Basic Line Follower Robot

Learning Outcomes:

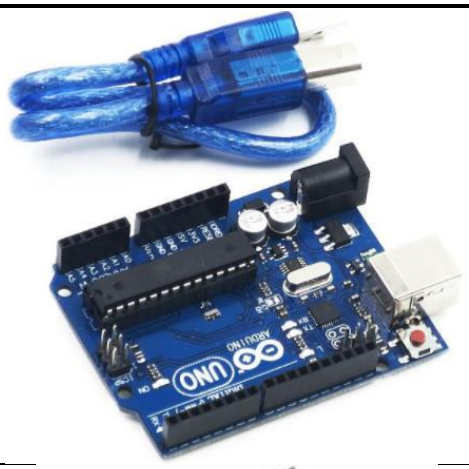
At the end of this module, the students will be able to:

- Design and build an Arduino-based line following robot hardware
- Write the software controller using Arduino platform
- Implement a robot that follows a black track against a bright background

Robot becomes widely used in industrial due to their characteristics. Robot able to work in 24 hours continuously without feeling. Line follower robot is a useful robot that is used in ware houses, industries, and stores, where it follows a dedicated path.

In this guide, you will create a simple line following robot that will follow specified path by giving appropriate signal to the microcontroller. Two DC motors are used interfaced to the microcontroller through a motor driver IC (L298N), and one caster wheel (free-directional). Input signals given to the microcontroller from the sensors and then the controller takes the appropriate action according to the program written in it and drives motors as desired.

Materials Needed:

Devices
1. Arduino Uno 
2. H-Bridge L298N Motor Driver 

3. Robot Car Chassis (2 wheels, 1 caster wheel)

4. Battery

5. Jumper Wires

6. Side cutter

Figure 1. Line Follower Materials

Tools needed in building the robot:

Tools
1. Screw driver (Philips) 
2. Glue Gun 

3. Soldering Gun & Lead (if necessary)

4. Wire Cutter/Stripper (if necessary)

5. Multimeter

Figure 2. Tools in Building the Line Follower

Principle of Line Follower

This kit is a 2-wheel drive with one caster wheel. This kit comes with an assembly manual. Just follow the instruction on how to do it. After assembly, the chassis should look like the pictures below.

Working of Arduino Line Follower Robot

Source: https://www.electronicshub.org/arduino-line-follower-robot/

The working of line follower by detecting the black line on the surface and move along that line. To do this, we need sensors to detect the line. For line detection logic, we need Infrared (IR) Sensors, which consists of IR LED and Photodiode. They are placed in a reflective way side – by – side so that whenever they come in to proximity of a reflective surface, the light emitted by IR LED will be detected by Photo diode.

The following image shows the working of a typical IR Sensor (IR LED – Photodiode pair) in front of a light coloured surface and a black surface. As the reflectance of the light coloured surface is high, the infrared light emitted by IR LED will be maximum reflected and will be detected by the Photodiode. In case of black surface, which has a low reflectance, the light gets completely absorbed by the black surface and doesn't reach the photodiode. Well, maybe up to 98% absorbance, the 2% gets reflected anyway.

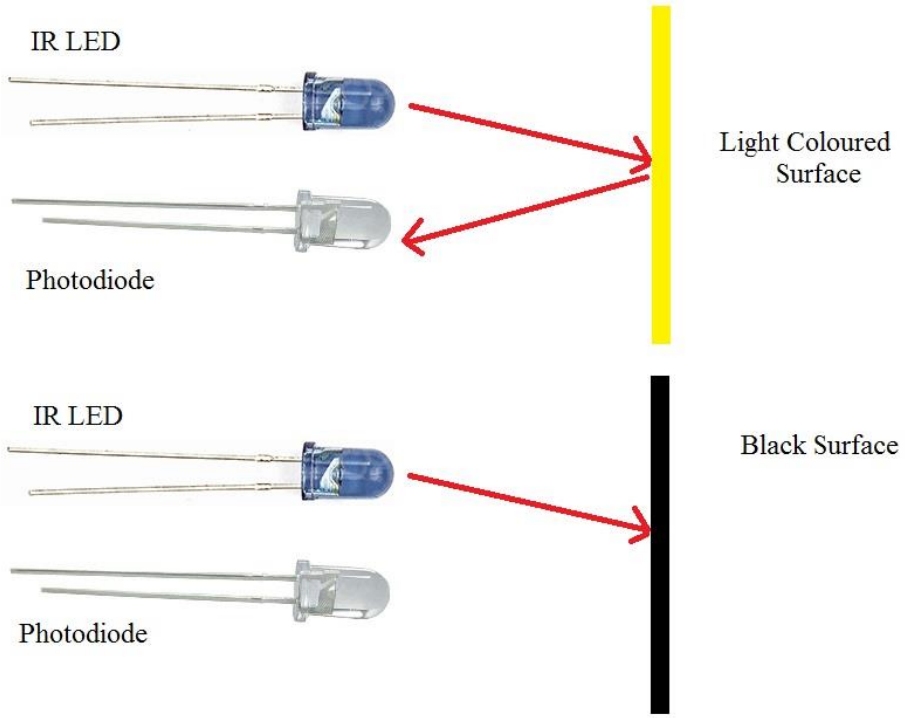


Figure 3. IR Sensors at Work

Using the same principle, we will setup the IR Sensors on the Line Follower Robot such that the two IR Sensors are on the either side of the black line on the floor. The setup is shown below.

Figure 4. IR sensors placed on either side of the black line

When the robot moves forward, both the sensors wait for the line to be detected. For example, if the IR Sensor 1 in the above image detects the black line, it means that there is a right curve (or turn) ahead. Arduino UNO detects this change and sends signal to motor driver accordingly. In order to turn right, the motor on the right side of the robot is slowed down using PWM (if enabled) or Logic LOW (if PWM is disabled), while the motor on the left side is run at normal speed.

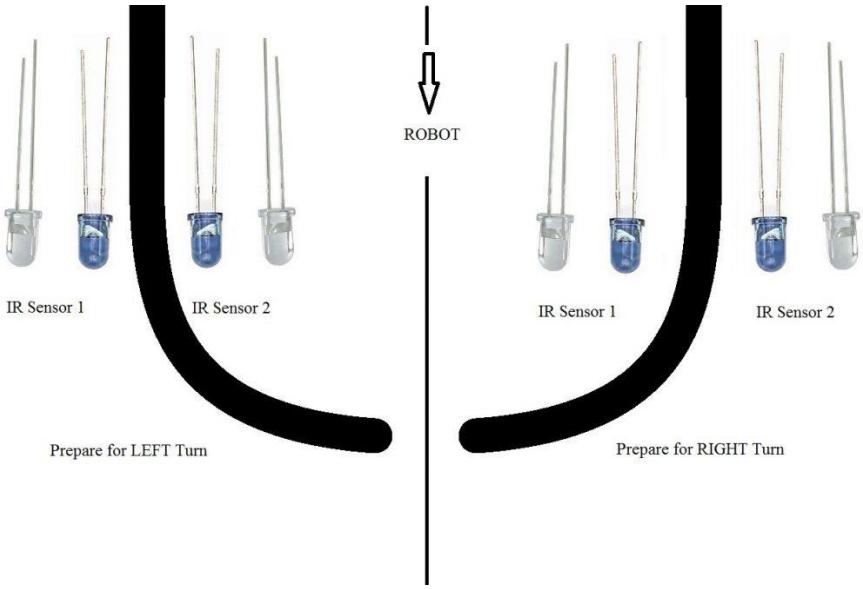


Figure 5. Light and Left Turn of the Robot

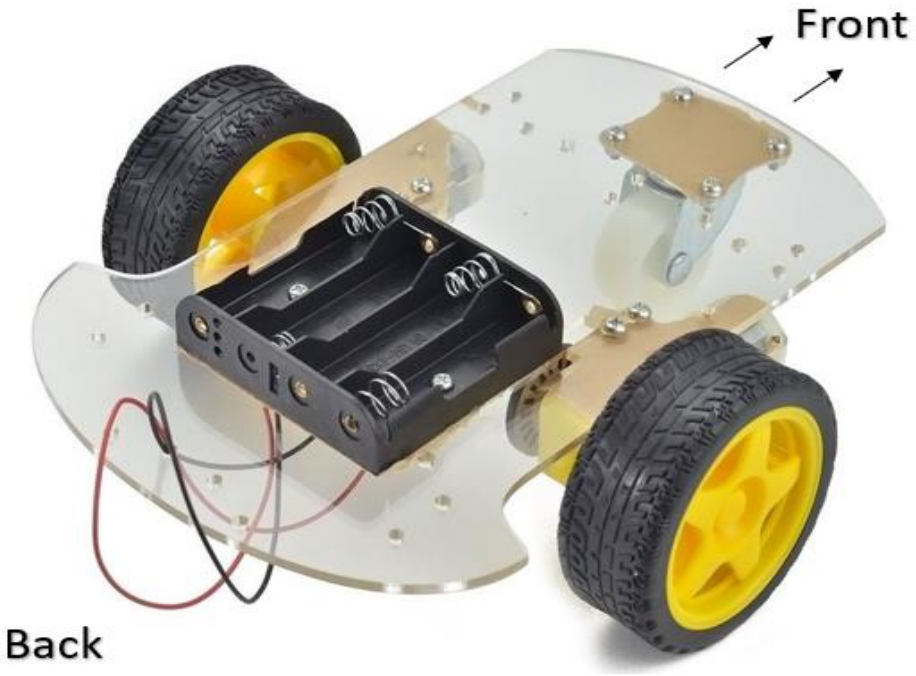
Similarly, when the IR Sensor 2 detects the black line first, it means that there is a left curve ahead and the robot has to turn left. For the robot to turn left, the motor on the left side of the robot is slowed down (or can be stopped completely or can be rotated in opposite direction) and the motor on the right side is run at normal speed. Arduino UNO continuously monitors the data from both the sensors and turns the robot as per the line detected by them.

Can I go beyond Basic Line Follower Design?

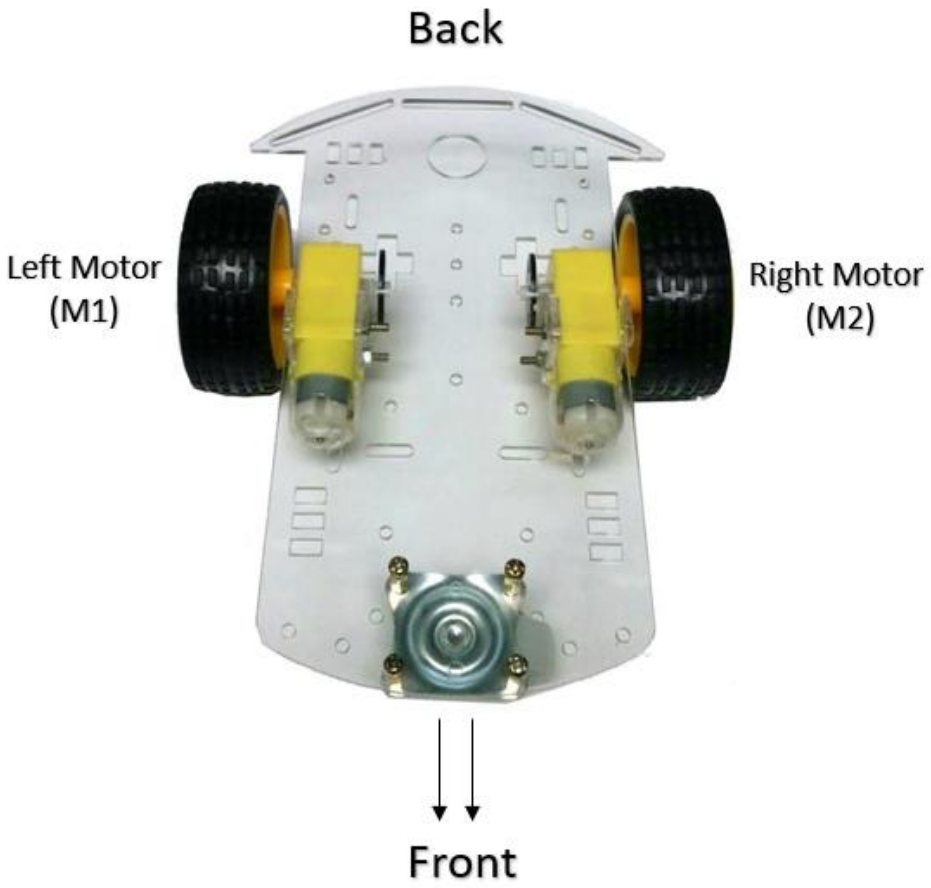
There is the so called “PID Controller”. PID stands for “proportional-integral-derivative”. This controller continuously calculates an error value using its own feedback loop that can consists of any type of sensors. This design is can be integrated in your code only. There is no need to change the hardware setup, just the code. This design is known to improve the performance of the line follower in terms of smooth line following as well as in terms of speed. See this resource for an example: https://tutorial.cytron.io/2016/10/13/make-line-following-robot-faster/. This project however, focuses only on the basic design. But you can go beyond the output I set, which is an excellent thing to do.

Robot Chassis Assembly

This kit is a 2-wheel drive with one caster wheel. This kit comes with an assembly manual. Just follow the instruction on how to do it. After assembly, the chassis should look like the pictures below.



(a)



(b)

Figure 6. Assembled Line Follower Chassis. (a) Top View (b) Bottom view

Arduino Connections

Arduino Board

Figure below shows Arduino Uno Board. Arduino Mega is just the same except that it has more pins for Input/Output, enhanced support for interrupt requests (IRQ), as well as higher storage capacity. But generally, same principles apply.

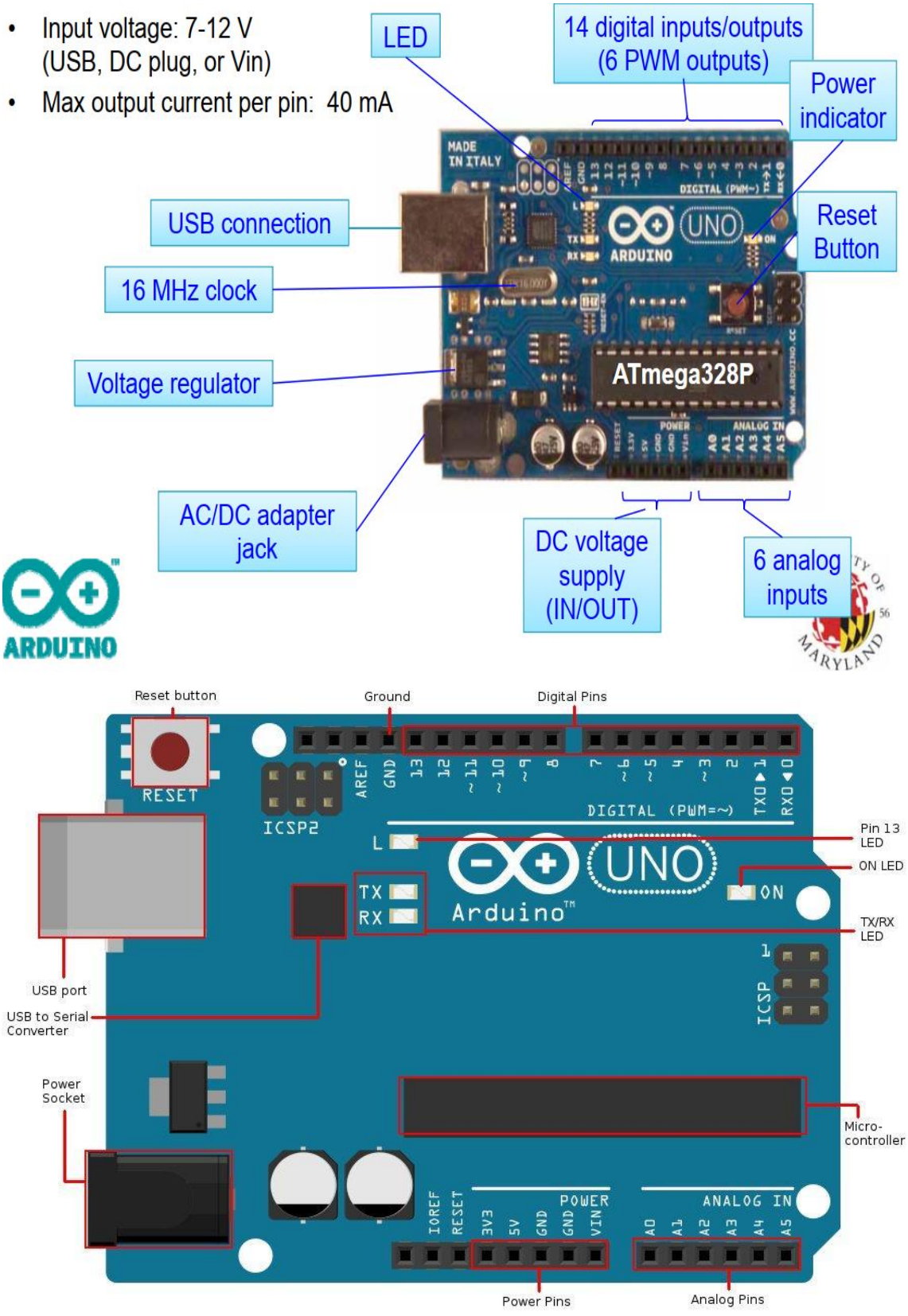


Figure 7. Arduino Board

Digital Line Tracker Sensor

Line tracking sensors comes with different designs and variety. There are single, 3-arrays, and 5-arrays configurations. This device senses signal through Infrared signalThe sample sensors are shown below.

Single Sensor

You need several of this depending on the sensors you need. For example, you need 3-sensors configuration, then you need to have obtain 3 pieces of this. Its pins are VCC (5 volts input pin), GND (Ground), and S1 (digital input pin). Each of these must be supplied with 5 volts and grounds using just one power source.

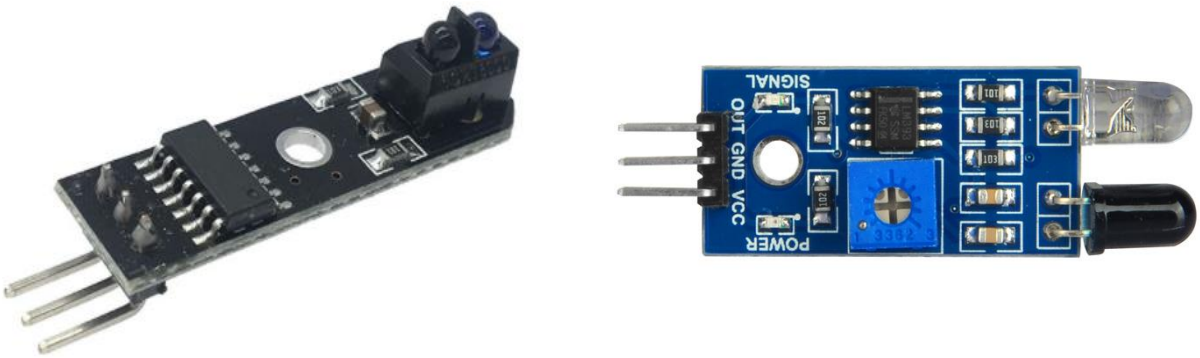


Figure 8. Single Sensor Modules

Multi-channel Sensors

With this configuration, several sensors are already packaged in one unit. Its power pins are labeled VCC (5 volts input), GND (Ground). Its digital sensor pins are S1 (sensor 1), S2 (sensor 2), and Sn (sensor n). As it comes with different label, you will see some designs have a label of SS1 or OUT1, SS2, OUT2, etc. These are just the same anyway. Please see the pictures.

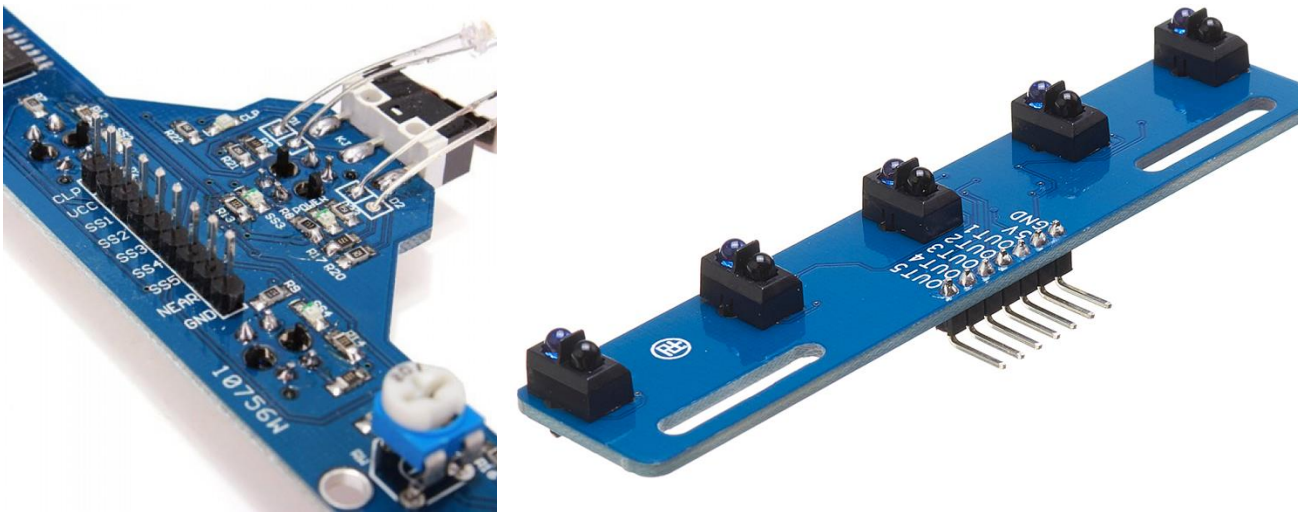


Figure 9. Multi Sensors Module

Connection from Arduino to Line Tracker Sensors

In this project, we will use 5 sensors. But you can tweak this to make it 2, 3, 4 sensors. In fact, I provided a code of 4 sensors line follower for your reference. To connect the sensor module to the Arduino board, we need Male- Female jumper cable.

Figure 10. Male-Female Connector

It's better to have some color code convention in your wiring for clarity. For VCC, use red wire. For GND, use black wire. For sensors, use the other colors.

Tip: In case you run out of male-female wire, you can cut one male-male wire and one female-female wire in the middle. Then connect the two ends of cut wires to produce male-female wire.

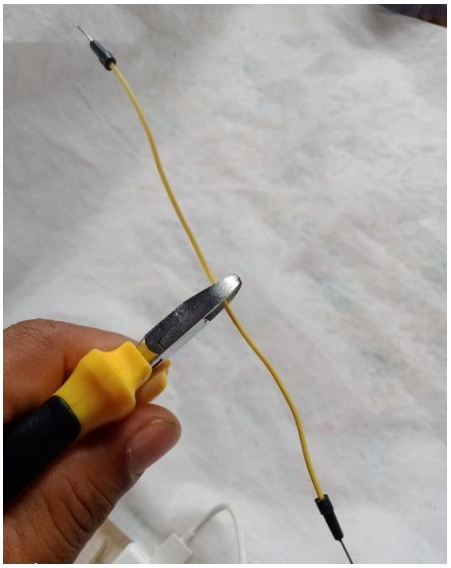


Figure 11. Cutting Male-Male Connector

The table below presents the pin connections used for your reference. You can actually change this if you want. Just be sure the changes you made in the hardware are reflected in your code.

Arduino Pins	Sensor Pins (S/SS/OUT)	Description
2	Sensor 1	Leftmost sensor
3	Sensor 2	2nd sensor from left
13	Sensor 3	Middle sensor
4	Sensor 4	2nd sensor from the right
5	Sensor 5	Rightmost sensor

Table 1. Arduino to Sensor Module Connection

Figure 9 provides the visual connection of the Arduino board and the multi-sensor module (This is what you'll do in this project)

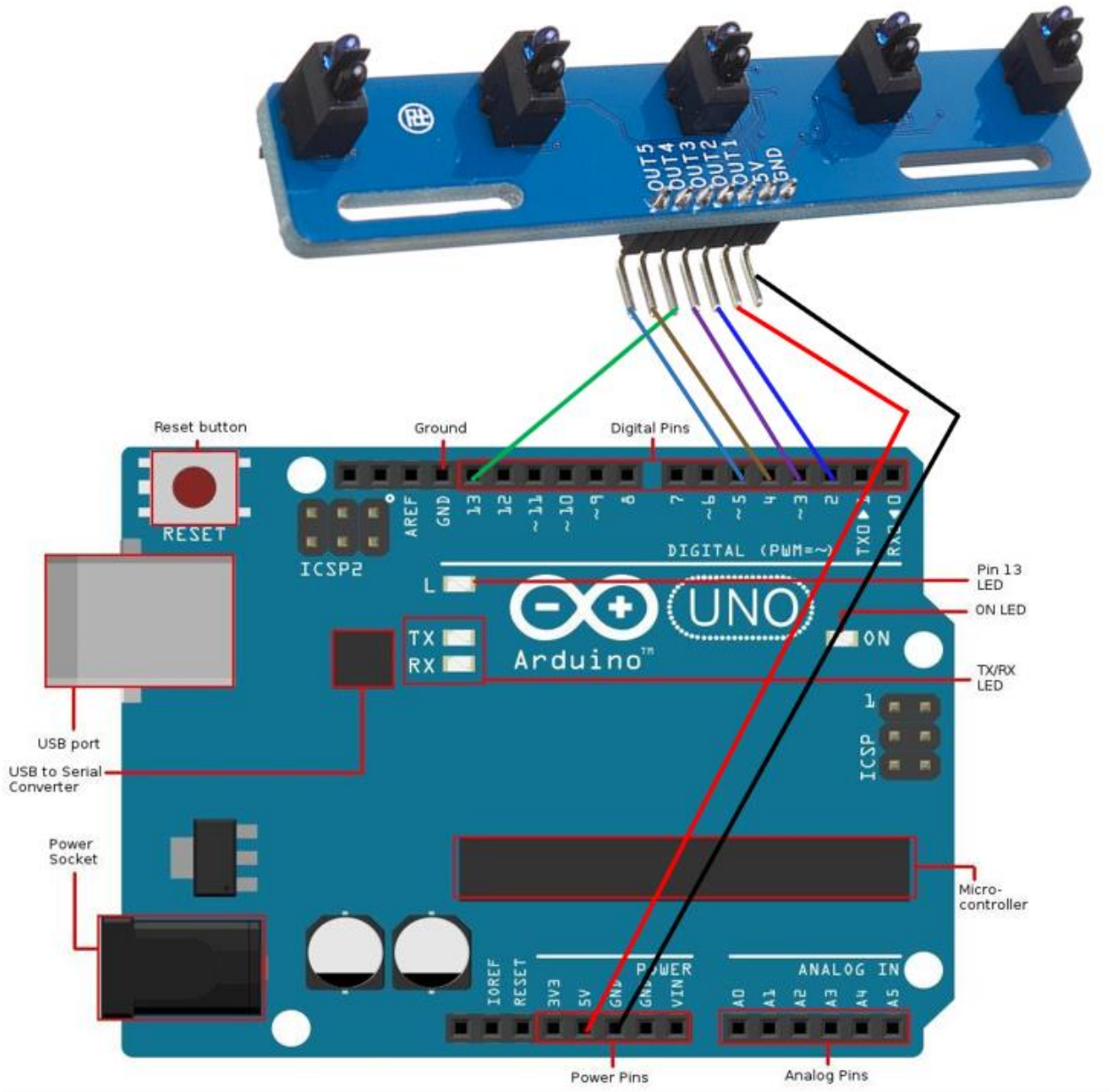


Figure 12. Arduino Connection to Multi-sensor Module

Note

As shown in Figure 9, There are 3 ground pins in the Arduino board, 2 pins from the Power Pins bank and 1 pin aligned with the digital input/output pins. You can connect the sensor module ground to any of these available Arduino GND pins.

Figure 10 shows the connection between the board and 5 single-sensor modules (For those who have single sensors only). For this, you might need a small breadboard to put all VCC and GND wires together.

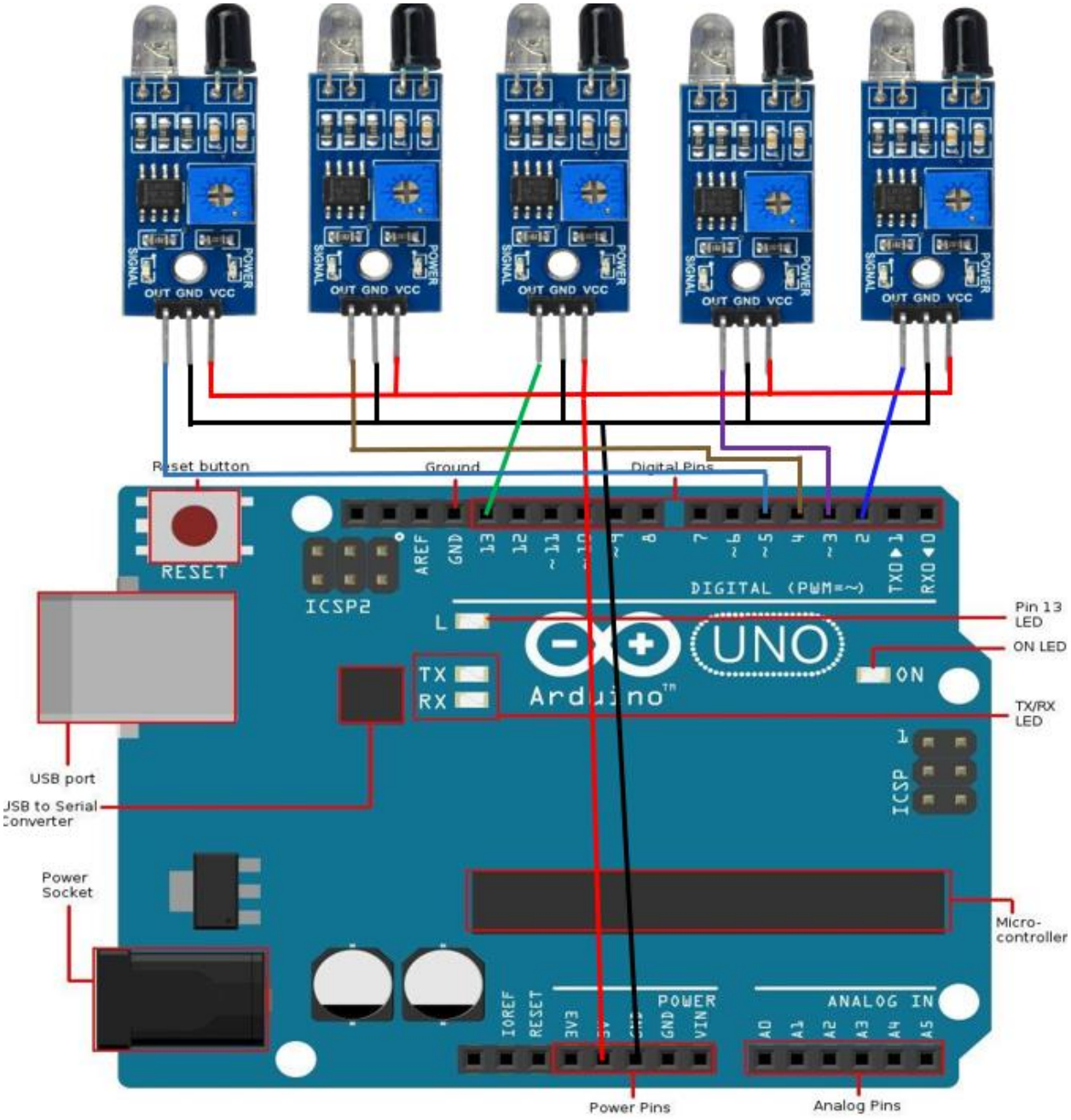


Figure 13. Arduino Connection to 5 Single-sensor modules

With this configuration, the line sensor module obtains its power from the Arduino board through its 5v output port. So, the Arduino must be provided with power first before the line sensor module can work. Figure below shows the actual picture of Arduino board and line sensor module connection.

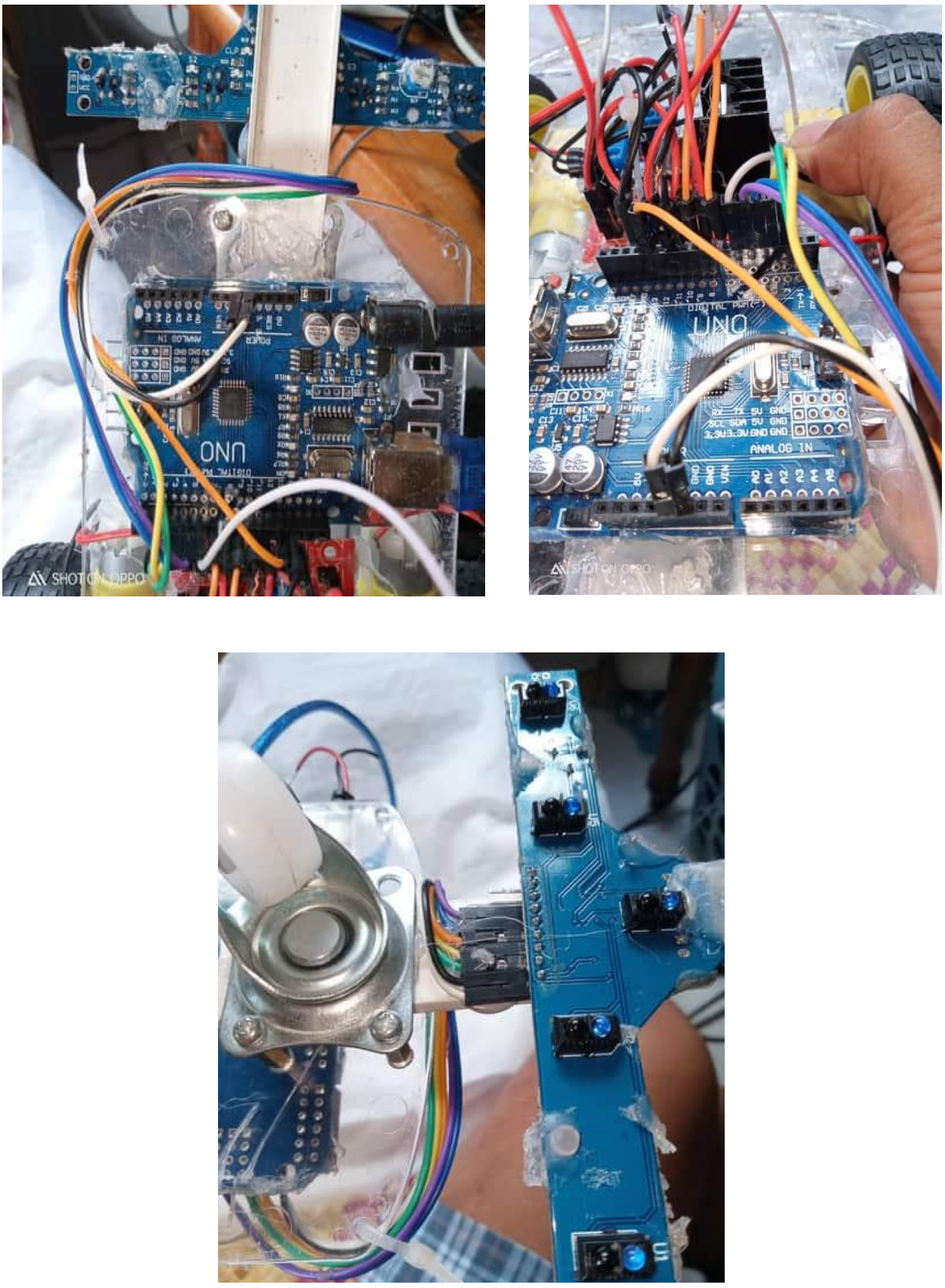


Figure 14. Arduino board connection to Line sensor module

Ways to Power Arduino Board

There are actually three ways to power Arduino board, but for this project, the two safer way will be used here.

Way 1. Use of Power Bank through the Arduino's USB port

Figure x shows the use of 5-volts power bank to Arduino board.

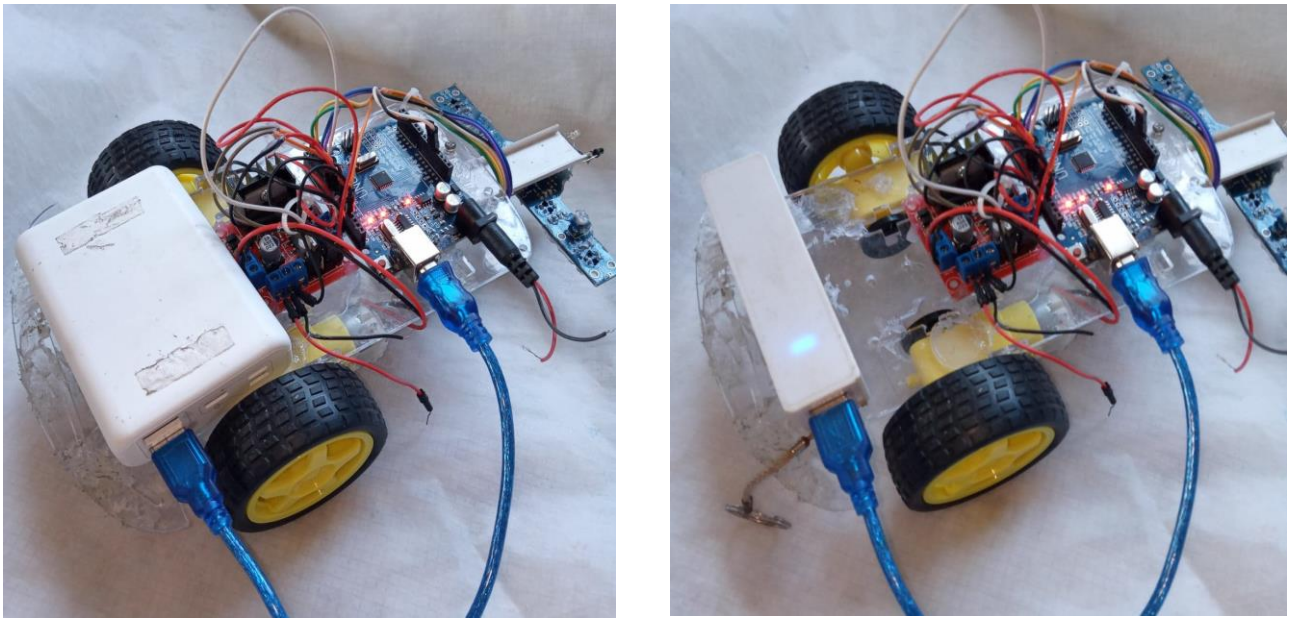


Figure 15. Power bank supply through Arduino's USB port

Way 2. Use of 9 volts battery through Arduino's power socket

Figure 13 illustrates the use of 9-volts power bank to Arduino board.

(a) (b)

Figure 16. Battery supply through Arduino's power socket. (a) 9v batt (b) 7.4 volts batt

Note: With Way 2, the battery supply voltage must be 7-12 volts for it to work properly.

Connection from Arduino Board to L298N Motor Driver Shield

Note: These are the web resources I instructed you to learn about how L298N works.

https://www.instructables.com/id/How-to-Use-L298n-to-Control-Dc-Motor-With-Arduino/

https://lastminuteengineers.com/l298n-dc-stepper-driver-arduino-tutorial/

https://www.youtube.com/watch?v=I7IFsQ4tQU8

https://www.youtube.com/watch?v=Da4HY7HZ6h0

It is advised that you study first how L298N works by following the tutorials and experiments presented in there before proceeding to the next lesson.

L298N Motor Driver Shield

As you learned on the web resources, I've uploaded in Schoology, L298N is an H-bridge type motor driver shield that supports 2 DC motors. H-bridge allows motors to run in forward or reverse direction by switching the designated pins HIGH or LOW. L298N receives signal from Arduino which in turn drives the DC motors connected to it.

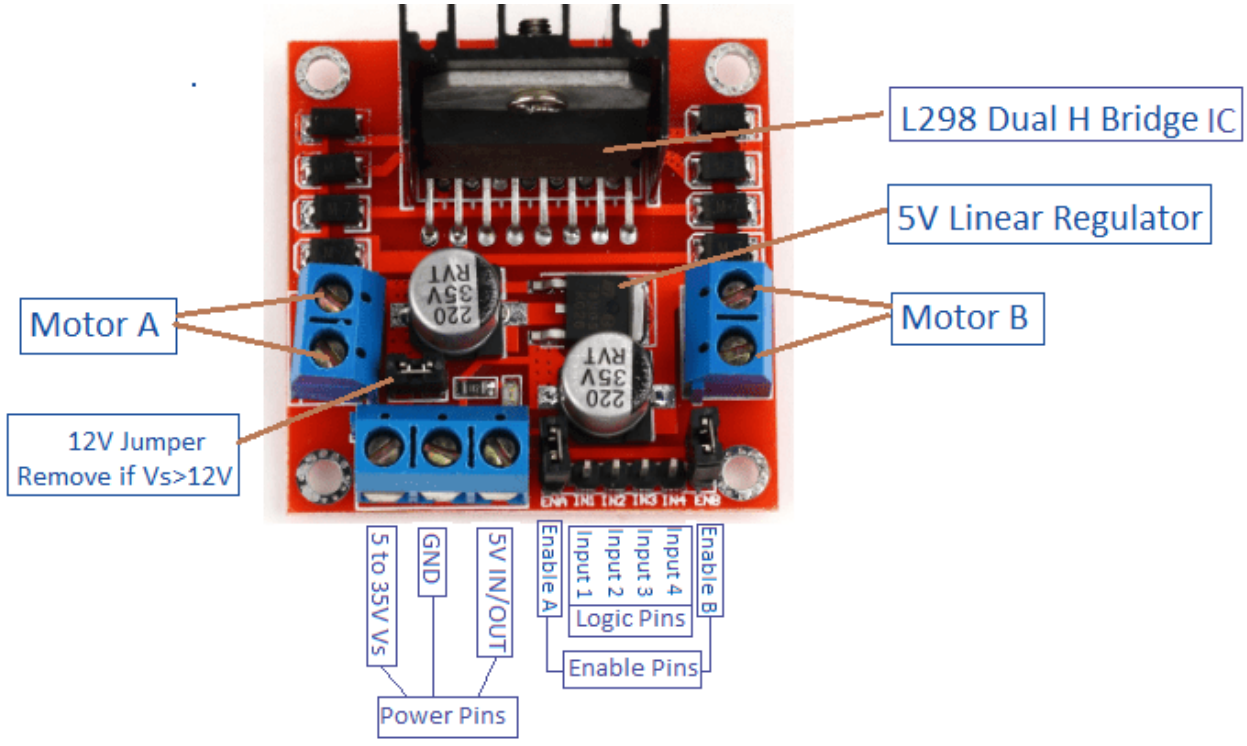


Figure 17. L298N Parts

Connecting L298N Logic pins to Arduino Board and Sharing Common Ground Point

The table below presents L298's Logic Connection pin connections used for your reference. You can actually change this if you want. Just be sure the changes you made in the hardware are reflected in your code. Again, use male-female wires here.

Arduino Pins	L298N Motor Driver	
	Pins	Description
12	EnA	Enables/disables motor A (HIGH/LOW) or (PWM)
7	EnB	Enables/disables motor B (HIGH/LOW) or (PWM)
11	IN1	Receives HIGH/LOW signal; tandems with IN2 that drives motor A
10	IN2	Receives HIGH/LOW signal; tandems with IN1 that drives motor A
9	IN3	Receives HIGH/LOW signal; tandems with IN3 that drives motor B
8	IN4	Receives HIGH/LOW signal; tandems with IN4 that drives motor B
GND	GND	Ground pin of L298N is used to share common Ground with Arduino

Table 2. Arduino to L298N Motor Logic Connection

Figure 15 shows the interconnection of Arduino board and L298N motor driver.

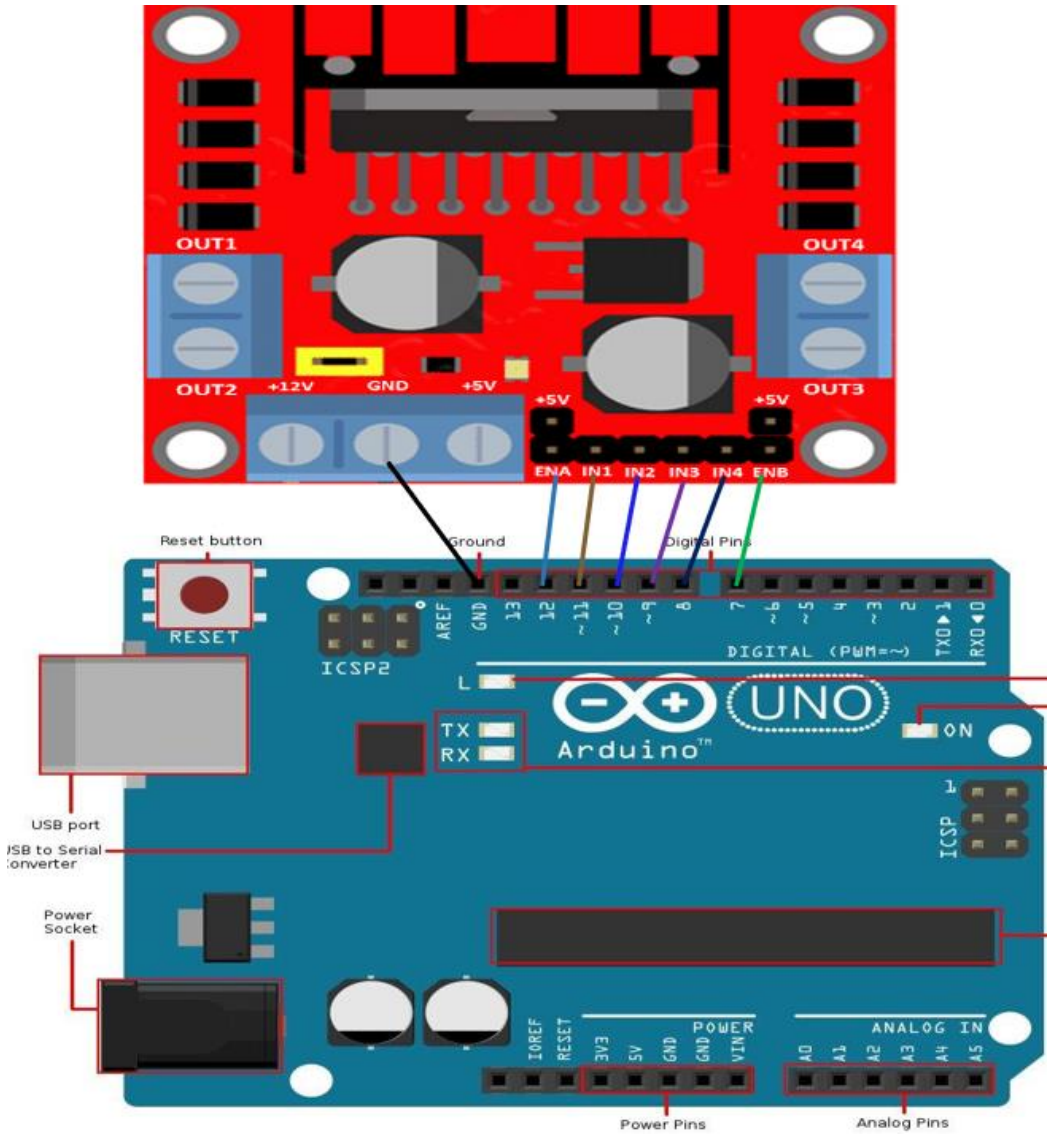


Figure 18. L298N Connection to Arduino Board

Connecting Motors

Figure 16 shows the connection of two DC motors to the Motor driver.

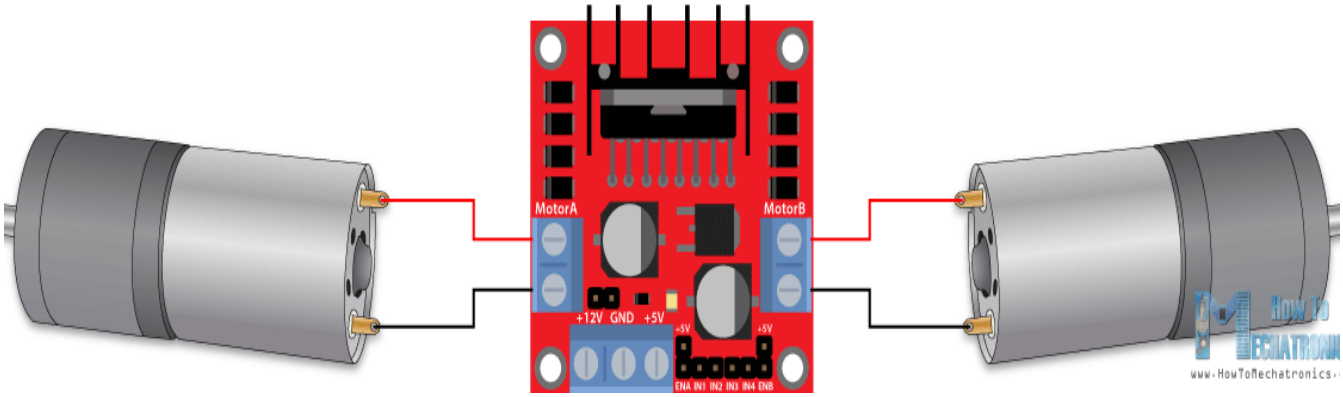


Figure 19. L298N Connection to DC motors

The wires are connected to the motor terminals - Motor A, Motor B, as well as power supply terminals by inserting them and making them firm using a small screw. See Figure 17 below.

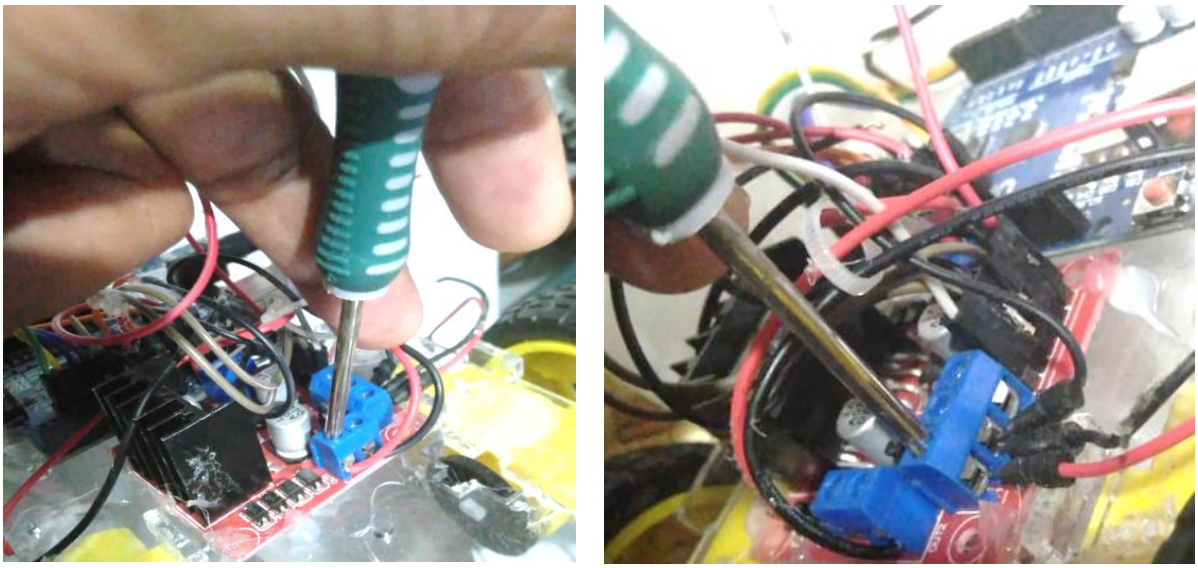


Figure 20. Tightening the wire using screw driver

Ways to Power L298N Motor Driver

There are two ways to power the L298N motor driver shield.

Way 1. Using power bank (5 volts)

Use this option if you do not intend to use pulse width modulation (PWM). To do this, you need to hack a USB cable by getting only the positive wire (red) and ground (black). USB cable has 4 wires inside. Usually, the positive and ground wires are on the leftmost and rightmost side. Use your trusted multi meter to determine the needed wires.

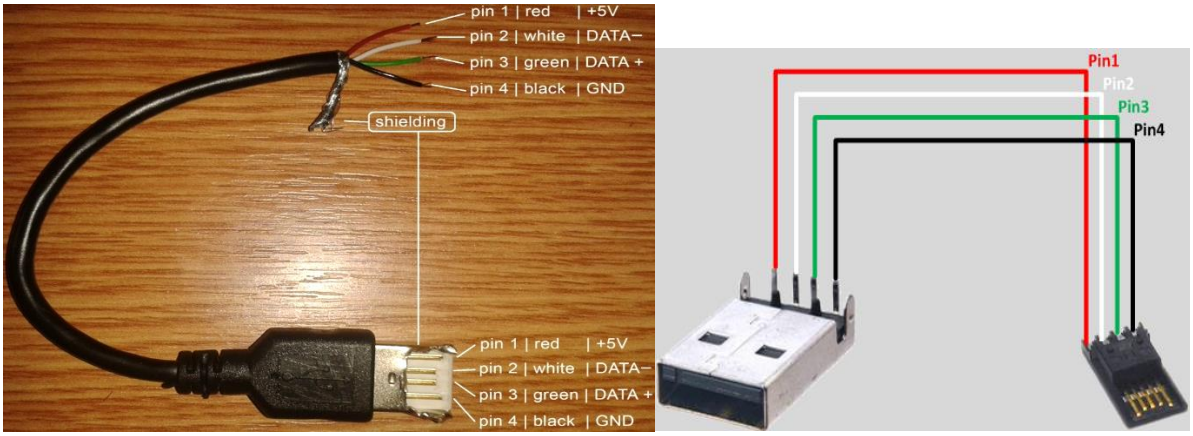


Figure 21. Parts of a USB Cable

Figure 19 shows the hacked USB cable and the use of 5-volts power bank to power L298N motor driver.

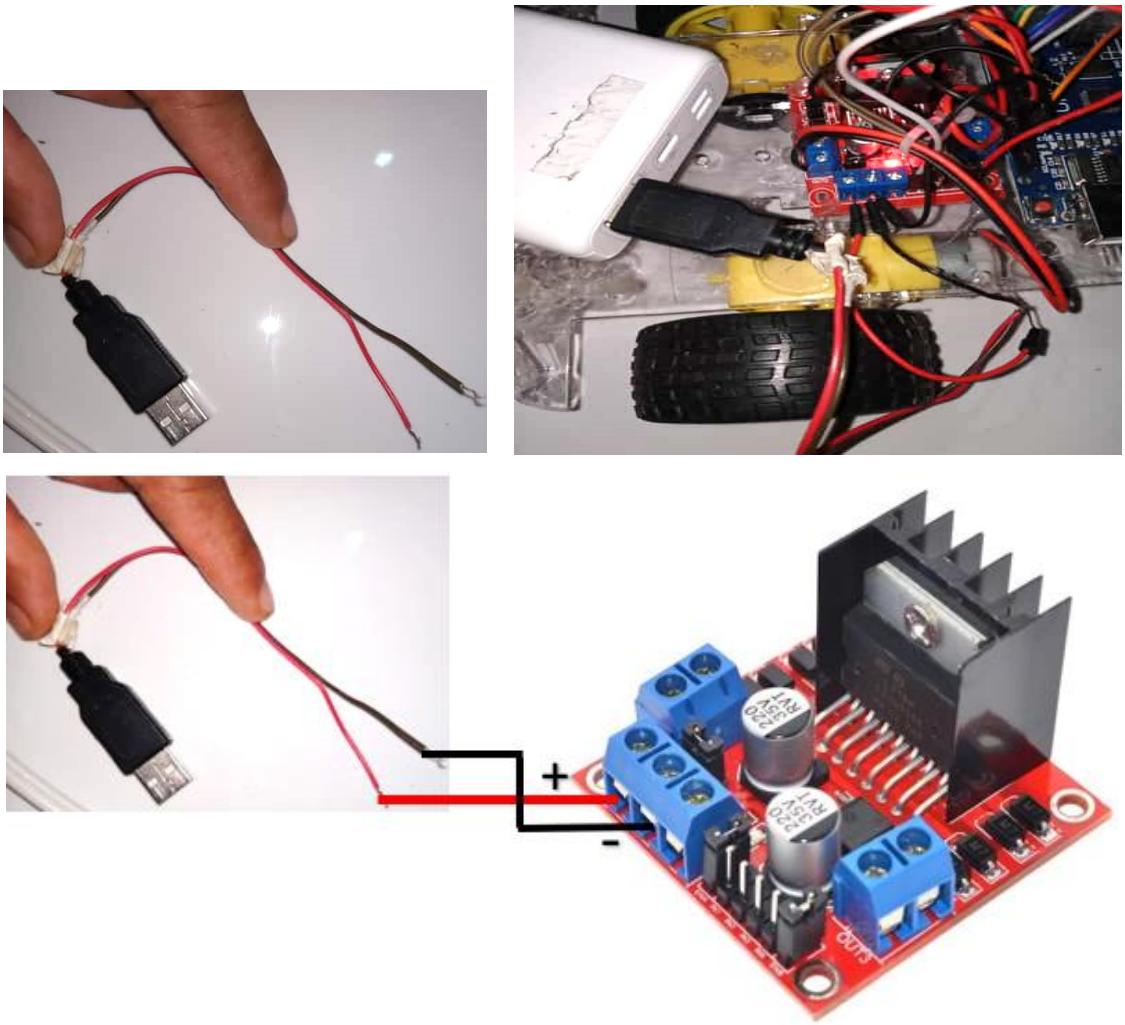


Figure 22. Power bank supplying 5v to L298N

Note: When using this option, you can power both the L298N driver and Arduino board at once using power bank. See the figure below.

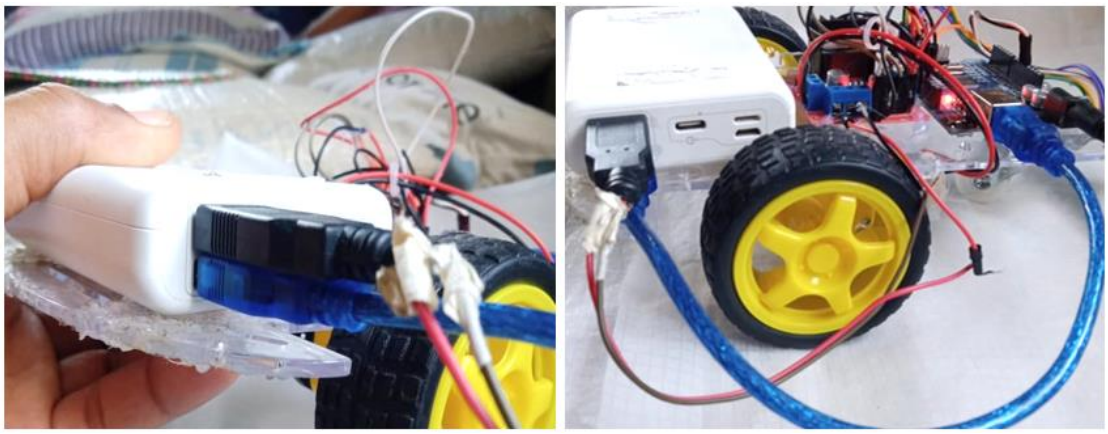
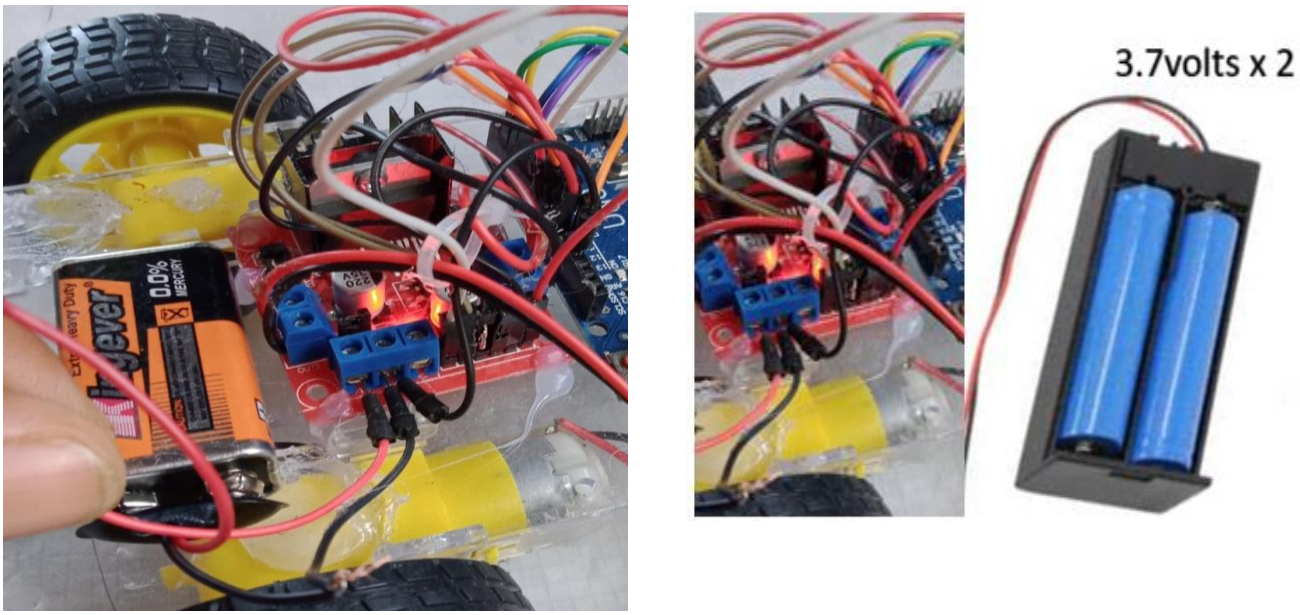


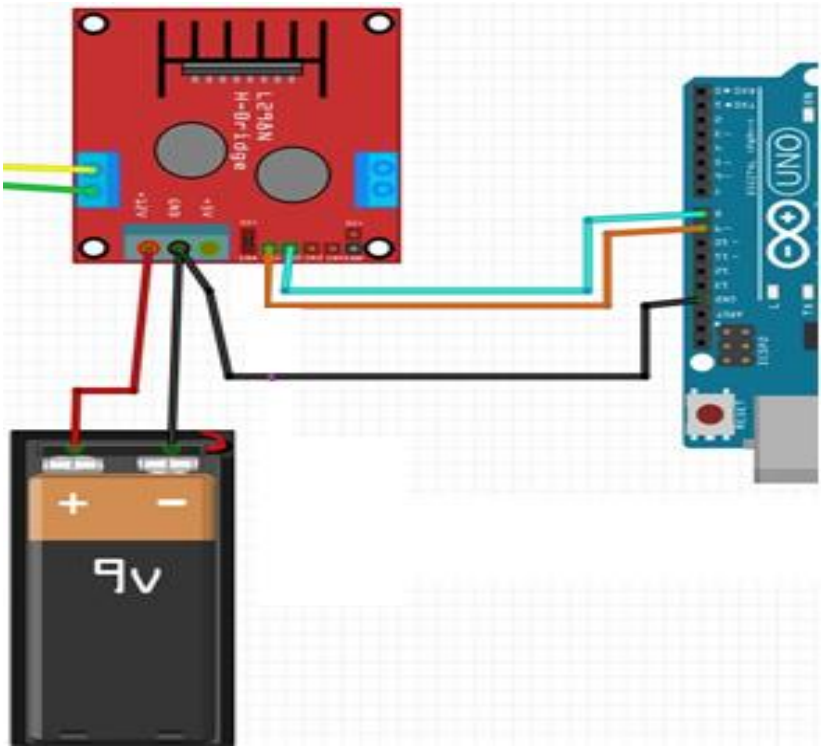
Figure 23. Power bank supplying 5v to L298N

Way 2. Use of more than 5 volts battery (7.4v, 9v, 12v, up to 35v)

If you wish to use PWM to control the speed of the DC motor through the motor driver, it is advisable that your power supply is at least 2 volts higher than 5 volts base requirement. Figure 13 illustrates the use of 9-volts and 7.4 volts batteries to power L298N. It is also recommended that you have a separate power supply for the Arduino board and the L298N motor driver.



(a)



(b)

Figure 24. Use of Battery. (a) Actual pictures (b) Circuit Diagram

Enabling/Disabling PWM

You can either disable the PWM to utilize only the full speed of the DC Motors or enable PWM to control motor's speed. To do this, there is a minor hardware setting you need to perform on the L298N motor driver. On L298N, there are two (2) jumper switches located on the Logic pins bank, inserted specifically on the ENA and ENB pins by default. Leaving it installed means that PWM is disabled. Meaning, you can only run the motor at full speed and a full stop. See Figure below.

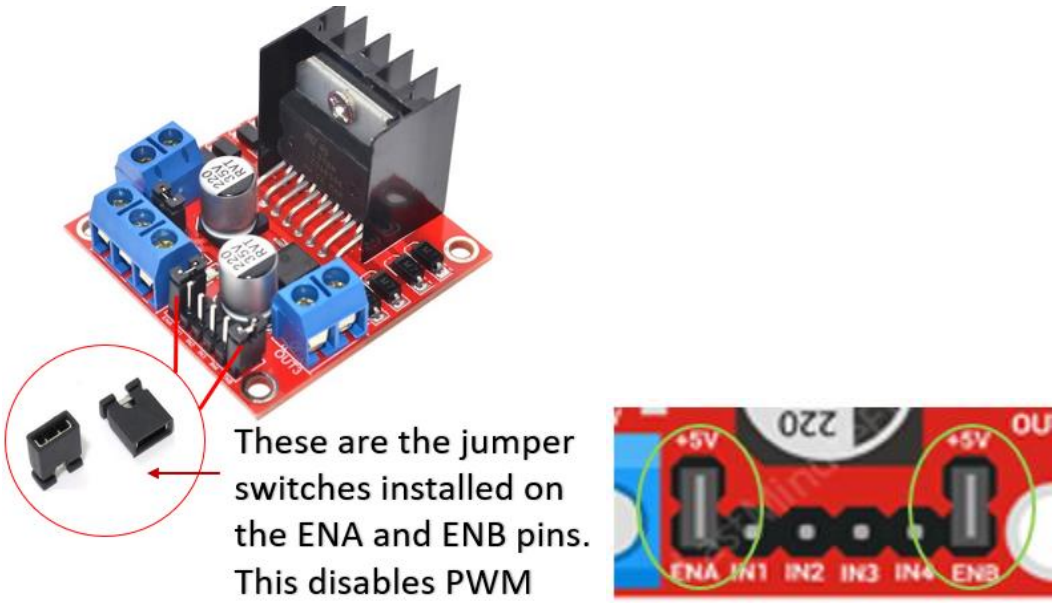


Figure 25. Jumper switches installed (PWM is disabled)

Obviously, if there are already wires connected to these pins, these two wires connected to the Arduino on pin 7 and pin 12 must be removed from the ENA and ENB pins in order for the jumpers to be inserted. See figure below.

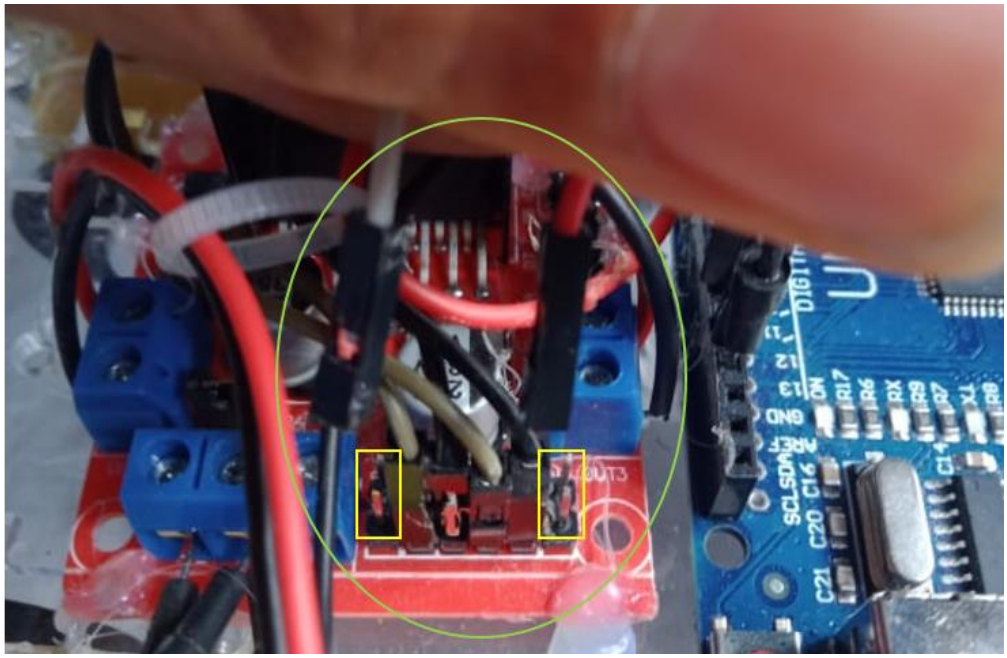


Figure 26. Removing wires from ENA and ENB pins before inserting jumper switches

To enable PWM, just remove the two jumper switches and insert the wire coming from pin 12 of Arduino to ENA pin, and insert the wire from pin 7 of Arduino to ENB pin of L298N. See figures below.

Figure 27. Enabling PWM

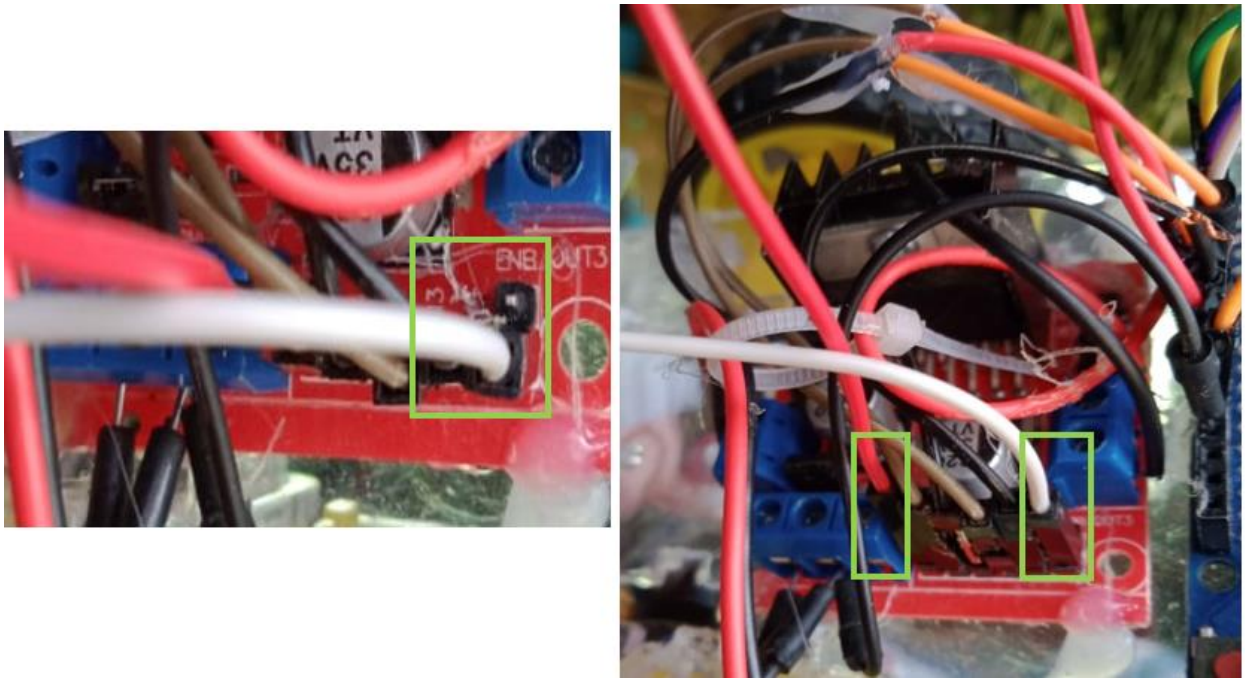


Figure 28. Wires inserted to ENA and ENB pins

A Guide on Component Placement and Measurement

This is some sort of guide only. Your whole design of the line follower will be depending on your creativity. I know that young people like you have brighter ideas. You can watch many designs from the Internet eg. website articles or Youtube and follow it. With that said, here is my sample prototype.

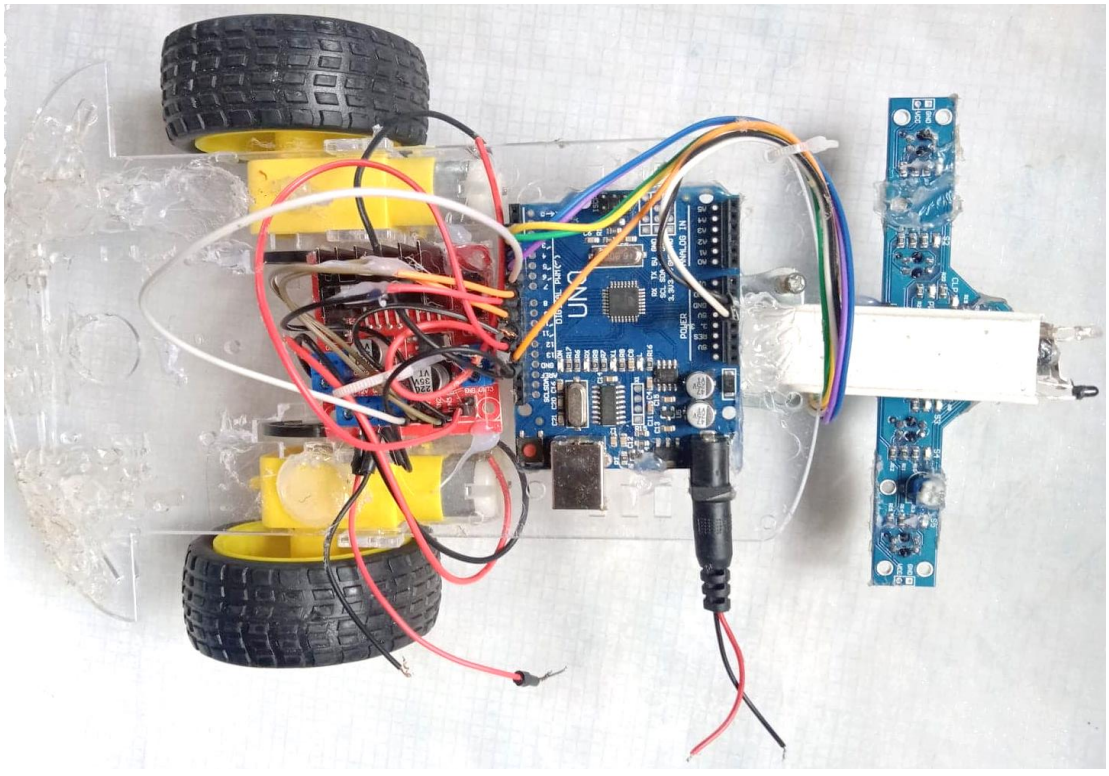


Figure 29. Top view of the prototype

KEEP IN MIND!

Take note that the weight of all the components on top of the chassis must be balance, and concentrate most of the weights at the middle. More weights you put at the back will make the robot wobbly while more weights you put at the front will make your robot a little difficult to turn.

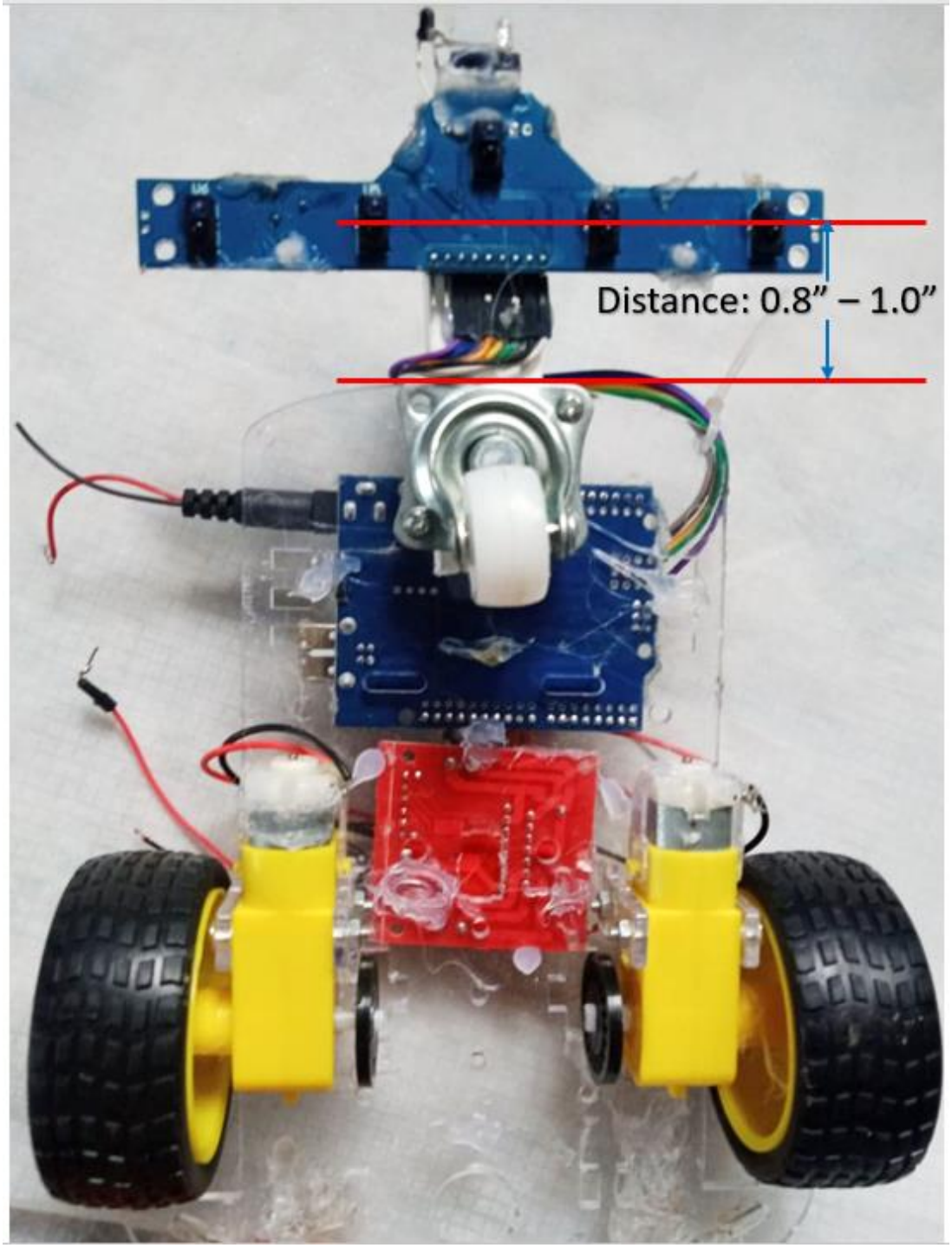


Figure 30. Bottom view of the prototype

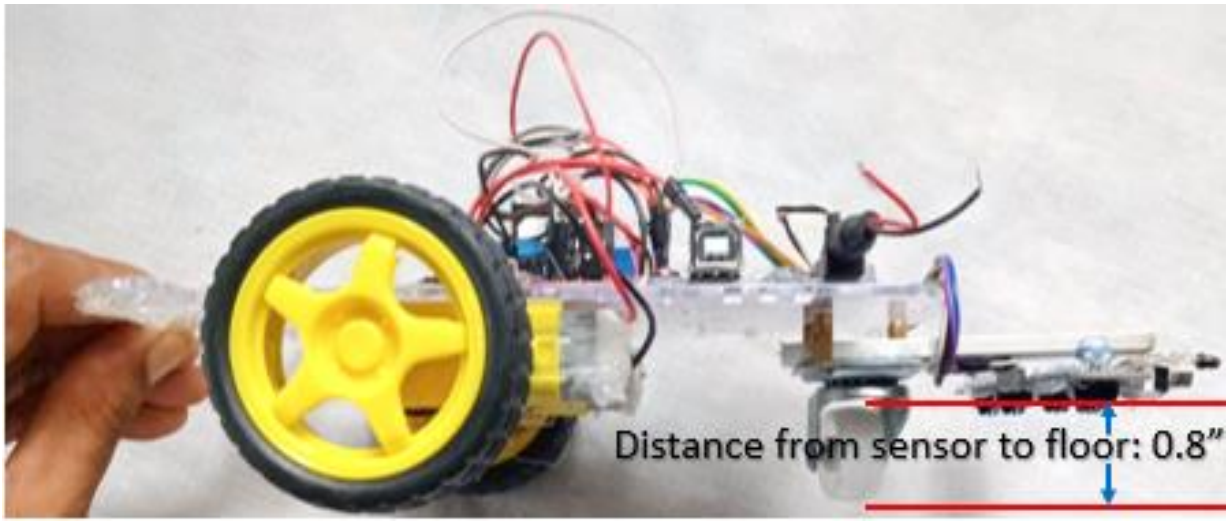


Figure 31. Side view of the prototype

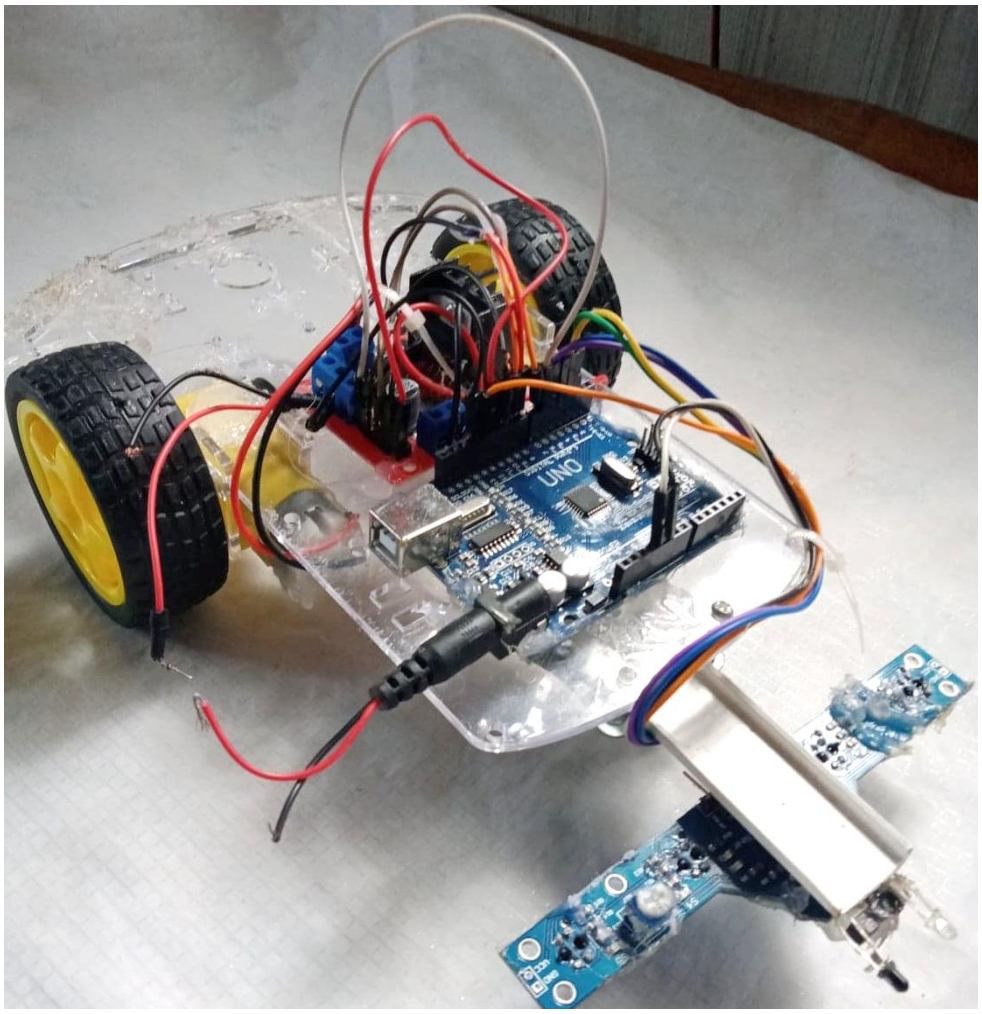


Figure 32. Diagonal view of the prototype

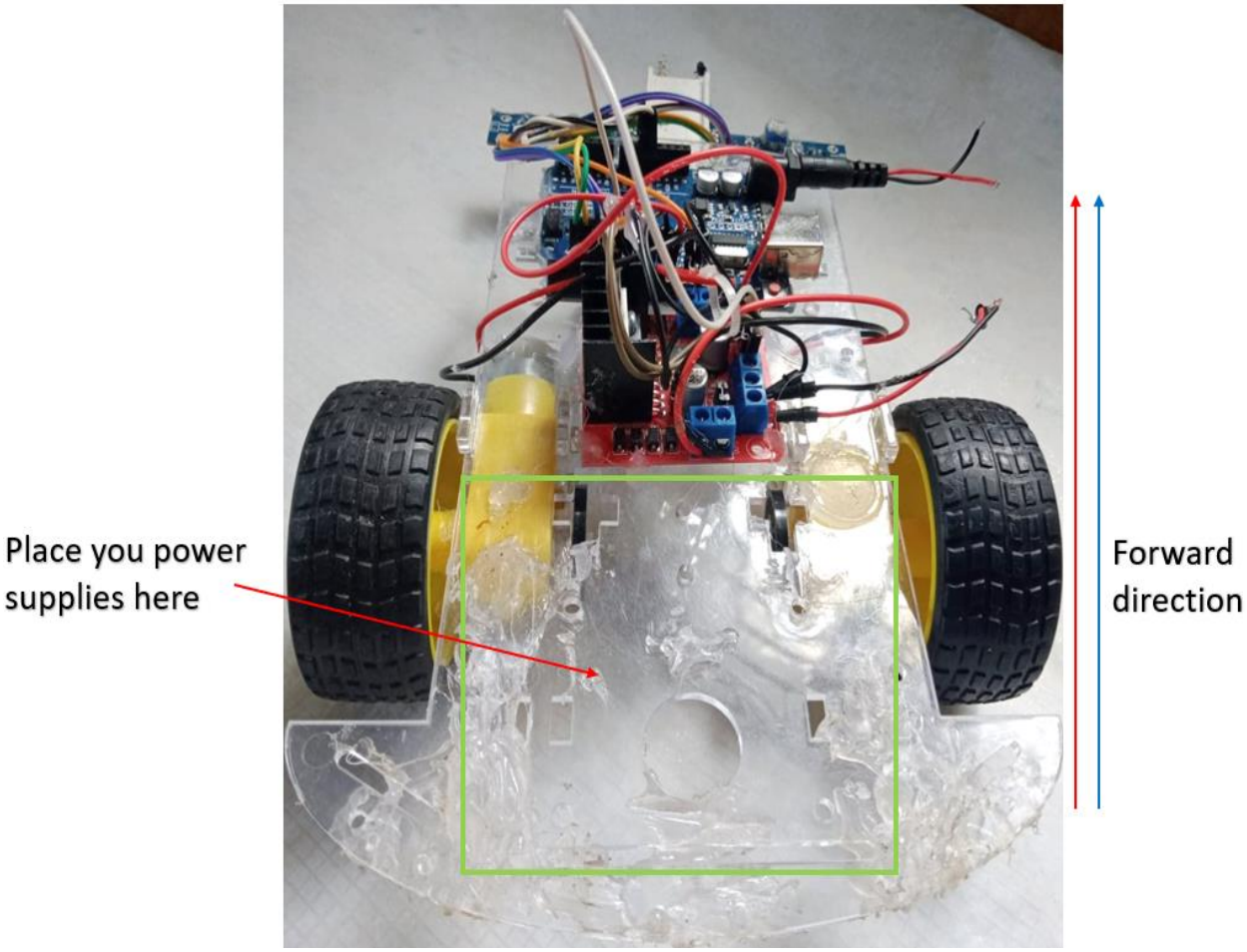


Figure 33. Diagonal view of the prototype

Simple Calibration

In Figure 27 and Figure 28, I suggested about the measurement from sensor to floor, but it is just purely suggestive. We need some kind of calibration for a more responsive and effective sensor system. We do not need tools here, some experimentation, trials and errors, and some luck 😊.

Calibrate Sensor Response on the Black Surface

As you know, we need to properly distance the sensor to the floor. Sensor which is too far to the floor would not capture or may be delay in sensing the line. Too near would make the sensor detect dark surface same as white surface due to the absorption of some reflected light from the dark surface. This in turn, will we be interpreted by the sensor as white surface.

To determine the best distance possible, you need to do this simple calibration before fixing the line tracking sensor on the front of your robot. This must be done iteratively until the right distance is found. *For this, just watch the video I uploaded in Schoology together with this tutorial on how to do a very simple calibration.*

Determine the width of line and distance of two sensors for the left and right edge of the line

Again, there is no exact measurement for this. Trial and error can be applied here too. But as a rule of thumb, the two sensors that detects left and right edge of the line must not be directly over the line, but instead, must be off the line. But then again, do some experimentation so as to maintain an optimum distance between the sensors and the edges of the line. Line materials can be a black cardboard, black paper, or even a black electrical tape. See figure below.

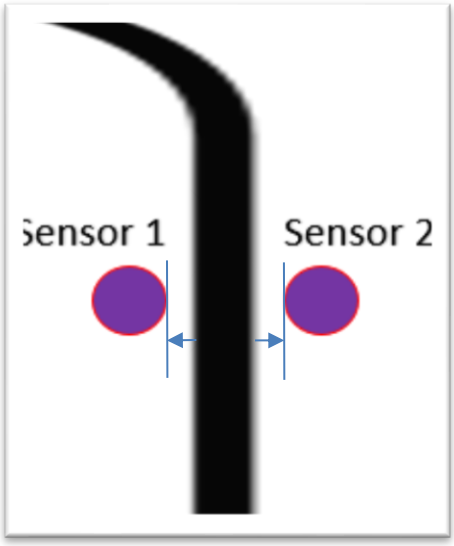
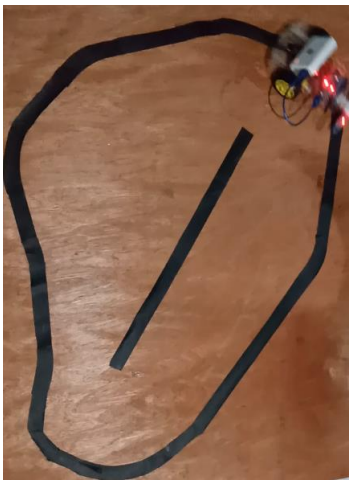


Figure 34. Distance of the sensors from the line edges

Challenge - The kind of track that the robot can follow so far is shown below.



Modify the code so the robot can follow the following turn:

References

1. https://www.electronicshub.org/arduino-line-follower-robot/
2. https://tutorial.cytron.io/2016/10/13/make-line-following-robot-faster/
3. *https://www.instructables.com/id/How-to-Use-L298n-to-Control-Dc-Motor-With-Arduino/*
4. *https://lastminuteengineers.com/l298n-dc-stepper-driver-arduino-tutorial/*
5. *https://www.youtube.com/watch?v=I7IFsQ4tQU8*
6. *https://www.youtube.com/watch?v=Da4HY7HZ6h0*
7. *Making Robots with the Arduino by Gordon McComb*
8. *Robotics, A Project-based Approach by Lakshima Prayaga, Chanora Prayaga, Alex Whiteside, and Ramakrishna Suri, pg. 36*