

PERVASIVE SOFTWARE ENGINEERING FINAL PROJECT DOCUMENTATION



Group 6:

Alvian Wijaya (2301891595)

Christy Natalia Jusman (2301890365)

Davin Pratama Chandra (2301891090)

Jeremy Ponto (2301891525)

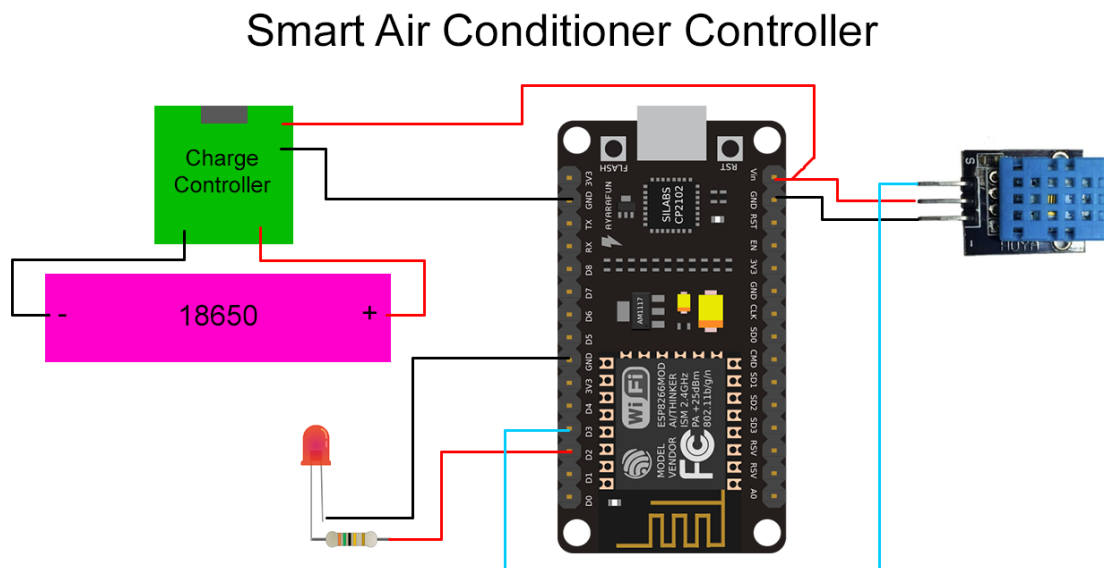
1. Product Summary

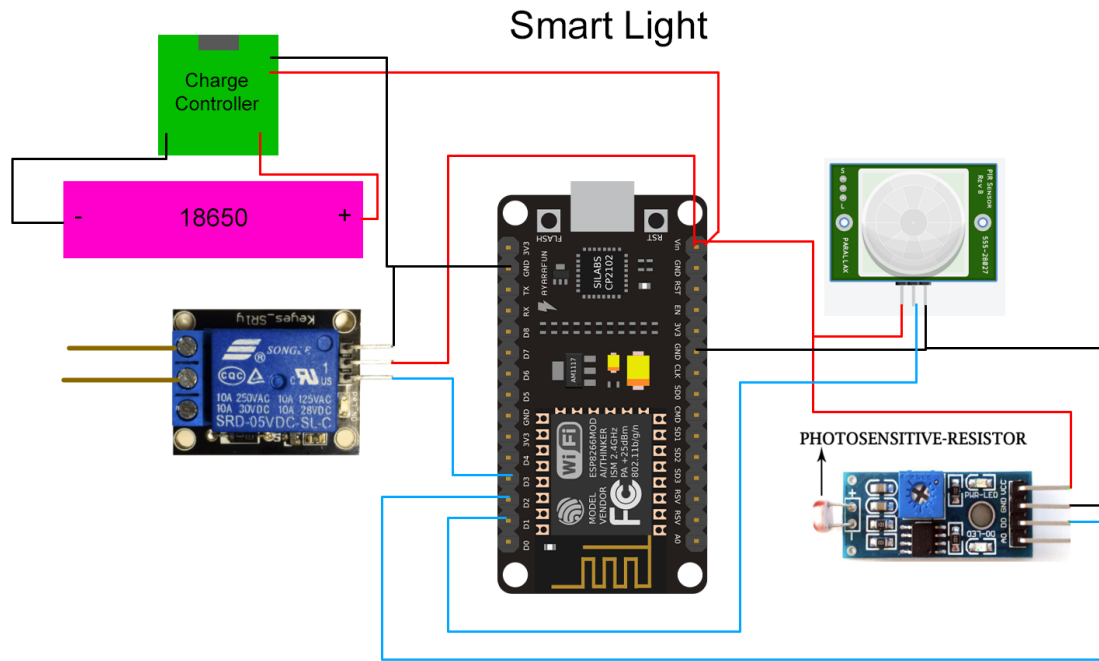
The smart home is an integrated project which consists of a light remote switch and an Air Conditioner (AC) remote switch. For the light remote switch, the motion sensor will detect the movement and a notification will be sent to the frontend through the backend so the user can decide to turn on the light or not on the web application. This also works the same way as the Air Conditioner (AC) remote switch. The temperature and humidity sensor will measure the room temperature and humidity and the user will receive a notification about this information on the frontend through the backend so the user can decide to turn on the Air Conditioner (AC) or not on the web application.

2. Components

- a. NodeMCU = x2
- b. IR Emitter = x1
- c. Humidity and Temperature Sensor = x1
- d. PCB = x2
- e. Motion Sensor = x1
- f. Light Sensor = x1
- g. Lithium Ion 18650 Cell = x2
- h. Lithium Ion Charge Controller = x2
- i. Relay = x1

3. Hardware Assembly





4. Software Development

The backend part itself is made with Express.js and MQTT broker. We created a main code file and a file dedicated to store routes. The routes file is filled with endpoints to get data from the device and send data to the device.

The endpoint /ac will return room temperature and room humidity. Then, the endpoint /lamp will return room brightness and room motion. For the room brightness, the device will send signal “0” and it will be defined as “dark”. While it sends signal “1”, it will define the room brightness as “light”. For the room motion, the value 0 itself means there is no movement in that room. However, when the value is 1, it detects movements.

For the AC, the /setacpower will send signal “0” as the value to turn off the AC. It will send “1” to turn on the AC. The /setactemp will send “x” as the AC temperature to the device where x is ranging from 18 to 32. The /setacfan will send “x” as the AC fan level to the device where x is ranging from 1 to 4. For the lamp, the /setlamppower will send signal “0” as the value to turn off the lamp. It will send “1” to turn on the lamp.

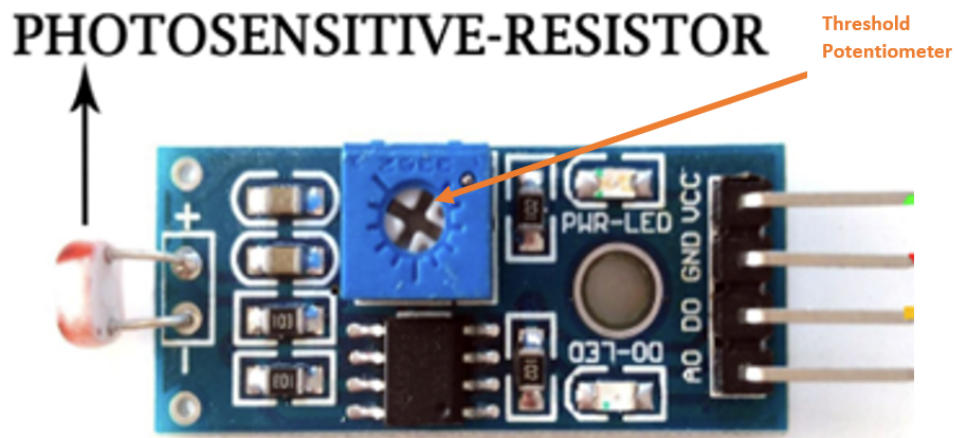
By using the API from the backend part, we can either send the request to the devices or receive the response from the devices. Therefore, to make the GUI, we use React as a library. Actually, we made many pages so the user can interact with our systems. However, since we did not have lots of time to work on the backend part, only one page can be used for this project. To make the page become more colorful and interactive, we use Bootstrap. As an example, when the cursor hovers at the top of the button, the font will become bolder.

Later on, it will be containerized by using Docker. For the frontend and backend, we have to build a docker container before deploying it to AWS. For the Mosquitto Broker, we just have to install the necessary software and open the ports to allow connection in the EC2 instance. Then, it was deployed by using 3 Amazon Web Service EC2. Each of them respectively for the Arduino, BackEnd part, and FrontEnd part.

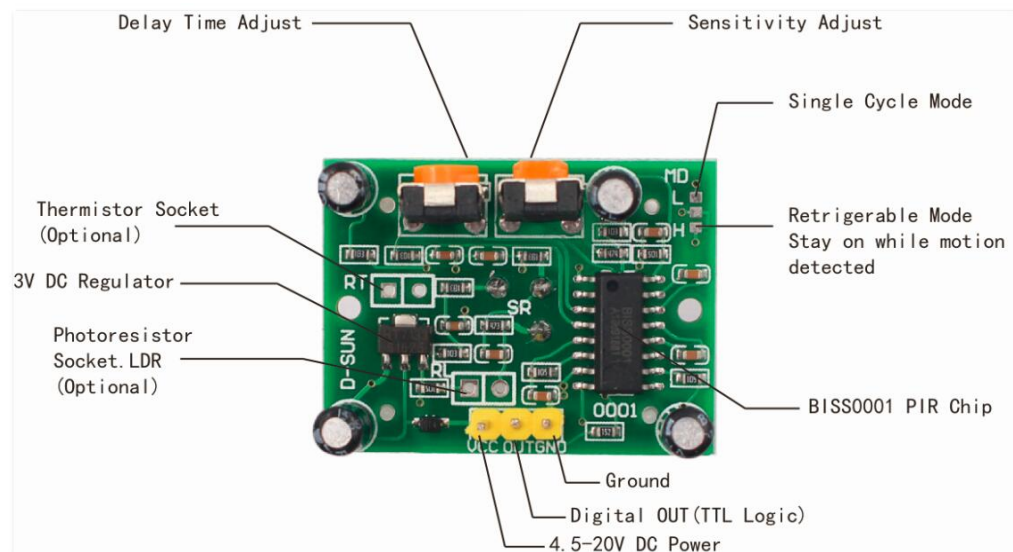
5. User Manual

User's interface:

- Go to <http://18.222.153.29> and it can be seen that there are 2 cards. The left one is for the Air Conditioner and the right one is for the lamp.
- On the left card, it will show the current temperature and humidity in that room. Therefore, if the users want to turn on the AC, they can click on the 'Turn On' button. Otherwise, if they want to turn it off, they have to click on the 'Turn Off' button. If the button color changes to blue, it means the action has been executed and successful.
- Users can also change the temperature of AC. The default temperature of the AC is 18 degrees celsius which is the lowest. The highest temperature is 33 degrees celsius. By clicking the (-) button, the temperature of the AC will be decreased. Furthermore, if (+) is clicked, the temperature will be increased. To save the current state, users can click on the 'Set' button.
- This device can also change the speed of the fan in the AC. The bigger the number of fans, the faster they become. The initial state of the fan is 1. To make it faster, users can click on the plus button, and to lower the speed, they can click on the minus button. Users must click on the 'Set' button to save it.
- The right card is for the lamp. It will show the brightness and also detect the movement in the room. The value itself will be updated every 5 seconds. To turn on the lamp, users can click on the 'Turn On' button and the 'Turn Off' button to turn off the lamp.
- To adjust the threshold between bright and dark, you can turn the potentiometer located in the light sensor.



- g. You also can adjust the sensitivity of the motion sensor by turning two potentiometers located at the side of the motion sensor. You can adjust the time needed to detect a movement and also the range of the motion detection (the sensitivity).



6. Contributors

- Alvian Wijaya : Made the hardware
- Christy Natalia Jusman : Made the frontend
- Davin Pratama Chandra : Made the backend
- Jeremy Ponto : Programmed the Arduino

7. Links

- Source Code Link:
[GitHub - alviancode/PSE_Final_Project: Pervasive Software Engineering Final Project](https://github.com/alviancode/PSE_Final_Project)
- Video Link:
<https://youtu.be/5K5PmtEraC0>