

# SMART Learning Kit

[www.quadstore.in](http://www.quadstore.in)



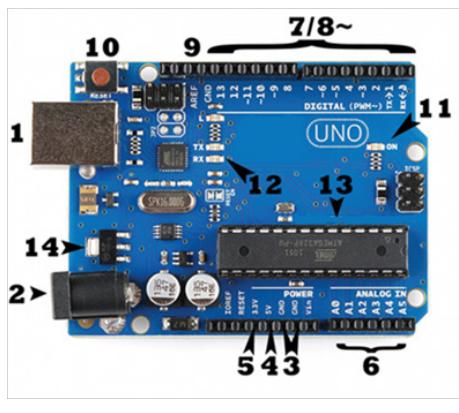
# TABLE OF CONTENTS

2

• BEGINNER LEVEL PROJECTS	5
1. Installing Arduino IDE and uploading code	5
2. Installing Libraries	11
3. Serial Monitor – Send Data	14
4. Serial Monitor – Receive Data	16
5. Blinking LED using UNO R3 boards built-in LED	17
6. Blinking LED using external LED	18
7. Smooth LED	19
8. Push Button	20
9. Potentiometer	21
10. Photoresistor	22
11. RGB LED - change	23
12. Active Buzzer	24
13. Passive Buzzer	25
14. Servo Motor	26
15. Servo Motor control using potentiometer	27
• INTERMEDIATE LEVEL PROJECTS	28
16. DHT Temperature Sensor using 3 pins	28
17. 1-digit 7 Segment Display	29
18. I2C 1602 LCD display	30
19. Ultrasonic Sensor HC-SR04	31
20. Relay	32
21. Bluetooth HC 05	33
22. L293D Motor Shield	39
23. Controlling DC Motor using Relay	40
• ADVANCE LEVEL PROJECTS	41
24. 2WD Car Chassis Assembly Instructions	41
25. Line Following Robot Car	44
26. Obstacle Avoidance Robot Car	51
27. Torch Light Following Robot Car	58
28. Bluetooth Controlled Car using Android phone app.	64
29. Wifi ESP8266 with Blynk app	74
30. Introduction to NodeMCU ESP8266	87

# INTRODUCTION: UNDERSTANDING ARDUINO UNO R3 BOARD

There are many varieties of Arduino boards that can be used for different purposes. Some boards look a bit different from the one below, but most Arduino's have majority of these components in common:



## POWER (USB / BARREL JACK)

Every Arduino board needs a way to be connected to a power source. The Arduino UNO can be powered from a USB cable coming from your computer or a wall power supply that is terminated in a barrel jack. In the picture above the USB connection is labeled (1) and the barrel jack is labeled (2).

### Note:

Do NOT use a power supply greater than 20 Volts as you will overpower (and thereby destroy) your Arduino.

The recommended voltage for most Arduino models is between 6 and 12 Volts.

Pins (5V, 3.3V, GND, Analog, Digital, PWM, AREF)

The pins on your Arduino are the places where you connect wires to construct a circuit (probably in conjunction with a breadboard and some wire). They usually have black plastic 'headers' that allow you to just plug a wire right into the board. The Arduino has several different kinds of pins, each of which is labeled on the board and used for different functions.

- \* **GND (3):** Short for 'Ground'. There are several GND pins on the Arduino, any of which can be used to ground your circuit.

- \* **5V (4) & 3.3V (5):** As you might guess, the 5V pin supplies 5 volts of power, and the 3.3V pin supplies 3.3 volts of power. Most of the simple components used with the Arduino run happily off of 5 or 3.3 volts.

- \* **Analog (6):** The area of pins under the 'Analog In' label (A0 through A5 on the UNO) are Analog In pins. These pins can read the signal from an analog sensor (like a temperature sensor) and convert it into a digital value that we can read.

- \* **Digital (7):** Across from the analog pins are the digital pins (0 through 13 on the UNO). These pins can be used for both digital input (like telling if a button is pushed) and digital output (like powering an LED).

\* **PWM (8)**: You may have noticed the tilde (~) next to some of the digital pins (3, 5, 6, 9, 10, and 11 on the UNO). These pins act as normal digital pins, but can also be used for something called Pulse-Width Modulation (PWM). We have a tutorial on PWM, but for now, think of these pins as being able to simulate analog output (like fading an LED in and out).

\* **AREF (9)**: Stands for Analog Reference. Most of the time you can leave this pin alone. It is sometimes used to set an external reference voltage (between 0 and 5 Volts) as the upper limit for the analog input pins.

## RESET BUTTON

Just like the original Nintendo, the Arduino has a reset button (10). Pushing it will temporarily connect the reset pin to ground and restart any code that is loaded on the Arduino. This can be very useful if your code doesn't repeat, but you want to test it multiple times. Unlike the original Nintendo however, blowing on the Arduino doesn't usually fix any problems.

## POWER LED INDICATOR

Just beneath and to the right of the word "UNO" on your circuit board, there's a tiny LED next to the word 'ON' (11). This LED should light up whenever you plug your Arduino into a power source. If this light doesn't turn on, there's a good chance something is wrong. Time to re-check your circuit!

## TX RX LEDs

TX is short for transmit, RX is short for receive. These markings appear quite a bit in electronics to indicate the pins responsible for serial communication. In our case, there are two places on the Arduino UNO where TX and RX appear – once by digital pins 0 and 1, and a second time next to the TX and RX indicator LEDs (12). These LEDs will give us some nice visual indications whenever our Arduino is receiving or transmitting data (like when we're loading a new program onto the board).

## MAIN IC

The black thing with all the metal legs is an IC, or Integrated Circuit (13). Think of it as the brains of our Arduino. The main IC on the Arduino is slightly different from board type to board type, but is usually from the ATmega line of IC's from the ATMEL company. This can be important, as you may need to know the IC type (along with your board type) before loading up a new program from the Arduino software. This information can usually be found in writing on the top side of the IC. If you want to know more about the difference between various IC's, reading the datasheets is often a good idea.

## VOLTAGE REGULATOR

The voltage regulator (14) is not actually something you can (or should) interact with on the Arduino. But it is potentially useful to know that it is there and what it's for. The voltage regulator does exactly what it says – it controls the amount of voltage that is let into the Arduino board. Think of it as a kind of gatekeeper; it will turn away an extra voltage that might harm the circuit. Of course, it has its limits, so don't hook up your Arduino to anything greater than 20 volts.

# Beginner Level Projects

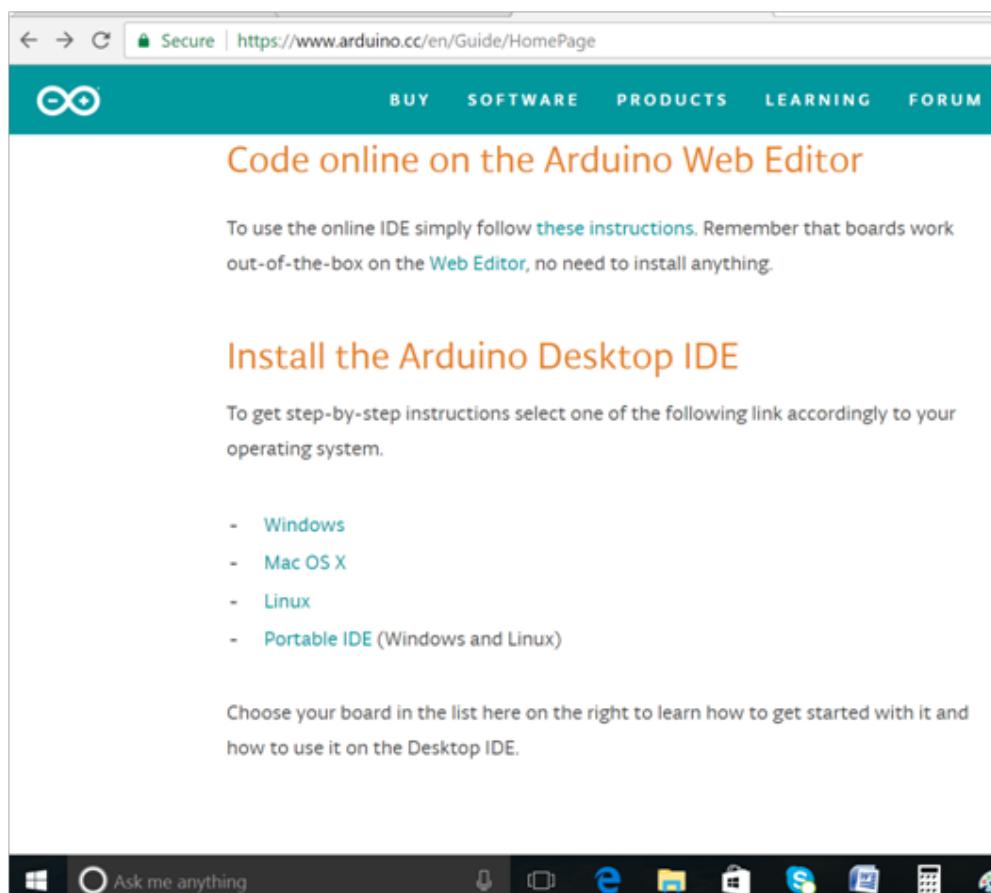
## PROJECT 1: INSTALLING ARDUINO IDE AND UPLOADING CODE

### STEP-1: DOWNLOAD THE ARDUINO IDE (INTEGRATED DEVELOPMENT ENVIRONMENT)

Access the Internet:

In order to get your Arduino up and running, you'll need to download some software first from [www.arduino.cc](https://www.arduino.cc) (it's free!). This software, known as the Arduino IDE, will allow you to program the Arduino to do exactly what you want. It's like a word processor for writing programs. With an internet-capable computer, open up your favorite browser and type in the following URL into the address bar:

[www.arduino.cc/en/Main/Software](https://www.arduino.cc/en/Main/Software)



The screenshot shows a web browser displaying the Arduino Software (IDE) installation guide for Windows. The page has a teal header with the Arduino logo and navigation links for BUY, SOFTWARE, PRODUCTS, LEARNING, FORUM, SUPPORT, and BLOG. A dropdown menu shows 'ENGLISH'. Below the header, a breadcrumb trail reads 'GETTING STARTED > Windows'. The main content title is 'Install the Arduino Software (IDE) on Windows PCs'. A sub-section title 'Download the Arduino Software (IDE)' is followed by a paragraph explaining the download options: 'Get the latest version from the [download page](#). You can choose between the installer (.exe) and the Zip packages. We suggest you use the first one that installs directly everything you need to use the Arduino Software (IDE), including the drivers. With the Zip package you need to install the drivers manually. The Zip file is also useful if you want to create a portable installation.' Below this, a note says 'When the download finishes, proceed with the installation and please allow the driver installation process when you get a warning from the operating system.' At the bottom of the page, there's a 'PREVIOUS' link.

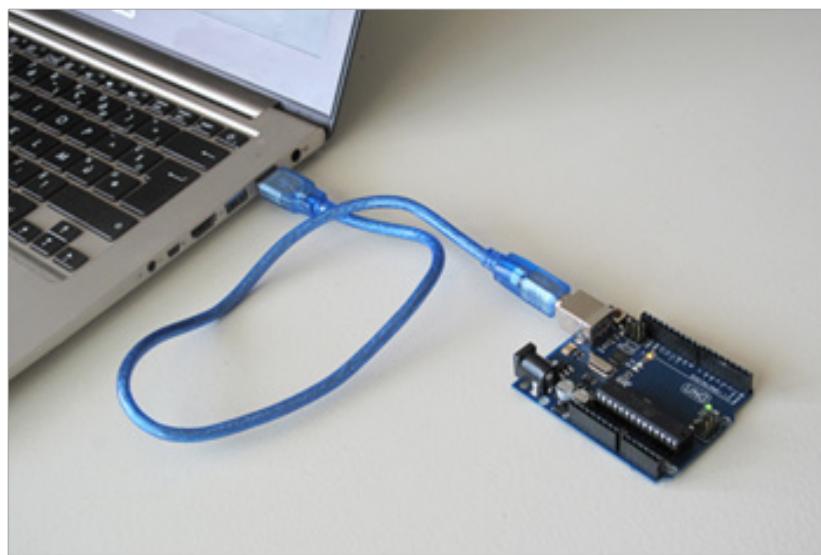
For different operating system platforms, the way of using Arduino IDE is different. Please refer to the following links: **Windows User:** <http://www.arduino.cc/en/Guide/Windows> **Mac OS X User:** <http://www.arduino.cc/en/Guide/MacOSX>

**Linux User:** <http://playground.arduino.cc/Learning/Linux>

For more detailed information about Arduino IDE, please refer to the following link: <http://www.arduino.cc/en/Guide/HomePage>

## STEP-2: CONNECT YOUR ARDUINO UNO TO YOUR COMPUTER

Use the USB cable provided in the kit to connect the Arduino to one of your computer's USB inputs.



# Beginner Level Projects

## STEP-3: INSTALL DRIVERS

Depending on your computer's operating system, you will need to follow specific instructions. Please go to the URLs below for specific instructions on how to install the drivers onto your Arduino Uno.



### Windows Installation Process:

Go to the web address below to access the instructions for installations on a Windows-based computer.

<http://arduino.cc/en/Guide/Windows>



### Macintosh OS X Installation Process:

Macs do not require you to install drivers. Enter the following URL if you have questions. Otherwise proceed to next page.

<http://arduino.cc/en/Guide/MacOSX>



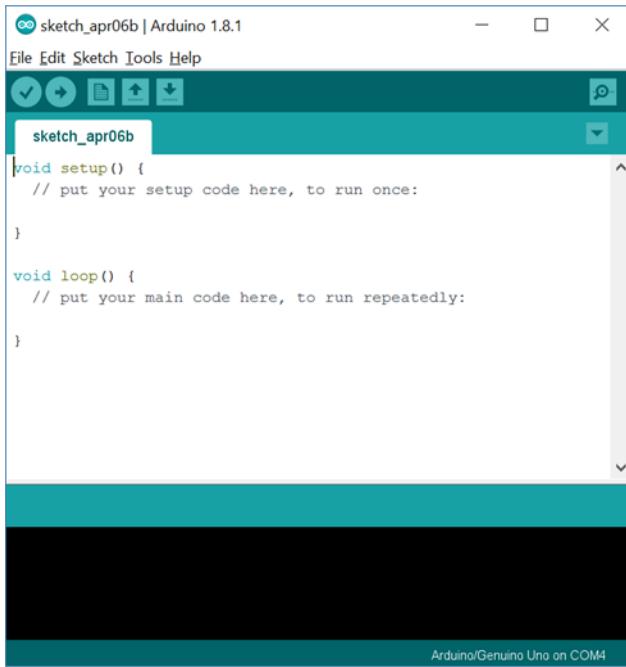
### Linux Installation Process:

32 bit / 64 bit, Installation Process Go to the web address below to access the instructions for installations on a Linux-based computer.

<http://www.arduino.cc/playground/Learning/Linux>

## STEP-4: OPEN THE ARDUINO IDE

Open the Arduino IDE software on your computer. Poke around and get to know the interface. We aren't going to code right away, this is just an introduction. The step is to set your IDE to identify your Arduino Uno.



### GUI (Graphical User Interface)



#### Verify

Checks your code for errors compiling it.



#### Upload

Compiles your code and uploads it to the configured board. See [uploading](#) below for details.

##### NOTE:

If you are using an external programmer with your board, you can hold down the "shift" key on your computer when using this icon. The text will change to "Upload using Programmer"



#### New

Creates a new sketch.



#### Open

Presents a menu of all the sketches in your sketchbook. Clicking one will open it within the current window overwriting its content.

##### NOTE:

due to a bug in Java, this menu doesn't scroll; if you need to open a sketch late in the list, use the File | Sketchbookmenu instead.



#### Save

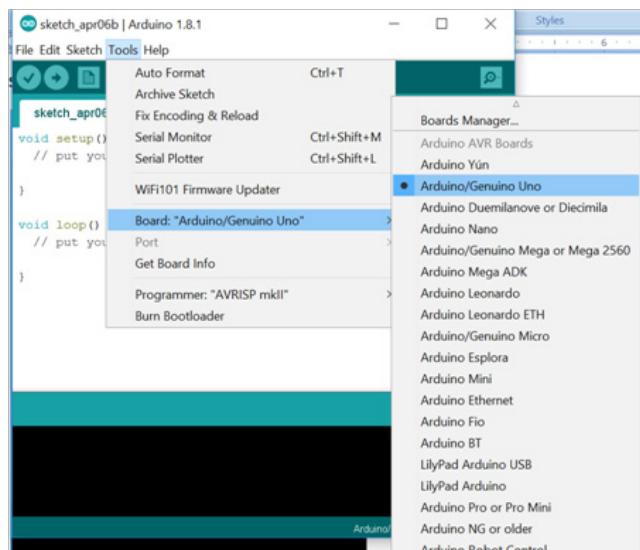
Saves your sketch.



#### Serial Monitor

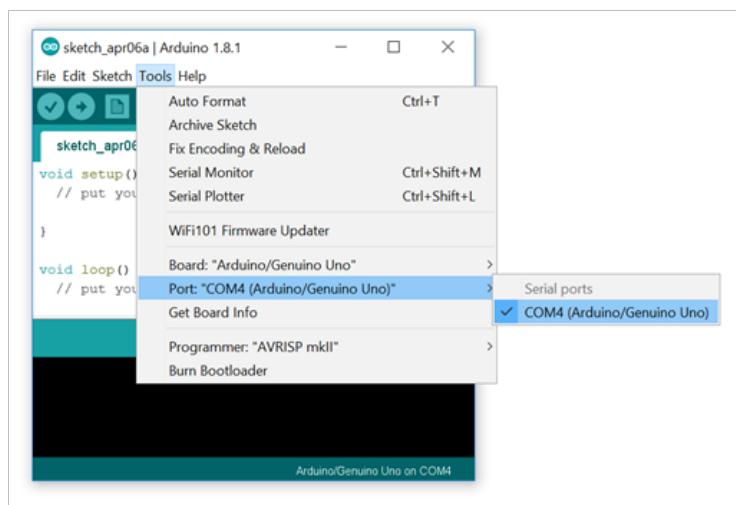
Opens the [serial monitor](#).

## STEP-5: SELECT YOUR BOARD: ARDUINO UNO



## STEP-6: SELECT YOUR SERIAL DEVICE

**Windows:** Select the serial device of the Arduino board from the Tools | Serial Port menu. This is likely to be com3 or higher (COM1 and COM2 are usually reserved for hardware serial ports). To find out, you can disconnect your Arduino board and re-open the menu; the entry that disappears should be the Arduino board. Reconnect the board and select that serial port.



**Mac OS:** Select the serial device of the Arduino board from the Tools > Serial Port menu. On the Mac, this should be something with /dev/tty.usbmodem (for the Uno or Mega 2560) or /dev/tty.usbserial (for older boards) in it.

**Linux:** <http://playground.arduino.cc/Learning/Linux>

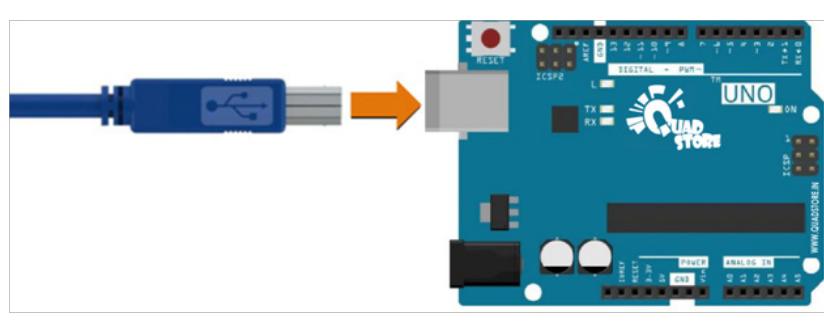


# Beginner Level Projects

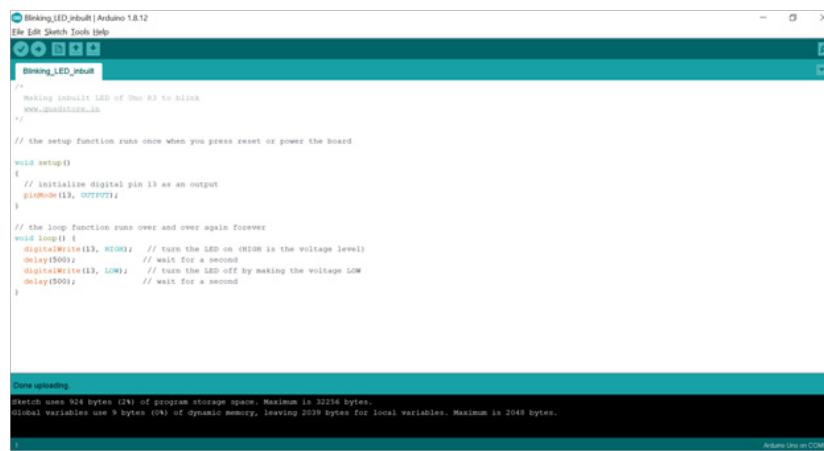
10

## HOW TO UPLOAD THE CODE TO UNO R3 BOARD ?

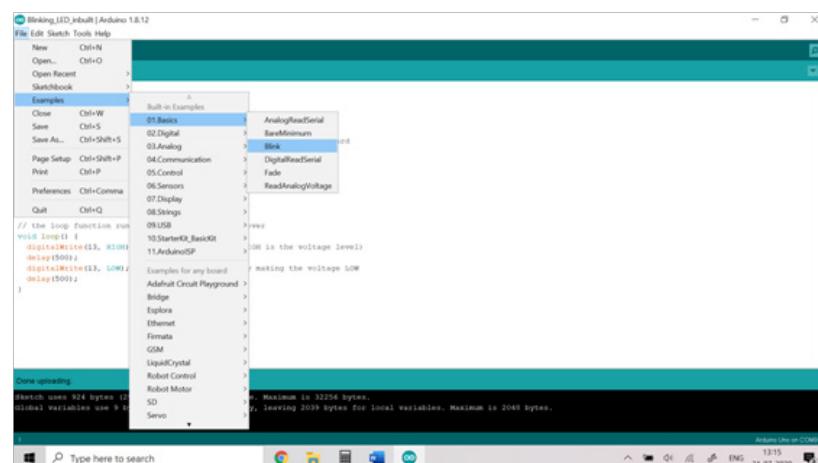
First connect your uno R3 board using usb cable to your laptop or computer.



If you already have arduino code in your systemfolders then just open it. Select the correct COM port for the bord. Then Click on verify button. Once the code is compiled without any error then click on upload button. Once the code is uploaded you will see a message at the bottom stating Done uploading. Now your code has been successfully uploaded.....



Alternatively, if you want to upload an example code you can go to File -> Examples -> and click on the example code you wish to upload.



## PROJECT 2: INSTALLING LIBRARIES

### WHAT ARE LIBRARIES?

Libraries are a collection of code that makes it easy for you to connect to a sensor, display, module, etc. For example, the built-in LiquidCrystal library makes it easy to talk to character LCD displays. There are hundreds of additional libraries available on the Internet for download. The built-in libraries and some of these additional libraries are listed in the reference. To use these additional libraries, you will need to install them.

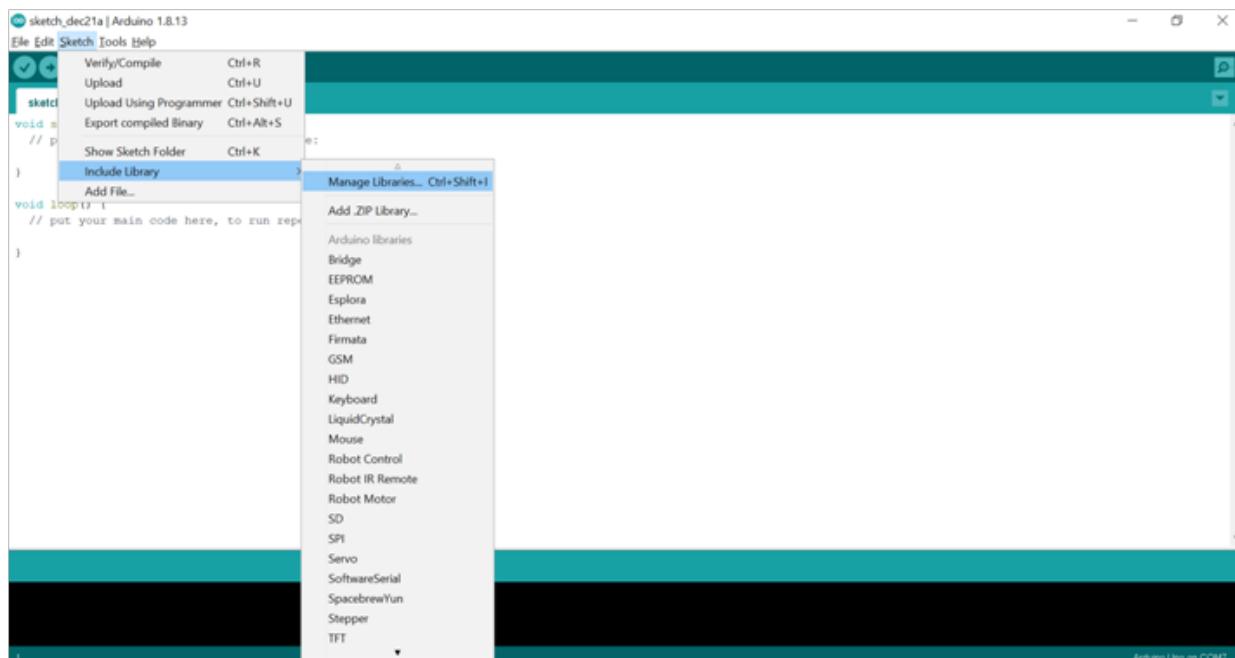
### HOW TO INSTALL LIBRARIES?

There are 3 different methods to install libraries, so let's see one by one.

#### Method- 1: Using library manager

To install a new library into your Arduino IDE, you can use the Library Manager.

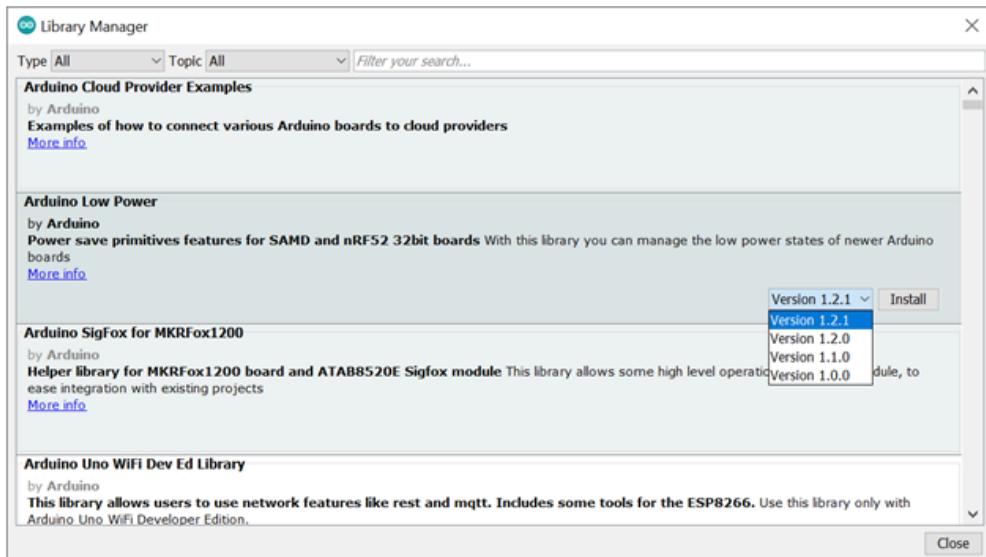
Open the IDE and click Sketch > Include > Library > Manage Libraries.



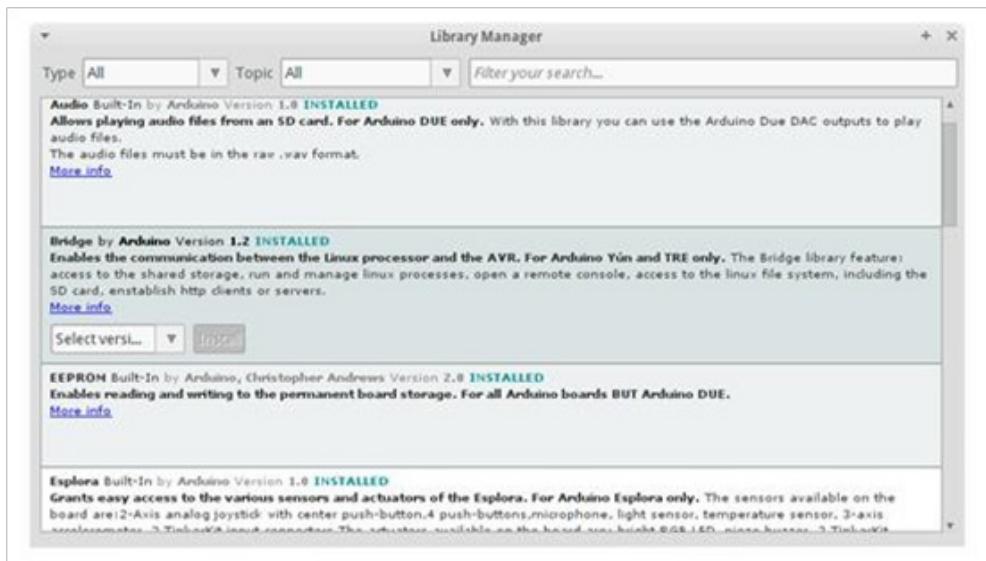
The library manager will open and you will find a list of libraries that are already installed or ready for installation. In this example, we will install the Bridge library. Scroll down the list to find it, then select the version of the library you want to install. Sometimes, only one version of the library is available. If the version selection menu does not appear, don't worry; it is normal.

# Beginner Level Projects

12



Finally click on install and wait for the IDE to install the new library. Downloading may take time depending on your connection speed. Once it has finished, an Installed tag should appear next to the Bridge library. You can close the library manager.

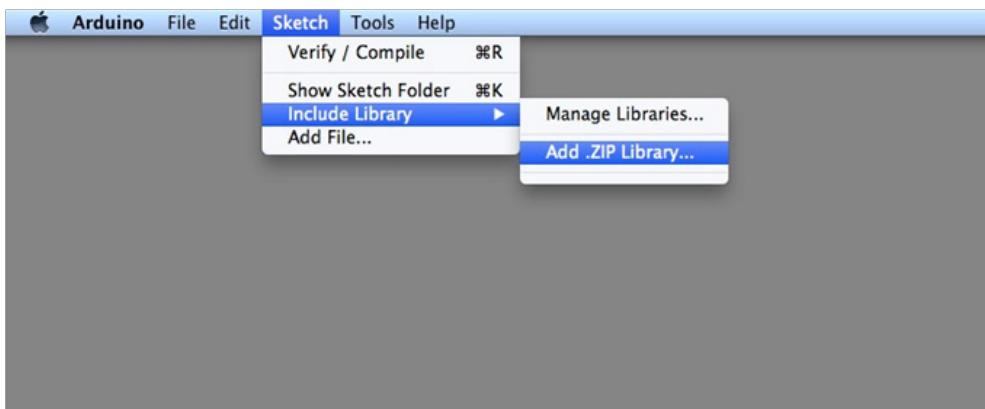


You can now find the new library available in the Include Library menu.

## Method-2: Importing a .zip library (Recommended for all the projects which we use)

Libraries are often distributed as a ZIP file or folder. The name of the folder is the name of the library. Inside the folder will be the following: .cpp file, .h file, often a keywords.txt file, examples folder, and other files required by the library. Starting with version 1.0.5, you can install third-party libraries in the IDE. Do not unzip the downloaded library; leave it as-is.

In the Arduino IDE, navigate to Sketch > Include Library > Add .ZIP Library.



You will be prompted to select the library you would like to add. Navigate to the .zip file's location and open it.

Return to the Sketch > Import Library menu. You should now see the library at the bottom of the dropdown menu. It is ready to be used in your sketch. The zip file will have been expanded in the libraries folder in your Arduino sketches directory.

NB: The Library will be available to use in sketches, but examples for the library will not be shown in the File > Examples until after the IDE has restarted.

### **Method-3: Manual Installation:**

To install the library, first, quit the Arduino application. Then unzip the ZIP file containing the library. For example, if you're installing a library called "ArduinoParty", uncompress ArduinoParty.zip. It should contain a folder called ArduinoParty, with files like ArduinoParty.cpp and ArduinoParty.h inside. (If the .cpp and .h files aren't in a folder, you'll need to create one. In this case, you'd make a folder called "ArduinoParty" and move into it all the files that were in the ZIP file, like ArduinoParty.cpp and ArduinoParty.h.)

Drag the ArduinoParty folder into this folder (your libraries folder). Under Windows, it will likely be called "My Documents\Arduino\libraries". For Mac users, it will likely be called "Documents/Arduino/libraries". On Linux, it will be the "libraries" folder in your sketchbook.

Your Arduino library folder should now look like this (on Windows):

[My Documents\Arduino\libraries\ArduinoParty\ArduinoParty.cpp](#)

[My Documents\Arduino\libraries\ArduinoParty\ArduinoParty.h](#)

[MyDocuments\Arduino\libraries\ArduinoParty\examples](#)

....

or like this (on Mac and Linux): [Documents/Arduino/libraries/ArduinoParty/ArduinoParty.cpp](#) [Documents/Arduino/libraries/ArduinoParty/ArduinoParty.h](#) [Documents/Arduino/libraries/ArduinoParty/examples](#)

....

There may be more files than just the .cpp and .h files so make sure they're all there. (The library won't work if you put the .cpp and .h files directly into the libraries folder or if they're nested in an extra folder. For example: [Documents\Arduino\libraries\ArduinoParty.cpp](#) and [Documents\Arduino\libraries\ArduinoParty\ArduinoParty\ArduinoParty.cpp](#) won't work.)

Restart the Arduino application. Make sure the new library appears in the Sketch > Import Library menu. That's it. You've installed a library!

## PROJECT 3: SERIAL MONITOR – SEND DATA

We will use the serial port on UNO R3 to send data to computer

### Parts Required:

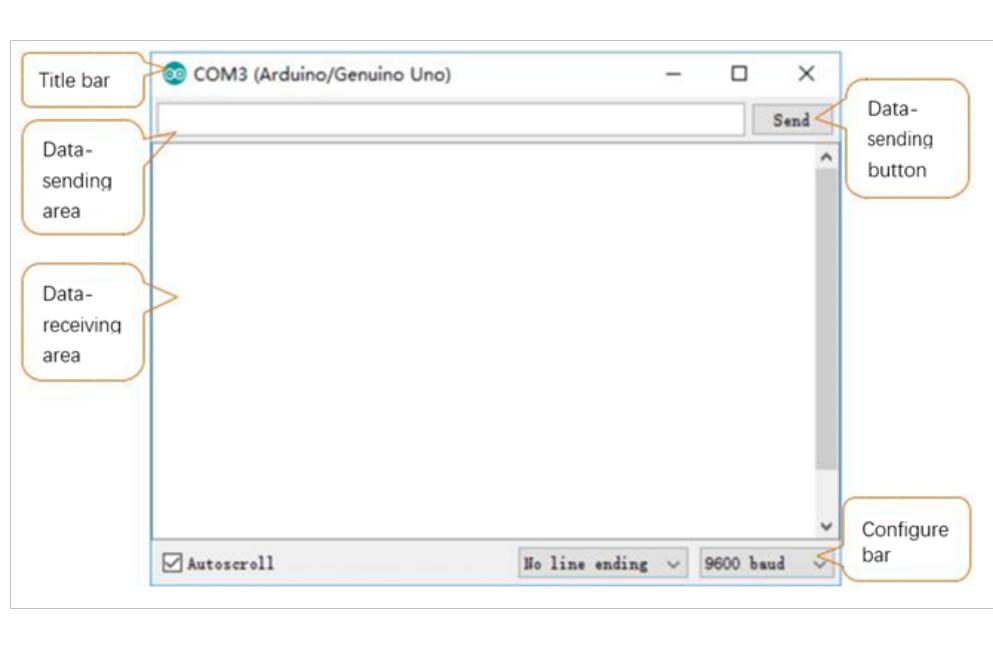
- Uno R3 board, USB Cable

### Serial port on Arduino

UNO has integrated USB to serial transfer, could communicates with computer when USB cable get connected to it. Arduino Software also uploads code to UNO through the serial connection. Computer identifies serial devices connected to your computer as COMx. We can use the Serial Monitor window of Arduino Software to communicate with UNO, connect UNO to computer through the USB cable, choose the right device, and then click the Serial Monitor icon to open the Serial Monitor window.

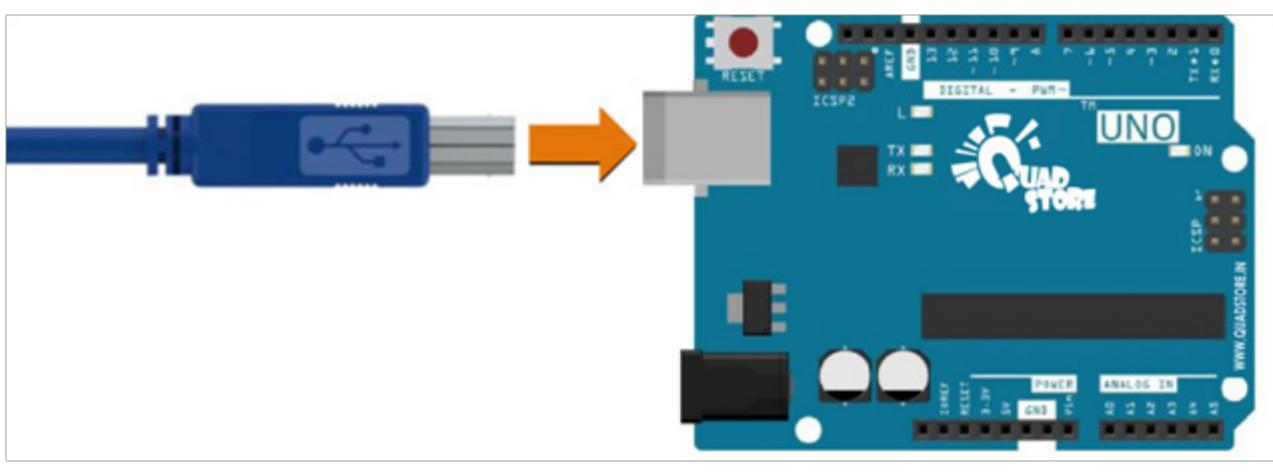


Interface of Serial Monitor window is as follows. If you can't open it, make sure UNO had been connected to the computer, and choose the right serial port in the menu bar "Tools".



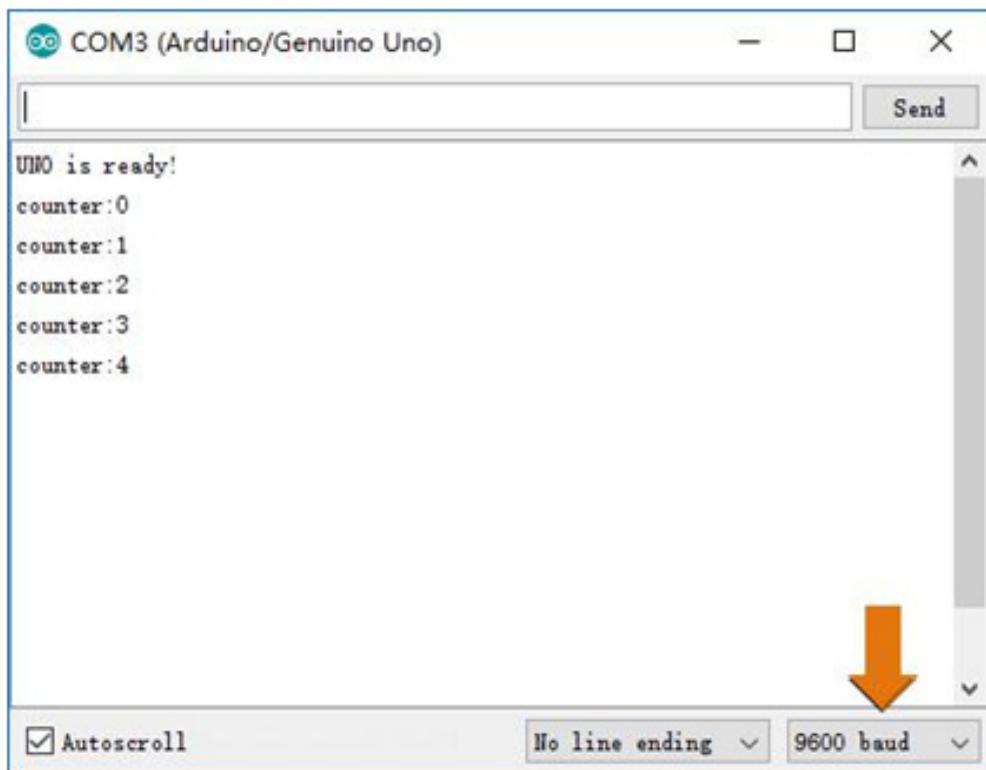
## Circuit Connection:

Connect UNO to the computer with USB cable.



**Code:** Verify and Upload the code "Serial\_Send\_Data.ino" to Uno R3 board

**Output:** Open the "Serial Monitor", then you'll see data sent from UNO r3.



## PROJECT 4: SERIAL MONITOR – RECEIVE DATA

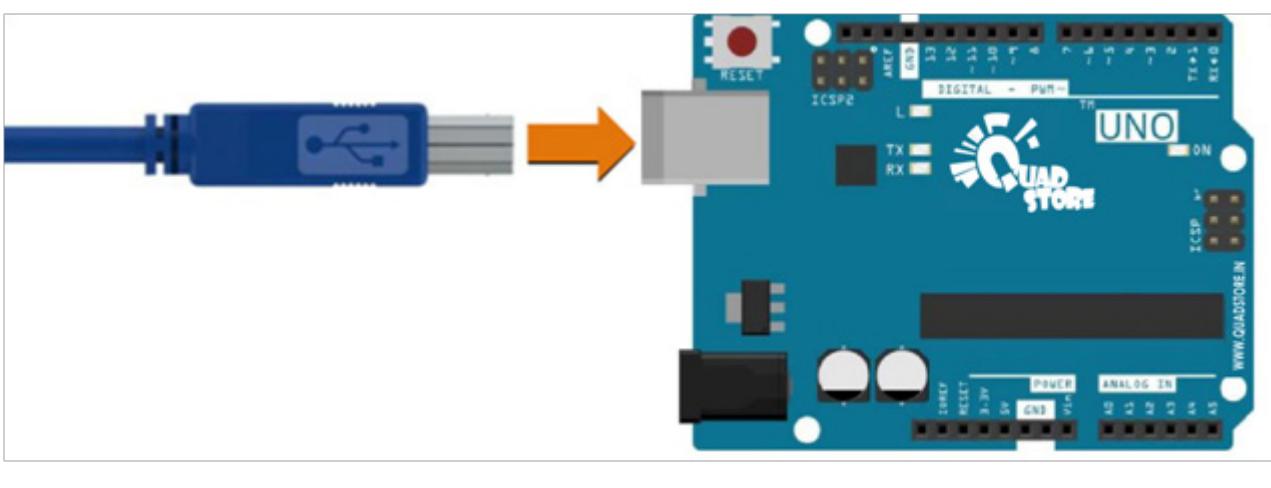
We will use the serial port to receive data from Uno R3.

### Parts Required:

- Uno R3 board, USB Cable

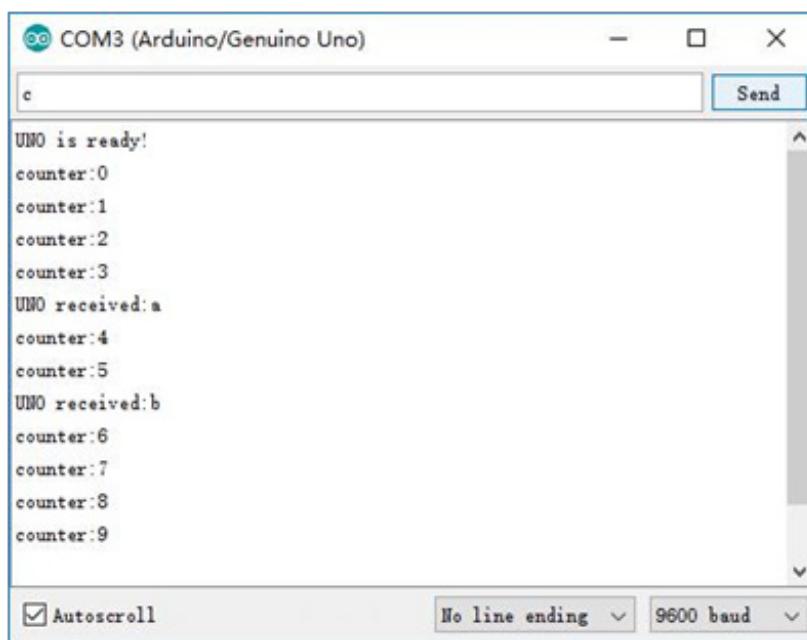
### Circuit Connection:

Connect UNO to the computer with USB cable



**Code:** Verify and Upload the code named “Serial\_Receive\_Data.ino” to Uno R3 board

**Output:** Open the “Serial Monitor”, then you’ll see the number constantly sent from UNO. Fill character in the sending area, and click the Send button, then you’ll see the string returned from UNO.



## PROJECT 5: BLINKING LED USING UNO R3 BOARDS BUILT-IN LED

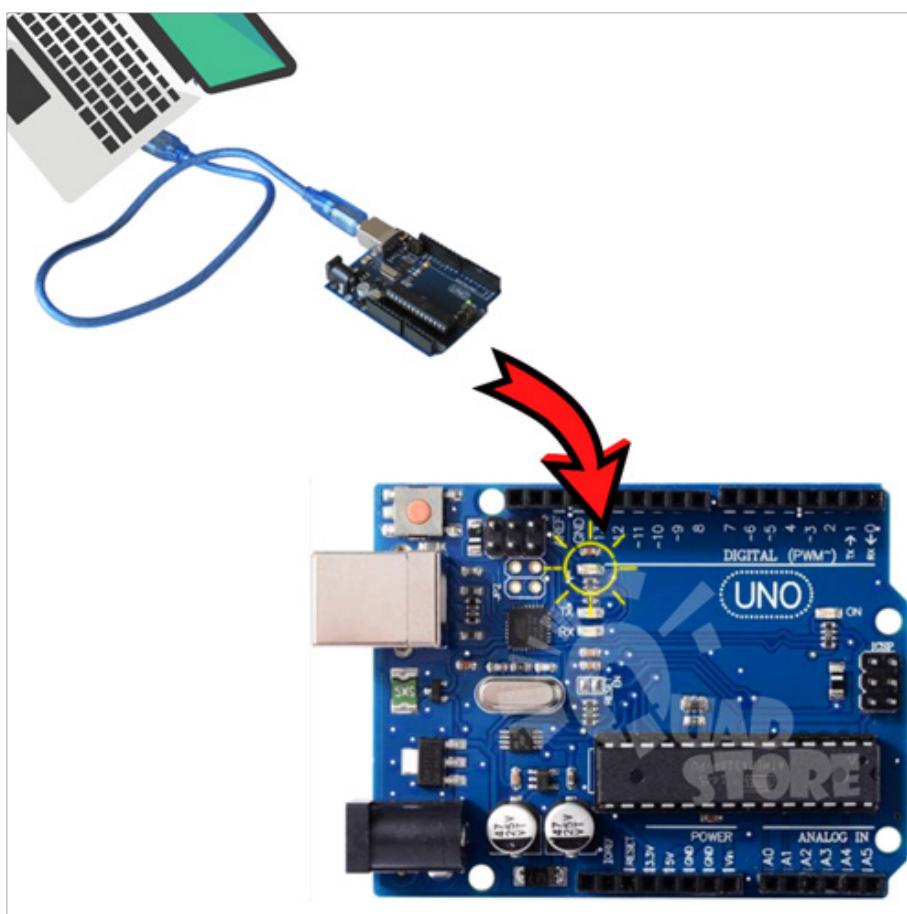
Make an inbuilt led below 13th pin blink.

### Parts Required:

- Uno R3 board, USB Cable

### Circuit Connection:

Connect the uno r3 board to the computer using the usb cable.



**Code:** Verify and Upload the code named “**Blinking\_LED\_inbuilt.ino**” to Uno R3 board.

**Output:** LED blinks continuously in certain interval.

## PROJECT 6: BLINKING LED USING AN EXTERNAL LED

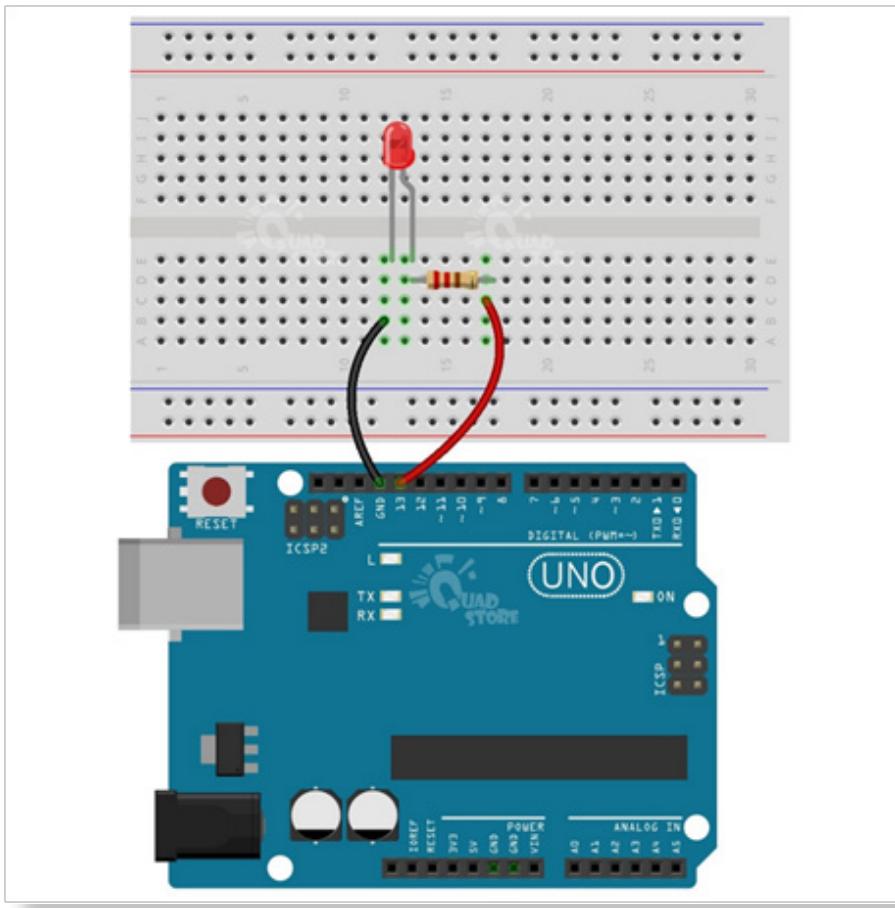
Make an external LED blink.

### Parts Required:

- \* Uno R3 board, USB Cable
- \* Breadboard
- \* Few jumper wires
- \* LED
- \* Resistor – 220ohm

### Circuit Connection:

Provide the connection as per the below circuit.



**Code:** Verify and Upload the code named “**Blinking\_LED\_external.ino**” to Uno R3 board

**Output:** LED blinks continuously in certain interval

## PROJECT 7: SMOOTH LED

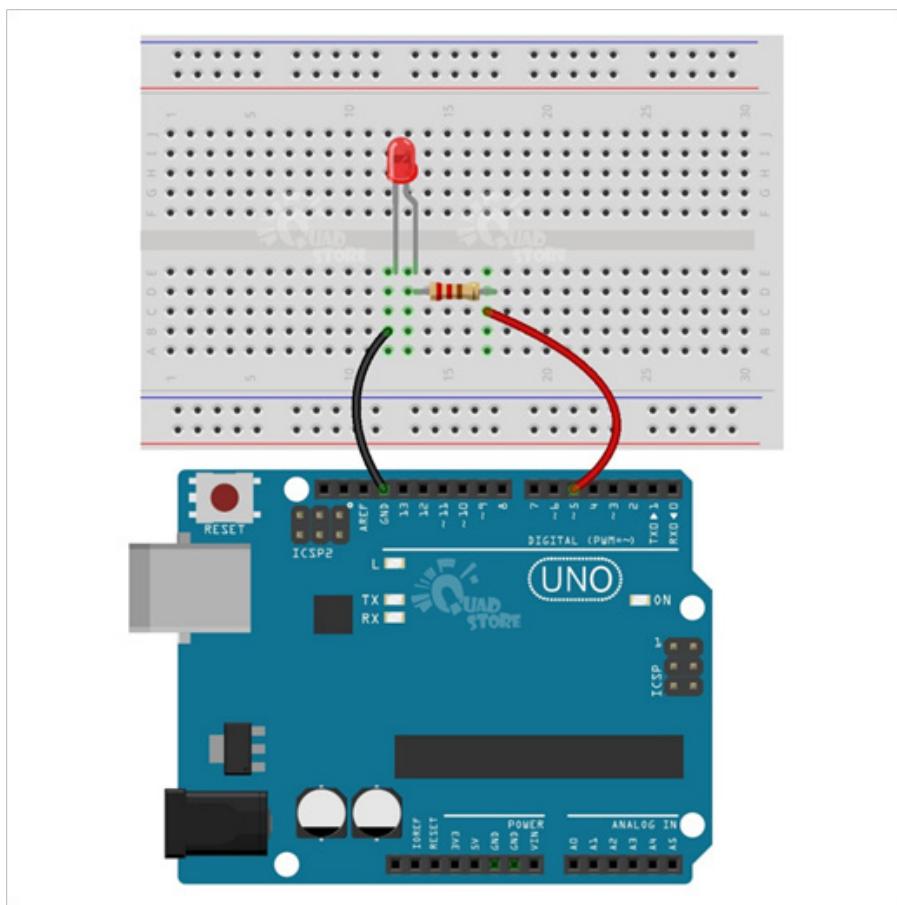
Make an LED blink smoothly.

### Parts Required:

- \* Uno R3 board, USB Cable
- \* Breadboard
- \* Few jumper wires
- \* LED
- \* Resistor – 220ohm

### Circuit Connection:

Provide the connection as per the below circuit.



**Code:** Verify and Upload the code named “**Blinking\_LED\_smoothly.ino**” to Uno R3 board

**Output:** LED will turn from low brightness to high and vice versa.

## PROJECT 8: SMOOTH LED

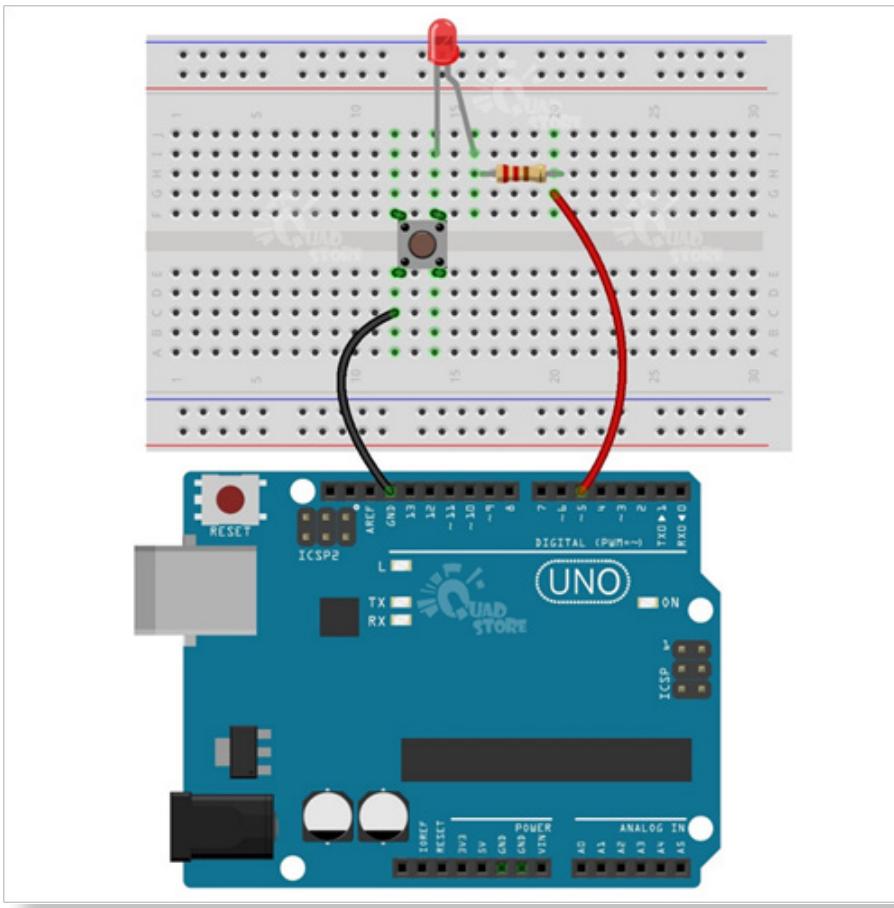
Turn the Led ON by pressing push button.

### Parts Required:

- \* Uno R3 board, USB Cable
- \* Breadboard
- \* Few jumper wires
- \* LED
- \* Resistor – 220ohm
- \* Push Button

### Circuit Connection:

Provide the connection as per the below circuit.



**Code:** Verify and Upload the code named “**PushButton.ino**” to Uno R3 board.

**Output:** Press the Push Button and you will see the LED glow.

## PROJECT 9: POTENTIOMETER

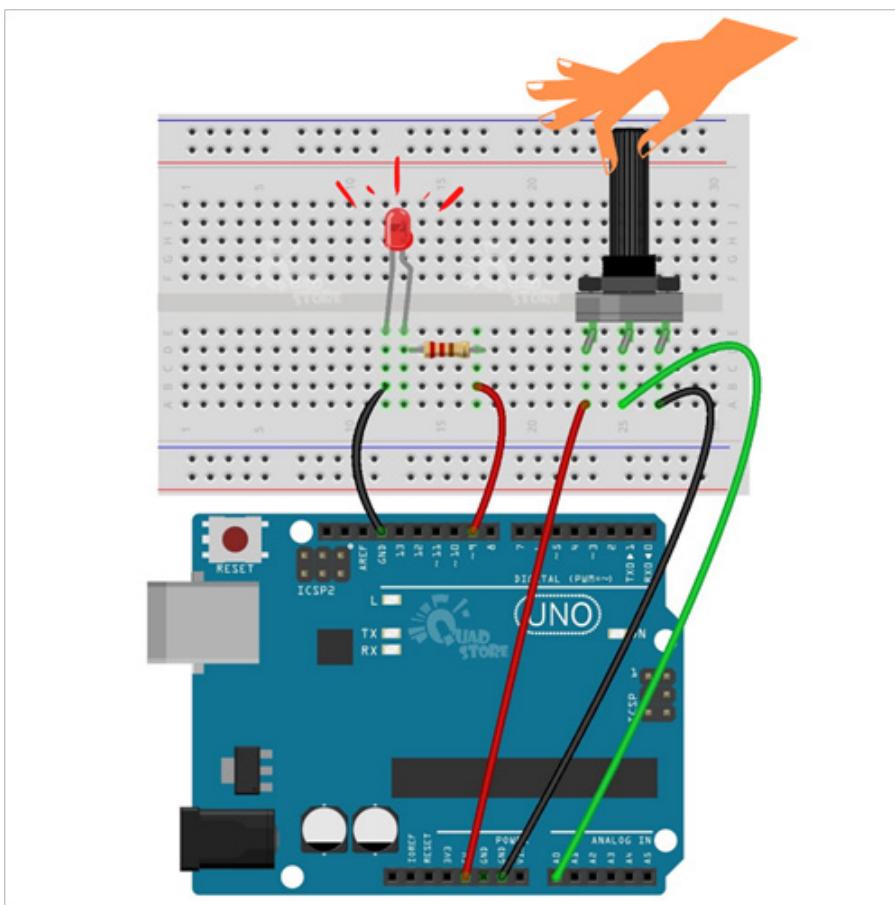
Adjust the brightness of Led using potentiometer.

### Parts Required:

- \* Uno R3 board, USB Cable
- \* Breadboard
- \* Few jumper wires
- \* LED
- \* Resistor – 220ohm
- \* Potentiometer

### Circuit Connection:

Provide the connection as per the below circuit.



**Code:** Verify and Upload the code named “**Potentiometer.ino**” to Uno R3 board

**Output:** Rotate the potentiometer from left to right to see the brightness of the LED changing.

## PROJECT 10: PHOTORESISTOR

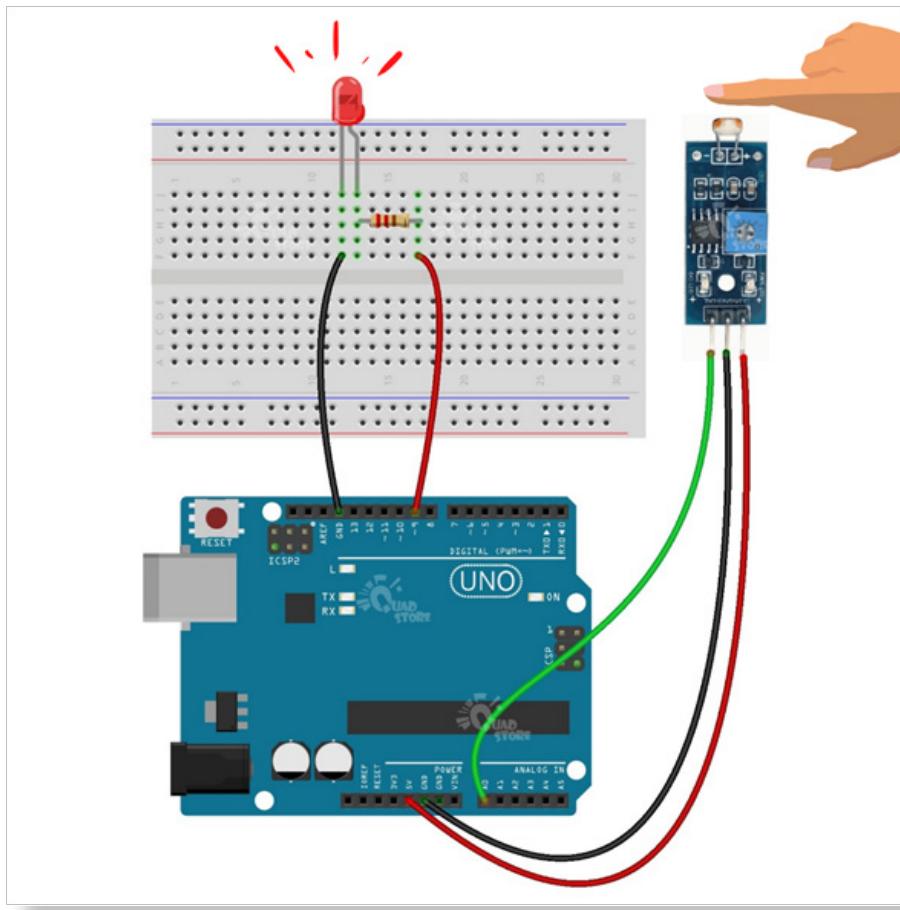
Adjust the brightness of Led using photoresistor.

### Parts Required:

- \* Uno R3 board, USB Cable
- \* Breadboard
- \* Few jumper wires
- \* LED
- \* Resistor – 220ohm for LED
- \* 3 pin Photoresistor

### Circuit Connection:

Provide the connection as per the below circuit.



**Code:** Verify and Upload the code named “**Photoresistor.ino**” to Uno R3 board

**Output:** Close the photoresistor with your finger to see the intensity of the LED increase.

## PROJECT 11: RGB LED

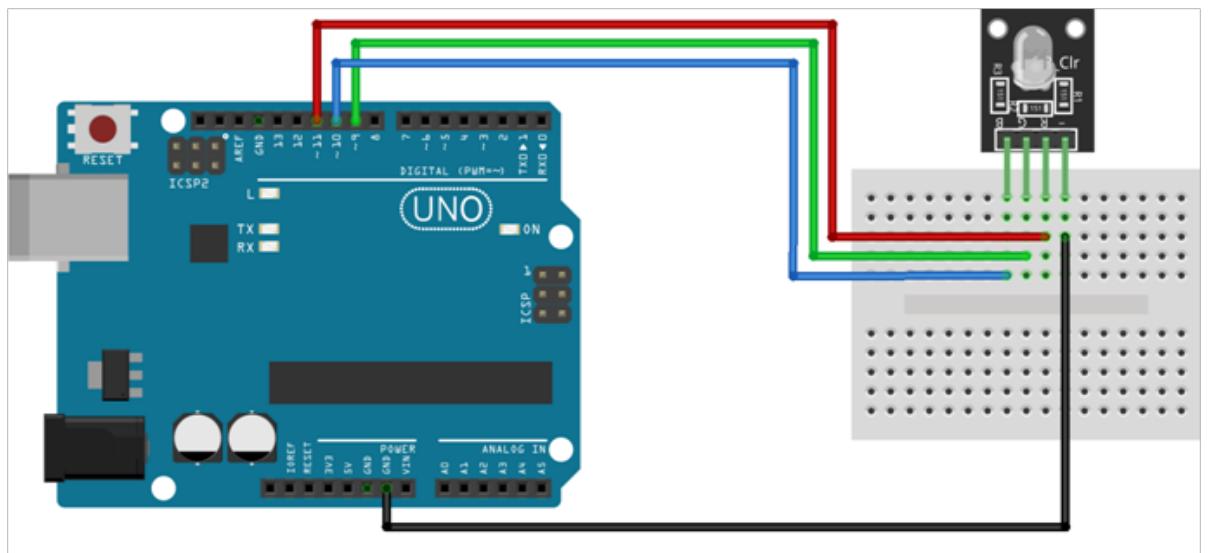
Turn on the RGB led to see different colors pop up.

### Parts Required:

- \* Uno R3 board, USB Cable
- \* Breadboard
- \* Few jumper wires
- \* RGB LED Module

### Circuit Connection:

Provide the connection as per the below circuit.



**Code:** Verify and Upload the code named “**RGB\_LED.ino**” to Uno R3 board

**Output:** Color of the LED changes from RED to GREEN to BLUE in cycles.

## PROJECT 12: ACTIVE BUZZER

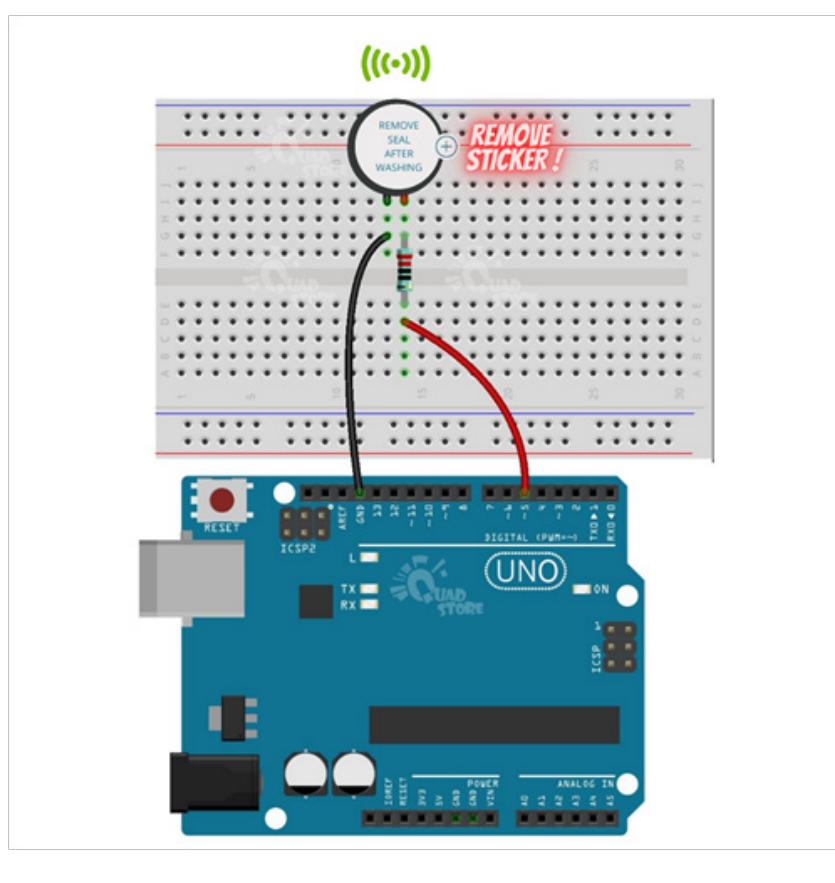
Make the buzzer beep in certain interval.

### Parts Required:

- \* Uno R3 board, USB Cable
- \* Breadboard
- \* Few jumper wires
- \* Active Buzzer (Identification: Has a small sticker on top of it. Remove this sticker before use)
- \* Resistor – 220ohm

### Circuit Connection:

Provide the connection as per the below circuit.



**Code:** Verify and Upload the code named “**Active\_Buzzer.ino**” to Uno R3 board

**Output:** Active Buzzer will beep continuously in certain interval.

# **PROJECT 13:**

## **PASSIVE BUZZER OR TONE GENERATOR**

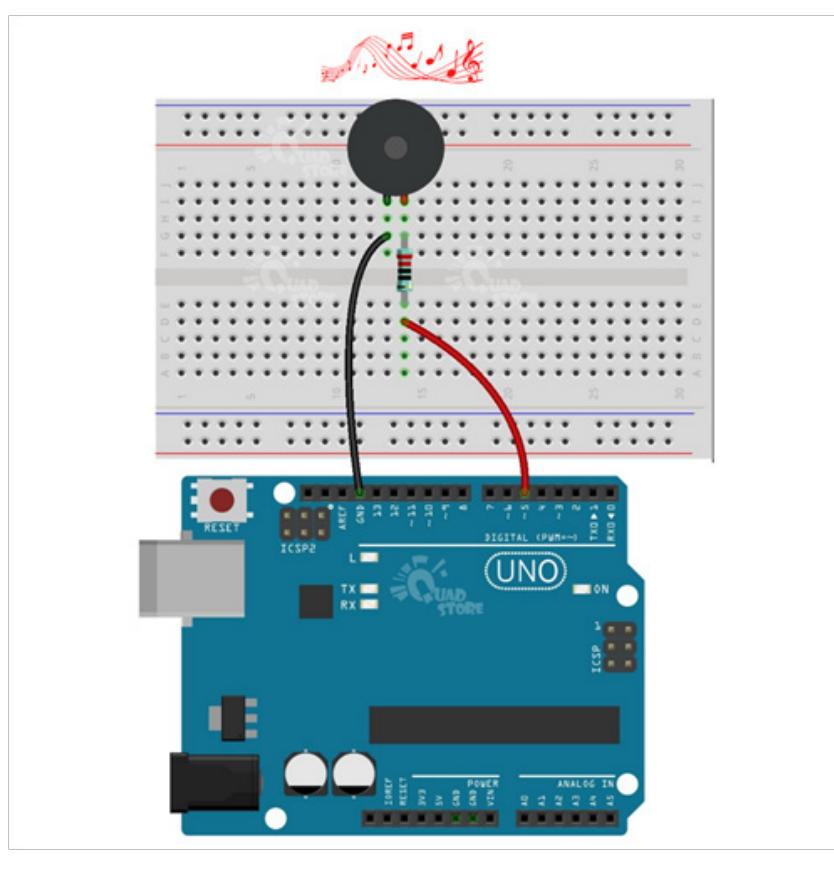
Hear the tone/music through passive buzzer.

## **Parts Required:**

- \* Uno R3 board, USB Cable
  - \* Breadboard
  - \* Few jumper wires
  - \* Passive Buzzer or Tone Generator
  - \* Resistor – 220ohm

## Circuit Connection:

Provide the connection as per the below circuit.



**Code:** Verify and Upload the code named “**Passive\_Buzzer.ino**” to Uno R3 board  
**Output:** Produces musical tones.

## PROJECT 14: PASSIVE BUZZER OR TONE GENERATOR

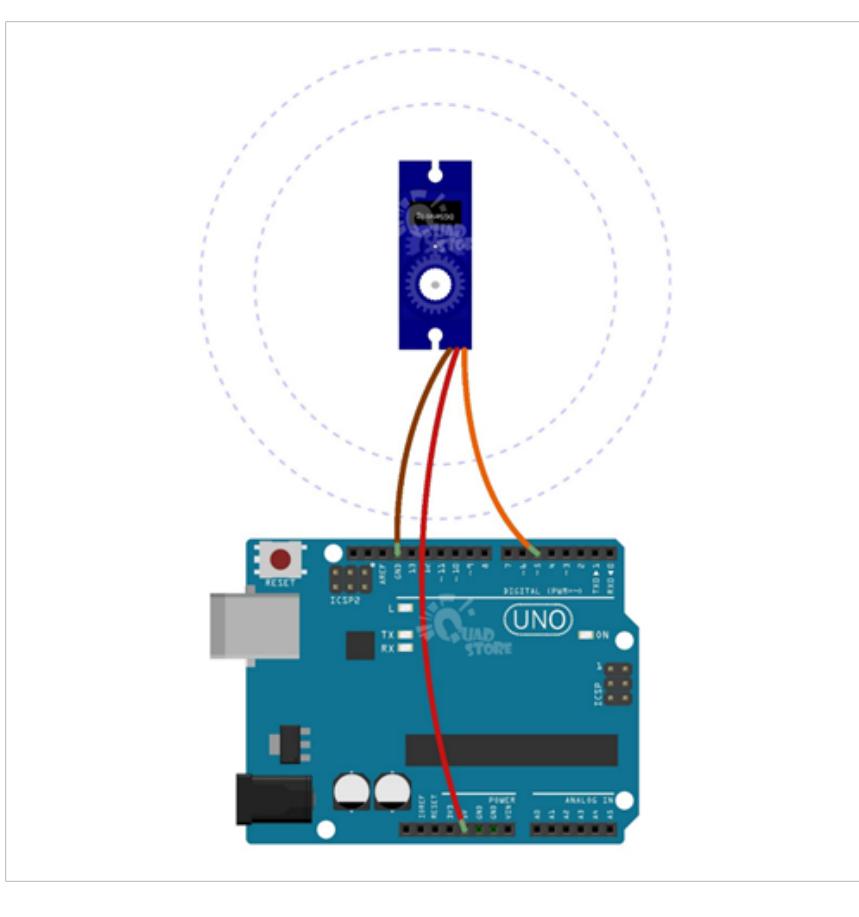
Make the servo rotate from 0 to 180 degrees and vice versa.

### Parts Required:

- \* Uno R3 board, USB Cable
- \* Male to Male Jumper Wires
- \* Servo Motor

### Circuit Connection:

Provide the connection as per the below circuit.



**Code:** Verify and Upload the code named “**Servo.ino**” to Uno R3 board

**Output:** Servo arm will rotate from 0 to 180 degrees and comes back to its original position

## PROJECT 15: SERVO CONTROL THROUGH POTENTIOMETER

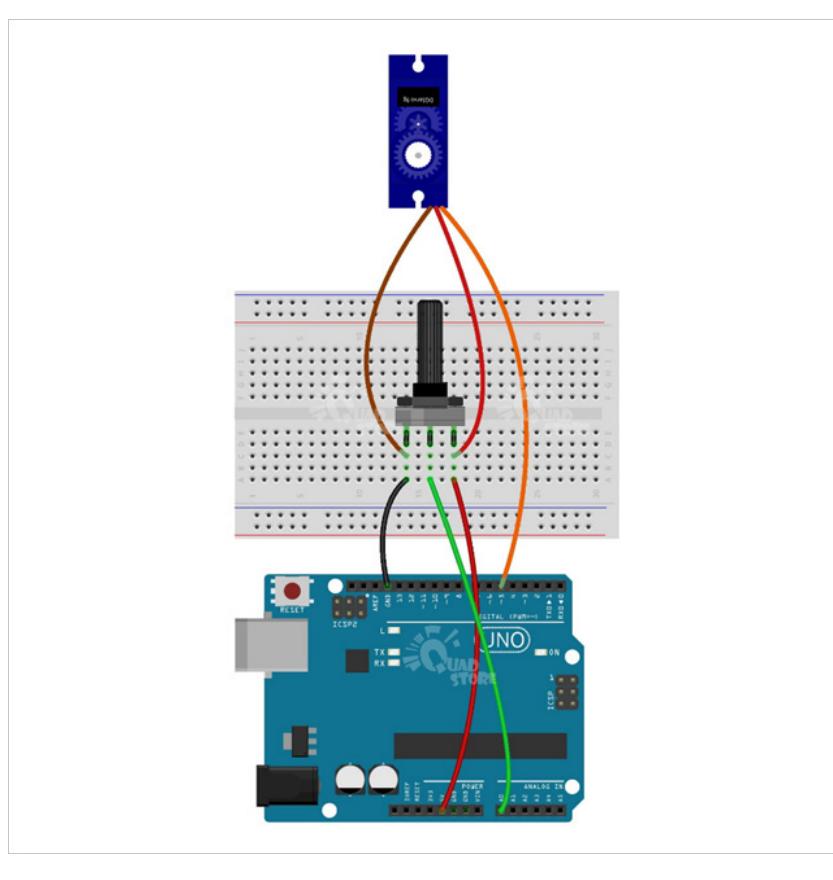
Control the rotation of servo using potentiometer.

### Parts Required:

- \* Uno R3 board, USB Cable
- \* Breadboard
- \* Male to Male Jumper Wires
- \* Servo Motor
- \* Potentiometer 10k

### Circuit Connection:

Provide the connection as per the below circuit.



**Code:** Verify and Upload the code named “**Servo\_Potentiometer.ino**” to Uno R3 board

**Output:** Rotate the potentiometer from left to right and see the rotation of the servo arm based on the rotation of potentiometer.

# Intermediate Level Projects

## PROJECT 16:

### DHT TEMPERATURE SENSOR USING 3 PINS

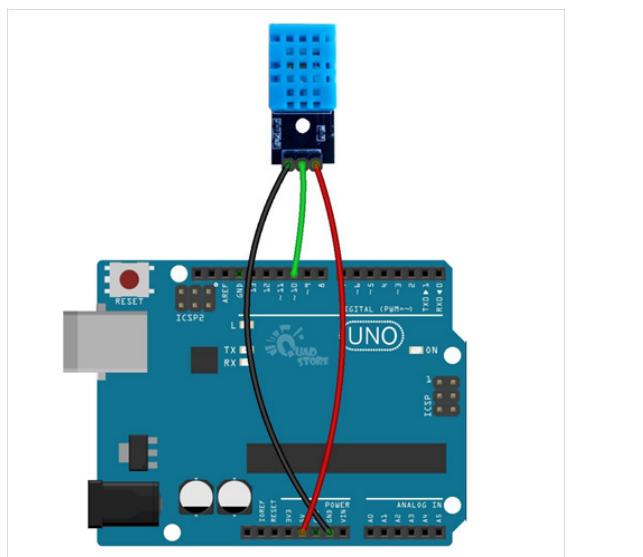
Display current temperature and humidity in serial monitor.

#### Parts Required:

- \* Uno R3 board, USB Cable
- \* Few jumper wires
- \* 3pin DHT11 Sensor

#### Circuit Connection:

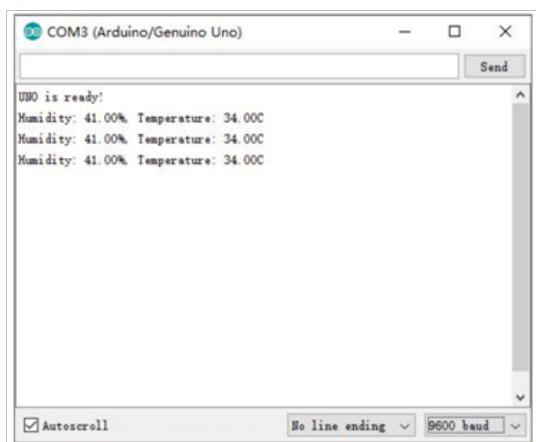
Provide the connection as per the below circuit.



**Library:** Install DHT.zip library before uploading the code. Otherwise your code will give error message.

**Code:** Verify and Upload the code named “DHT11\_TempSensor\_3pin.ino” to Uno R3 board.

**Output:** Open “Serial Monitor” to see the current temperature and humidity displayed.



## PROJECT 17: 1-DIGIT 7 SEGMENT DISPLAY

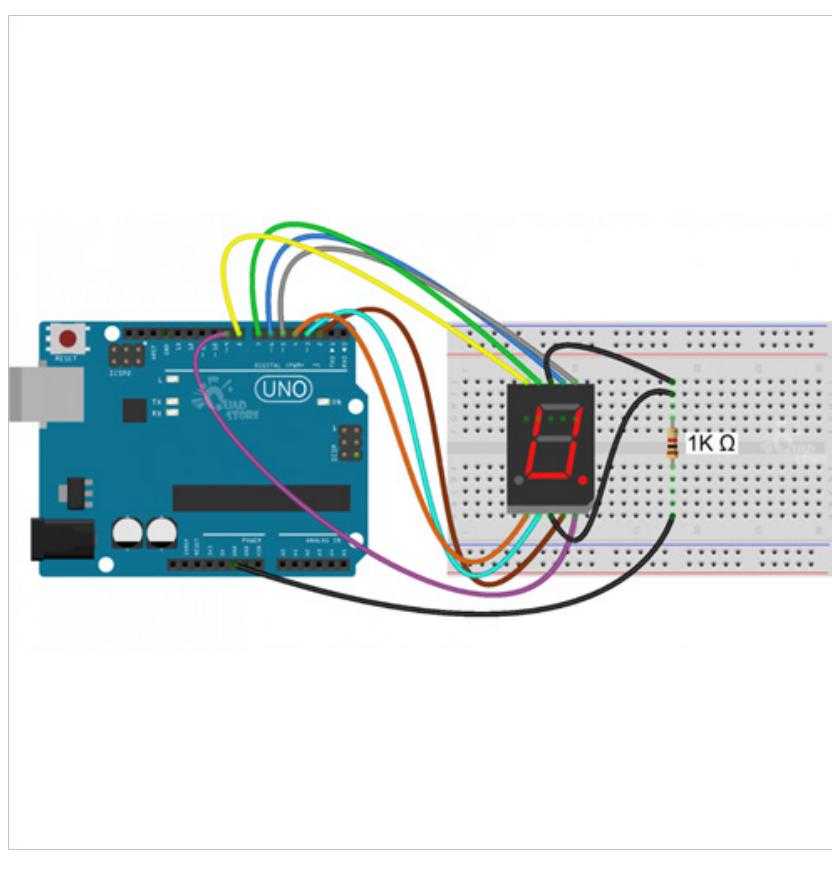
Display numbers from 0 to 9 in the 1-digit 7 Segment display.

### Parts Required:

- \* Uno R3 board, USB Cable
- \* Breadboard
- \* Few jumper wires
- \* Resistor – 1Kohm
- \* 1-digit 7 Segment Display

### Circuit Connection:

Provide the connection as per the below circuit.



**Library:** Install SevSeg.zip library before uploading the code. Otherwise your code will give error message.

**Code:** Verify and Upload the code named “**1Digit\_7Segment.ino**” to Uno R3 board

**Output:** You can see the numbers displayed from 0 to 9.

## PROJECT 18: I2c 1602 LCD DISPLAY

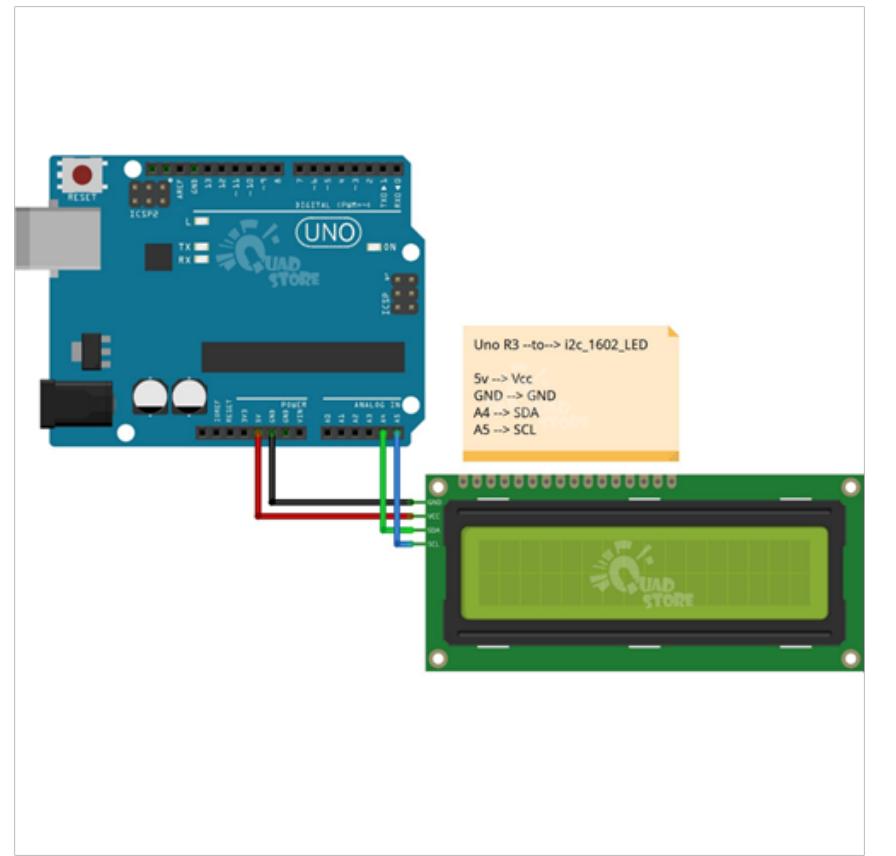
Display counter number in 1602 display.

### Parts Required:

- \* Uno R3 board, USB Cable
- \* Few jumper wires
- \* I2c 1602 LCD display

### Circuit Connection:

Provide the connection as per the below circuit.



**Library:** Install Newliquidcrystal\_1.3.5 library before uploading the code. Otherwise your code will give error message.

**Code:** Verify and Upload the code named "**i2c\_1602\_LCDdisplay.ino**" to Uno R3 board

**Output:** You will see LCD display counters with increase in number for every blink.

## PROJECT 19:

### ULTRASONIC SENSOR HC-SR04

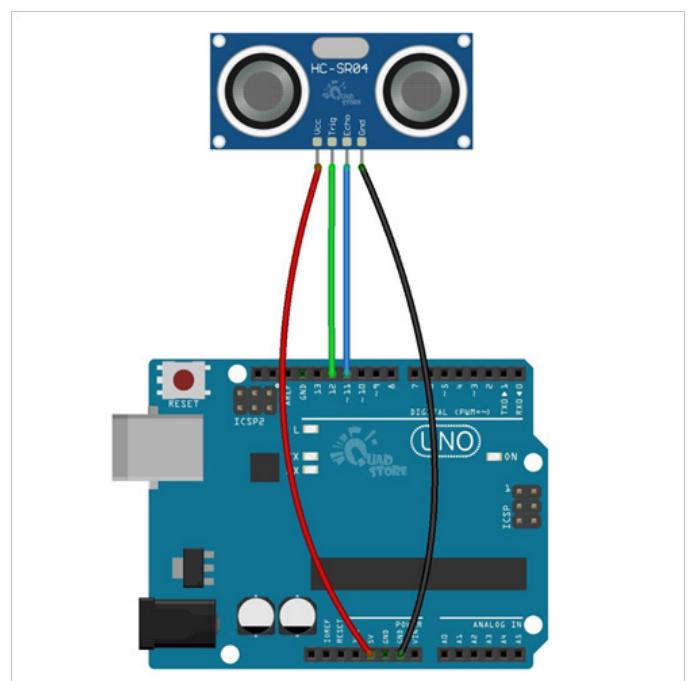
Display the distance of object in serial monitor.

#### Parts Required:

- \* Uno R3 board, USB Cable
- \* Few jumper wires
- \* Ultrasonic Sensor HC-SR04

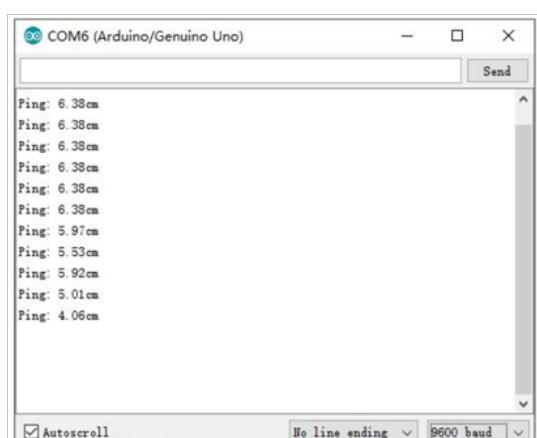
#### Circuit Connection:

Provide the connection as per the below circuit.



**Code:** Verify and Upload the code named “**Ultrasonic\_Sensor.ino**” to Uno R3 board

**Output:** Open “Serial Monitor”. Place any object in front of the ultrasonic sensor and move the object closer and further to see the distance displayed in serial monitor.



## PROJECT 20: 1 CHANNEL RELAY

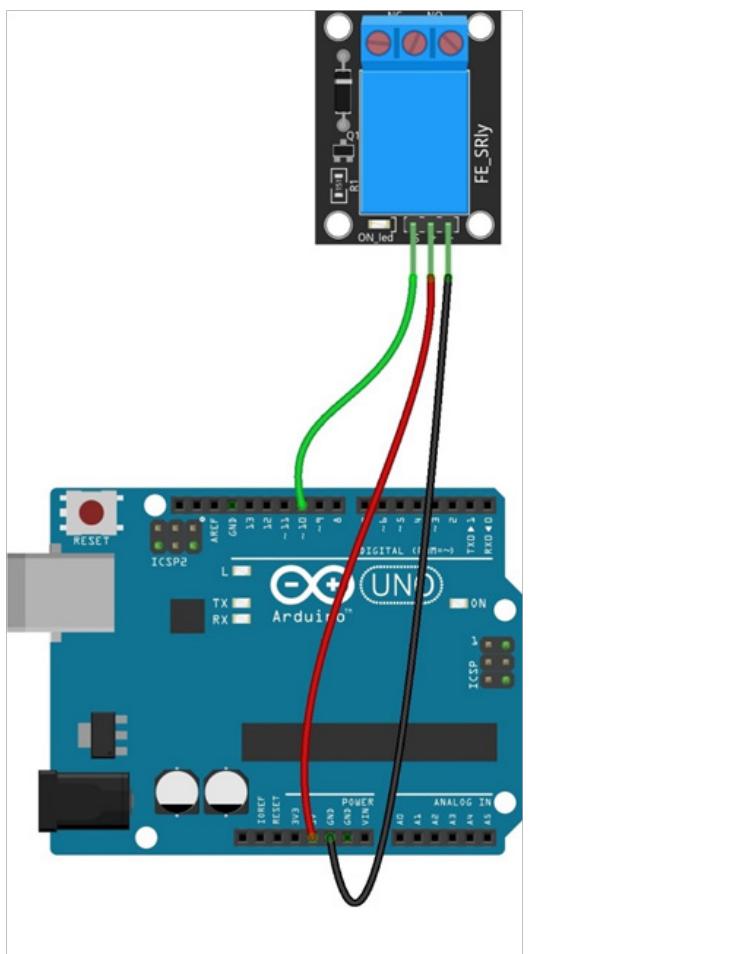
Turn on the LED and DC motor using relay module.

### Parts Required:

- \* Uno R3 board, USB Cable
- \* Breadboard
- \* Few jumper wires
- \* 1 Channel Relay.

### Circuit Connection:

Provide the connection as per the below circuit.



**Code:** Verify and Upload the code named “**Relay.ino**” to Uno R3 board

**Output:** The relay will switch ON and OFF in certain interval continuously.

## PROJECT 21: BLUETOOTH HC-05 OR HC-06

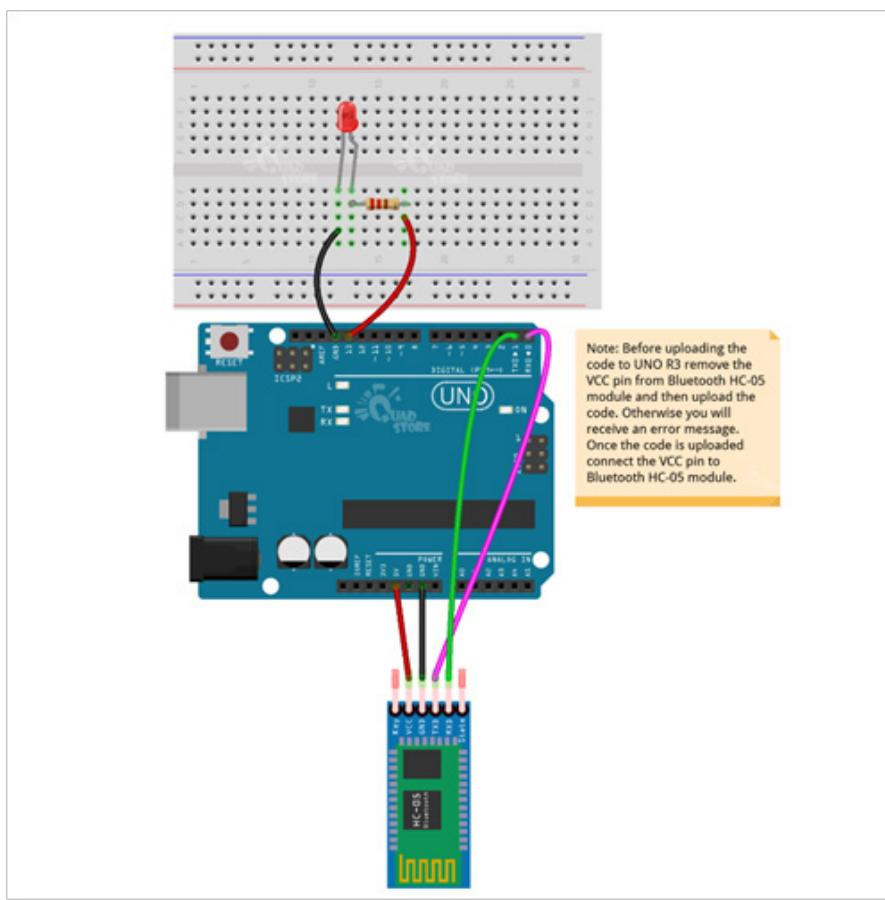
Control LED turn ON and OFF using Bluetooth mobile app.

### Parts Required:

- \* Uno R3 board, USB Cable
- \* Breadboard
- \* Few jumper wires
- \* Bluetooth HC-05 module(or) HC-06 module
- \* LED
- \* Resistor – 220ohm
- \* Install “Arduino Bluetooth Controller” App from Playstore

### Circuit Connection:

Provide the connection as per the below circuit. Connection is same for both HC-05 or HC-06 modules.



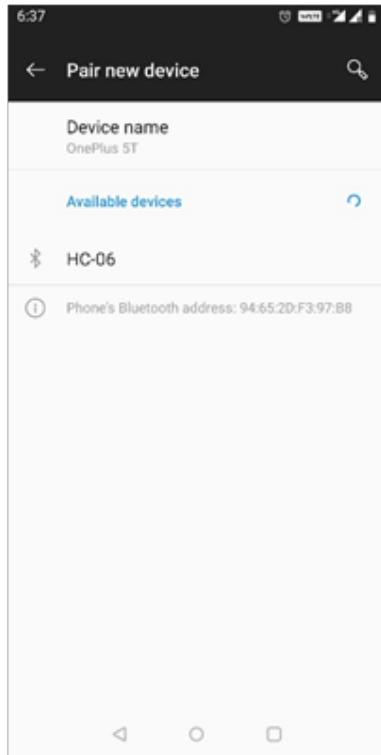
Follow the below steps to perform proper connection between phones Bluetooth app and HC-05/06 module.

1. Before uploading the code, you need to remove the VCC pin from Bluetooth HC-05/06 module.
2. **Code:** Verify and Upload the code named “**Bluetooth\_HC05.ino**” to Uno R3 board

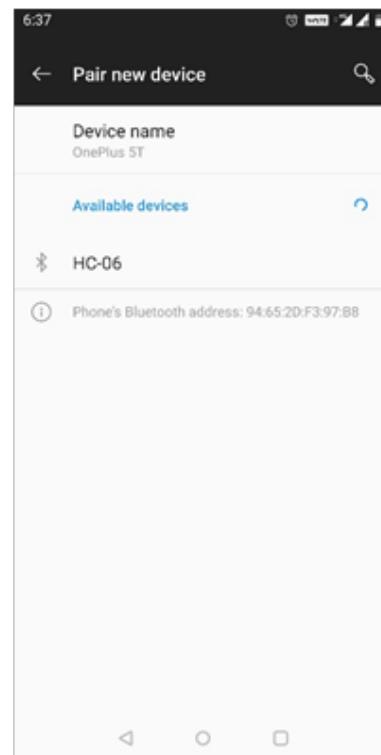
# Intermediate Level Projects

34

3. Connect the VCC pin back to Bluetooth HC-05/06 module.
4. Enable Bluetooth in your mobile setting and click on Pair new device.
5. You should see the device name HC-05 or HC-06 based on your bluetooth module.

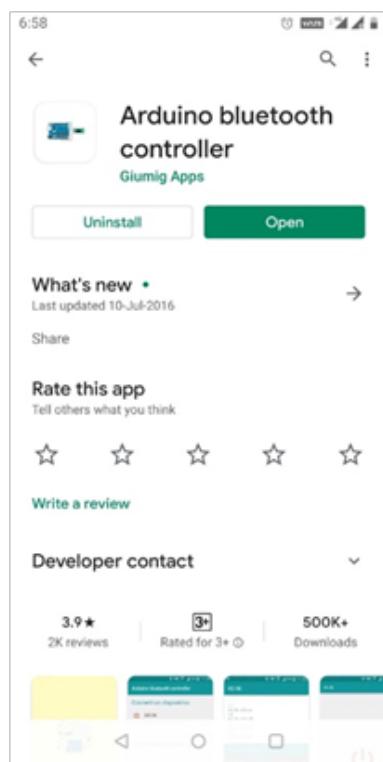


6. Click on HC-05 or HC-06 and it will prompt for password.
7. Enter the password as 1234 or 0000 and click OK



# Intermediate Level Projects

8. Device will be successfully paired with your mobile. But still you will see the led in the bluetooth HC-05 or HC-06 module blinking.
9. Install the app named “Arduino bluetooth Controller” from Playstore



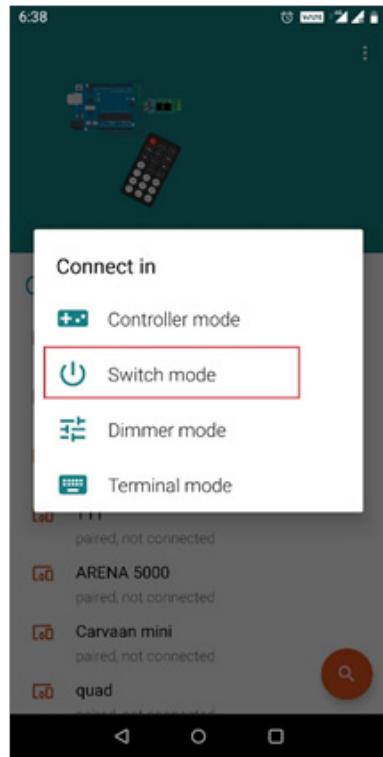
10. Open the app and under Connect to a device select HC-05 or HC-06



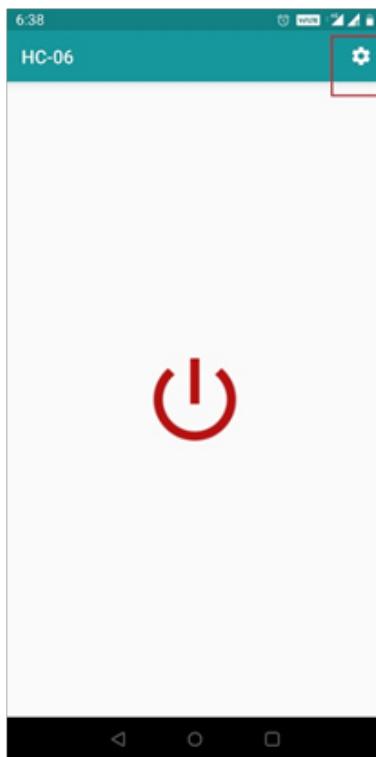
# Intermediate Level Projects

36

11. Connect in "Switch Mode" option

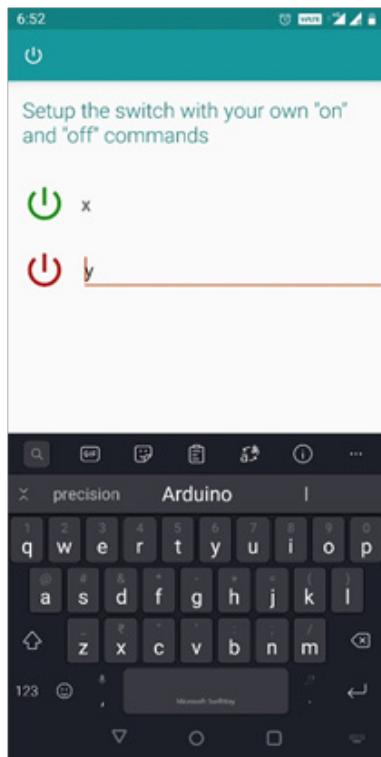


12. Click on Setting icon to configure the button options.

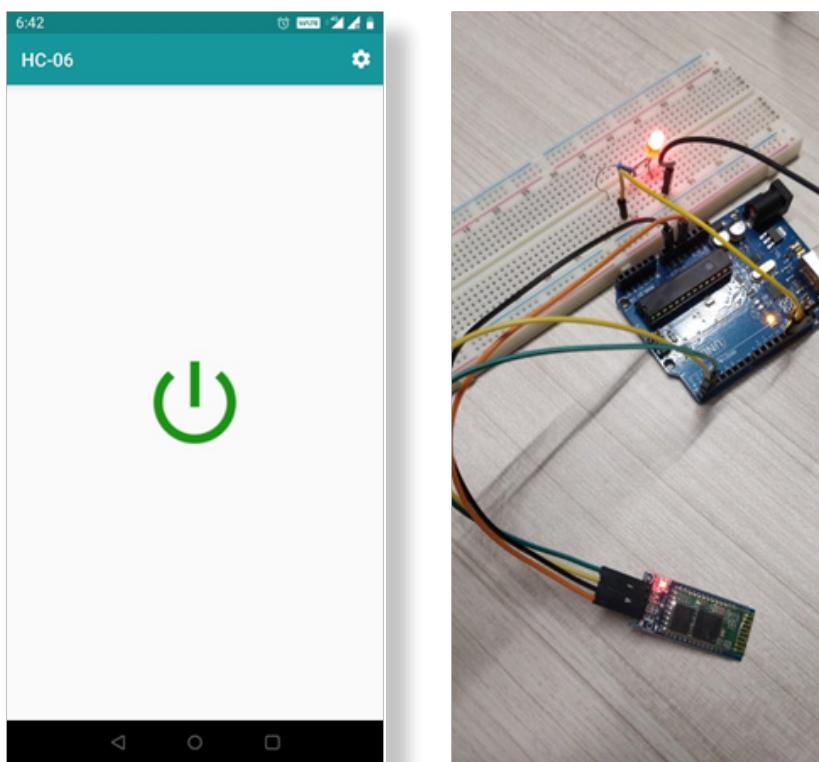


# Intermediate Level Projects

13. Enter x for ON button and y for OFF button and then go back to previous screen.



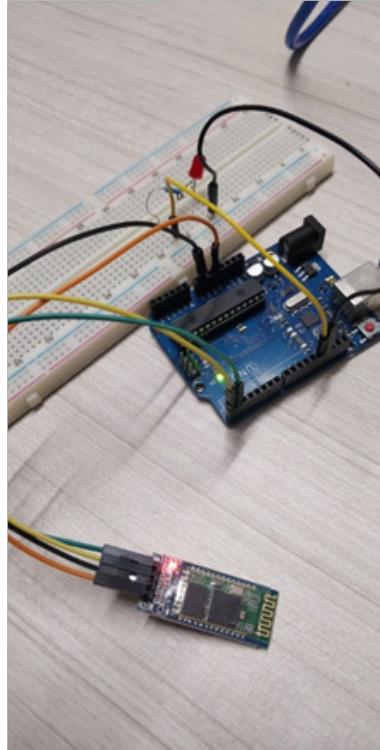
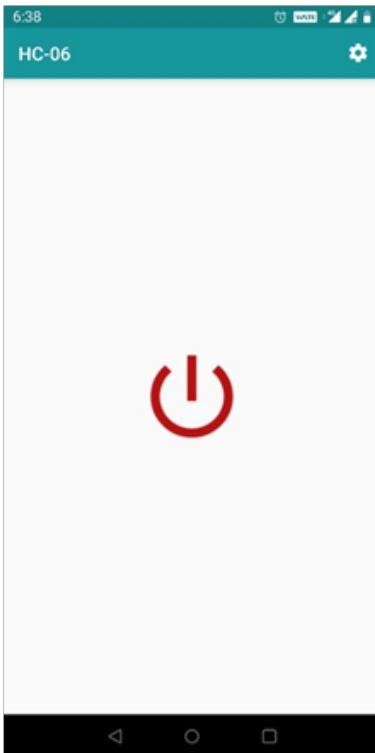
14. Click on the button icon to turn the Led ON.



# Intermediate Level Projects

38

15. Click on the button once again to turn the Led OFF



## PROJECT 22: L293D MOTOR SHIELD

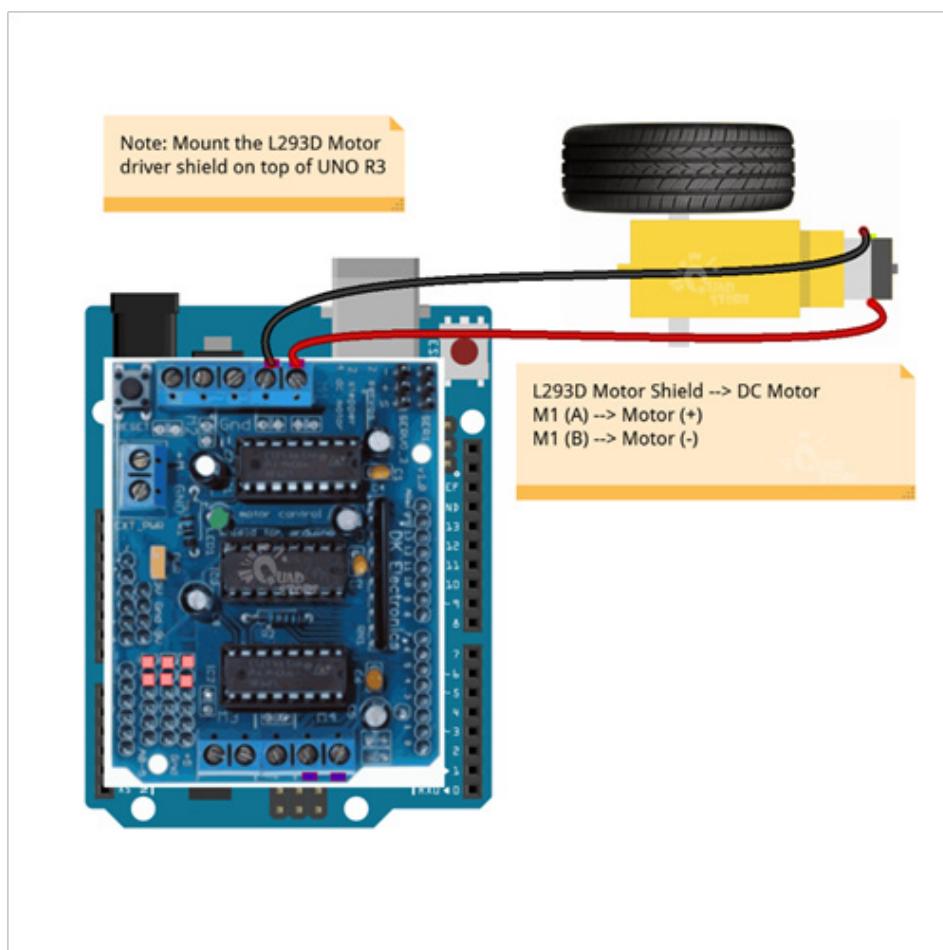
Make the DC motor run using L293D motor shield.

### Parts Required:

- \* Uno R3 board, USB Cable
- \* Few jumper wires

### Circuit Connection:

First Insert the L293D Motor shield on top of Uno R3 board and Provide the connection as per the below circuit.



**Code:** Verify and Upload the code named "**L293D\_Motor\_Shield.ino**" to Uno R3 board  
**Output:** You will see the DC motor rotate clockwise and anticlockwise alternatively.

## PROJECT 23: CONTROLLING DC MOTOR USING RELAY

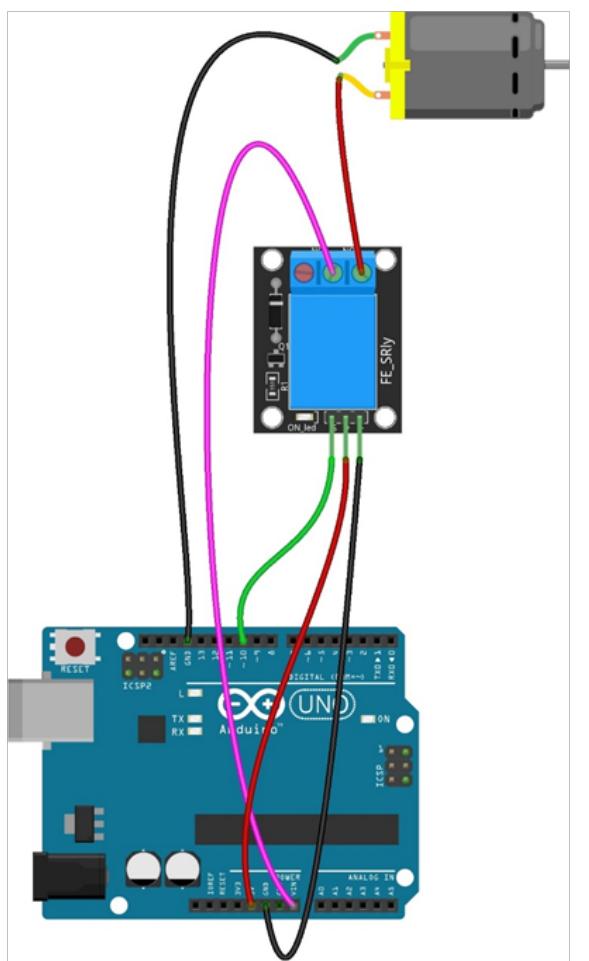
Turn on the LED and DC motor using relay module.

### Parts Required:

- \* Uno R3 board, USB Cable
- \* Breadboard
- \* Few jumper wires
- \* 2 Channel Relay.
- \* LED
- \* Resistor – 220ohm
- \* DC motor

### Circuit Connection:

Provide the connection as per the below circuit.



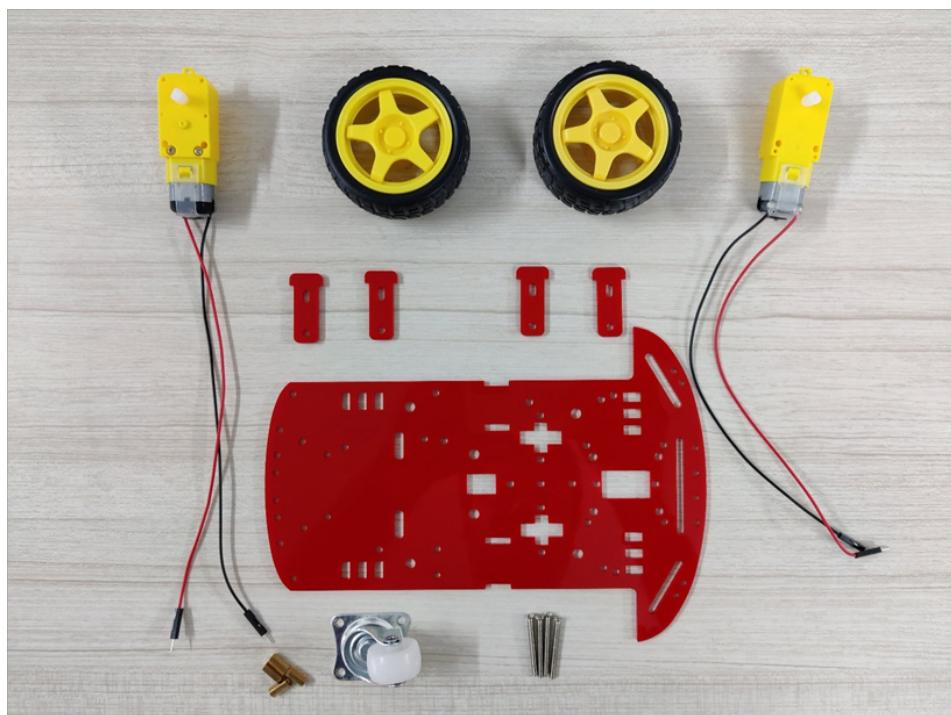
**Code:** Verify and Upload the code named “DCMotor.ino” to Uno R3 board

**Output:** The relay will switch ON and OFF and you will see DC motor also turns ON and OFF while the relay does.

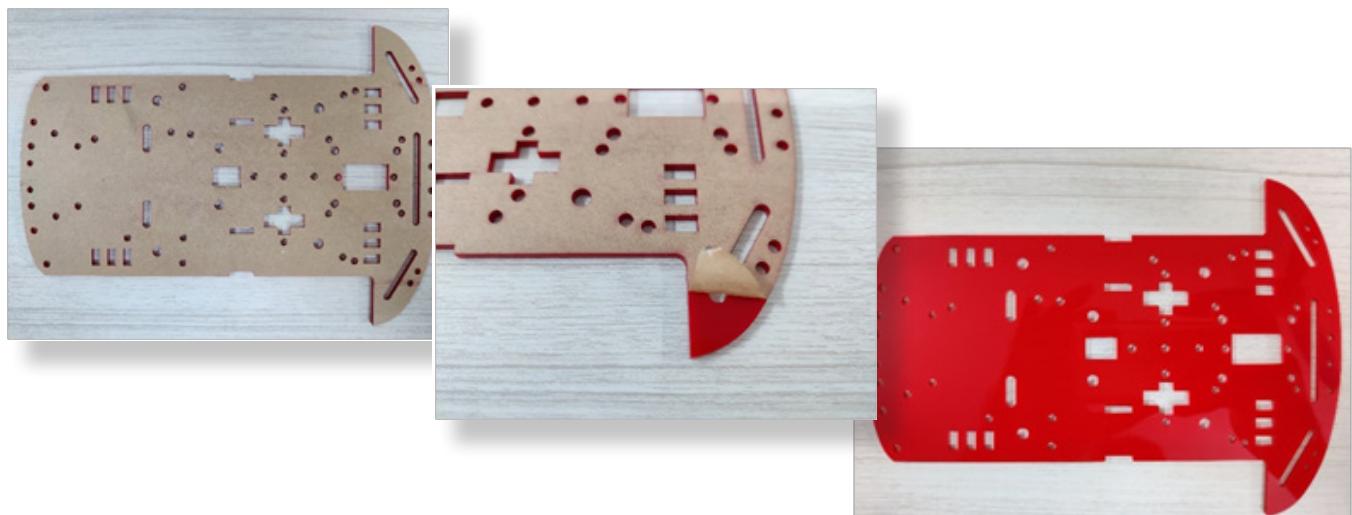
# Advance Level Projects

## PROJECT 24: 2WD CAR CHASSIS ASSEMBLY INSTRUCTIONS

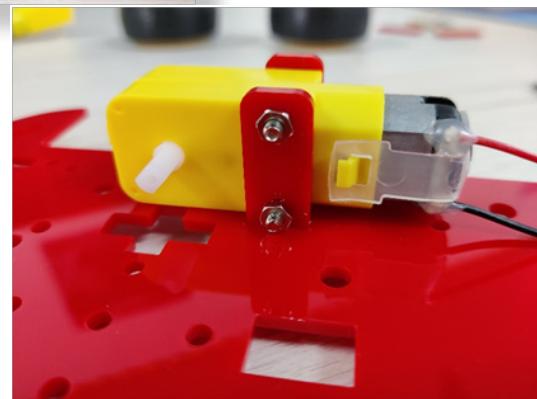
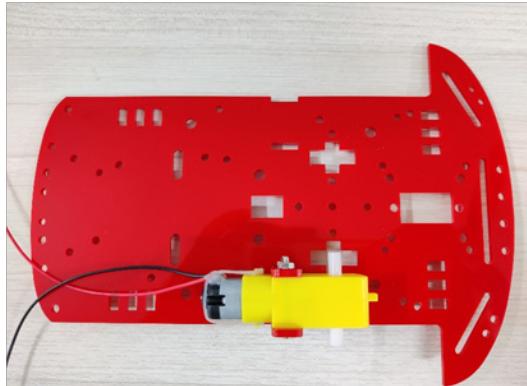
The basis for all the robot car is assembling the chassis and we will show you step by step how to assemble the car.



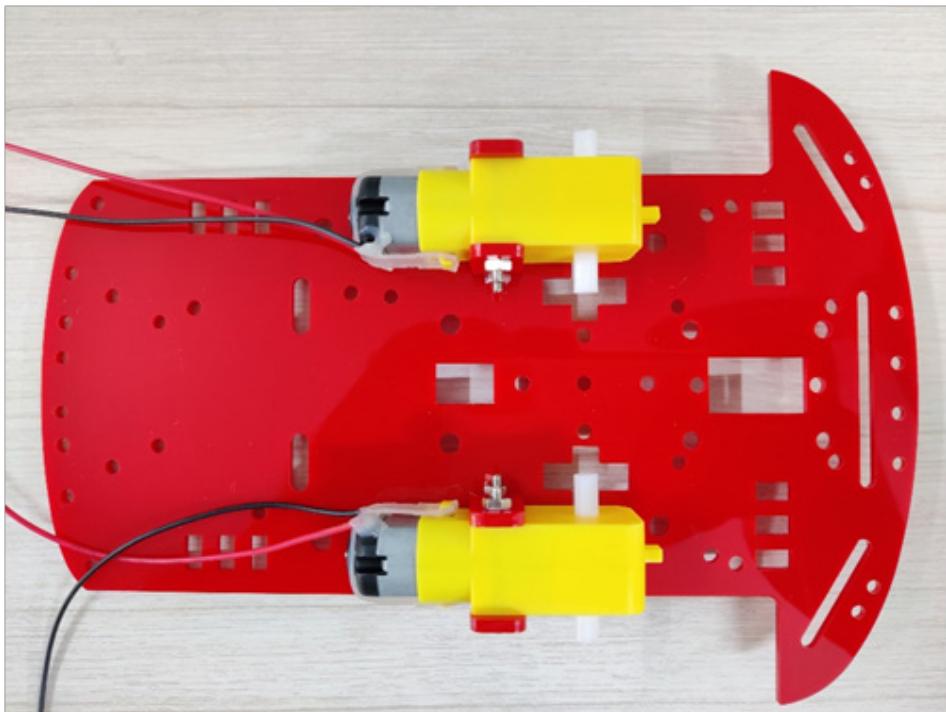
Step-1: Peel the outer sticker from the car chassis and the motor holders to make the chassis reveal its original bright color.



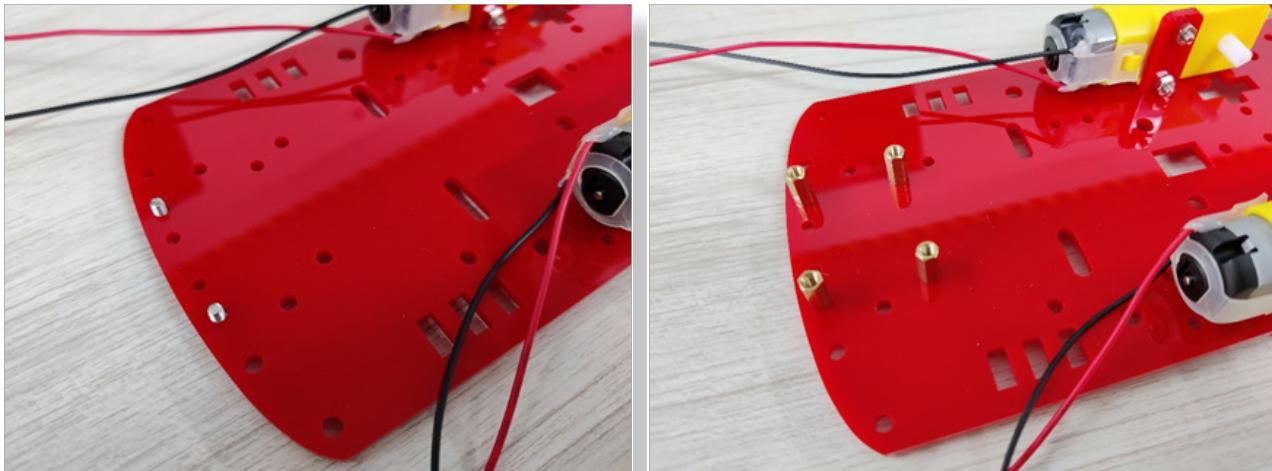
Step-2: Insert the motor holder into its respective slot and install the gear motor using the screws and nuts provided.



Step-3: Repeat the steps and install the gear motor in other side of the chassis as well.



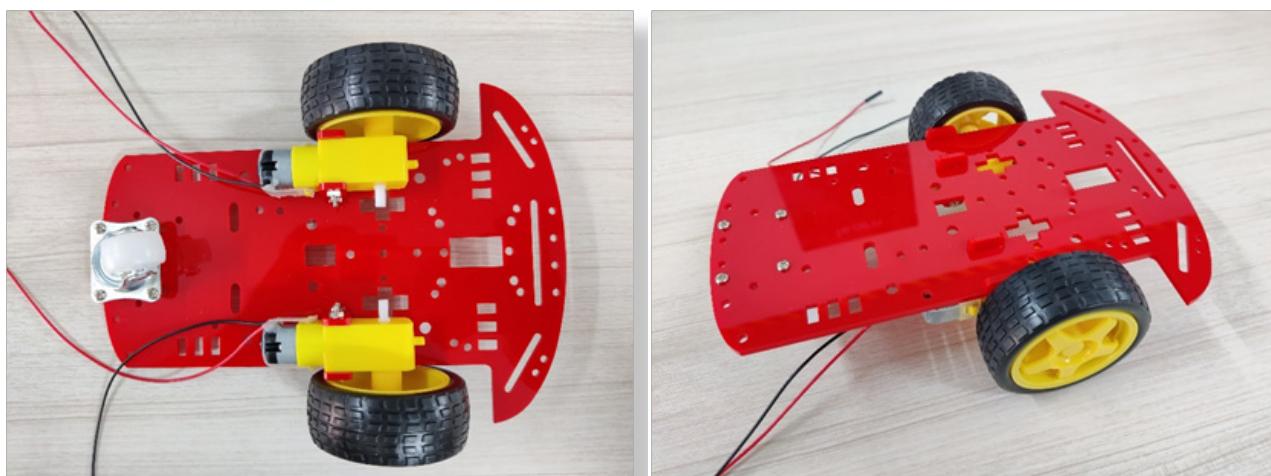
Step-4: Use the small screws and install all 4 Hex mount at the back end of the car chassis.



Step-5: Install the castor wheel on top of the hex mount using the screws.



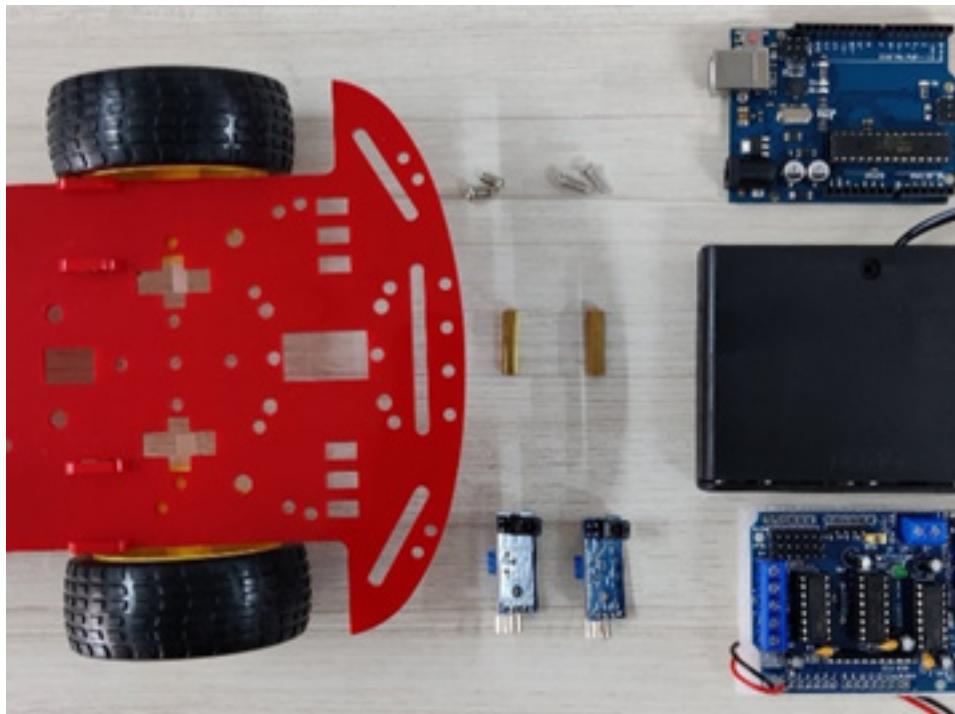
Step-6: Insert the wheels into the gear motor. Please align the wheel according the slot of the gear motor to insert properly. This completes the assembly instructions of 2WD car chassis.



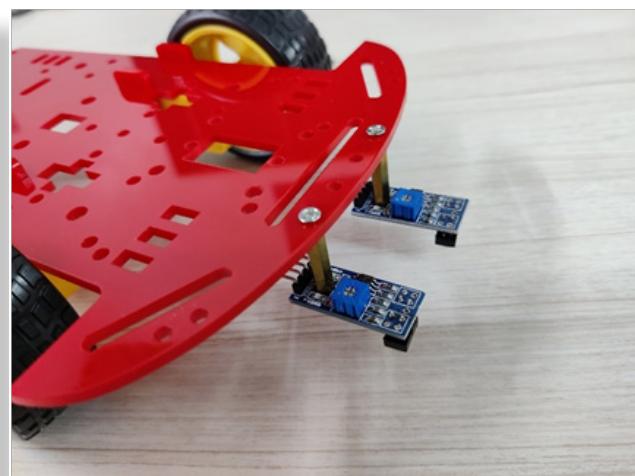
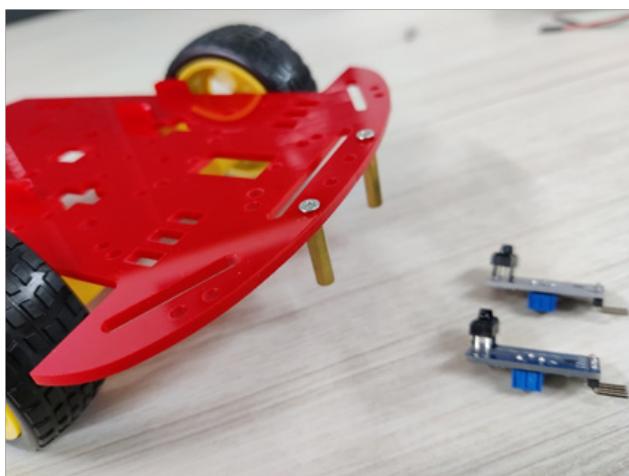
## PROJECT 25: LINE FOLLOWING ROBOT CAR

In this project we will build a line following car which will follow the black line in a white background.

Step-1: Take an assembled 2WD Car chassis. Just in case you are not aware how to assemble the 2WD car chassis please refer to the "2WD Card Chassis Assembly Instruction" section for details.

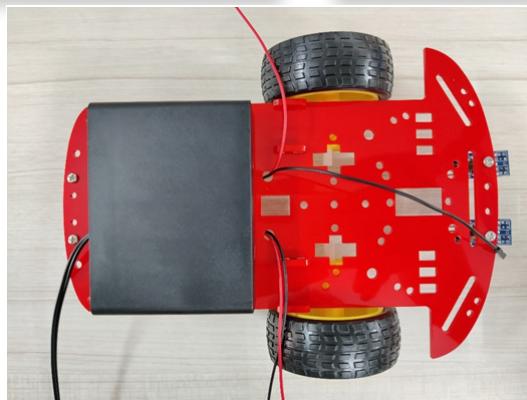


Step-2: Install 2 hex connectors and screw the line sensors to them as shown in the picture.

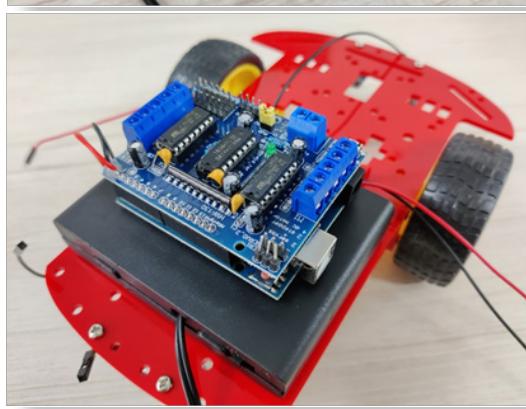
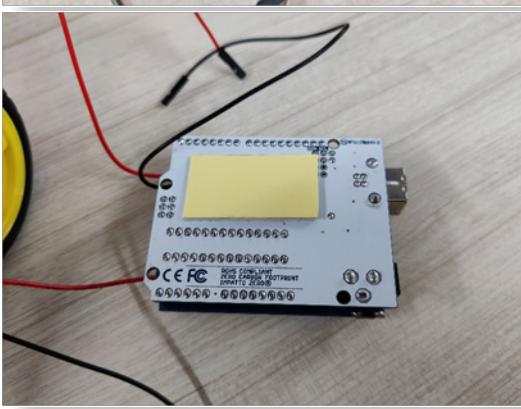
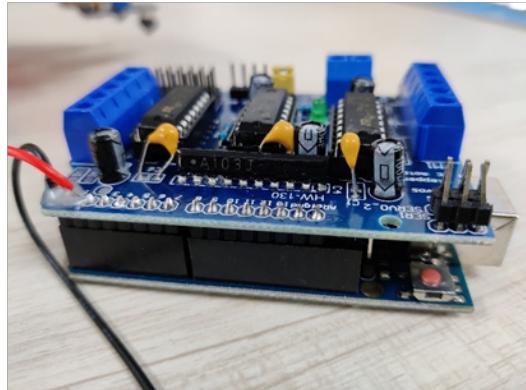
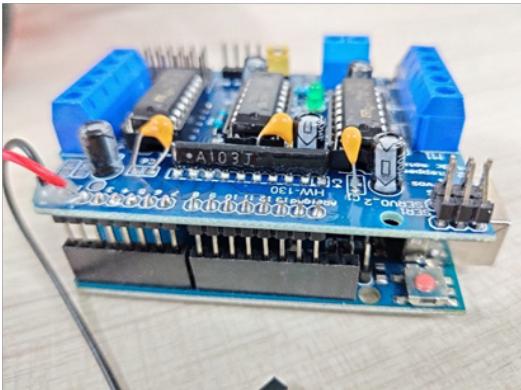


# Advance Level Projects

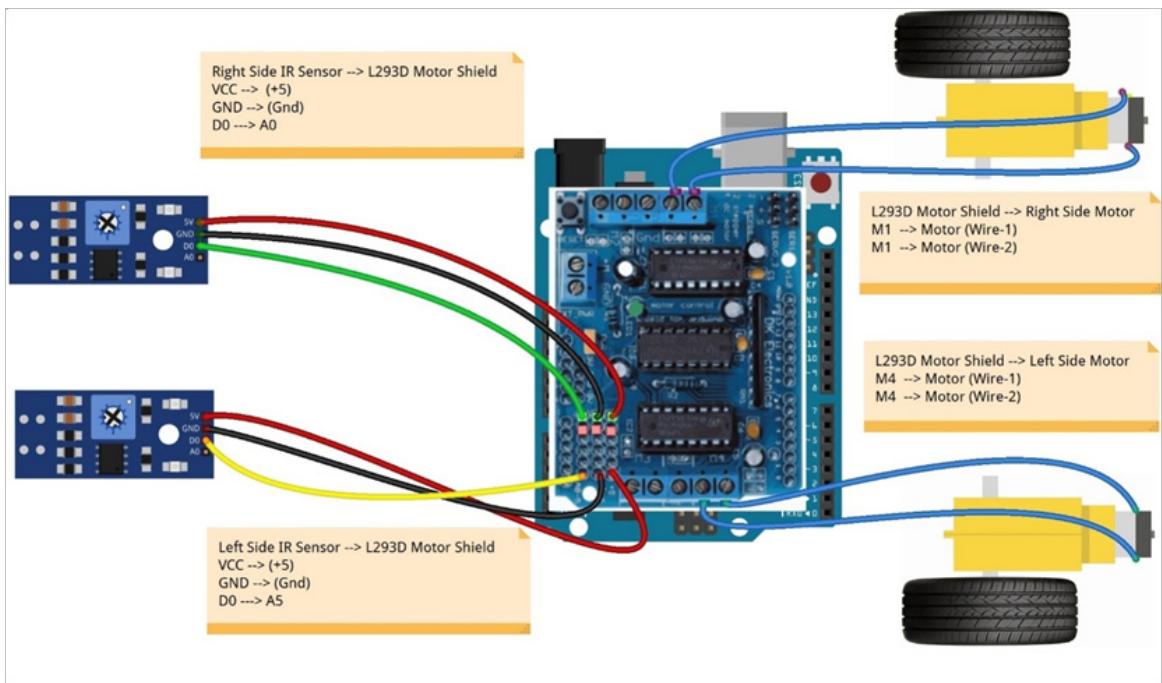
Step-3: Insert 6 x AA batteries in battery holder and place it in the back side of the car chassis using double side stickers.



Step-4: Insert the L293D motor driver shield on top of the Uno R3 board and use a double side tape and fix it on top of the battery holder.



## Step-5: Circuit Connection



Motors to L293D driver shield

### Left Side Motor L293 Shield mounted on Uno R3

Wire-1 (Red) M4

Wire-2 (Black) M4

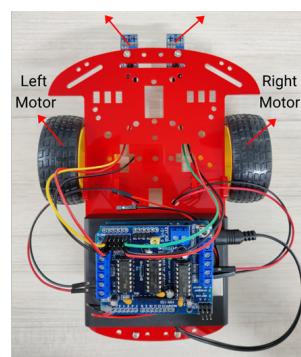
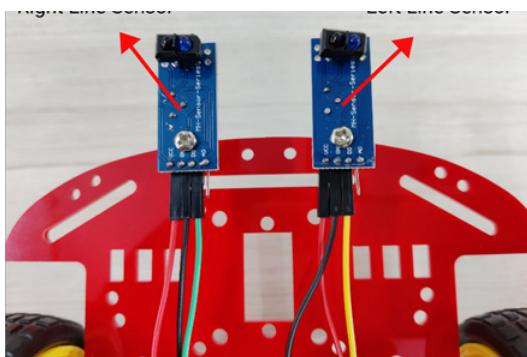
### Right Side Motor L293 Shield mounted on Uno R3

Wire-1 (Red) M1

Wire-2 (Black) M1

#### NOTE:

Kindly note that there is no polarity in gear motors. Hence the red and black wires provided with the motor is just for reference purpose. They can be interchanged if the motor is required to rotate in different directions.



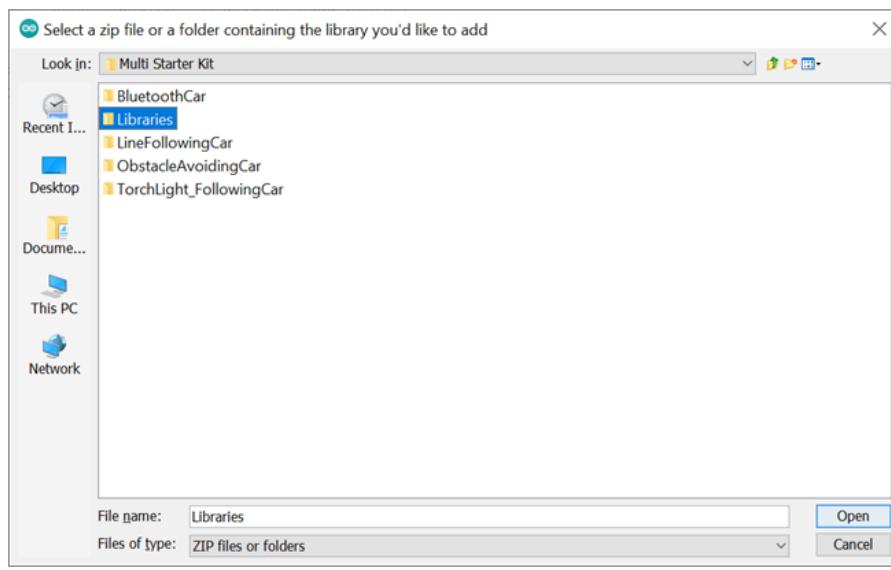
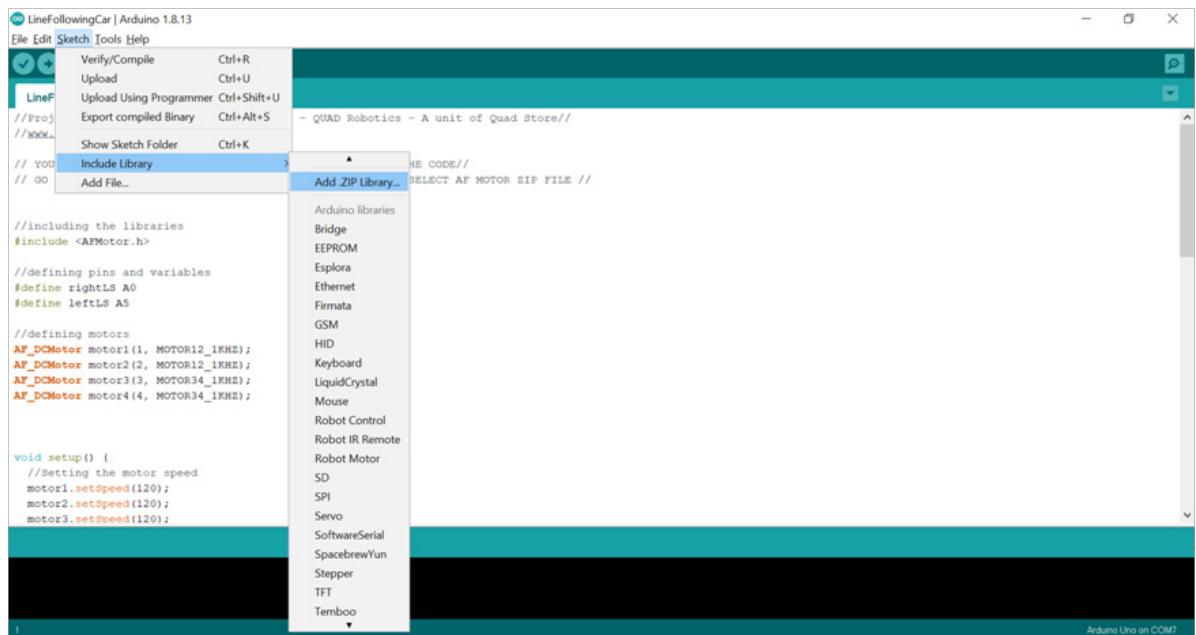
## Upload Code to Uno R3 board:

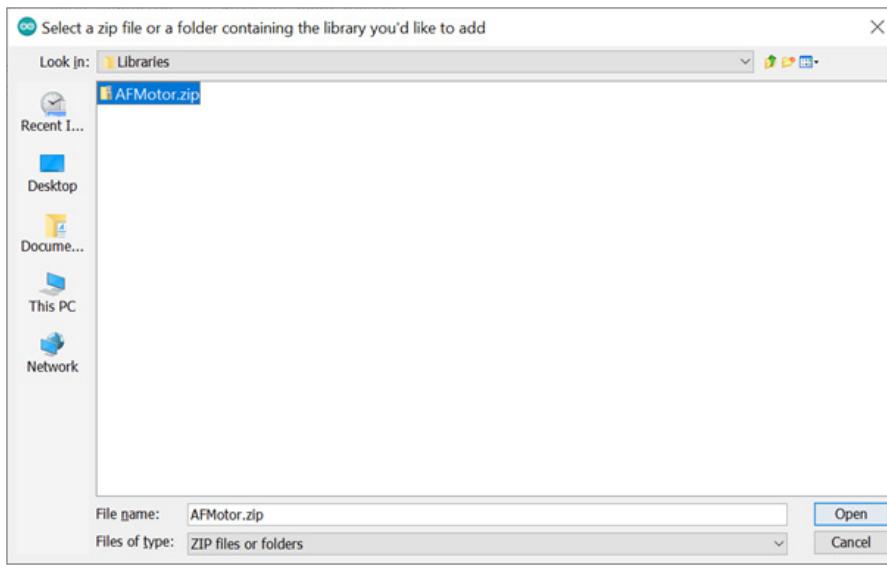
Use the provided USB cable and connect the Uno R3 board to your laptop/desktop computer.

## Installing Libraries:

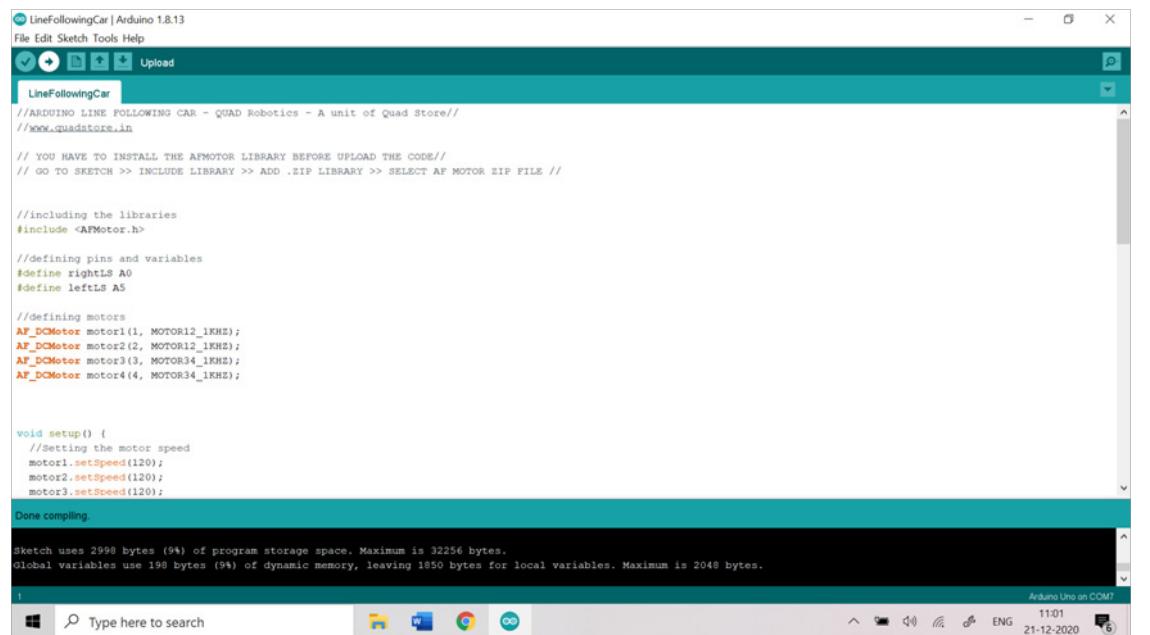
Open the **LineFollowingCar.ino** code using the Arduino IDE.

Before uploading the code to the Uno R3 board you need to install the "**AFMotor.zip**" library. Follow the below steps to install the library.





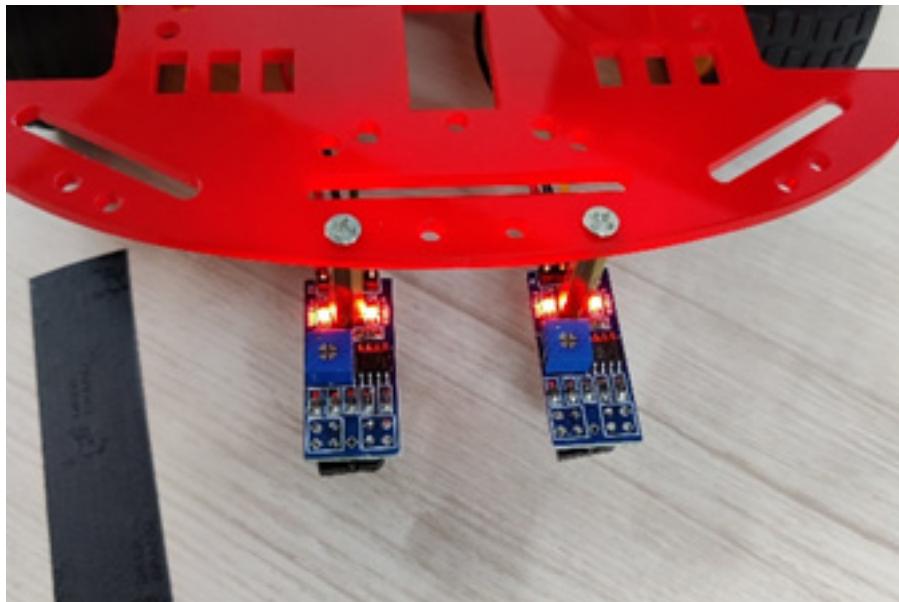
Compile and Upload the code to Uno R3 board.



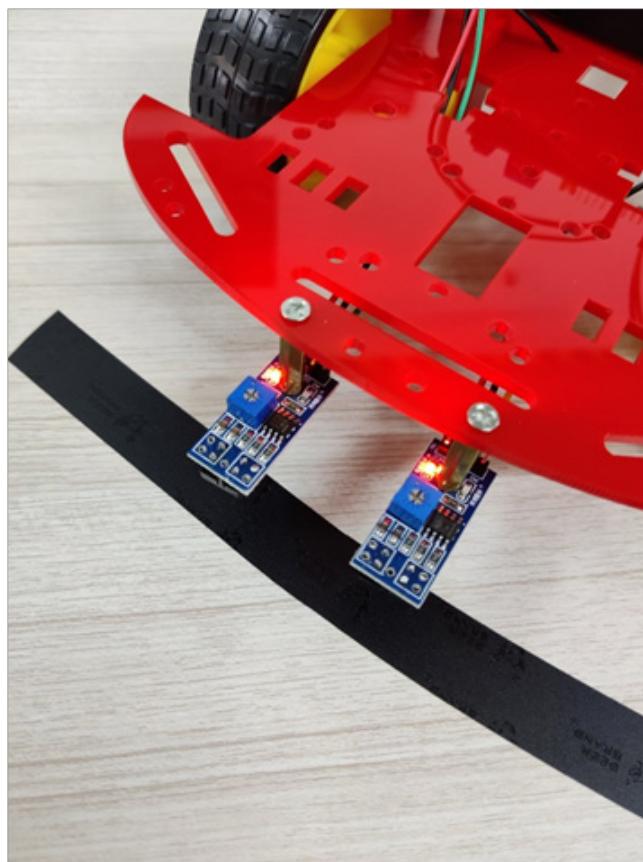
Step-6: Insert the DC power jack from battery holder into Uno R3 dc jack and Power ON the Uno R3 board. Now you should make sure the line sensors are working as expected.

**Condition-1:** When line sensor is in white surface then you should notice the below points.

- \* You should see both LED's in each line sensor should be turned ON.
  - \* Both wheels of the car should rotate in forward direction. If any of the wheels or both wheels rotation in reverse direction, then please interchange the red and black wire connection which goes to the L293D motor driver shield.

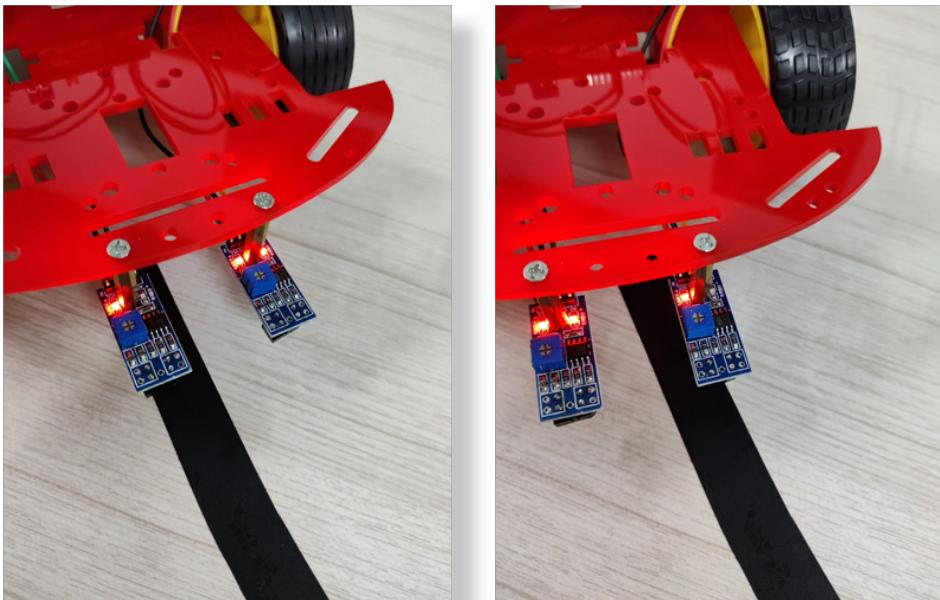


**Condition-2:** Black Surface: When line sensors is in black surface you see only one Led in each light sensor should be turned ON.

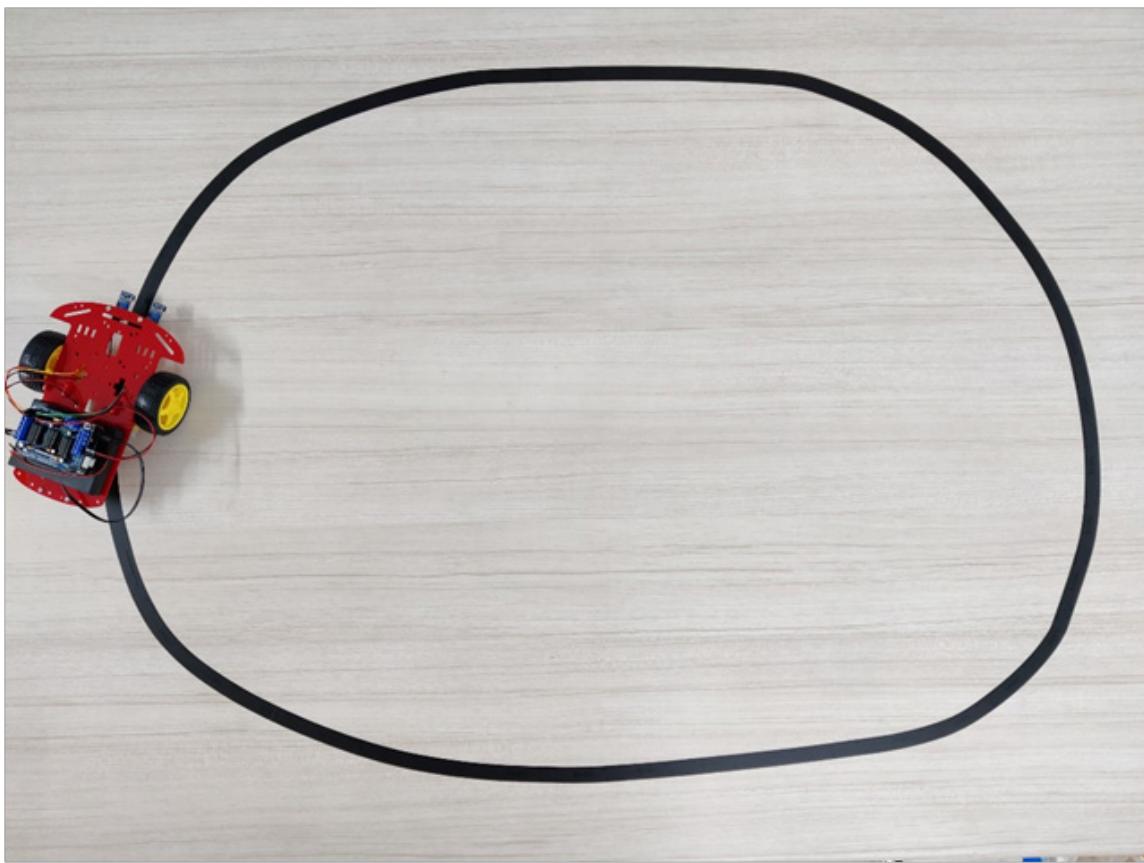


Note: If the above steps are not working as expected then please adjust the potentiometer in the line sensor using the screwdriver to its desired level to make it work.

Demonstration when one of the line sensor is in black surface and when the other one is in white surface. Make sure your line sensors also work in the same manner.

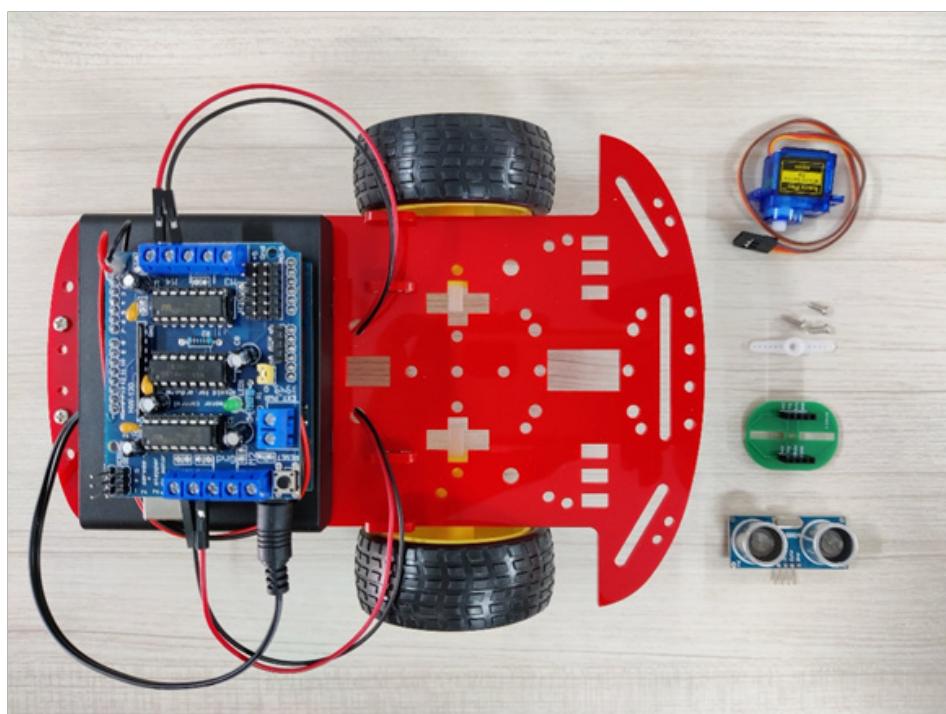


Step-7: Make a circuit in oval shape or round shape using black insulation tape. Kindly note the surface should be white with black line in it. If you are using reflective surfaces like floor, marble, granite, tiles it might not work well. So please use a chart paper or matt white surface to get the best results.

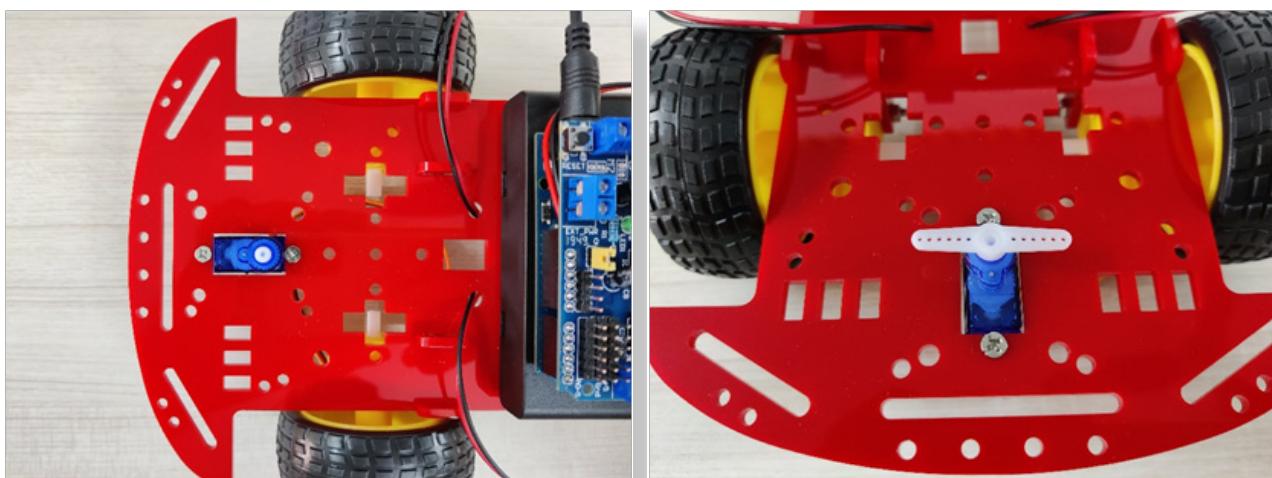


## PROJECT 26: OBSTACLE AVOIDANCE ROBOT CAR

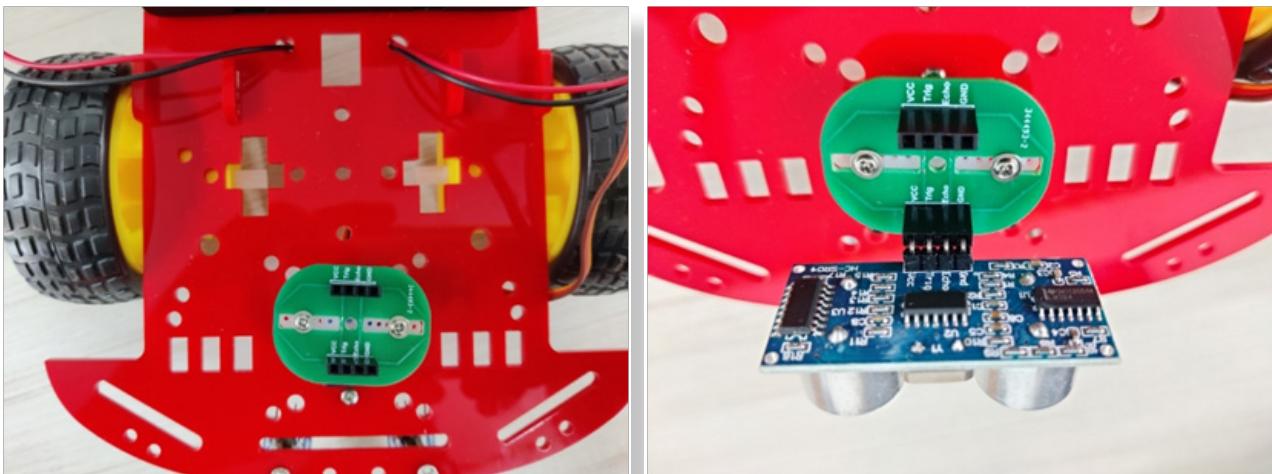
Step-1: Take an assembled 2WD Car chassis. Just in case you are not aware how to assemble the 2WD car chassis please refer to the "2WD Card Chassis Assembly Instruction" section for details.



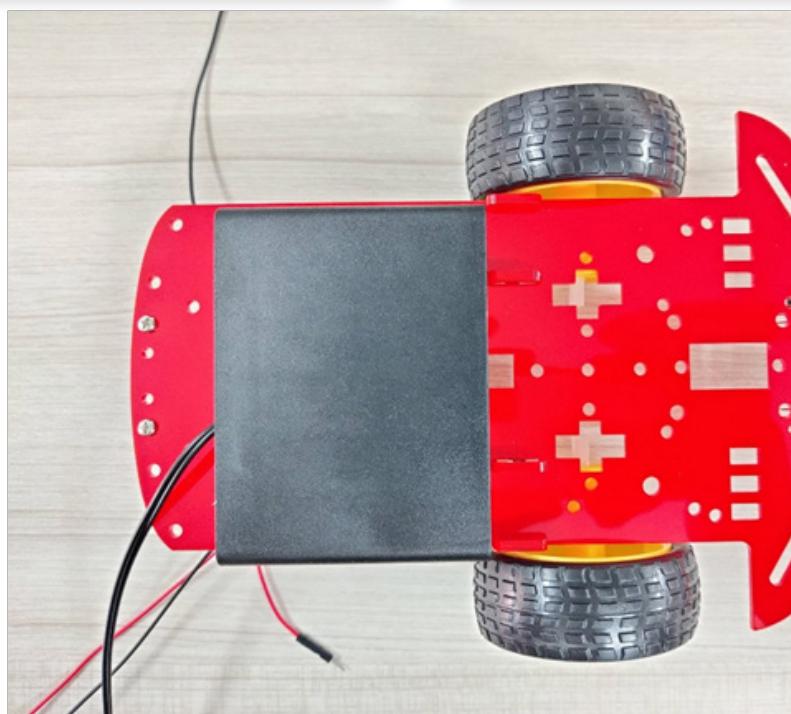
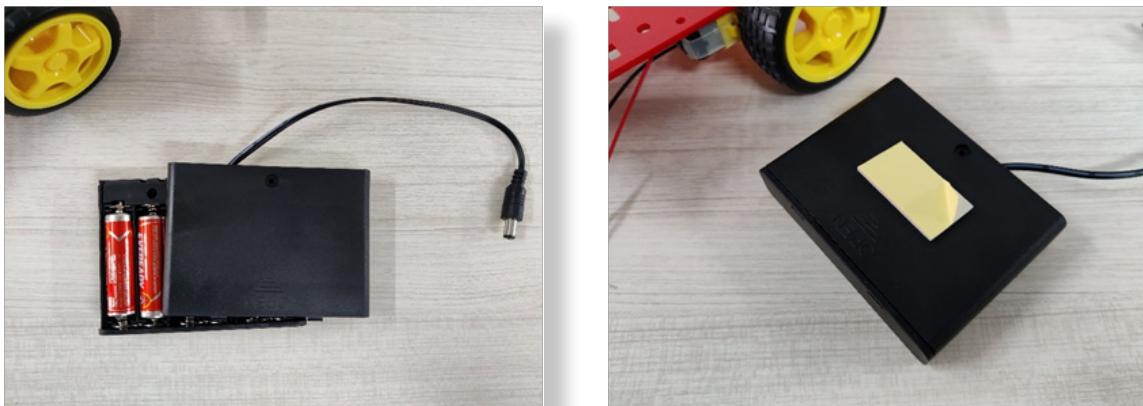
Step-2: Insert Servo motor from bottom to the servo slot and screw them. Insert servo arm to the servo.



Step-3: Place the Ultrasonic sensor holder on the servo arm and screw them. Kindly note the direction in which the ultrasonic sensor holder is mounted, because it has to match with the pin terminals mentioned in the ultrasonic sensor. Insert the ultrasonic sensor into the holder.

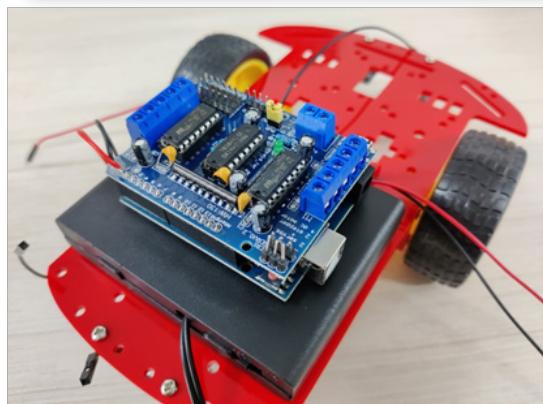
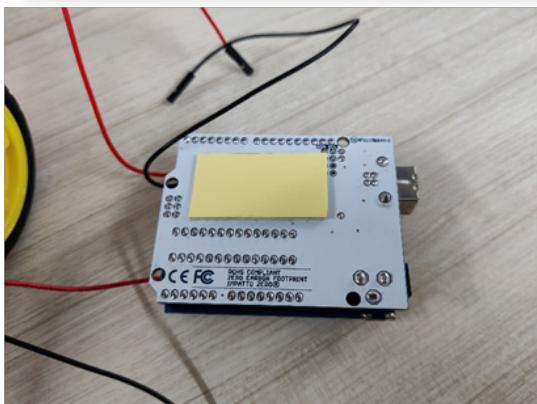
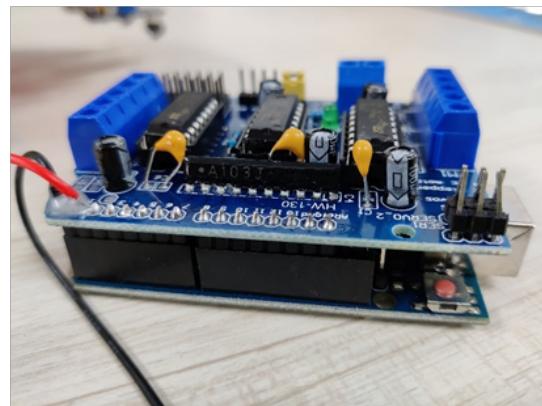
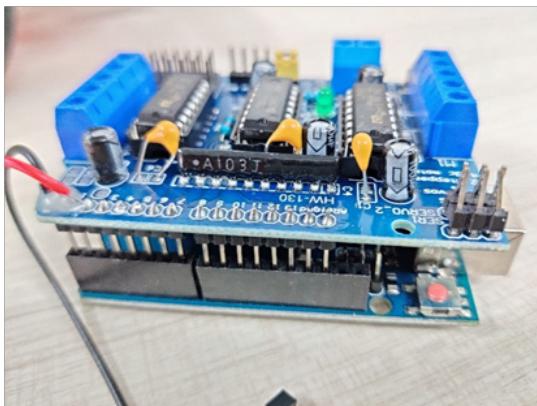


Step-4: Insert 6 x AA batteries in battery holder and place it in the back side of the car chassis using double side stickers.

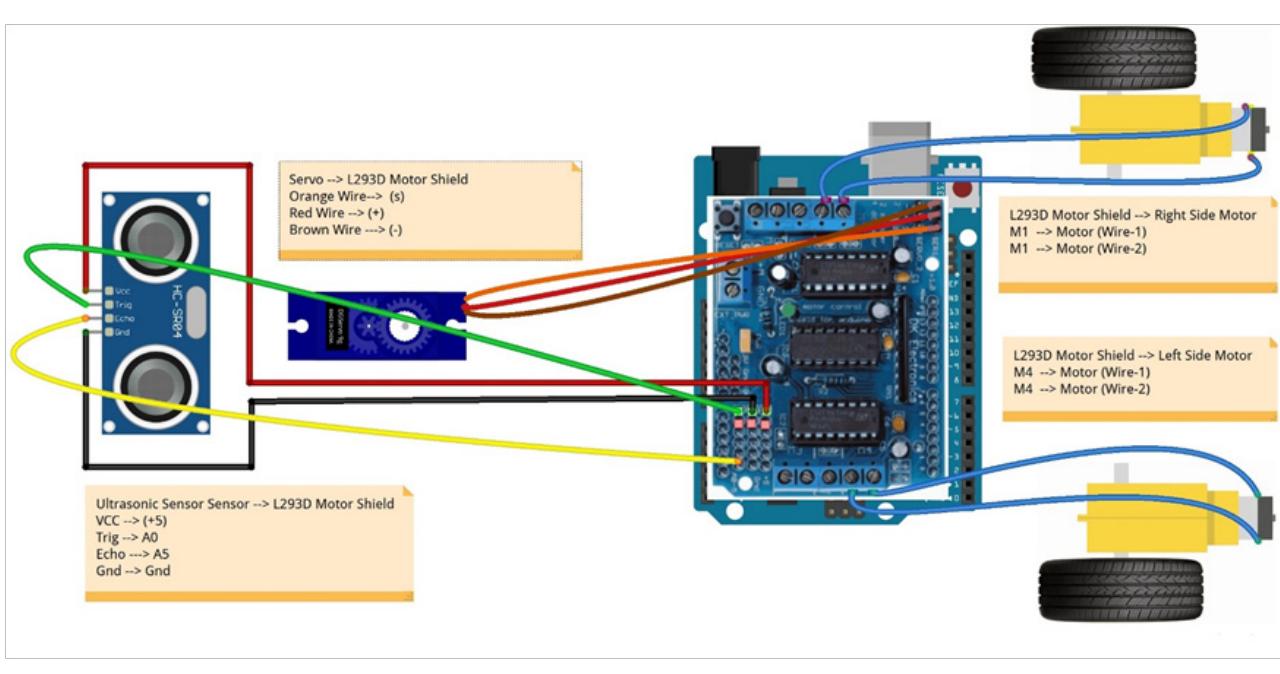


# Advance Level Projects

Step-5: Insert the L293D motor driver shield on top of the Uno R3 board and use a double side tape and fix it on top of the battery holder.



Step-6: Circuit Connection



Motors to L293D driver shield

## Left Side Motor L293 Shield mounted on Uno R3

Wire-1 (Red) M4

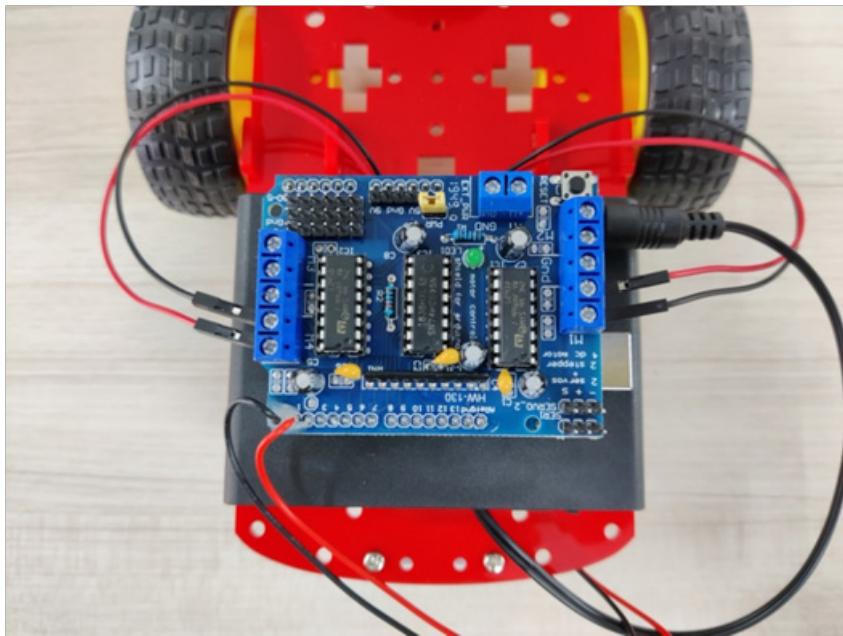
Wire-2 (Black) M4

## Right Side Motor L293 Shield mounted on Uno R3

Wire-1 (Red) M1

Wire-2 (Black) M1

Note: kindly note that there is no polarity in gear motors. Hence the red and black wires provided with the motor is just for reference purpose. They can be interchanged if the motor is required to rotate in different directions.



Ultrasonic Sensor to L293D driver shield:

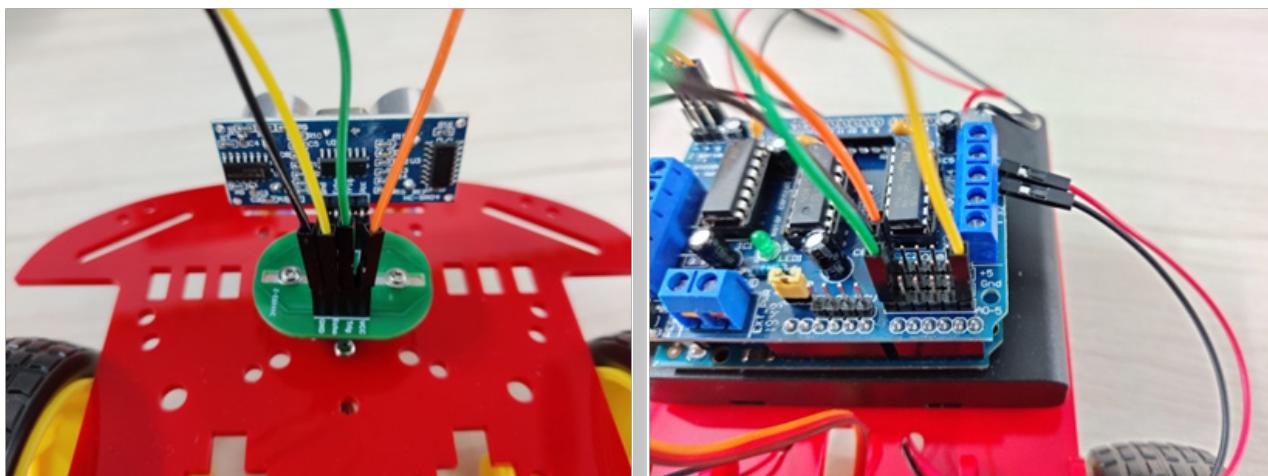
## Ultrasonic Sensor L293 Shield mounted on Uno R3

Gnd Gnd

Echo A5

Trig A0

Vcc +5



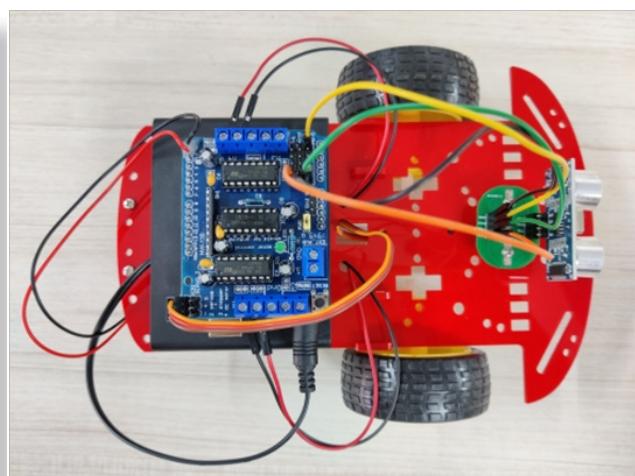
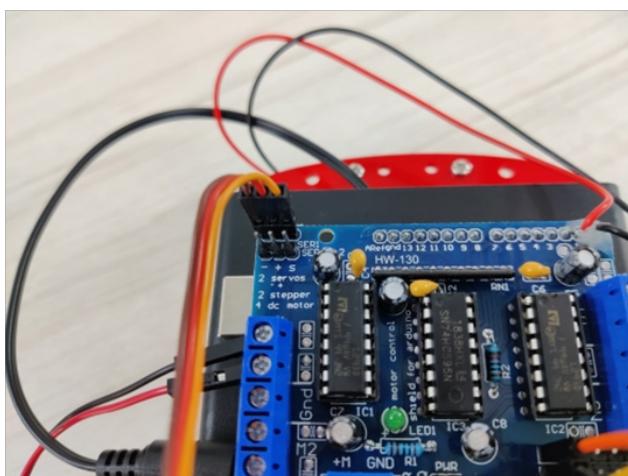
Servo Motor to L293D driver shield:

<b>Servo Motor</b>	<b>L293 Shield mounted on Uno R3</b>
--------------------	--------------------------------------

Brown Color Wire -

Red Color Wire +

Orange color  
Wire s



**Upload Code to Uno R3 board:**

Use the provided USB cable and connect the Uno R3 board to your laptop/desktop computer.

**Installing Libraries:**

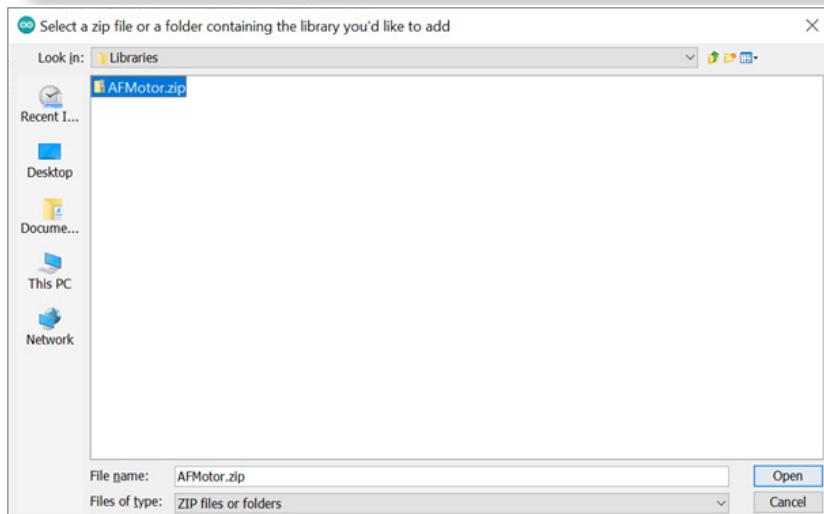
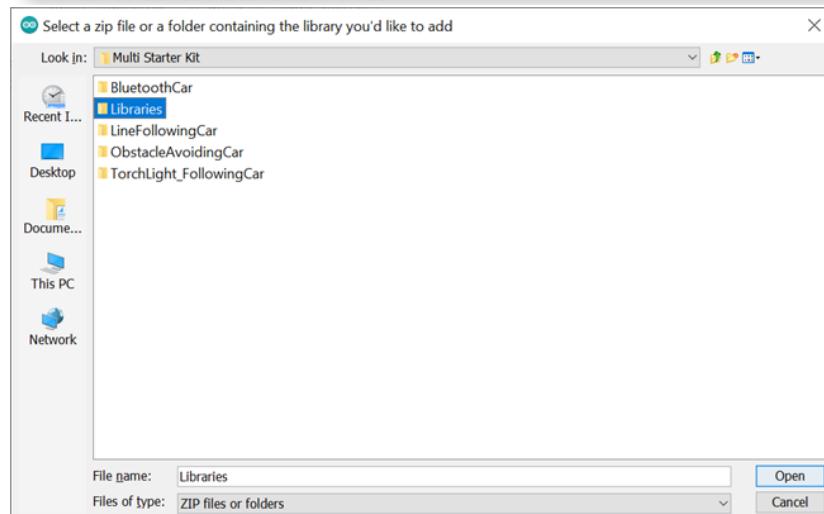
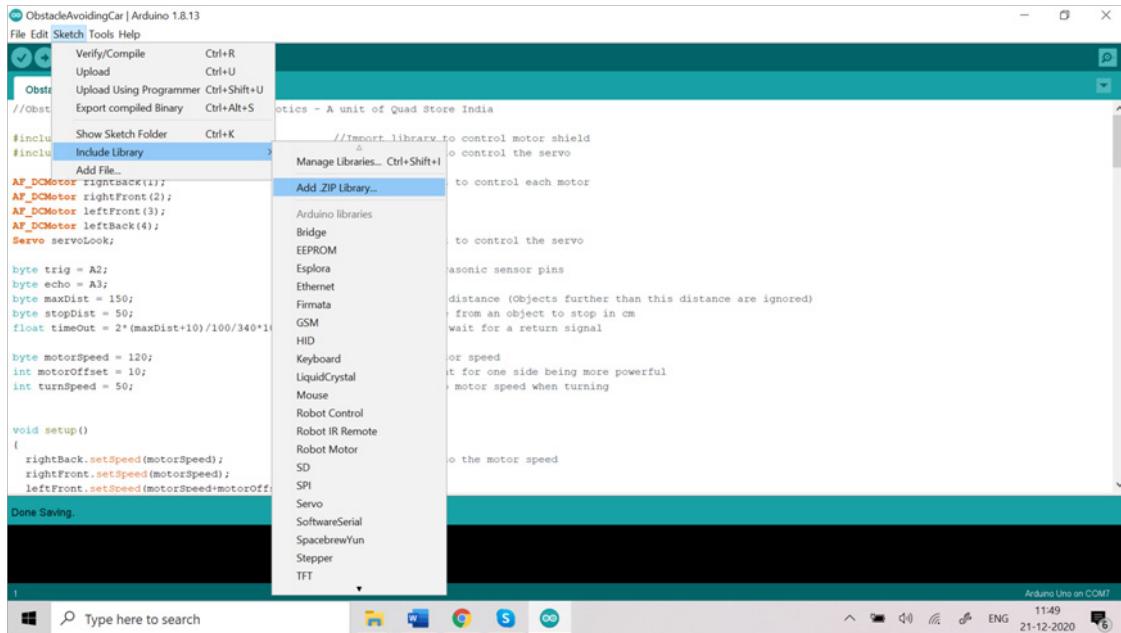
Open the **ObstacleAvoidingCar.ino** code using the Arduino IDE.

Before uploading the code to the Uno R3 board you need to install the "**AFMotor.zip**" library.

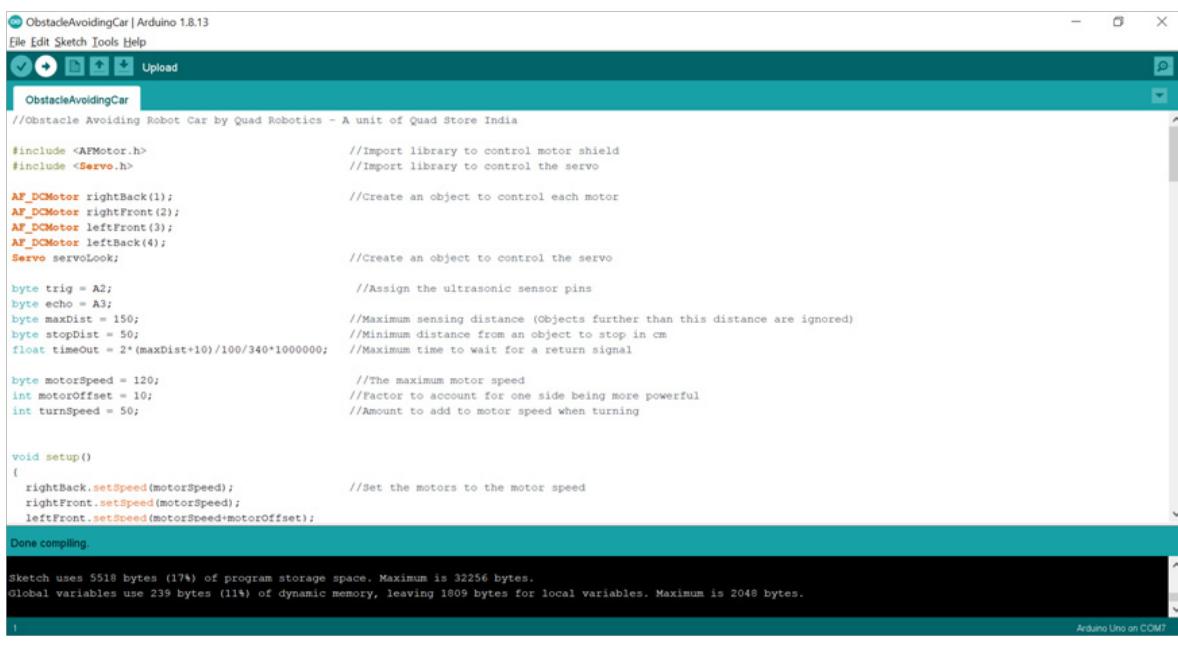
# Advance Level Projects

56

NOTE: If you had already installed this AFMotor.zip library during the previous projects then you can ignore this step. Else follow the below steps to install the library.



Compile and Upload the code to Uno R3 board.



```

ObstacleAvoidingCar | Arduino 1.8.13
File Edit Sketch Tools Help
Upload
ObstacleAvoidingCar
//Obstacle Avoiding Robot Car by Quad Robotics - A unit of Quad Store India

#include <AFMotor.h> //Import library to control motor shield
#include <Servo.h> //Import library to control the servo

AF_DCMotor rightBack(1);
AF_DCMotor rightFront(2);
AF_DCMotor leftFront(3);
AF_DCMotor leftBack(4);
Servo servoLook; //Create an object to control the servo

byte trig = A2; //Assign the ultrasonic sensor pins
byte echo = A3;
byte maxDist = 150; //Maximum sensing distance (Objects further than this distance are ignored)
byte stopDist = 50; //Minimum distance from an object to stop in cm
float timeout = 2*(maxDist+10)/100/340*1000000; //Maximum time to wait for a return signal

byte motorSpeed = 120; //The maximum motor speed
int motorOffset = 10; //Factor to account for one side being more powerful
int turnSpeed = 50; //Amount to add to motor speed when turning

void setup()
{
    rightBack.setSpeed(motorSpeed); //Set the motors to the motor speed
    rightFront.setSpeed(motorSpeed);
    leftFront.setSpeed(motorSpeed+motorOffset);
}

Done compiling.

Sketch uses 5518 bytes (17%) of program storage space. Maximum is 32256 bytes.
Global variables use 239 bytes (1%) of dynamic memory, leaving 1809 bytes for local variables. Maximum is 2048 bytes.

Arduino Uno on COM7

```

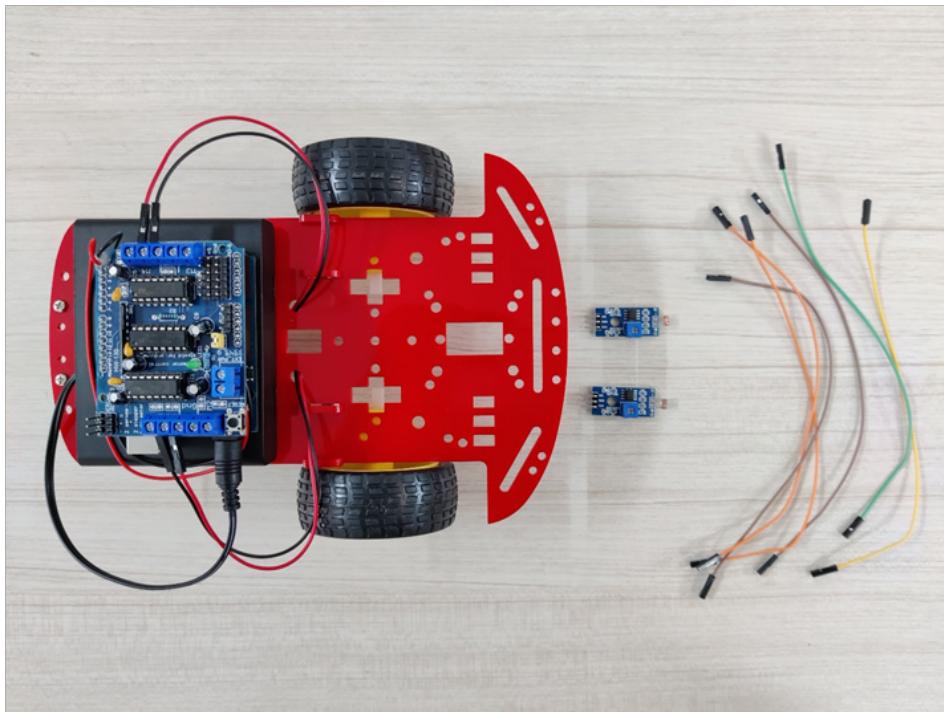
Step-7: Insert the DC power jack from battery holder into Uno R3 dc jack and Power ON the Uno R3 board.

Result: You will see the car avoiding the obstacles and going towards the longest path available where there is no obstacle in front of it.

## PROJECT 27: LIGHT FOLLOWING CAR

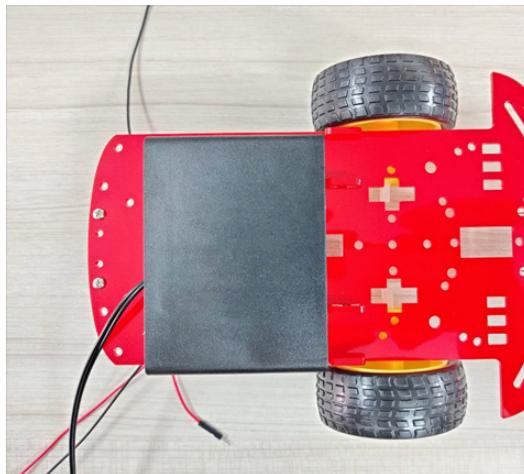
In this project we will build a light following car which will follow the light in the dark.

Step-1: Take an assembled 2WD Car chassis. Just in case you are not aware how to assemble the 2WD car chassis please refer to the "2WD Card Chassis Assembly Instruction" section for details.

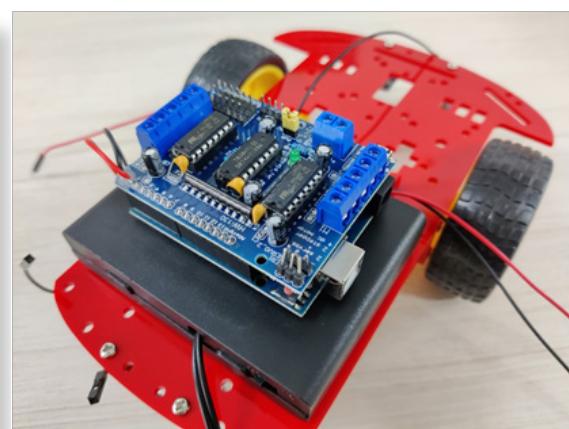
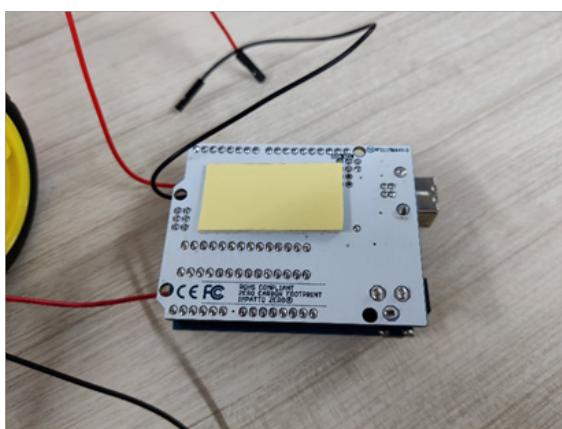
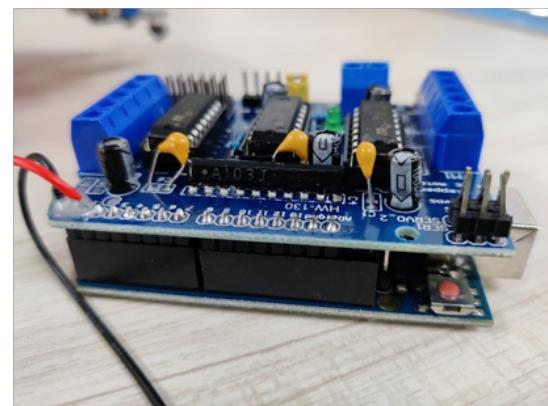
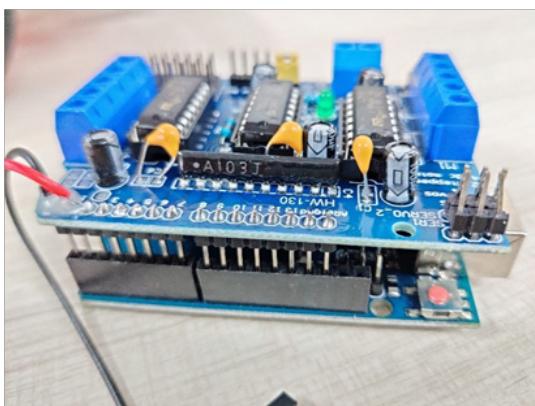


Step-2: Insert 6 x AA batteries in battery holder and place it in the back side of the car chassis using double side stickers.

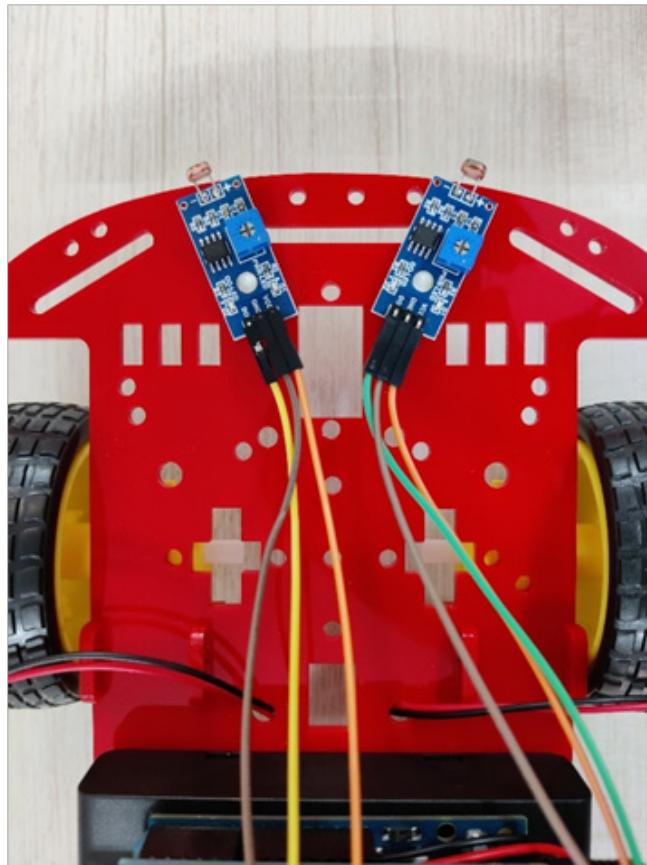




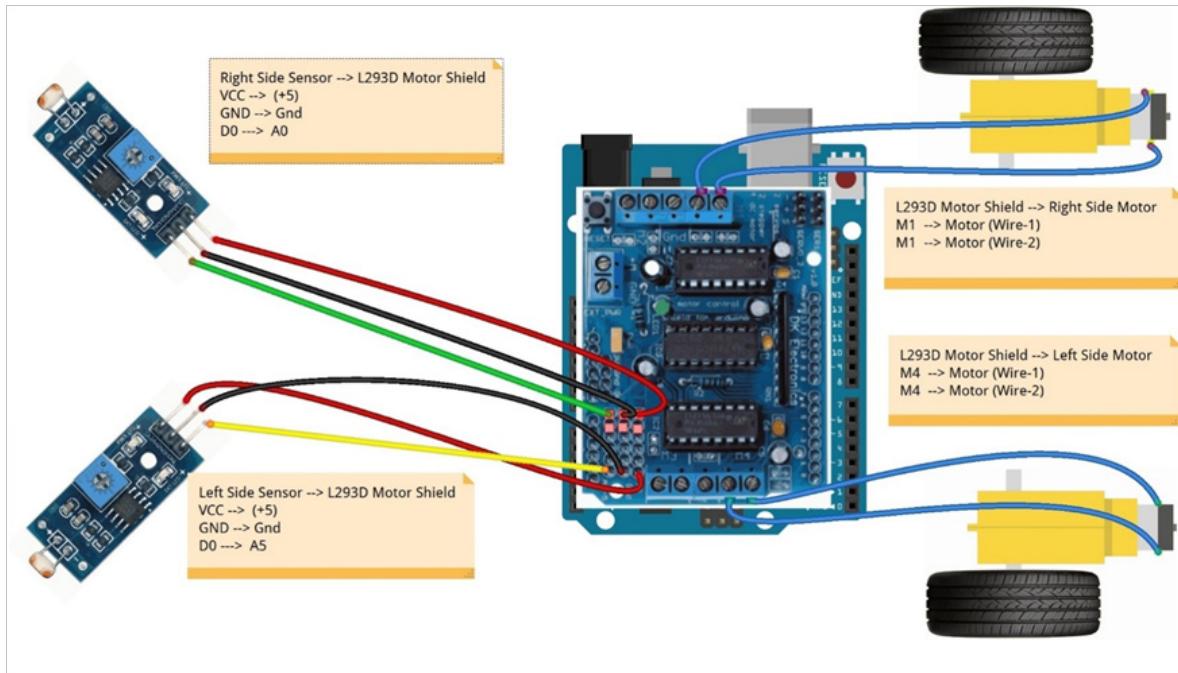
Step-3: Insert the L293D motor driver shield on top of the Uno R3 board and use a double side tape and fix it on top of the battery holder.



Step-4: Use double side tape and fix the Photoresistor in the 45 degree angle as shown in the picture below.



Step-6: Circuit Connection



Motors to L293D driver shield

**Left Side Motor L293 Shield mounted on Uno R3**

Wire-1 (Red) M4

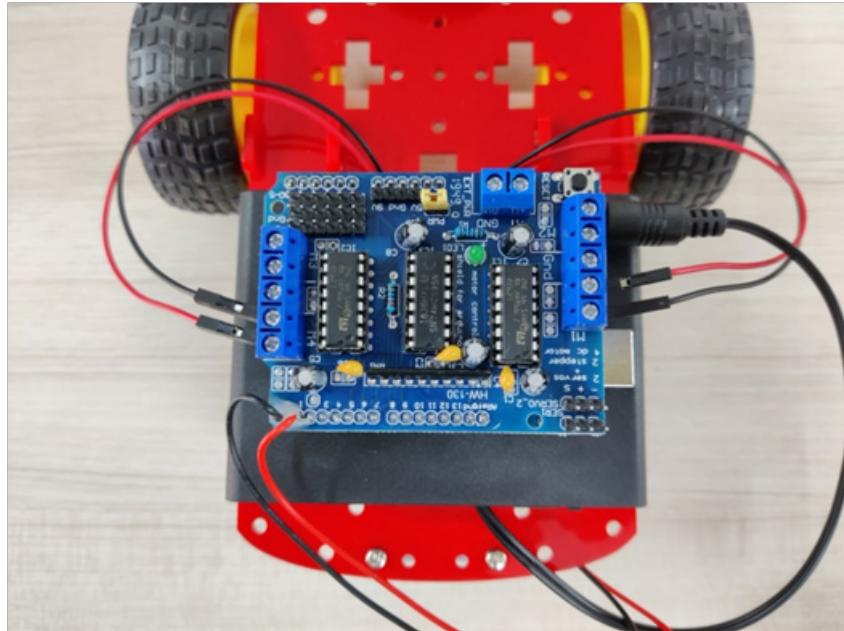
Wire-2 (Black) M4

**Right Side Motor L293 Shield mounted on Uno R3**

Wire-1 (Red) M1

Wire-2 (Black) M1

Note: kindly note that there is no polarity in gear motors. Hence the red and black wires provided with the motor is just for reference purpose. They can be interchanged if the motor is required to rotate in different directions.



**Left Side Photoresistor L293 Shield mounted on Uno R3**

VCC +5

GND Gnd

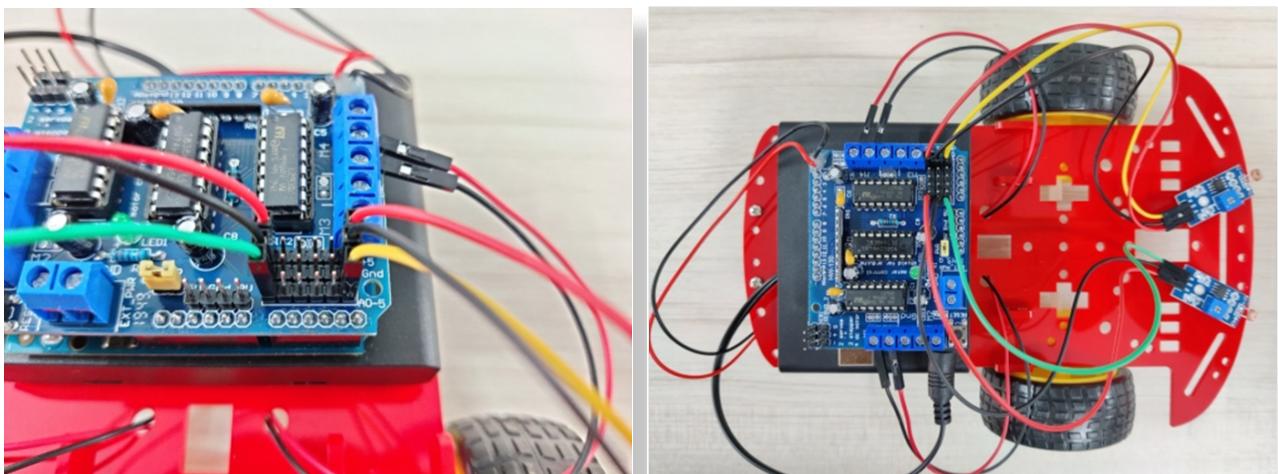
D0 A5

**Right Side Photoresistor L293 Shield mounted on Uno R3**

VCC +5

GND Gnd

D0 A0



### Upload Code to Uno R3 board:

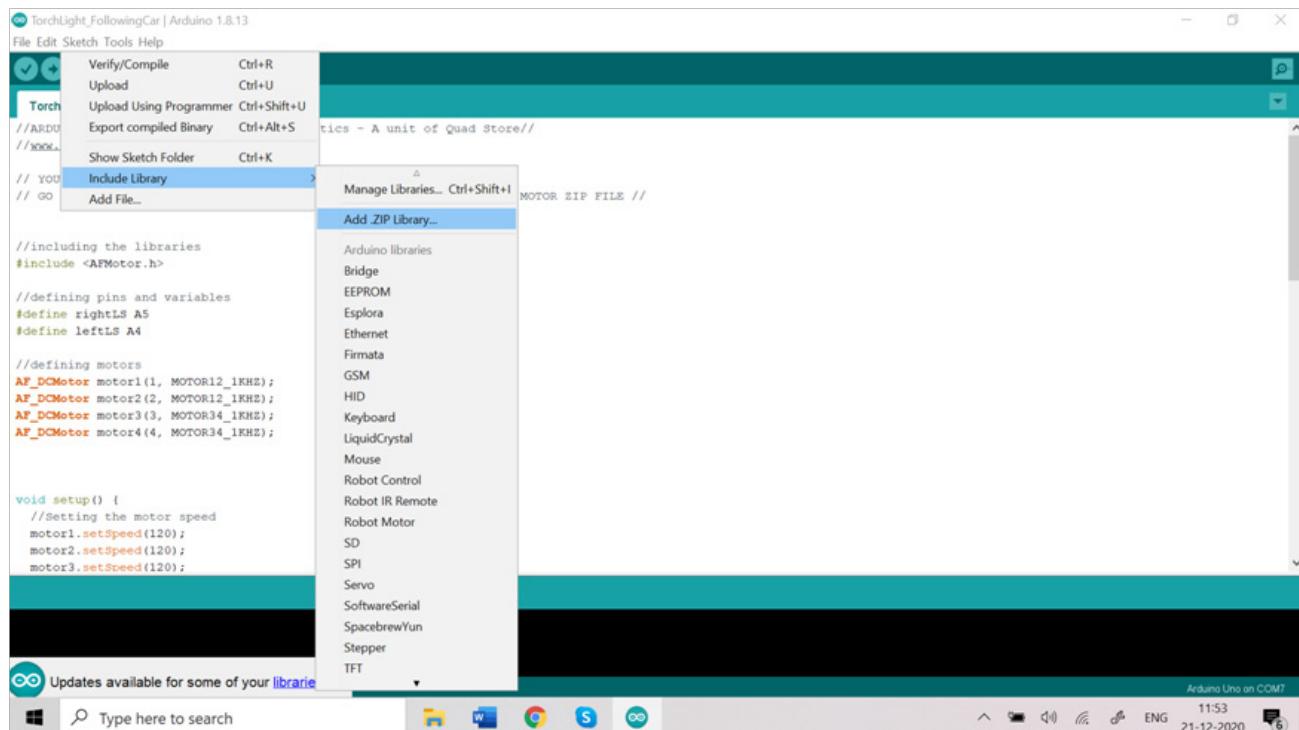
Use the provided USB cable and connect the Uno R3 board to your laptop/desktop computer.

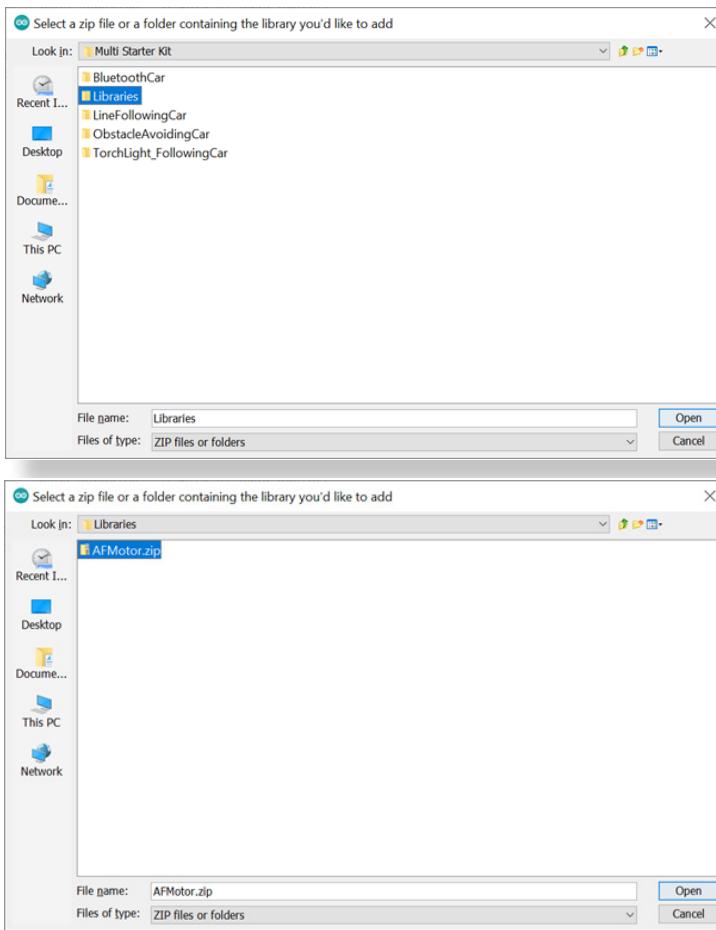
### Installing Libraries:

Open the **TorchLight\_FollowingCar.ino** code using the Arduino IDE.

Before uploading the code to the Uno R3 board you need to install the "**AFMotor.zip**" library.

**NOTE:** If you had already installed this AFMotor.zip library during the previous projects then you can ignore this step and directly compile and upload the code to UNO R3 board. Else follow the below steps to install the library.





Compile and Upload the code to Uno R3 board.

```

TorchLight_FollowingCar | Arduino 1.8.13
File Edit Sketch Tools Help
Upload
TorchLight_FollowingCar
//ARDUINO Light Following Car - QUAD Robotics - A unit of Quad Store//
//www.quadstore.in

// YOU HAVE TO INSTALL THE AFMOTOR LIBRARY BEFORE UPLOAD THE CODE//
// GO TO SKETCH >> INCLUDE LIBRARY >> ADD .ZIP LIBRARY >> SELECT AF MOTOR ZIP FILE //

//including the libraries
#include "AFMotor.h"

//defining pins and variables
#define rightLS A5
#define leftLS A4

//defining motors
AF_DCMotor motor1(1, MOTOR12_1KHZ);
AF_DCMotor motor2(2, MOTOR12_1KHZ);
AF_DCMotor motor3(3, MOTOR34_1KHZ);
AF_DCMotor motor4(4, MOTOR34_1KHZ);

void setup() {
  //Setting the motor speed
  motor1.setSpeed(120);
  motor2.setSpeed(120);
  motor3.setSpeed(120);
}

```

Step-8: Insert the DC power jack from battery holder into Uno R3 dc jack and Power ON the Uno R3 board.

Result: In a dark room, show the light in front of the cars photoresistor sensors and see the robot car following the light.



## PROJECT 28: BLUETOOTH CONTROLLED CAR

**Introduction:** In this project we will demonstrate how to assemble a Bluetooth controlled car using Android app. (Works only with Android mobile phones). Does not support iOS.

### Parts Required:

- \* 2WD Car Chassis Assembled

- \* Uno R3

- \* L293D motor driver shield

- \* HC05 Bluetooth module

- \* Arduino Bluetooth RC Car app – Download from PlayStore.

**\* IMPORTANT NOTE:** Kindly upload the code for Bluetooth car to UNO R3 board before installation. If you want to upload the code after installation, then you need to remove the power supply of the Bluetooth module before uploading. If you try to upload the code with Bluetooth module power supply, it will throw error message.

### Step-1: Upload the Code to Uno R3 board

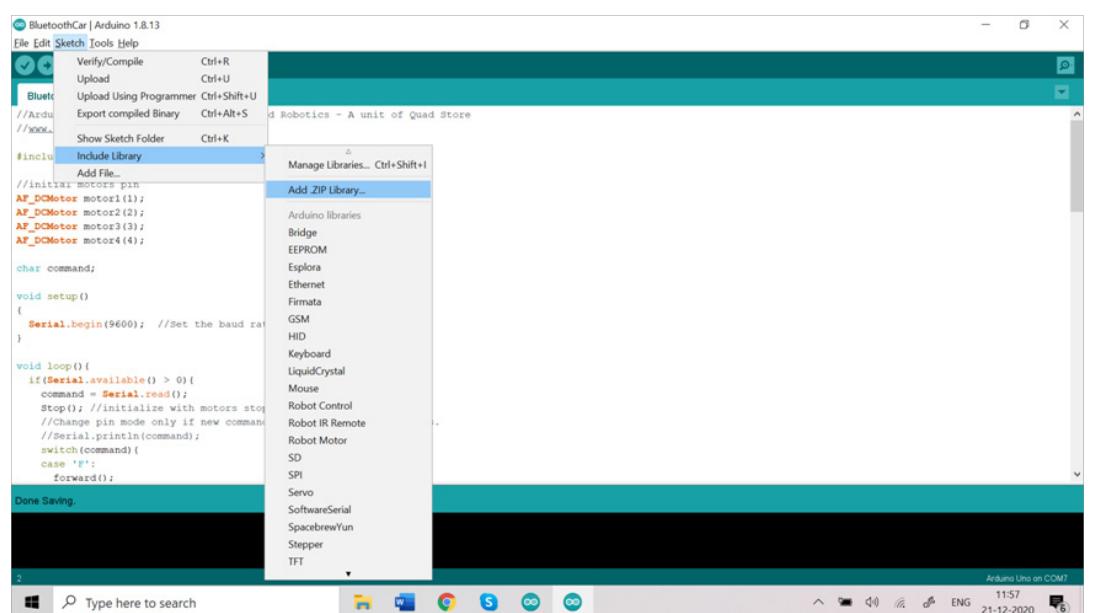
Use the provided USB cable and connect the Uno R3 board to your laptop/desktop computer.

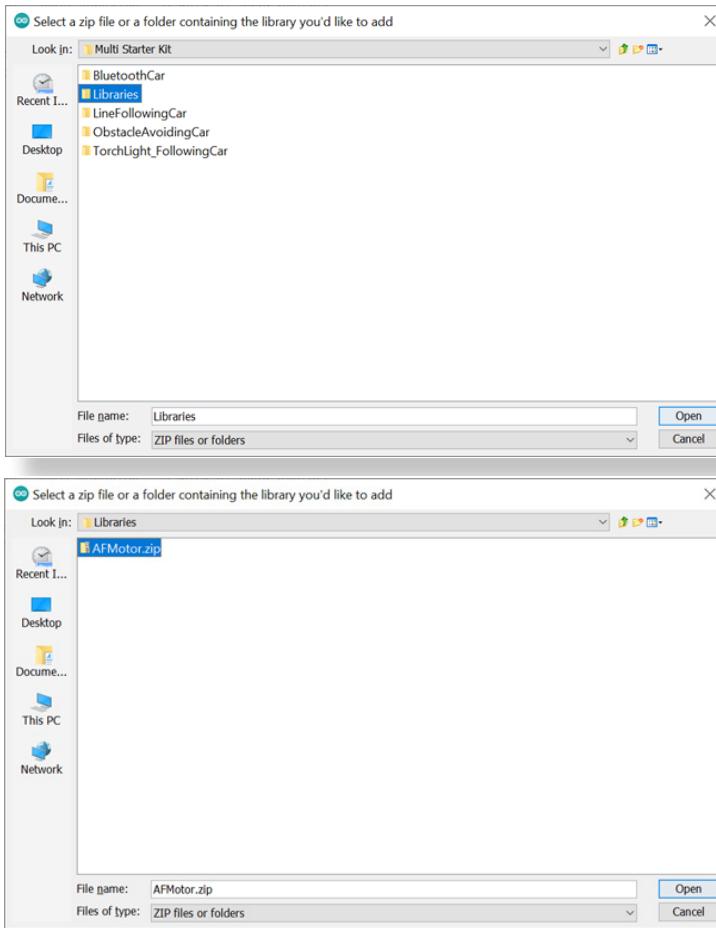
### Installing Libraries:

Open the BluetoothCar.ino code using the Arduino IDE.

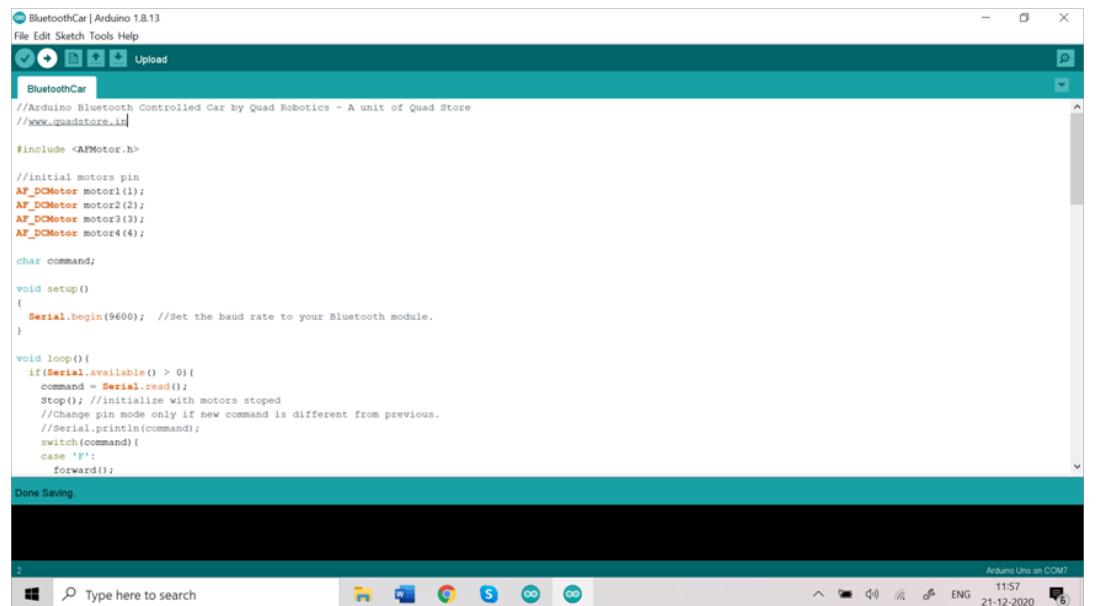
Before uploading the code to the Uno R3 board you need to install the "AFMotor.zip" library.

**NOTE:** If you had already installed this AFMotor.zip library during the previous projects then you can ignore this step and directly compile and upload the code to UNO R3 board. Else follow the below steps to install the library.

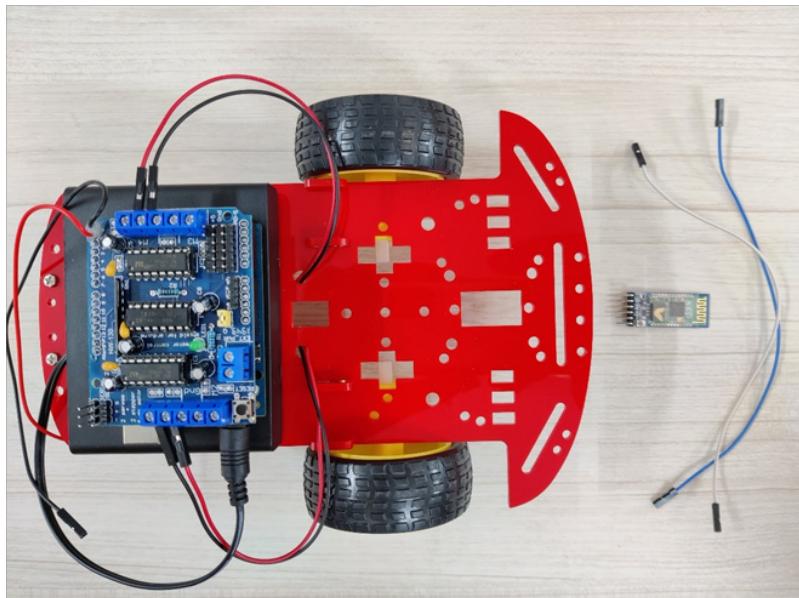




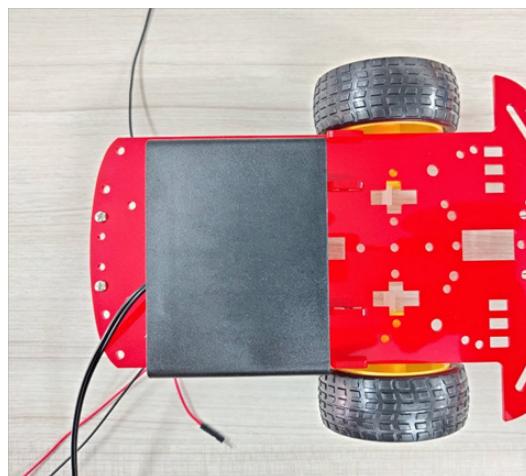
Compile and Upload the code to Uno R3 board.



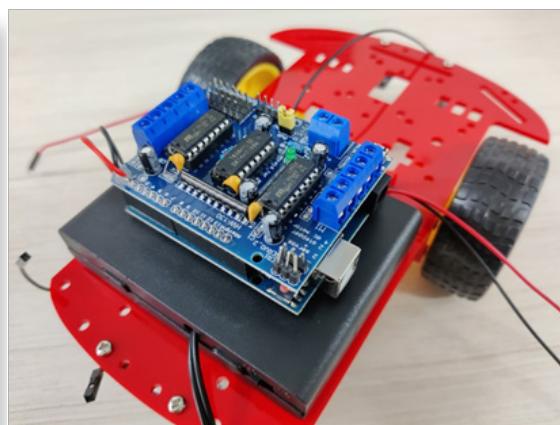
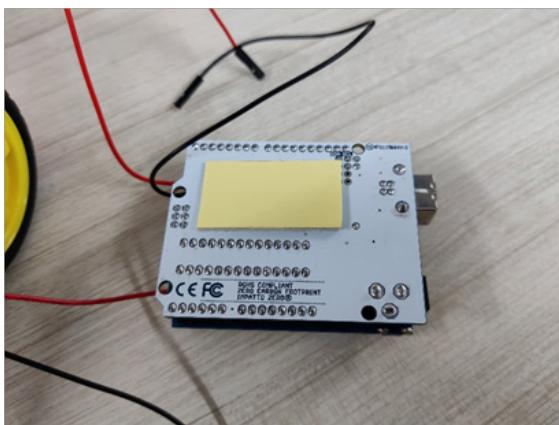
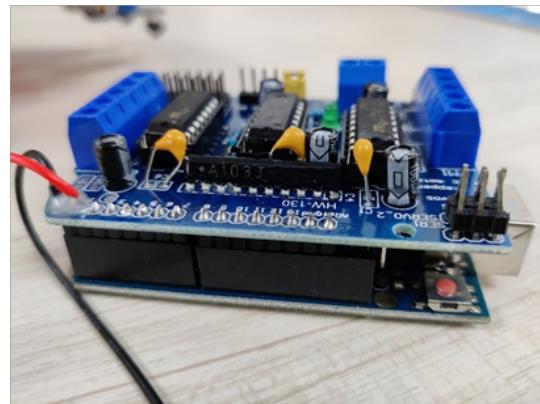
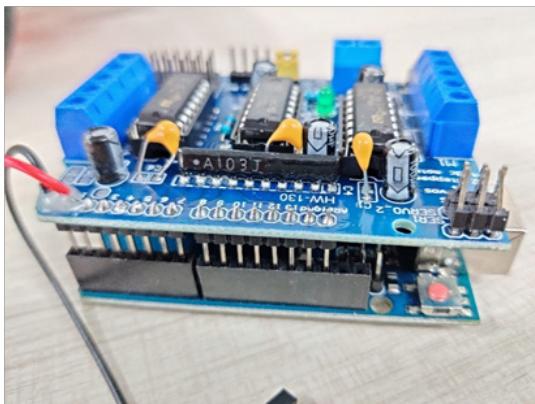
Step-2: Take an assembled 2WD Car chassis. Just in case you are not aware how to assemble the 2WD car chassis please refer to the "2WD Card Chassis Assembly Instruction" section for details.



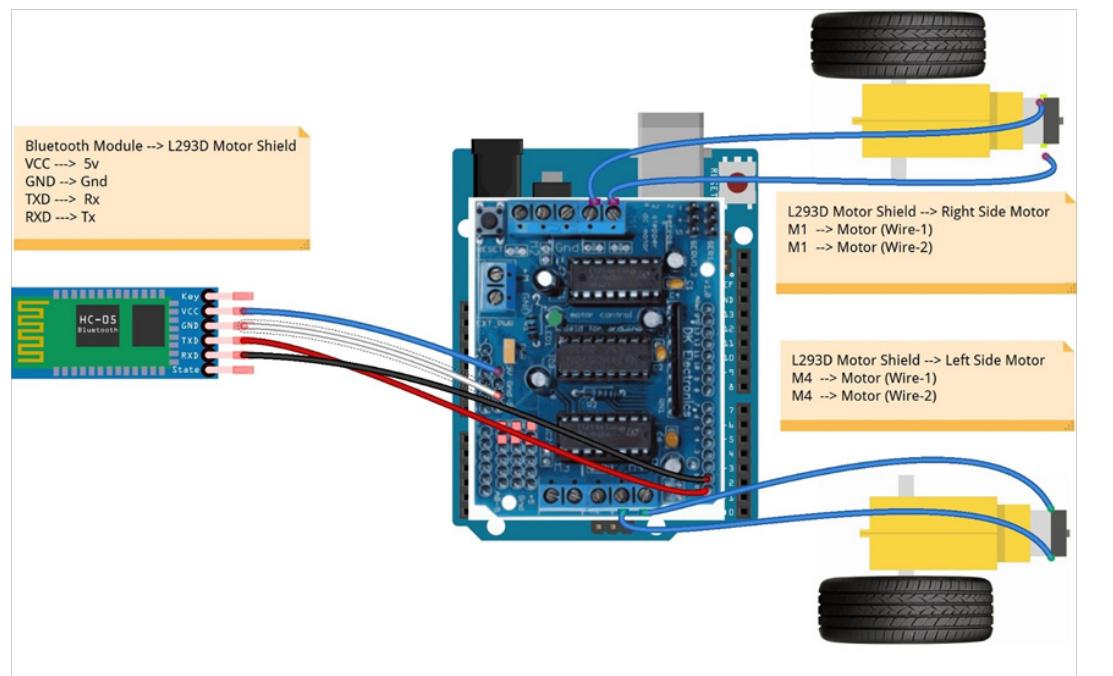
Step-3: Insert 6 x AA batteries in battery holder and place it in the back side of the car chassis using double side stickers.



Step-4: Insert the L293D motor driver shield on top of the Uno R3 board and use a double side tape and fix it on top of the battery holder.



## Step-5: Circuit Connection



Motors to L293D driver shield

## Left Side Motor L293 Shield mounted on Uno R3

Wire-1 (Red) M4

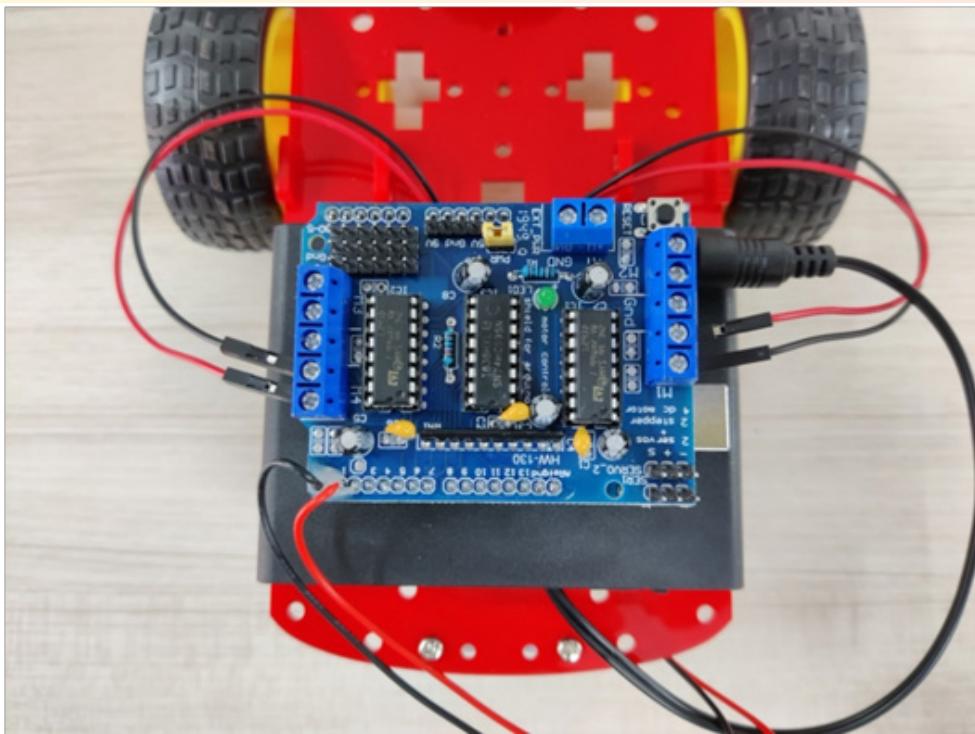
Wire-2 (Black) M4

## Right Side Motor L293 Shield mounted on Uno R3

Wire-1 (Red) M1

Wire-2 (Black) M1

Note: kindly note that there is no polarity in gear motors. Hence the red and black wires provided with the motor is just for reference purpose. They can be interchanged if the motor is required to rotate in different directions.



Bluetooth Module to L293D motor driver shield:

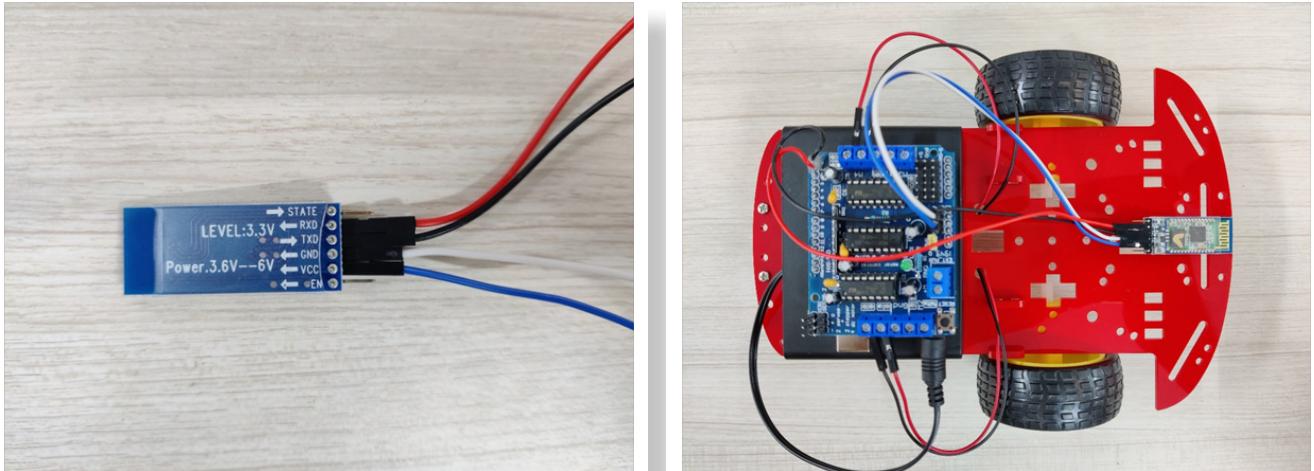
## Bluetooth Module L293 Shield mounted on Uno R3

RX Tx (Pin 1)

TX Rx (Pin 0)

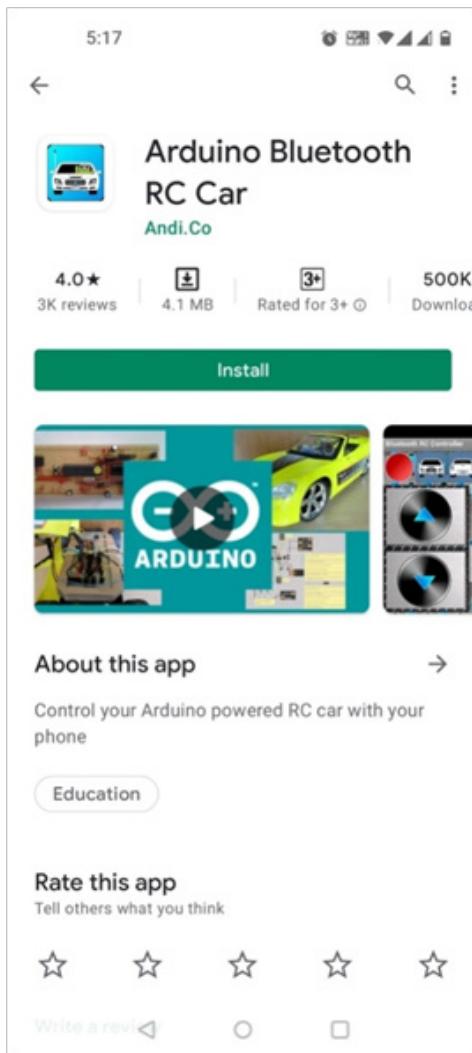
GND Gnd

VCC 5v



Installing the Bluetooth RC controller app from PlayStore in your Android mobile phone:

Go to playstore and search for "Arduino Bluetooth RC Car" and install it.



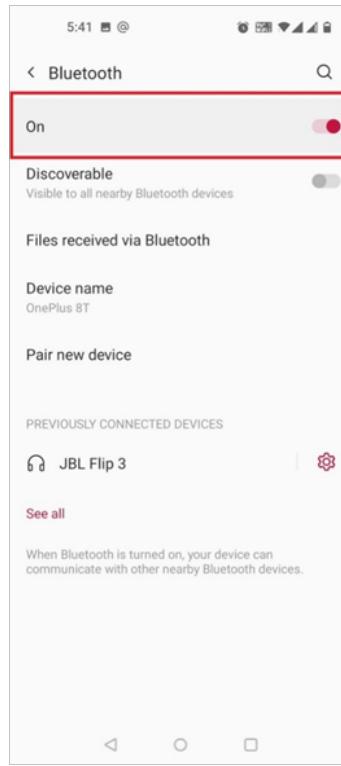
Power on the Car and you would see the RED led in the Bluetooth module blinking. This means the Bluetooth module is waiting for paring with the app.

# Advance Level Projects

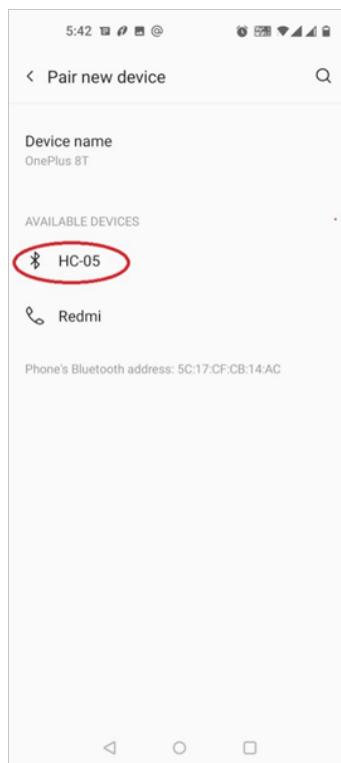
70

In your mobile phone

Go to Settings -> Bluetooth -> Turn on Bluetooth Option

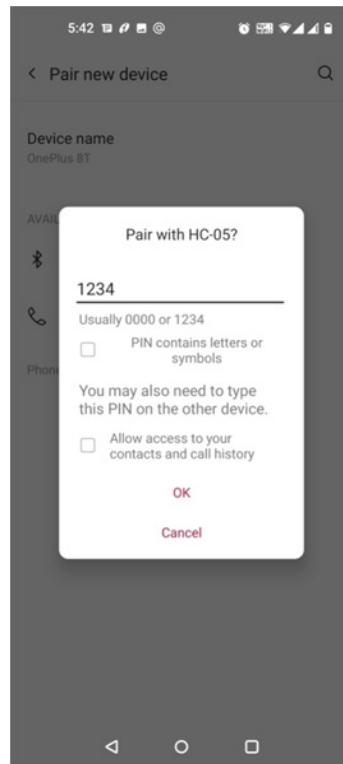


Click "Pair new device" and from the Available devices select "HC-05"



# Advance Level Projects

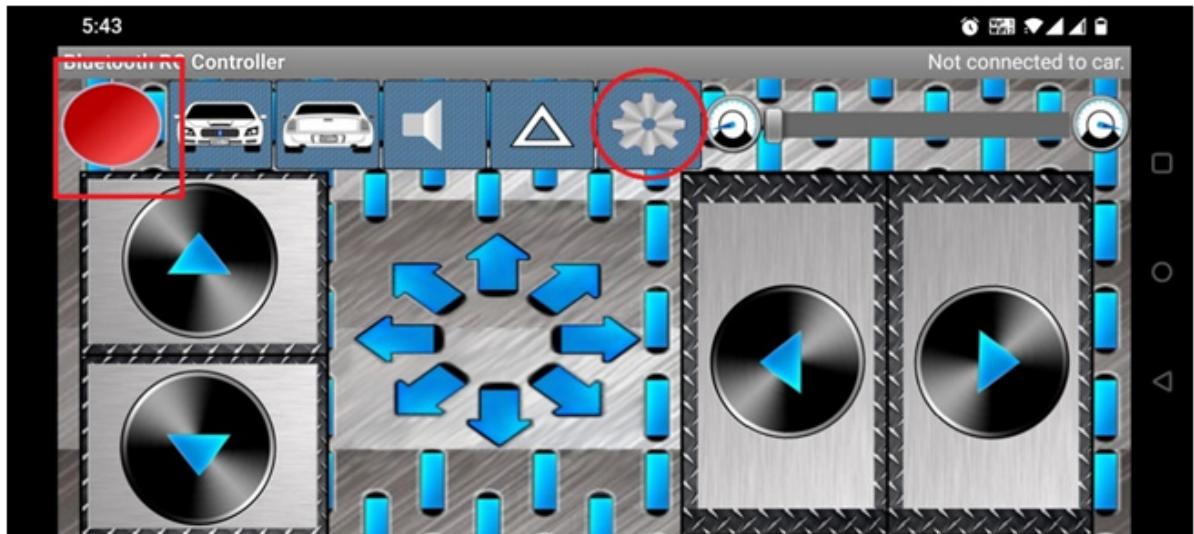
When prompted for password please enter 1234 and click Ok.



Kindly note still your Bluetooth module is NOT yet paired with the app.

Now open the "Arduino Bluetooth RC Car" app. You will see a RED circle indicator blinking on the left corner of the app which indicates that bluetooth module is not yet paired with the app.

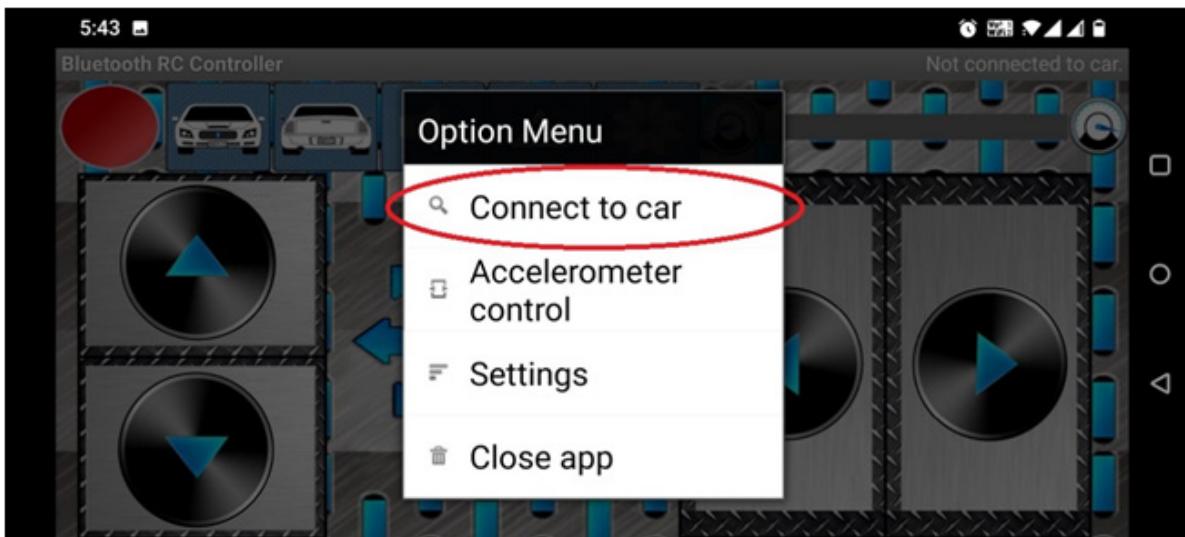
Now Click on the setting wheel icon.



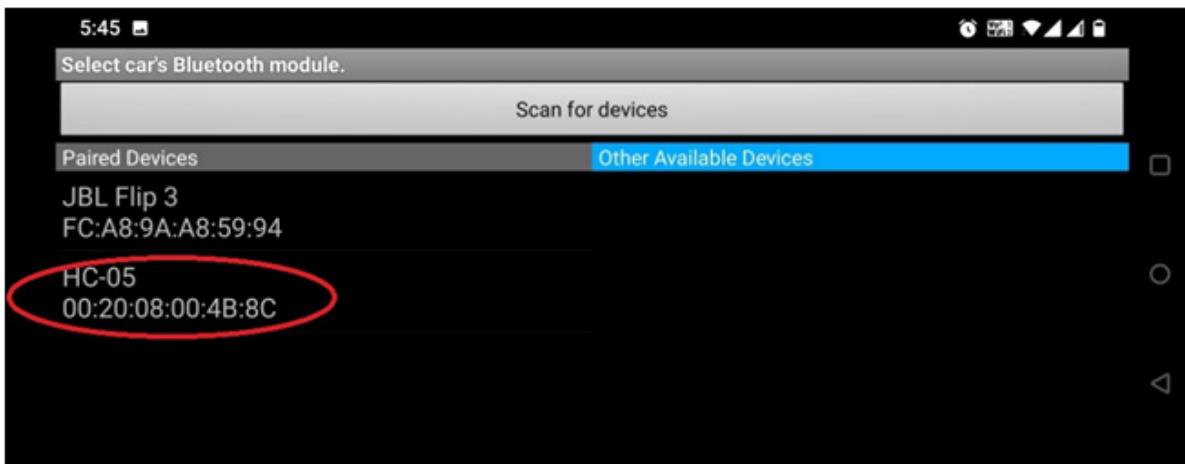
# Advance Level Projects

72

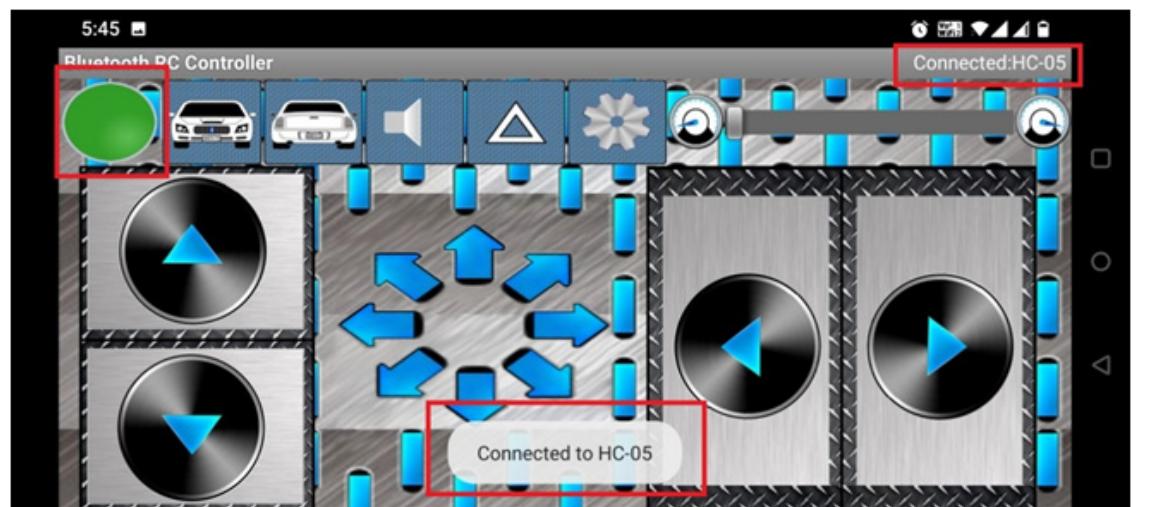
Click "Connect to Car" option

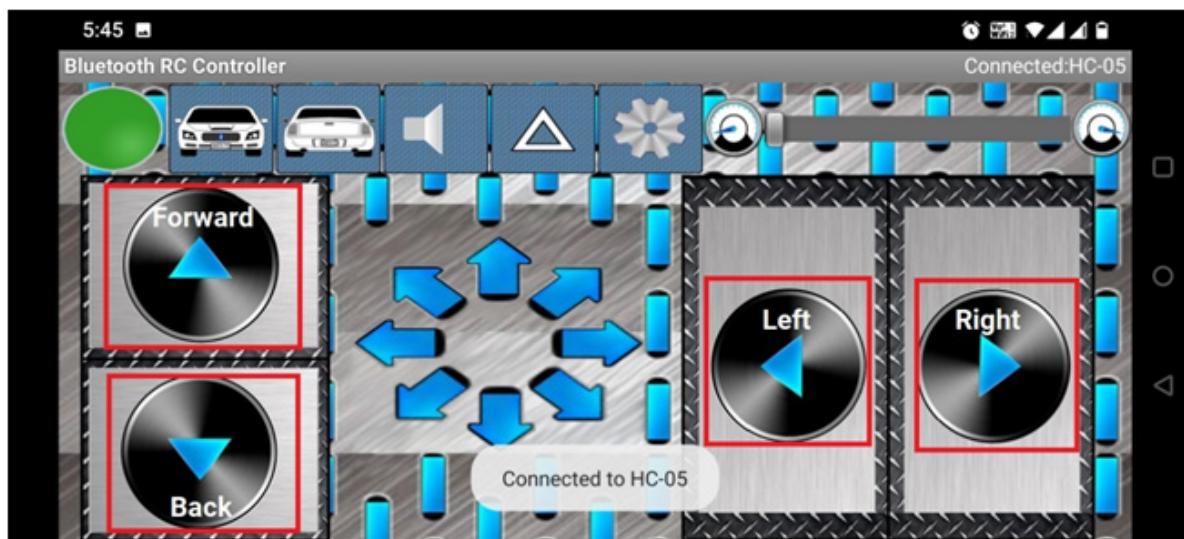


In the list of Bluetooth devices listed select "HC-05" option.



Now you should see the red flashing led turning to SOLD Green and also a text which reads Connected to HC-05 being displayed.





**Result:** Now that the app is connected to your Bluetooth car use the Forward, Backward, Left and Right direction button to make the car move in desired direction.

## PROJECT 29: WIFI ESP8266-01 WITH BLYNK APP

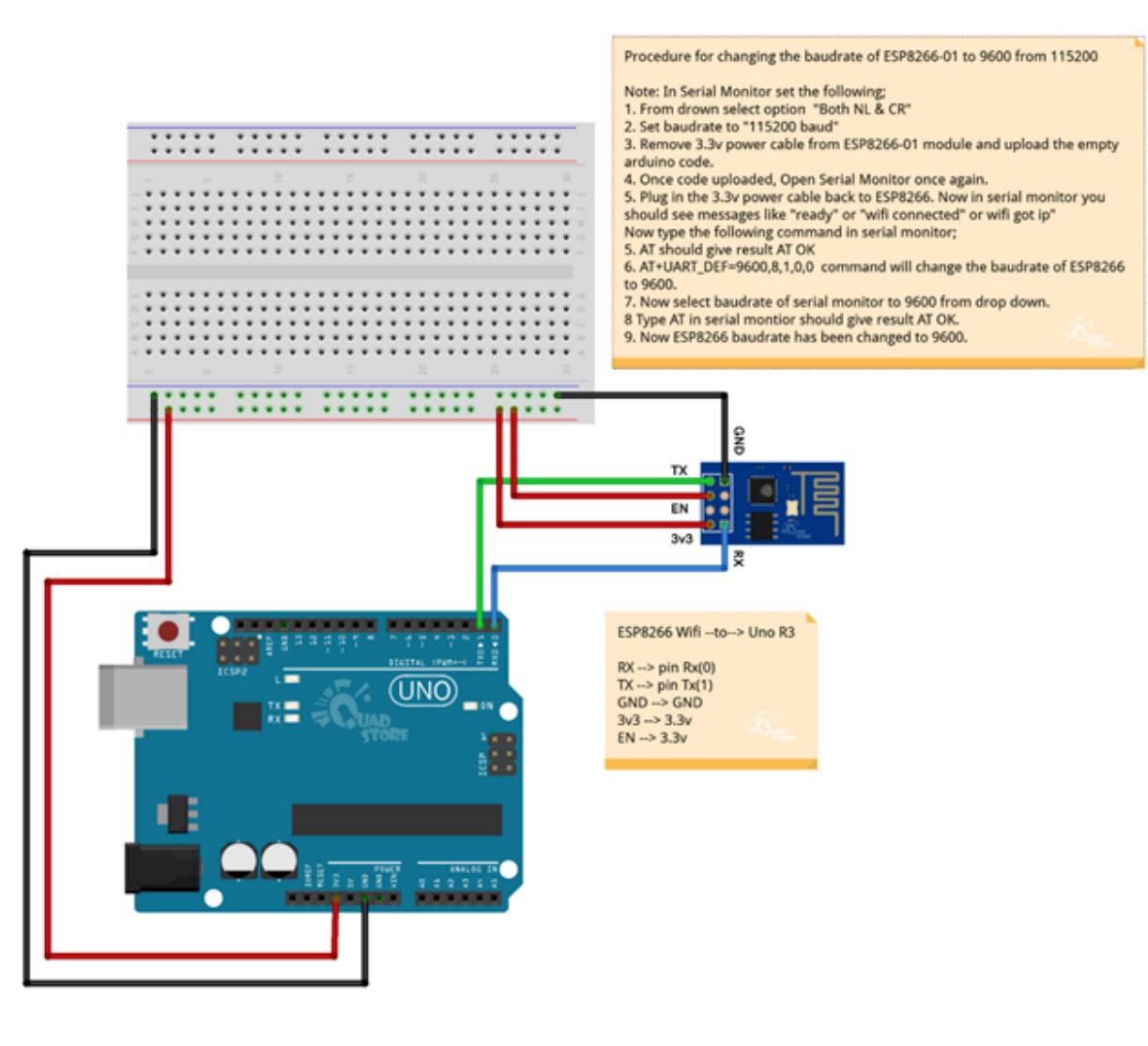
Introduction: Control LED using wifi and blynk app from your mobile.

Parts Required:

- \* Uno R3 board, USB Cable
- \* Breadboard
- \* Few jumper wires
- \* ESP8266-01 module
- \* LED
- \* Resistor – 220ohm
- \* Blynk app need to be installed

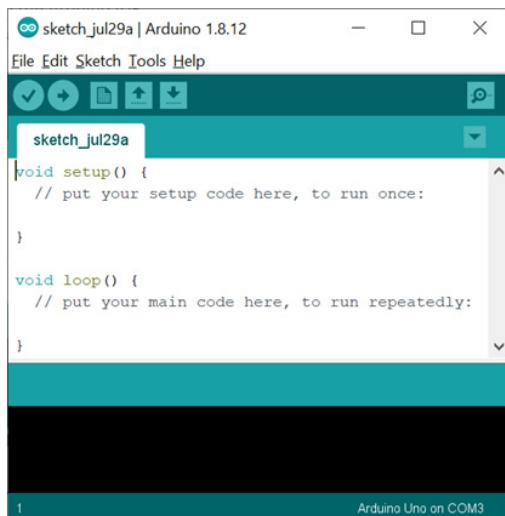
**Note:** You need to change the circuit connection twice for setting up the ESP8266-01 wifi module.

**First circuit connection for changing the baudrate of ESP8266-01 module:** Provide the connection as per the below circuit.

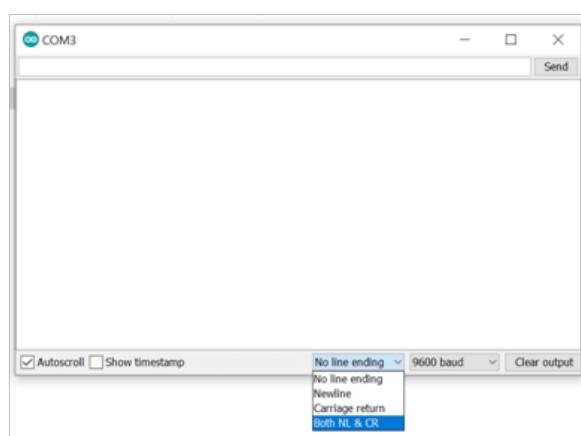


**Action-1: Procedure for changing the baudrate of ESP8266-01 to 9600 from 115200**

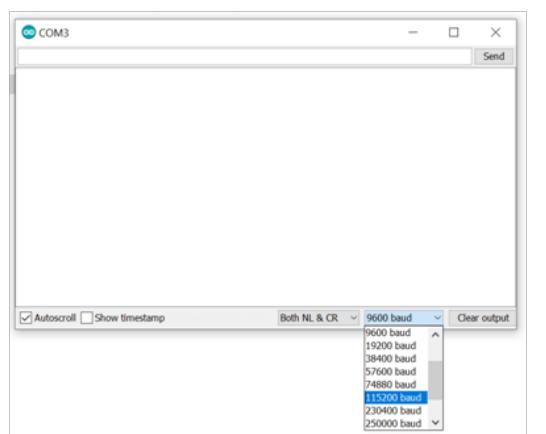
1. Open New Sketch from the Arduino IDE.



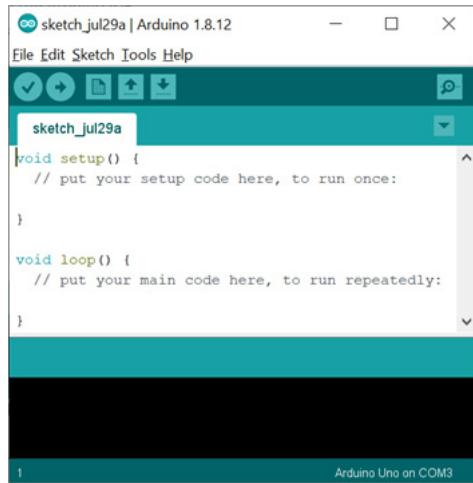
2. Open Serial Monitor and change the following options
3. From dropdown select option "Both NL & CR"



4. Set baudrate to "115200 baud"



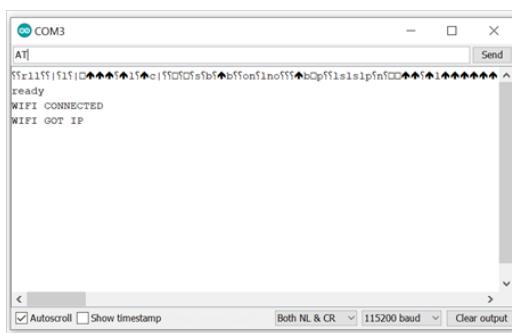
5. Remove **3.3v power cable** from ESP8266-01 module.
6. Upload the New empty Sketch/Code to the Uno R3 board.



7. Once code uploaded, Open Serial Monitor once again.
8. Plug in the **3.3v power cable** back to ESP8266. Now in serial monitor you should see messages like "ready" or "WIFI CONNECTED" or "WIFI GOT IP"



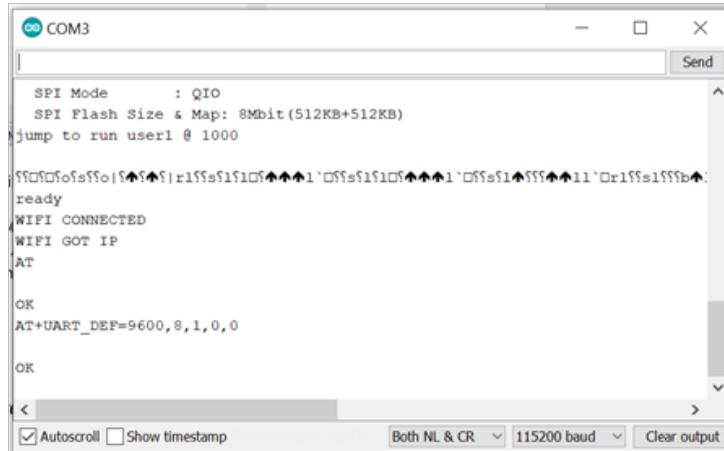
9. Now type the command **AT** in serial monitor and click Send button.



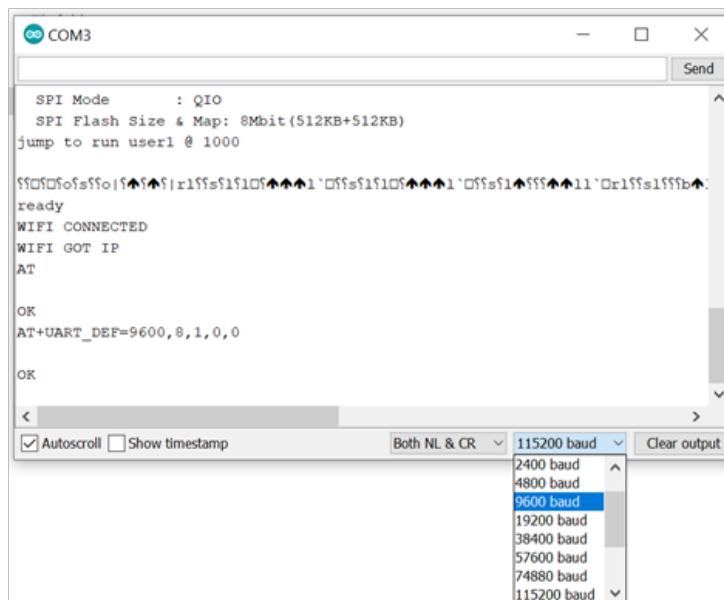
10. AT should give result **AT OK**



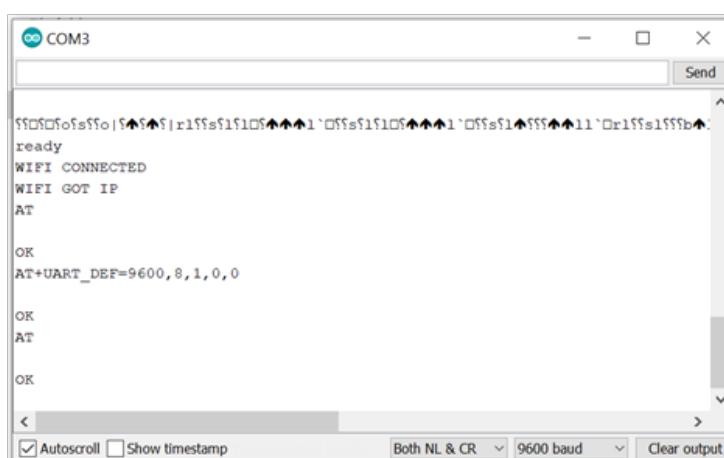
11. Now type command **AT+UART\_DEF=9600,8,1,0,0** command will change the baudrate of ESP8266 to 9600.



12. Now select baudrate of serial monitor to 9600 from drop down.



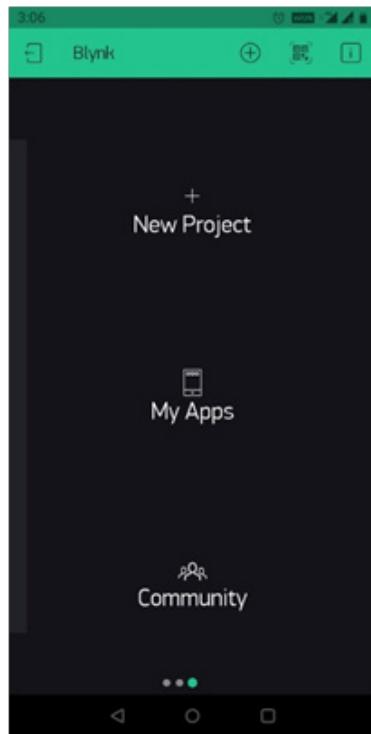
13. Type AT in serial monitor should give result AT OK.



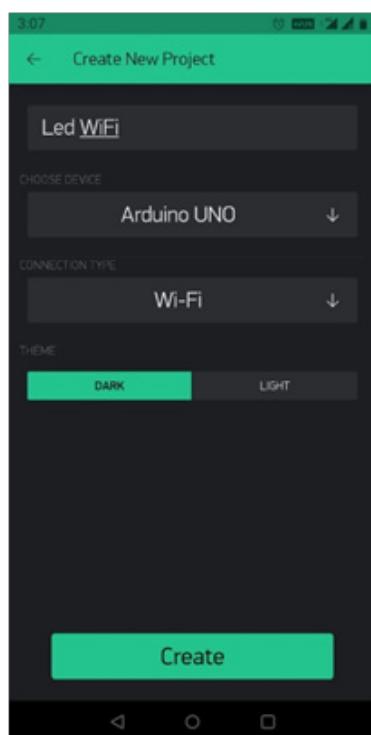
14. Success! Now the baudrate of ESP8266-01 has been changed to 9600.

### Installation and setup of Blynk App:

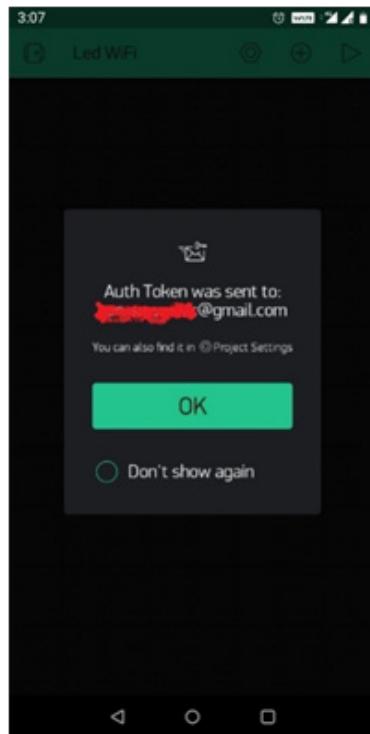
1. Download and install Blynk app from Google Playstore or Appstore
2. Open Blynk App and create "New Project"



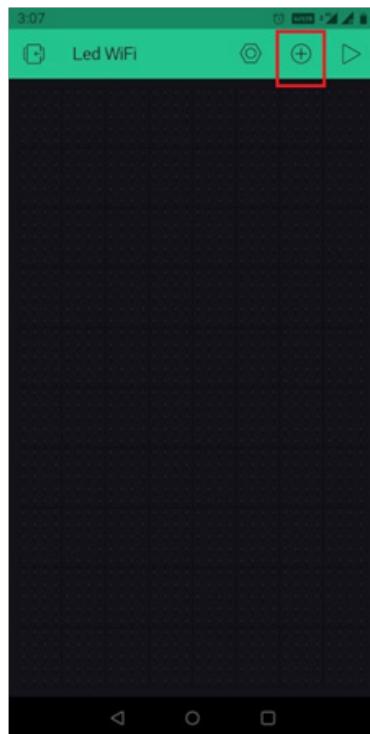
3. Enter the "Project Name" as Led WiFi.
4. Select the "Board Type" as Arduino Uno and click Create



5. Auth token will be generated and sent to your email address.



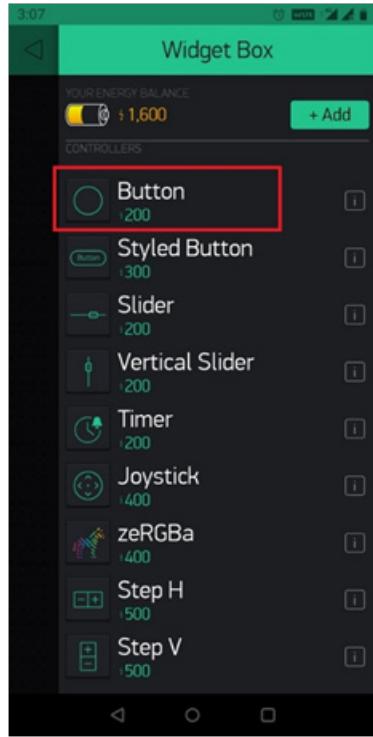
6. Click on the + icon to open widget box



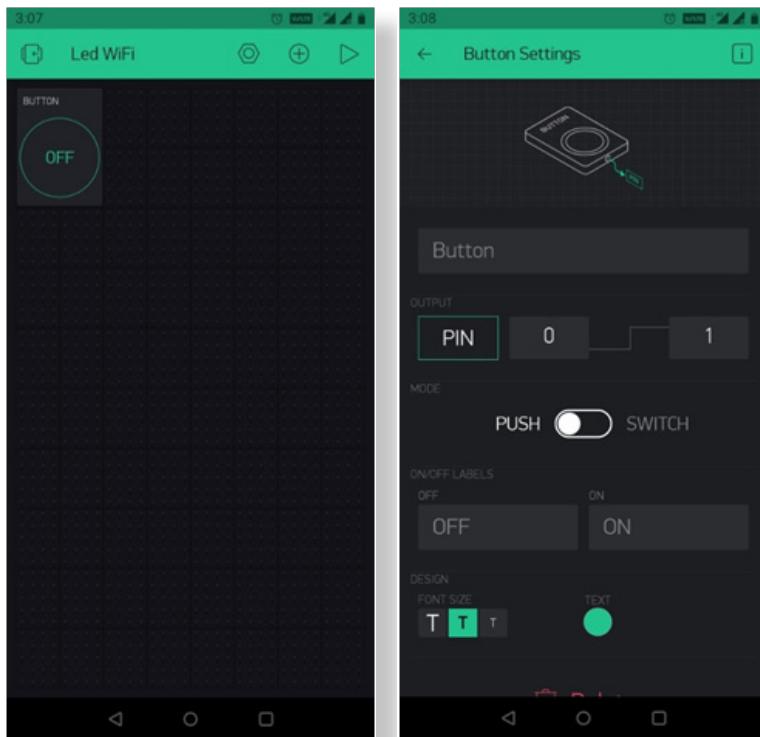
# Advance Level Projects

80

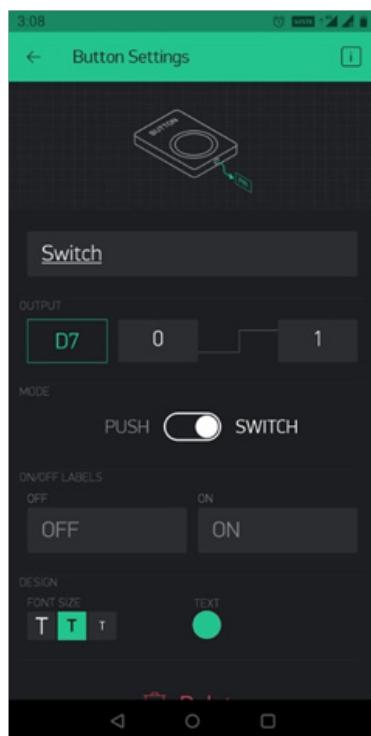
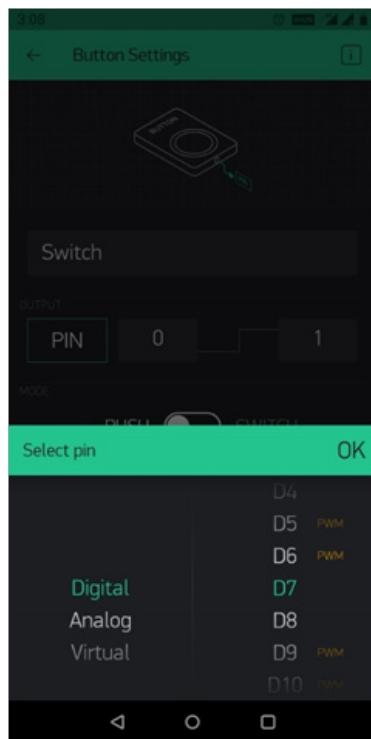
7. Add a new widget "Button"



8. Click on the button widget will take to its setting

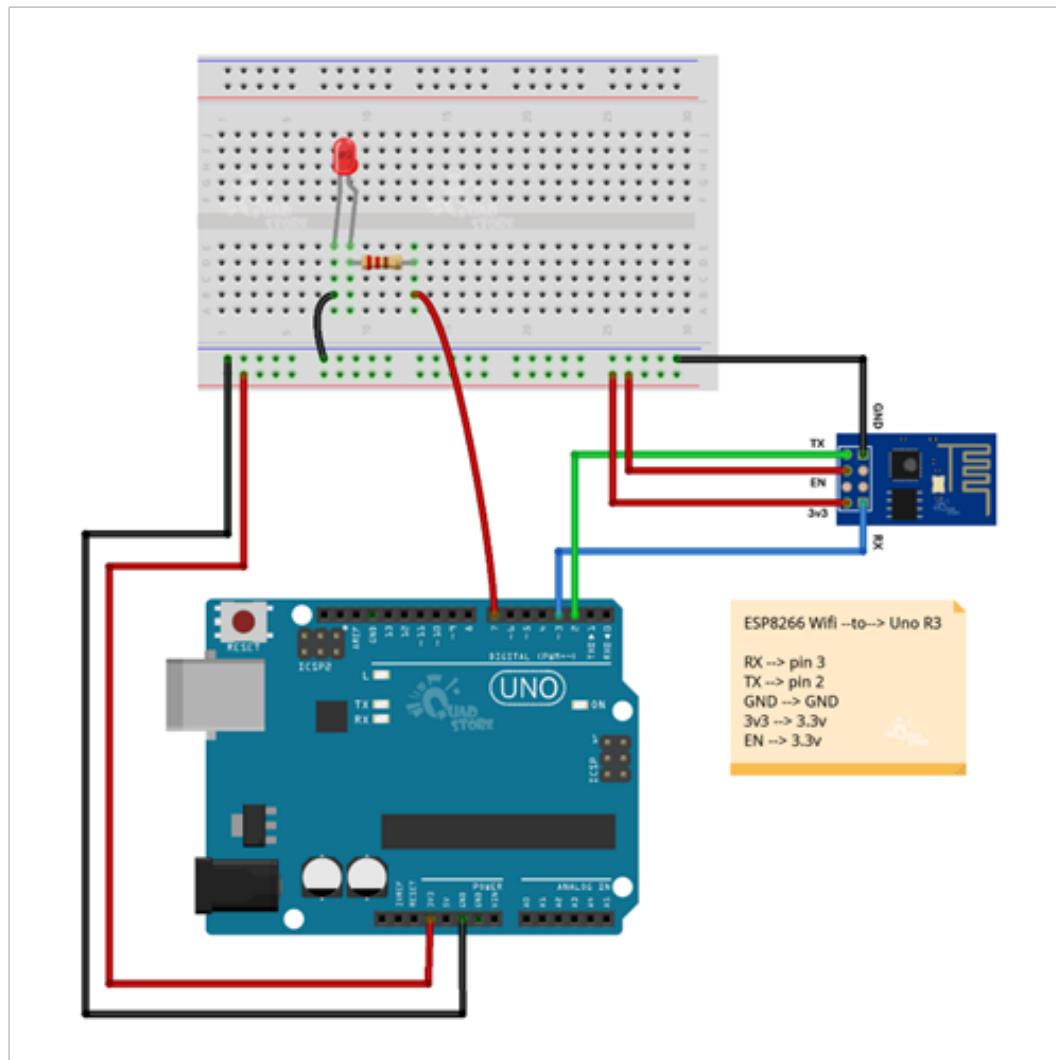


9. Change the PIN to "Digital - pin 7", rename the widget to "Switch" and change the PUSH option to SWITCH and go back.



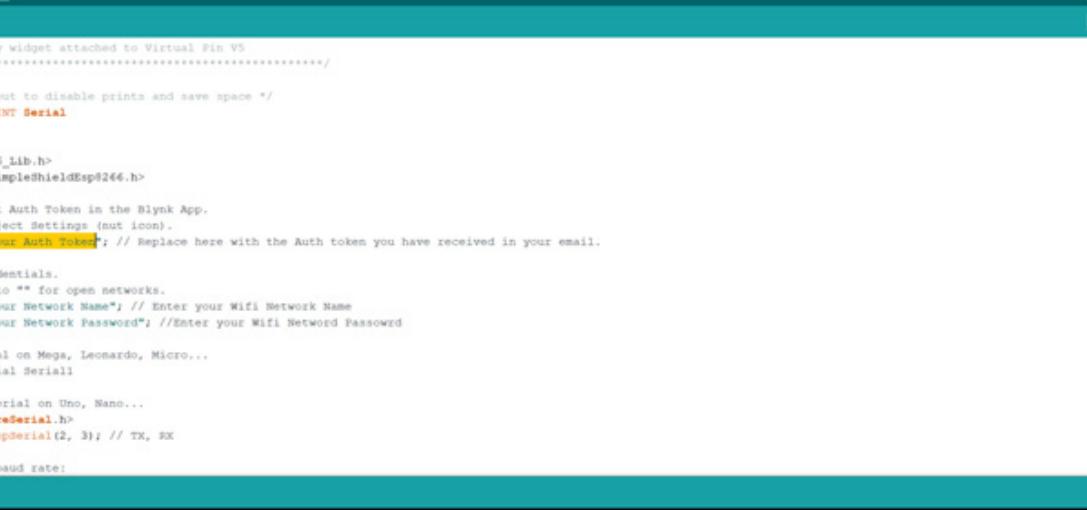
10. Now all the configuration is completed at Blynk app.

**Second circuit connection for using the ESP8266-01 wifi module with your phones Blynk app:** Provide the connection as per the below circuit.



## Update your CODE:

1. **Library:** Install Blynk.zip and BlynkESP8266\_Lib.zip libraries. Otherwise your code will give error message.
2. **Open the code named “wifi\_blynk.ino”**
3. Make the following changes in your CODE to reflect the correct “Auth Token”, “Network Name” and “Network Password”



The screenshot shows the Arduino IDE interface with the following details:

- Title Bar:** WiFi\_Blynk | Arduino 1.8.12
- Menu Bar:** File Edit Sketch Tools Help
- Toolbar:** Standard Arduino toolbar icons.
- Sketch Area:** The code for the WiFi\_Blynk sketch is displayed. It includes comments for Blynk\_PRINT Serial, WiFi credentials (SSID and Password), and hardware serial definitions (SoftwareSerial or HardwareSerial).
- Status Bar:** Shows "Done Saving." at the bottom left and "Arduino Uno on COM3" at the bottom right.

```
// WiFi_Blynk | Arduino 1.8.12
File Edit Sketch Tools Help
WiFi_Blynk

Value Display Widget attached to Virtual Pin V5
*****/* Comment this out to disable prints and save space */
#define BLYNK_PRINT Serial

#include <ESP8266_Lib.h>
#include <BlynkSimpleShieldEsp8266.h>

// You should get Auth Token in the Blynk App.
// Go to the Project Settings (nut icon).
char auth[] = "Your Auth Token"; // Replace here with the Auth token you have received in your email.

// Your WiFi credentials.
// Set password to "" for open networks.
char ssid[] = "Your Network Name"; // Enter your WiFi Network Name
char pass[] = "Your Network Password"; //Enter your WiFi Network Password

// Hardware Serial on Mega, Leonardo, Micro...
// #define ESPSERIAL Serial1

// or Software Serial on Uno, Nano...
#include <SoftwareSerial.h>
SoftwareSerial EspSerial(2, 3); // TX, RX

// Your ESP8266 baud rate:
49
Done Saving.
Arduino Uno on COM3
```

Note: You would have received the Auth Token in your email. Kindly copy and paste it in the code.

4. Next enter your Wifi network "User Name" and "Password".

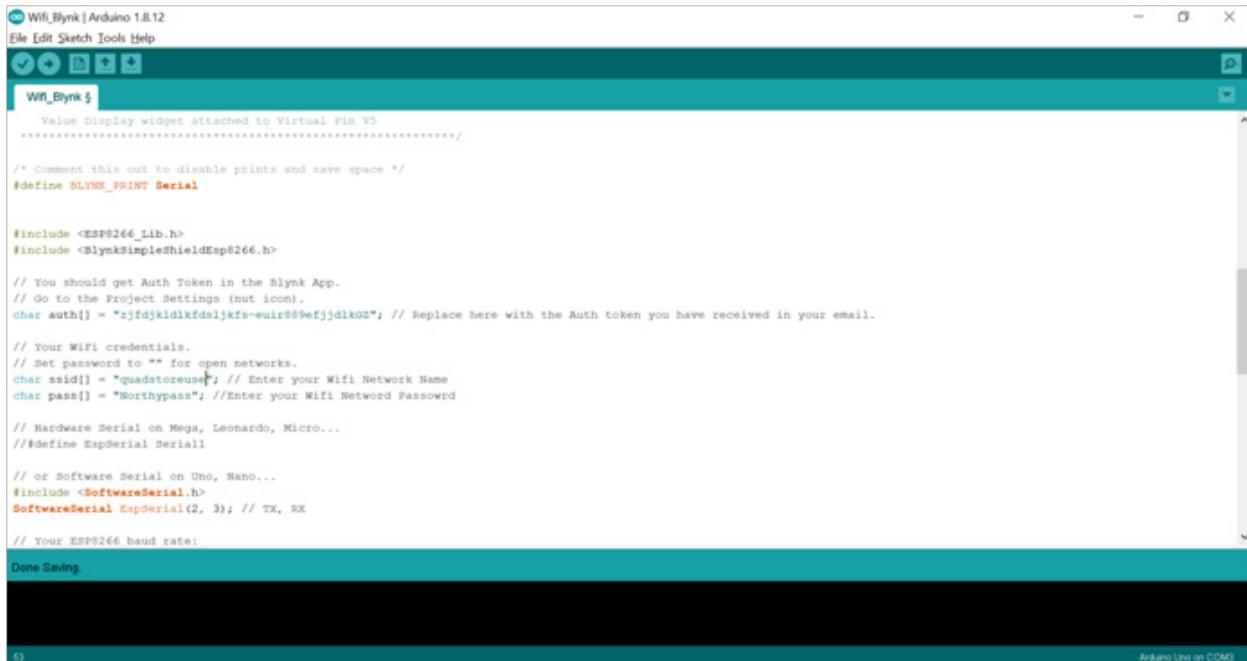


The screenshot shows the Arduino IDE interface with the following details:

- Title Bar:** WiFi\_Blynk | Arduino 1.8.12
- Menu Bar:** File Edit Sketch Tools Help
- Toolbar:** Includes icons for Save, Undo, Redo, Cut, Copy, Paste, Find, Select All, and Print.
- Sketch Name:** WiFi\_Blynk
- Code Area:** Displays the C++ code for the WiFi\_Blynk sketch. The code includes comments for Blynk API keys, WiFi credentials, and serial communication setup. A portion of the code is highlighted in yellow, specifically the line `char auth[] = "rzvEcXtYY5SSf0we8y-eLen5buccbyGZ";`.
- Status Bar:** Shows "Done Saving." at the bottom left and "Arduino Uno on COM3" at the bottom right.



5. Once you have entered the Auth token, User Name and Passowrd you code should look something like this;



The screenshot shows the Arduino IDE interface with the file 'WiFi\_Blynk.ino' open. The code is a Blynk library example for connecting an ESP8266 to a WiFi network and displaying sensor data via Blynk. It includes comments for auth token, WiFi credentials, and serial setup. A message 'Done Saving.' is visible at the bottom.

```
Wifi_Blynk | Arduino 1.8.12
File Edit Sketch Tools Help
WiFi_Blynk.ino
Value Display widget attached to Virtual Pin V5
-----
/* Comment this out to disable prints and save space */
#define BLYNK_PRINT Serial

#include <ESP8266_Lib.h>
#include <BlynkSimpleShieldEsp8266.h>

// You should get Auth Token in the Blynk App.
// Go to the Project Settings (nut icon).
char auth[] = "zjfdjklidkfdsljkfs-euir@@efjjdkGZ"; // Replace here with the Auth token you have received in your email.

// Your WiFi credentials.
// Set password to "" for open networks.
char ssid[] = "quadstoreuse"; // Enter your WiFi Network Name
char pass[] = "Northypass"; //Enter your WiFi Network Password

// Hardware Serial on Mega, Leonardo, Micro...
// #define EspSerial Serial1

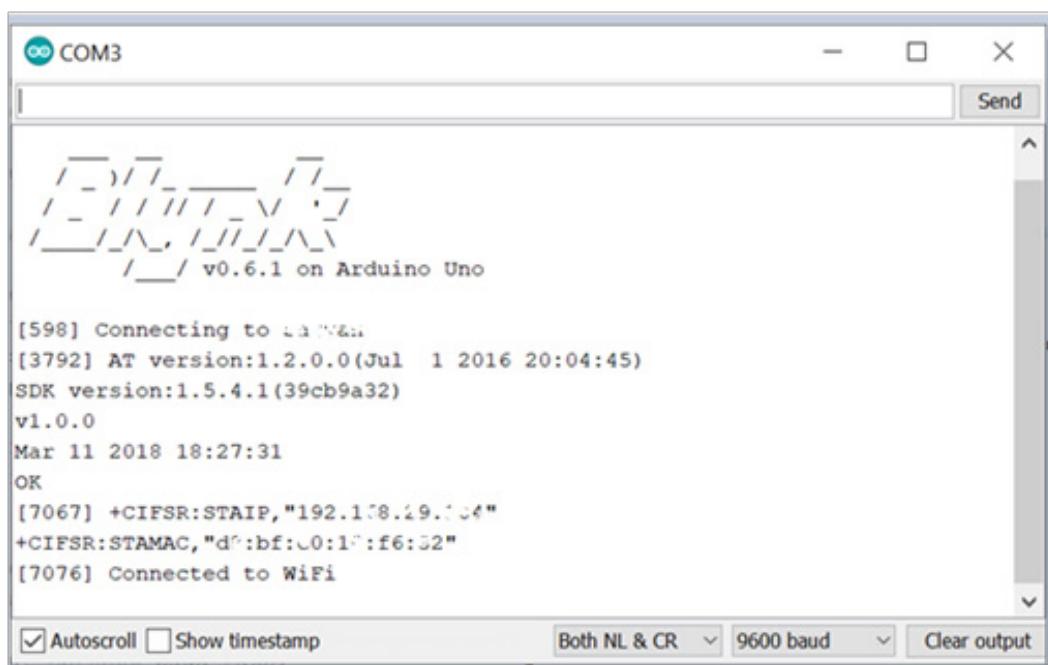
// or Software Serial on Uno, Nano...
#include <SoftwareSerial.h>
SoftwareSerial EspSerial(2, 3); // TX, RX

// Your ESP8266 baud rate:
Done Saving.

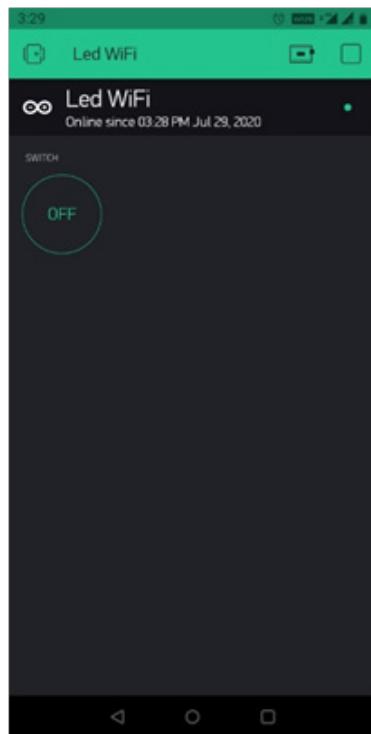
53
Arduino Uno on COM3
```

6. Now perform the below steps to get connected with the wifi module:

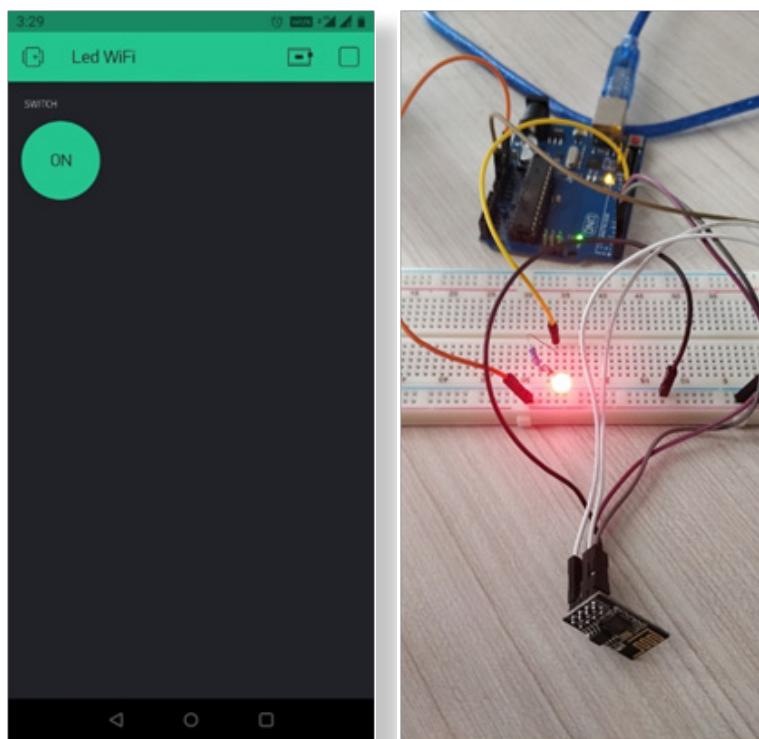
7. Remove 3.3v power cable from ESP8266-01 module and then upload the updated code.
8. Plug in the 3.3v power cable back to ESP8266-01 module.
9. Open "Serial Monitor" and check if the ESP8266-01 module is getting connected to your WiFi network successfully.
10. Ensure the baudrate is at 9600 then you should see like the screen below where the WiFi module gets connected.



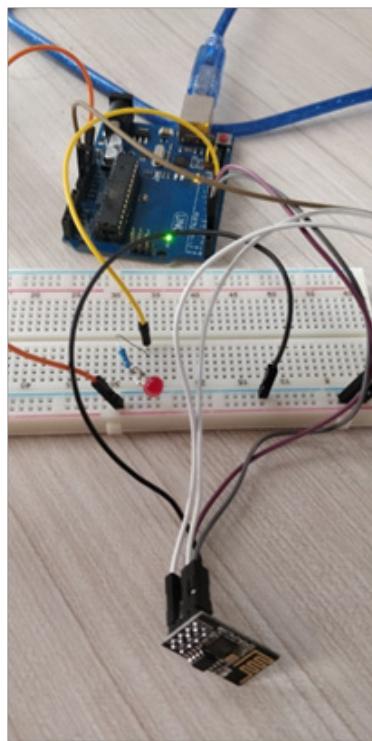
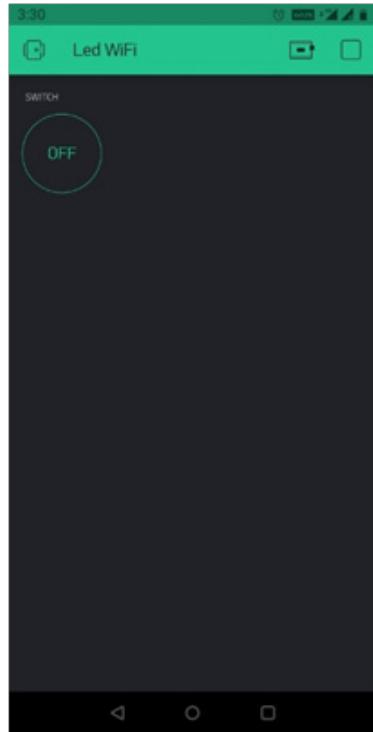
11. Open Blynk app and you should see the project getting connected to your Uno R3 board through wifi automatically.



12. **Output:** Click ON and OFF button through Blynk app to control the LED.  
13. When ON button is clicked you would see the LED turn ON through Wifi.



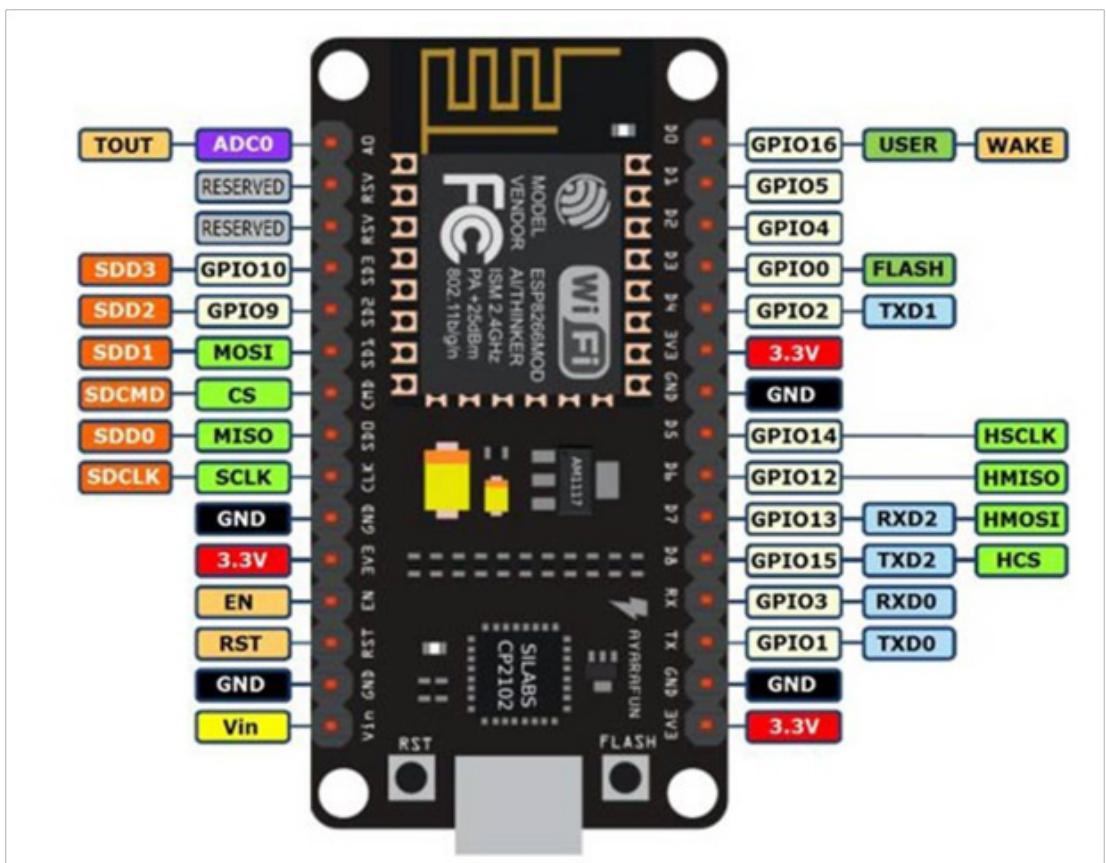
Similarly, when OFF button is clicked the LED turns OFF through Wifi.



## PROJECT 30: INTRODUCTION TO NODEMCU ESP8266

The Node MCU is an open source firmware and development kit that helps you to prototype your IoT product with Arduino IDE or in few Lau script lines.

It includes firmware which runs on the ESP8266 Wi-Fi SoC. And hardware which is based on the ESP-12 module. In this tutorial we explain how to use NodeMCU with Arduino IDE



### How to Connect NodeMCU with Arduino IDE

- \* Open up the Arduino IDE.
  - \* Go to File -> Preferences -> Additional Boards Manager URLs: [http://arduino.esp8266.com/stable/package\\_esp8266com\\_index.json](http://arduino.esp8266.com/stable/package_esp8266com_index.json) -> click OK
  - \* Close the IDE and open it up again.
  - \* Go to Tools -> Board (where you'd select your version of Arduino) -> Boards Manager, find the ESP8266 and click Install. You now should be able to use the ESP8266 as an Arduino.
  - \* Simply select the NODEMCU 1.0 as your board with Port and you should be ready to code.
- Now, with ESP8266 board installed to Arduino IDE, we can program NodeMCU using Arduino IDE directly.



# Thank You

