

**Name:**Vidya Jethwa  
**Class:**MSc CS I

**AcademicYear:**2021-2022  
**RollNo. :**7

## **Advanced Embedded**

### **SystemsIndex**

<b>Sr.No.</b>	<b>Title</b>	<b>PageNo.</b>
1	SwitchingONandOFFLEDusingPushbuttonwithArduinoUNO.	2
2	Using 7-segment display with Arduino UNO to display from 0 to 9 after specific interval of time.	5
3	Configuring a digital object counter device using 7-segment display with Arduino UNO and IR proximity sensor.	10
4	Print message on LCD display with Arduino UNO.	17
5	Use 4x4 keypad to give the input in Arduino UNO serial monitor.	21
6	Interfacing of buzzer with Arduino UNO.	24
7	Interfacing of ultrasonic sensor with Arduino UNO.	27
8	Interfacing of servomotor with Arduino UNO.	31
9	Interfacing of DHT11 with Arduino UNO to read temperature and humidity which is then printed on serial monitor.	34
10	Interfacing of LED with NodeMCU and controlling it remotely with Blynk application on mobile.	37

# Practical1

**Aim:** Switching ON and OFF LED using Push button with Arduino UNO.

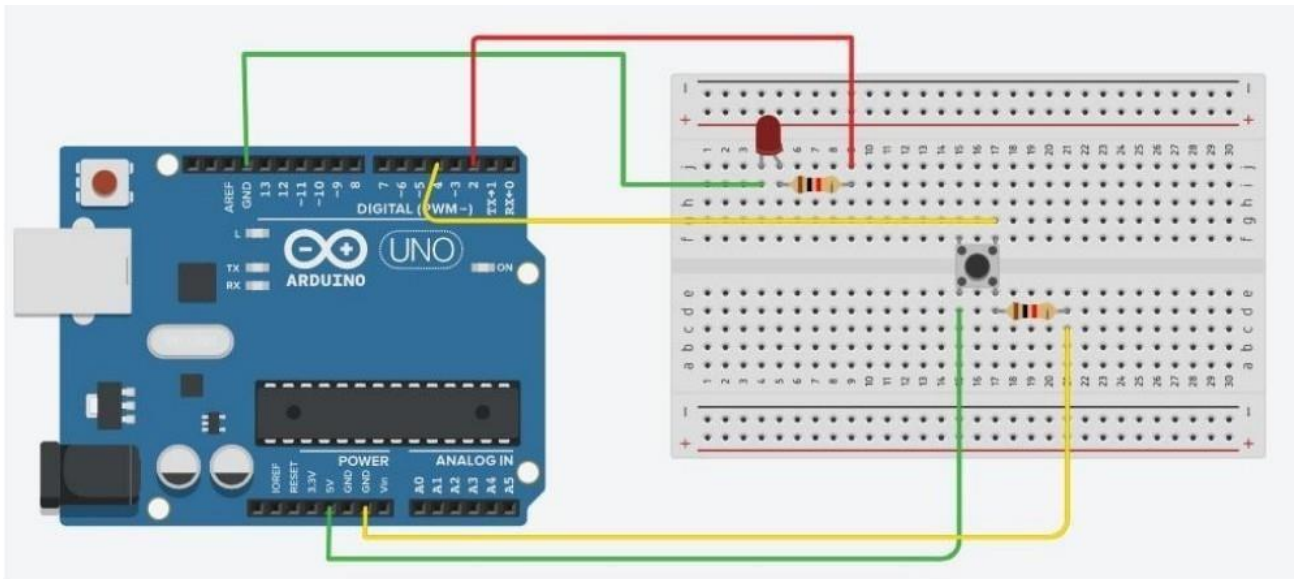
## Description:

- **Arduino:**
  - Arduino is an open-source platform used for building electronics projects.
  - Arduino consists of both a physical programmable circuit board and a piece of software, or IDE runs on your computer, used to write and upload computer code to the physical board.
  - Arduino UNO has 14 digital pins and 6 analog pins.
- **Breadboard:**
  - It is a way of constructing electronics without having to use a soldering iron.
  - Components are pushed into the sockets on the breadboard and then extra 'jumper' wires are used to make connections.
- **Push button:**
  - The push button is a component that connects two points in a circuit when you press it.
- **LED:**
  - A light-emitting diode (LED) is a semiconductor device that produces light from electricity. LEDs last a long time and do not break easily.

## Required Components:

- Arduino Uno (1x)
- Breadboard (1x)
- LED (1x)
- Resistors (2x)
- Push Button (1x)
- Jump Wires

## CircuitDiagram:



## Connections:

Components	Connections	
	From	To
ArduinotoPushButton	5V	Pin1(Usingbreadboard)
	GND	Pin2(Usingbreadboardandthroughresistance)
	Pin4	Pin3(Usingbreadboard)
ArduinotoLED	Pin2	Positive end (Usingbreadboardandthroughresistance)
	GND	Negativeend(Usingbreadboard)

## SourceCode:

```
const int ledpin =
2;const int btnpin =
4;intbtnstate=0;

voidsetup(){
  // put your setup code here, to run
  once:Serial.begin(9600);
  pinMode(ledpin,
  OUTPUT);pinMode(ledpin,I
  NPUT);
}

voidloop(){
  // put your main code here, to run
  repeatedly:btnstate=digitalRead(btnpin);
  if(btnstate==HIGH)
  {
    digitalWrite(ledpin,
    HIGH);Serial.println("LEDON
    ");
  }
  else
  {
    digitalWrite(ledpin,
    LOW);Serial.println("LEDOF
    F");
  }
}
```

# Practical2

**Aim:** Using 7-segment display with Arduino UNO to display from 0 to 9 after specific interval of time.

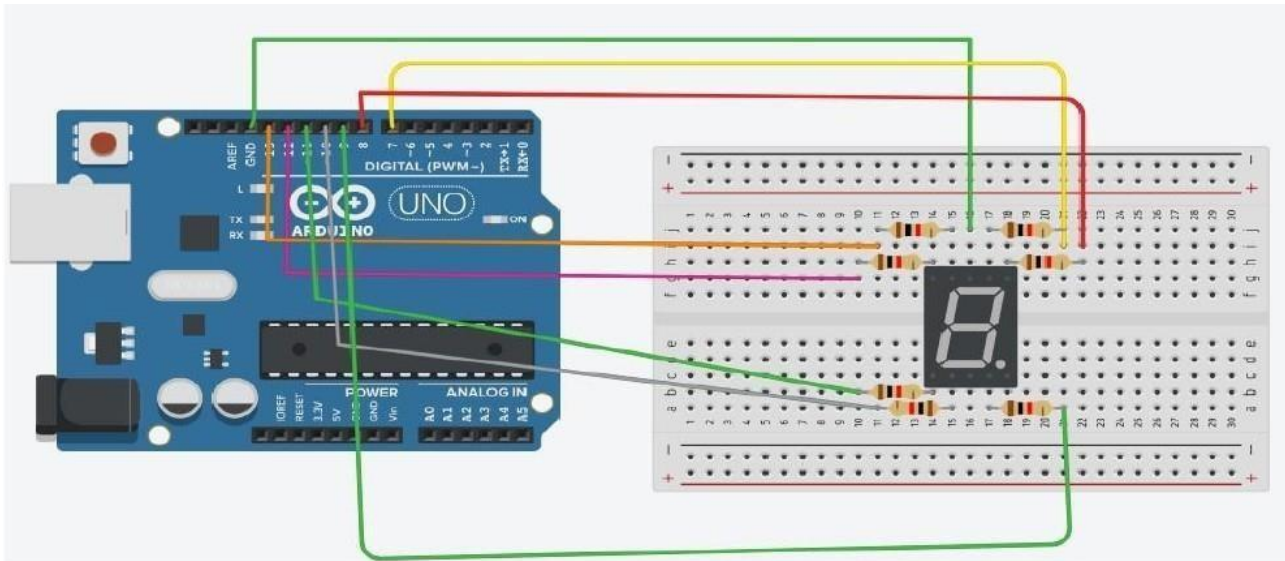
## Description:

- **Arduino:**
  - Arduino is an open-source platform used for building electronics projects.
  - Arduino consists of both a physical programmable circuit board and a piece of software, or IDE runs on your computer, used to write and upload computer code to the physical board.
  - Arduino UNO has 14 digital pins and 6 analog pins.
- **Breadboard:**
  - It is a way of constructing electronics without having to use a soldering iron.
  - Components are pushed into the sockets on the breadboard and then extra 'jumper' wires are used to make connections.
- **Seven-segment:**
  - The seven-segment display has seven LEDs arranged in the shape of number eight.
- **Resistors:**
  - It's a passive two-terminal electrical component that implements electrical resistance as a circuit element.
  - In electronic circuits, resistors are used to reduce current flow, adjust signal levels, to divide voltages, bias active elements, and terminate transmission lines, among other uses.

## Required Components:

- ArduinoUno (1x)
- Breadboard(1x)
- Seven-segmentDisplay(1x)
- Resistors(7x)
- Jumpwires

## CircuitDiagram:



## Connections:

Components	Connections	
	From	To
Arduinoto7-segmentDisplay	GND	COMorGND
	13	Pinf(throughresistance)
	12	Ping(throughresistance)
	11	Pine(throughresistance)
	10	Pind(throughresistance)
	9	Pinc(throughresistance)
	8	Pinb(throughresistance)
	7	Pina(throughresistance)

## SourceCode:

```
int f =
13;int g =
12;int e =
11;int d =
10;int c =
9;int b =
8;inta=7;
int delay_ms =
1000;int count =
0x00;voidsetup(){
// put your setup code here, to run
once:pinMode(13,OUTPUT);
pinMode(12,
OUTPUT);pinMode(11,
OUTPUT);pinMode(10,
OUTPUT);pinMode(9,
OUTPUT);pinMode(8,
OUTPUT);pinMode(7,OU
TPUT);
}

voidloop(){
    //putyourmaincodehere,torunrepeatedly:

    digitalWrite(a,1);
    digitalWrite(b,1);
    digitalWrite(c,1);
    digitalWrite(d,1);
    digitalWrite(e,1);
    digitalWrite(f,1);
    digitalWrite(g,

0);delay(delay_ms);//

0

    digitalWrite(a,0);
    digitalWrite(b,1);
    digitalWrite(c,1);
    digitalWrite(d,0);
    digitalWrite(e,0);
    digitalWrite(f,0);
    digitalWrite(g,

0);delay(delay_ms);
```

```
//1digitalWrite(a,1)
```

```
;
```



```
digitalWrite(b,1);  
digitalWrite(c,0);  
digitalWrite(d,1);  
digitalWrite(e,1);  
digitalWrite(f,0);  
digitalWrite(g,
```

```
1);delay(delay_ms);//
```

2

```
digitalWrite(a,1);  
digitalWrite(b,1);  
digitalWrite(c,1);  
digitalWrite(d,1);  
digitalWrite(e,0);  
digitalWrite(f,0);  
digitalWrite(g,
```

```
1);delay(delay_ms);//
```

3

```
digitalWrite(a,0);  
digitalWrite(b,1);  
digitalWrite(c,1);  
digitalWrite(d,0);  
digitalWrite(e,0);  
digitalWrite(f,1);  
digitalWrite(g,1);
```

```
delay(delay_ms);
```

```
//4digitalWrite(a,1)
```

```
;  
digitalWrite(b,0);  
digitalWrite(c,1);  
digitalWrite(d,1);  
digitalWrite(e,0);  
digitalWrite(f,1);  
digitalWrite(g,1);
```

```
delay(delay_ms);
```

```
//5digitalWrite(a,0)
```

```
;  
digitalWrite(b,0);  
digitalWrite(c,1);  
digitalWrite(d,1);  
digitalWrite(e,1);  
digitalWrite(f,1);  
digitalWrite(g,1);
```

```

delay(delay_ms); //6

digitalWrite(a,1);
digitalWrite(b,1);
digitalWrite(c,1);
digitalWrite(d,0);
digitalWrite(e,0);
digitalWrite(f,0);
digitalWrite(g,

0);delay(delay_ms); //

7

digitalWrite(a,1);
digitalWrite(b,1);
digitalWrite(c,1);
digitalWrite(d,1);
digitalWrite(e,1);
digitalWrite(f,1);
digitalWrite(g,

1);delay(delay_ms); //

8

digitalWrite(a,1);
digitalWrite(b,1);
digitalWrite(c,1);
digitalWrite(d,0);
digitalWrite(e,0);
digitalWrite(f,1);
digitalWrite(g,1);

delay(delay_ms); //9
}

```

# Practical3

**Aim:**Configuringadigitalobjectcounterdeviceusing7-segmentdisplaywithArduinoUNOandIRproximitysensor.

## Description:

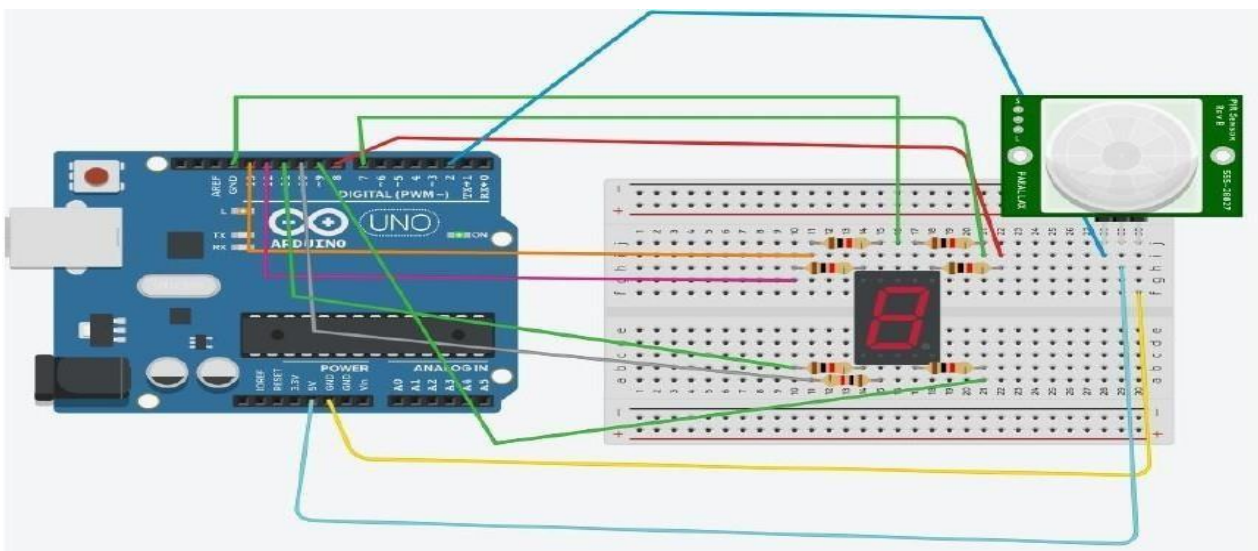
- **Arduino:**
  - Arduinoisanopen-sourceplatformusedforbuildingelectronicsprojects.
  - Arduinoconsistsofbothaphysicalprogrammable circuitboardand a piece of software, or IDE runs on your computer, used towriteandupload computercodetothephysicalboard.
  - ArduinoUNOhas14digitalpinsand6analogpins.
- **Breadboard:**
  - Itisawayofconstructingelectronicswithouthavingtouseasoldering iron.
  - Componentsarepushedintothesocketsonthebreadboardandthene xtra'jumper' wiresareusedtomakeconnections.
- **Seven-segment:**
  - Theseven-segmentdisplayhassevenLEDsarrangedintheshapeofnumber eight.
- **Resistors:**
  - It'sapassivetwo-terminalelectricalcomponentthatimplementslectricalresistanceas acircuitelement.
  - In electronic circuits, resistors are used to reduce current flow,adjustsignallevels,todividevoltages,biasactiveelements,and terminatetransmission lines,among otheruses.

- **InfraredSensor:**
  - Infrared (IR) communication is a widely used and easy to implement wireless technology that has many useful applications.
  - The most prominent examples in day-to-day life are TV/video remote controls, motion sensors, and infrared thermometers.

### Required Components:

- Arduino Uno (1x)
- Breadboard (1x)
- Seven-segment Display (1x)
- Resistors (7x)
- IR Proximity Sensor (1x)
- Jumpwires

### Circuit Diagram:



## Connections:

Components	Connections	
	From	To
<b>Arduinoto7-segmentDisplay</b>	GND	COMorGND
	13	Pinf(throughresistance)
	12	Ping(throughresistance)
	11	Pine(throughresistance)
	10	Pind(throughresistance)
	9	Pinc(throughresistance)
	8	Pinb(throughresistance)
	7	Pina(throughresistance)
<b>ArduinotoIRProximitySensor</b>	5V	V <sub>CC</sub>
	GND	GND
	2	O/P

## SourceCode:

```

int f =
13;int g =
12;int e =
11;int d =
10;int c =
9;int b =
8;inta=7;
int delay_ms =
1000;int irsensorpin =
5;int irsensorstate =
0;intp=0;

voidsetup(){
// put your setup code here, to run
once:pinMode(13,OUTPUT);
pinMode(12,
OUTPUT);pinMode(11,
OUTPUT);pinMode(10,
OUTPUT);pinMode(9,
OUTPUT);pinMode(8,OU
TPUT);

```

```

pinMode(7,
OUTPUT);pinMode(5, I
NPUT);
}

voidloop() {
    //putyourmaincodehere,torunrepeatedly:

    irsensorstate =
digitalRead(irsensorpin);if(irsensorstate
==HIGH)
{
    p++;
}

if(p==0)
{
    digitalWrite(a,1);
    digitalWrite(b,1);
    digitalWrite(c,1);
    digitalWrite(d,1);
    digitalWrite(e,1);
    digitalWrite(f,1);
    digitalWrite(g,0);

    delay(delay_ms);//0
}

if(p==1)
{
    digitalWrite(a,0);
    digitalWrite(b,1);
    digitalWrite(c,1);
    digitalWrite(d,0);
    digitalWrite(e,0);
    digitalWrite(f,0);
    digitalWrite(g,0);

    delay(delay_ms);//1
}

if(p==2)
{
    digitalWrite(a,1);
    digitalWrite(b,1);
    digitalWrite(c,0);
    digitalWrite(d,1);

```

```
digitalWrite(e,1);
```



```

        digitalWrite(f,0);
        digitalWrite(g,1);

        delay(delay_ms);//2
    }

    if(p==3)
    {
        digitalWrite(a,1);
        digitalWrite(b,1);
        digitalWrite(c,1);
        digitalWrite(d,1);
        digitalWrite(e,0);
        digitalWrite(f,0);
        digitalWrite(g,1);

        delay(delay_ms);//3
    }

    if(p==4)
    {
        digitalWrite(a,0);
        digitalWrite(b,1);
        digitalWrite(c,1);
        digitalWrite(d,0);
        digitalWrite(e,0);
        digitalWrite(f,1);
        digitalWrite(g,1);

        delay(delay_ms);//4
    }

    if(p==5)
    {
        digitalWrite(a,1);
        digitalWrite(b,0);
        digitalWrite(c,1);
        digitalWrite(d,1);
        digitalWrite(e,0);
        digitalWrite(f,1);
        digitalWrite(g,1);

        delay(delay_ms);//5
    }

    if(p==6)

```

```

{
    digitalWrite(a,0);
    digitalWrite(b,0);
    digitalWrite(c,1);
    digitalWrite(d,1);
    digitalWrite(e,1);
    digitalWrite(f,1);
    digitalWrite(g,1);

    delay(delay_ms);//6
}

if (p==7)
{
    digitalWrite(a,1);
    digitalWrite(b,1);
    digitalWrite(c,1);
    digitalWrite(d,0);
    digitalWrite(e,0);
    digitalWrite(f,0);
    digitalWrite(g,0);

    delay(delay_ms);//7
}

if (p==8)
{
    digitalWrite(a,1);
    digitalWrite(b,1);
    digitalWrite(c,1);
    digitalWrite(d,1);
    digitalWrite(e,1);
    digitalWrite(f,1);
    digitalWrite(g,1);

    delay(delay_ms);//8
}

if (p==9)
{
    digitalWrite(a,1);
    digitalWrite(b,1);
    digitalWrite(c,1);
    digitalWrite(d,0);
    digitalWrite(e,0);
    digitalWrite(f,1);

```

```
        digitalWrite(g,1);  
        delay(delay_ms);//9  
    }  
}
```

# Practical4

**Aim:** Print message on LCD display with Arduino UNO.

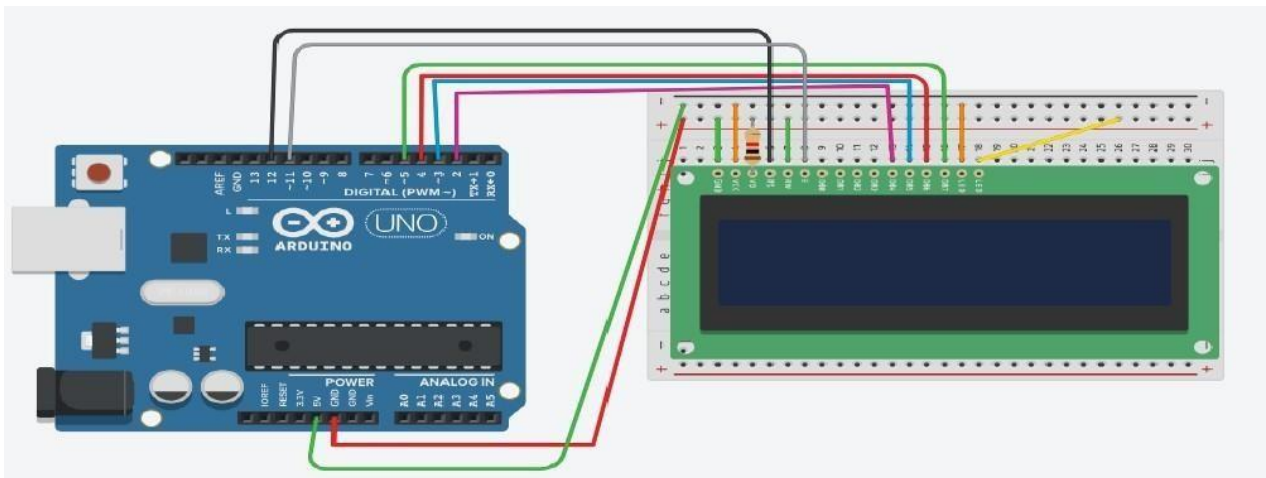
## **Description:**

- **Arduino:**
  - Arduino is an open-source platform used for building electronics projects.
  - Arduino consists of both a physical programmable circuit board and a piece of software, or IDE runs on your computer, used to write and upload computer code to the physical board.
  - Arduino UNO has 14 digital pins and 6 analog pins.
- **Breadboard:**
  - It is a way of constructing electronics without having to use a soldering iron.
  - Components are pushed into the sockets on the breadboard and then extra 'jumper' wires are used to make connections.
- **LCD:**
  - A liquid-crystal display (LCD) is a flat-panel display or other electronically modulated optical device that uses the light-modulating properties of liquid crystals combined with polarizers.
  - Liquid crystals do not emit light directly, instead using a backlight or reflector to produce images in colour or monochrome.
  - It is a 16x2 LCD display. That is, it has 16 columns and 2 rows.

### Required Components:

- ArduinoUno (1x)
- Breadboard(1x)
- LCD16x2(1x)
- Resistors(?x)
- JumpWires

### CircuitDiagram:



## Connections:

Components	Connections	
	From	To
ArduinotoLCD display	12	RS
	11	Enable
	5	D4
	4	D5
	3	D6
	2	D7
	GND	R/W
	GND	V <sub>ss</sub>
	5V	V <sub>cc</sub>

## SourceCode:

**Note:** Install the *LiquidCrystal* library by navigating to Tools > Manage Libraries or by using the shortcut Ctrl+Shift+I.

```
#include<LiquidCrystal.h>
const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 =
2;LiquidCrystallcd(rs,en,d4,d5,d6,d7);

voidsetup(){
    // put your setup code here, to run
    once:lcd.begin(16,2);
    lcd.print("Hello,world!");
}

voidloop(){
    // put your main code here, to run
    repeatedly:lcd.setCursor(13,0);
    lcd.print("Ok");lcd.s
    etCursor(5,1);
```

```
for(intthisChar=0;thisChar<10;thisChar++)  
{  
    lcd.print(thisChar);  
    delay(500);  
}  
}
```

# Practical5

**Aim:** Use 4x4 keypad to give the input in Arduino UNO serial monitor.

## Description:

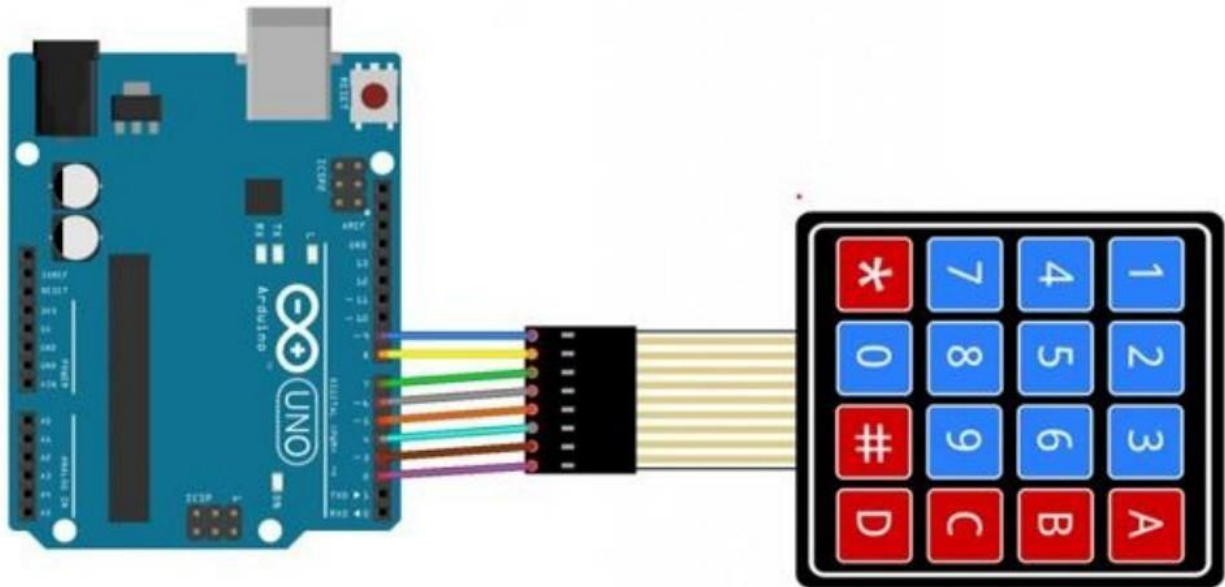
- **Arduino:**
  - Arduino is an open-source platform used for building electronics projects.
  - Arduino consists of both a physical programmable circuit board and a piece of software, or IDE runs on your computer, used to write and upload computer code to the physical board.
  - Arduino UNO has 14 digital pins and 6 analog pins.
- **Keypad:**
  - The buttons on a keypad are arranged in rows and columns. A 3x4 keypad has 4 rows and 3 columns, and a 4x4 keypad has 4 rows and 4 columns.
  - Keypad 4x4 is used for loading numerics into the microcontroller.
  - It consists of 16 buttons arranged in a form of an array containing four lines and four columns.
  - It is connected to the development system by regular IDC 10 female connector plugged into some development system's port.

## Required Components:

- Arduino Uno (1x)
- 4x4 Keypad (1x)
- Jump Wires



## CircuitDiagram:



## Connections:

Components	Connections	
	From	To
Arduinoto4x4K eypad	9	R1
	8	R2
	7	R3
	6	R4
	5	C1
	4	C2
	3	C3
	2	C4

## SourceCode:

**Note:** Install the Keypad and KeyMatrix library by navigating to Tools > Manage Libraries or by using the shortcut Ctrl+Shift+I.

```
#include<Keypad.h>#i
nclude<Key.h>const
byte rows =
4;constbytecols=4;
charkeys[rows][cols]={
    {'1','2','3','A'},
    {'4','5','6','B'},
    {'7','8','9','C'},
    {'*','0','#','D'}
};

byte colPins[rows] = {5, 4, 3, 2}; //Connect to the row pinouts
ofkeypad.
byte rowPins[cols] = {9, 8, 7, 6}; //Connect to the row pinouts
ofkeypad.
Keypad keypad = Keypad(makeKeymap(keys), rowPins, colPins,
rows,cols);

voidsetup(){
    // put your setup code here, to run
    once:Serial.begin(9600);
}

voidloop(){
    // put your main code here, to run
    repeatedly:charkey=keypad.getKey();
    if(key)
    {
        Serial.println(key);
    }
}
```

# Practical6

**Aim:**InterfacingofbuzzerwithArduinoUNO.

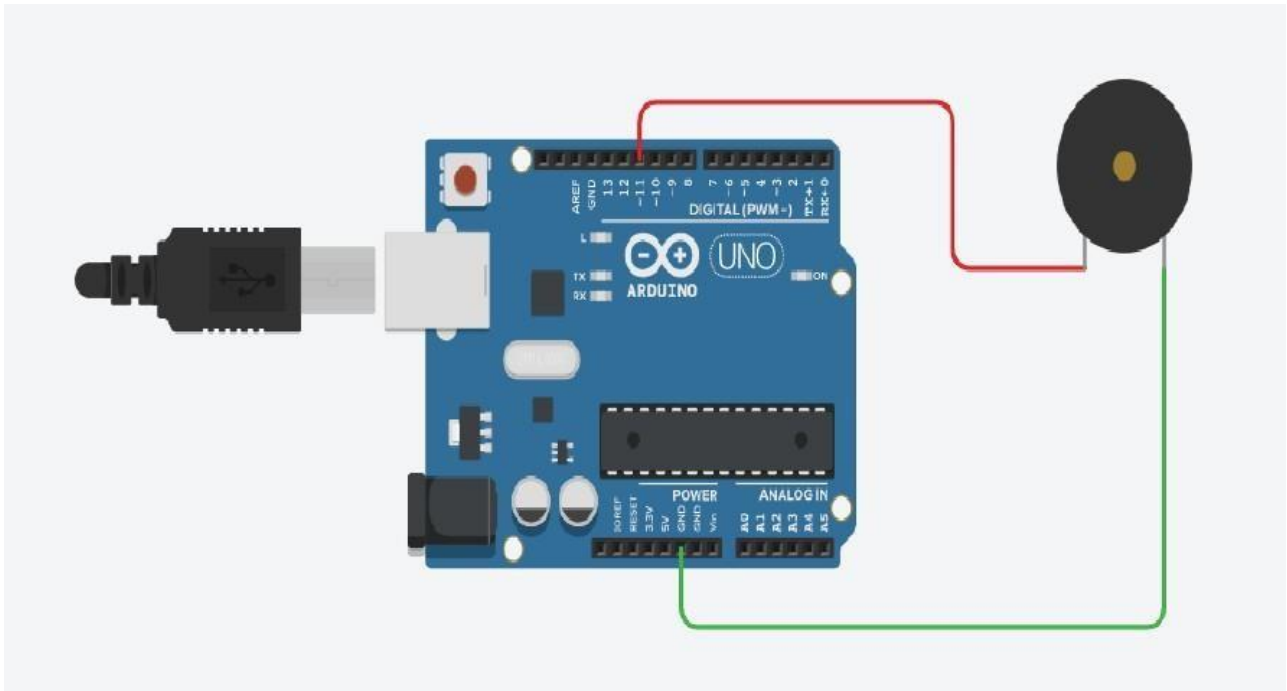
## **Description:**

- **Arduino:**
  - Arduinoisanopen-sourceplatformusedforbuildingelectronicsprojects.
  - Arduinoconsistsofbothaphysicalprogrammable circuitboardanda piece of software, or IDE runs on your computer, used towriteandupload computercodetothephysicalboard.
  - ArduinoUNOhas14digitalpinsand6analogpins.
- **Buzzer:**
  - A buzzer or beeper is an audio signalling device, which may bemechanical,electromechanical,orpiezoelectric(piezoforshort).
  - Typical uses of buzzers and beepers include alarm devices,timers,andconfirmationofuserinputsuchasamouse clickorkeystroke.

## **RequiredComponents:**

- ArduinoUno (1x)
- Buzzer(1x)
- JumpWires

## CircuitDiagram:



## Connections:

Components	Connections	
	From	To
ArduinotoBuzzer	11	+vepin
	GND	-vepin

## SourceCode:

1)

```
intbuzzer=11;

voidsetup() {
    //putyoursetupcodehere,torunonce:

}

voidloop() {
    // put your main code here, to run
    repeatedly:tone(buzzer,450);
    delay(500);noTo
    ne(buzzer);dela
    y(500);
}
```

2)

```
intbuzzer=11;

voidsetup() {
    //putyoursetupcodehere,torunonce:

}

voidloop() {
    // put your main code here, to run
    repeatedly:inti =0;
    do
    {
        i++;
        tone(buzzer,
        450);delay(200);no
        Tone(buzzer);delay
        (200);
    }
    while(i <
    3);delay(3000
    );
}
```

# Practical7

**Aim:** Interfacing of ultrasonic sensor with Arduino UNO.

## **Description:**

- **Arduino:**

- Arduino is an open-source platform used for building electronics projects.
- Arduino consists of both a physical programmable circuit board and a piece of software, or IDE runs on your computer, used to write and upload computer code to the physical board.
- Arduino UNO has 14 digital pins and 6 analog pins.

- **Breadboard:**

- It is a way of constructing electronics without having to use a soldering iron.
- Components are pushed into the sockets on the breadboard and then extra 'jumper' wires are used to make connections.

- **LED:**

- A light-emitting diode (LED) is a semiconductor device that produces light from electricity. LEDs last a long time and do not break easily.

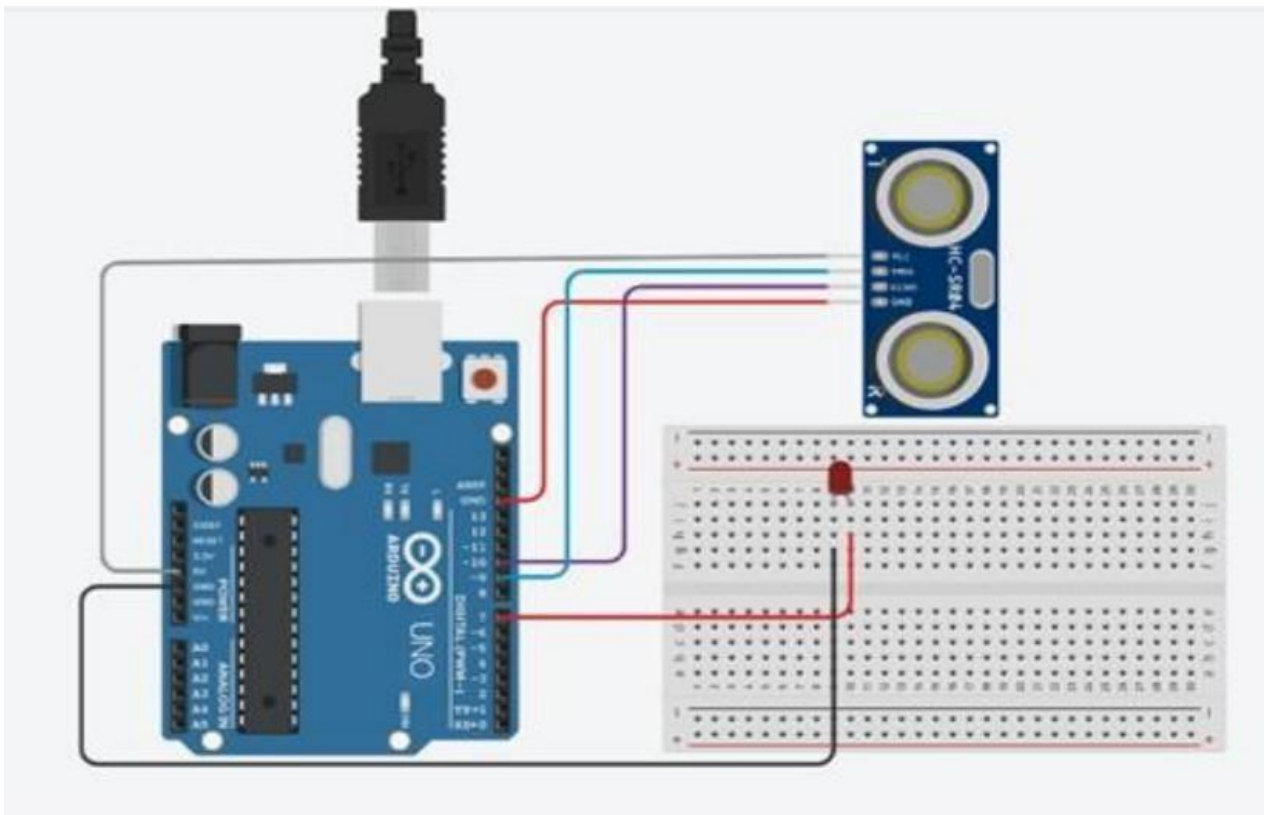
- **Ultrasonic Sensor:**

- Ultrasonic distance sensor determines the distance to an object by measuring the time taken by the sound to reflect back from that object.
- A typical ultrasonic distance sensor consists of two membranes. One membrane produces sound, another catches reflected echo. Basically they are a speaker and microphone.

## Required Components:

- ArduinoUno (1x)
- Breadboard(1x)
- LED(1x)
- UltrasonicSensor(1x)
- JumpWires

## CircuitDiagram:



**Connections:**

<b>Components</b>	<b>Connections</b>	
	<b>From</b>	<b>To</b>
<b>ArduinotoUltrasonic Sensor</b>	5V	VCC
	GND	GND
	9	Trig
	10	Echo
<b>ArduinotoLED</b>	7	+veend
	GND	-veend



## SourceCode:

```
int trigpin =
9;int echopin =
10;intled=7;

voidsetup(){
  // put your setup code here, to run
  once:Serial.begin(9600);
  pinMode(led,
  OUTPUT);pinMode(trigpin,
  OUTPUT);pinMode(echopin,I
  NPUT);
}

voidloop(){
  // put your main code here, to run
  repeatedly:longduration,distance;
  digitalWrite(trigpin,
  HIGH);delayMicroseconds(1000);digi
  talWrite(trigpin, LOW);duration =
  pulseIn(echopin, HIGH);distance =
  (duration / 2) /
  29.1;Serial.print(distance);Serial
  .println("CM");
  delay(10);if(dista
  nce<=10)
  {
    digitalWrite(led,HIGH);
  }
  elseif(distance>10)
  {
    digitalWrite(led,LOW);
  }
}
```

# Practical8

**Aim:**InterfacingofservomotorwithArduinoUNO.

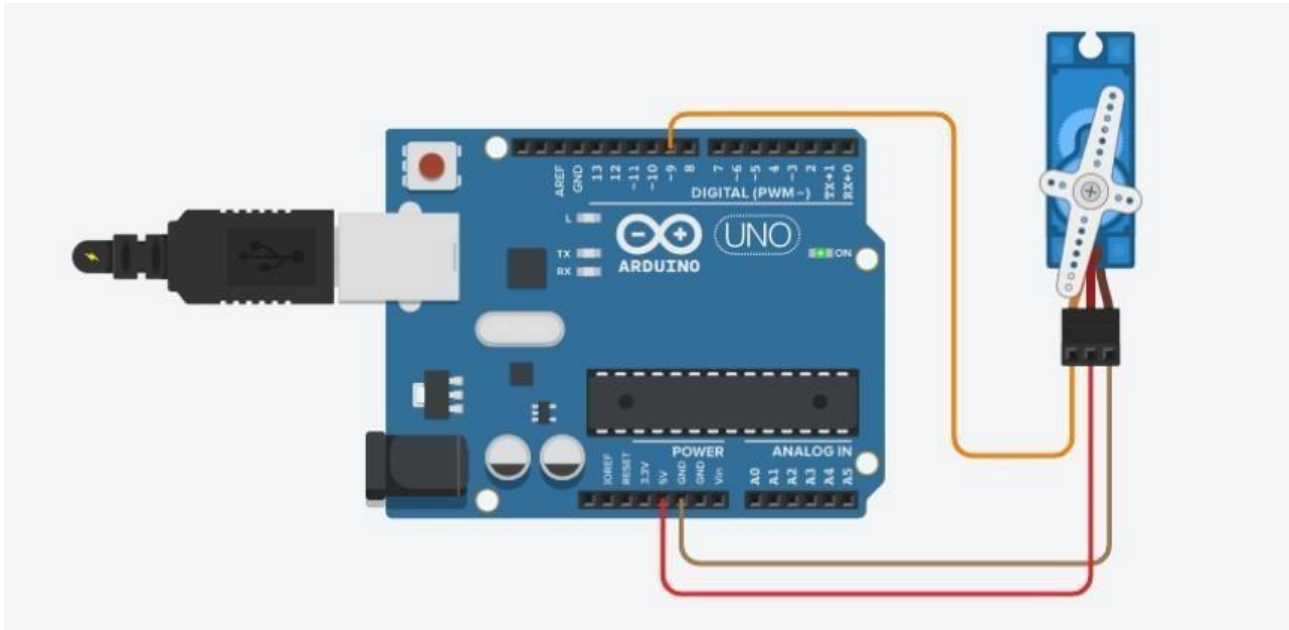
## **Description:**

- **Arduino:**
  - Arduinoisanopen-sourceplatformusedforbuildingelectronicsprojects.
  - Arduinoconsistsofbothaphysicalprogrammable circuitboardanda piece of software, or IDE runs on your computer, used towriteandupload computercodetothephysicalboard.
  - ArduinoUNOhas14digitalpinsand6analogpins.
- **ServoMotor:**
  - Aservomotorisanelectricaldevicewhichcanpushorrotateanobjectwithgreatprecision.
  - Ifyouwanttorotateanobjectatsomespecificanglesordistance,then youuseservo motor.

## **RequiredComponents:**

- ArduinoUno (1x)
- ServoMotor(1x)
- JumpWires

## CircuitDiagram:



## Connections:

Components	Connections	
	From	To
ArduinotoServo Motor	9	Orangepin
	5V	Redpin
	GND	Brownpin

## SourceCode:

Note:InstalltheServolibrarybynavigatingtoTools>ManageLibrariesorby usingtheshortcut Ctrl+Shift+I.

```
#include<Servo.h>
//Create Servo object to control a
ServoServomy servo;
// 12 servo objects can be created on most
boards.int pos;
void setup() {
    // put your setup code here, to run
    once:my servo.attach(9); //Variable to store the servo position.
}

void loop() {
    // put your main code here, to run
    repeatedly:for (pos=0; pos<=180; pos+=1)
    {
        // Goes from 0 to 180 degrees with a step of
        1my servo.write(pos); //Tell servo to go to position
        'pos'.delay(15); //Wait 15ms for the servo to reach the position
        'pos'.
    }

    for (pos=180; pos>=0; pos-=1)
    {
        // Goes from 180 to 0 degrees with a step of -
        1my servo.write(pos); //Tell servo to go to position
        'pos'.delay(15); //Wait 15ms for the servo to reach the position
        'pos'.
    }
}
```

# Practical9

**Aim:** Interfacing of DHT11 with Arduino UNO to read temperature and humidity which is then printed on serial monitor.

## Description:

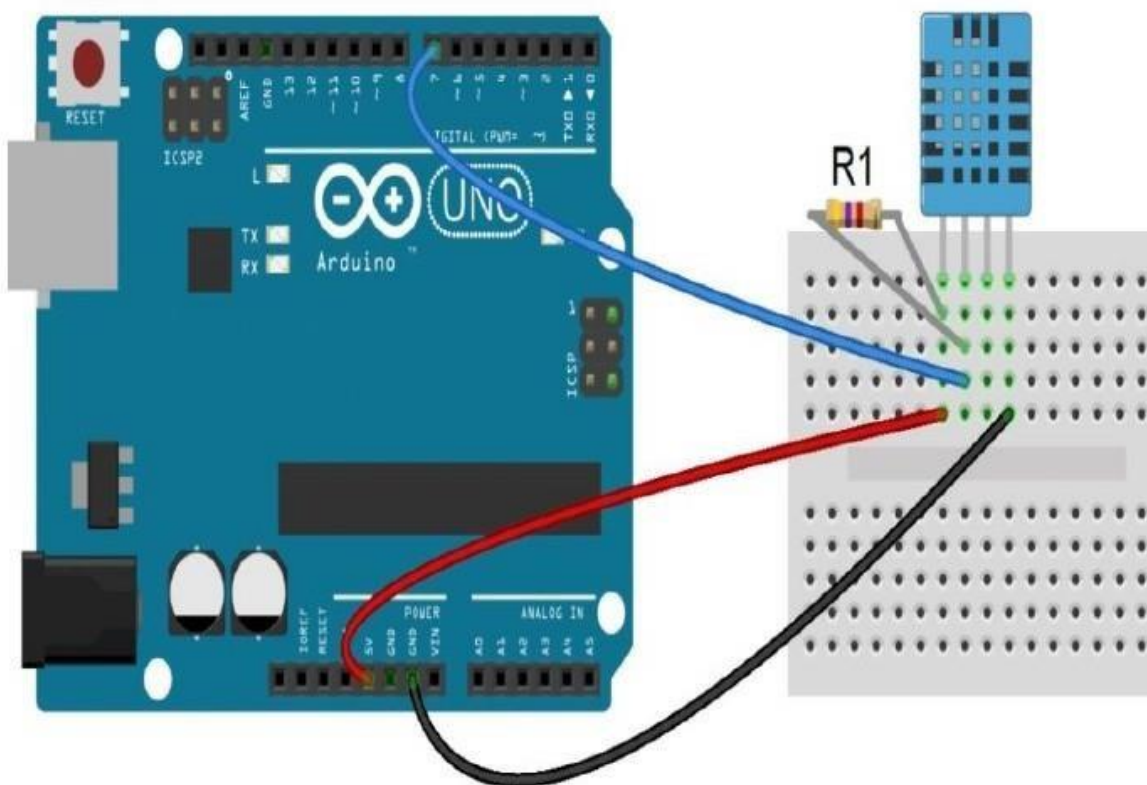
- **Arduino:**
  - Arduino is an open-source platform used for building electronics projects.
  - Arduino consists of both a physical programmable circuit board and a piece of software, or IDE runs on your computer, used to write and upload computer code to the physical board.
  - Arduino UNO has 14 digital pins and 6 analog pins.
- **Breadboard:**
  - It is a way of constructing electronics without having to use a soldering iron.
  - Components are pushed into the sockets on the breadboard and then extra 'jumper' wires are used to make connections.
- **Resistors:**
  - It's a passive two-terminal electrical component that implements electrical resistance as a circuit element.
  - In electronic circuits, resistors are used to reduce current flow, adjust signal levels, to divide voltages, bias active elements, and terminate transmission lines, among other uses.
- **DHT11:**
  - DHT11 is a low-cost digital sensor for sensing temperature and humidity.
  - This can be easily interfaced with any microcontroller like Arduino, Raspberry Pi etc to measure humidity and temperature instantaneously.

- This sensor is used various applications such as measuring humidity and temperature values in heating, ventilation and AC systems.
- Offices, cars, greenhouses use this sensor for measuring humidity value and safety measure. This can be used for smart gardening.

### Required Components:

- Arduino Uno (1x)
- Breadboard (1x)
- Resistor (1x)
- DHT11 Sensor (1x)
- Jump Wires

### Circuit Diagram:



## Connections:

Components	Connections	
	From	To
ArduinotoDHT11	5V	V <sub>CC</sub>
	GND	GND
	7	SDA
ResistortoDHT11	V <sub>CC</sub>	SDA

## SourceCode:

**Note:** Install the DHTLib library by navigating to Tools > Manage Libraries or by using the shortcut Ctrl+Shift+I.

```
#include<dht.h>#def
ine DHT11_PIN
7dhtDHT;

voidsetup(){
  // put your setup code here, to run
  once:Serial.begin(9600);
}

voidloop(){
  // put your main code here, to run
  repeatedly:int chk =
  DHT.read11(DHT11_PIN);Serial.print("Temperatur
e =
");Serial.println(DGT.temperature);Serial.prin
t("Humidity = ");Serial.println(DHT.humidity);
  delay(1000);
}
```

# Practical10

**Aim:** Interfacing of LED with NodeMCU and controlling it remotely with Blynk application on mobile.

## Description:

- **NodeMCU:**
  - NodeMCU is low-cost, open source IOT platform.
  - It initially included firmware which runs on the ESP8266 Wi-Fi SoC.
  - Arduino UNO does not have an inbuilt Wi-Fi module. It provides access to the GPIO.
  - It has 10 digital pins and only 1 analog pin.
  - It can also be programmed directly using Arduino IDE.
  - It consumes ten times of power than Arduino UNO.
- **Blynk App:**
  - Blynk is a platform with iOS and Android apps to control Arduino, Raspberry Pi and soon.
  - It's a digital dashboard where we can build a graphic interface for our project by simply dragging and dropping widgets.
- **Breadboard:**
  - It is a way of constructing electronics without having to use a soldering iron.
  - Components are pushed into the sockets on the breadboard and then extra 'jumper' wires are used to make connections.
- **LED:**
  - A light-emitting diode (LED) is a semiconductor device that produces light from electricity. LEDs last a long time and do not break easily.



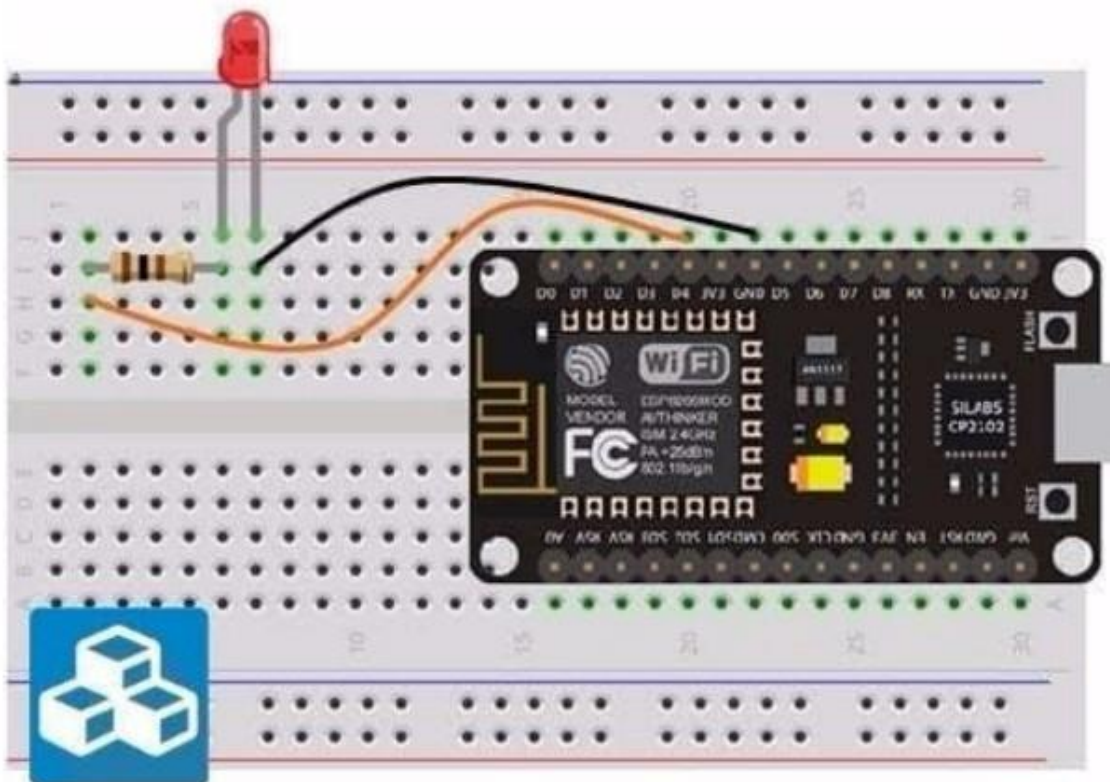
- **Resistors:**

- It's a passive two-terminal electrical component that implements electrical resistance in a circuit element.
- In electronic circuits, resistors are used to reduce current flow, adjust signal levels, to divide voltages, bias active elements, and terminate transmission lines, among other uses.

### **Required Components:**

- NodeMCU(1x)
- Breadboard(1x)
- LED(1x)
- Resistor(1x)
- Jump Wires

### **Circuit Diagram:**



Connections:

Components	Connections	
	From	To
NodeMCUtoLED	D5	+vepin
	GND	-vepin

Blynk IotAppConfiguration:

TocontrolLEDconnectedtoNodeMCUremotelyweneedtoinstallBlynkIoTappinourmo  
bileasfollows–

- 1. DownloadBlynkIoTappfromplaystoreorappstore.

3:08

78%



Sign Up

Log In

\_\_\_\_\_

## 2. After downloading, create an account by clicking on Sign Up.

3:09  78%

← Sign Up

Simply fill in your email address and we will  
send an account activation link.

EMAIL

 your.email@email.com 

Enter email

☐ I agree to [Terms and Conditions](#) and  
accept [Privacy Policy](#)

Continue

3. Enter your Email ID and then you will receive a link in your inbox where you need to enter a password for your account.

4G+ 3:20 97% LTE

← Set a password

Create a password which is hard to guess.

PASSWORD

password

Enter new password

VERY WEAK

- Make it longer than 8 symbols
- Use uncommon words
- Use non-standard uPPerCaSing,
- Use creatif spelllllllling
- Use non-obvi0u\$ number\$ & symbo1s

Continue

4. After this, you need to enter your first name to finish the setup of your account.

4G+ 3:21 96% LTE

← User Profile

1 of 1

Fill in profile information

FIRST NAME

Enter...

Enter first name

Next

4G+ 3:21 96% LTE

Blynk

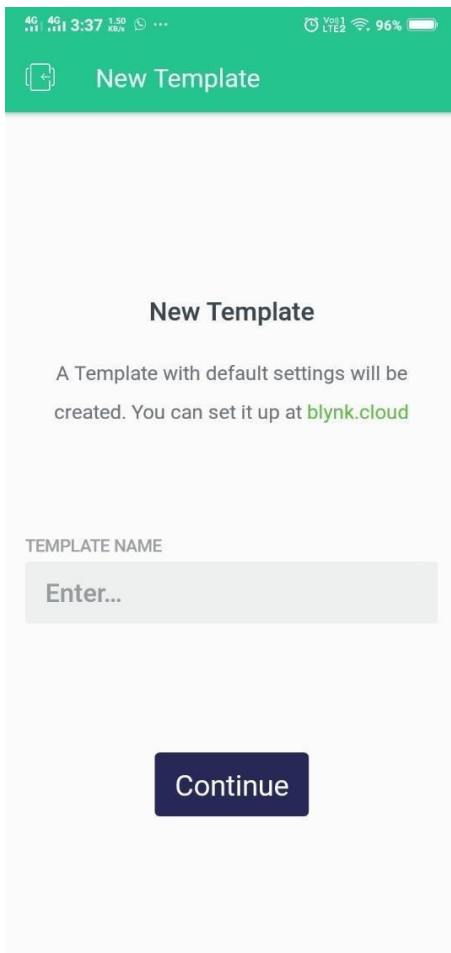
No devices yet

You don't have any devices connected yet

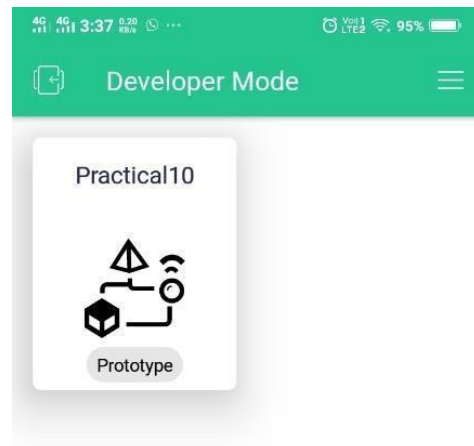
Add new device

Developer Mode

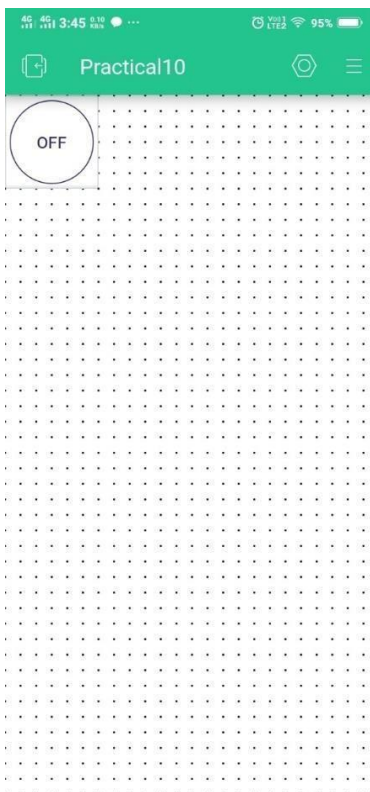
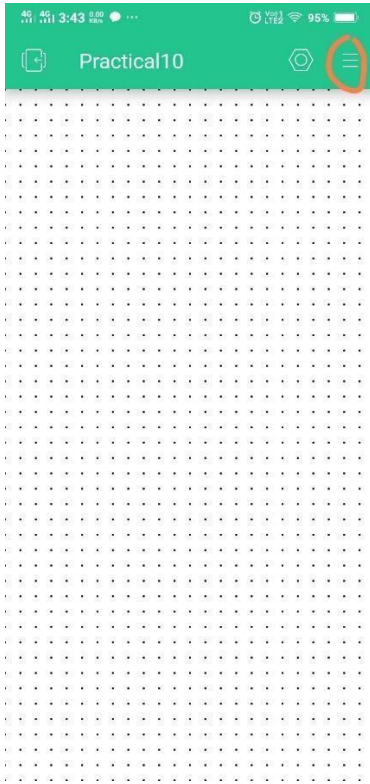
5. Click on Developer mode. Then you will be asked to enter a name for the template. Give a suitable template name and then click on continue.



The screenshot shows the 'New Template' screen in the Blynk app. The status bar at the top indicates 4G LTE, 3:37, and 96% battery. The app header is green with a back arrow and the text 'New Template'. The main content area has a title 'New Template' and a paragraph: 'A Template with default settings will be created. You can set it up at [blynk.cloud](https://blynk.cloud)'. Below this is a text input field labeled 'TEMPLATE NAME' with the placeholder 'Enter...'. At the bottom is a dark blue button labeled 'Continue'.



6. Now click on Prototype of Practical 10 a canvas will get opened. Then click on the circled part shown on the image and after that under controllers click on Button. A button with OFF written on it will be created on the canvas.



7. Now click on the button created it will open Button Settings. Enter title for the button and select the mode of the button to **SWITCH**.

LED

TITLE ALIGNMENT

DATASTREAM

Select Data Stream

MODE

PUSH SWITCH

ON/OFF LABELS

OFF ON

OFF ON

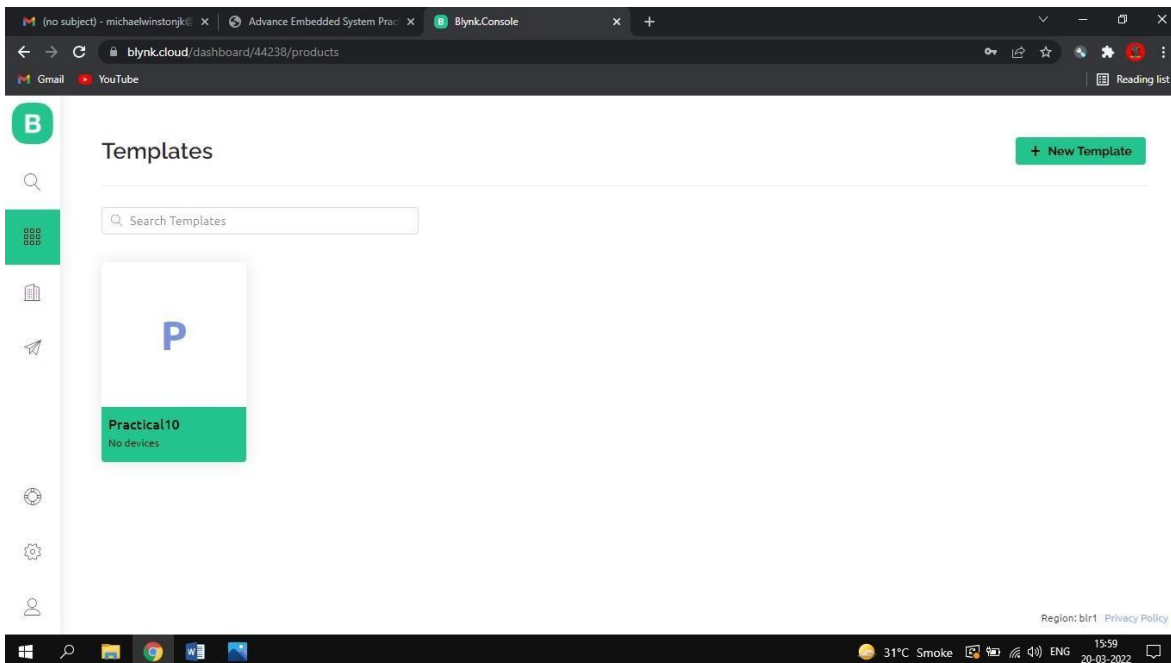
FONT SIZE

Medium

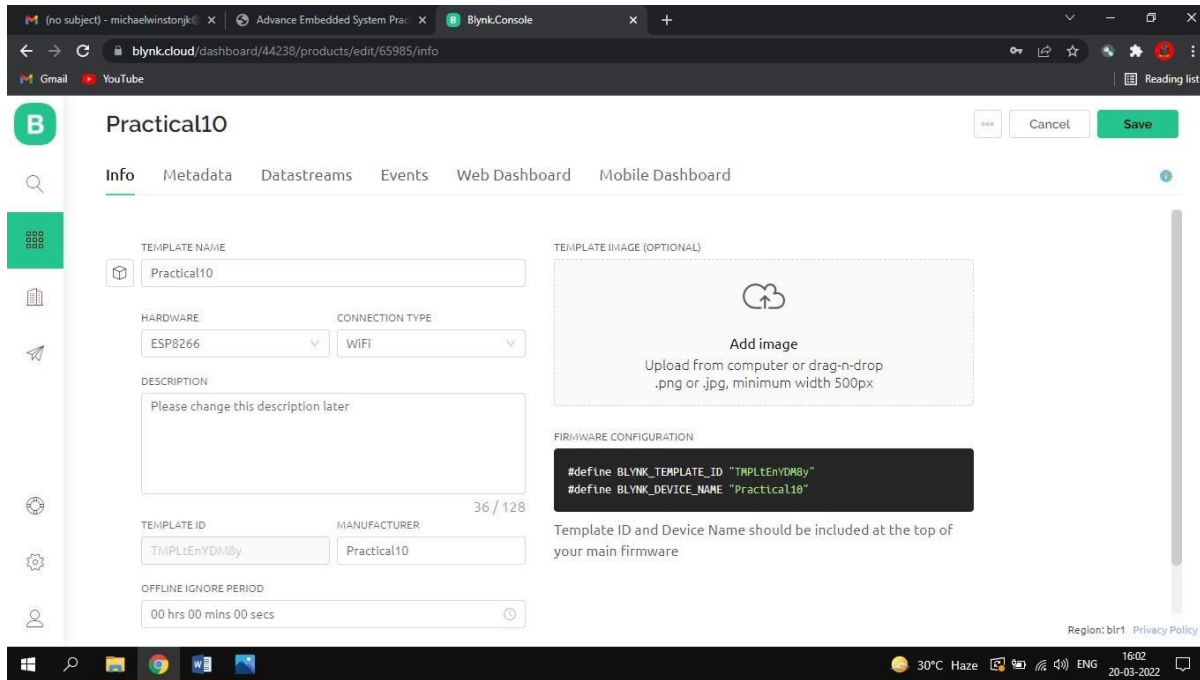
DESIGN

TEXT

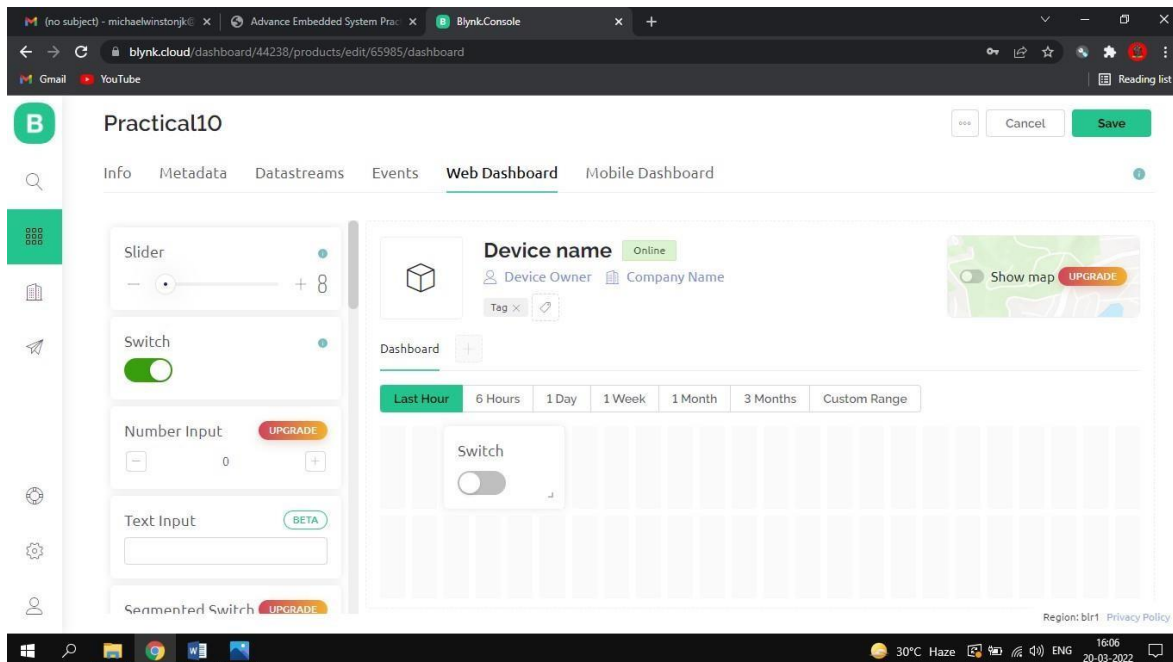
8. Now, Login to the desktop site of blynk i.e blynkcloud and then go to Templates section there you can see the template which we created on our mobile is visible click on it.



9. Now click on edit and change the **HARDWARE** to **esp8266** by selecting **esp8266** from the list and then click on save.

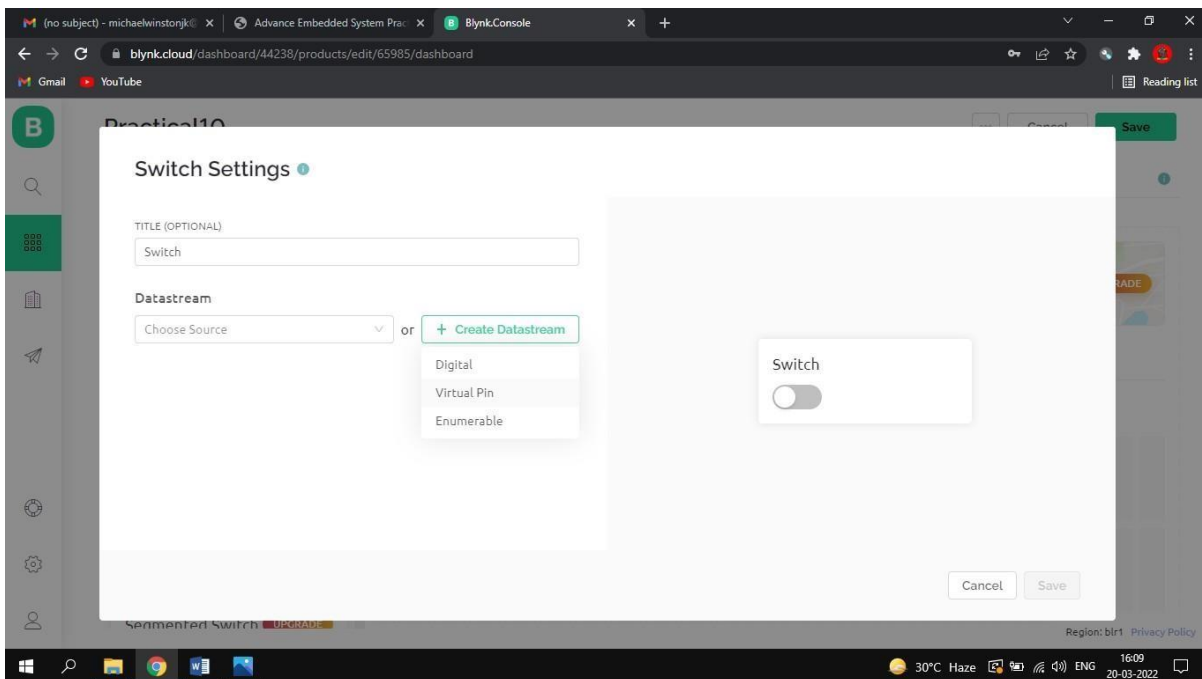


10. Now head towards the **Web Dashboard** of four template and then click on edit. After that drag and drop **Switch** widget from the widgets present on left of the page to the canvas.

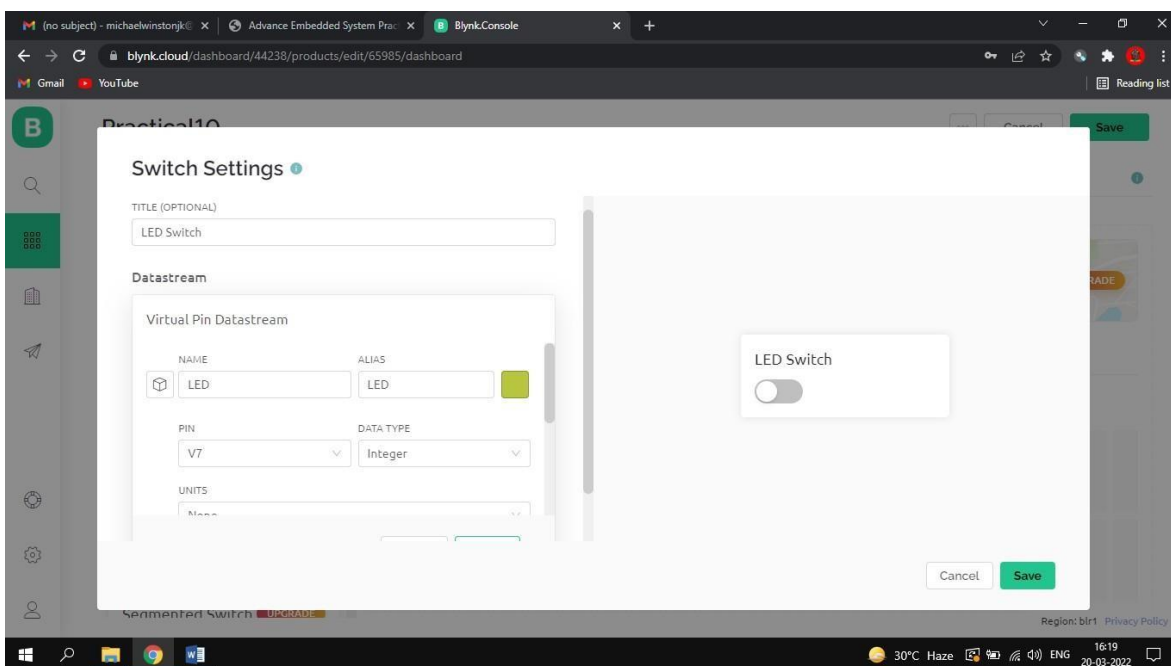




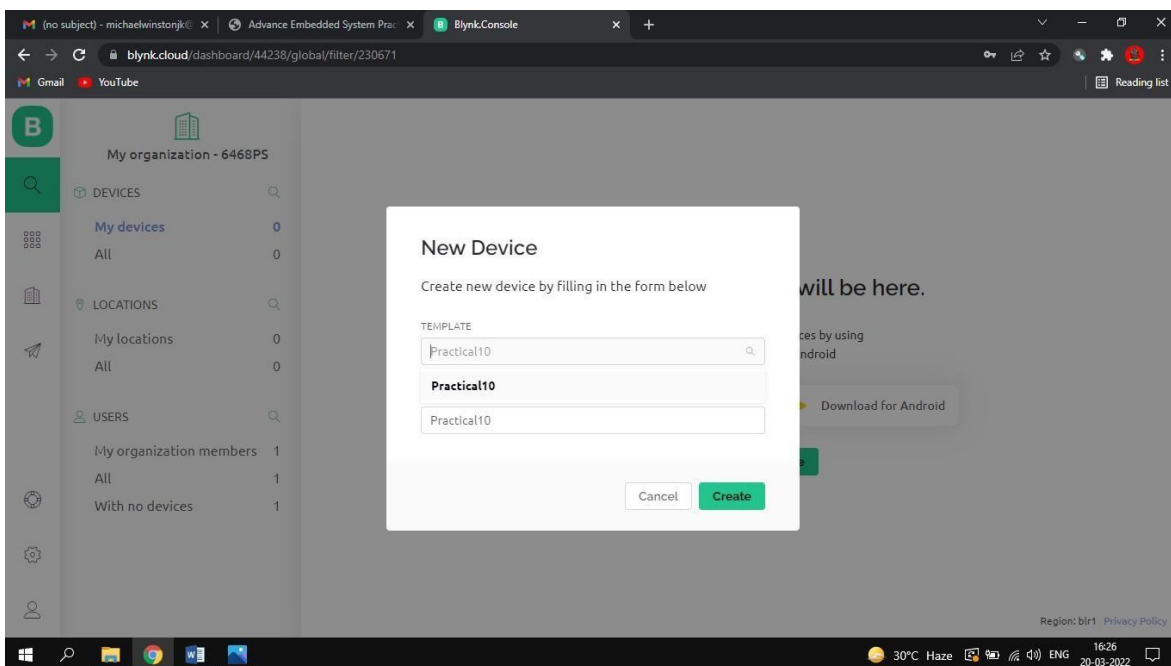
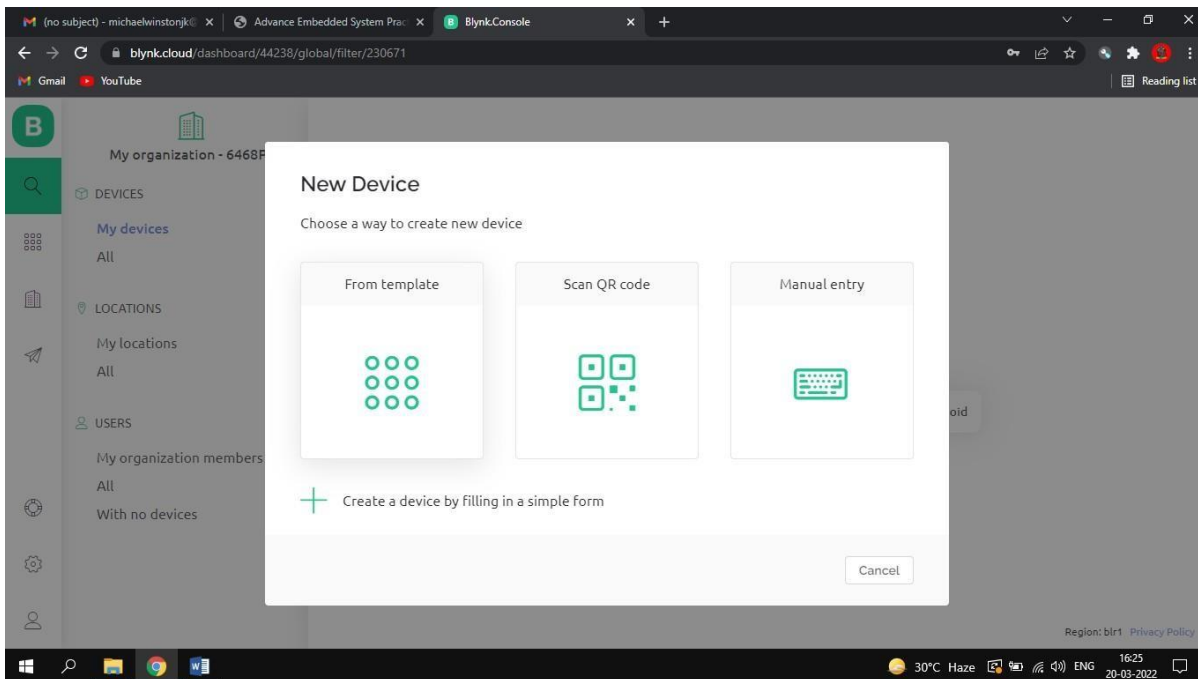
11. Now click on settings button of the switch widget it will open Switch Settings in that click on **Create Datastream** select **Virtual Pin**.



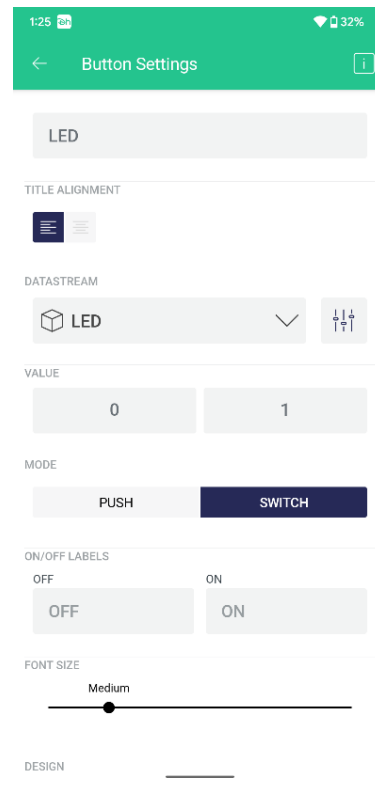
12. Now enter Title and name in Virtual Pin Destination and then click on create and after that click on save.



13. Now click on search on top left side and then click on **My devices**. Then click on **New Device** and click on **From Template** after that select template to **Practical10** or any suitable name given by you and then click on **Create**.



14. After that go to Blynk IoT App and click on the LED Button created on the template. Then click on Select Data Stream. A list of Datastream appears in that select 'LED'.



With this Blynk IoT app configuration is done.

## Uploading Firmware:

1. Download following zip file:-

[https://github.com/blynkkk/blynk-library/releases/download/v0.6.1/Blynk\\_Release\\_v0.6.1.zip](https://github.com/blynkkk/blynk-library/releases/download/v0.6.1/Blynk_Release_v0.6.1.zip)

2. Extract this file → we can see 2 folders **tools** and **libraries**
3. The content of tools will be copied and pasted on C:\Program Files(x86)\Arduino\tools
4. The content of libraries will be copied and pasted on C:\Program Files(x86)\Arduino\libraries

# ArduinoIDESetup:

OpenarduinoIDEandperformfollowingconfiguration–

1. File → Preferences → In **Additional Boards Manager** text box  
enter:[https://arduino.esp8266.com/stable/package\\_esp8266com\\_index.js](https://arduino.esp8266.com/stable/package_esp8266com_index.js)  
[on](#)→OK
2. Tools→Boards→BoardManager→Searchfor**esp8266byESP8266community2.6.3**→Install
3. Tool→Board→SelectNodeMCU
4. Tools→SelectCOMportforcommunication

**Code:**

```
#define
BLYNK
_PRINT
Serial#i
nclude
<ESP8266
WiFi.h>
#include<BlynkSimpleEsp8266_SSL.h>
//Youshouldget AuthTokenintheBlynkApp.
//GototheProjectSettings (nuticon)→AuthTokens
→Copy allchar auth[]="YourAuthToken";
//YourWiFicredentials.
//Setpasswordto""foropennetworks.
charssid[]="YourNetworkName"; //Wi-FiName charpass[]="YourPassword";

//

Wi-FiPasswordvoidsetup()
{
//
Debugcons
oleSerial.be
gin(9600);
Blynk.begin
(auth,
ssid,pass);
}
voidloop()
{
Blynk.run();
}
```

**Note:** - Before uploading, make sure to paste your authorization token into the auth [] variable. Also make sure to load your Wifi network settings into the Blynk.begin(auth, "ssid", "pass") function.

Now compile and run the code.

### **Output:**

Click the button from Blynk IoT app to switch ON and OFF the LED.  
can test from remotely operating.

We