Exp.11: Temperature-Controlled Fan Interface with ESP8266 and Blynk IoT

OBJECTIVES:

1. To interface a temperature-controlled fan with the ESP8266 NodeMCU board.
2. To monitor and control the fan based on temperature readings using the DHT11 sensor.
3. To visualize temperature data and control the fan remotely using the Blynk IoT app.

MATERIALS REQUIRED:

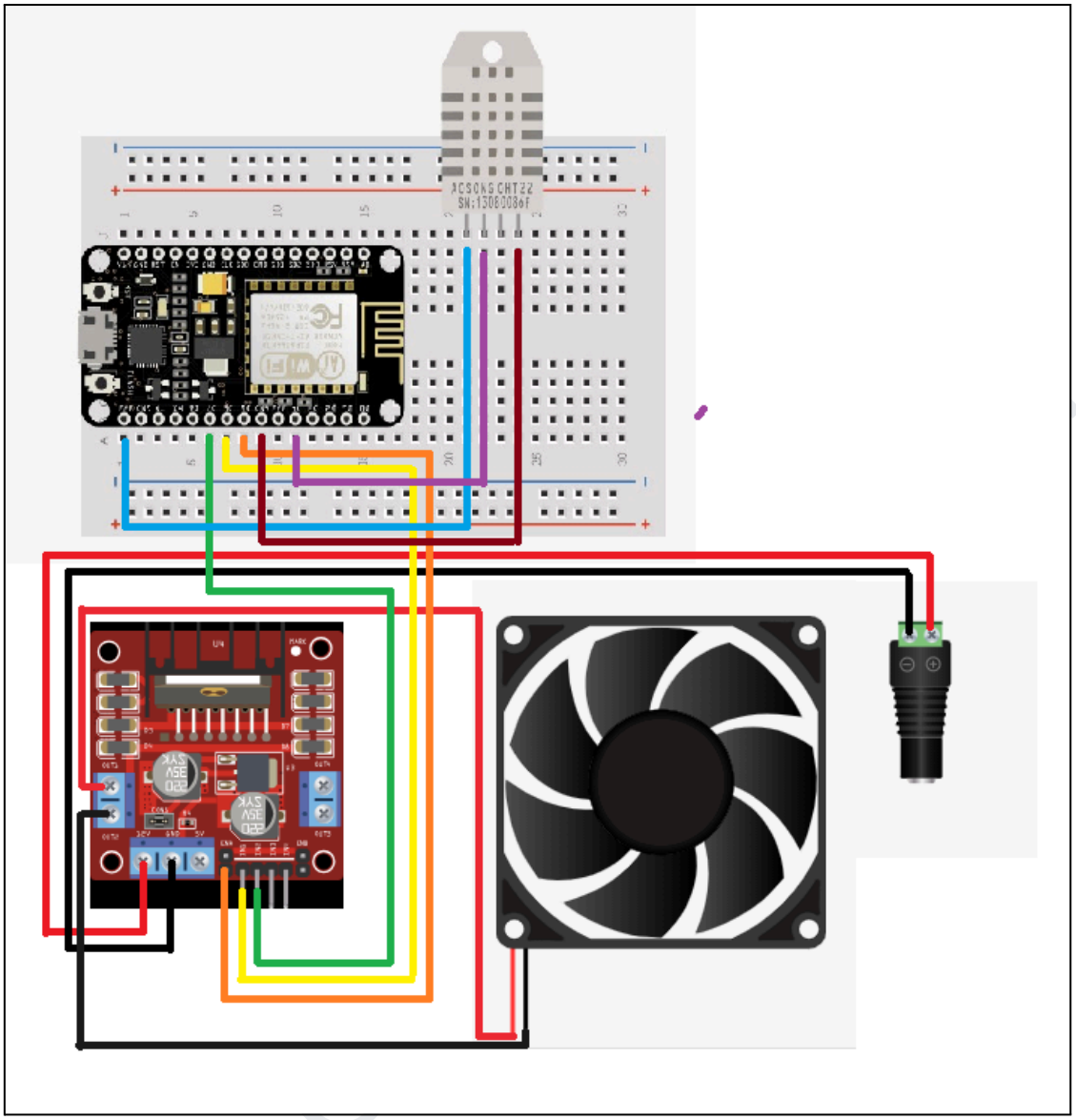
- ESP8266 NodeMCU board
- Motor driver (e.g., L298N)
- DHT11 temperature and humidity sensor
- AC fan
- Breadboard
- Jumper wires
- USB cable
- Computer with Arduino IDE installed
- Blynk app installed on a smartphone or tablet

THEORY:

The ESP8266 NodeMCU board can control an AC fan based on temperature readings from the DHT11 sensor. A motor driver module, such as the L298N, is used to control the fan's operation.

The fan is turned on or off based on the temperature data received from the DHT11 sensor. The ESP8266 connects to Wi-Fi, enabling remote monitoring and control through the Blynk app.

CIRCUIT DIAGRAM:



Motor Driver Pin Configuration:

- IN1, IN2: Control pins for the fan (connect to ESP8266 GPIO pins)
- OUT1, OUT2: Connected to the fan terminals
- VCC: Connected to a suitable power supply (e.g., 12V)
- GND: Connected to ground

DHT11 Sensor Pin Configuration:

- V_{CC}: Connected to 3.3V or 5V
- GND: Connected to ground
- DATA: Connected to a GPIO pin on ESP8266 (e.g., D4)

Fan Connection:

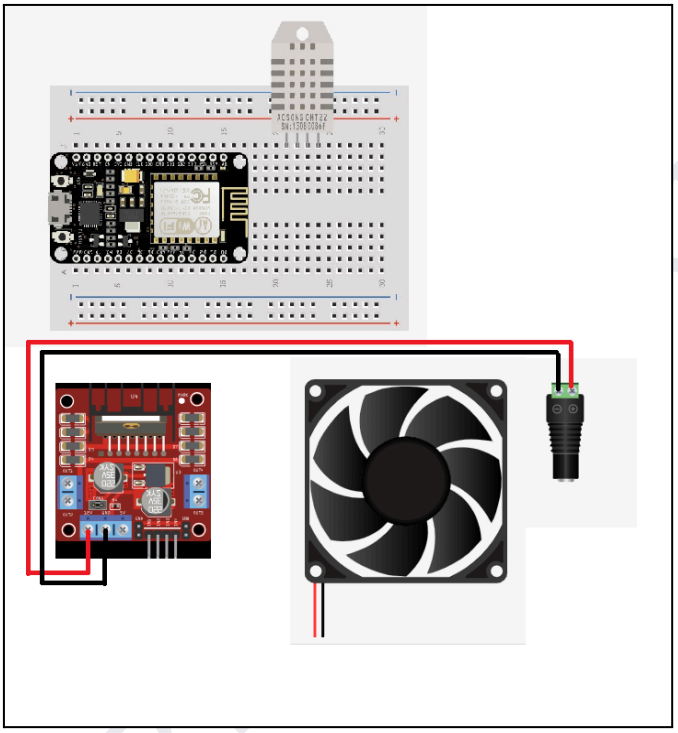
Fan Terminals: Connected to the motor driver outputs (OUT1 and OUT2)

PROCEDURE:

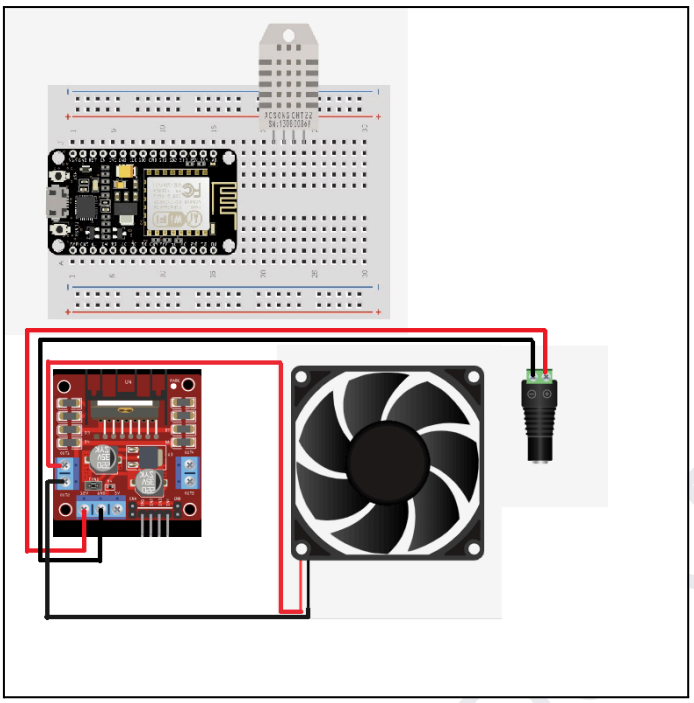
1. Hardware Setup:

- Connect the motor driver to power supply:

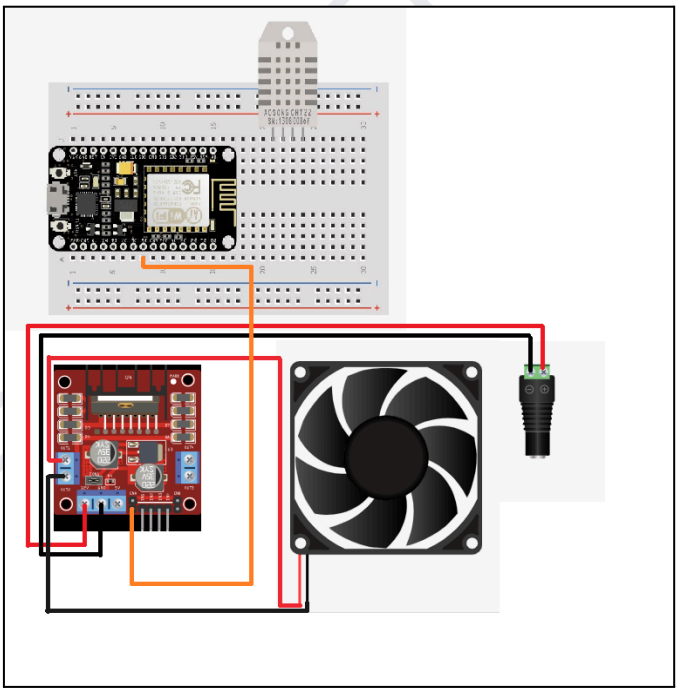
1. V_{CC} to a suitable power supply (e.g., 12V)
2. GND to GND



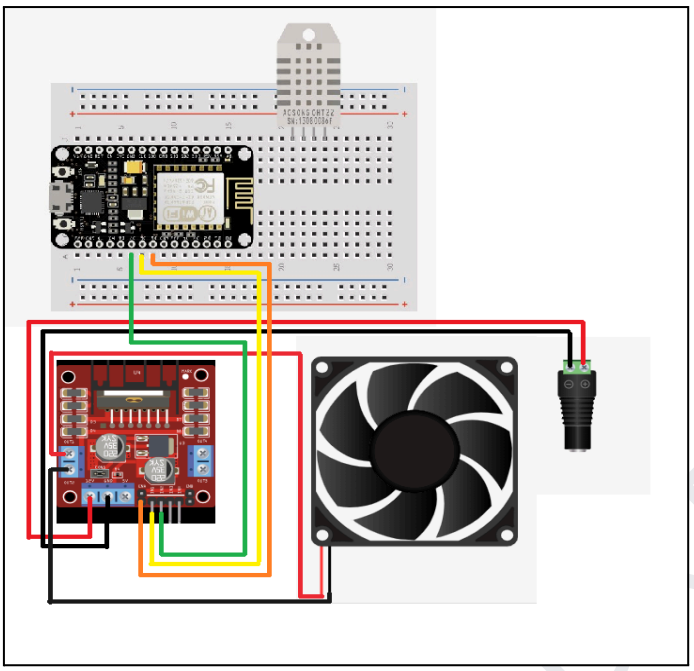
- Connect the fan to the motor driver as per the motor driver specifications.



- ENA to D5 on ESP8266

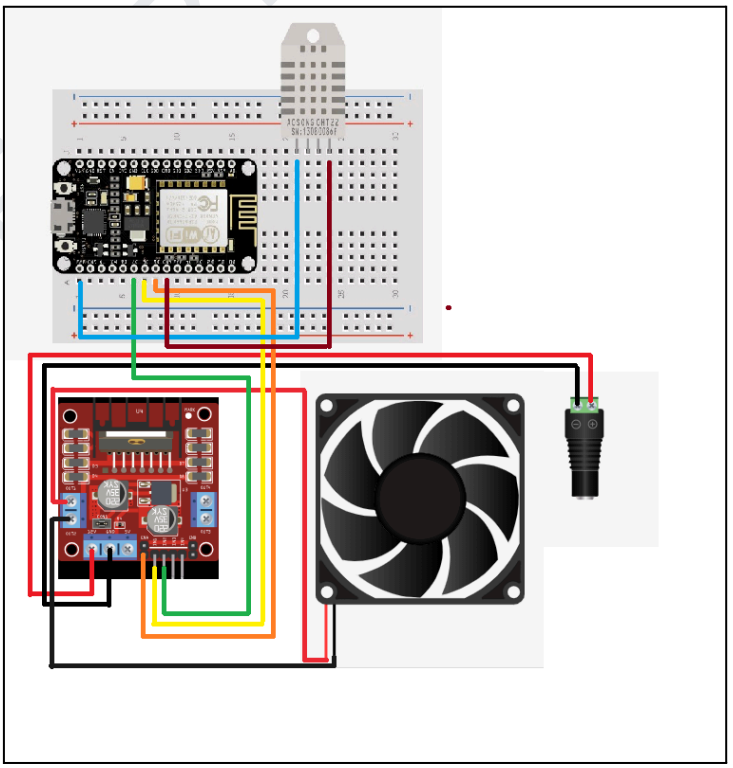


- IN1 to D6 on ESP8266
- IN2 to D7 on ESP8266

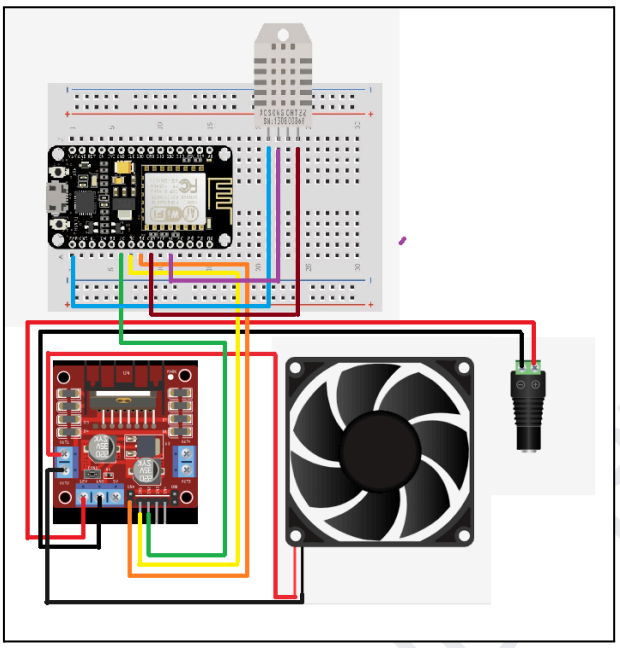


- Connect the DHT11 sensor to the ESP8266:

1. V_{CC} to 3.3V (or 5V if using a 5V DHT11 sensor)
2. GND to GND



3. DATA to D4 on ESP8266



2. Software Setup:

- Open the Arduino IDE on your computer.
- Install the necessary libraries: Blynk library, DHT sensor library, and any required dependencies.
- Install the libraries by navigating to Sketch -> Include Library -> Manage Libraries and searching for “Blynk” and “DHT sensor library”.

3. Blynk App Setup:

- Open the Blynk app on your smartphone/tablet.
- Create a new project and note the Auth Token.
- Add a ”Value Display” widget to monitor temperature (set to Virtual Pin V1).
- Add a “Button” widget to manually control the fan (set to Virtual Pin V0).

4. Programming:

- Connect the ESP8266 to your computer using a USB cable.
- In the Arduino IDE, write the following code:

```cpp
// Blynk credentials
#define BLYNK_TEMPLATE_ID "__"
#define BLYNK_TEMPLATE_NAME "__"
```

```
#define BLYNK_AUTH_TOKEN "__"

#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include <DHT.h>

// Replace these with your network credentials
char ssid[] = "WIFI_Name";
char pass[] = "Password";

// Define the relay control pins
const int motorPin1 = D5; // GPIO14
const int motorPin2 = D6; // GPIO12

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

// Function to control fan based on temperature
BLYNK_WRITE(V0) {
  int pinValue = param.asInt(); // Get the value from the button
  if (pinValue == 1) {
    digitalWrite(motorPin1, HIGH); // Turn fan on
    digitalWrite(motorPin2, LOW);
  } else {
    digitalWrite(motorPin1, LOW); // Turn fan off
    digitalWrite(motorPin2, LOW);
  }
}

// Function to read temperature and send to Blynk app
void sendSensorData() {
  float humidity = dht.readHumidity();
  float temperature = dht.readTemperature();
  if (isnan(temperature) || isnan(humidity)) {
    Serial.println("Failed to read from DHT sensor!");
    return;
  }
  Blynk.virtualWrite(V1, temperature); // Send temperature data to Blynk app
  if (temperature > 30) { // Set your desired temperature threshold
    digitalWrite(motorPin1, HIGH); // Turn fan on
    digitalWrite(motorPin2, LOW);
  } else {
    digitalWrite(motorPin1, LOW); // Turn fan off
    digitalWrite(motorPin2, LOW);
  }
}
```

```
void setup() {
  // Start serial communication
  Serial.begin(115200);

  // Initialize motor control pins as output
  pinMode(motorPin1, OUTPUT);
  pinMode(motorPin2, OUTPUT);
  digitalWrite(motorPin1, LOW);
  digitalWrite(motorPin2, LOW);

  // Initialize DHT sensor
  dht.begin();

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

  // Initialize Blynk
  Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass);
}

void loop() {
  Blynk.run(); // Run Blynk
  sendSensorData(); // Read sensor and control fan
  delay(5000); // Update every 5 seconds
}
```

- Replace `"WIFI_NAME"`, `"Password"`, and `BLYNK_AUTH_TOKEN` with your actual Wi-Fi credentials and Blynk Auth Token.
- Upload the code to the ESP8266.

5. Running the Experiment:

- Open the Serial Monitor in the Arduino IDE to observe the connection status and temperature readings.
- Open the Blynk app and monitor the temperature via the Value Display widget. Use the Button widget to manually control the fan.

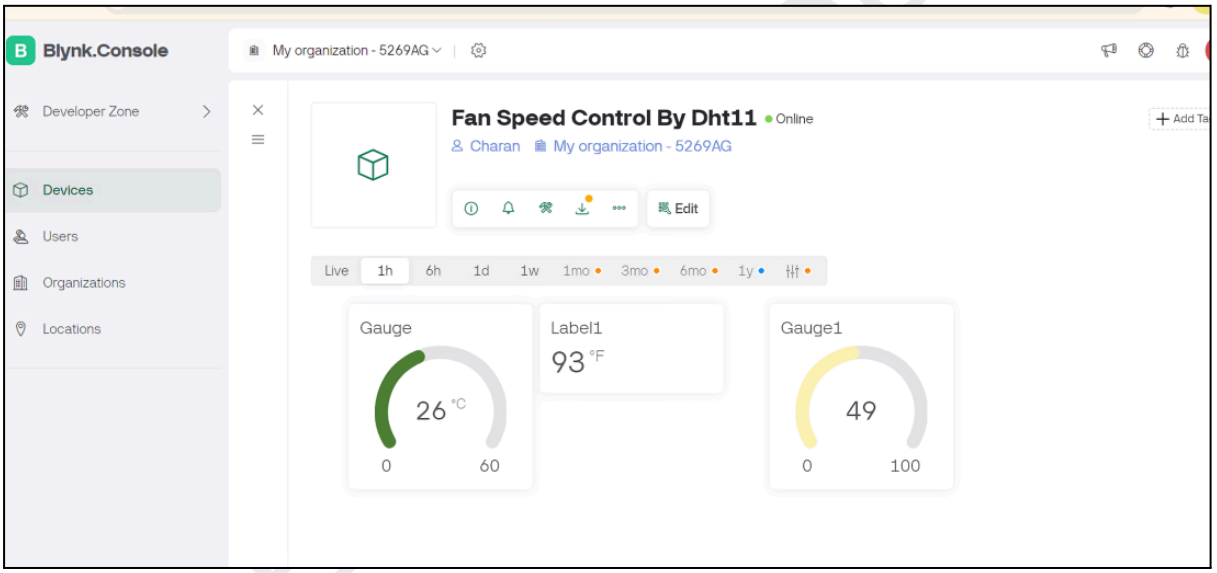
Observations:

S.No	Temperature (ºC\ºF)	Fan Speed

Table 1: Fan speed at different temperature ranges in celsius or fahrenheit

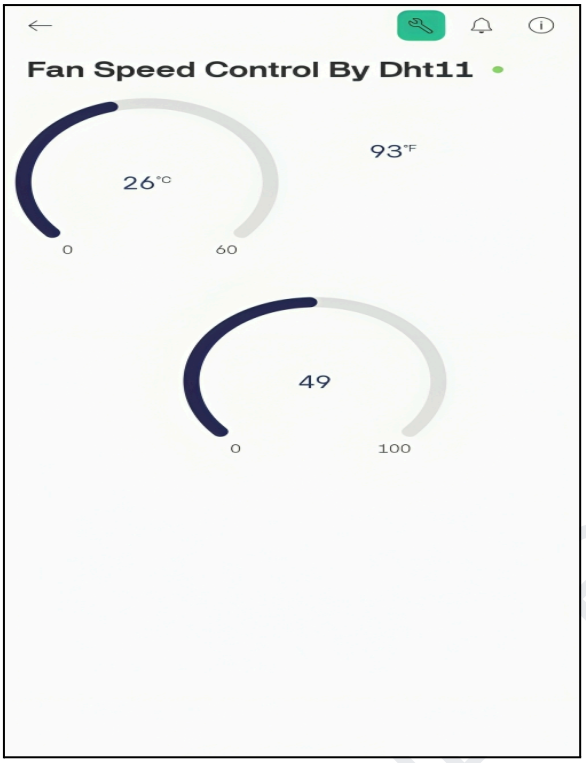
Observation in Blynk Website:

- Verify the real-time temperature readings displayed on the Blynk dashboard.



Observation in Blynk IoT Mobile App:

- Check if the fan turns on or off based on the temperature threshold and manual control via the Button widget.



Result:

The temperature-controlled fan was successfully managed using the DHT11 sensor and motor driver through the ESP8266 NodeMCU board. Temperature data was displayed on the Blynk app, and the fan was controlled both manually and automatically based on temperature readings.

Conclusion:

This experiment demonstrates how to integrate a temperature sensor, motor driver, and fan with the ESP8266 board, and how to control and monitor them remotely using the Blynk IoT platform. The successful implementation highlights the capability of ESP8266 and Blynk for creating IoT-based temperature control systems.

Appendix:

A. Symbols, Units, and Abbreviations:

- V: Voltage
- V_{CC}: Voltage Common Collector
- GND: Ground
- GPIO: General Purpose Input/Output
- DHT: Digital Humidity and Temperature Sensor

B. Tools Required:

- ESP8266 NodeMCU board
- Motor driver (e.g., L298N)

- DHT11 sensor
- AC fan
- Breadboard
- Jumper wires
- USB cable
- Computer with Arduino IDE installed
- Blynk app installed on a smartphone or tablet

C. Additional Resources:

- ESP8266 Documentation
- Arduino IDE Installation Guide
- Blynk Documentation
- DHT11 Sensor Guide
- Motor Driver Guide

D. Reference link with QR code

https://www.youtube.com/watch?v=klcjGD58hzo

This format provides a clear and comprehensive guide for conducting the experiment, ensuring students can follow along and achieve the desired outcomes.