Senior Design
ENG EC 463

Memo

EcoSwitch

To: Alan Pisano

From: Michelle Thevenin, Jiawei Liao, Keven DeOliveira, Samarah Uriarte, Michael Harkess

Team: 31 - EcoSwitch

Date: 11/16/21

Subject: First Prototype Testing

1.0 Required Materials

Hardware:

- Microprocessor: ESP32-WROOM-32
- A computer (to act as the power adapter and server)
- DHT11 Temperature and Humidity Sensor

Software:

- Mobile application using React Native, Node.js
- Server: XAMPP with MySQL and Apache 2.4.51
- Arduino IDE

2.0 Setup

The setup is divided into hardware and software: the former consists of the ESP32 microcontroller and the DHT11 temperature and humidity sensor while the latter consists of the mobile app and computer server. The DHT11 sensor detects the room temperature and humidity levels and sends 3.3 V signals to the ESP32 via pin D4 (GPIO 4). The ESP32 program then sends this data to the server that is running on the computer. Upon reaching the server, the data is stored in the MySQL database and the mobile app then queries for this data, developed with React Native on the phone to display the values on the interface.

2.1 Pre-Testing Setup Procedure

Server side:

1. Ensure that the server is not running.
2. Run XAMPP with admin privileges.
3. Activate Apache and MySQL.

Mobile app side:

1. Start the temporary backend service and Expo project.
    a. Ensure IP address is properly configured.

2. Open the EcoSwitch app using Expo (Expo Go if on mobile) and proceed through the login page.
3. Press the “Update” button to update the temperature and humidity values with the most recent entry.
    a. Alternatively, wait 30 seconds for automatic updates.

ESP32 side:

1. Ensure the ESP32 is fully connected to the computer.
2. Open the Arduino IDE and navigate to the appropriate directory.
3. Upload script.

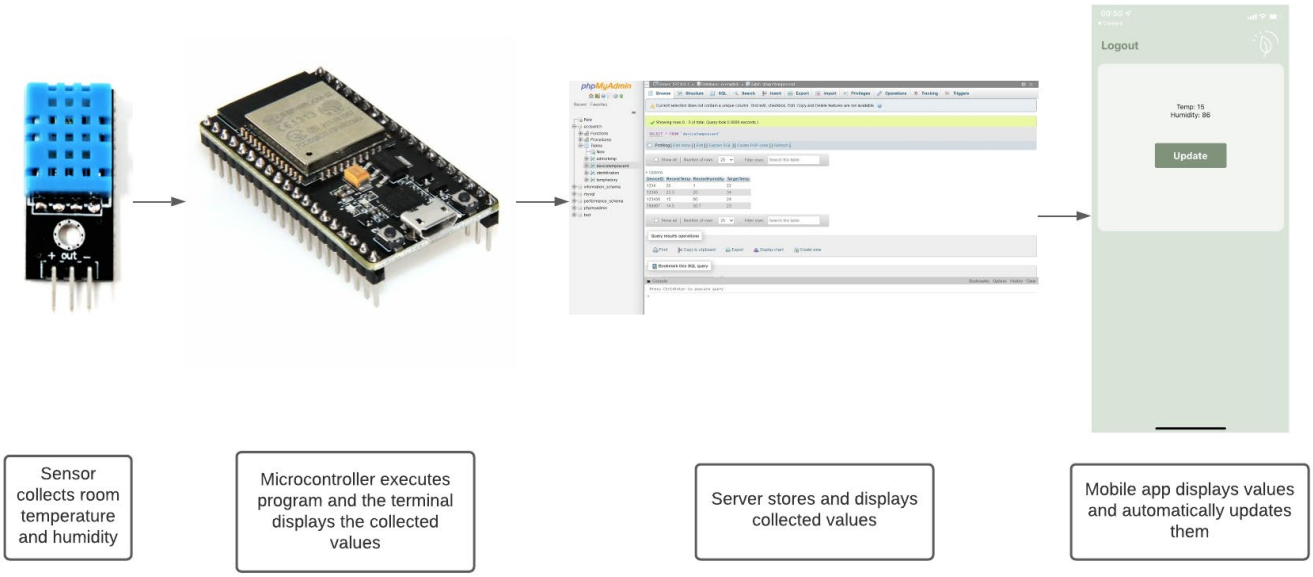

Figure 1: Illustration of Setup and Process flow

2.2 Testing Procedure

1. Flash and monitor the ESP32 program.
2. Observe the terminal output consisting of room temperatures and humidity values.
3. Observe the room temperature and humidity value displayed in the mobile app and database.
4. Ensure the values detailed in steps 2 and 3 match.
5. Ensure the room temperature and humidity values successfully update every half-minute.
    a. Note: This will be changed to automatically update every minute in the final prototype.

2.3 Measurable Criteria

The criteria for successful running and output is as follows:

I. The microcontroller should successfully communicate with the sensor to read in accurate data.
II. The server should successfully receive sensor data from the microcontroller.
III. The values should be stored in the database.

IV. The mobile app should successfully query for server data consisting of the temperature and humidity values.
V. The mobile app should successfully display these values through its interface.
VI. The displayed room temperature and humidity value should match the collected data.
VII. The room temperature and humidity values should successfully update every half-minute.
    A. Note: this will be changed to every minute in the final prototype.

3.0 Score Sheet

Minutes Passed	**½**	**1**	**1 ½**	**2**	**2 ½**	**3**
Displayed Room Temperature in Terminal (°C)						
Displayed Room Humidity in Terminal (%)						
Collected Room Temperature (°C)						
Collected Humidity Value (%)						
In the Database? (Y or N)						
Same? (Y or N)						