

EXPERIMENT NO. 9 (Group B)

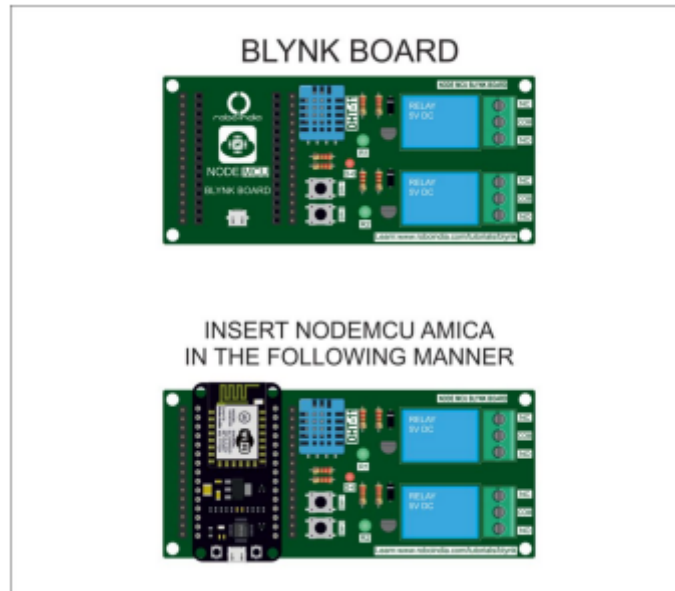
- **Aim:** Write a program to show the temperature and shows a graph of the recent measurements.
- **Outcome:** Understanding working principle of DHT11 temperature sensor, Blynk IOT Platform
- **Hardware Requirement:** Arduino(Node MCU), LM35, DHT11, etc
- **Software Requirement:** Arduino IDE
- **Theory:**

1. Introduction:

In this project using an esp8266, to show the temperature and humidity DHT11 sensor on your Smartphone or tablet. The NodeMCU collects the temperature and humidity from the DHT11 sensor and sends it to the Blynk app every second.

1.2 Hardware required

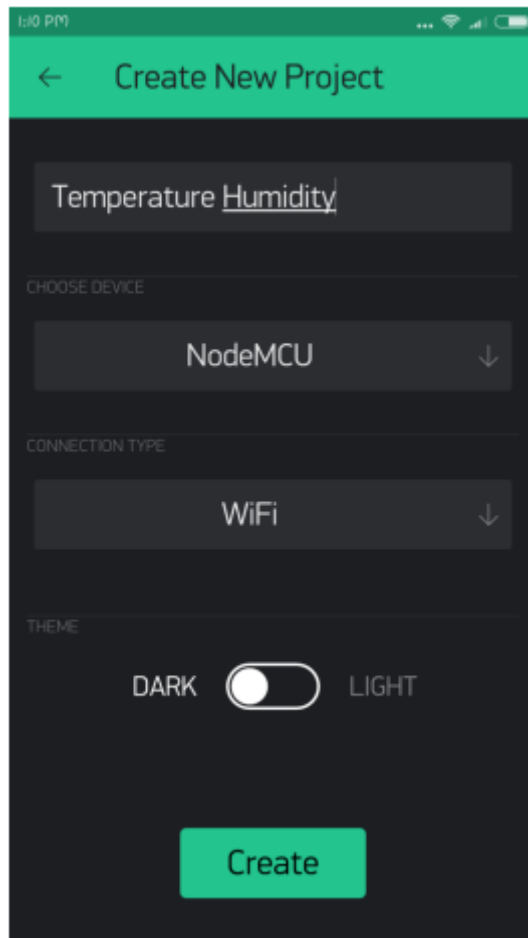
Blynk Board and NodeMCU is used in this example. Inset NodeMCU to the Blynk board as shown in the image ahead then connect NodeMCU to PC or Laptop through USB cable.



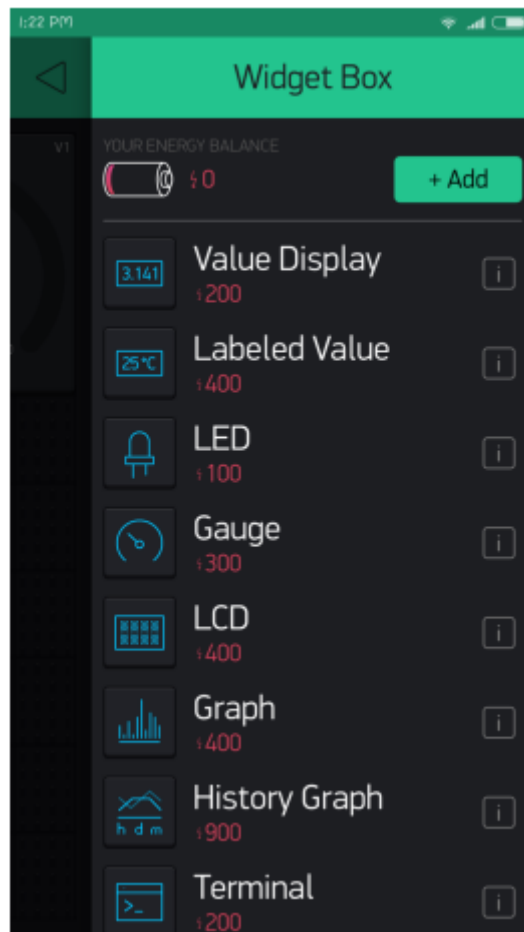
2. On Blynk App

You need to perform following steps on Blynk App.

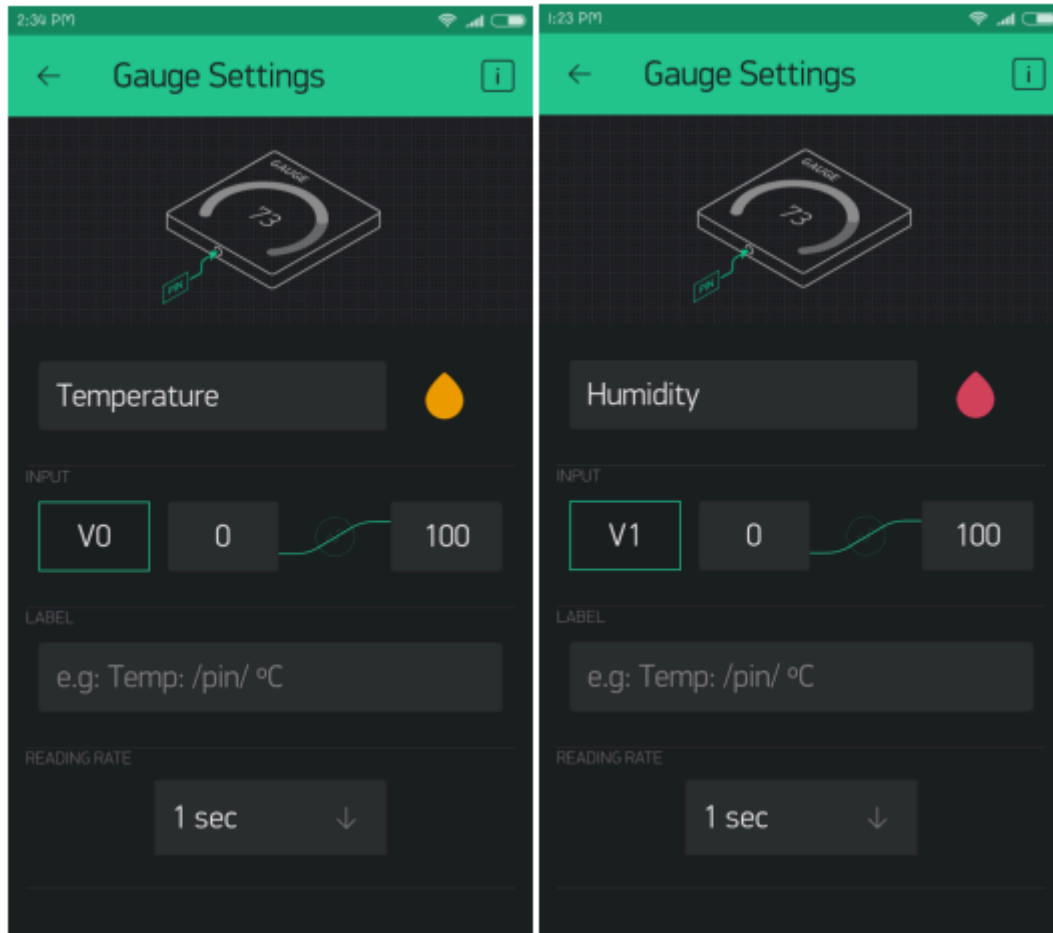
2.1 Create a New Project in BLYNK app. Write Project name Temperature Humidity and Select NodeMCU from drop down.



2.2 An AUTH token will be sent to your registered email, note this down. Tap on the screen and add a 2 Gauges.



2.3 Tap on the Widget and select the respective Virtual pins for temperature and humidity data (V0 for temperature and V1 for humidity).



Note: Make sure to setup Reading rate as '1' second for all Widgets. And add gauges for both Humidity and Temperature.

3. Code the NodeMCU with the following code.

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.

Following code may be downloaded from [here](#).

```
// Robo India Tutorial
// Digital Output on LED
// Hardware: NodeMCU Blynk Board

#define BLYNK_PRINT Serial

#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include "DHT.h" // including the library of DHT11 temperature and humidity sensor
#include <SimpleTimer.h> //including the library of Simple Timer

#define DHTTYPE DHT11 // DHT 11
```

```

#define dht_dpin 14
DHT dht(dht_dpin, DHTTYPE);
SimpleTimer timer;
char auth[] = "Your Auth. Key"; // You should get Auth Token in the Blynk App.

(nut icon). // Go to the Project Settings

char ssid[] = "Your Wifi Network name"; // Your WiFi credentials.
char pass[] = "Password of your network"; // Set password to "" for open networks.
float t; // Declare the variables
float h;

void setup()
{
  Serial.begin(9600); // Debug console
  Blynk.begin(auth, ssid, pass);
  dht.begin();
  timer.setInterval(2000, sendUptime);
}

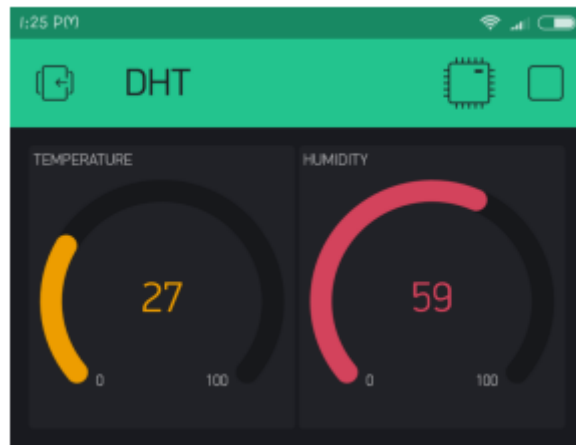
void sendUptime()
{
  float h = dht.readHumidity();
  float t = dht.readTemperature();
  Serial.println("Humidity and temperature\n\n");
  Serial.print("Current humidity = ");
  Serial.print(h);
  Serial.print("% ");
  Serial.print("temperature = ");
  Serial.print(t);
  Blynk.virtualWrite(V0, t);
  Blynk.virtualWrite(V1, h);
}

void loop()
{
  Blynk.run();
  timer.run();
}

```

4. Output

After Uploading the Arduino code IDE. Press the play button on blynk app to show the output.



Conclusion: -

Experiment no.9

```
#include "DHT.h"

#define DHTPIN 2

#define DHTTYPE DHT11

#define switchPin 3

bool lastButton=LOW;

bool currentButton=LOW;

bool tempORhum = LOW; //temp = LOW or Hum HIGH to be displayed on the Serial Plotter

DHT dht(DHTPIN, DHTTYPE);

void setup() {

    pinMode(switchPin, INPUT);

    Serial.begin(9600);

    dht.begin();

}

void loop() {

    float h = dht.readHumidity();

    float t = dht.readTemperature();

    if (isnan(t) || isnan(h)) {

        return;

    }

    currentButton = debounce(lastButton);

    if (lastButton == LOW && currentButton == HIGH) {

        tempORhum != tempORhum;

    }

    lastButton = currentButton;

    if (tempORhum == LOW) {

        Serial.println(t);

    } else {
```

```
    Serial.println(h);  
}  
delay(2000);  
}
```

```
boolean debounce(boolean last) {  
    boolean current = digitalRead(switchPin);  
  
    if (last != current) {  
  
        delay(2000);  
  
        current = digitalRead(switchPin);  
    }  
  
    return current;  
    delay(2000);  
}
```