QUAD STORE

Ultimate kit
for Arduino

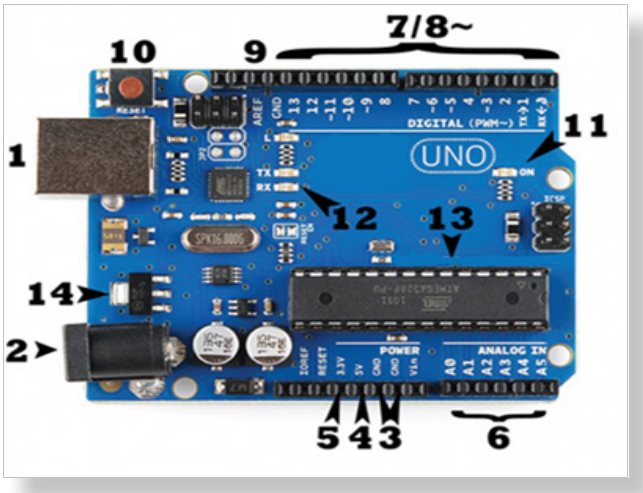
www.quadstore.in

Table of contents

Table of contents

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 conjuction 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.

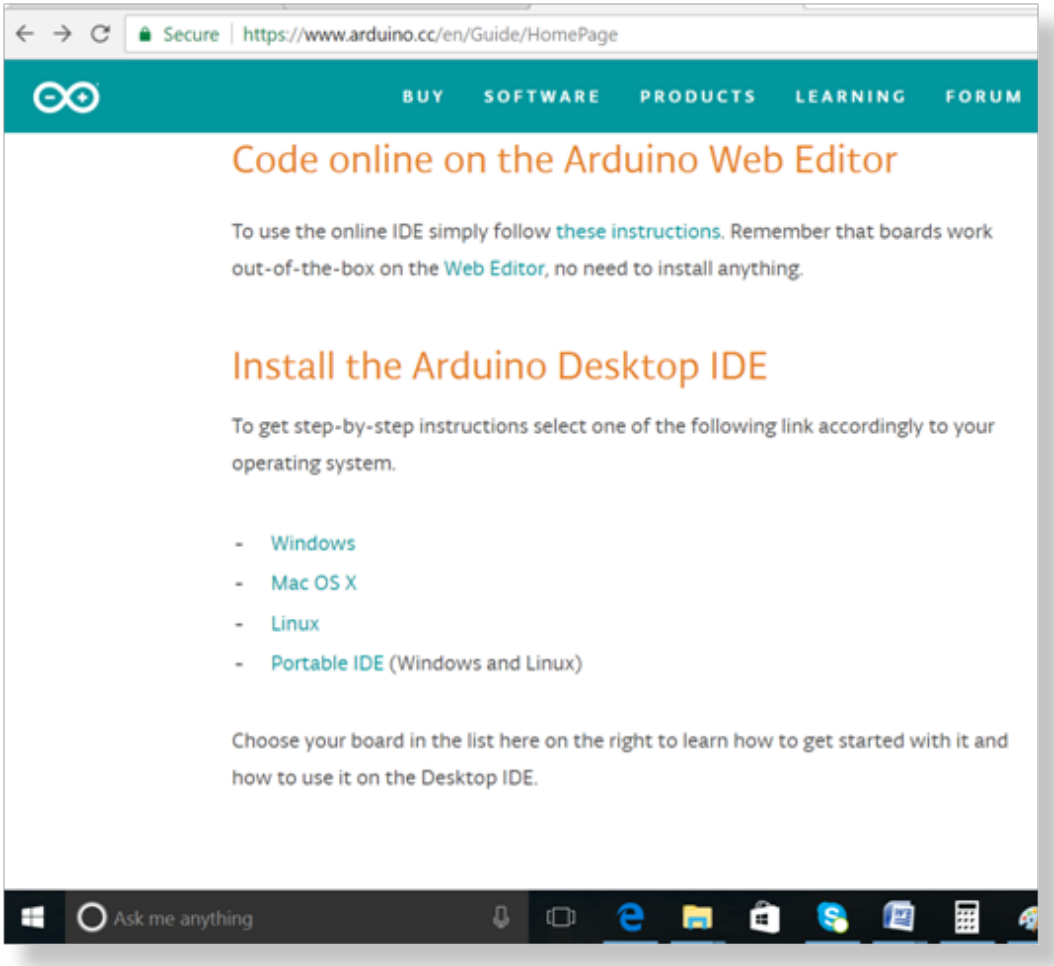
Project 1: Installing Arduino IDE and uploading code

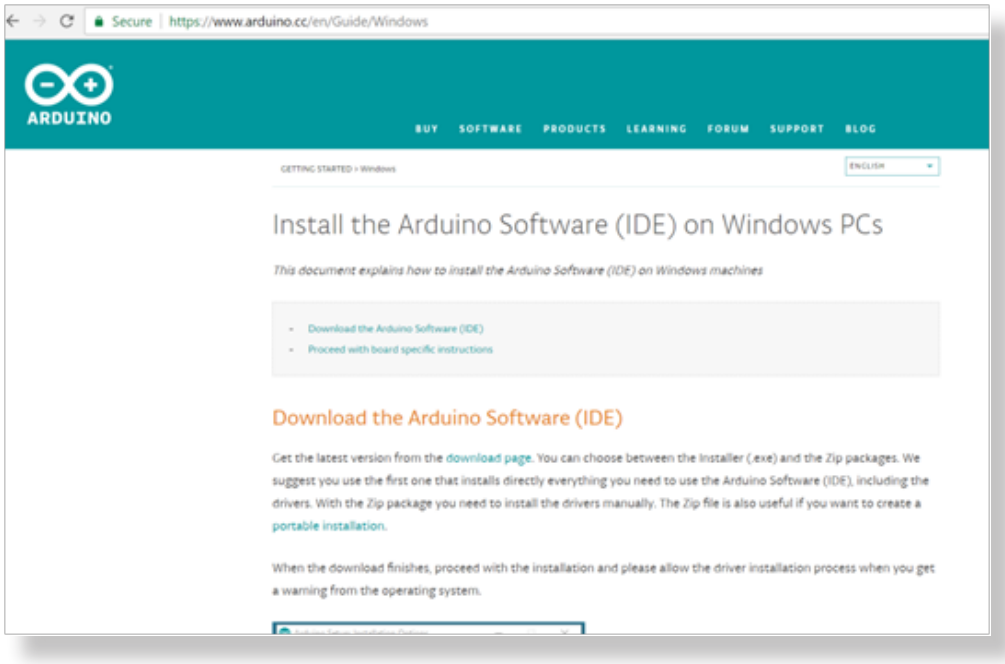
STEP-1:Download the Arduino IDE (Integrated Development Environment)

Access the Internet:

In order to get your Arduino up and running, you'll need to download some software first from www.arduino.cc (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 favourite browser and type in the following URL into the address bar:

www.arduino.cc/en/Main/Software

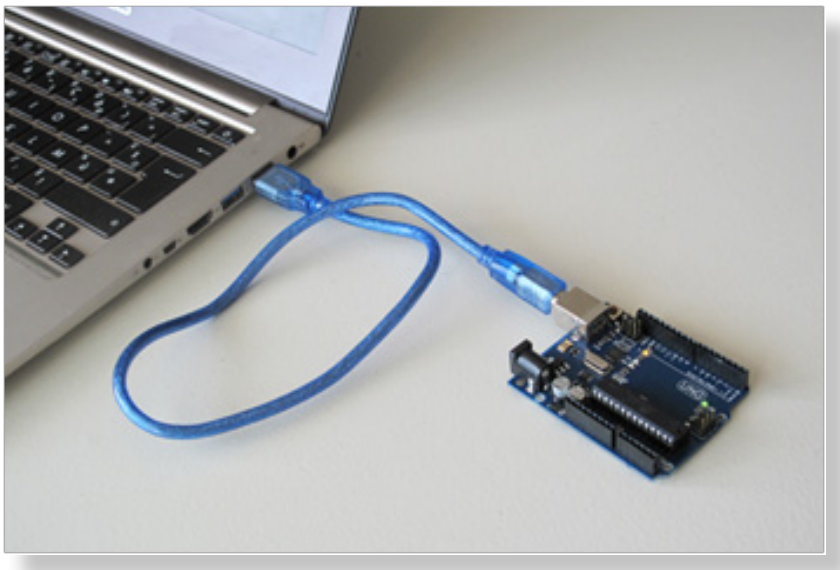




For different operating system platforms, the way of using Arduino IDE is different. Please refer to the following links: Windows User: http://www.arduino.cc/en/Guide/Windows Mac OS X User:http://www.arduino.cc/en/Guide/MacOSX Linux User:http://playground.arduino.cc/Learning/Linux For more detailed information 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.



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

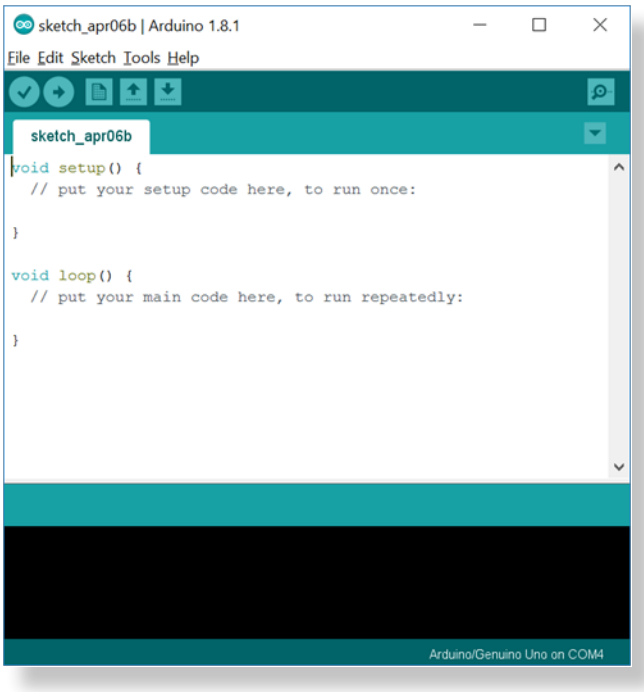
Linux Installation Process:

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.

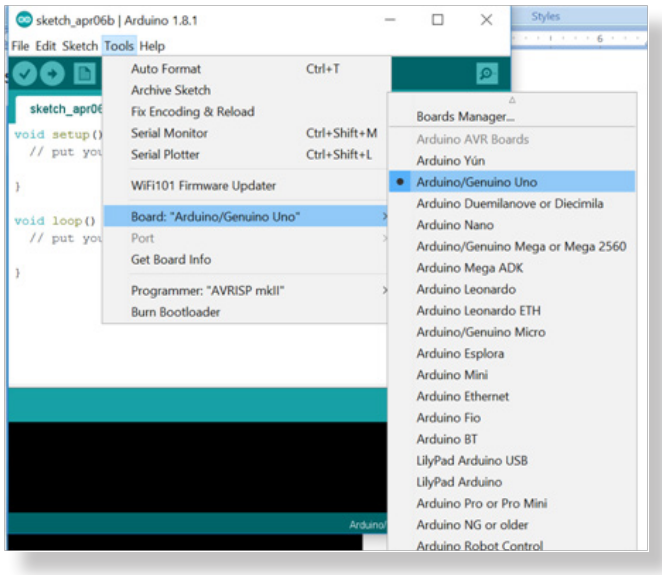
Save

Saves your sketch.

Serial Monitor

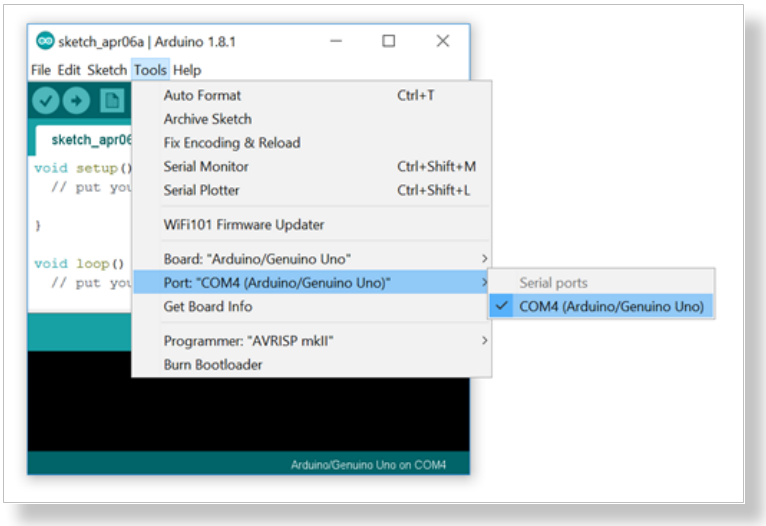
Opens the **serial monitor**.

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.

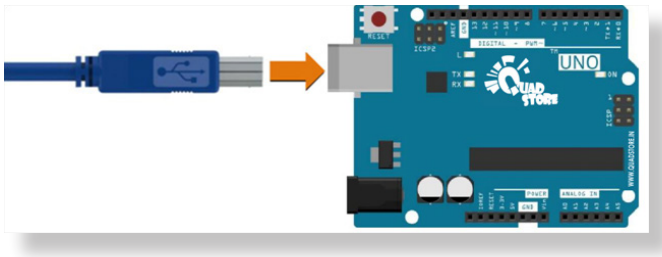


Mac OS: Select the serial device of the Arduino board from the Tools > Serial Port menu. On the Mac, this should be something with /dev/tty.usbmodem (for the Uno or Mega 2560) or /dev/tty.usbserial (for older boards) in it.

Linux: http://playground.arduino.cc/Learning/Linux

How to upload the code to Uno R3 board ?

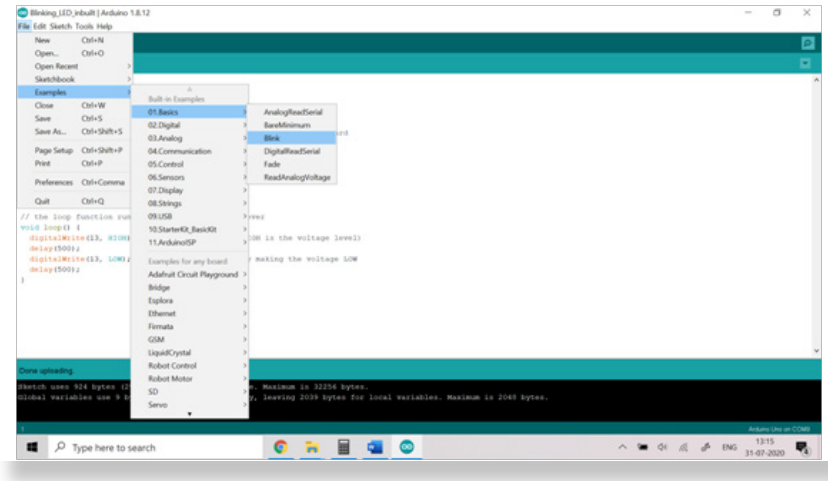
First connect your uno R3 board using usb cable to your laptop or computer.



If you already have arduino code in your system folders 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.

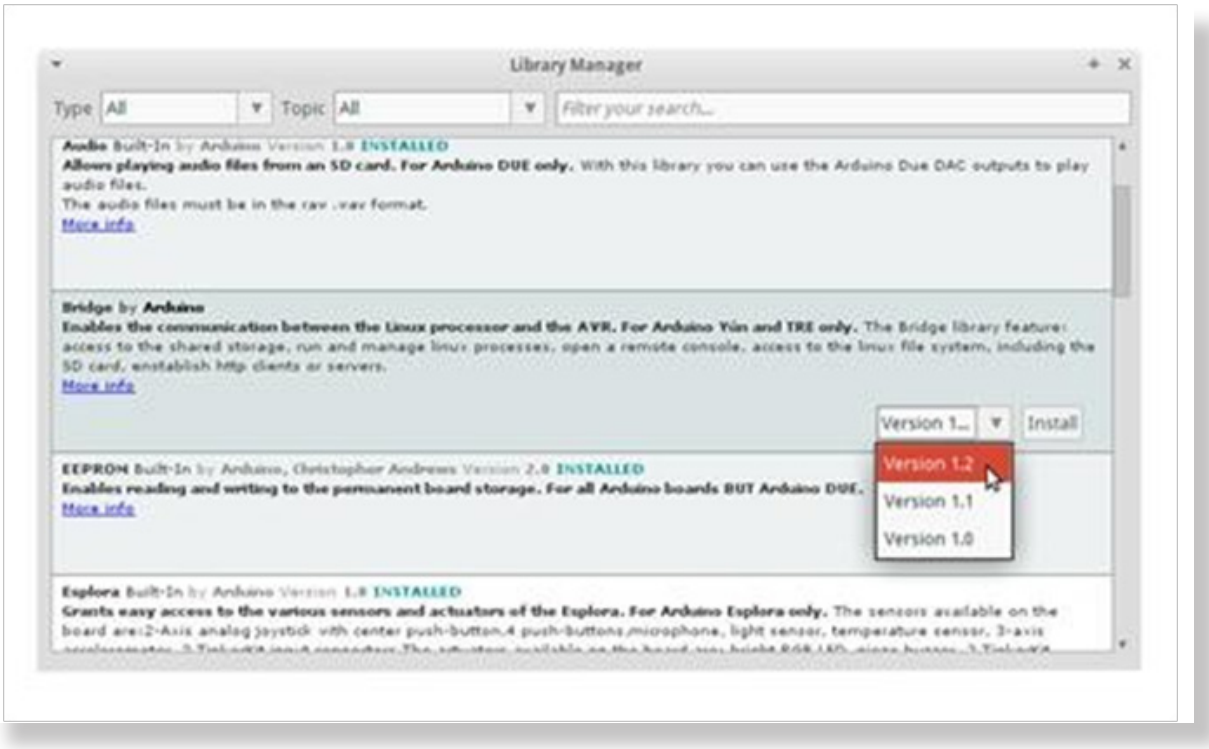
How to install libraries?

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

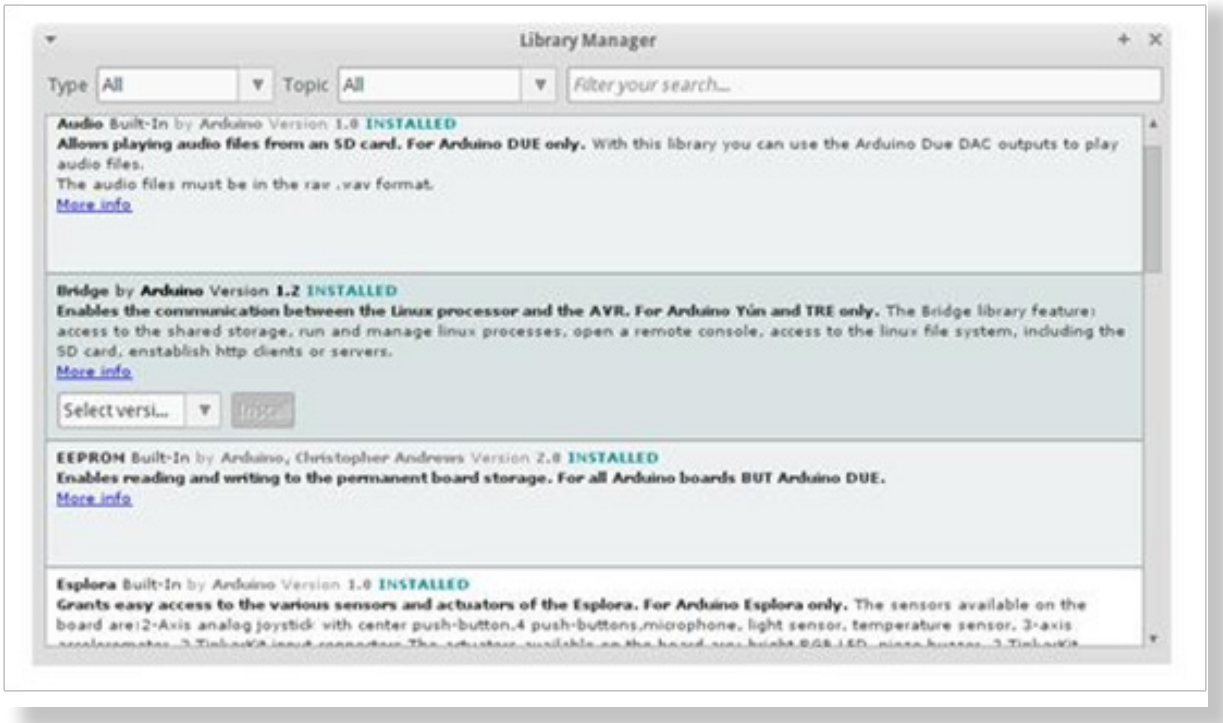
Method-1: Using library manager

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 orready for installation. In this example, we will install the Bridge library. Scroll down the list to findit,then select the version of the library you want to install. Sometimes,onlyoneversionof the library is available. If the version selection menu does not appear, don't worry; it is normal.



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.

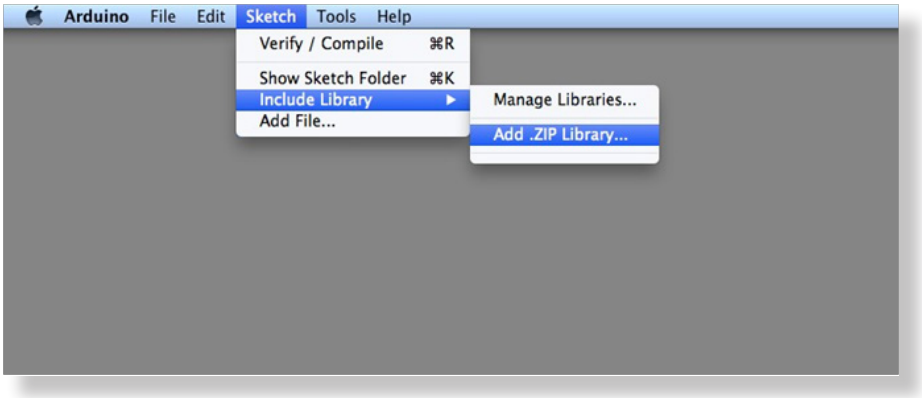


You can now find the new library available in the Include Library menu.

METHOD-2: IMPORTING A .ZIP LIBRARY
(RECOMMENDED FOR ALL THE PROJECTS WHICH WE USE)

Libraries are often distributed as a ZIPfileorfolder. 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 itas-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 drop-down 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 in to it all the files that were in the ZIP file, like ArduinoParty.cpp andArduinoParty.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 nestedinanextrafolder. Forexample:Documents\Arduino\libraries\ArduinoParty.cppand Documents\Arduino\libraries\ArduinoParty\ArduinoParty\ArduinoParty.cpp won'twork.)

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.

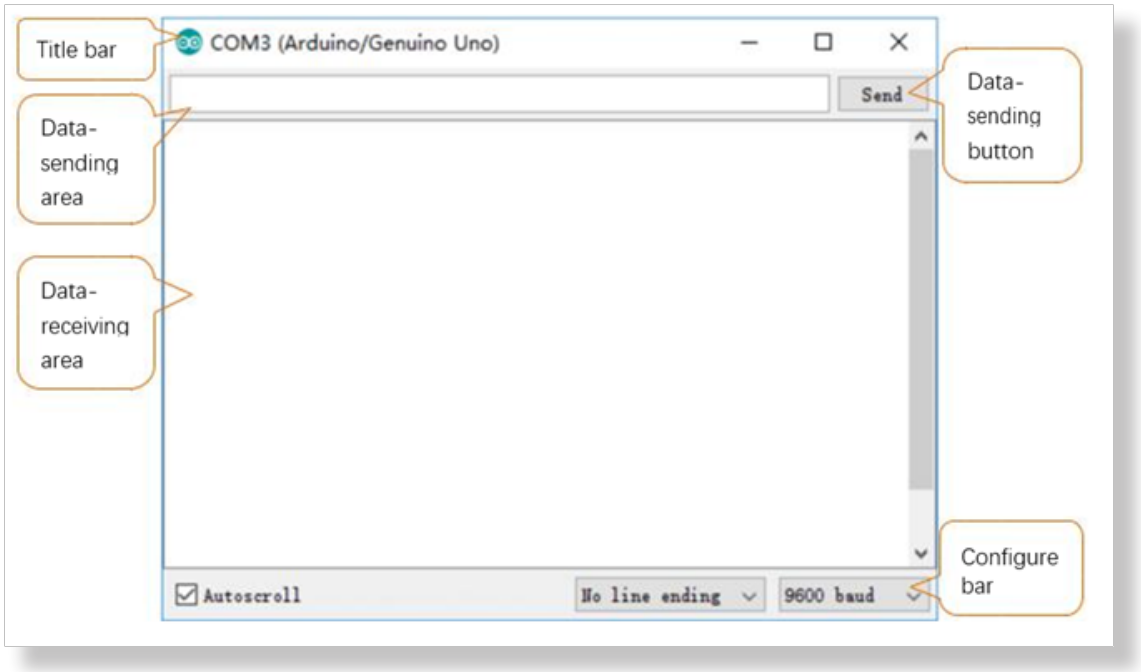
Parts Required:

- Uno R3 board, USB Cable

Serial port on Arduino

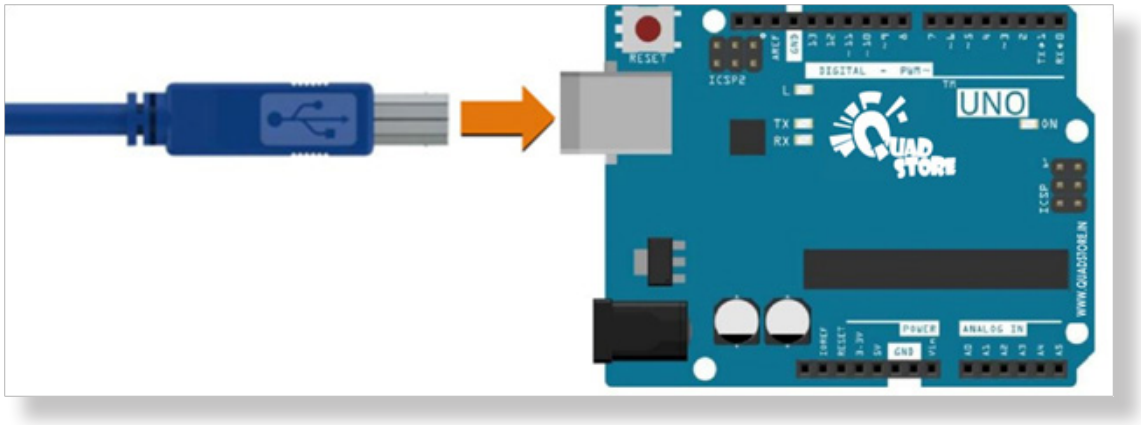
UNO has integrated USB to serial transfer, could communicates with computer when USB cable get connected to it. Arduino Software also uploads code to UNO through the serial connection. Computer identifies serial devices connected to your computer as COMx. We can use the Serial Monitor window of Arduino 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.

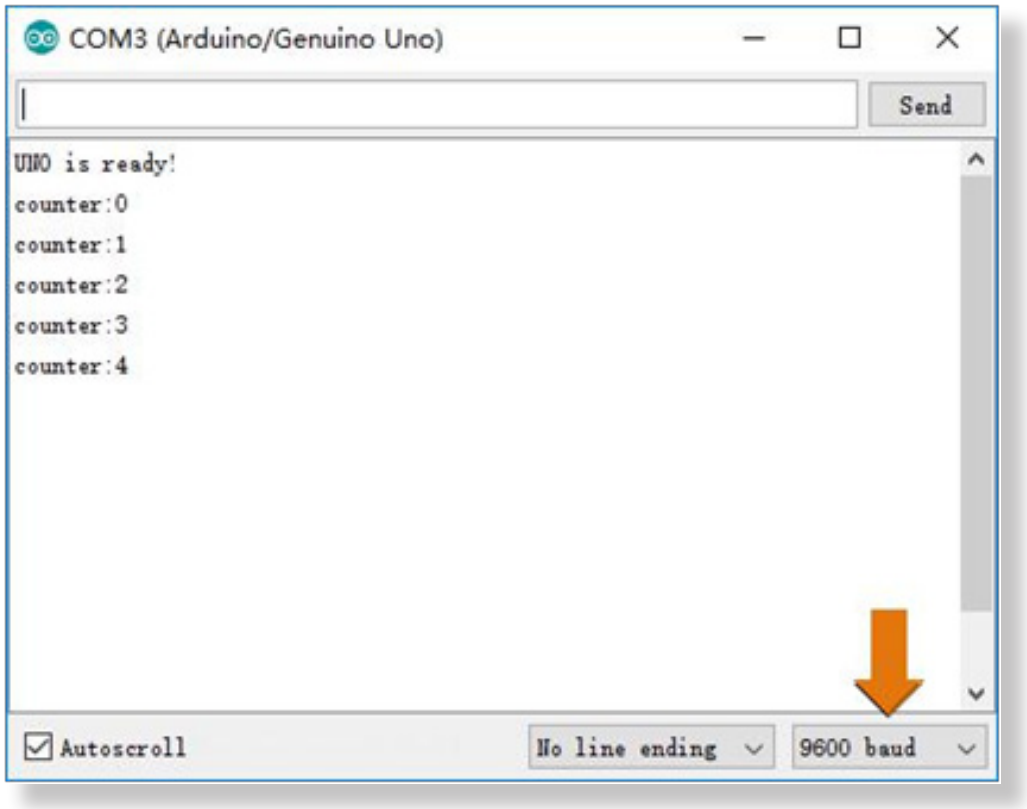


Code:

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

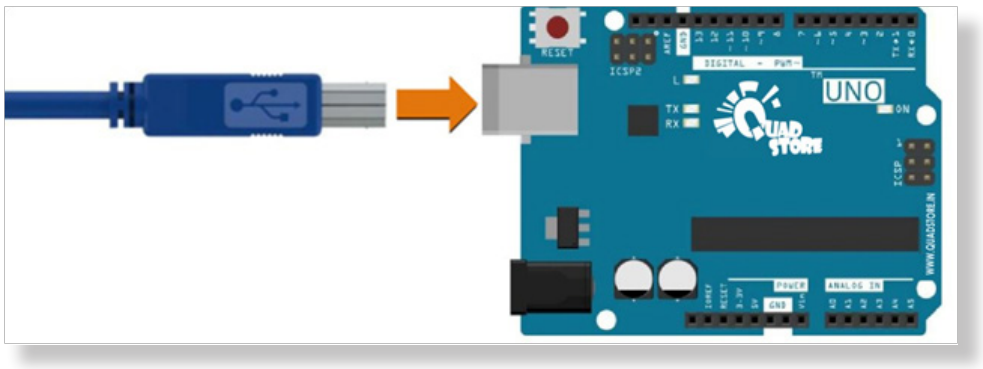
We will use the serial port to receive data from Uno R3.

Parts Required:

- Uno R3 board, USB Cable

Circuit Connection:

Connect UNO to the computer with USB cable

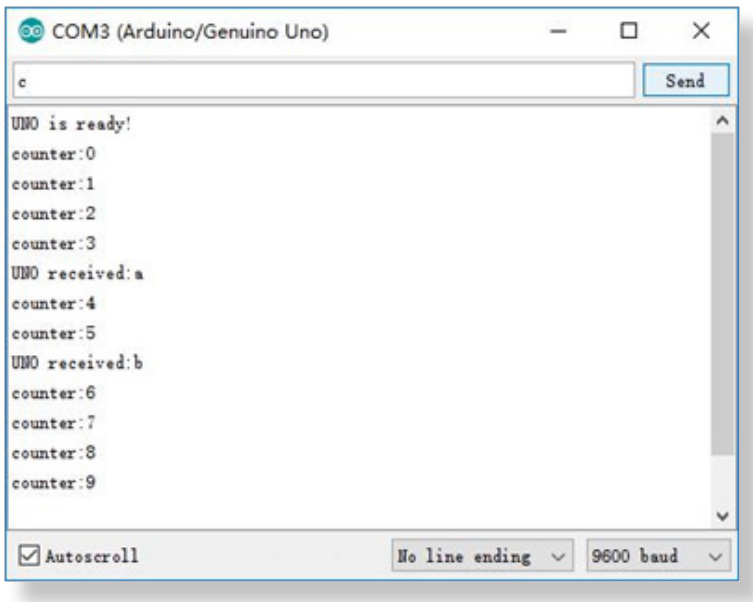


Code:

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

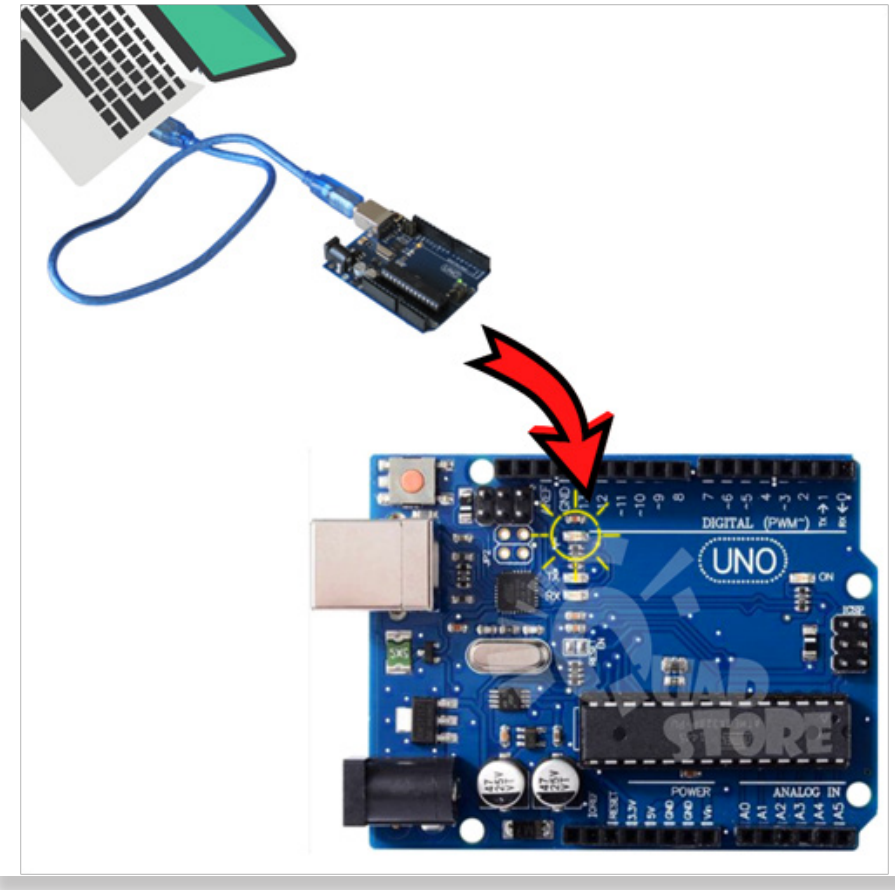
Make an inbuilt led below 13th pin blink.

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

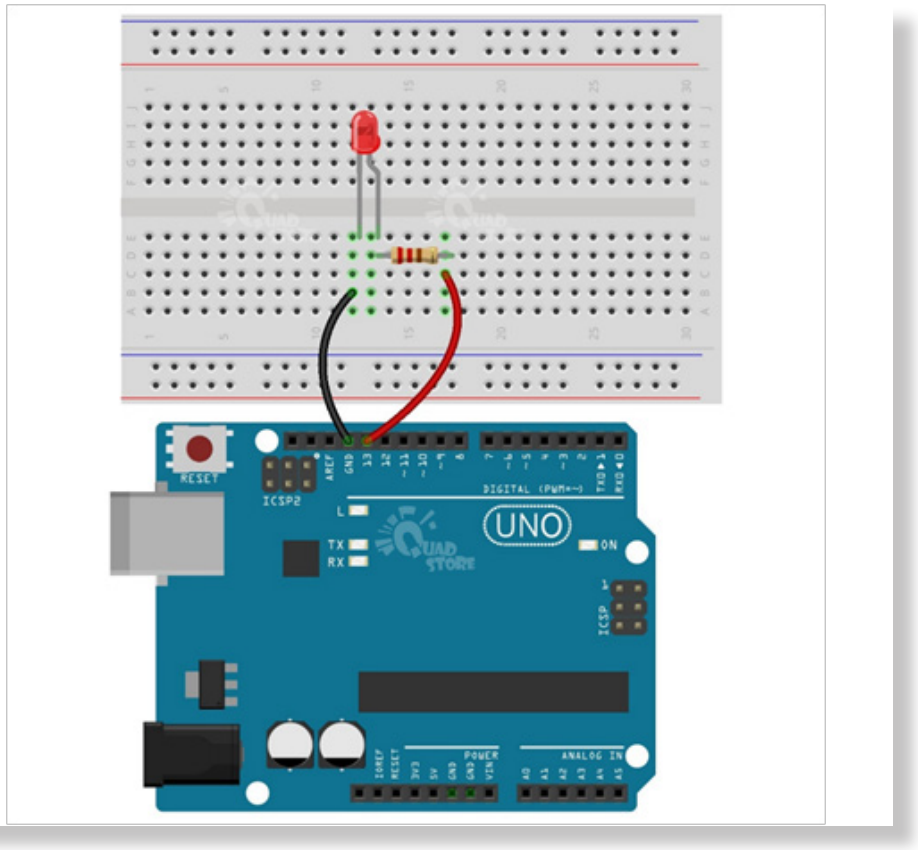
Make an external LED blink.

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

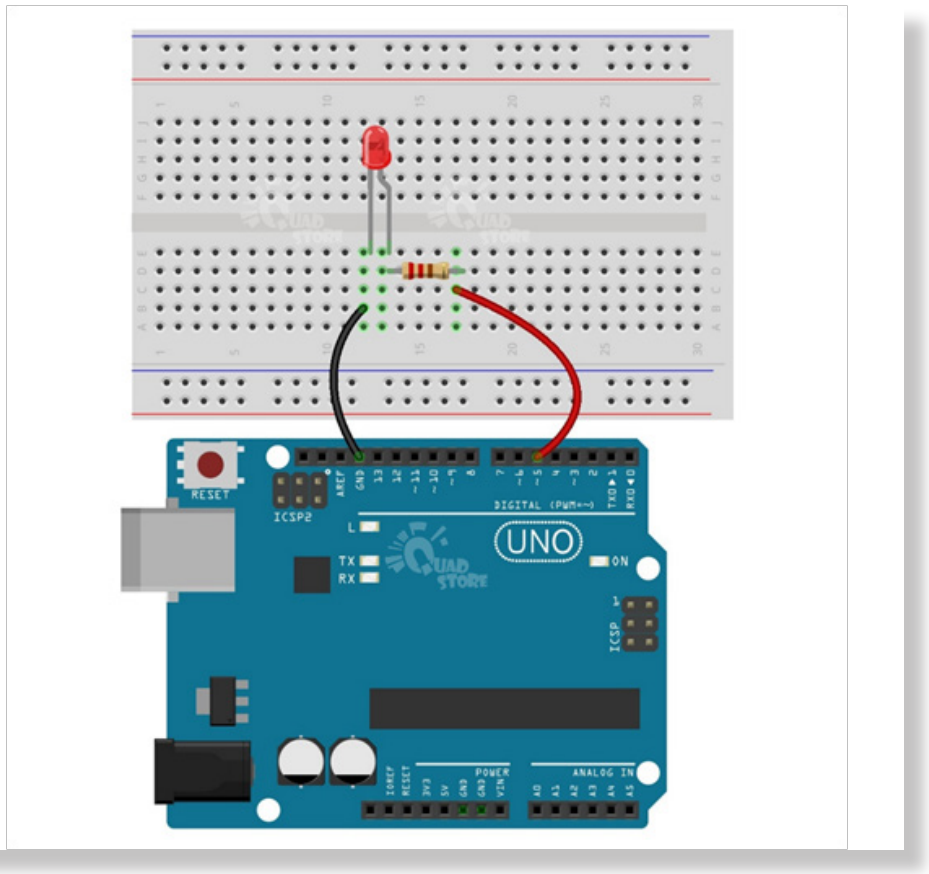
Make an LED blink smoothly.

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:
Push Button

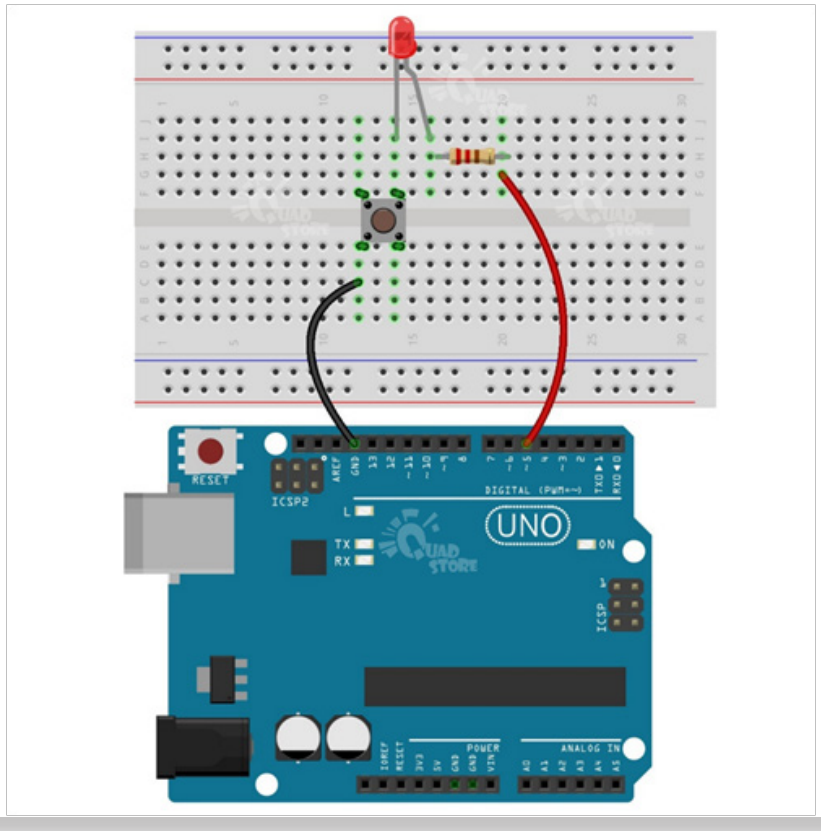
Turn the Led ON by pressing push button.

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

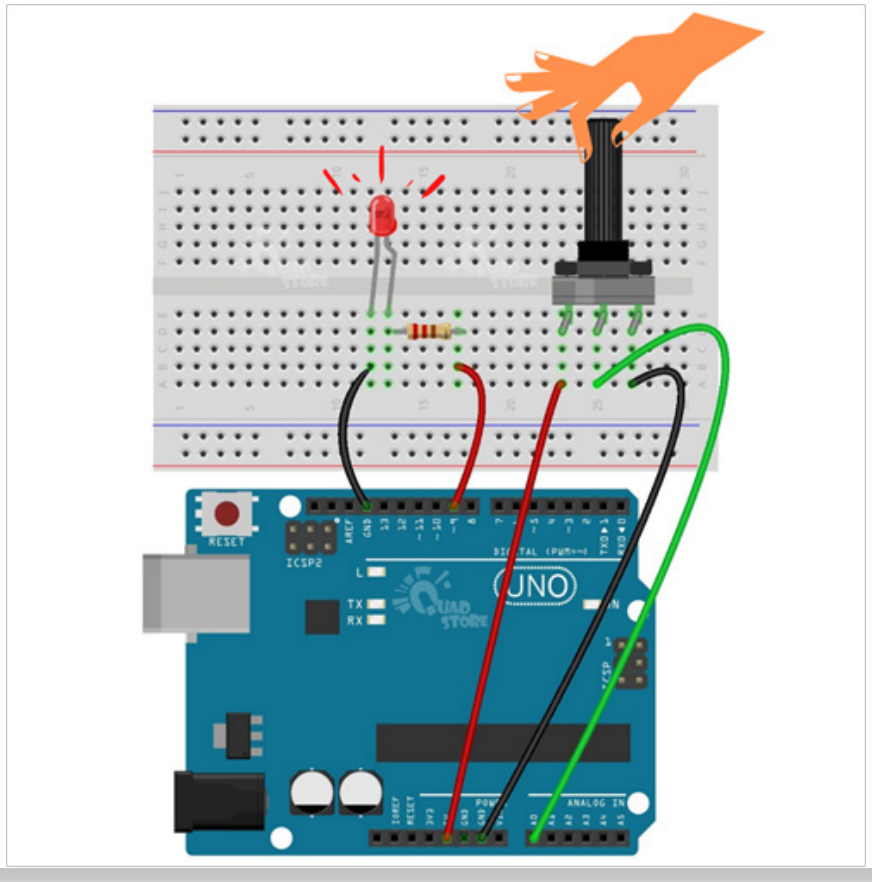
Adjust the brightness of Led using potentiometer.

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

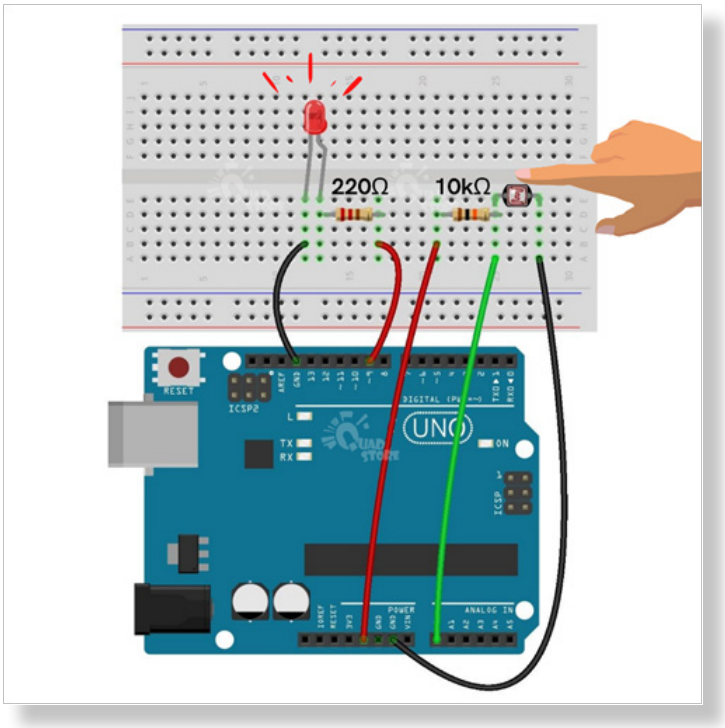
Adjust the brightness of Led using photoresistor.

Parts Required:

- Uno R3 board, USB Cable
- Breadboard
- Few jumper wires
- LED
- Resistor – 220ohm for LED
- Resistor – 10Kohm for photoreistor
- 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 photoresitor with your finger to see the intensity of the LED increase.

Project 11: RGB Led

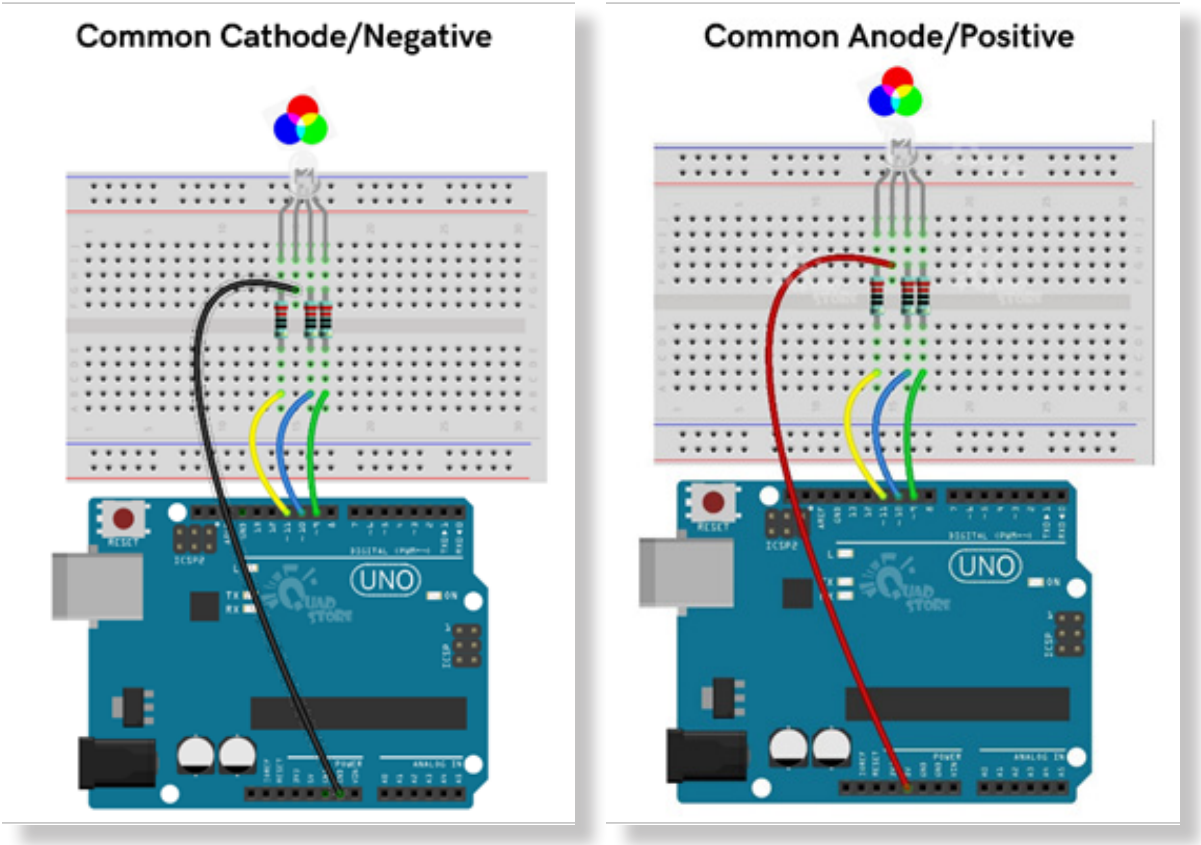
Turn on the RGB led to see different colors pop up

Parts Required:

- Uno R3 board, USB Cable
- Breadboard
- Few jumper wires
- RGB LED (Please check if you have received common Anode or common Cathode)
- Resistor – 220ohm x 3

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 etc

Project 12: Active Buzzer

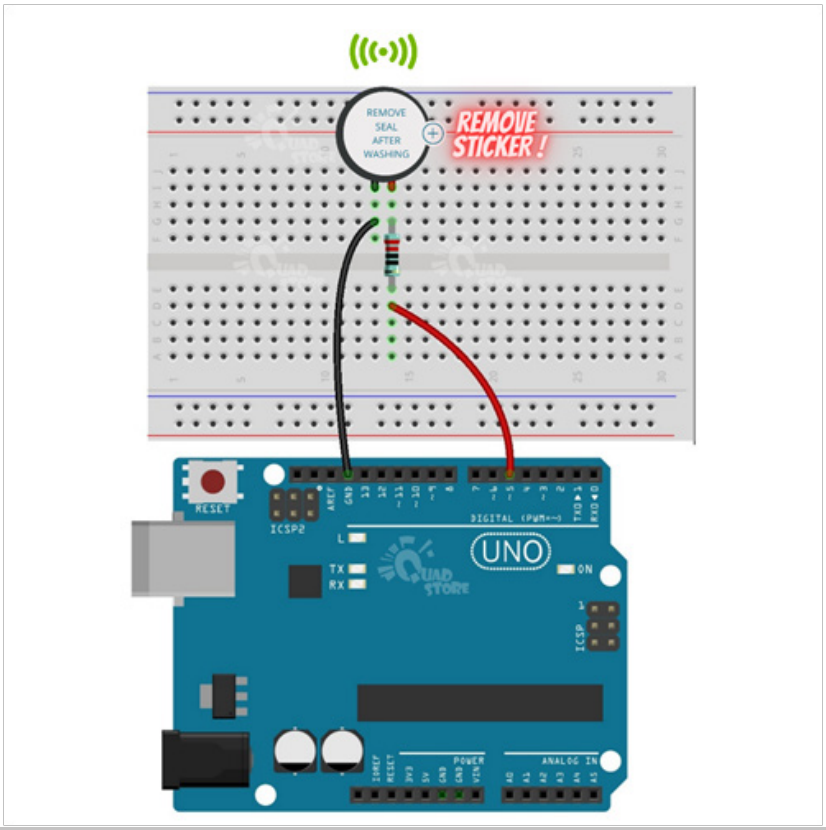
Make the buzzer beep in certain interval.

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

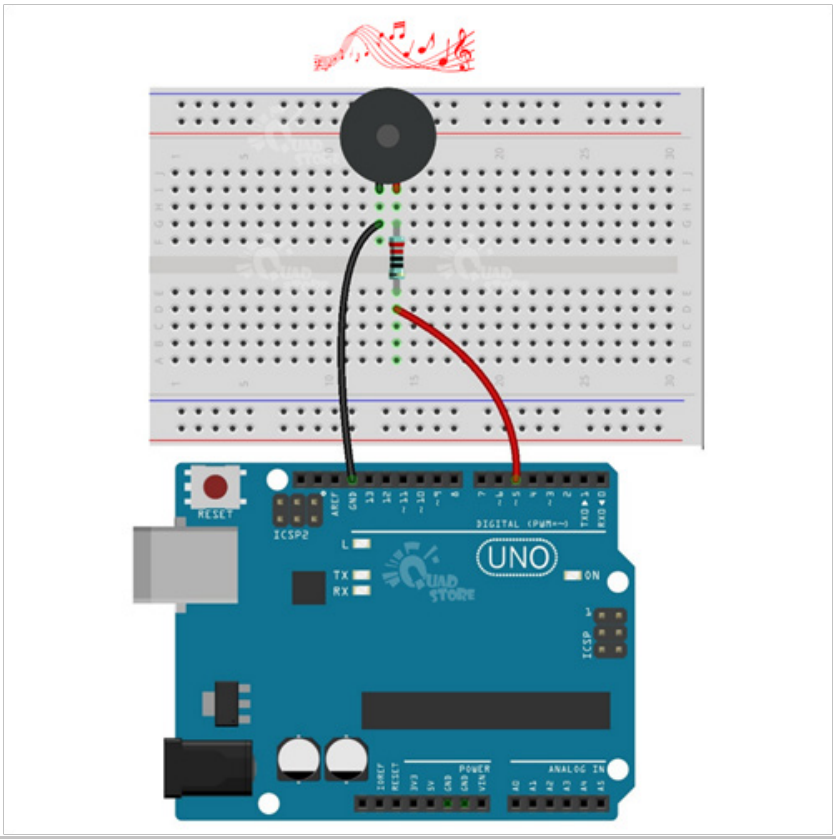
Hear the tone/music through passive buzzer.

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: Servo Motor

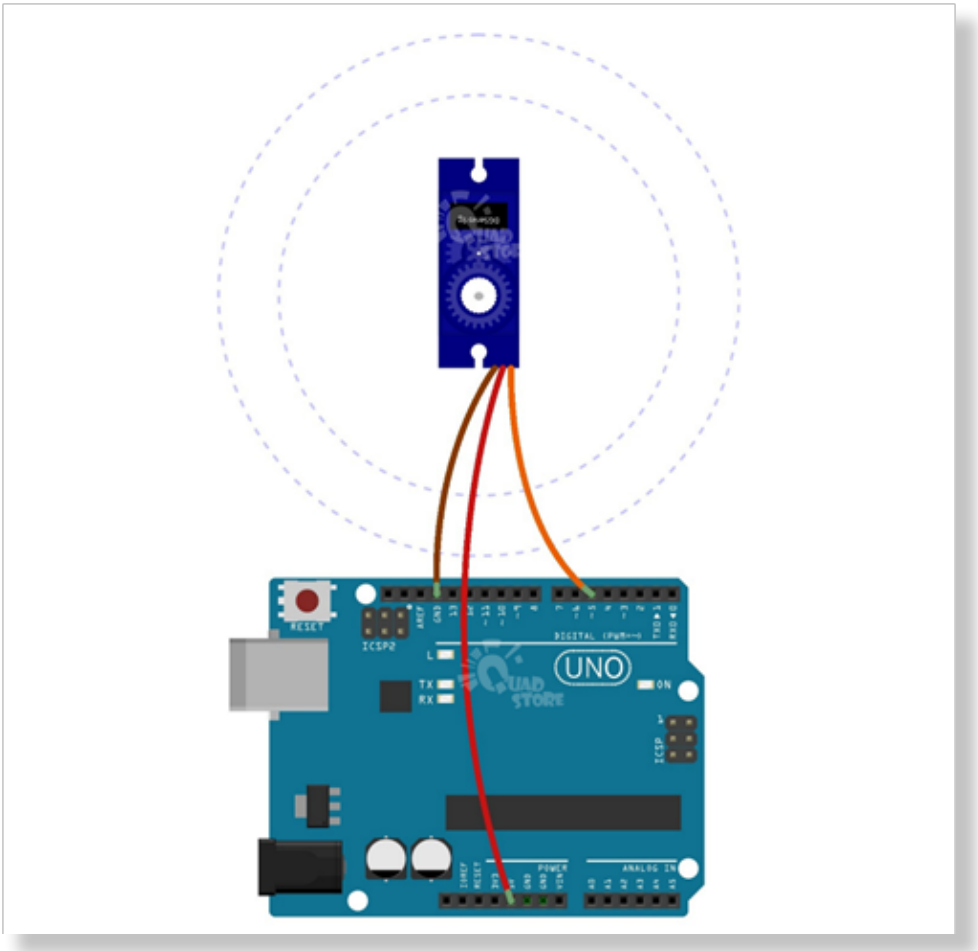
Make the servo rotate from from 0 to 180 degrees and vice versa.

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

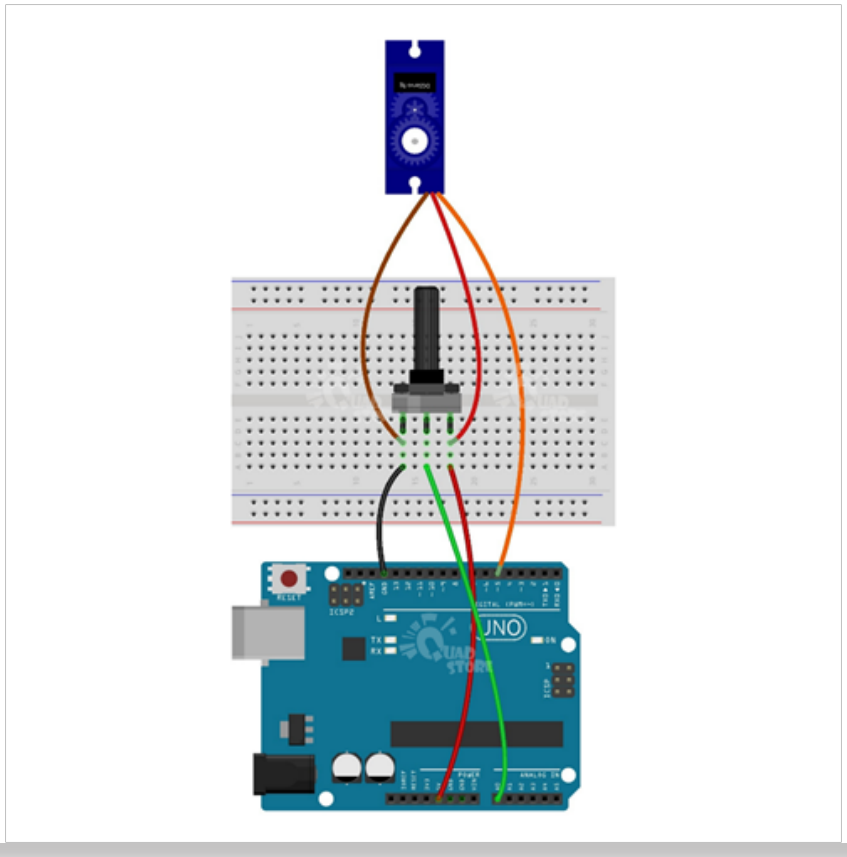
Control the rotation of servo using potentiometer

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.

Project 16: Stepper Motor

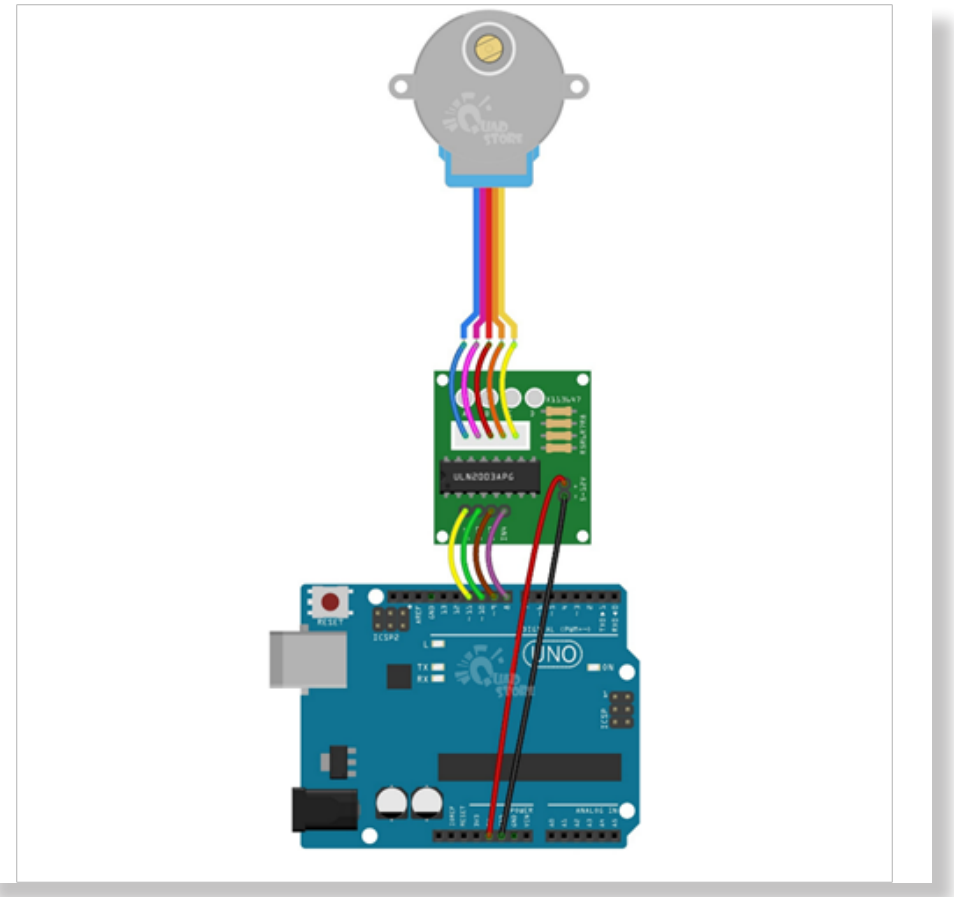
Make the stepper motor rotate clockwise and anticlockwise

Parts Required:

- Uno R3 board, USB Cable
- Few jumper wires
- Stepper Motor (28BYJ-48)
- Stepper Motor Driver (ULN2003)

Circuit Connection:

Provide the connection as per the below circuit.



Code:

Verify and Upload the code named **"Stepper_Motor.ino"** to Uno R3 board

Output:

You will see the stepper motor rotate one complete rotation clock wise and anticlockwise repeatedly.

Project 17:
DHT Temperature Sensor using 3 pins

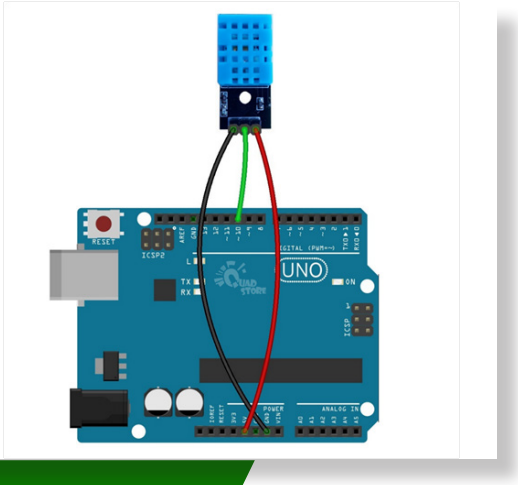
Display current temperature and humidity in serial monitor.

Parts Required:

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

Circuit Connection:

Provide the connection as per the below circuit.



Library:

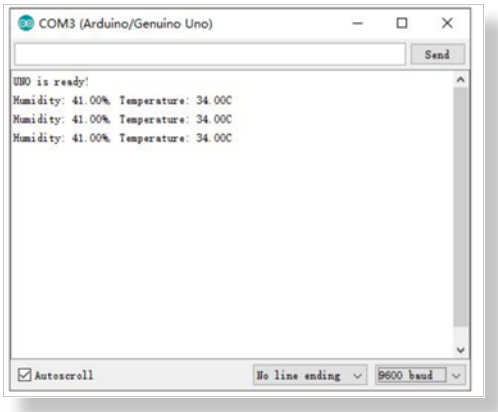
Install DHT.zip library before uploading the code. Otherwise your code will give error message.

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 18:
DHT Temperature Sensor using 4 pins

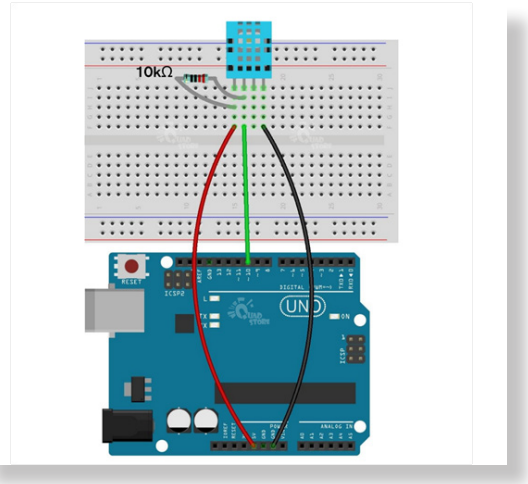
Display current temperature and humidity in serial monitor.

Parts Required:

- Uno R3 board, USB Cable
- Breadboard
- Few jumper wires
- Resistor - 10Kohm
- 4pin DHT11 Sensor

Circuit Connection:

Provide the connection as per the below circuit.



Library:

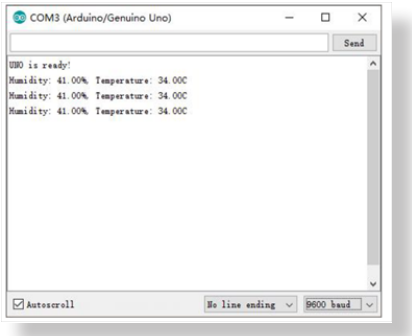
Install DHT.zip library before uploading the code. Otherwise your code will give error message.

Code:

Verify and Upload the code named **"DHT11_TempSensor_4pin.ino"** to Uno R3 board.

Output:

Open **"Serial Monitor"** to see the current temperature and humidity displayed.



Project 19: Joystick

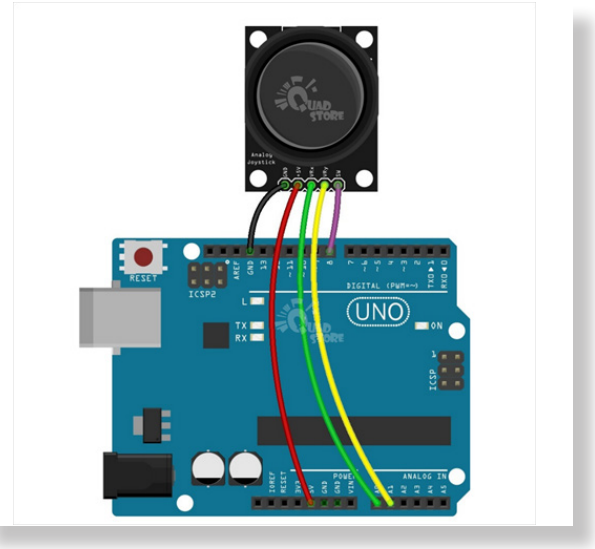
Display current temperature and humidity in serial monitor.

Parts Required:

- Uno R3 board, USB Cable
- Few jumper wires
- Joystick Module

Circuit Connection:

Provide the connection as per the below circuit.

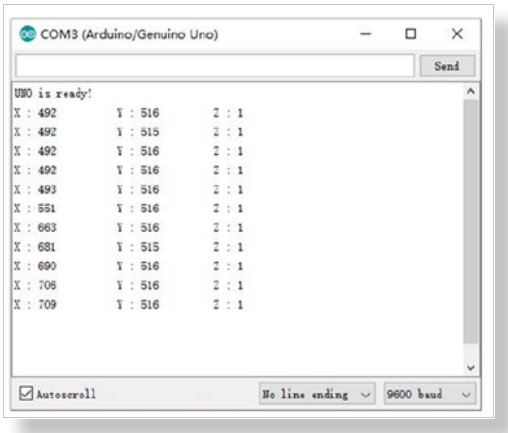


Code:

Verify and Upload the code named **"Joystick.ino"** to Uno R3 board

Output:

Open "**Serial Monitor"** then you can see the current Joystick state value sent by UNO board. More or press the joystick with your finger, then you can see the change of value in serial monitor.



Project 20:
1-digit 7 Segment Display

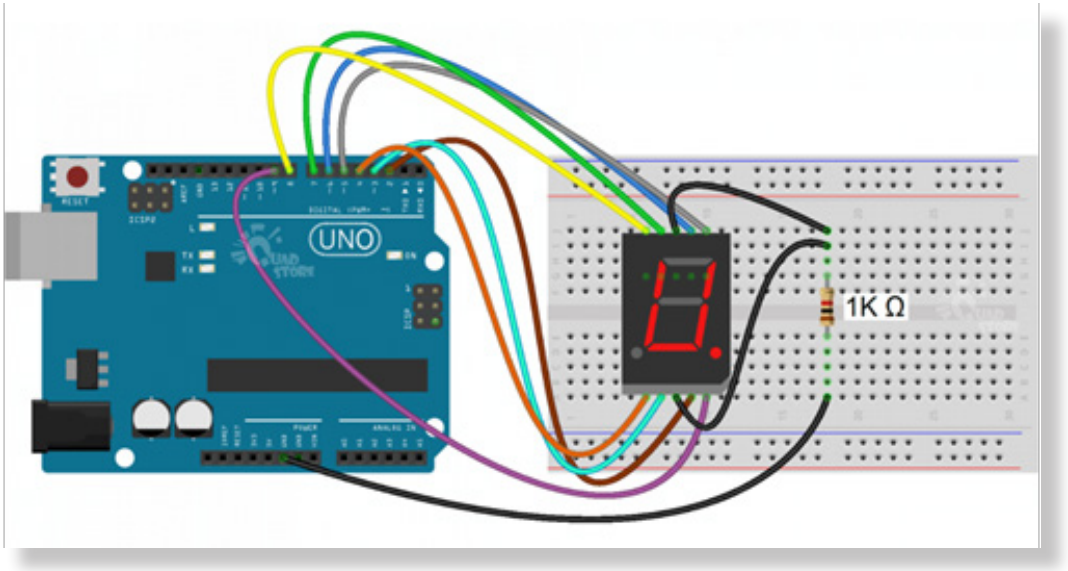
Display numbers from 0 to 9 in the 1-digit 7 Segment display.

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.inio"** to Uno R3 board

Output:

You can see the numbers displayed from 0 to 9.

Project 21: 4-digit 7 Segment Display

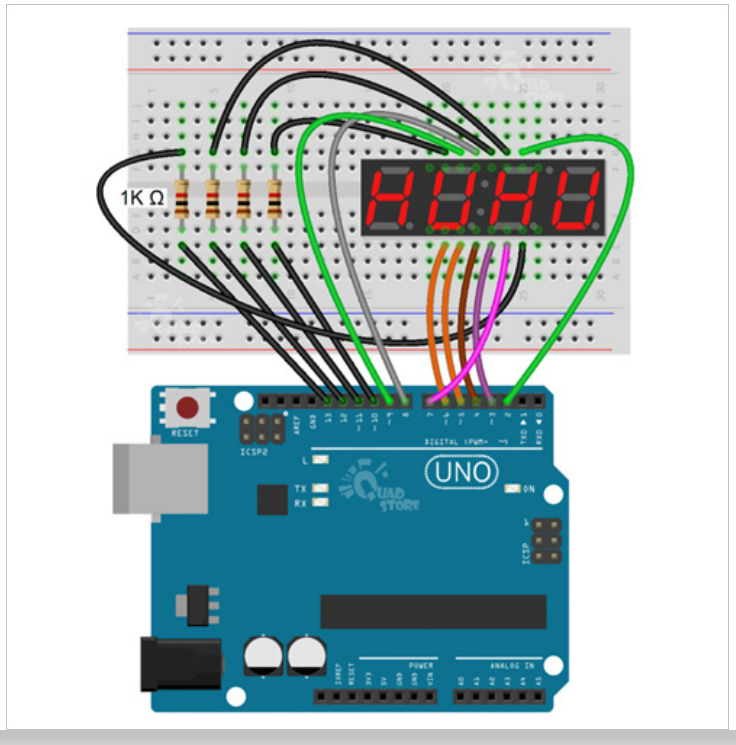
Display the numbers 4.999 in the 4-digit 7 Segment display.

Parts Required:

- Uno R3 board, USB Cable
- Breadboard
- Few jumper wires
- Resistor – 1Kohm x4
- 4-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 **"4Digit_7Segment.ino"** to Uno R3 board

Output:

You will see the number 4.999 displayed.

Project 22:
8*8 Led Matrix Display

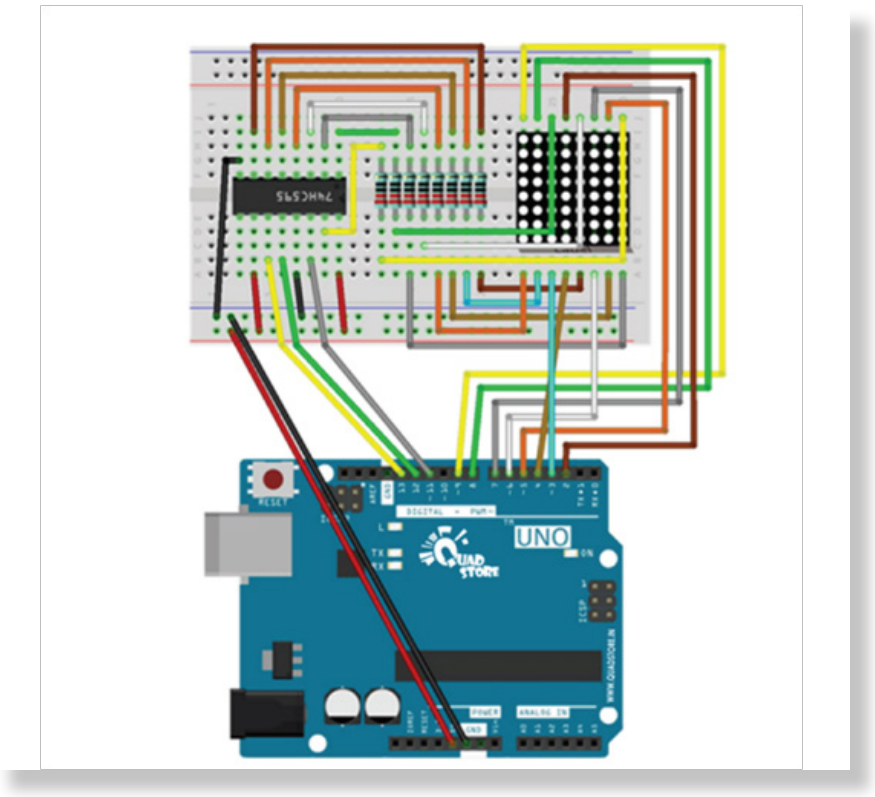
Display smiley, digits and character in 8*8 led matrix display.

Parts Required:

- Uno R3 board, USB Cable
- Breadboard
- Few jumper wires
- Resistors – 220ohm x8
- 74HC595 Chip
- 8*8 Led Matrix

Circuit Connection:

Provide the connection as per the below circuit.



Code:

Verify and Upload the code named **"LED_Matrix.ino"** to Uno R3 board

Output:

LED Matrix begins to display the static smiley pattern. A few secondslater, LED Matrix will display the scrolling number 0-9 and the letter A-F.

Project 23:
I2c 1602 Lcd Display

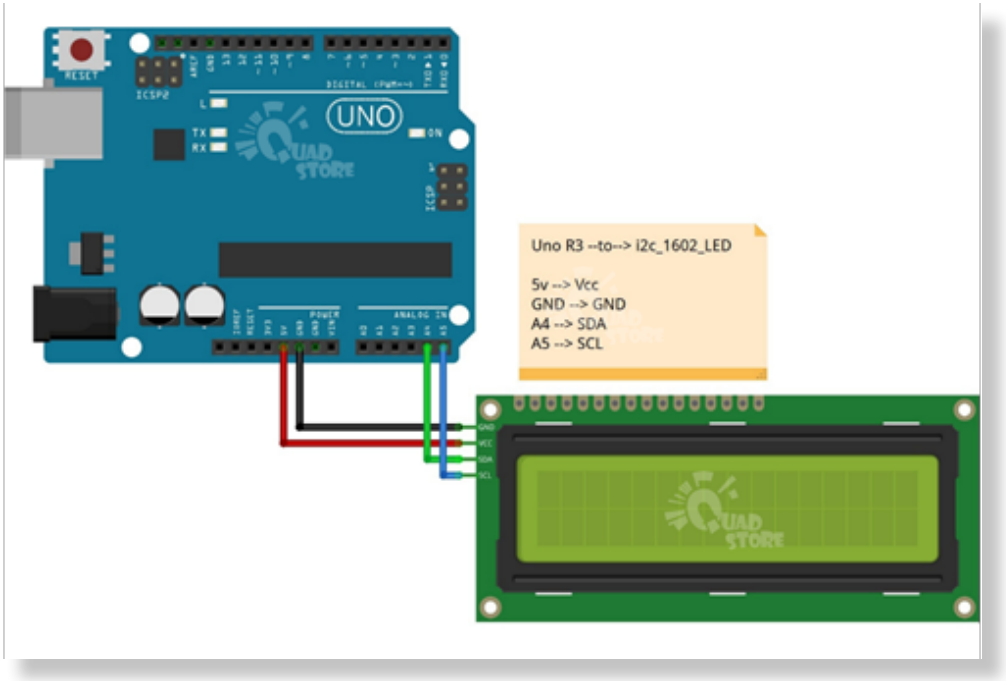
Display counter number in 1602 display.

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_1.3.5library 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 24:
1602 display with manual soldered pins

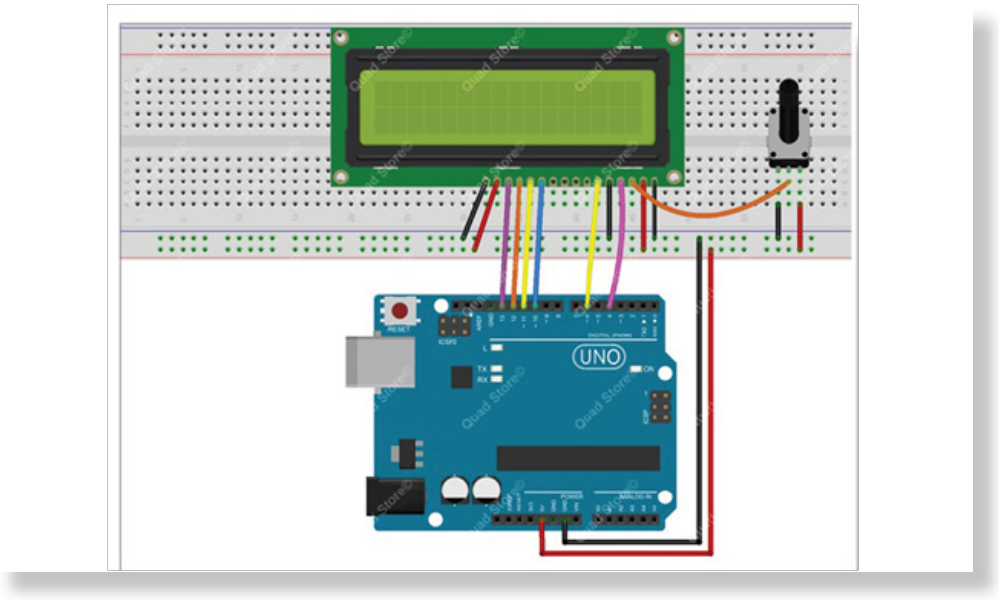
Display text in your 1602 displays screen.

Parts Required:

- Uno R3 board, USB Cable
- Breadboard
- Few jumper wires
- 1602 lcd display
- 16 pin male header for soldering with 1602 display (**Note:** you need to solder the male header pins with the 1602 display to work properly. If you try to plug in the wires directly to 1602 display it will not work.)
- Potentiometer

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 **"1602_Soldered.ino"** to Uno R3 board

Output:

Now adjust the potentiometer level for the brightness setting to display the text "Hello Geeks!" followed by "Quad Store" and then "www.quadstore.in" displayed.

Project 25: Bar LED

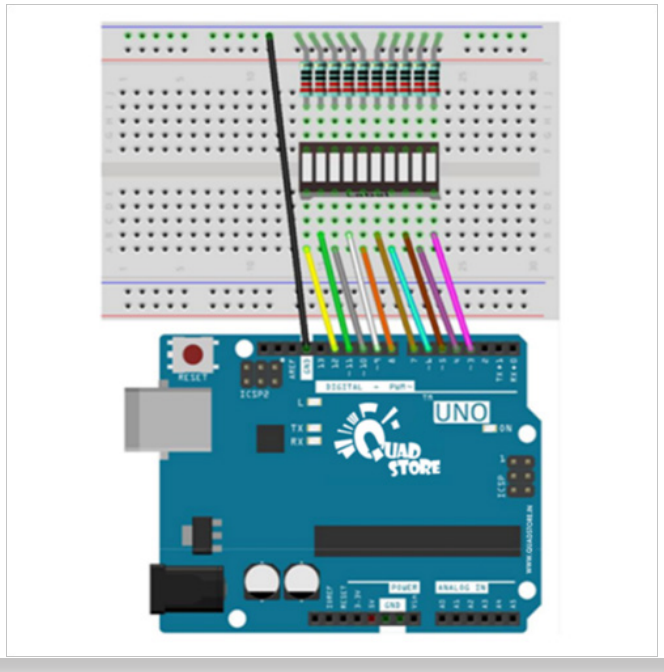
Make the Led light move from left to right and vice versa.

Parts Required:

- Uno R3 board, USB Cable
- Breadboard
- Few jumper wires
- Resistors – 220ohm x10
- Bar LED

Circuit Connection:

Provide the connection as per the below circuit.



Code:

Verify and Upload the code named **"Bar_LED.ino"** to Uno R3 board

Output:

You will see the LED light moving from right to left and vice versa on LED bar graph

Project 26: Keypad

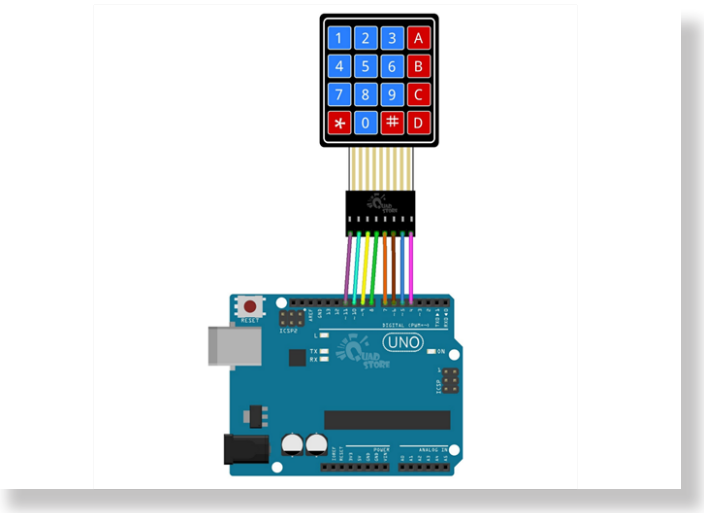
Display the keypad character presses in serial monitor.

Parts Required:

- Uno R3 board, USB Cable
- Few jumper wires
- 4x4 Keypad

Circuit Connection:

Provide the connection as per the below circuit.



Library:

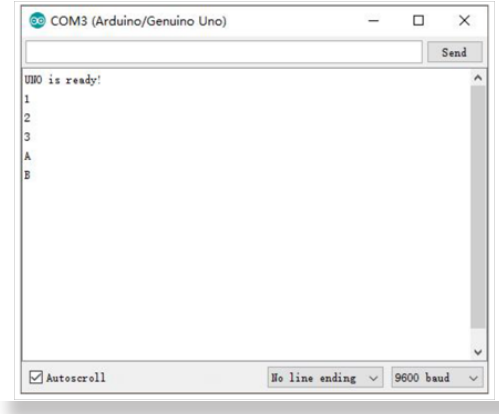
Install Keypad.zip library before uploading the code. Otherwise your code will give error message.

Code:

Verify and Upload the code named **"Keypad.ino"** to Uno R3 board

Output:

Open the "**Serial Monitor"** and press the keypad, then you will see thecharacter you press is printed out.



Project 27: Keypad

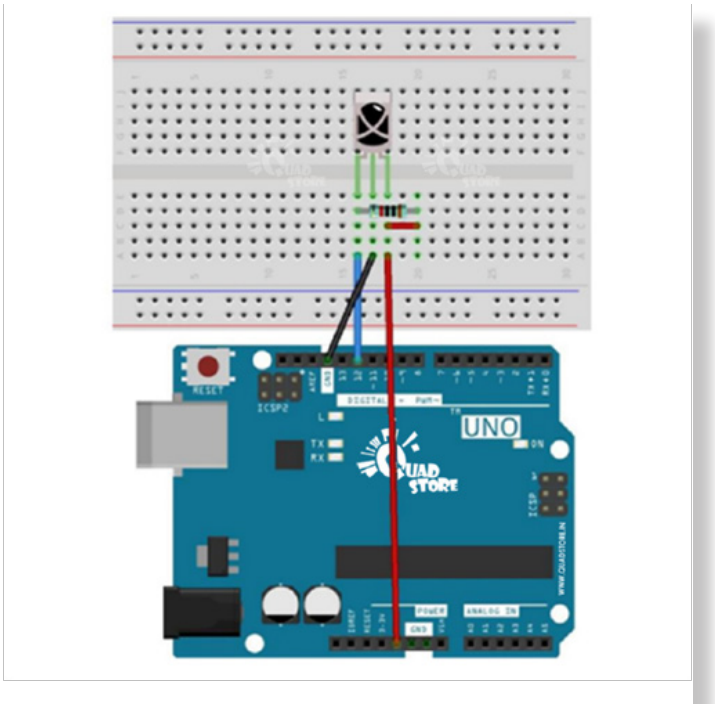
Display the HEX character valuesfrom IR remote to serial monitor

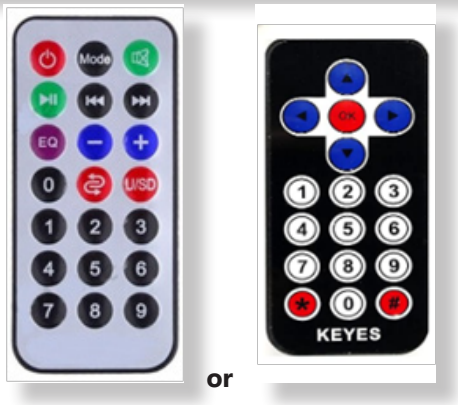
Parts Required:

- Uno R3 board, USB Cable
- Breadboard
- Few jumper wires
- IR Remote
- IR Receiver
- Resistor – 10Kohm

Circuit Connection:

Provide the connection as per the below circuit.





or

LIBRARY:

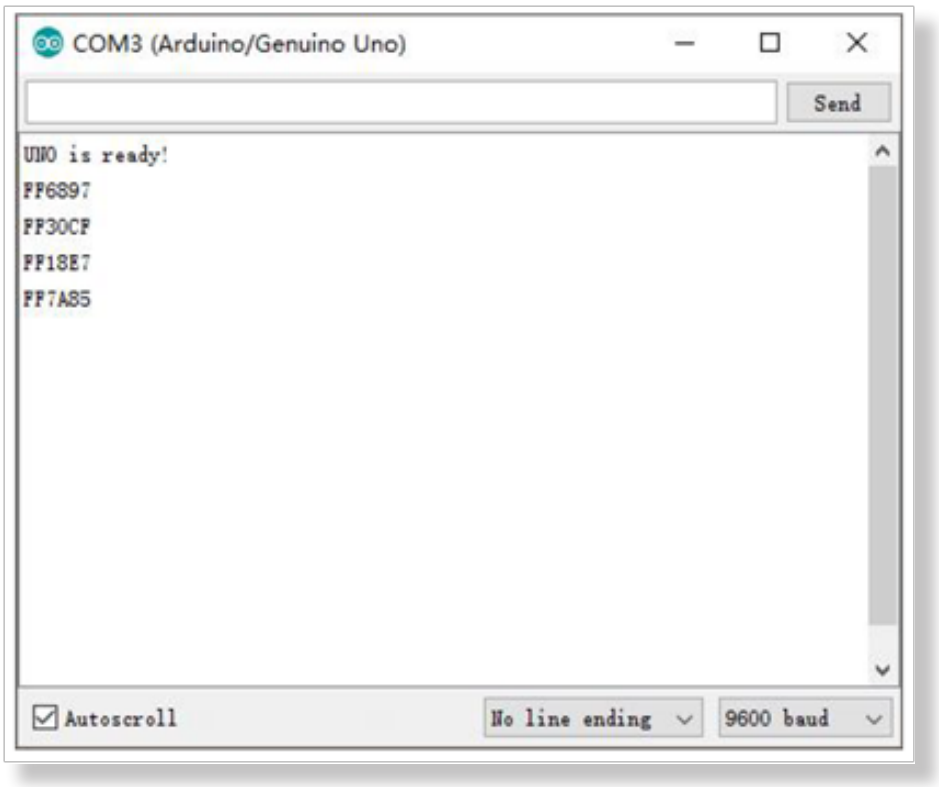
Install IRremote.zip library before uploading the code. Otherwise your code will give error message.

CODE:

Verify and Upload the code named **"Infrared_Remote_Control.ino"** to Uno R3 board

OUTPUT:

Open the **"Serial Monitor",** and press any button from IR remote, then you can see the corresponding code will be printed out for that button.



Project 28:
Vibration Switch

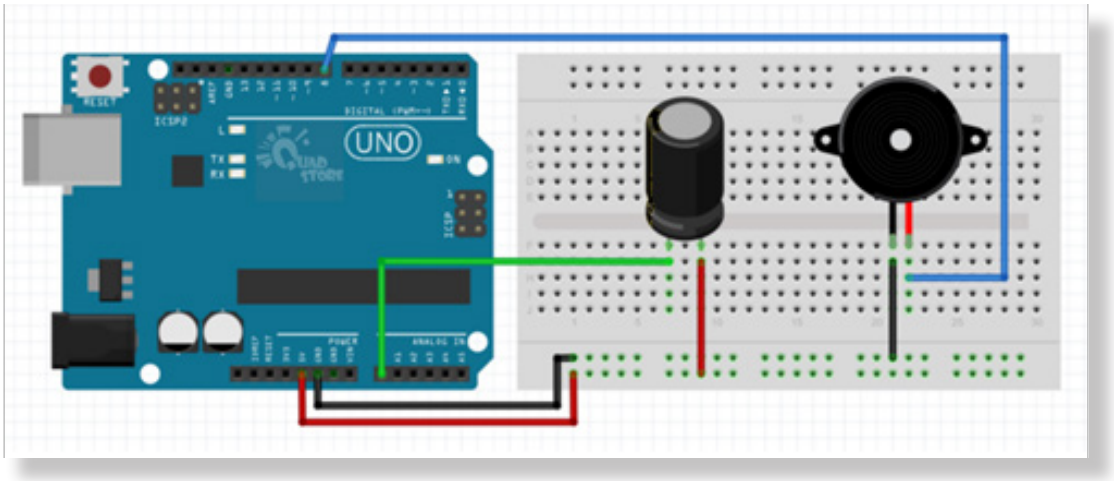
Make the buzzer beep on vibration

Parts Required:

- Uno R3 board, USB Cable
- Breadboard
- Few jumper wires
- Vibration Switch
- Active Buzzer

Circuit Connection:

Provide the connection as per the below circuit.



Code:

Verify and Upload the code named **"Vibration_Switch.ino"** to Uno R3 board

Output:

Shake the vibration switch and you should hear the buzzer beep.

Project 29: Infrared(IR) Sensor Module

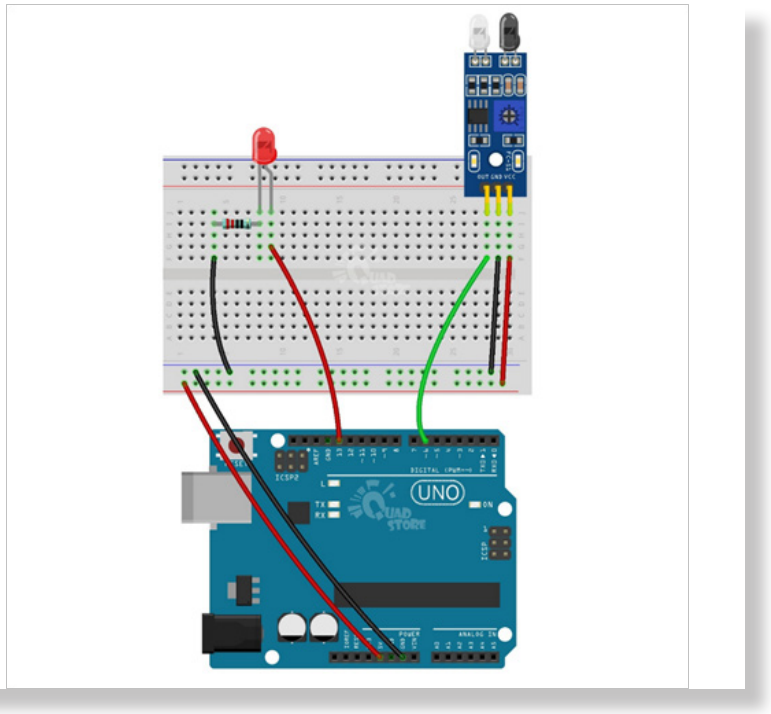
Wave your hand or object in front of IR sensor and make LED turn ON.

Parts Required:

- Uno R3 board, USB Cable
- Breadboard
- Few jumper wires
- IR Sensor Module
- LED
- Resistor – 220ohm

Circuit Connection:

Provide the connection as per the below circuit.



Code:

Verify and Upload the code named **"IR_Sensor.ino"** to Uno R3 board

Output:

Place the hand or any other object in front of IR sensor and you should see the LED turn ON. When you remove the object then LED turns OFF.

Note:

You can adjust the sensitivity of the IR sensor using the inbuilt potentiometer. Give a try !

Project 30: Ultrasonic Sensor HC-SR04

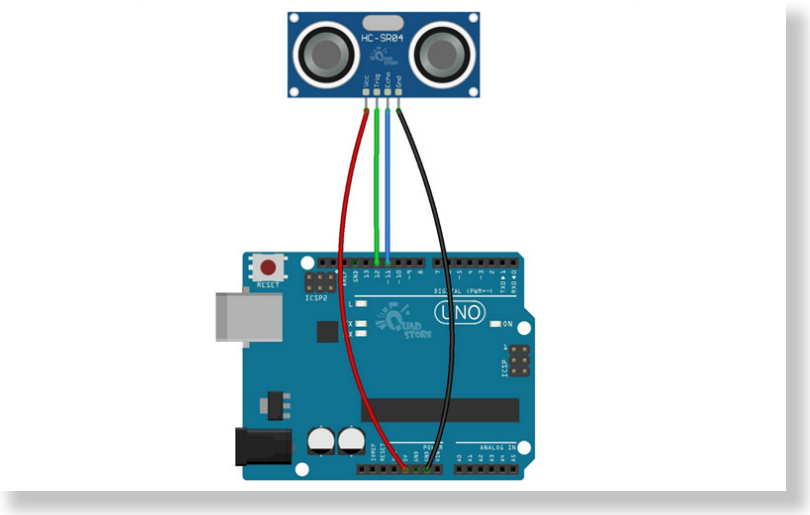
Display the distance of object in serial monitor.

Parts Required:

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

Circuit Connection:

Provide the connection as per the below circuit.

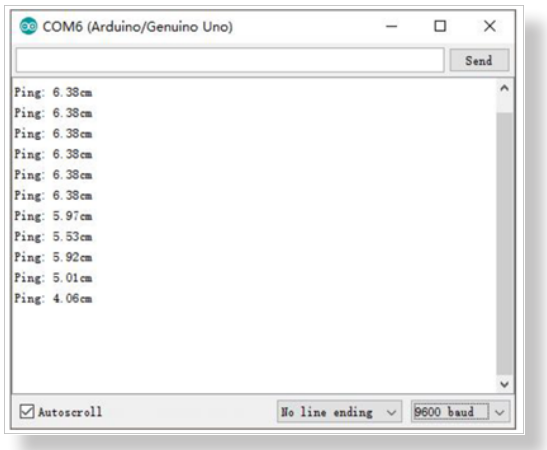


Code:

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 31: Relay

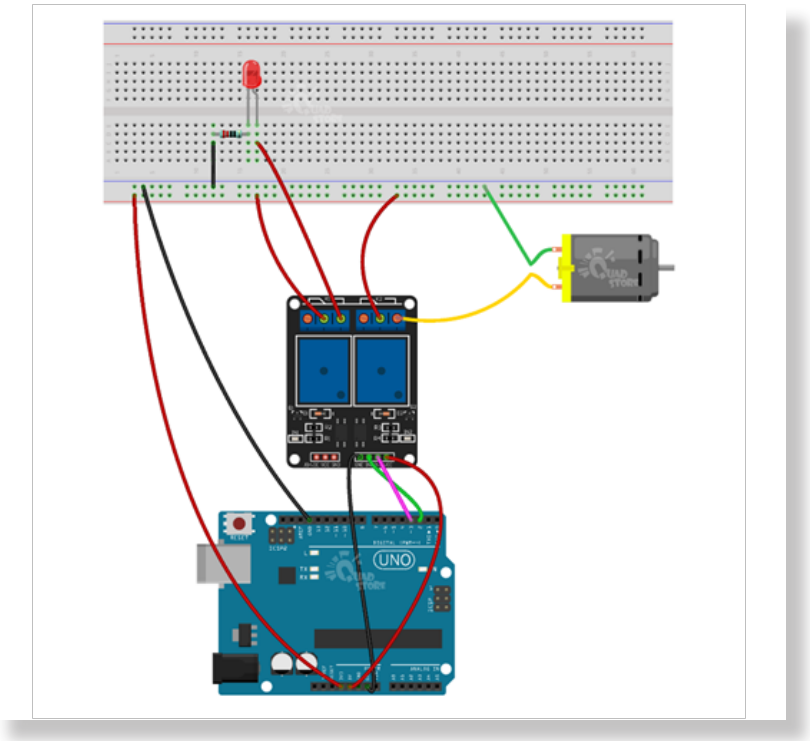
Durn on the LED and DC motor using relay module.

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 **"Relay2Channel.ino"** to Uno R3 board

Output:

The relay will switch ON and OFF between the 2 channels and you will see when Channel-1 turns on LED will glow and when Channel-2 turns on the DC motor will run.

Project 32:
Real Time Clock (RTC) – DS3231

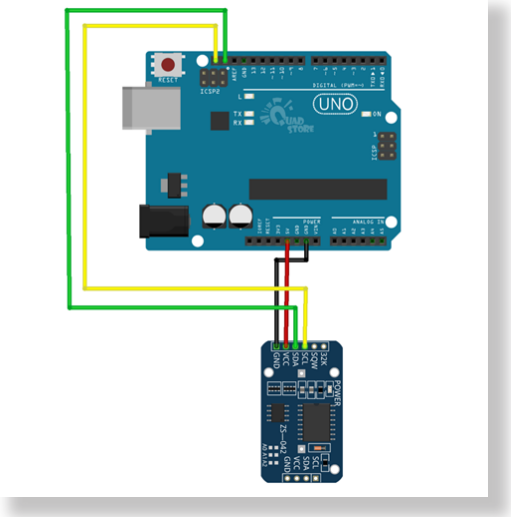
Display current date and time in serial monitor.

Parts Required:

- Uno R3 board, USB Cable
- Few jumper wires
- Real time clock module – DS3231(Install the battery for RTC module)

Circuit Connection:

Provide the connection as per the below circuit.



Library:

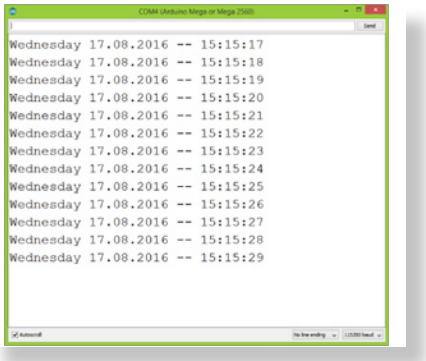
Install ds3231-master.zip library before uploading the code. Otherwise your code will give error message.

Code:

First change the date to current date and time in the code. Then verify and Upload the code named **"RTC_ DS3231.ino"** to Uno R3 board

Output:

Open "Serial Monitor" and you will see the current date and time displayed.



Project 33:
PIR Motion Sensor

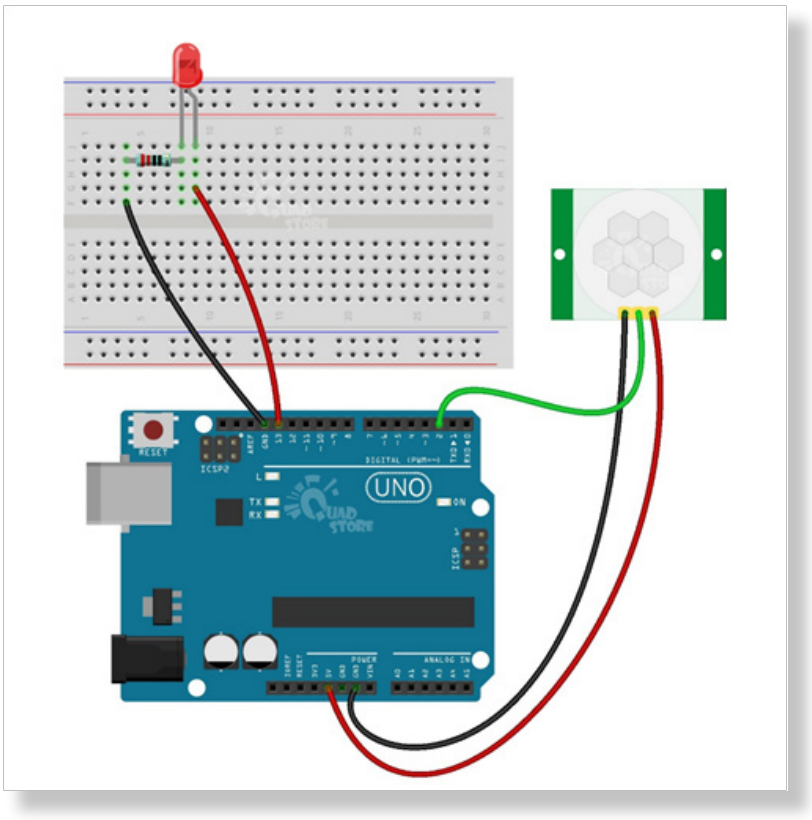
Deduct the motion and turn ON the led light.

Parts Required:

- Uno R3 board, USB Cable
- Breadboard
- Few jumper wires
- PIR Motion Sensor
- LED
- Resistor – 220ohm

Circuit Connection:

Provide the connection as per the below circuit.



Code:

Verify and Upload the code named **"PIR_Motion_Sensor.ino"** to Uno R3 board

Output:

Wave your hand in front of the PIR sensor or move in front of the sensor to trigger the LED turn ON.

PROJECT 34: RFID

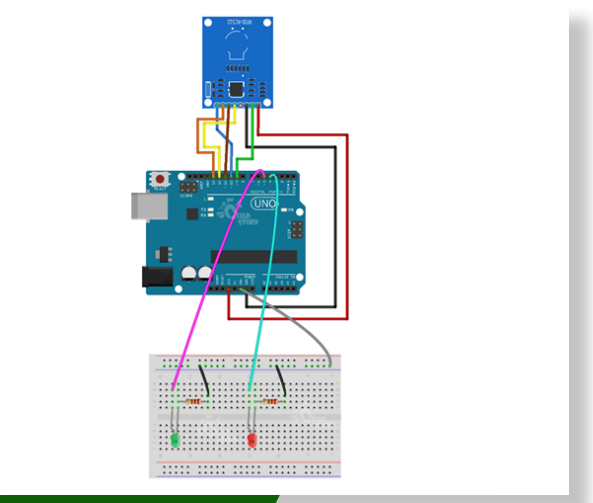
Demo to try out valid and invalid entry using RFID cards and tags.

PARTS REQUIRED:

- Uno R3 board, USB Cable
- Breadboard
- Few jumper wires
- RFID Scanner
- RFID Tag and Card

CIRCUIT CONNECTION:

Provide the connection as per the below circuit.



LIBRARY:

Install rfid-master.zip library before uploading the code. Otherwise your code will give error message.

UPLOAD-1:CODE:

Verify and Upload the code named **"RFID.ino"** to Uno R3 board

ACTION:

Open "Serial Monitor". Place one of the RFID card under the scanner and it should display the corresponding HEX value of that card. Copy the HEX value of the card and paste in the Code line which says [if (content. substring(1) == "XX XX XX XX")] replacing value XX XX XX XX. Now you have given access to the card. Save the code.

UPLOAD-2:CODE:

Verify and Upload the updated code named "**RFID.ino"** to Uno R3 boardonce again after the value of the card is updated.

OUTPUT:

Place the valid card under the scanner and you should see the GREEN Led turn ON. When you place a card

Project 35: RFID

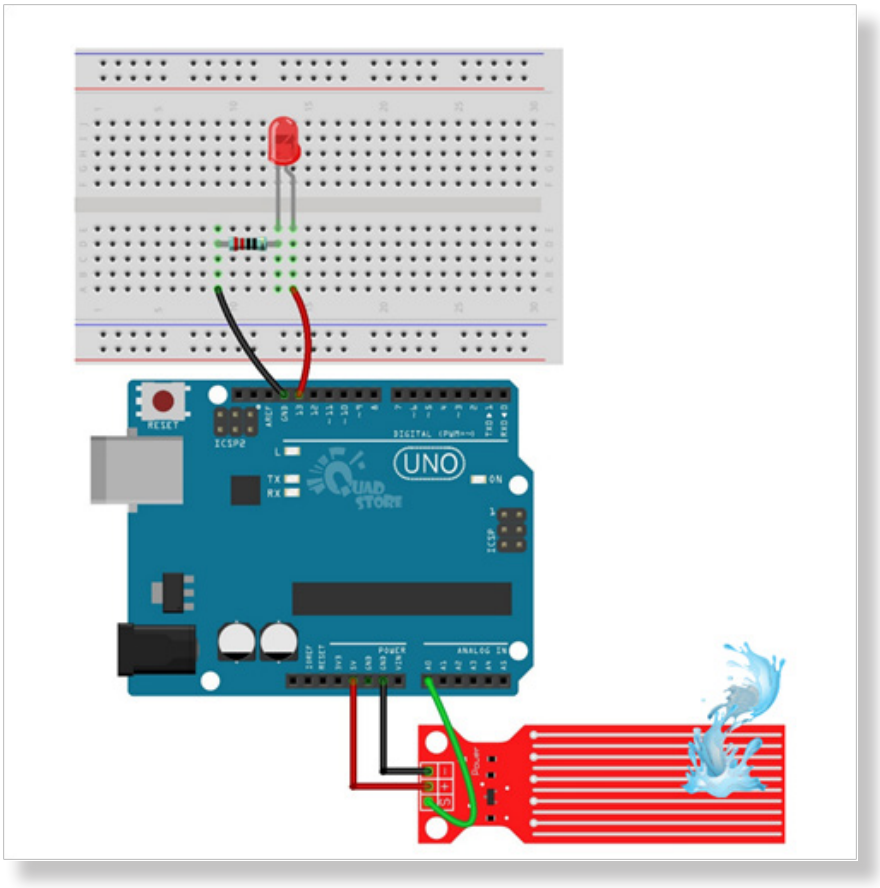
Detect water to make LED turn ON.

Parts Required:

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

Circuit Connection:

Provide the connection as per the below circuit.



Code:

Verify and Upload the code named "**Water_Sensor.ino**" to Uno R3 board

Output:

When you put the sensor into water you will see the LED turn ON and when you remove sensor from water LED turns OFF.

Project 36: Rain Sensor

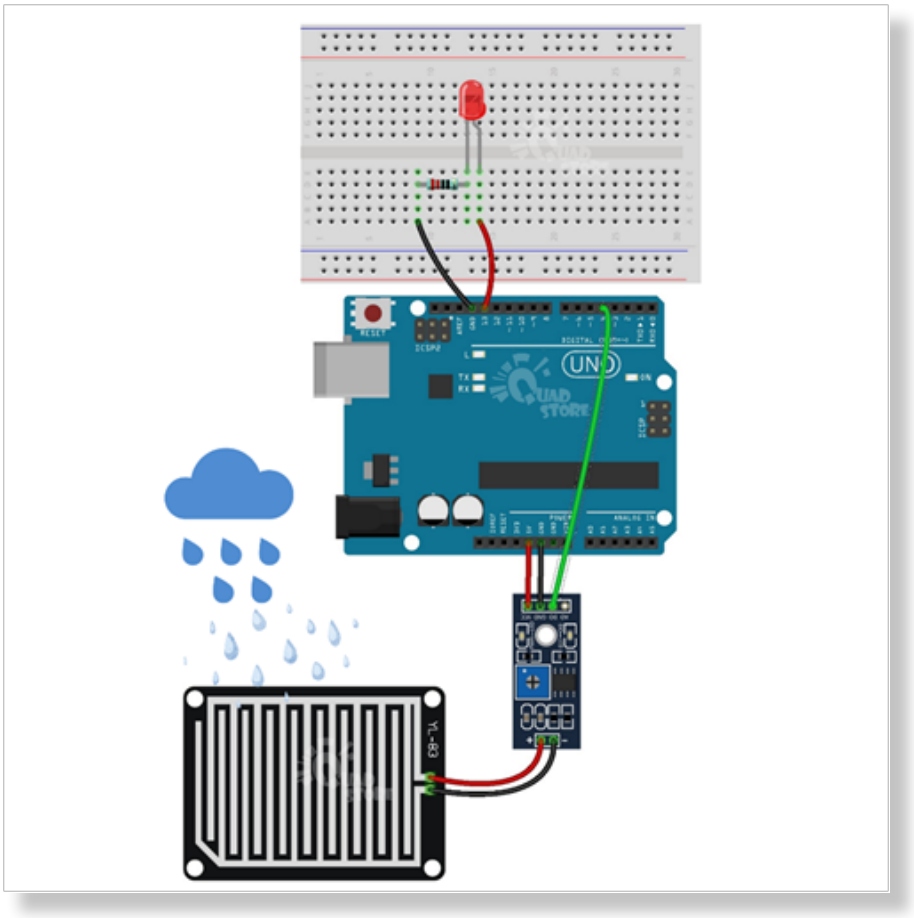
Detect rain to turn the LED on.

Parts Required:

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

Circuit Connection:

Provide the connection as per the below circuit.



Code:

Verify and Upload the code named "**Rain_Sensor.ino"** to Uno R3 board

Output:

When water droplets fall in sensor LED turns ON. You can replace LED with buzzer and give a try !.

Project 37: Soil Sensor

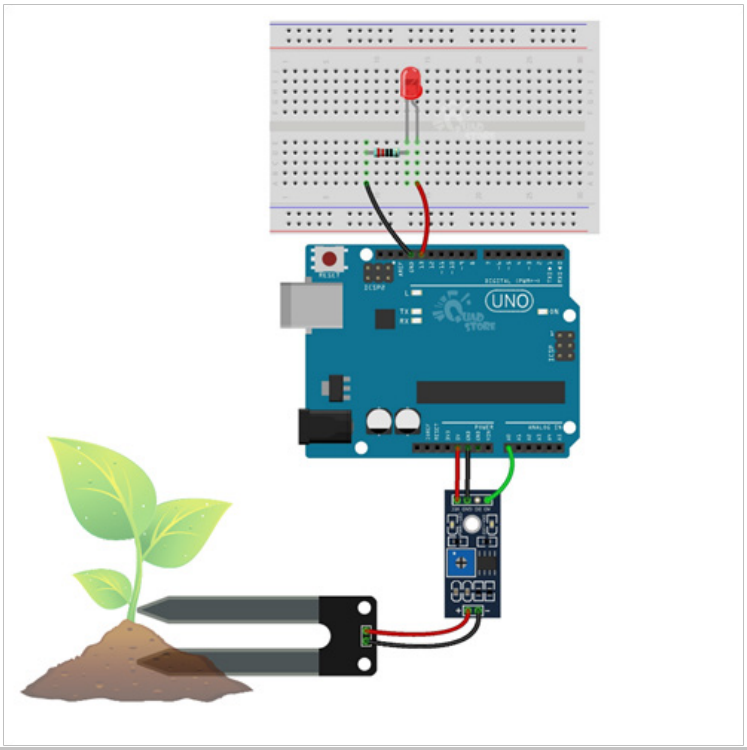
Validate moisture content in soil to turn on LED.

Parts Required:

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

Circuit Connection:

Provide the connection as per the below circuit.



Code:

Verify and Upload the code named "**Soil_Sensor.ino"** to Uno R3 board

Output:

Insert the sensor into the soil. When the dampness of the soil reduces LED will turn ON indication that you need to water the plants. When you water the soil LED will turn OFF

Note:

You can adjust the sensitivity of the sensor using the inbuilt potentiometer

Project 38: Touch Sensor

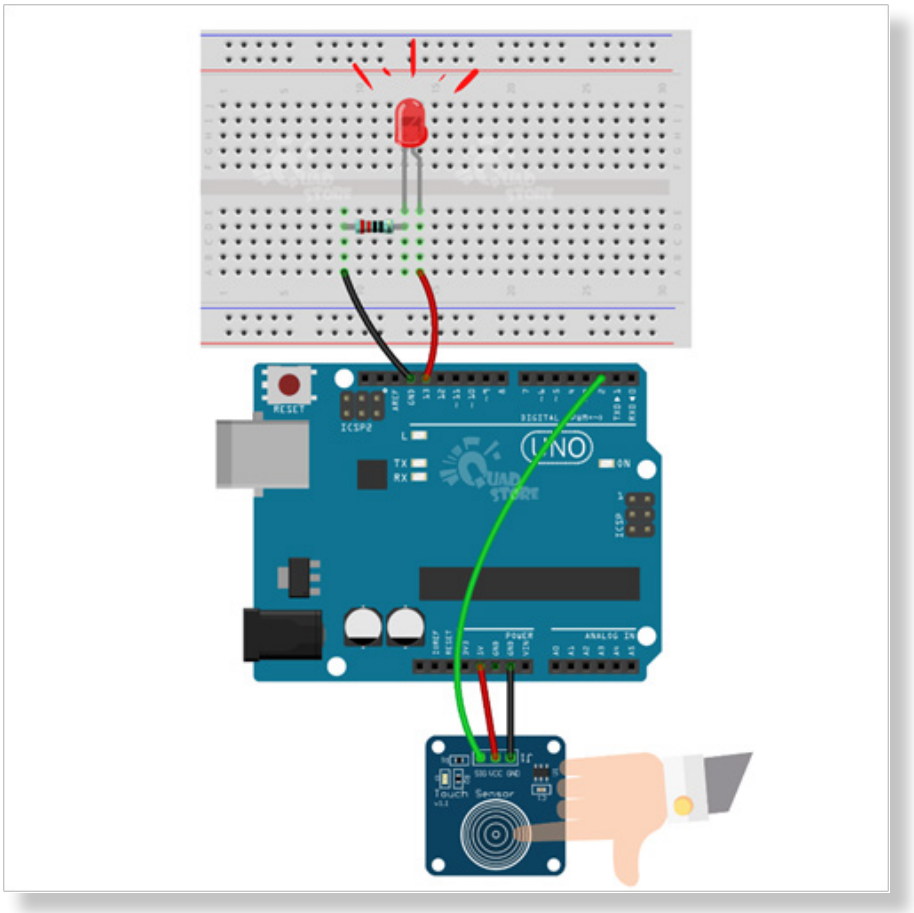
Touch the sensor to make the LED turn ON.

Parts Required:

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

Circuit Connection:

Provide the connection as per the below circuit.



Code:

Verify and Upload the code named "**Touch_Sensor.ino**" to Uno R3 board

Output:

Place you finger in touch sensor and you will see the LED turn ON. When you remove your finger LED will turn OFF.

Project 39: Bluetooth HC-05 or HC-06

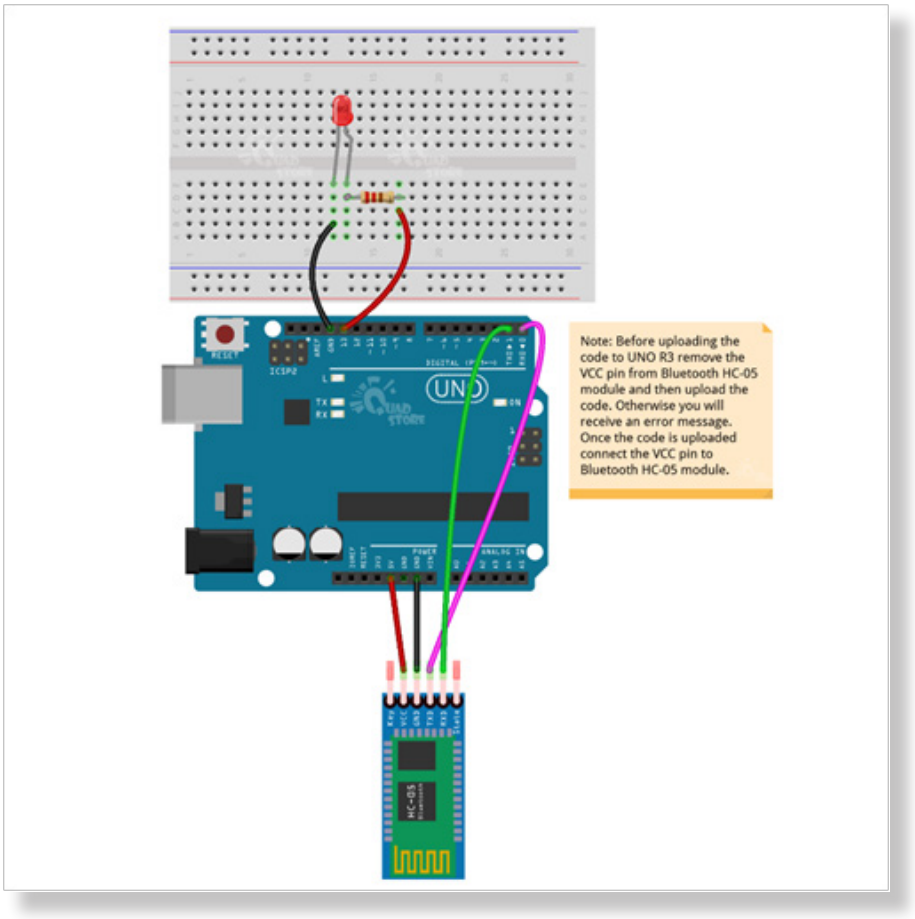
Control LED turn ON and OFF using Bluetooth mobile app.

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:

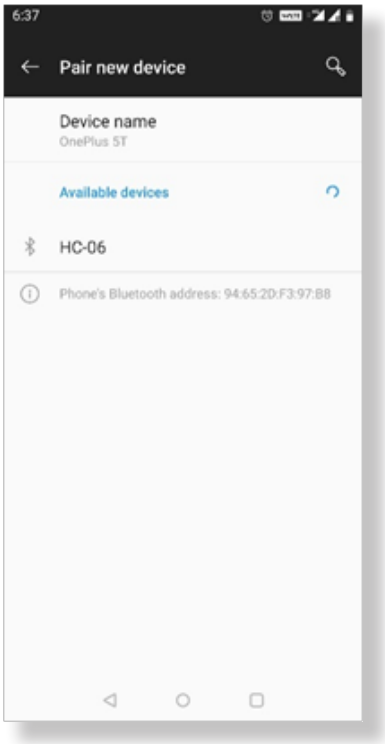
Provide the connection as per the below circuit. Connection is same for both HC-05 or HC-06 modules.



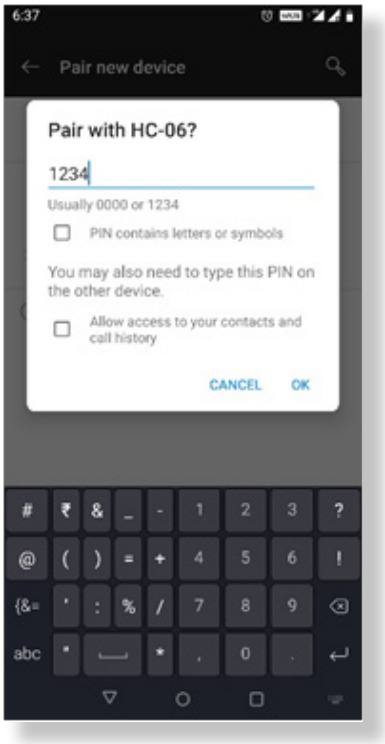
Follow the below steps to perform proper connection between phones Bluetooth app and HC-05/06 module. 1. Before uploading the code, you need to remove the VCC pin from Bluetooth HC-05/06 module.

2. **Code:** Verify and Upload the code named **"Bluetooth_HC05.ino"** to Uno R3 board

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.

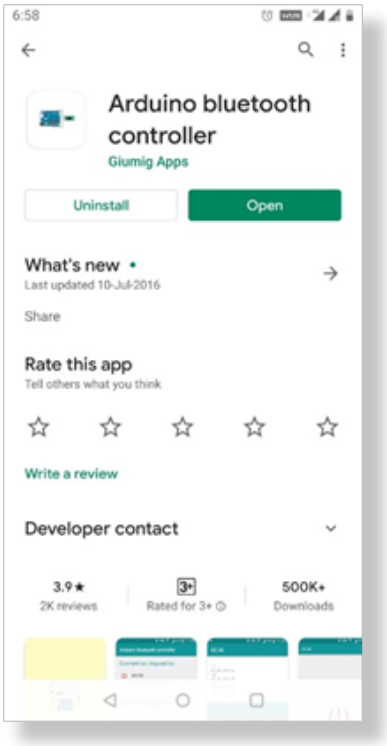


6. Click on HC-05 or HC-06 and it will prompt for password.
7. Enter the password as 1234 or 0000 and click OK

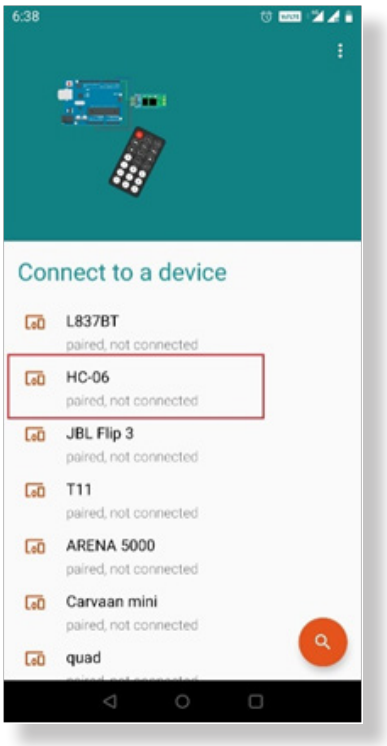


8. Device will be successfully paired with your mobile. But still you will see the led in the bluetooh HC-05 or HC-06 module blinking.

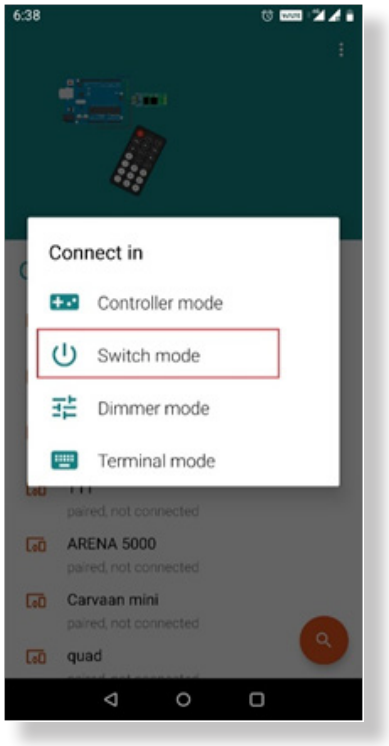
9. Install the app named "Arduino bluetooth Controller" from Playstore



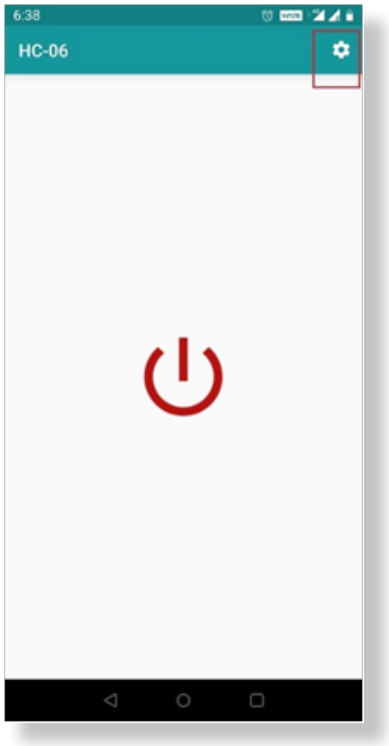
10. Open the app and under Connect to a device select HC-05 or HC-06



11. Connect in “Switch Mode” option

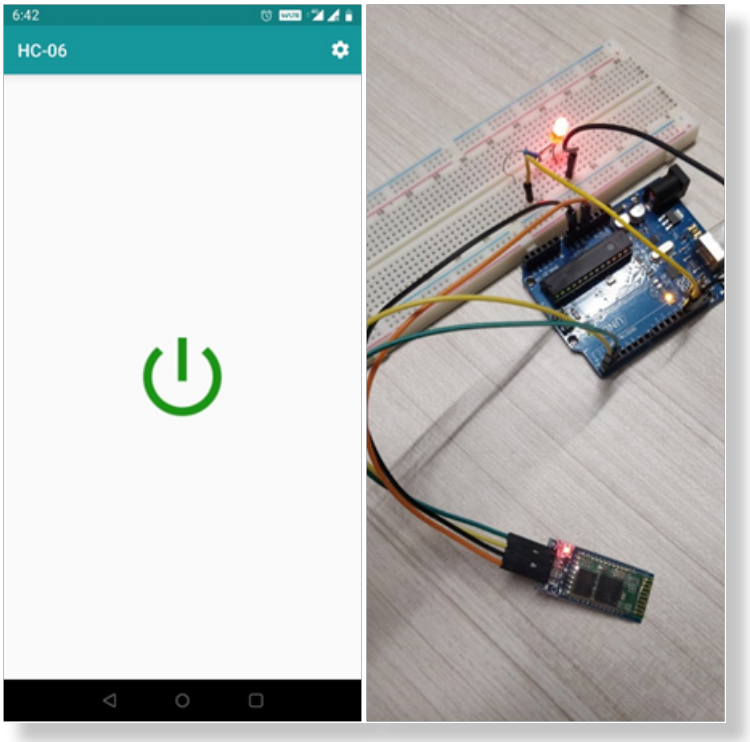


12. Click on Setting icon to configure the button options.

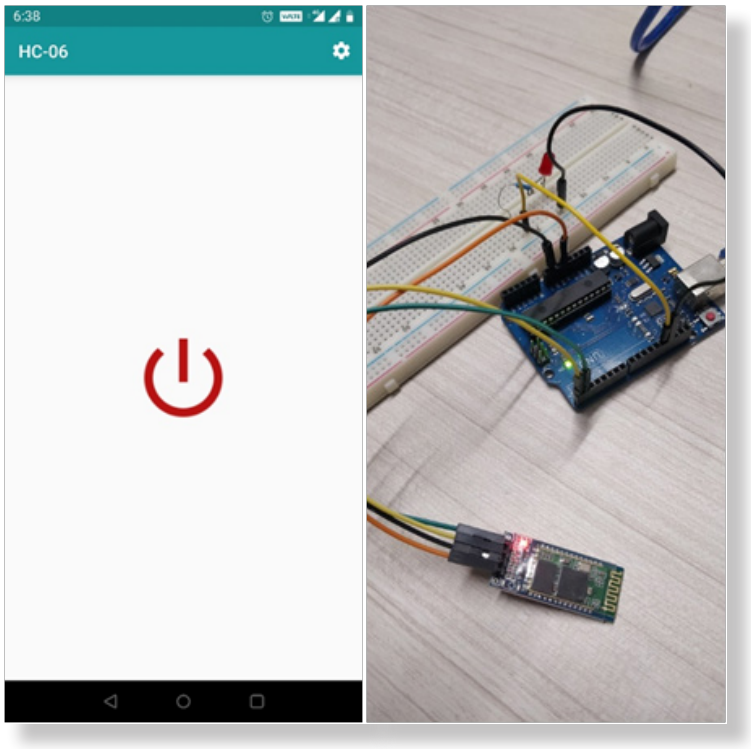


13. Enter x for ON button and y for OFF button and then go back to previous screen.

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



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



Project 40: L298N Motor Driver

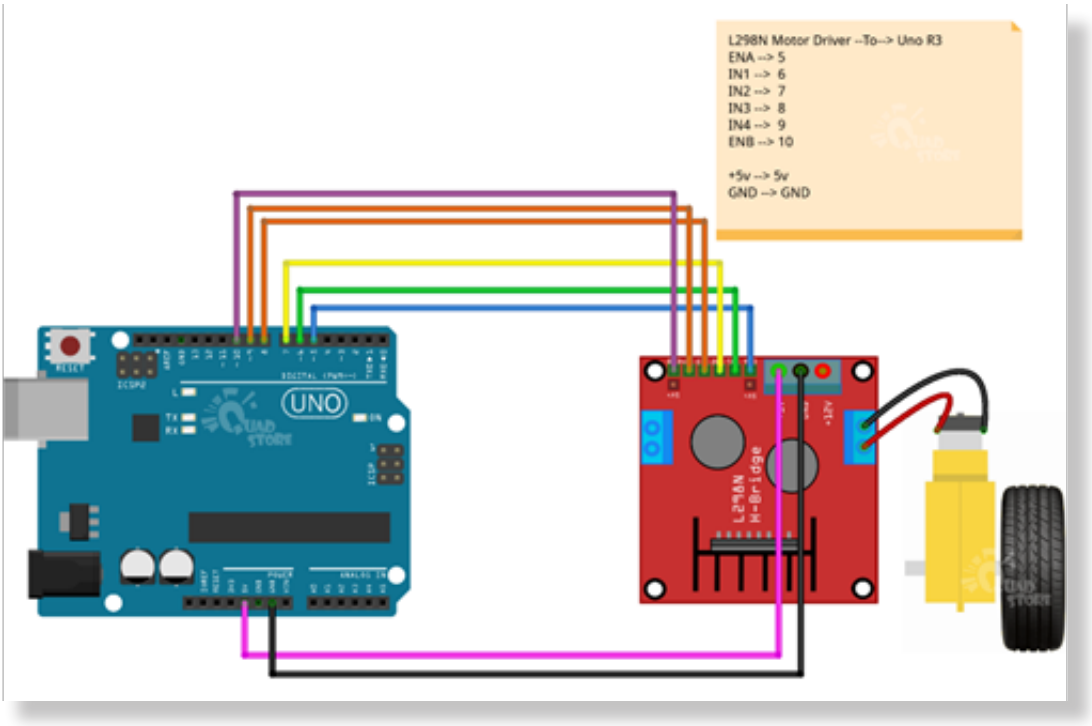
Make the DC motor run using L298N motor driver

Parts Required:

- Uno R3 board, USB Cable
- Few jumper wires
- L298N motor driver
- DC motor

Circuit Connection:

Provide the connection as per the below circuit.



Code:

Verify and Upload the code named "**L298N_Motor_Driver.ino**" to Uno R3 board

Output:

You will see the DC motor rotate clockwise and anticlockwise alternatively.

Project 41: L293D Motor Shield

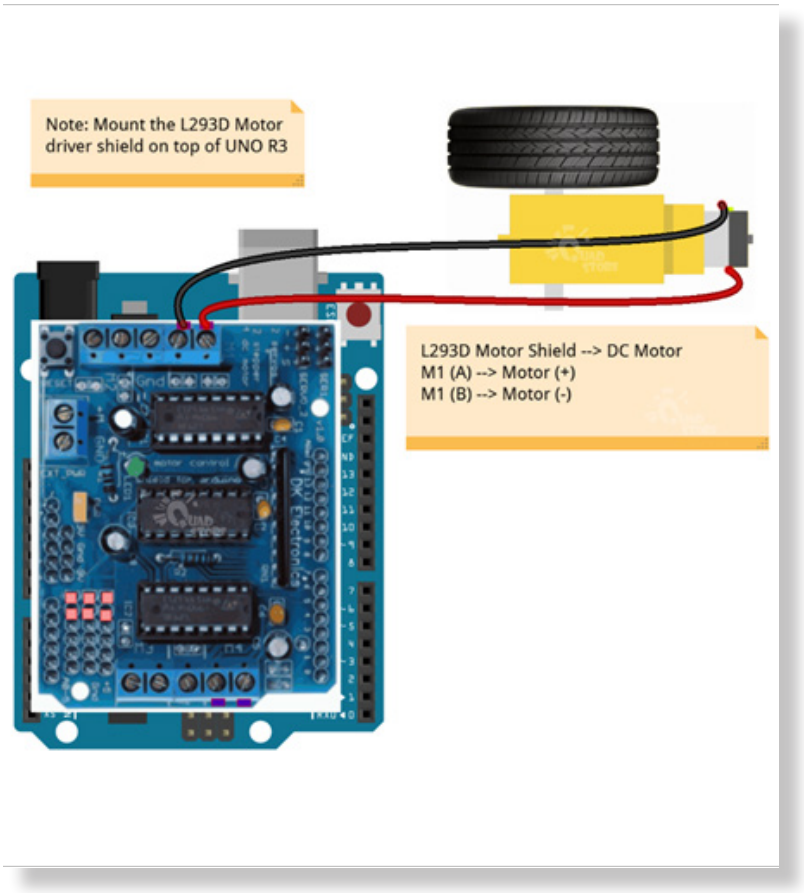
Make the DC motor run using L293D motor shield

Parts Required:

- Uno R3 board, USB Cable
- Few jumper wires

Circuit Connection:

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 42: L293D Motor Shield

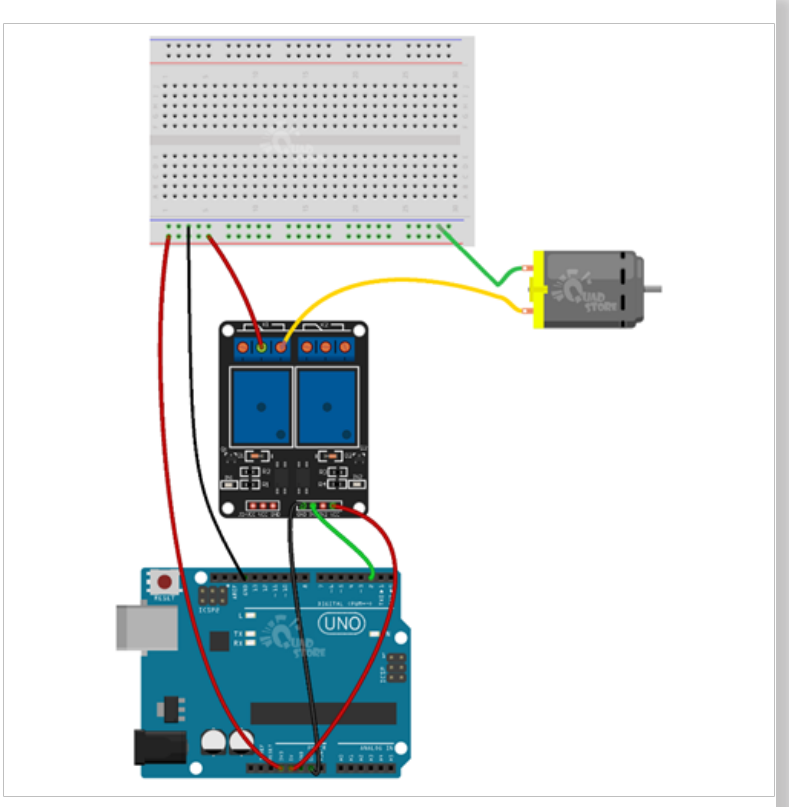
Make the DC motor run using relay module

Parts Required:

- Uno R3 board, USB Cable
- Breadboard
- Few jumper wires
- 1 Channel or 2 Channel relay
- DC Motor

Circuit Connection:

Provide the connection as per the below circuit.



Code:

Verify and Upload the code named "**DC_Motor.ino"** to Uno R3 board

Output:

When relay turns on the motor runs. You can control the timing of turning on and off the relay in the code. Give a try!

Project 43: Water Pump

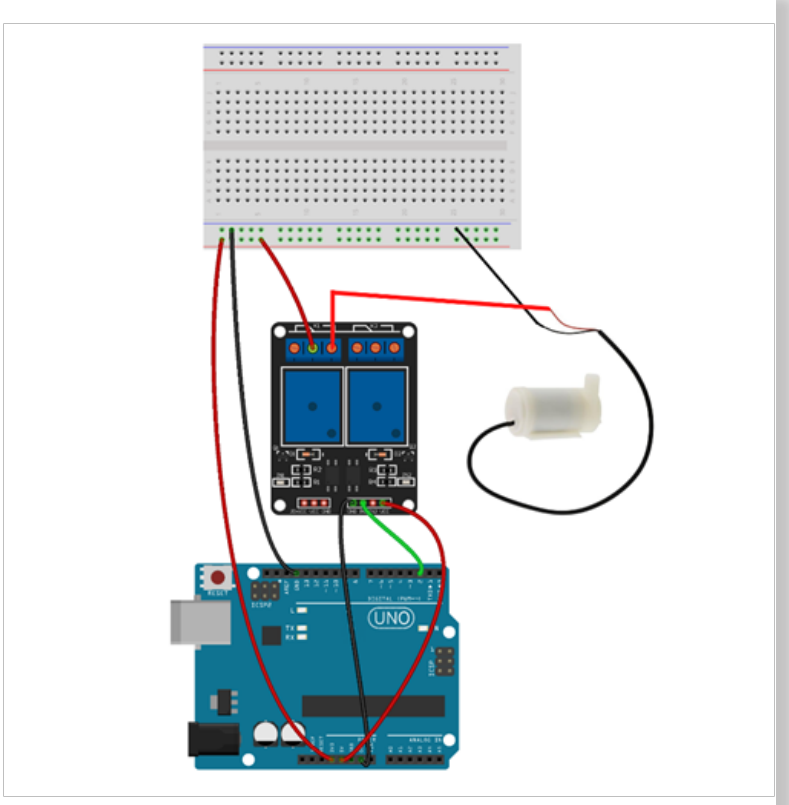
Make the water pump turn ON using relay module.

Parts Required:

- Uno R3 board, USB Cable
- Breadboard
- Few jumper wires
- 1 Channel or 2 Channel relay
- Water Pump Motor

Circuit Connection:

Provide the connection as per the below circuit.



Code:

Verify and Upload the code named "**Water_Pump.ino**" to Uno R3 board

Output:

When relay turns on the water pump runs. You can control the timing of turning on and off the relay in the code. Give a try!

Project 44:
Flame Sensor 2 pin

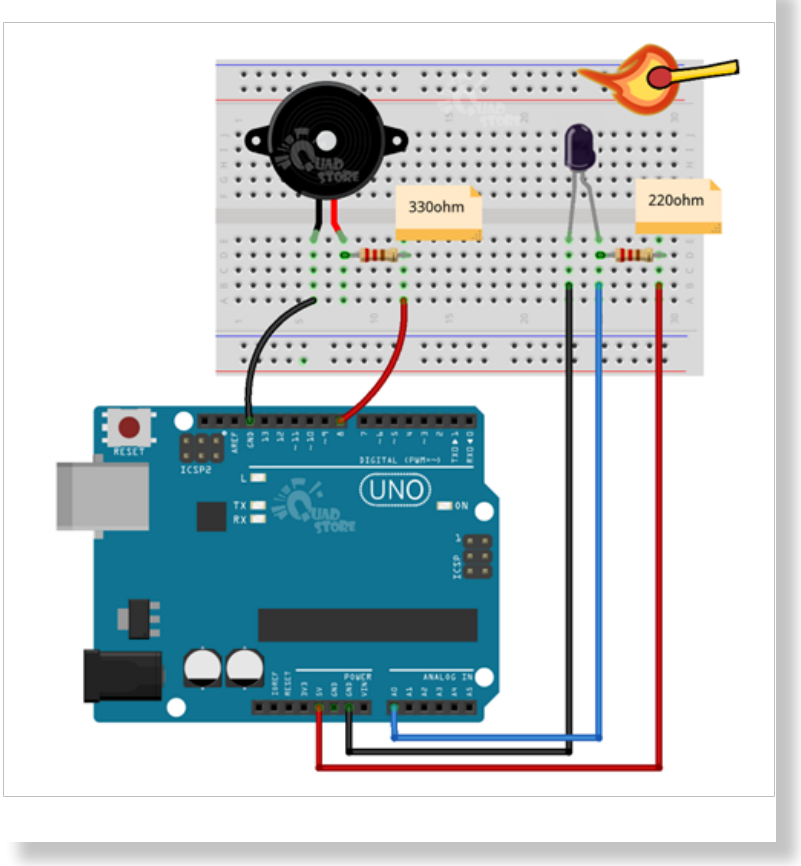
Make the buzzer beep during contact with flame.

Parts Required:

- Uno R3 board, USB Cable
- Breadboard
- Few jumper wires
- 2pin Flame Sensor
- Active Buzzer
- Resistor – 220ohm for flame sensor
- Resistor – 330ohm for Active Buzzer

Circuit Connection:

Provide the connection as per the below circuit.



Code:

Verify and Upload the code named "**Flame_Sensor_2pin"** to Uno R3 board

Output:

When you bring flame near the sensor buzzer will beep.

Project 45:
Flame Sensor 4 pin

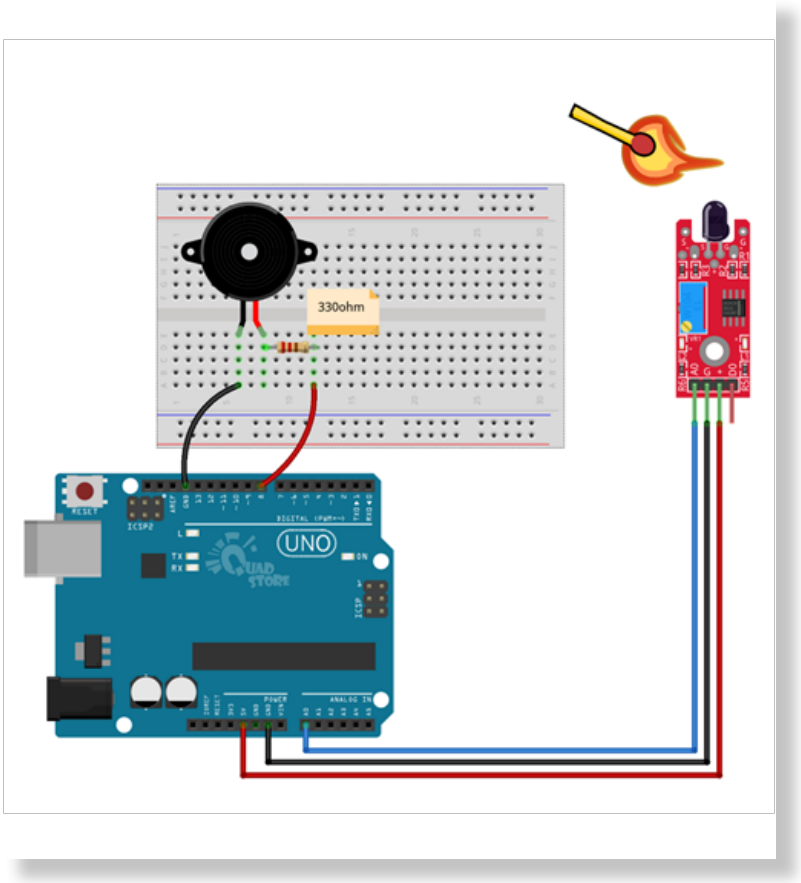
Make the buzzer beep during contact with flame.

Parts Required:

- Uno R3 board, USB Cable
- Breadboard
- Few jumper wires
- 4pin Flame Sensor
- Active Buzzer
- Resistor – 330ohm for Active Buzzer

Circuit Connection:

Provide the connection as per the below circuit.



Code:

Verify and Upload the code named "**Flame_Sensor_4pin.ino"** to Uno R3 board

Output:

When you bring flame near the sensor buzzer will beep.

Project 46:
Wifi ESP8266-01 with Blynk app

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

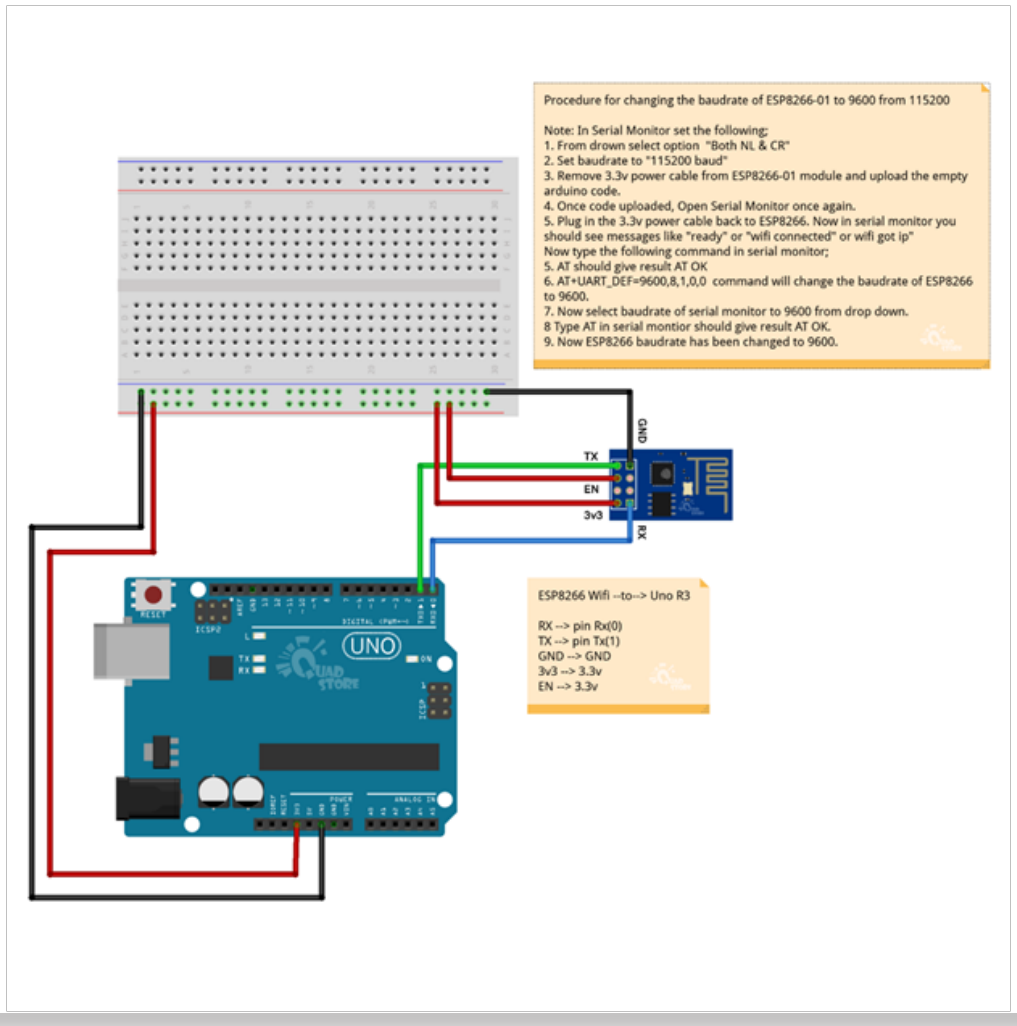
Parts Required:

- 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.

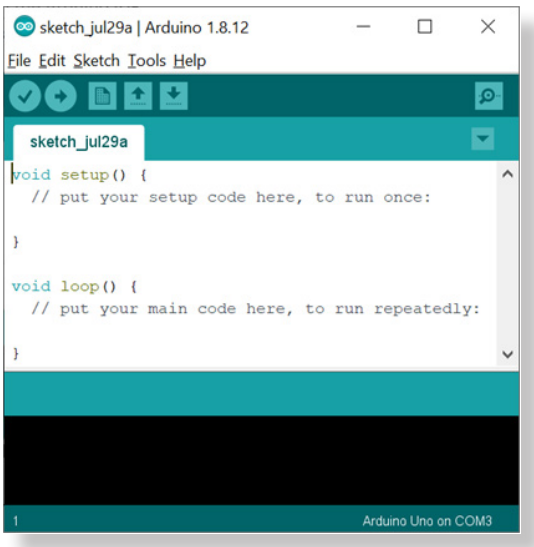
First circuit connection for changing the baudrate of ESP8266-01 module:

Provide the connection as per the below circuit.

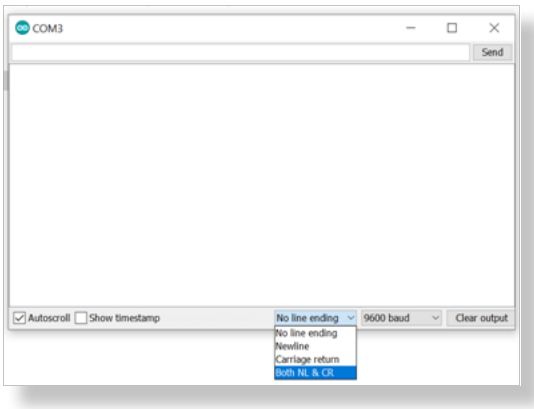


ACTION-1: PROCEDURE FOR CHANGING THE BAUDRATE OF ESP8266-01 TO 9600 FROM 115200

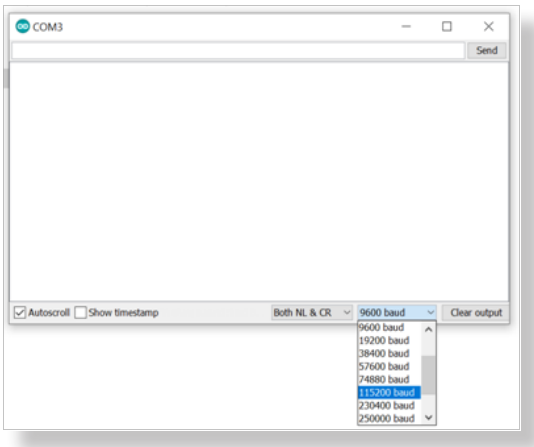
1. Open New Sketch from the Arduino IDE.



2. Open Serial Monitor andchange the following options
3. From drown select option “Both NL & CR"

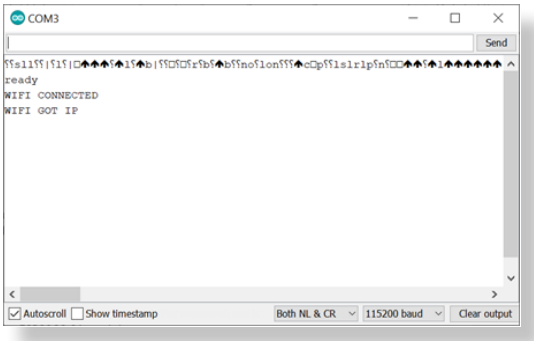


4. Set baudrate to "115200 baud"

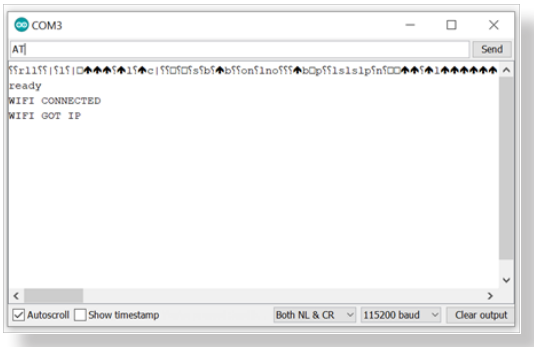


5. Remove **3.3v power cable** from ESP8266-01 module.
6. Upload the New empty Sketch/Code to the Uno R3 board.

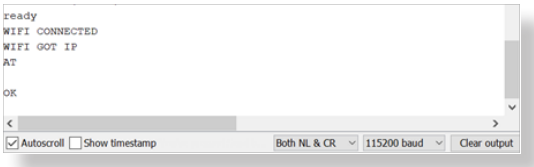
7. Once code uploaded, Open Serial Monitor once again.
8. Plug in the **3.3v power cable** back to ESP8266. Now in serial monitor you should see messages like "ready" or "WIFI CONNECTED" or "WIFI GOT IP"



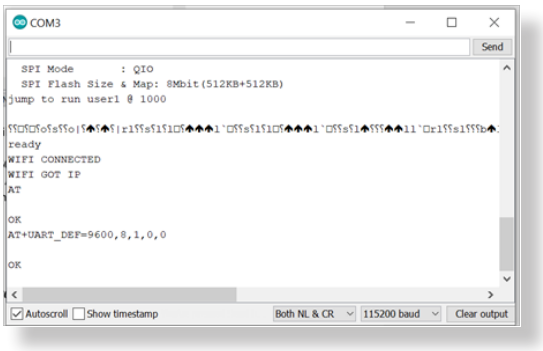
9. Now type the commandAT in serial monitor and click Send button.



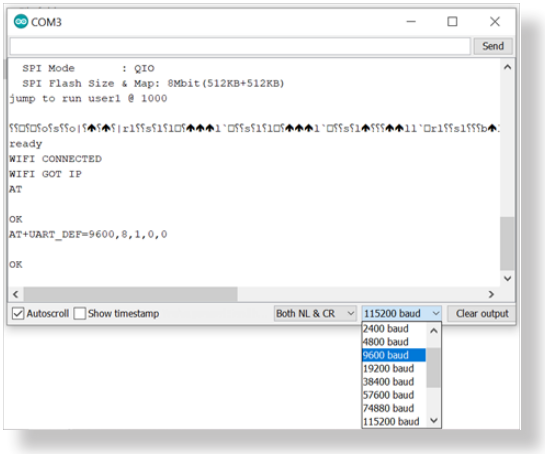
10. AT should give result AT OK



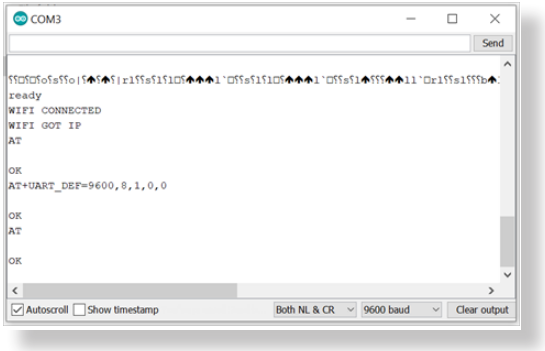
11. Now type command AT+UART_DEF=9600,8,1,0,0 command will change the baudrate of ESP8266 to 9600.



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



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

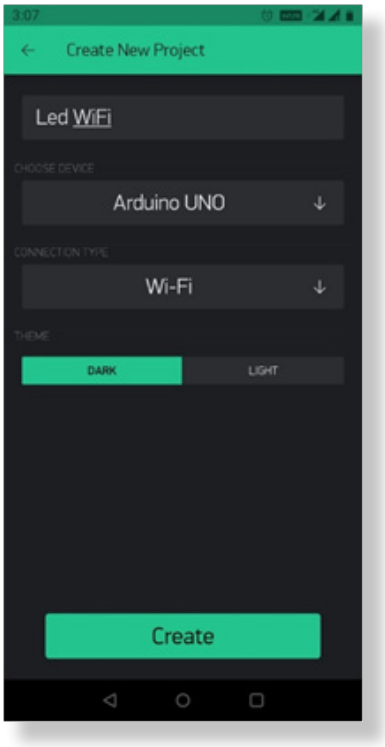


14. Success! Now the baudrate of ESP8266-01 has been changed to 9600.

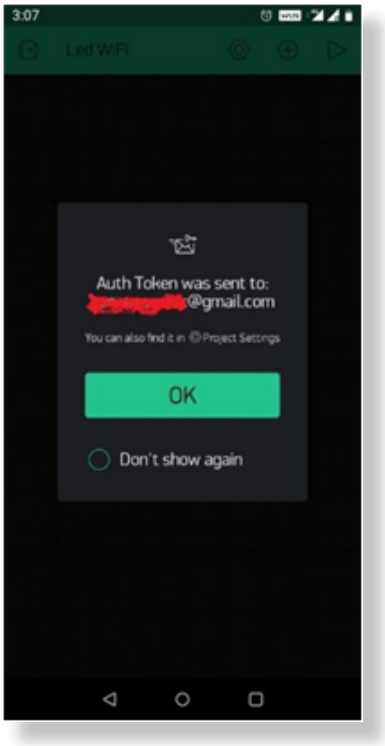
Installation and setup of Blynk App:

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

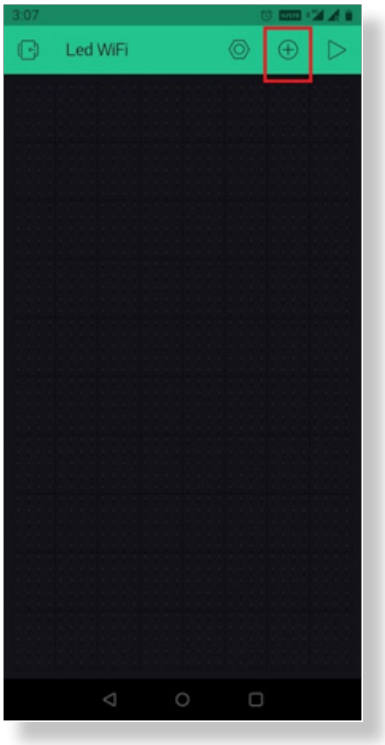
3. Enter the "Project Name" as Led Wifi.
4. Select the "Board Type" as Arduino Uno and click Create



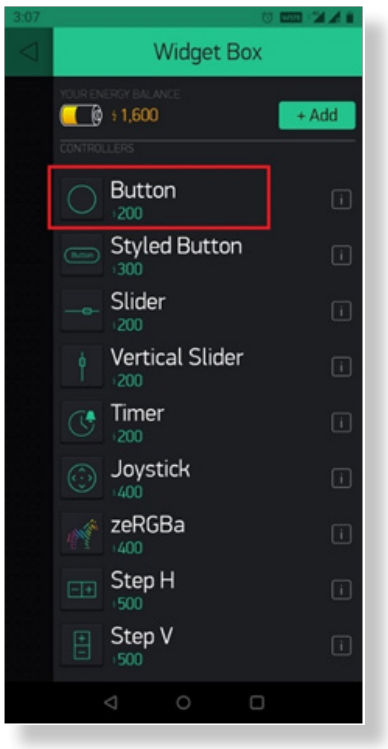
5. Auth token will be generated and sent to your email address.



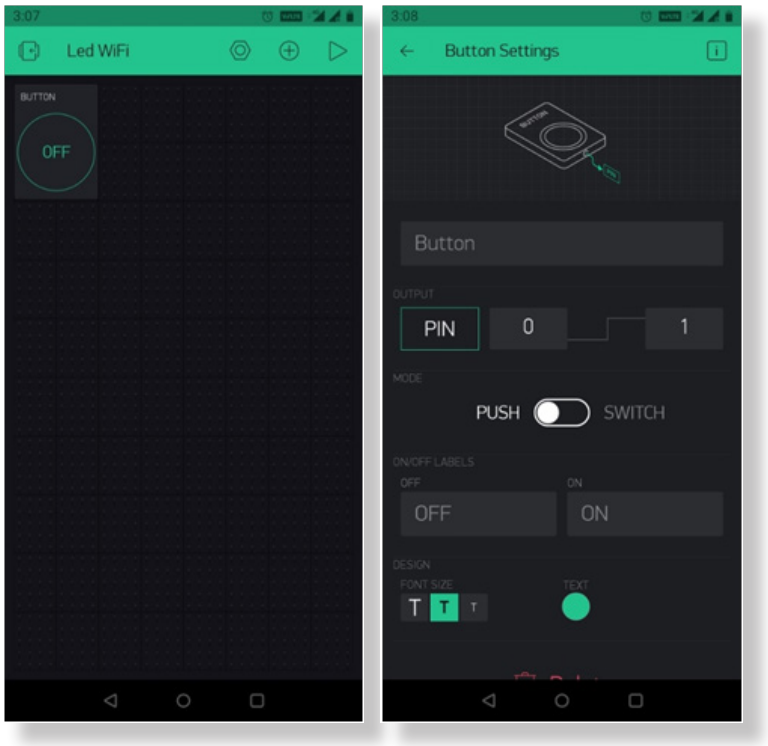
6. Click on the + icon to open widget box



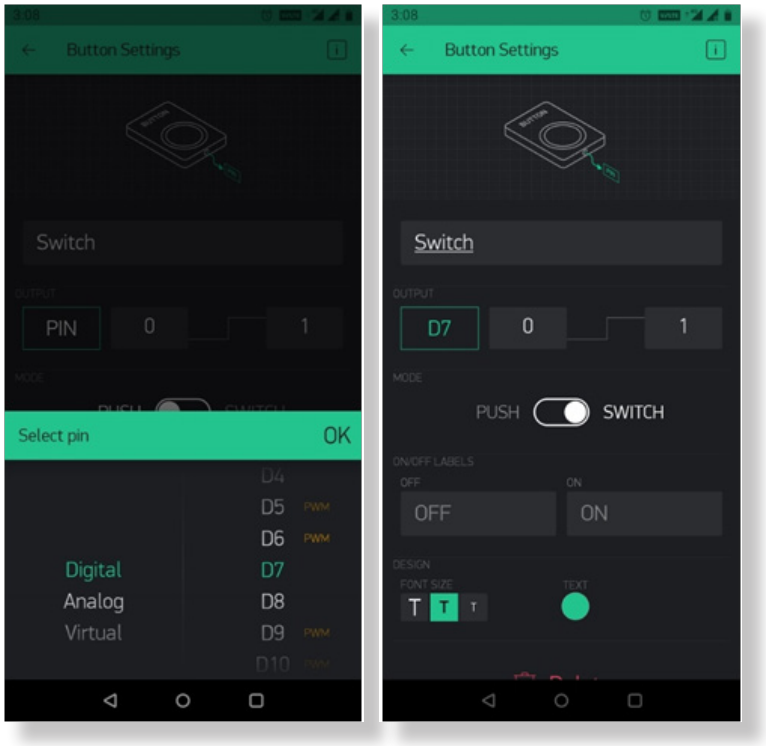
7. Add a new widget “Button”



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



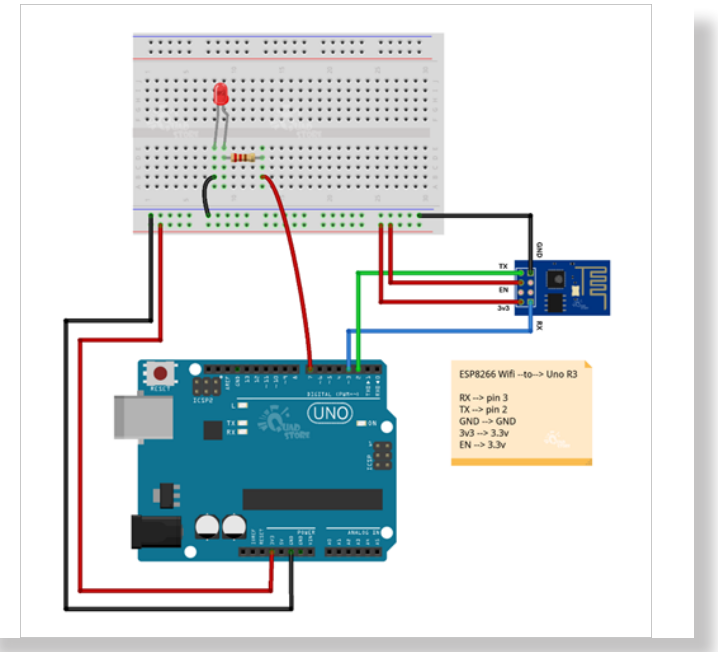
9. Change the PIN to “Digital - pin 7”, rename the widget to “Switch” and change the PUSH option to SWITCH and go back.



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.



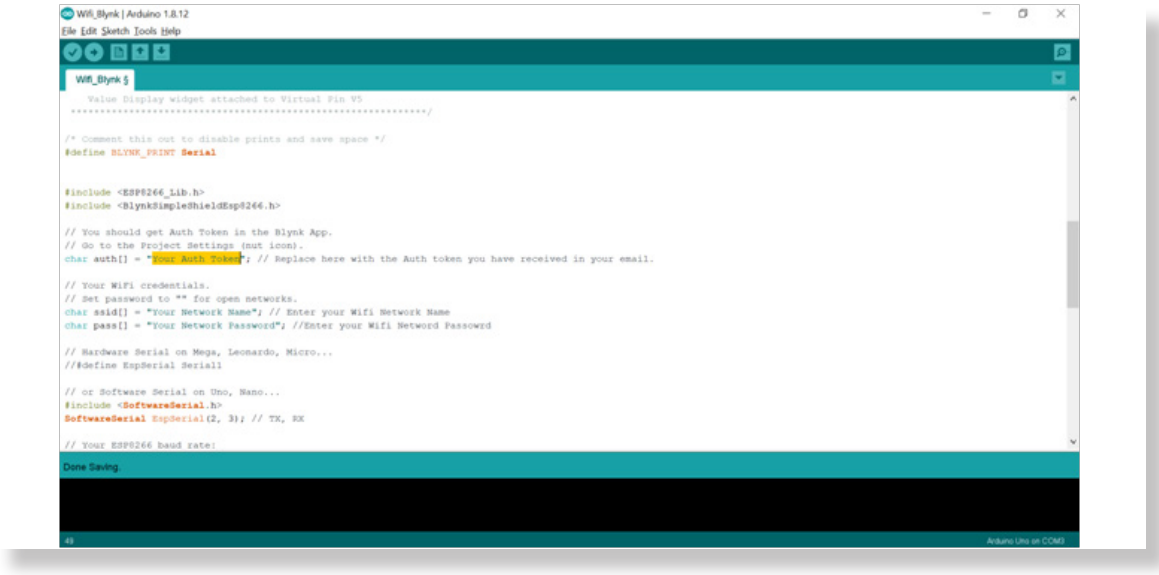
Update your CODE:

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"**

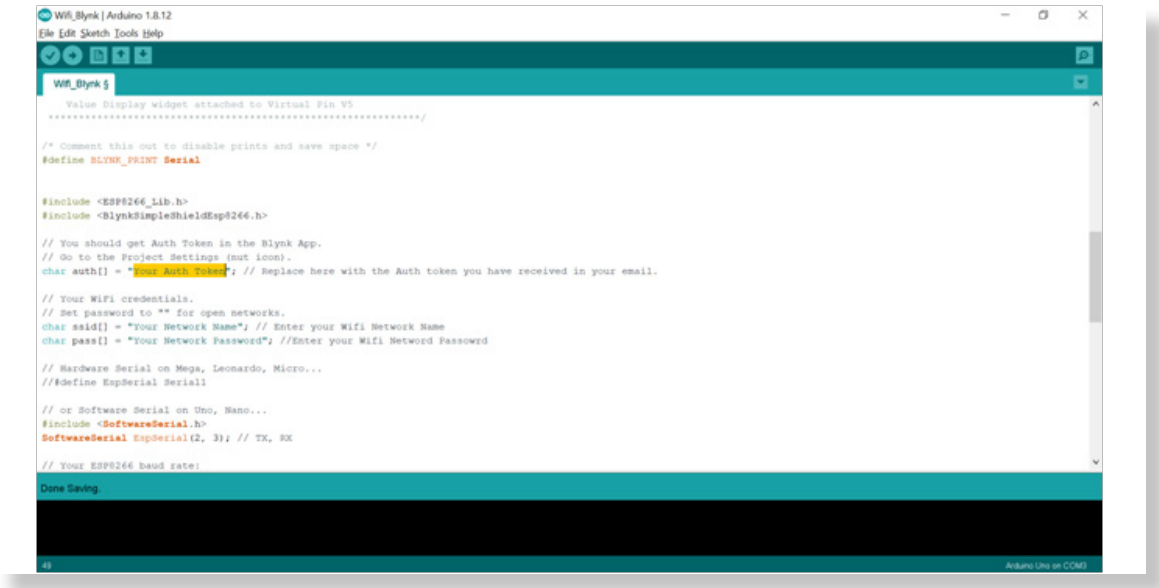
3. Make the following changes in your CODE to reflect the correct **"Auth Token", "Network Name"** and **"Network Password"**



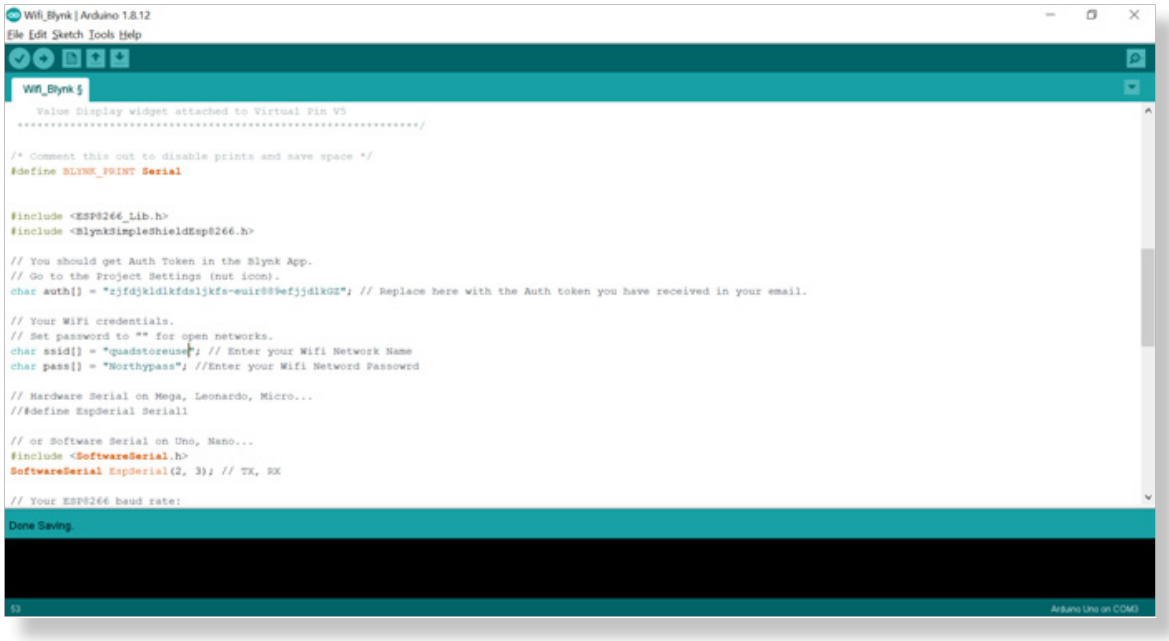
Note:

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".



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



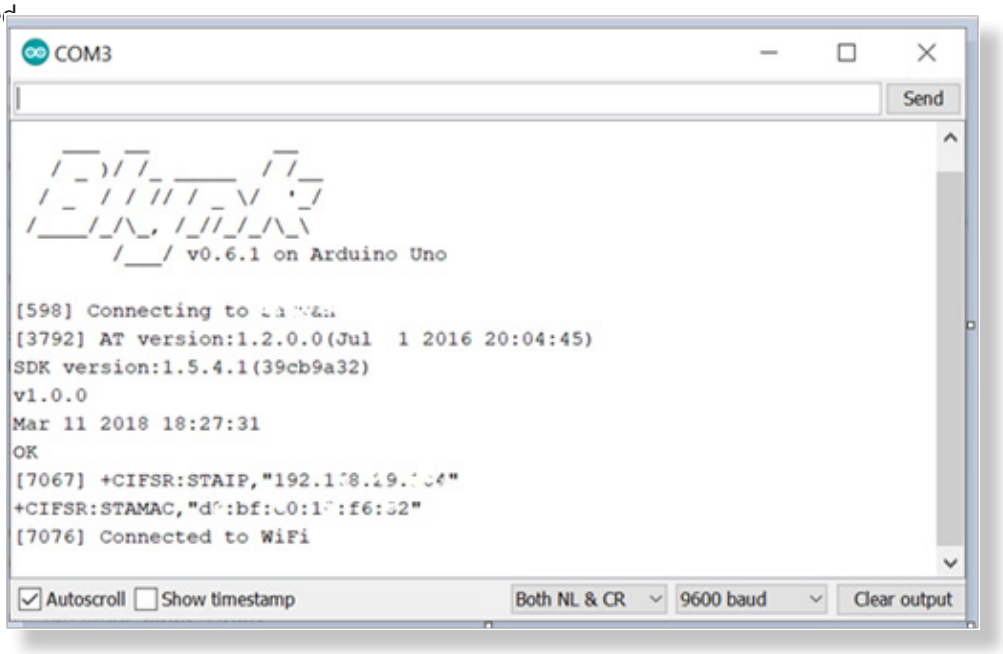
6. Now per form the below steps to get connected with the wifi module:

7. Remove 3.3v power cable from ESP8266-01 module and then upload the updated code.

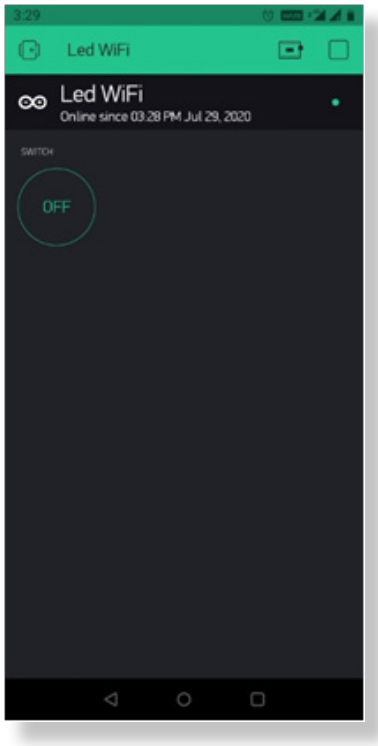
8. Plug in the 3.3v power cable back to ESP8266-01 module.

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

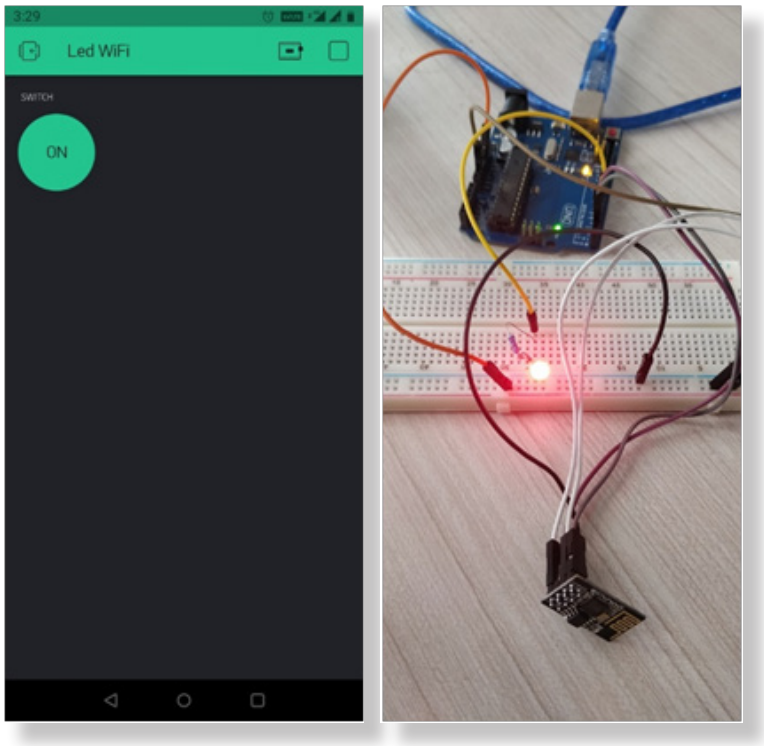


11. Open Blynk app and you should see the project getting connected to your Uno R3 board through wifi automatically.

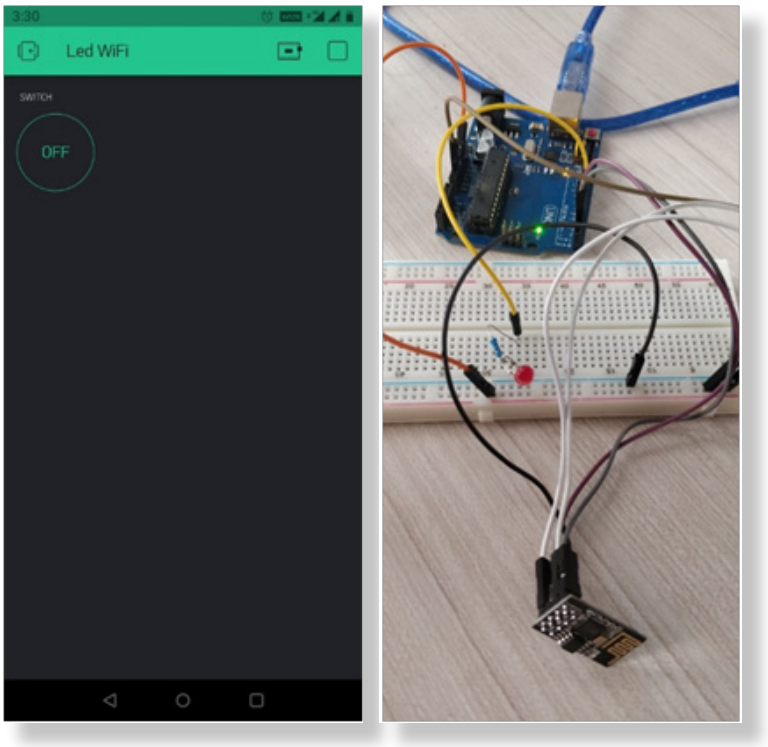


12. Output: Click ON and OFF button through Blynk app to control the LED.

13. When ON button is clicked you would see the LED turn ON through Wifi.



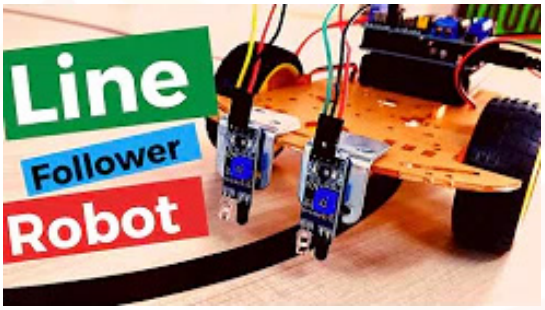
Similarly, when OFF button is clicked the LED turns OFF through Wifi.



Now time to build other project by your own !

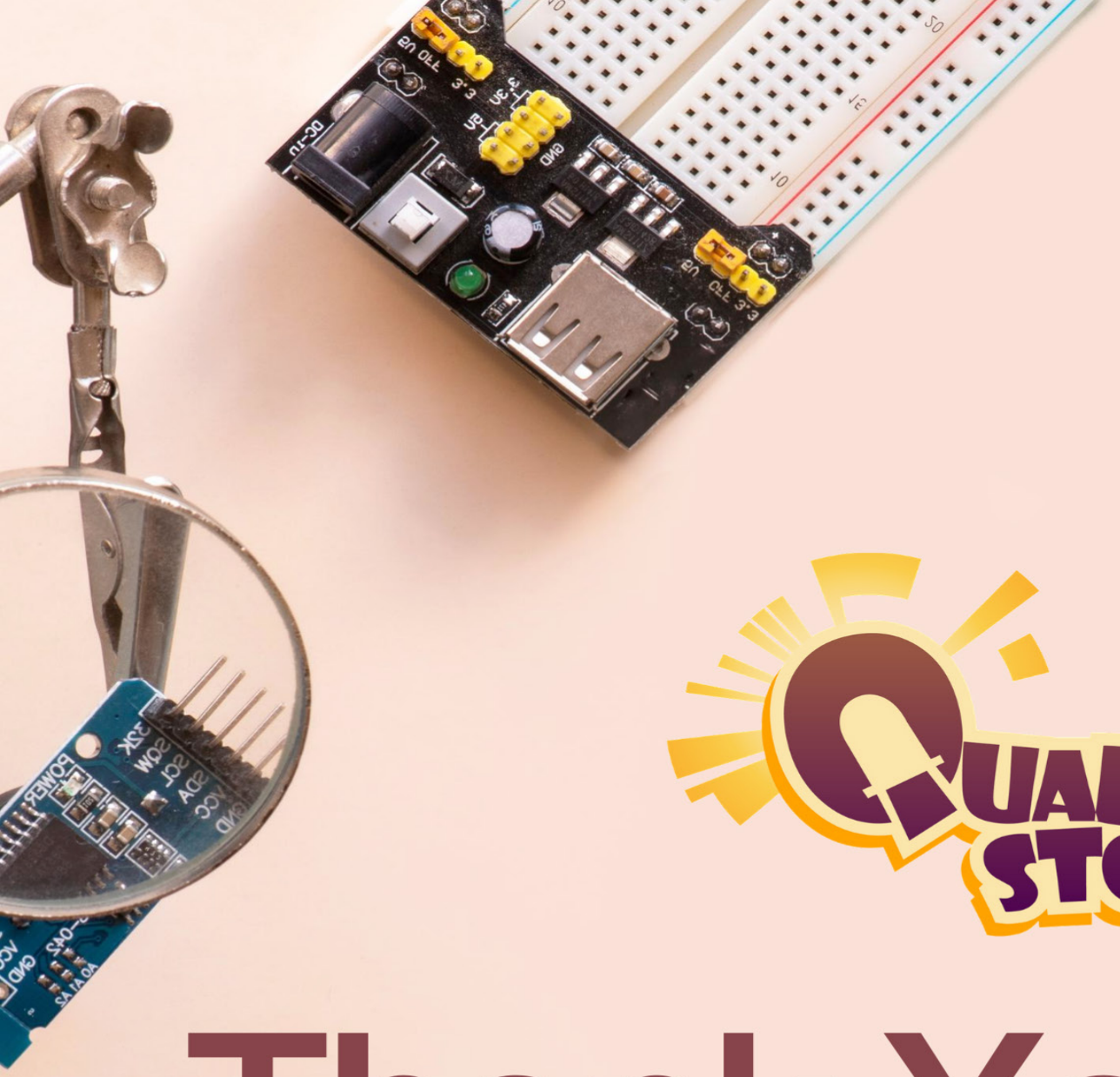
Check out Quad Store's Youtube page for other interesting projects like:

Smart dustbin



Line following robot

Obstacle Avoiding robot etc.



QUAD STORE

Thank You