

Edge Computing Lab

Class: TY-AIEC

School of Computing, MIT Art Design Technology

University *Academic Year: 2024-25*

Experiment No. 2

Name: Viresh Kamlapure

Class: TY AIEC

Enrollment No: MITU23BTCSD082

Roll No: D2233082

Experiment Title: Real-Time Temperature and Humidity Monitoring using DHT11 Sensor and Flask

Objective:

To interface a DHT11 sensor with a Raspberry Pi and create a web application using Flask to display real-time temperature and humidity data.

Step 1: Install Required Libraries

1. Update the package list:

```
1. sudo apt update
2.
```

2. Install Python 3 and pip:

```
1. sudo apt install python3-pip -y
2.
```

3. Install the Adafruit DHT library:

```
1. pip3 install adafruit-circuitpython-dht
2.
```

4. Install additional dependencies for DHT11 on Raspberry Pi:

```
1. sudo apt-get install libgpiod2 -y
2.
```

5. Install Flask:

```
1. pip3 install flask
2.
```

Step 2: Connect the DHT11 Sensor

1. Connect the DHT11 sensor to the Raspberry Pi GPIO pins:

- **VCC:** Connect to the 3.3V pin on the Raspberry Pi.
- **GND:** Connect to any ground (GND) pin on the Raspberry Pi.
- **DATA:** Connect to a GPIO pin (e.g., GPIO4).

Step 3: Flask Application and AJAX Integration

3.1: Flask App to Serve JSON Data

Create a Python file named `dht11_ajax.py` and add the following code:

```
from flask import Flask, render_template, jsonify
```

```
import adafruit_dht
import board
```

```
app = Flask(__name__)
DHT_SENSOR_PIN = board.D4 # GPIO4
```

```
def read_dht_sensor():
    dht_sensor = adafruit_dht.DHT11(DHT_SENSOR_PIN)
    try:
        temperature = dht_sensor.temperature
        humidity = dht_sensor.humidity
        return temperature, humidity
    except RuntimeError:
        return None, None
    finally:
        dht_sensor.exit()
```

```
@app.route('/')
def index():
    return render_template('index.html')
```

```
@app.route('/sensor-data')
def sensor_data():
    temperature, humidity = read_dht_sensor()
    if temperature is not None and humidity is not None:
        data = {
            "temperature": f"{temperature:.1f} °C",
            "humidity": f"{humidity:.1f} %"
        }
    else:
```

```

        data = {"error": "Unable to read sensor data."}
        return jsonify(data)

if __name__ == '__main__':
    app.run(host='0.0.0.0', port=5000, debug=False)

```

Code Explanation:

- **Imports:**

- Flask: For creating the web application.
- render_template: To render the HTML file.
- jsonify: To return sensor data in JSON format.
- adafruit_dht and board: To interact with the DHT11 sensor.

- **DHT Sensor Initialization:** The DHT_SENSOR_PIN is set to GPIO4.

- **read_dht_sensor Function:**

- Reads the temperature and humidity values from the sensor.
- Handles RuntimeError if the sensor cannot be read.

- **Routes:**

- /: Serves the main webpage.
- /sensor-data: Provides temperature and humidity data in JSON format.

3.2: HTML Template

Create a new folder named templates in the same directory as dht11_ajax.py. Inside this folder, create a file named index.html with the following content:

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Temperature and Humidity</title>
</script>
function updateSensorData() {
  fetch('/sensor-data')
    .then(response => response.json())
    .then(data => {
      if (data.error) {
        document.getElementById('temperature').innerText = data.error;
        document.getElementById('humidity').innerText = "";
      } else {

```

```

        document.getElementById('temperature').innerText = "Temperature: " +
data.temperature;
        document.getElementById('humidity').innerText = "Humidity: " + data.humidity;
    }
})
.catch(error => console.error('Error fetching sensor data:', error));
}

// Refresh sensor data every 2 seconds
setInterval(updateSensorData, 2000);
window.onload = updateSensorData;
</script>
</head>
<body>
  <h1>Temperature and Humidity</h1>
  <p id="temperature">Loading...</p>
  <p id="humidity"></p>
</body>
</html>

```

Code Explanation:

• JavaScript Function:

- updateSensorData: Fetches sensor data from /sensor-data using the fetch API.
- Updates the webpage with temperature and humidity values every 2 seconds.

• Dynamic Update:

- Data is fetched asynchronously without reloading the entire webpage.
- Ensures a smooth user experience.

Step 4: Run the Application

1. Start the Flask application:

```

1. python3 dht11_ajax.py
2.

```

2. Open a web browser and navigate to:

```

1. http://<raspberrypi-ip>:5000
2.

```

Replace <raspberrypi-ip> with your Raspberry Pi's IP address.

Observation and Results

- Observe real-time temperature and humidity data updating every 2 seconds on the webpage.
- Verify the accuracy of the sensor readings.

Conclusion

In this experiment, you successfully:

- Interfaced a DHT11 sensor with a Raspberry Pi.
- Built a Flask web application to serve real-time sensor data.
- Utilized AJAX to dynamically update webpage content without refreshing the page.

References

1. Adafruit CircuitPython DHT Documentation
2. Flask Documentation
3. Raspberry Pi GPIO Pinout

