enior Design
ENG EC 463

Memo

EcoSwitch

To: Alan Pisano

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

Team: 31 - EcoSwitch

Date: 4/1/22

Subject: Final Prototype Testing

1.0 Required Materials

Hardware:

- Microcontroller: ESP32-WROOM-32
- A computer (to show server data)
- DHT22 Temperature and Humidity Sensor
- 12V 1.5A Stepper Motor
- 12V Battery Pack
- L7805 5V Voltage Regulator
- L298N Motor Driver
- 3D printed casing
- 3D printed attachment

Software:

- Mobile application: React Native, Node.js
- Website: ReactJS, Node.js
- Server: Amazon EC2 running Linux (Debian) and Apache 2.4.51
- Database: Amazon RDS (MySQL, MariaDB)
- Arduino IDE

2.0 Setup

The setup is divided into hardware and software: the hardware consists of the ESP32 microcontroller, the DHT22 temperature and humidity sensor, the L298N Motor driver, a 12V stepper motor. The software consists of the mobile app, website, and server and database.

The DHT22 sensor detects the room temperature and humidity levels and sends signals to the ESP32 via pin D13 (GPIO 13). The ESP32 program then sends this data to the server that is running on a local computer. Upon reaching the server, the data is recorded in Amazon RDS (in a MySQL database) and the mobile app then queries for this data on a local phone to display the values through the interface. The website also does this, along with displaying additional test devices and their corresponding data, and it offers a registration functionality that adds and removes student information to and

from a device user database when entered by the user as well. Additionally, the website provides an interface for turning off all the devices in one location. When the temperature read from the sensor is out of range with the temperature input from the mobile app, the motor will slightly turn, which would allow it to switch an FCU dial in practice.

2.1 Pre-Testing Setup Procedure

Server:

1. Log into AWS Console.
2. Navigate to the RDS Console and ensure MySQL database instance is running.
3. Navigate to EC2 Console and ensure the (Apache) server instance is running.

Mobile app:

1. Launch application using Expo (*expo start*).
2. Open application on mobile device (or laptop if WiFi is unreliable).
3. Proceed to the main page.

Website:

1. Launch project using npm (*npm start*).
2. Navigate website to use the available functionalities.

ESP32:

1. Ensure the ESP32 is fully connected to the computer.
2. Open the Arduino IDE and navigate to the appropriate directory.
3. Upload and run the script *DHT22_with_L298N_driver_test_circuit.ino*.

Hardware:

1. Attach the motor to the 3d printed attachment and put it inside the casing.
2. Add rubber strips around the FCU switch and align the attachment to the switch.
3. Align and attach the casing to the FCU with the screws on the side.

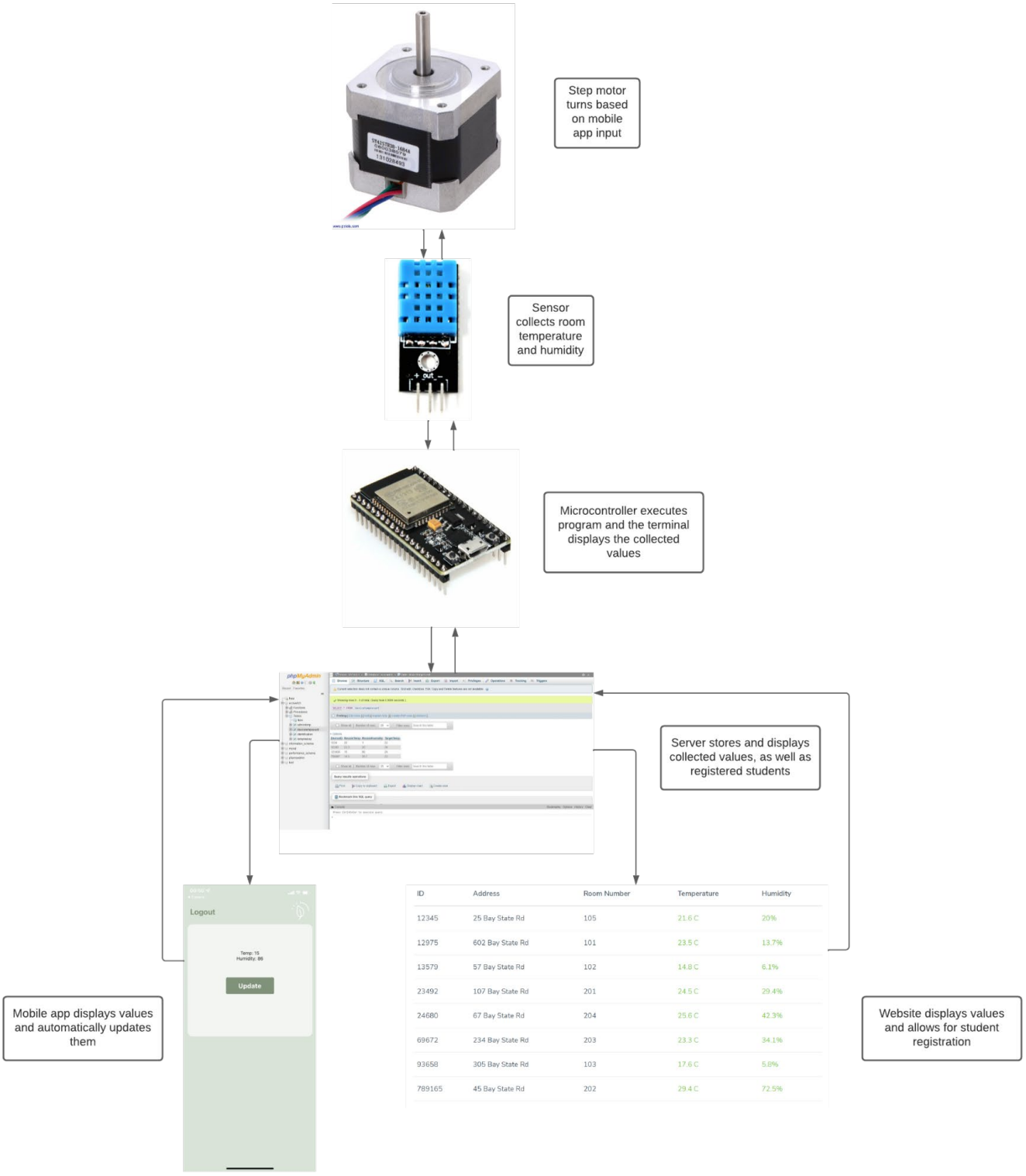

Figure 1: Illustration of Setup and Process flow

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 temperatures and humidity values displayed in the database.
4. Observe the room temperature and humidity value displayed in the mobile app.
5. Observe the room temperatures and humidity values displayed in the website.
6. Ensure the appropriate values match.
7. Register a test student using the website's "Add a Student" feature.
8. Observe the test student information in the database and ensure it matches what was entered in Step 7.
9. Remove a test student using the website's "Remove a Student" feature.
10. Observe that the test student from step 9 was removed from the database correctly.
11. Turn off all the devices in a single location using the website's "Device Adjustment" page.
12. Observe the state in the database for the location entered in step 11 and ensure it is "0".
13. Adjust the device temperature using the mobile application.
14. Observe the corresponding device information in the database and ensure it matches what was entered in Step 9.

2.3 Measurable Criteria

The criteria for successful running and output is as follows:

I. The mobile application successfully displays desired temperatures from the database.
II. The website successfully displays data from multiple devices.
III. The website successfully sends test student information from the correct database.
IV. The website successfully removes test student information from the correct database.
V. The website successfully changes the state of a user-entered location to "0", or OFF.
VI. The server successfully receives data from the mobile device.
VII. The values are stored in the database when received from the ESP32.
VIII. The ESP32 successfully retrieves the desired temperature data from the server.
IX. The ESP32 shows that the temperature was received and that the data matches what was sent.
X. The actuator operates accordingly.

3.0 Score Sheet

Action	Successful? (Y/N)
Did the mobile app successfully display the data read from the device?	
Did the server successfully receive data when sent from the mobile app?	

Are the data values stored in the database?	
Is the ESP32 receiving the input temperature data from the server?	
Did the motor turn based on the input temperature?	
Did the website successfully display the data read from the device, as well as the hardcoded data used for testing?	
Did the website successfully add a test student to the corresponding database?	
Did the website successfully remove a test student from the corresponding database?	
Did the website successfully turn off the devices registered under one location by setting their states to "0"?	

4.0 Hardware Pinout

ESP32 Pin #	Usage/Description
VIN	5V voltage supply->L298N Vcc, DHT22 Vcc, Vout of voltage regulator
GND	Ground->L298N GND, GND of voltage regulator, DHT22, two capacitors
27	Data transmit->L298N In1
26	Data transmit->L298N In2
25	Data transmit->L298N In3
33	Data transmit->L298N In4
13	Data transmit->DHT22 data pin