

Experiment No. 3: Interface a Sensor with NodeMCU and Post Values on Blynk App

Aim: The aim of this experiment is to monitor the temperature and humidity using a DHT11 sensor and post the values on the Blynk App using the NodeMCU (ESP8266).

Apparatus Required:

NodeMCU (ESP8266) – 1 unit

DHT11 sensor – 1 unit

Jumper wires

Breadboard

Smartphone (Android/iOS) with Blynk app installed

USB Cable for powering the NodeMCU and uploading the code

Internet connection (for NodeMCU and smartphone to connect)

Blynk app account (for accessing the data)



Blynk Features



Theory:

Blynk is an IoT (Internet of Things) platform that allows users to control and monitor devices remotely. It provides a user-friendly interface that enables interaction with hardware devices like Arduino, ESP8266, ESP32, and Raspberry Pi. Blynk allows you to create projects and control hardware through a mobile app.

Working:

- **Mobile App:** You create a project on the Blynk app by selecting a hardware device (e.g., NodeMCU, Arduino) and setting up widgets to control and display values. These widgets can display sensor data (e.g., temperature and humidity) or control outputs (e.g., LEDs, motors).
- **Blynk Cloud:** The app communicates with Blynk Cloud, a remote server, which acts as a bridge between your mobile app and the hardware. The cloud sends and receives data between your app and NodeMCU via the internet.

Hardware Device: The hardware (NodeMCU, ESP8266) runs a Blynk sketch, which connects to the Blynk Cloud using Wi-Fi. It sends data from sensors or receives data from control widgets on the app.

- **Sensor (DHT11):** The DHT11 sensor is connected to the NodeMCU and measures the temperature and humidity.
- **NodeMCU:** It reads the sensor values and sends them to the Blynk App using Wi-Fi.
- **Blynk App:** The app receives and displays the sensor data on the mobile phone in real-time. You can monitor the temperature and humidity remotely.

Key Features of Blynk:

- **Remote Monitoring:** You can view sensor data remotely on the app.
- **Widgets:** Blynk offers various widgets such as buttons, sliders, graphs, and more to display or control the hardware.
- **Easy to Use:** No advanced programming skills are required to set up the mobile app interface.

- **Real-Time Data:** It supports real-time data monitoring.
- **Cloud Integration:** Data is sent to Blynk's cloud servers, which can be accessed from anywhere.

Stepwise Procedure:

Step 1: Setting Up the Blynk App

1. Install Blynk App:

- Download and install the **Blynk app** on your smartphone (available for both Android and iOS).

2. Create an Account and New Project:

- Open the app and sign up (if you don't have an account).
- Create a new project in the app. Choose **ESP8266** as the device.
- Set up a **new project** and note the **Auth Token** that Blynk sends to your email. You will need this token in your code.

3. Add Widgets:

- Add a **Value Display** widget for displaying the temperature and humidity readings.
- Set the output pins for the widgets.
- Link the widgets to virtual pins (e.g., V1 for temperature and V2 for humidity).

4. Get the Auth Token:

- You will receive an **Auth Token** via email. This token is used to authenticate the device to the Blynk server.

Step 2: Connecting the DHT11 Sensor to NodeMCU

1. Wiring the DHT11 Sensor:

- Connect the **VCC** pin of the DHT11 to the **3V3** pin of the NodeMCU. Connect the **GND** pin of the DHT11 to the **GND** pin of the NodeMCU.
- Connect the **Data** pin of the DHT11 to **D2** (GPIO4) pin of NodeMCU (you can choose another GPIO pin, but modify the code accordingly).

Step 3: Installing the Required Libraries

1. Install Libraries in Arduino IDE:

- Open the Arduino IDE and go to **Sketch > Include Library > Manage Libraries**.
- Search for and install **DHT sensor library** by Adafruit.
- Install the **Blynk library** for NodeMCU.

Step 4: Writing the Code

```
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include <DHT.h>
// Replace with your Blynk Auth Token
char auth[] = "YourBlynkAuthToken";

// Replace with your network credentials
char ssid[] = "YourWiFiSSID";
char pass[] = "YourWiFiPassword";

// Set the DHT11 pin and type
#define DHTPIN D2
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);

void setup() {
  // Start serial communication for debugging
  Serial.begin(9600);

  // Connect to Wi-Fi
  WiFi.begin(ssid, pass);
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.println("Connecting to WiFi...");
  }
  Serial.println("Connected to WiFi");
```

```
// Start Blynk
Blynk.begin(auth, ssid, pass);

// Initialize DHT sensor
dht.begin();
}
void loop() {
  // Read temperature and humidity
  float temp = dht.readTemperature();
  float humidity = dht.readHumidity();

  // Send the values to Blynk app
  Blynk.virtualWrite(V1, temp); // Send temperature to V1
  Blynk.virtualWrite(V2, humidity); // Send humidity to V2

  // Wait for 2 seconds before reading again
  delay(2000);

  // Run Blynk process
  Blynk.run();
}
```

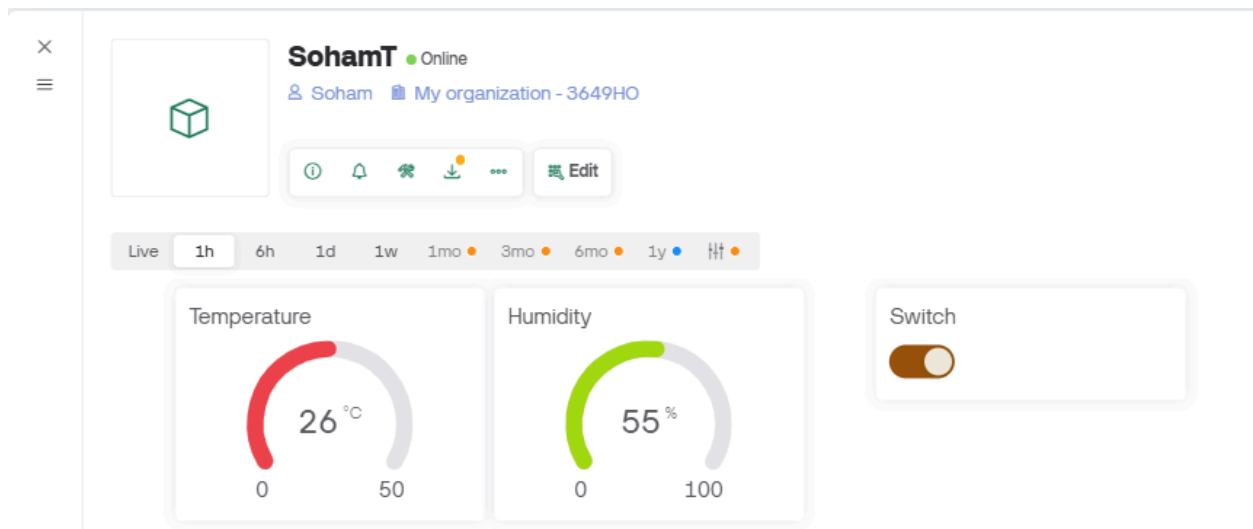
Step 5: Uploading the Code to NodeMCU

1. Select the correct **Board (NodeMCU 1.0)** and **Port** in the **Tools** menu of Arduino IDE.
2. Click on the **Upload** button to upload the code to the NodeMCU.

Step 6: Monitor Data on Blynk App

1. Open the Blynk app on your smartphone.
2. You should now see the temperature and humidity data displayed on the widgets you created.
3. The values should update every 2 seconds as per the delay in the code.

Output:



```
Temperature: 26.20 °C Humidity: 53.30 %
Roll no : 54 59 61 68
Temperature: 26.20 °C Humidity: 53.30 %
Roll no : 54 59 61 68
Temperature: 26.20 °C Humidity: 53.30 %
Roll no : 54 59 61 68
Temperature: 26.40 °C Humidity: 59.60 %
Roll no : 54 59 61 68
Temperature: 26.50 °C Humidity: 58.50 %
Roll no : 54 59 61 68
```