +5V of Arduino
- GND: Connect to Ground of Arduino
- TXD: With digital pin 10 (defined as RXD)
- RXD: With digital pin 11 (defined as TXD)

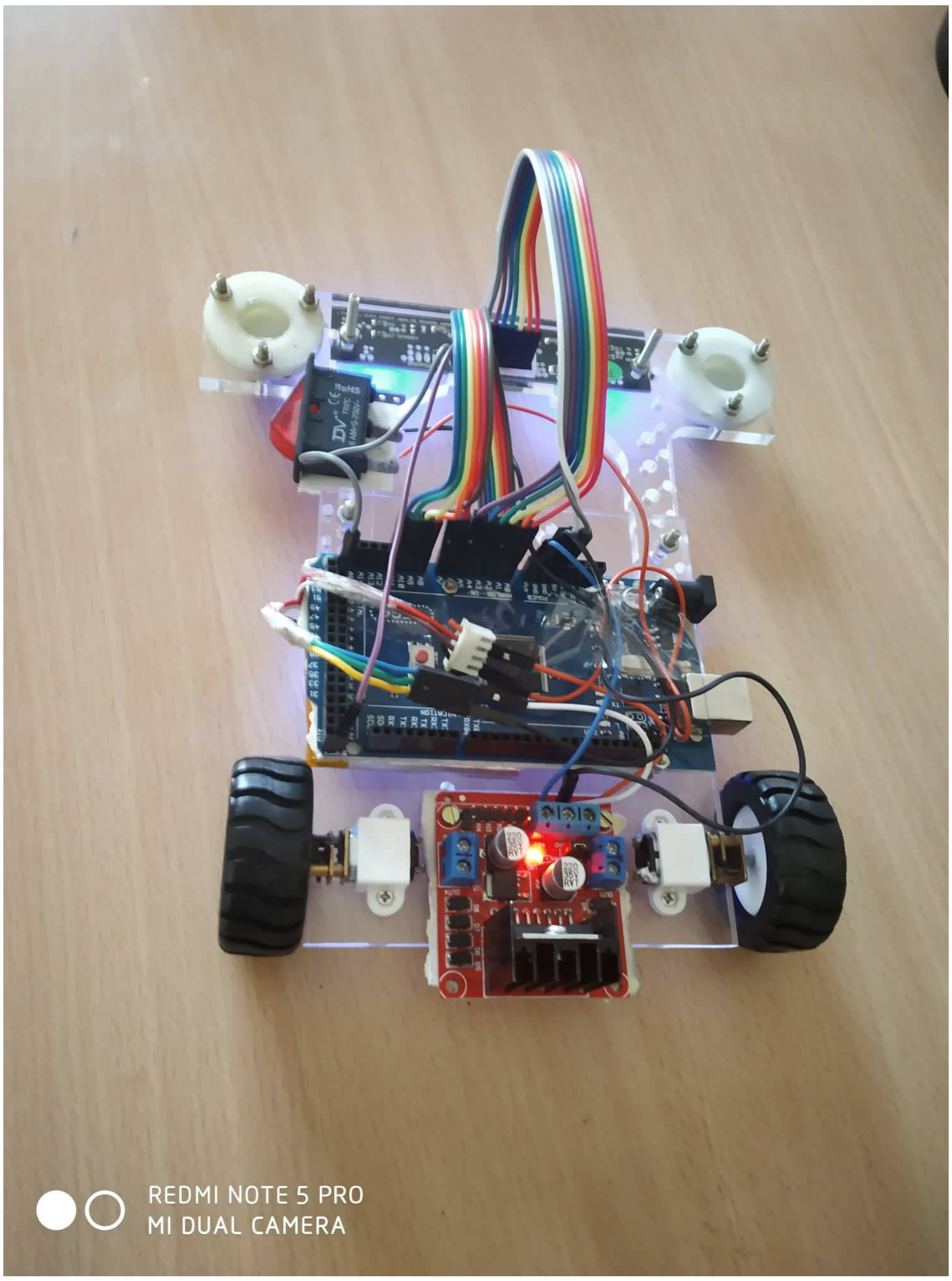


Figure 17: Assembled bot with electronics

7 Implementation

7.1 Testing Pipeline

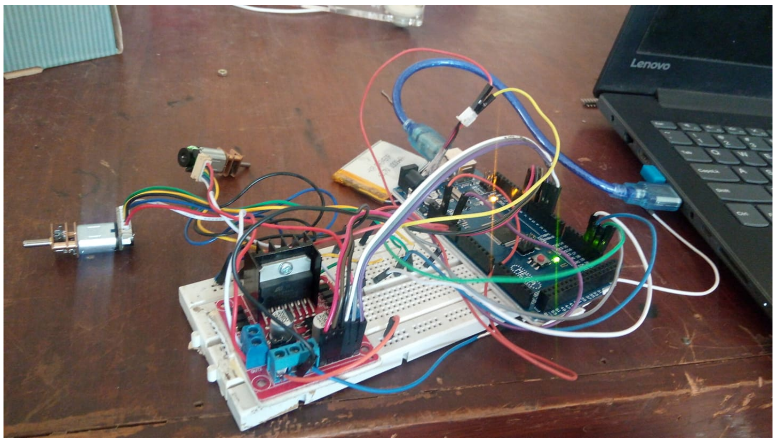


Figure 18: Motor Testing

Before implementing all the algorithms on the robot, we individually tested each module using a breadboard and only the required electronic components. An Arduino sketch implementing a specific module/algorithm was first written and compiled. The necessary connections were then made using the breadboard, the Arduino MEGA and other components. The compiled sketch was then uploaded onto the Arduino and its output was observed. If the output was not as desired, changes were made to the sketch and it was tested again using the same procedure. Each algorithm module is individually described below: