

Edge Computing Lab
Class: TY-AIEC
School of Computing, MIT Art Design Technology University
Academic Year: 2024-25

Experiment No. 2

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:

```
1. from flask import Flask, render_template, jsonify
2. import adafruit_dht
3. import board
4.
5. app = Flask(__name__)
6. DHT_SENSOR_PIN = board.D4 # GPIO4
7.
8. def read_dht_sensor():
9.     dht_sensor = adafruit_dht.DHT11(DHT_SENSOR_PIN)
10.    try:
11.        temperature = dht_sensor.temperature
12.        humidity = dht_sensor.humidity
13.        return temperature, humidity
14.    except RuntimeError:
15.        return None, None
16.    finally:
17.        dht_sensor.exit()
18.
19. @app.route('/')
20. def index():
21.     return render_template('index.html')
22.
23. @app.route('/sensor-data')
24. def sensor_data():
25.     temperature, humidity = read_dht_sensor()
26.     if temperature is not None and humidity is not None:
27.         data = {
28.             "temperature": f"{temperature:.1f} °C",
29.             "humidity": f"{humidity:.1f} %"
30.         }
31.     else:
32.         data = {"error": "Unable to read sensor data."}
33.     return jsonify(data)
34.
35. if __name__ == '__main__':
36.     app.run(host='0.0.0.0', port=5000, debug=False)
37.
```

7

Lab Manual

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:

```
1. <!DOCTYPE html>

2. <html lang="en">
3. <head>
4. <meta charset="UTF-8">
5. <meta name="viewport" content="width=device-width, initial-scale=1.0">
6. <title>Temperature and Humidity</title>
7. <script>
8. function updateSensorData() {
9.   fetch('/sensor-data')
10.  .then(response => response.json())
11.  .then(data => {
12.    if (data.error) {
13.      document.getElementById('temperature').innerText = data.error;
14.      document.getElementById('humidity').innerText = "";
15.    } else {
16.      document.getElementById('temperature').innerText = "Temperature: " + data.temperature;
17.      document.getElementById('humidity').innerText = "Humidity: " + data.humidity;
18.    }
19.  })
20.  .catch(error => console.error('Error fetching sensor data:', error));
21. }
22.
23. // Refresh sensor data every 2 seconds
24. setInterval(updateSensorData, 2000);
25. window.onload = updateSensorData;
26. </script>
27. </head>
28. <body>
29. <h1>Temperature and Humidity</h1>
30. <p id="temperature">Loading...</p>
31. <p id="humidity"></p>
32. </body>
33. </html>
34.
```

8

Lab Manual

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.

Viva Questions

Viva Questions

Name : Deepika Modgi

Batch : TYAIEEC - C

Roll no : 2223928

Subject : EC Lab.

Experiment 2 : DHT11 with Raspberry Pi and
Web server Using Flask

Questions :

Q1] What is the function of the DHT11 sensor, and how does it measure temperature and humidity?

Ans. DHT11 sensor is used to measure temperature and humidity

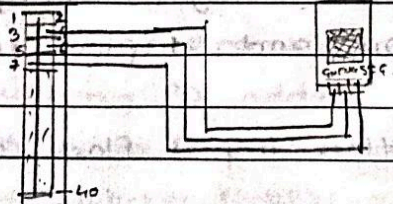
i] It has a thermistor to measure temperature

ii] It has a capacitive humidity sensor to measure moisture

iii] The sensor communicates via a single wire digital sensor and sends temperature ($^{\circ}\text{C}$) and humidity (in %) as a 40-bit data packet.

Q2] How do you interface the DHT11 with Raspberry Pi using GPIO pins? Provide a simple circuit diagram.

Ans. Connect GND to pin 6, VCC to pin 4 and SIG to pin 7.





Q3] What is flask and how can it be used to create a web page on Raspberry Pi?

Ans. Flask is a lightweight Python web framework used to create web applications and APIs.

i] Serves sensor data via a web page.

ii] Provides a simple dashboard for IoT projects.

iii] Can control hardware via a web interface.

Steps:

i] install flask : `pip install flask`.

ii] Create simple app (app.py):

```
from flask import Flask
```

```
app = Flask(__name__)
```

```
@app.route('/')  
def home():
```

```
    return "Hello, RaspberryPi!"
```

```
if __name__ == '__main__':
```

```
    app.run(host='0.0.0.0', port=5000)
```

iii] Run server:

```
python3 app.py
```

iv] Access webpage from another device:

```
hostname -I
```

open browser and visit :

```
http://<Raspberry-Pi-IP>:5000
```

Q4] Write basic Python script to read data from DHT11 and display it on flask.

Ans.

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

```
import Adafruit_DHT
```

```
app = Flask(__name__)
```

```
DHTsensor = Adafruit_DHT.DHT11
```

```
DHT_PIN = 7
```

```
@app.route('/')
```

```
def index():
```

```
    humidity, temperature = Adafruit_DHT.read  
                                (DHTsensor, DHT_PIN)
```

```
    if humidity is not None and temperature  
        is not None:
```

```
        return f"Temperature: {temperature}°C,  
                Humidity: {humidity}% "
```

```
    else:
```

```
        return "Failed to retrieve data from  
                sensor"
```

```
if __name__ == '__main__':
```

```
    app.run(host='0.0.0.0', port=5000,  
            debug=True)
```

Q5] How can you make the flask web server accessible from another device in same network?

Ans. i] Run flask with hostname = '0.0.0.0'

ii] Find Raspberry Pi's address : hostname -I

iii] Access web server from another device.

http://<Raspberry-Pi-IP>:5000

References

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