Boston University
Electrical & Computer Engineering
EC464 Capstone Senior Design Project

User's Manual

EcoSwitch

Submitted to

Harshavardhan Ogoti
State Street Global Advisors
1 Iron St, Boston MA 02210
hogoti@bu.edu

by

Team 31
Team EcoSwitch

Team Members

Keven DeOliveira kevend@bu.edu
Samarah Uriarte samcu@bu.edu
Michael Harkess mharkess@bu.edu
Michelle Thevenin thevenin@bu.edu
Jiawei Liao liaophie@bu.edu

Submitted: 4/18/2022

1. EcoSwitch

2. Table of Contents

2.

3. Executive Summary

Manually controlled FCUs (fan coil units) in BU brownstone dorms lead to rooms becoming uncomfortably warm due to its inefficiency. As a result, students either turn off their heating unit or open a window in order to regulate their dormitory conditions. These methods are either inconvenient to the student or waste energy. As a solution, we propose EcoSwitch, a system consisting of a smart device networked with online portals to allow students and administrators to monitor and adjust the temperatures of these rooms. EcoSwitch will establish applications that can help Boston University achieve its Zero Waste goals while increasing the students' quality of life.

1 Introduction

Many of the brownstones on the Boston University campus have a heating system that is a manually controlled fan coil unit, also known as an FCU. The FCU works by heating up air with a coil and circulating that air in the room with a fan. It is controlled through a dial on the FCU unit with simple settings such as low, medium, high, and off. These preset settings cause students to switch their units on and off or open a window to release some of the overgenerated heat. The constant switching becomes an inconvenience to students and a waste of energy for the university.

The device's hardware requirements include safe mounting on the FCU, a swappable attachment that connects to the dial, and tamper-proof features so that it remains in the dorm. On the software side, students should be able to monitor and change their dorm room temperatures on a cross platform mobile app, and building maintenance should have access to an administrator website that they can use to monitor all dorms through the EcoSwitch devices.

As a solution to these issues, we proposed EcoSwitch. EcoSwitch is a smart device that allows for user interaction through a mobile app and website, features that are typical to smart home systems. The device controls room temperatures based on user specifications submitted through the app for desired temperatures. The mobile application displays the temperature of the dorms to both students and housing administrators. The website, which is for the latter, allows them to monitor all units. These features will allow our system to create the most comfortable temperatures for students through a convenient and efficient process which will ultimately reduce energy waste. EcoSwitch will help BU reach their mission to achieve zero waste by 2030. Our solution is cost-effective and compatible with the existing heating infrastructure in the brownstones.

EcoSwitch resolves the aforementioned problems by automating the process of changing dorm temperatures. By doing so, EcoSwitch can keep the temperature of a room consistent. When a user inputs a temperature, EcoSwitch will turn the switch of the FCU to keep the temperature as close to that value as possible. If the student is comfortable, there will no longer be a need to open a window or change the temperature as dorm conditions inevitably drift away from that preferred temperature. EcoSwitch also takes humidity and local weather conditions into account to ensure that the room is within an acceptable range of the user's chosen temperature.

EcoSwitch will also determine if an FCU is failing in a particular dormitory. Administrators will be able to observe this from the temperature and humidity trends shown in the graph within the website. It can therefore inform facility workers of the issue immediately.

Administrators will be able to add students to the appropriate EcoSwitch devices based on the students' housing assignments, as well as remove them from those residences once the students have moved out. New residents will have to simply sign up on the app and immediately be able to connect to their devices upon move in.

The remainder of this document includes information on the system, installation, operation, and safety of the EcoSwitch device. It is worth noting that there were technical issues with the device's ability to rotate the FCU dial due to the high torque that the dial requires to be turned. The motor we currently have integrated into the circuit cannot overcome this torque, but the rest of the system and the logic behind the device's operation works as intended.

2 System Overview and Installation

This section will cover the overall structure of the EcoSwitch system as well as how to set up each of its main components.

2.1 Overview Block Diagram

The EcoSwitch system can be generalized into three main components: the EcoSwitch device, the administrator website, and the student mobile app. Figure 2.1.1 contains the block diagram for the overall system, demonstrating how the aforementioned components communicate and interact with each other. While sensor data is being fed to both the administrator website and the student mobile app, it is worth noting that the latter will only have access to the data that is being retrieved from their residence's corresponding device.

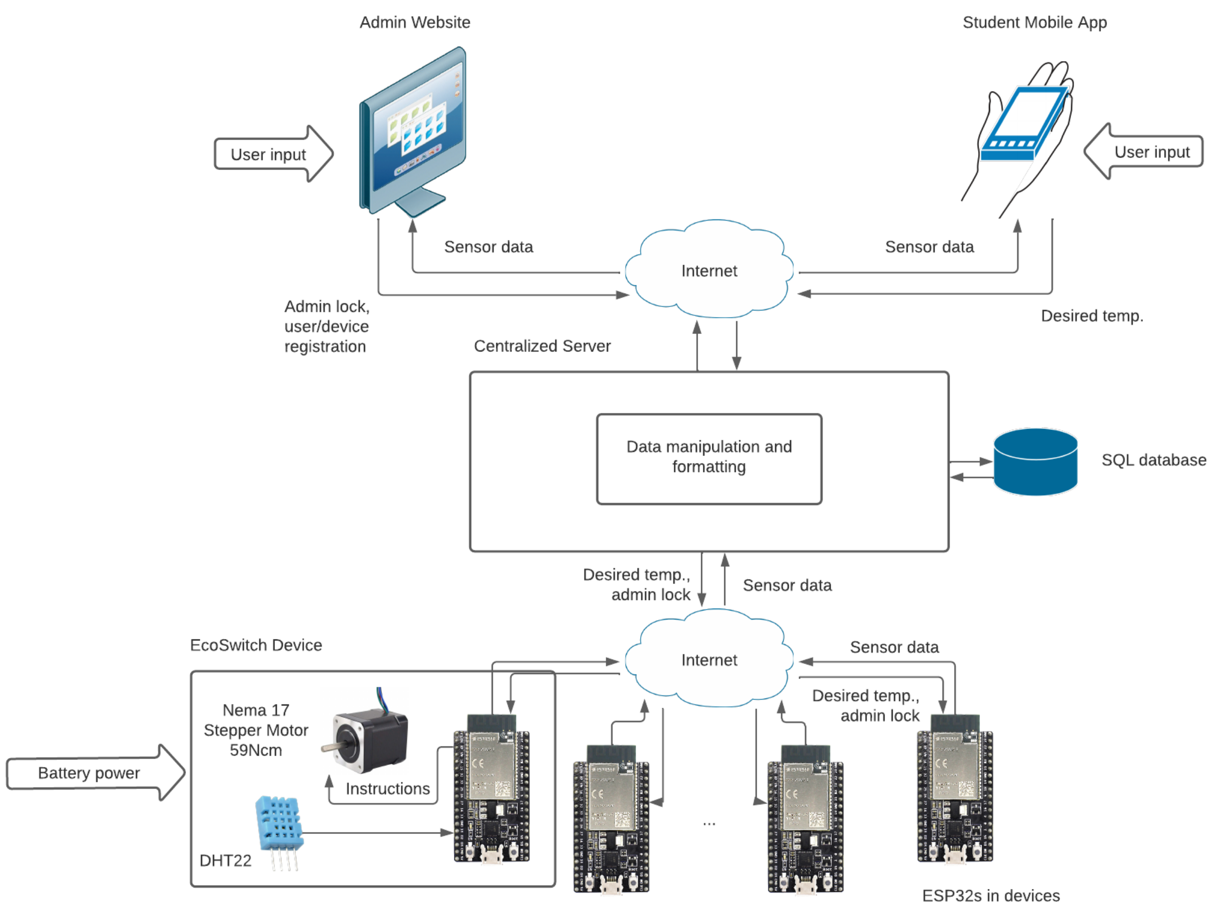

Figure 2.1.1 High-level block diagram of the entire EcoSwitch system, including the device, mobile application, and website. Some electrical components are left out for simplicity.

The EcoSwitch device's logic is available in greater detail in Figure 2.1.2. We have not included the conversion to actuator instructions in this block diagram because it takes place in the ESP32 script rather than in the server.

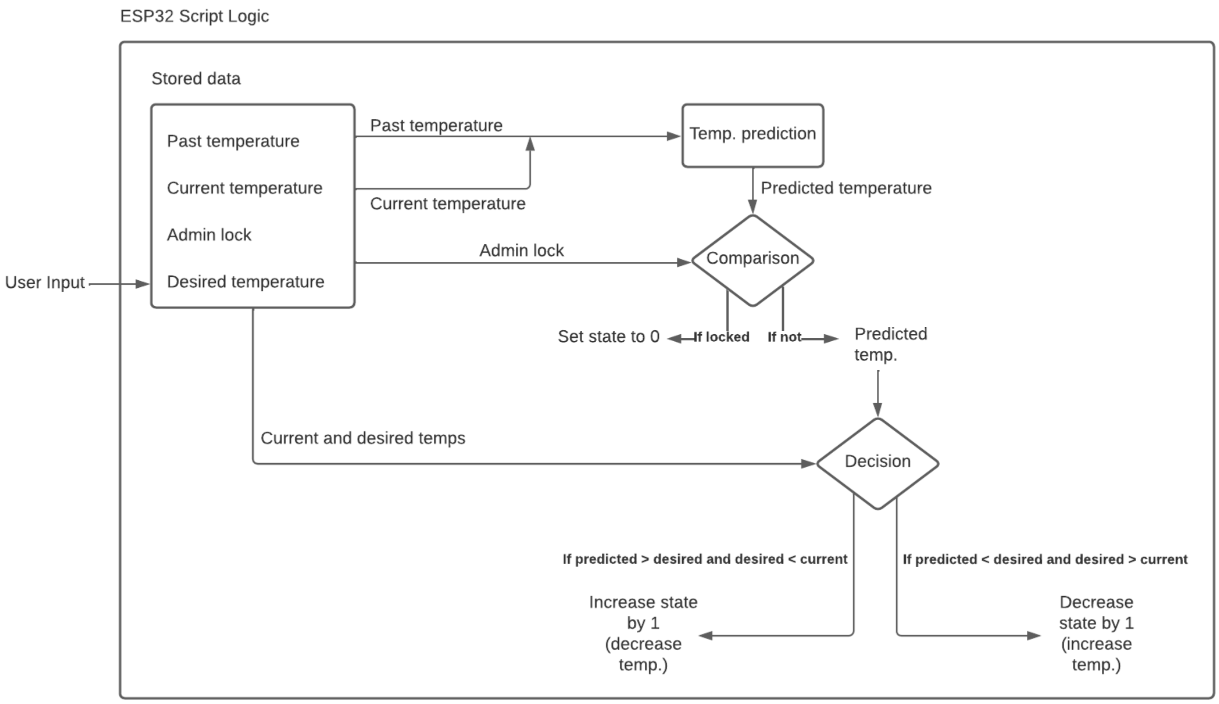

Figure 2.1.2 The EcoSwitch device's block diagram, illustrating the logic that will be operating on all devices and connecting them to their users' input.

The admin lock input is set by the housing administrator through the "Device Adjustment" page on the website. If an administrator indicates all the FCUs in a building should be turned off by submitting the appropriate form, students cannot overwrite that decision by setting the desired temperature through the app. Furthermore, temperature predictions are calculated in this script with equation 2.1.1, where the change in temperature is evaluated by subtracting a previous temperature from the current one.

$$\text{prediction} = \text{current} + (\Delta\text{temp}) \cdot \text{minutes} \tag{2.1.1}$$

The device's logic is based on an emulation of the settings on the FCU dial, which are enumerated by states 0 through 3. State 0 is off, 1 is high, 2 is medium, and 3 is low. Additionally, the dial has restricted movement, as it cannot turn completely around and must instead move through the existing states to reach another one. We have taken this into account in our design, and the simplified state machine for this available movement is shown in Figure 2.1.3.

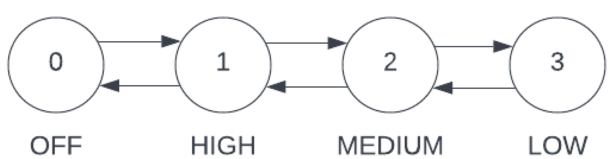

Figure 2.1.3 This is the only way the EcoSwitch device can move between states.

2.2 User Interface

The UI includes the administrator website as well as the mobile application to be used by students. Figure 2.2.1 shows the three pages of the mobile app: the login page, the main page, and the credits page, respectively. Figure 2.2.2 shows the dashboard of the admin website. This website will allow administrators to view information on all installed EcoSwitch devices, such as device ID, location, and data.

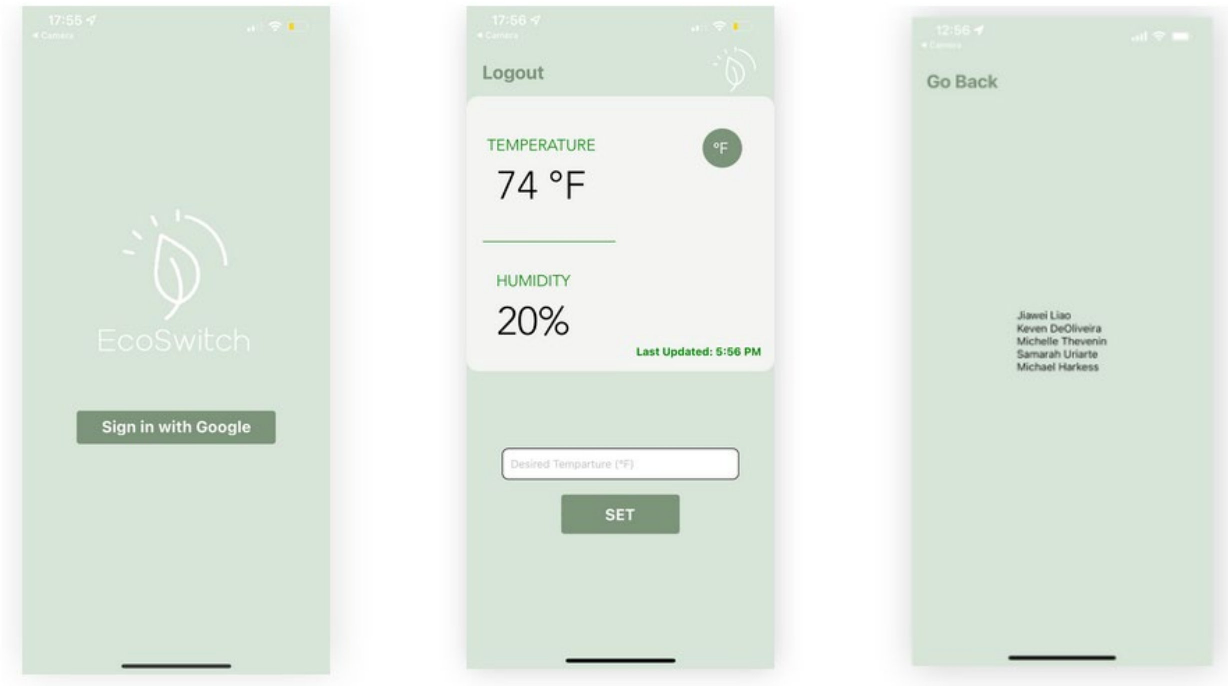

Figure 2.2.1 The mobile application GUI.

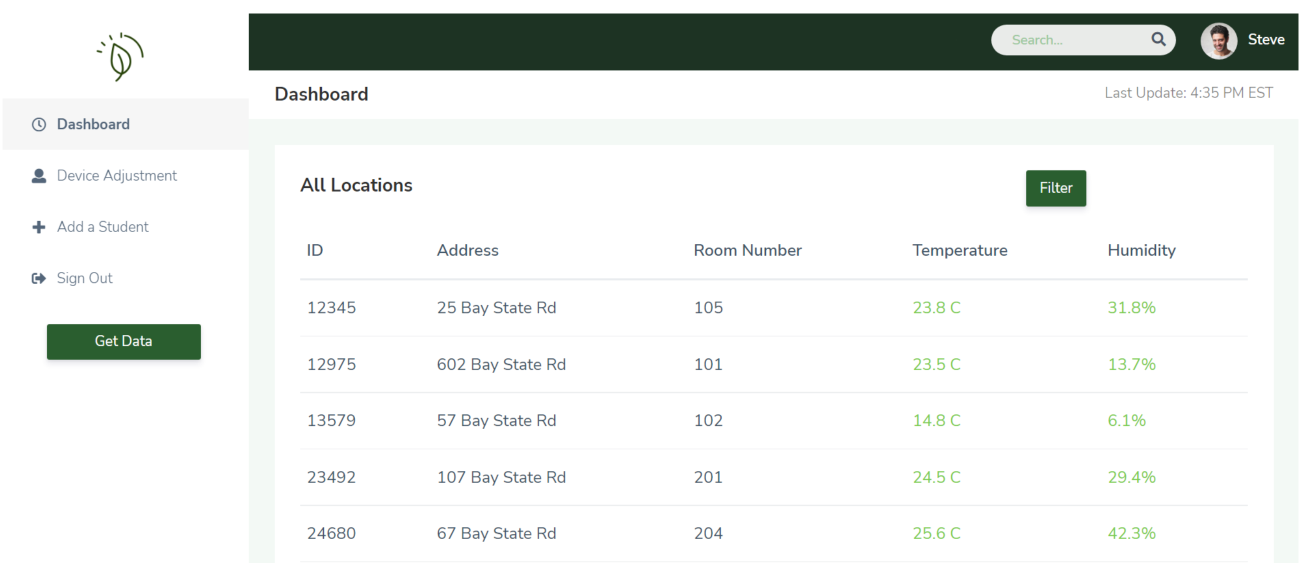

Figure 2.2.2 The dashboard page of the admin website.

2.3 *Physical Description*

The physical EcoSwitch device consists of a circuit connected to the motor, an attachment, and an outer shell. The motor directly connects to the attachment to turn the dial switch while the other components of the circuit rest inside the outer shell. The outer shell is connected to the FCU surface without making contact with the dial. This design allows the motor and connected attachment to turn freely while providing stability to the overall device.

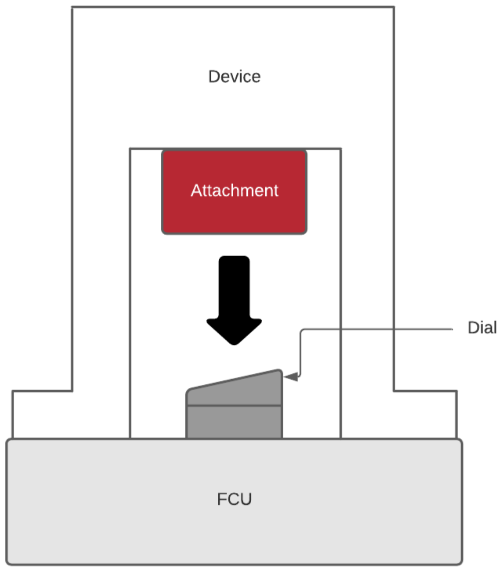

Figure 2.3.1 The EcoSwitch concept that illustrates how the device attaches to the FCU dial switch.

The EcoSwitch circuit consists of an ESP32 microcontroller, a DTH22 temperature and humidity sensor, a L298N motor driver, a Nema 17 stepper motor, an LCD screen, and a 12V battery pack with 8 AA alkaline batteries. It also includes a 5V regulator, a 10 kΩ resistor, an 100 nF ceramic capacitor, and a 1uF electrolytic capacitor. Figure 2.3.2 demonstrates how the circuit is put together.

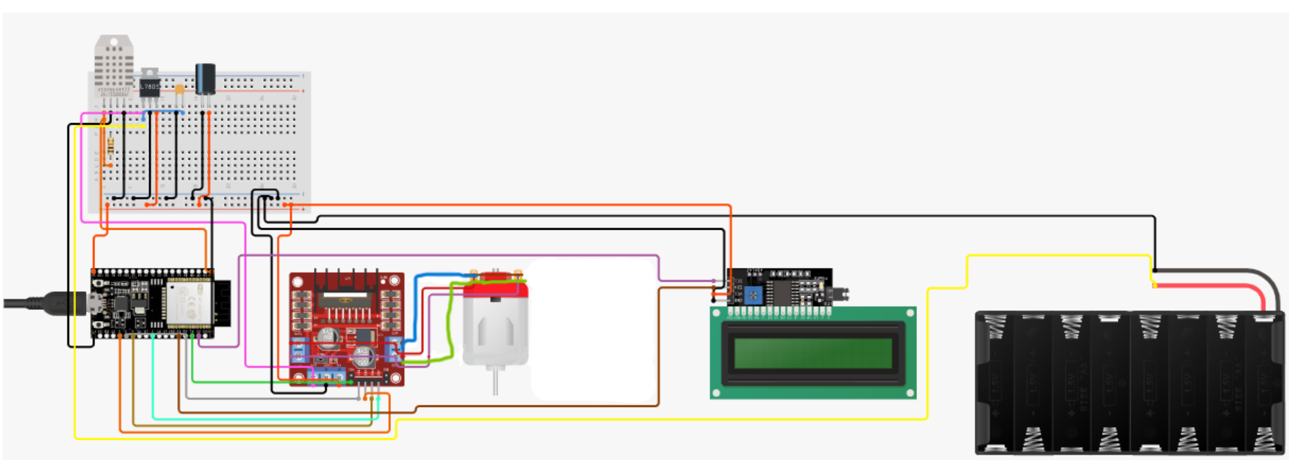

Figure 2.3.2 The circuit diagram of the EcoSwitch device.

The bottom of the attachment is specifically designed to fit on top of the FCU switch in order to turn the dial with the motor. The hole at the top of the attachment is fit to the diameter of the motor shaft.

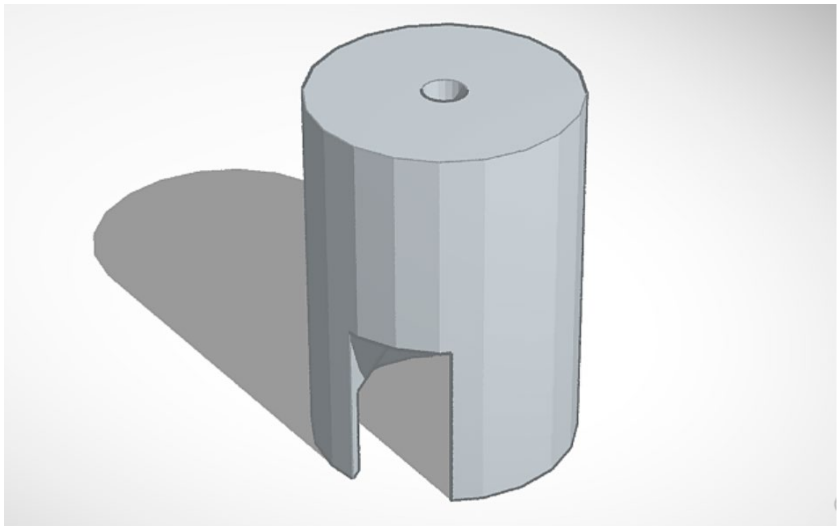

Figure 2.3.3 The FCU switch attachment.

The outer shell of the device is used to apply a downward force on the motor so that it stands still enough to apply the full torque onto the switch.

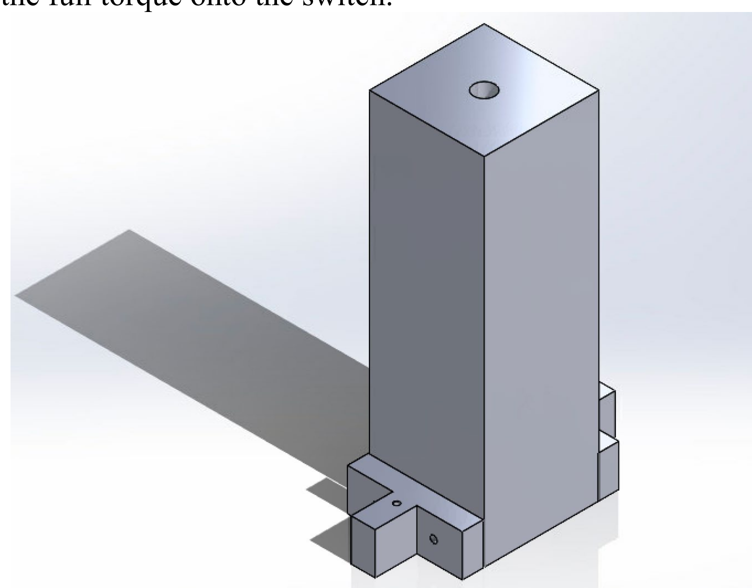

Figure 2.3.4 The EcoSwitch device's outer shell.

2.4 Installation, Setup, and Support

Each EcoSwitch device will come prebuilt with a motor, an attachment, a circuit board within the rectangular outer shell, and a battery pack. The user needs to mount the device on the FCU dial, making sure that the attachment is securely attached to the knob. Then, the user needs to put batteries in the battery pack to power the device and assemble it as follows:

1. Put two T-shape bars beside the rectangle box.
2. Use four M4x20 mm screws and drill them inside the two holes of the T-shape bars. Also, fix the M4 screw locks at the end to stabilize the screws.
3. Use two M3x20 mm screws to drill on the top hole of the two T-shape bars.
4. Fix the M3 screw locks at the end to stabilize the screws.

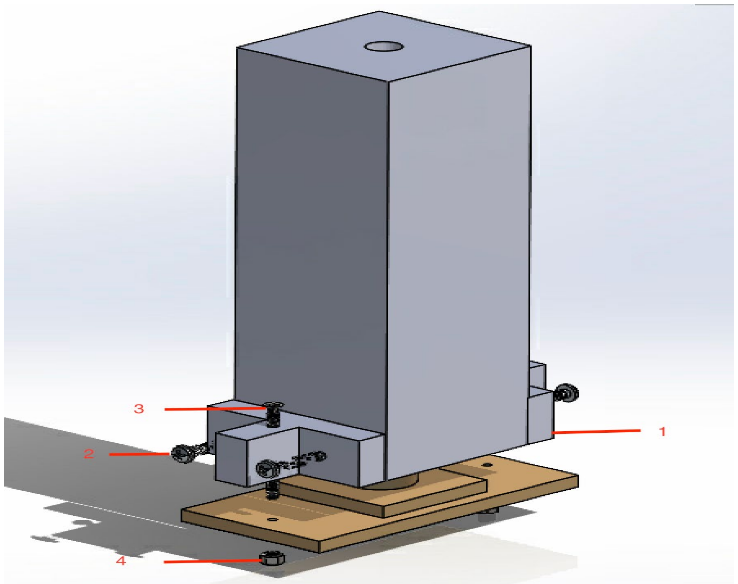

Figure 2.4.1 The installation diagram in assembly.

To maximize the functionality of the EcoSwitch system, the user should consider several factors. The EcoSwitch device is an IoT device that requires internet connectivity. Therefore, it would be ideal to have it positioned at a location with a stable WiFi connection. Also, the EcoSwitch device contains a temperature and humidity sensor inside the insulator, as well as other electrical components. Consequently, the users should not bring any external heat or water source near the device. For example, if the user wants the EcoSwitch device in the kitchen, it should not be right next to the sink in case of a leakage or other emergencies that may cause the device to malfunction.

Setup:

1. For facility workers: register the device through the website by scanning the barcode contained in the packaging and entering the requested information. We are unable to implement this portion of the setup process because it requires a larger scale of production; however, it is included to provide a better understanding of the overall product. Then, assemble and install the EcoSwitch device onto the FCU dial and surface.
2. For students: download the EcoSwitch mobile application from the appropriate app store and sign up using a Google account.
3. For administrators: navigate to the website and sign up using a Google account or an email and password.

Support:

The user might find EcoSwitch intimidating due to the sensors and motors inside the device. Therefore, we provide an interactive mobile application and website. The mobile application allows the user to see the current room temperature and humidity levels and the battery level. The website also offers this functionality, except it includes this data for all EcoSwitch devices, rather than just one.

3 Operation of the Project

This section includes details on the operation of the EcoSwitch system under two modes: normal and abnormal. In addition, safety issues that may occur during either of these aforementioned operations will be discussed.

3.1 Operating Mode 1: Normal Operation

A detailed description of the system's operating mode, as well as its consequences and results, is provided below.

a. 3.1.1 Operating Mode in Detail

We will not be implementing the components that are required to complete this procedure. This is due to our limited number of existing EcoSwitch devices. However, we have developed the process that facility workers will be able to follow to set up the EcoSwitch device and provided it in detail below:

1. Connect the device to BU WiFi by using a smartphone with Bluetooth capabilities. To do this, the user must allow the EcoSwitch device to pair with the smartphone after Bluetooth is turned on. Then, the user can simply follow the directions that are shown on the smartphone screen.
2. Register the device by using their smartphone to scan the barcode that will be available in the device's packaging.
3. Enter the location the device will be stationed at in the form that the barcode connects the user to.
4. Ensure there is no error in the verification process that attempts to check that the EcoSwitch device does not already exist in the database and that the residency is valid (ensuring it contains an FCU to control).

Otherwise, students may view the device's battery monitor or numerical display to access information about the device's battery levels or their dormitory's temperature levels, respectively.

Student users can access the mobile application functionalities, mentioned in 3.1.3 with the following steps:

1. Sign in using any Google email account through Google Authentication.
2. Gain access to their assigned device's data reading via the main menu.
3. Adjust the temperature scale.
4. Set a desired temperature range for the room.

The user can access the website functionalities, as mentioned in section 3.1.3, with the following steps:

1. Enter the URL for the website; this will bring users to the home page.
2. Navigate to the "Administrator Portal" login page.
3. Log in using their Gmail account or the email and password they have used to sign up.
4. View the dashboard and scroll down to see the graphs that show trends in data.
5. Navigate to the sidebar and click the functionality of interest.
6. For the "Device Adjustment" page, in order to turn off all the FCUs in one location, enter the address of the building and click the button below to make the request. This will lock that residence and prevent students from being able to control any devices in it. To unlock the location, navigate to the appropriate tab and enter the address of the building they wish to unlock. Then, click the button below to make the request.

7. For the “Add a Student” page, enter the student’s BU email address, residence, and room number, and then click the button below to make the request. To remove a student, repeat these steps but on the appropriate tab.
8. Click the profile icon in the upper right corner of the page to access profile information.
9. Once finished, the user can click “Sign Out.”

b. 3.1.2 User Interface

The user interface will vary depending on the role of the user: students will have access to the mobile application, and administrators, including facility workers, will have access to the website as well as the device setup interface. The former two UI options are pictured above in the *User Interface (2.2)* section of this manual.

The device setup interface is still a work in progress. Currently the setup process allows admins to set up a WiFi connection to the device, or factory reset the device.

c. 3.1.3 Normal Consequences of User Actions

For student users, the mobile application will allow the following:

1. Secure login using a valid Google account.
2. Viewing recently observed (within the past 5 minutes) temperature and humidity data.
3. The ability to set a desired temperature/temperature range.

For housing administrators, the website will allow the following:

1. Secure login using a valid Google account or an email and password.
2. View device information such as ID, location, data for every installed EcoSwitch device.
3. Turn off all the FCUs in a specified location, effectively locking it so students cannot control the EcoSwitch devices it contains.
4. Manually assign and remove students to and from devices.
5. Register and verify new devices.
    a. **Note:** This feature cannot be implemented until more Ecoswitch devices are created. The concept, however, describes our solution to guarantee an easy setup process.

Also for administrators, including facility workers, the device configuration interface will allow the following:

1. Set up WiFi connection for a device.
2. Factory reset the device.
3. Wirelessly complete setup for a device.

d. 3.1.4 Potential Abnormal Results

Due to the unstable nature of BU WiFi, the EcoSwitch device may randomly go offline. In this situation, the device will attempt to reconnect to BU WiFi until it successfully does so or until it is turned off. This process may interfere with the device’s ability to send the data it has most recently recorded to the server, resulting in the temperature and humidity database containing outdated values. Additionally, the device cannot be controlled during this interval. However, once the device is online again, it will continue to send data as expected in 5-minute intervals that are consistent with the server’s clock and users will be able to remotely change their dormitory conditions through the app. Currently, the system is still in development mode, so the temperature and humidity levels are read and uploaded to the database more frequently. The 5-minute timeframe that the device is meant to operate on will be established prior to deployment.

Additionally, as previously mentioned, the device may unexpectedly shut down if the battery runs out. If the device loses power, its data syncing process that uploads current values to the appropriate database will also be interrupted, which will result in a potentially inaccurate representation of the corresponding dormitory's temperature and humidity conditions. Users will also find that their temperature inputs will not affect the device, as they will not be able to access the device itself because it will be powered off. The device can easily be powered by replacing its batteries, which will allow it to reconnect to BU WiFi and communicate with the server again.

Finally, the website and mobile application have simple functionalities that make it unlikely for abnormal behavior to occur.

e. 3.1.5 Exiting Operating Mode

The EcoSwitch device can simply be shut off by moving the "On" switch to the "Off" setting. This will shut down the ESP32 script and turn off all the individual components in the circuit as well. However, it is worth noting that the script will automatically be exited if the device runs out of power during its normal operating mode, since it relies on battery power. The batteries must remain charged in order to shut the system down properly. Additionally, if the user wants to remove the batteries in order to charge them, the device should be turned off beforehand.

There is no need to exit or shutdown the server, as it is an AWS instance that is constantly running in the background. The server is required to stay this way in order for the EcoSwitch devices to continue updating the database with their read-in values.

Similarly, the mobile application and administrator website operate independently with a permanent backend that is combined with the aforementioned AWS instance. They are both deployed through AWS every time the corresponding directory in the project's GitHub repository is updated. Development for both applications still occur through local hosts with servers that must be manually shut down by sending a SIGINT through the terminal.

3.2 Operating Mode 2: Abnormal Operations

Describe any anticipated abnormal states (e.g. out-of-range data) and recovery. Have you explicitly built into the project operating modes like 'diagnostic' or 'self-test'? Describe how the project can enter and recover from such states. Is user intervention needed? Where can the users find help when a problem arises?

f. 3.2.1 Connection Error

1. If the EcoSwitch cannot connect to the WiFi network, restart the device and try again.
    a. On device setup, the device will go into a boot loop if the device constantly fails connecting to the specified WiFi network. If the device fails to connect to the WiFi network after multiple attempts, the network itself could be causing the issue.
2. Depending on network conditions, the EcoSwitch may not be able to connect to the internet to upload sensor data or retrieve the user's target temperature. Restarting the device can fix this issue. The device can operate without a connection; however, functionality will be extremely limited. Users will not be able to control the device from the mobile application while the device is offline.
3. If the user cannot connect to the EcoSwitch interface via the EcoSwitch WiFi network:
    a. Try refreshing the page to see if the interface will load. If this does not work, reconnect to the EcoSwitch WiFi network and then have the page reload.

g. 3.2.2 Temperature Change

1. If the device is not regulating the temperature to the target temperature:
    a. It could be an issue with the WiFi network and either the connection between the mobile app or the device itself. Try restarting the device or reconnecting to the WiFi network on your mobile device.
    b. The target temperature may be outside the acceptable temperature range that has been set for the room. In this case, adjust the temperature you have attempted to input to one that falls within a normal range.
    c. The device may not be powered due to the device being turned off or the batteries being drained. If so, turn on the device or replace the batteries.
    d. The device may be locked by an administrator. If so, contact your administrator to unlock the device.

h. 3.2.3 Power Failure

1. If the EcoSwitch suddenly loses power due to depleted batteries, replace the batteries and turn on the device. Normal operation should resume after the booting process is finished.

3.3 Safety Issues

4. The EcoSwitch device uses a lithium battery pack. Users are advised to avoid putting the battery pack or the device near fire or expose it to sunlight for a long time.
5. Users are also advised to be careful not to spill liquids near the device. Spilling liquids on the top of the EcoSwitch devices may damage the electronics in the circuit board, such as the temperature sensor or microcontroller. Consequently, please do not put the device near or in wet areas.
6. Users should also take care of the device when the driver is in an abnormal or error status, which will cause the motor to fail to operate normally. If that occurs, students should contact the facility workers or housing administrators to follow the provided solutions above for maintenance.

4 Technical Background

This section includes the technical description of the EcoSwitch system and a detailed description of the interactivity between different components of the EcoSwitch system.

a. 4.1 Hardware Component

The EcoSwitch device uses an ESP32 microcontroller and a DHT22 sensor to fulfill its hardware functionalities. An ESP32 is ideal because it supports both WiFi and Bluetooth, which was necessary to carry out our system design. Additionally, the size of the ESP32 was suitable for the sizing limitations we were given.

The main programming language used to program the EcoSwitch device was C/C++. This allowed for control of the sensor and motor, as well as to connect to the internet. The device also uses a 3D-printed attachment, which was designed to fit onto a dial so that the device may turn the dial efficiently. The DHT22 sensor is used to accurately detect the current temperature and relative humidity of the surrounding area. An insulation material will surround the sensor and other electronics, ensuring these sensor readings are accurate.

The dial on the heating unit is moved with a stepper motor. This motor is housed in a 3D-printed shell to protect the component. Additionally, it is controlled by a L298N motor driver, which is powered by a 12V power source using 8 AA batteries. The circuit includes a 5V voltage regulator as the ESP32 pushes at most a 5V signal. The circuit also consists of both a ceramic and electrolytic capacitor in order to minimize capacitor impedance across frequencies.

The ESP32 is connected to an LCD display which will provide user feedback on the current operation of the EcoSwitch. This feedback will include current temperature and humidity levels of the room, as well as other feedback regarding device setup.

b. 4.2 Software Component

The EcoSwitch operates with three core software components: frontend, backend, and API. The frontend is split into two instances: one is a website for administrators to monitor the EcoSwitch devices and another is a mobile application for students to interact with their specific devices. Both frontend components are developed in React, with React JS used for the website and React Native for the mobile app. The mobile application also uses Expo to build the application onto different platforms. The backend was written in PHP and runs on an EC2 instance on AWS (Amazon Web Services).

API requests are facilitated by the use of an HTTP or HTTPS connection to the EC2 instance on AWS. The HTTP/HTTPS request supports a TCP connection. This makes the transference of data using this connection more secure and reliable. For a POST request made using an API call, once the necessary data is retrieved from the request, the backend will parse the data, convert the data into an acceptable data type, and then update the appropriate entries in the MySQL database. For a GET request, the backend will search the MySQL database for the requested record based on the parameters passed to the API. If a record is found, the record is passed back to the requester in a JSON format. If there are any errors searching for data in the database, an error is returned to the user.

The EcoSwitch API is the primary tool that allows for communication between the ESP32, mobile application, administration website, and backend. The main purpose of this API is to facilitate the movement of data between the source and destination. The interaction between these components are:

1. The EcoSwitch device sends sensor data to the MySQL database via a POST request using the API.
2. Both the mobile application and administrator website display the data via a GET request using the API.
    a. The mobile application provides the most recent conditions of the room while the admin website presents the same information for all the EcoSwitch devices registered at the time.
3. The mobile application sends target temperatures to the MySQL database via a POST request.
4. The EcoSwitch device retrieves these target temperatures via a GET request from the database.
5. The EcoSwitch device calculates how far to turn the dial based on the target temperature, current temperature, and the current rate the temperature is changing in the room.
    a. Based on the current rate of change, the ESP32 will “predict” if the target temperature will be achieved in 10 minutes. If the temperature will not be achieved, the device will move the heater to a more extreme setting; otherwise, the setting will remain the same.
    b. This algorithm initiates automatically over a set interval.

Other functionalities that both frontend components share include a method to change the displayed temperatures between Celcius and Fahrenheit and an authentication feature for users to log into their accounts to control the EcoSwitch devices. The administrator website also has additional functionalities not previously mentioned. This section of the frontend can make a POST request using the EcoSwitch API to add authorized students to the system as well by adding new entries to the appropriate tables in the MySQL database.

5 Relevant Engineering Standards

IEEE 802.11 – Wireless Local and Metropolitan Area Networks:

This is a standard for frame-based data networks such as local area networks. The IEEE 802 architecture also includes a specification for the identification of public, private, and standard protocols. It relates to our project because BU WiFi is a LAN network and is used to connect the device, mobile app, and website.

IEEE 1 – General Principles for Temperature Limits in the Rating of Electrical Equipment and for the Evaluation of Electrical Insulation:

This standard includes all temperature indexes for materials and temperature classes for insulation systems. It relates to our project as temperature is an important factor to consider when choosing materials.

IEEE 1B – Guiding Principles for the Specification of Service Conditions in Electrical Standards

This standard focuses on the different situations regarding usual service conditions. It applies to our project because it is expected that the device will need to be serviced by maintenance regularly.

IEEE 830 – Recommended Practice for Software Requirements Specifications

This standard describes the content and qualities of a good software requirements specification. It is aimed at defining requirements of software to be developed but also aids in the selection of software products. It relates to our project as relevant software products had to be decided on.

IEEE 270 – Standard Definitions for Selected Quantities, Unites, and Related Terms, with Special Attention to the International System of Units

This standard includes definitions for physical quantities, units commonly used in technology, and related terms that concern systems of measurement. It relates to our project due to the SI units often used to describe quantities in our project.

IEEE 82079 – Standard for Preparation of Information for Use of Products

This standard gives principles and general requirements for conveying applicable information to users. Information for use is for anyone who encounters a product for the first time. It relates to our product as it is important for us to accurately describe installation and set up of the device.

6 Cost Breakdown

Project Costs for Production of Beta Version (Next Unit after Prototype)				
Item	Quantity	Description	Unit Cost	Extended Cost
1	1	ESP-WROOM-32 ESP32 ESP-32S Development Board	$10.99	~
2	1	DHT22 Temperature Humidity Sensor	$9.95	~
3	1	12V battery pack	$8.00	~
4	1	Nema 17 Stepper Motor Bipolar 2A 59Ncm	$13.99	~
5	1	L298N Motor Drive Controller Board DC Dual H-Bridge Robot Stepper Motor Control	$6.99	~
6	1	CopperTop AA Alkaline Batteries (8-ct)	$7.99	~
7	1	GeeekPi 2-Pack I2C 1602 LCD Display Module	$10.99	~
8	1	Thin Neoprene Rubber Strip 1/16	$12.59	~
9	1	Black Aluminum Heatsink for TO-220 Transistor	$8.99	~
			Beta Version – Total Cost	$90.48

The main constraint of this project was constructing a device as low-cost as possible to compete against currently available alternatives on the market. The biggest budget concern was the motor choice as it was the most expensive component and required some replacing throughout testing. Despite this, more than half of the budget remains.

7 Appendices

The EcoSwitch system's specifications will be presented in this section, as well as the academic achievements and professional plans of the individuals that are responsible for its creation.

7.1 Appendix A – Specifications

Requirements	Value, Range, Tolerance, and Units
Case/Device	• Dimensions: 55mm x 55mm x 170mm • Safe to mount and tamper-proof
Power Supply	• 12V lithium battery pack • 5V and 160-260 mA in active mode for the ESP32 • 12V and 11.5 W for the stepper motor driver
Attachment Size	• Diameter: 34 mm • Height: 48.5 mm • Securely mounted on the knob of the FCU dial
Actuator	• Nema 17 Bipolar 59 Ncm (84 oz.in) 2A 42x48 mm 4 Wires w/ 1 m Cable & Connector: • The actuator is programmed to turn the FCU dial to a specific angle to switch the mode of FCU • The actuator is able to turn the dial forward and backward
Space of Server	• CPU (2.5 GHz), 1 GiB of Memory, and 8 GB SSD storage • OS: Amazon Linux 2 (Kernel 5.10)
User Interaction	• The app periodically fetches from the server every 5 minutes • The app is in cross platform
Sensor	• DHT22 Temperature Humidity Sensor • The sensor collects data every 5 minutes

Microprocessor	• ESP-WROOM-32 • Operates with the temperature sensor and sends data to the server every 5 minutes

c.

7.2 Appendix B – Team Information

Keven DeOliveira	Keven DeOliveira will be graduating with a bachelor’s degree in computer engineering and will then go on to work as a software engineer at Hewlett Packard Enterprise.
Samarah Uriarte	In addition to her computer engineering major, Samarah Uriarte has a concentration in Machine Learning and will be interning at Red Hat following graduation.
Michael Harkess	Michael Harkess will go on to graduate school to receive a master’s degree in Computer Engineering after graduating with a bachelor’s degree in the same field.
Michelle Thevenin	Michelle Thevenin will be working as a process integration engineer for Raytheon Missiles and Defense after graduating with a bachelor’s degree in electrical engineering.
Jiawei Liao	After graduating with a bachelor’s degree in electrical engineering, Jiawei Liao will be working as a system engineer for MBTA.