

IOT LAB 49+50

22MIA1062

PRIYANSHU TEOTIA

EXPERIMENT 2

AIM : USING DHT 11 SENSOR DISPLAY THE TEMPERATURE AND HUMIDITY

PROCEDURE:

Library Installation:

1. Open the **Arduino IDE**.
2. Go to **Sketch > Include Library > Manage Libraries...**
3. Search for "**DHT11**" and install a compatible DHT11 library (e.g., "**DHT11 by Rob Tillaart**").

Circuit Connections:

Connect the DHT11 sensor to the Arduino Uno as follows:

1. **DHT11 Pins:**
 - **VCC** (power) → **5V** on Arduino
 - **GND** (ground) → **GND** on Arduino
 - **DATA** (signal) → **Digital Pin 2** on Arduino
2. **Optional Pull-up Resistor:**
 - Connect a **10k ohm resistor** between the **DATA pin** and **VCC (5V)** to ensure a stable signal.

Steps to Upload the Code:

1. **Connect the Arduino Uno** to your computer using the USB cable.
2. Open the **Arduino IDE**.
3. **Select the correct board and port:**
 - Go to **Tools > Board > Arduino Uno**.
 - Go to **Tools > Port** and select the port corresponding to your Arduino.
4. **Copy and paste** the code into the Arduino IDE.
5. Click on the **Upload button** (right-arrow) to upload the code.

INPUT:

```
// Include the DHT11 library for interfacing with the sensor.
#include <DHT11.h>

// Create an instance of the DHT11 class.
// - For Arduino: Connect the sensor to Digital I/O Pin 2.
// - For ESP32: Connect the sensor to pin GPIO2 or P2.
// - For ESP8266: Connect the sensor to GPIO2 or D4.
DHT11 dht11(2);

void setup() {
    // Initialize serial communication to allow debugging and data
    readout.
    // Using a baud rate of 9600 bps.
    Serial.begin(9600);

    // Uncomment the line below to set a custom delay between sensor
    readings (in milliseconds).
    // dht11.setDelay(500); // Set this to the desired delay. Default is
    500ms.
}

void loop() {
    int temperature = 0;
    int humidity = 0;

    // Attempt to read the temperature and humidity values from the DHT11
    sensor.
    int result = dht11.readTemperatureHumidity(temperature, humidity);

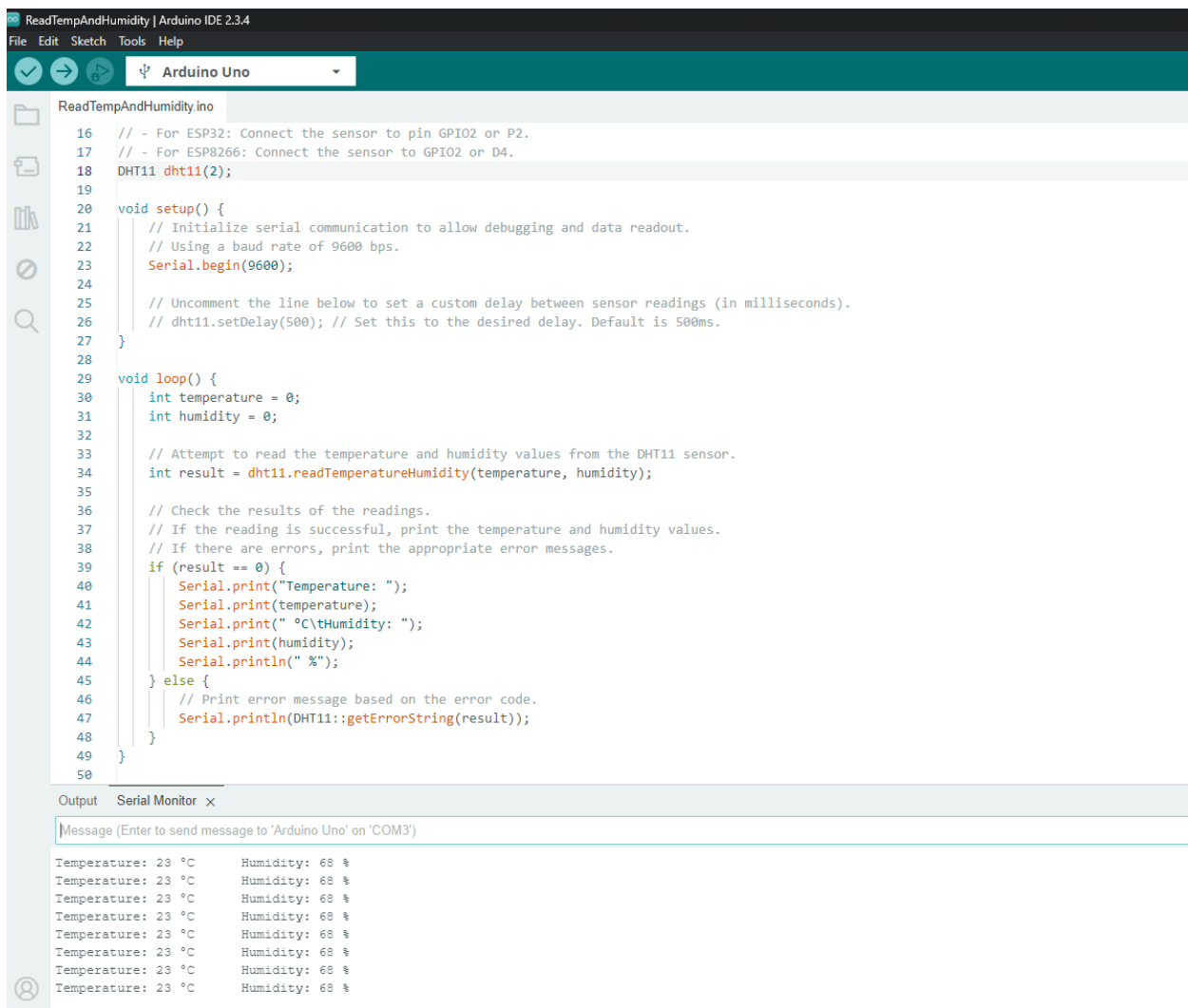
    // Check the results of the readings.
    // If the reading is successful, print the temperature and humidity
    values.
    // If there are errors, print the appropriate error messages.
    if (result == 0) {
        Serial.print("Temperature: ");
        Serial.print(temperature);
        Serial.print(" °C\tHumidity: ");
        Serial.print(humidity);
    }
}
```

```

        Serial.println(" %");
    } else {
        // Print error message based on the error code.
        Serial.println(DHT11::getErrorString(result));
    }
}

```

OUTPUT:



The screenshot shows the Arduino IDE interface. The top toolbar includes icons for file operations, compilation, and uploading. The 'Board' dropdown is set to 'Arduino Uno'. The main editor displays the code for 'ReadTempAndHumidity.ino'. The code includes comments for ESP32 and ESP8266 pin configurations, a DHT11 sensor declaration, and functions for setup and loop. The loop function reads temperature and humidity data from the DHT11 sensor and prints the results to the Serial Monitor. The Serial Monitor at the bottom shows the output of the program, displaying temperature and humidity values repeatedly.

```

16 // - For ESP32: Connect the sensor to pin GPIO2 or P2.
17 // - For ESP8266: Connect the sensor to GPIO2 or D4.
18 DHT11 dht11(2);
19
20 void setup() {
21     // Initialize serial communication to allow debugging and data readout.
22     // Using a baud rate of 9600 bps.
23     Serial.begin(9600);
24
25     // Uncomment the line below to set a custom delay between sensor readings (in milliseconds).
26     // dht11.setDelay(500); // Set this to the desired delay. Default is 500ms.
27 }
28
29 void loop() {
30     int temperature = 0;
31     int humidity = 0;
32
33     // Attempt to read the temperature and humidity values from the DHT11 sensor.
34     int result = dht11.readTemperatureHumidity(temperature, humidity);
35
36     // Check the results of the readings.
37     // If the reading is successful, print the temperature and humidity values.
38     // If there are errors, print the appropriate error messages.
39     if (result == 0) {
40         Serial.print("Temperature: ");
41         Serial.print(temperature);
42         Serial.print(" °C\tHumidity: ");
43         Serial.print(humidity);
44         Serial.println(" %");
45     } else {
46         // Print error message based on the error code.
47         Serial.println(DHT11::getErrorString(result));
48     }
49 }
50

```

Output Serial Monitor ×

Message (Enter to send message to 'Arduino Uno' on 'COM3')

```

Temperature: 23 °C Humidity: 68 %
Temperature: 23 °C Humidity: 68 %
Temperature: 23 °C Humidity: 68 %
Temperature: 23 °C Humidity: 68 %
Temperature: 23 °C Humidity: 68 %
Temperature: 23 °C Humidity: 68 %
Temperature: 23 °C Humidity: 68 %
Temperature: 23 °C Humidity: 68 %

```

RESULT: hence, the led bulb will glow successfully.

