Name:______________________ Class: D16 Roll No: __

EXPERIMENT NO: 04

Aim: To monitor the Humidity and Temperature through Blynk App and Node MCU (ESP8266) & DHT11

Apparatus: Blynk App(https://sgp1.blynk.cloud/dashboard/853457/get-started) , Node MCU (ESP8266), DHT11

Theory:

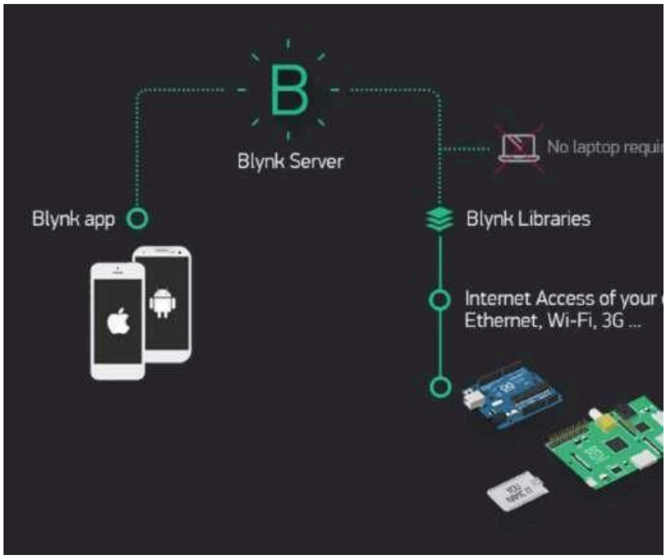

Fig. 1 *Workflow of Blynk app*

- Blynk is an loT platform with customizable mobile apps, private cloud, rules engine, and device management analytics dashboard, designed for easy and customizable Internet of Things applications.
- It can control hardware remotely, it can display sensor data, it can store data, visualize it and do many other things.
- It's a digital dashboard where one can build a graphic interface for the project by simply dragging and dropping widgets
- Blynk is not tied to some specific board or shield. Instead, it's supporting hardware of the user's choice
- Whether it is Arduino or Raspberry Pi which is linked to the Internet over Wi-Fi, Ethernet or ESP8266 chip, Blynk will get it online and ready for the Internet Of Things.

WORKING OF BLYNK APP:

There are three major components in the platform

Blynk App - allows the user to create amazing interfaces for the projects using various widgets provided **Blynk Server** - responsible for all the communications between the smartphone and hardware. One can use the Blynk Cloud or run the private Blynk server locally

Blynk Libraries - for all the popular hardware platforms - enable communication with the server and process all the incoming and outcoming commands

FEATURES:

- Supports majority of development boards like Arduino, RPI, esp8266
- Easy to use
- Awesome widgets like LCD, push buttons, labeled value, graphs
- Not restricted to local Wifi network
- Direct pin manipulation with no code writing
- Easy to integrate and add new functionality using virtual pins

PROCEDURE:

(A) Setting Up Blynk With Arduino IDE

1. Install the Blynk App library.
2. Once the Zip file is downloaded, extract it and individually copy all the folder to the libraries folder of the Arduino
3. Once done just open Arduino IDE and go to Sketch - > Include libraries and blynk can be seen in the menu.
4. It means that libraries have been included successfully.

(B)Blynk App Dashboard Setup for Monitoring Temperature & Humidity

1. After installing the app, sign in to create an account
2. After that Press on click on New Project and a screen will be displayed
3. Enter the name of the project and save it

4. Then Select the Board as ESP8266 and then the authentication token number will be sent to your registered email id
5. And then Finally click on the create button.
6. Now a dashboard screen appears Just click on the top most button "+" on the right comer to add gauges to the project.
7. Then configure its settings as Pin -
8. Virtual Select V1 for Temperature & V2 for Humidity.
9. The gauge can be set as a push type or as a switch.
10. Then upload the code to the node MCU to find the result.

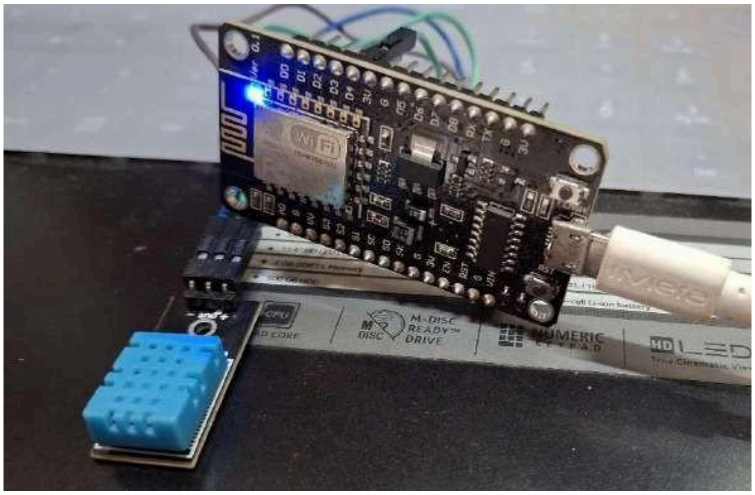

Fig. 2 *Hardware setup for Temperature and humidity reading*

CODE:

```
#define BLYNK_TEMPLATE_ID "TMPL6EQyNZGmf"
#define BLYNK_TEMPLATE_NAME "Weather Monitor"
#define BLYNK_AUTH_TOKEN "csO29wsYOwiV9aGX311ifPXNF23CTmy-"

#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>

#include <DHT.h>


char auth[] = BLYNK_AUTH_TOKEN;

char ssid[] = "shreyash";  // type your wifi name
char pass[] = "shreyash";  // type your wifi password
```

```
BlynkTimer timer;


#define DHTPIN D7 //Connect Out pin to D4 in NODE MCU
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);


void sendSensor()
{
  float h = dht.readHumidity();
  float t = dht.readTemperature(); // or dht.readTemperature(true) for
Fahrenheit

  if (isnan(h) || isnan(t)) {
    Serial.println("Failed to read from DHT sensor!");
    return;
  }
  // You can send any value at any time.
  // Please don't send more that 10 values per second.
    Blynk.virtualWrite(V0, t);
    Blynk.virtualWrite(V1, h);
    Serial.print("Temperature : ");
    Serial.print(t);
    Serial.print("    Humidity : ");
    Serial.println(h);
}
void setup()
{

  Serial.begin(115200);


  Blynk.begin(auth, ssid, pass);
  dht.begin();
  timer.setInterval(100L, sendSensor);

  }

void loop()
{
  Blynk.run();
```

```
 timer.run();
}
```

OUTPUT:

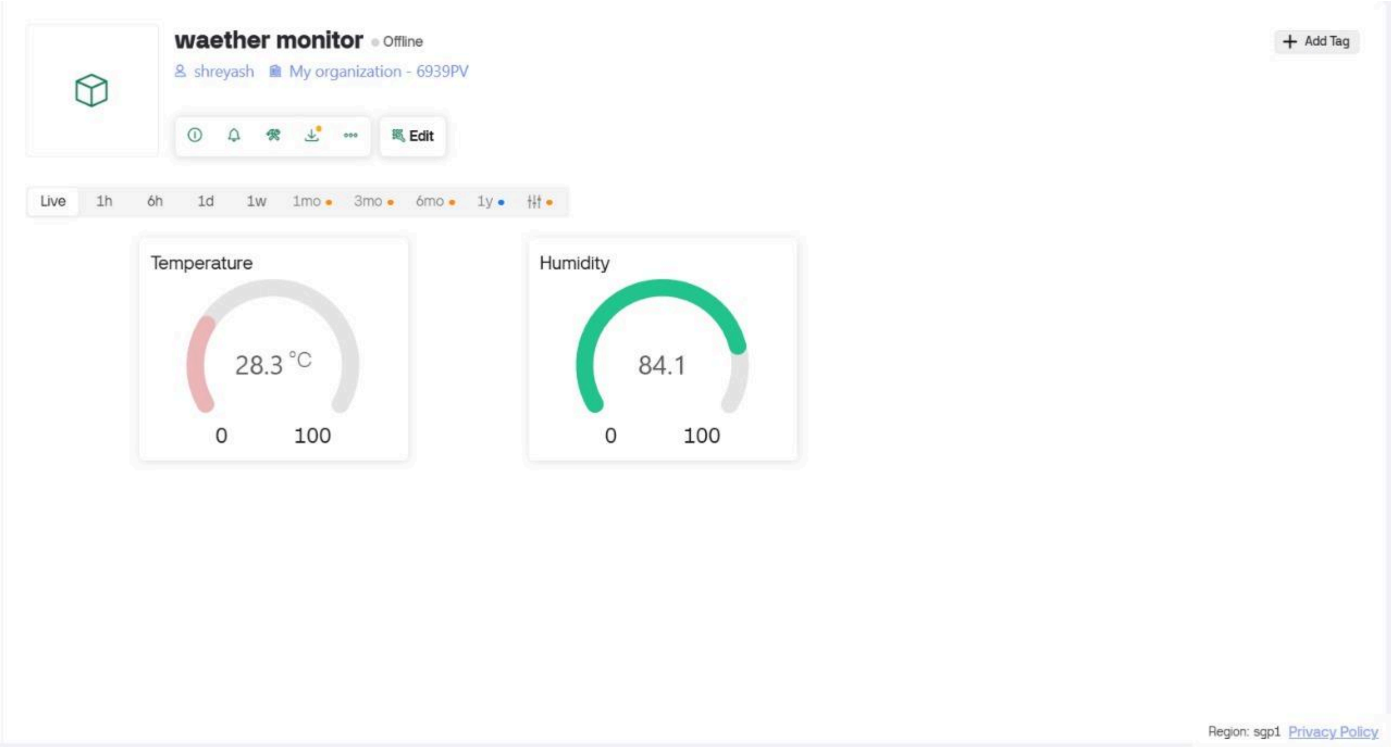

Fig. 3 *Blynk Dashboard with radial Gauge graph showing temperature and humidity*

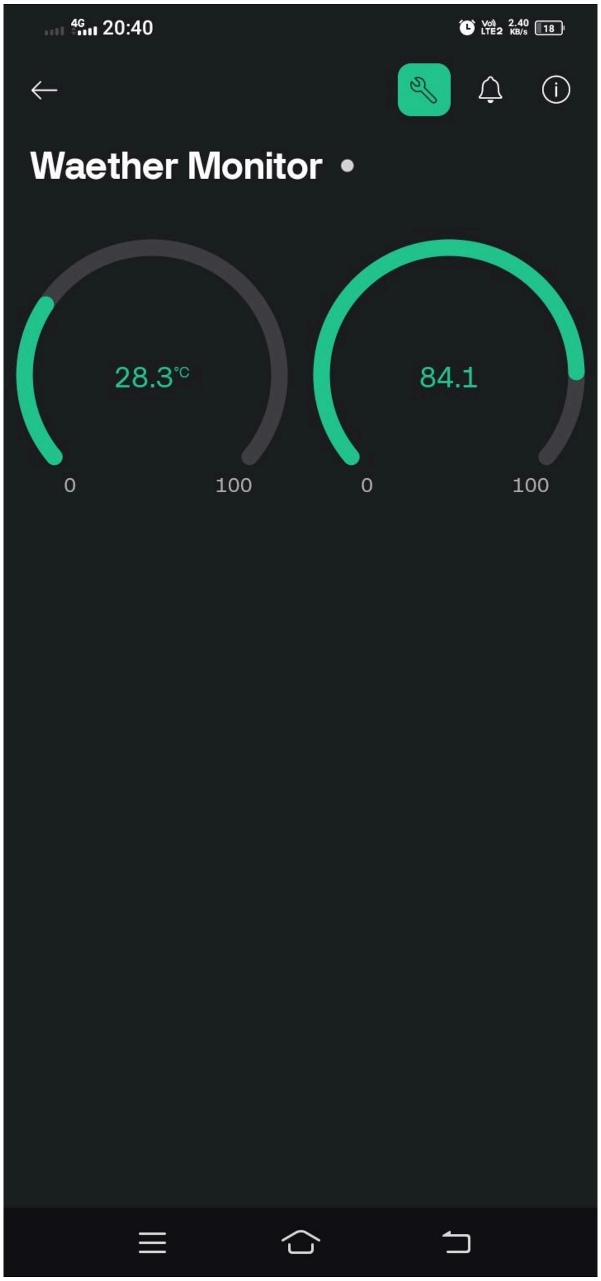

Fig.4 *Blynk weather monitor Dashboard showing temperature and humidity in Mobile phone*

CONCLUSION:

1. Interfacing NodeMCU with Blynk App.
2. Use of Blynk app key for data transfer.
3. Creating a Project on Blynk.
4. Adding gauges to the project for humidity & temperature.
5. Including library in Arduino IDE for Blynk App.