HOUSE LIGHT SENSING SYSTEM

By

Amritpaul Singh

Julia Luzinski

Rajiv Tatineni

Final Report for ECE 445, Senior Design, Spring 2020

TA: Vassily Petrov

8 May 2020

Project No. 05

Abstract

The aim of this project was to lower the amount of electricity used by a user in their home. To accomplish this, we designed a system that allows a user to remotely monitor and control their home lights. Our system consists of four parts: a server, light modules, relay modules, and a LED board. These components allow the user to know exactly what lights are on in the home via the light modules and allows them to toggle lights on or off via the LED board and relay module. The original project sought to accomplish the same goal by using a coat hanger that would turn off each connected light when the hanger was not holding an item. This was accomplished via specialized wireless light switches that took input from the hanger. Our solution gives the user more control over specific lights and is cheaper when the amount of lights controlled is large.

Contents

1 Introduction **4**
1.1 Problem Statement 4
1.2 Solution 4
1.3 High-Level Requirements 5
1.5 Block Diagram 6
2 Second Project Implementation **7**
2.1 Implementation of Arduino Code to Connect to IOT Cloud 7
2.2 Power Supply 8
2.3 Light Module 9
2.3.1 Ambient Light Sensor 10
2.4 Relay Module 11
2.4.1 Tolako 5V Relay 11
2.4.1.1 Power Draw 11
2.4.1.2 Longevity 12
2.5 Tolerance Analysis 13
2.5.1 Light Sensing Accuracy 13
2.5.2 Window Casing Calculation 14
3 Conclusion **16**
3.1 Implementation Summary 16
3.1.1 Arduino code to connect to IOT cloud. 16
3.1.2 Power Supply 16
3.1.3 Light Module 16
3.1.4 Relay 16
3.2 Unknowns, Uncertainties, Testing Needed 16
3.2.1 Ambient Light Sensor Verification 17
3.3 Ethics and Safety 17
3.3.1 Ethics 17
3.3.2 Safety 17
3.4 Project Improvements 17
4 Progress on First Project **19**
4.1 Main Circuit PCB 19
5 References **20**
Appendix A **21**

1 Introduction

1.1 Problem Statement

It is very easy to accidentally leave a light on in the home when you are either going outside or going to sleep. Whenever this occurs, a person's electricity bill can rise by a large amount depending on the amount of lights that were kept on and their duration. The root cause of this problem is that it is difficult for a user to know exactly what lights are on in their home and it can be annoying to turn off unnecessary lights.

We decided to work on this type of project due to how expensive many existing solutions are for monitoring light fixtures. For this project we knew that we could make a cheap light monitoring system when compared to current offerings.

1.2 Solution

In order to solve the above problem we will utilize 4 different subsystems: a ec2 web server, light modules, relay modules, and a LED board. The first subsystem is an Amazon ec2 server that is used to store the reading of the light modules and the position of the switches on the LED board. We wrote a python script that reads the data from the server and determines whether a light is on and if a light in the home needs to be turned off or on. The script uses the data from the light modules to determine what LEDs need to be illuminated on the LED board, and it uses the status of the switches on the LED board to toggle the specific light fixtures' relay. The second subsystem, the light module, is responsible for monitoring whether a light is on via an ambient light sensor. The light modules are each connected to their own wifi module so that they can transmit their readings to the server. The relay module is used to turn on or off a specific light fixture and it gets connected in series with the power to the light fixture. This module has a wired connection to the light module's atmega328p IC and is activated when someone presses a switch on the LED board. The last subsystem, the LED board, displays to the user what lights are on in the home via LEDs and it also has switches which lets the user control certain lights. The LED board like the light modules is equipped with a wifi module that allows for its data to be sent to the server. With our solution a user will be able to save money because they will know the status of each of their lights and they can turn each one off remotely.

The past project used a coat hanger to turn off every connected light when the hanger is no longer holding an item. Our solution varies in the sense that we are using a board that tells the user exactly what lights are on in the home via LEDs and labels, which the user would have to make. This board also contains switches that allows the user to toggle on/off that certain light. The past project used a Z-wave hub which then controlled Z-wave switches in order to automate the light fixtures. In our project we are using AWS which stores the status of each light module, via the wifi module in each light module, and uses this to transmit to the LED board. The AWS system for the project also takes in data from the LED board, via the wifi module in the board, on what switch was pressed and it will use this to turn off that particular light fixture. Essentially the past project sought to limit light usage by having a main kill switch for each connected light while we have decided to have independent control over each light that is connected to our light module and relay.

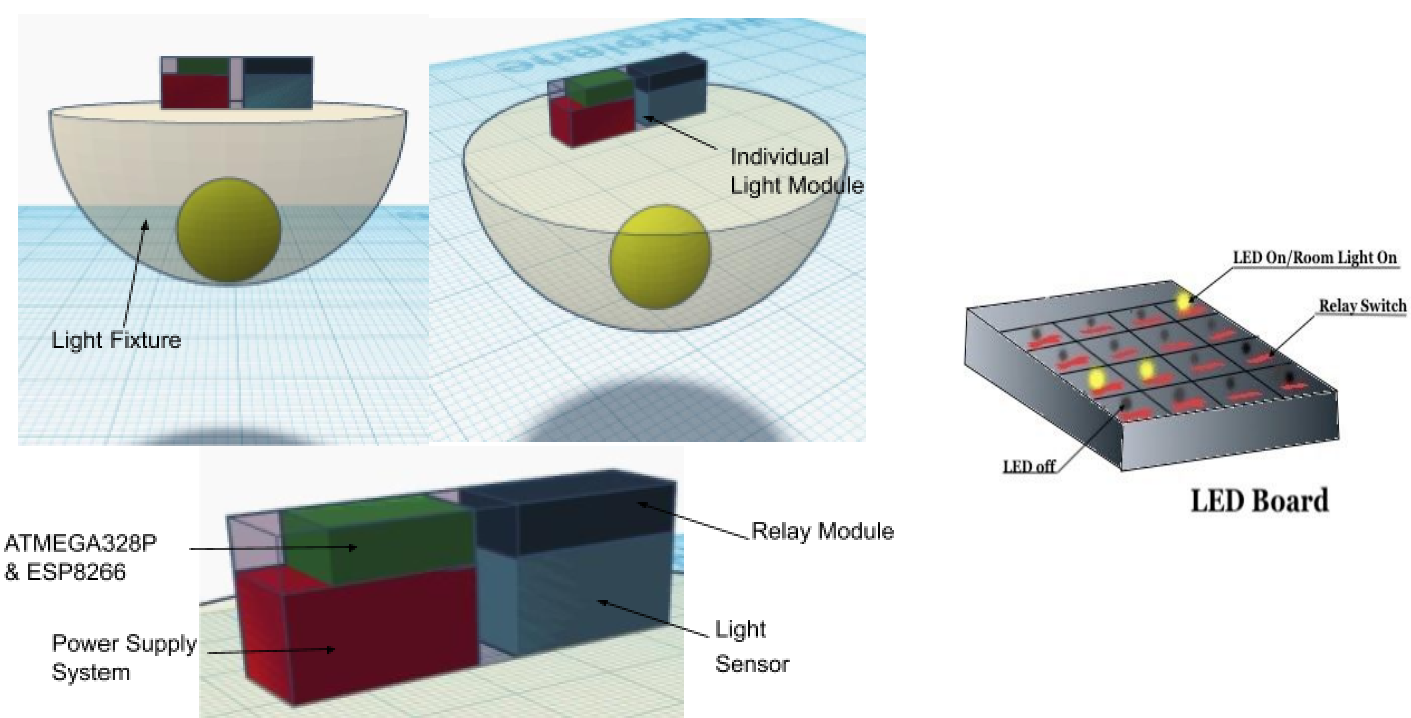
Numerous different brands of smart lights are on the market and each one can be monitored and controlled wirelessly by the user. The problem with this solution is that the bulbs are very expensive; for example, the starter kit for the Philips Hue bulbs which contains 4 lights has a price of 200 dollars [1]. Very quickly buying these bulbs and making an entire house's light system 'smart' can cost a lot of money. Our solution will allow for the user to monitor and control their lights at a much cheaper price. Along with this unlike smart bulbs, our system is not tied to the bulb so if the bulb dies then the user can just buy a cheap bulb to replace the dead one without having to worry about wireless monitoring and control.

1.3 High-Level Requirements

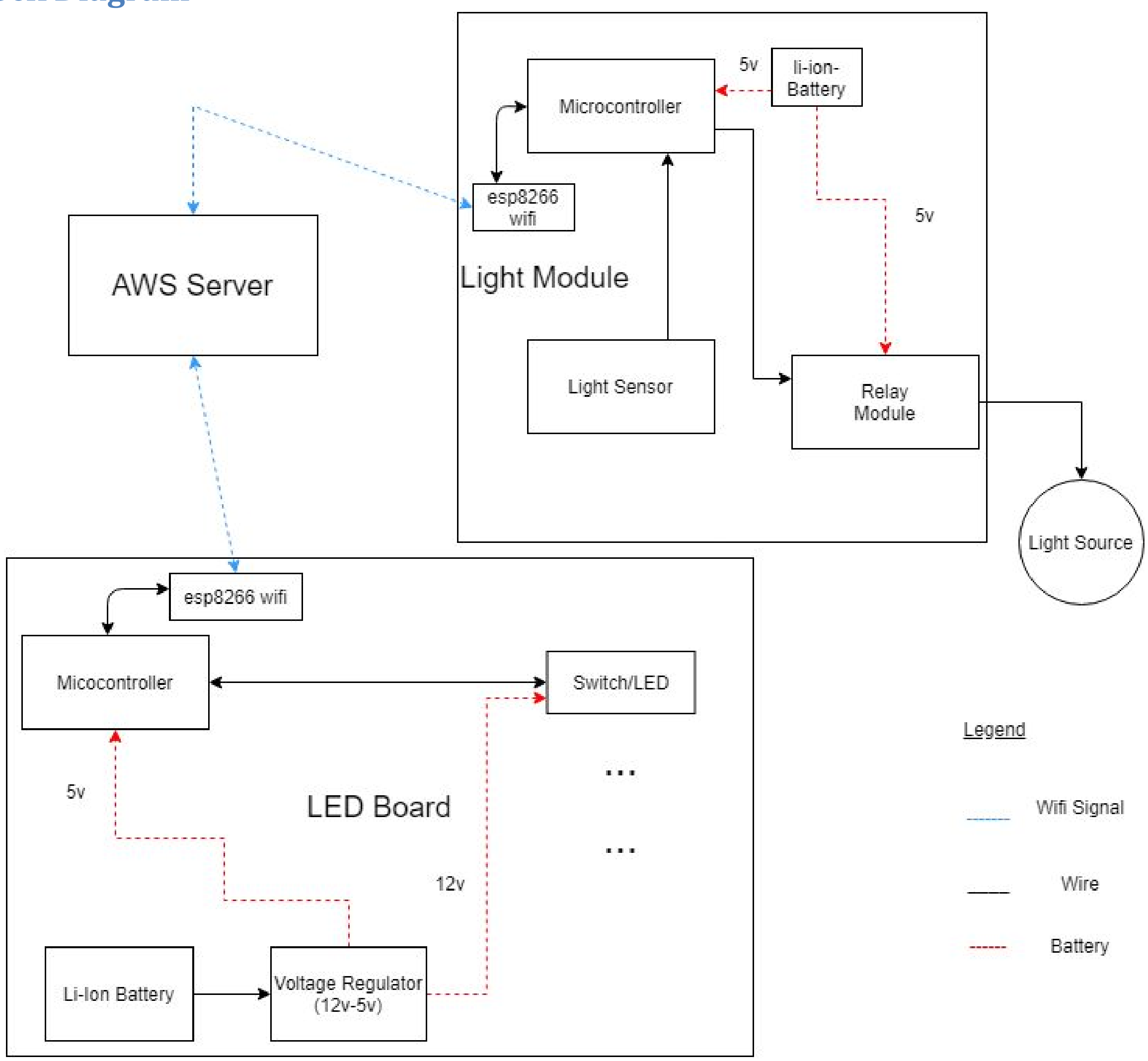
- A Light Module signals a light is on only when the OPT3001 detects light of intensity equal to or greater than the luminosity threshold determined from its main light source.
- The microcontroller must be able to interpret light sensor data wirelessly and turn the correct board LEDs on or off.
- The User must be able to turn off individual lights from the LED board.

1.4 Visual Aid

Figure 1. Individual Light Module (Left) LED Board (Right)



1.5 Block Diagram



Light modules can be expanded similarly with the blueprint above adding another switch/LED and installing another light module on a Light Source

Note:Light source comes from light in your house the voltage may differ based on the light you install this light module on

Figure 2. Block Diagram

The design involves primarily choosing a light source such as a lamp or other light in your house and installing a light module near that light source. This will involve connecting the relay, the light sensor and wifi enabled microcontroller. This coupled with the Led board that has the corresponding switches for the light modules active will allow communication through an aws infrastructure.

2 Second Project Implementation

2.1 Implementation of Arduino Code to Connect to IOT Cloud

Introduction

An integral part of this project relies on successfully communicating light sensor data to the cloud. If this requirement is not met then it will be impossible to signal to the LED board wirelessly that an led must be HIGH or LOW and similarly for the light module to know whether a switch was turned on or off on the led board. This requirement will be fulfilled by showing an example of how a microcontroller can communicate light data to the cloud by connecting to IOT cloud web service.

Limitations of implementation

1) Unable to verify or test any meaningful simulation because I do not own the required parts and am unable to purchase the esp8266 wifi module as well as the light sensor required to send any real data to the cloud.

2) Limited by my access to a functioning lab due to covid-19 and therefore unable to have access to resistors, breadboards, capacitors, voltage regulators and so on.

Implementation Steps

Our goal is to be able to send data wirelessly to aws iot web service which will allow us to interact with other aws web services seamlessly such as an EC2 instance or interact with buckets in s3 storage. Once we are able to establish a working connection from microcontroller to aws iot then we will be able to send light sensor data easily to s3 storage or directly access them in a ec2 instance running a python script.

1) Setup AWS IOT with each individual light module used in the house. For example, we will consider the case of the living room lamp. This lamp will connect to the cloud through IOT core service from AWS and will communicate data through a http protocol where we can send data such as light_status through the response body of a http request.

   Please refer to Appendix A figure 1 for example IOT core setup.

2) In order to set up the iot device, or the microcontroller of the living room lamp, login onto a root aws account and go to IOT core and create an iot device and save the public and private keys of this device in order for later connection details. Refer to Appendix A1.

3) This will essentially setup the device onto the aws ecosystem, now in order for this implementation to work the microcontroller must be initialized onto a wifi network and then be able to make a http request to the aws ecosystem in order to send data wirelessly from the living room lamp module to aws cloud.
    - Utilize boilerplate functions provided by aws to connect your microcontroller to an aws iot ecosystem.
    - Fill in details of private and public keys and local wifi network in arduino code.
    - Obtain signal from the server in a similar fashion through http protocol and based on the relay signal turn on or off the light if the signal is active.
    - The final step would include initializing the light sensor and connecting the circuit properly as detailed in the circuit schematic for the light module then sending that data under the first comment in the loop function.

Please refer to Appendix A, figures 2,3 for detailed code.

2.2 Power Supply

To meet our high-level requirements, each module requires an adequate power supply.There are two critical power modules- the LED Board Power System and Light Module Power System. The LED Board Power System is composed of a 12 V Lithium-Ion battery pack and a 12 to 5 V fixed output linear voltage regulator (ADP3333). To filter out noise the ADP3333 requires two 1μF bypass capacitors [16]. Light Module Power Systems include a 5 V Li-Ion battery pack and a 5 to 3.3 V fixed output linear voltage regulator with 1μF bypass capacitor.

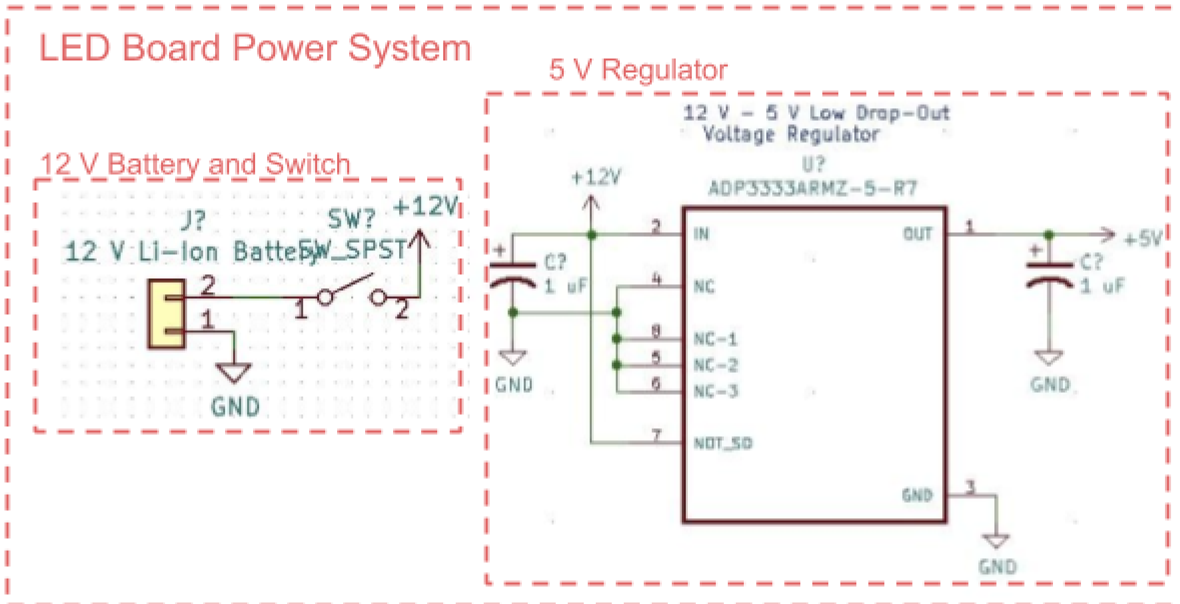


Figure. 3 LED Board Power System Schematic

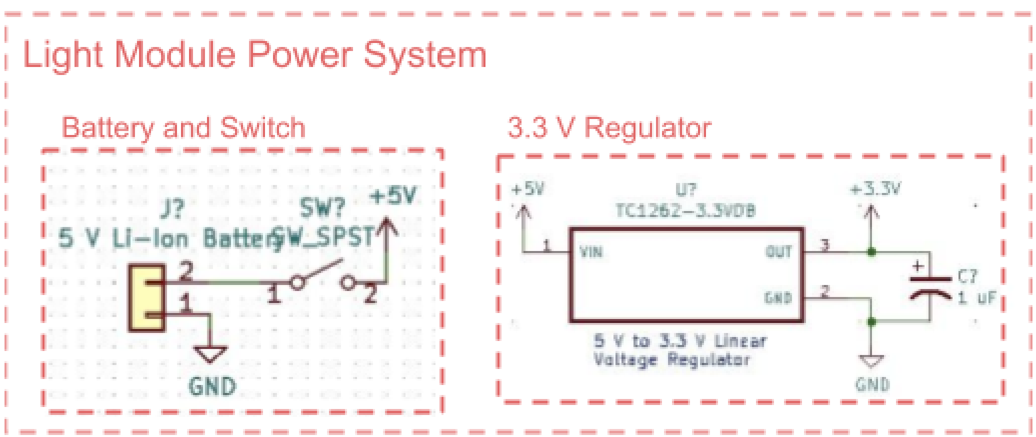


Figure. 4 Light Module Power System Schematic

2.3 Light Module

A light module is responsible for determining the status of a light, communicating on/off light source data wirelessly to the server, and communicating server commands to its relay module. Each room's main light source will have its own light sensor module. Individual Light Modules consist of a wifi-module (ESP8266), microcontroller (Atmega328), OPT3001 Digital Ambient Light Sensor, and power system explained in section 2.2.

2.3.1 Ambient Light Sensor

The OTP3001 must be able to detect the lux level of the light source during both day and night conditions.The only peripheral circuits needed for the IC chip are a bypass capacitor and pull-down resistor. A 100 nF bypass capacitor is recommended to filter out noise from the power supply [12]. Connecting a 10 kΩ pull-down resistor between the sensor’s INT, SCL, and SDA pins and the 3.3 V power supply results in a pin current of

$$3.3\ \mathbf{V}/10\mathrm{k}\Omega = 3.3\ \mu\mathrm{A}$$

This is within the 10 mA absolute maximum current rating into any pin [12].

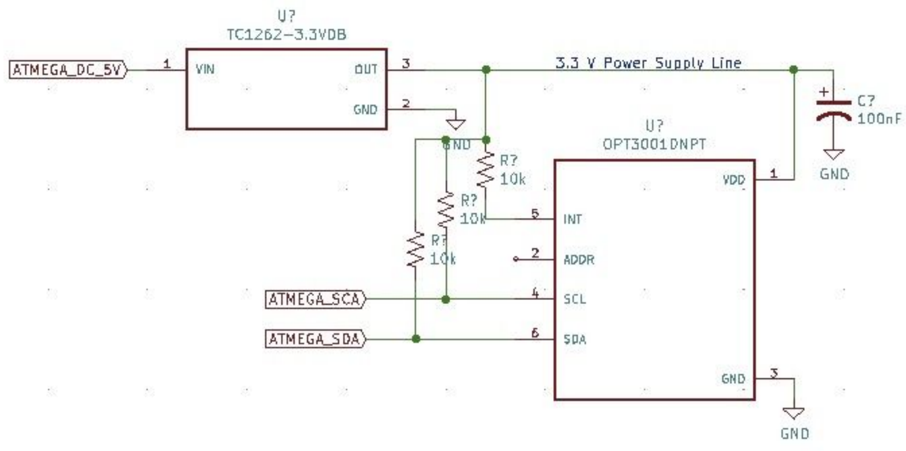
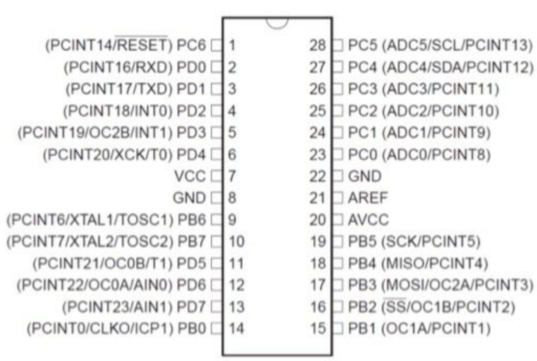


Figure 5. OPT3001 Circuit Schematic



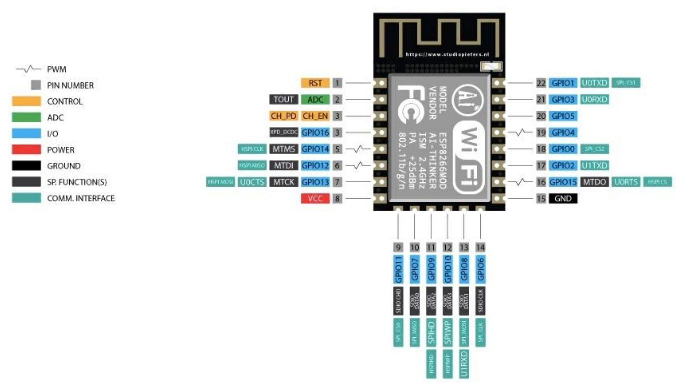


Figure 6. ATMEGA328P Pin Configuration

Figure7. ESP8266 Pin Configuration

2.4 Relay Module

Each of the light fixtures in our project need to be connected to our relay module in order for them to be controlled wirelessly. Without the relay we cannot fulfil our third high level requirement. The relay is connected in series with the light fixture and power. More specifically the light fixture's power is connected in the normally closed pin as shown in the schematic below. The reason for this is that when the relay detects a high value at pin 1 then it will turn off the light which is how we desired for the relay to operate. The first pin takes in input from the atmega328p that in the light module. We did this so that we did not need to buy additional esp8266 wifi modules for each relay.

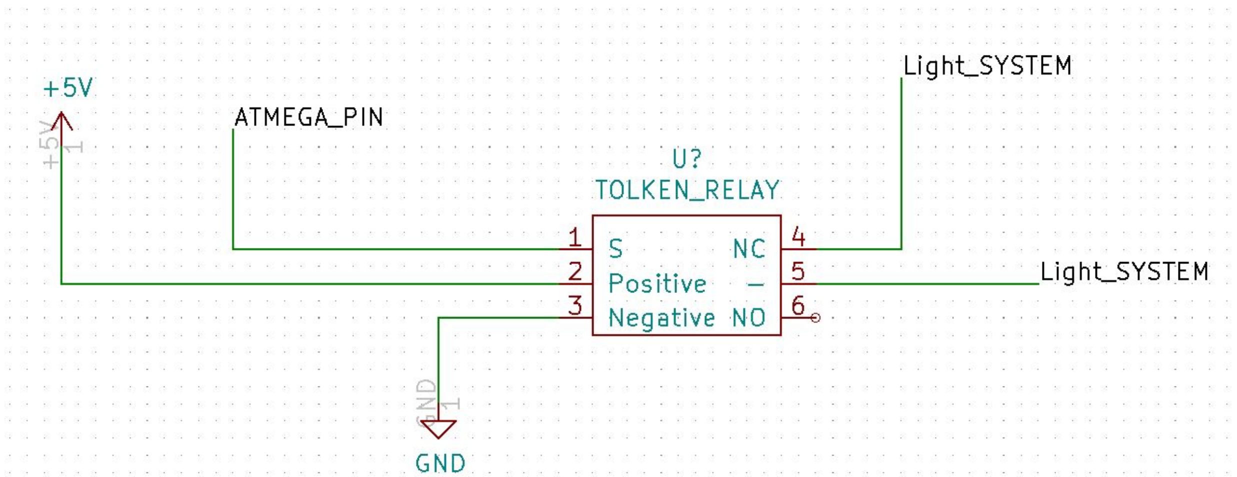


Figure 8. Circuit Schematic For Relay Configuration

2.4.1 Tolako 5V Relay

There are two main concerns with using a relay for controlling light fixtures: its power draw and its longevity.

2.4.1.1 Power Draw

At the 5v DC that we would operate the relay at there is a power draw of about .36 watts [15]. This is shown below in table 6. At this power, the relay experiences a temperature increase of 25 degrees celsius which is shown in figure 9 [15]. The power draw of the relay is not a major concern due to the amount of power that the light module's power source can support. However the temperature increase is not ideal so we would need to create a housing for the relay that is perforated so that heat can dissipate better.

6. COIL DATA CHART (AT20°C)

Coil Sensitivity	Coil Voltage Code	Nominal Voltage (VDC)	Nominal Current (mA)	Coil Resistance (Ω) ±10%	Power Consumption (W)	Pull-In Voltage (VDC)	Drop-Out Voltage (VDC)	Max-Allowable Voltage (VDC)
SRD (High Sensitivity)	03	03	120	25	abt. 0.36W	75%Max.	10% Min.	120%
	05	05	71.4	70				
	06	06	60	100				
	09	09	40	225				
	12	12	30	400				
	24	24	15	1600				
	48	48	7.5	6400				
SRD (Standard)	03	03	150	20	abt. 0.45W	75% Max.	10% Min.	110%
	05	05	89.3	55				
	06	06	75	80				
	09	09	50	180				
	12	12	37.5	320				
	24	24	18.7	1280				
	48	48	10	4500	abt. 0.51W			

Table 6.

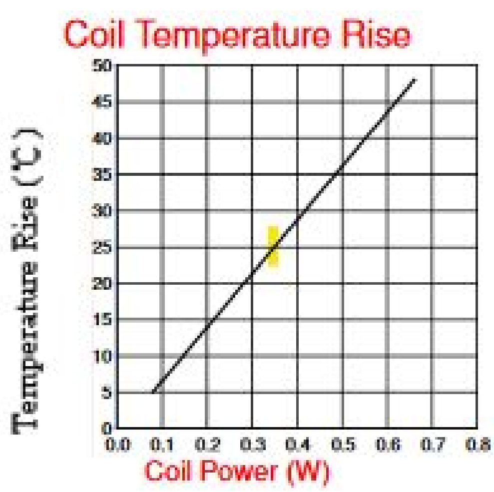


Figure 9. Coil Temperature Rise

2.4.1.2 Longevity

Another concern with the relay is its longevity due to the high voltage that the relay needs to switch on and off. The chart below is operating the relay at 24v DC but according to table 6 the power is still about .36 watts. Assuming that the user wants to control a lamp that uses 3 light bulbs that are 60 watts. The total current drawn by the lamp at 120v AC is (3*60)/120 = 1.5 amps. Looking at the chart below the relay operates for about (20*10000) = 200,000 operations. This number is fine for our project since the user will not always use the board to control their lights and thus the overall usage rate of the relay is lower.

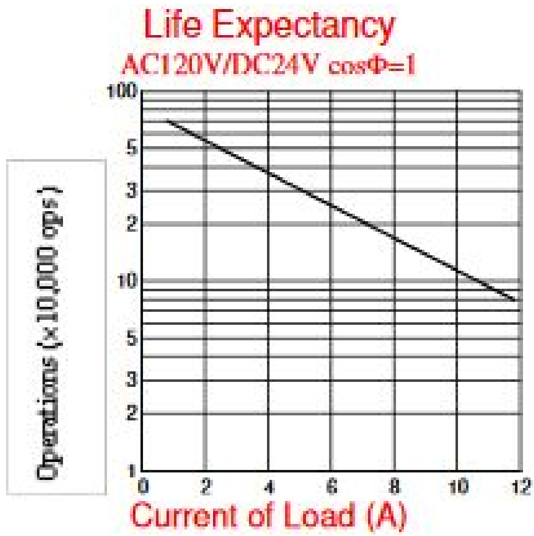


Figure 10. Life Expectancy

2.5 Tolerance Analysis

2.5.1 Light Sensing Accuracy

A Light Module signals a light is on only when the OPT3001 detects light of intensity equal to or greater than the luminosity threshold determined from its main light source. The varying lux levels between light sources should not affect the functionality of our design. Ideally the sensor will only detect light from its main light source and it will not be affected by varying daylight conditions in the room. Light from night conditions is negligible, however, we need to prevent daylight from interfering with sensor reading.

Average light levels in a home range from 500-1000 lx [11]. On a clear day light levels from a window may be 25-30 lux in the middle area of a room. Setting the minimum luminosity threshold of the OPT to be 350 lx in a bedroom with a light source of 450 lx gives a normalized response of 0.778. This is within tolerance since the OPT3001 has a normalized response of 0.96 at $\pm$ 15°. The sensor should not detect lux levels below 336 lx.

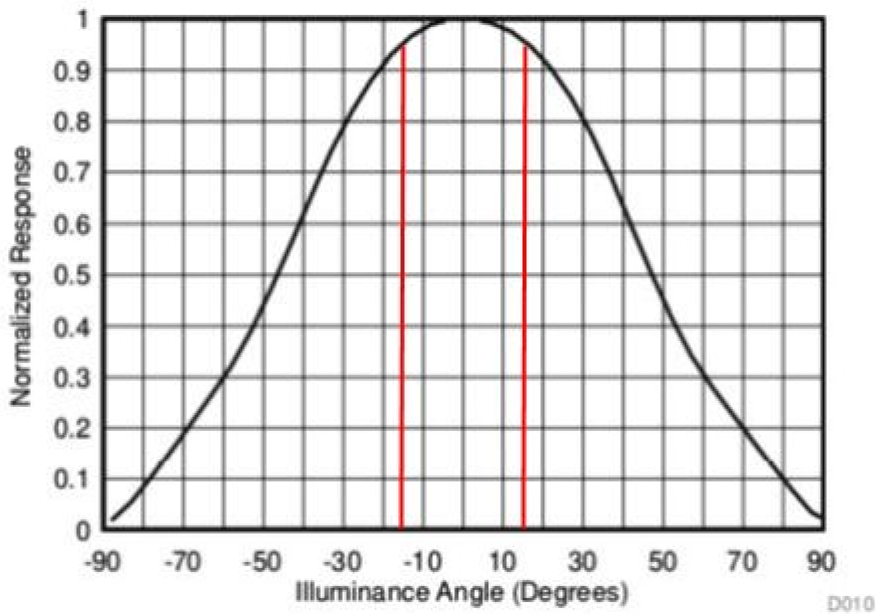


Figure 11. OPT3001 Normalized Angular Response

2.5.2 Window Casing Calculation

The OPT3001 has a field-of-view of approximately ±45°, where 50% or more of the incident light is detected [10]. Since our design is a targeted application, a smaller FOV can be used. To keep outside light from reaching the sensor, the device should be situated so that the main light source is at a normal incidence angle to the sensor. We are able to adjust the window casing for the OPT3001 so that its FOV is ±15°. This should prevent detection of luminous flux from outside sources. Below is the window casing calculation.

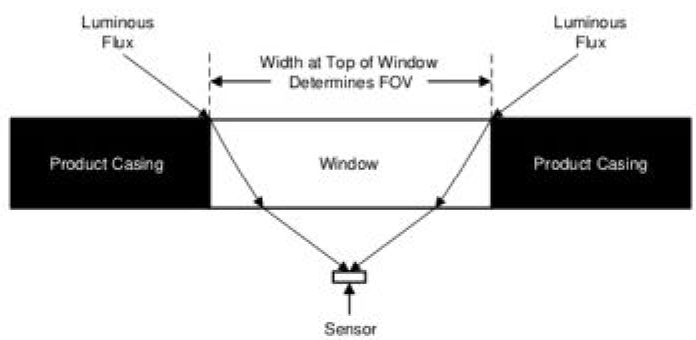


Figure 12. Window Casing OPT3001

Relevant equations and variables obtained from [10]:

- $W = (SensorWidth(XorY)) + 2 * (WFOV + \Delta W)$
- $WFOV = h' * \tan\Theta 1$
- $h' = h - hs$
- $\Theta 1 = \pm FOV°$

- $\Delta W = t * \tan\Theta 2$
- $\sin(\Theta 2)= \frac{n1}{n2} * \sin(\Theta 1)$
- $\Theta 2=\arcsin(\frac{n1}{n2} * \sin(\Theta 1))$

h_s = sensor height above the PCB, typically 0.38 mm [12]

h = product casing height above the PCB

W = width of window

W_{FOV} = window dimension defined by FOV angle and distance from sensor

ΔW = window dimension defined by light bending as a function of window refractive index and thickness

N_1 = refractive index of material between sensor and bottom of window (air)

N_2 = refractive index of window material

t = glass thickness

Sensor X = measurement (in mm) of sensor in the X direction [12]

Sensor Y = measurement (in mm) of sensor in the Y direction [12]

Θ_1 = angle from surface normal to the incident light ray

Θ_2 = angle from surface normal to the incident light ray in material of refractive index N_2

System Level Requirements:

- Desired FOV is ±15°
- Thickness of window is 1 mm
- Height from the PCB to the bottom of the window (h) is 5 mm
- Index of refraction of the window material (N2) is 1.5
- Sensor Width X is 0.49 mm, Sensor Width Y is 0.39 mm

Our prototype design requirements are calculated below.

$$h' = 5\text{ mm}-0.38\text{ mm} = 4.62\text{ mm} \tag{1}$$

$$WFOV = 4.62\text{ mm} * \tan 15° = 1.238\text{ mm} \tag{2}$$

$$\Theta 2 = 9.936° \tag{3}$$

$$\Delta W = 1\text{ mm} * \tan 9.936° = 0.175607\text{ mm} \tag{4}$$

$$Wx = 0.49\text{ mm} + 2 * (1.238\text{ mm} + 0.175607\text{ mm}) = 3.317\text{ mm} \tag{5}$$

$$Wy = 0.39\text{ mm} + 2 * (1.238\text{ mm} + 0.175607\text{ mm}) = 3.217\text{ mm} \tag{6}$$

3 Conclusion

3.1 Implementation Summary

3.1.1 Arduino code to connect to IOT cloud.

This example connection shows how one of our many light modules will seamlessly connect and send data to the aws ecosystem. This model also shows how data can be sent back to the microcontroller as well in order to communicate to a relay connected to the light source whether to turn on or off a light that has a light module installed on it. From here the next steps would include grabbing the data from IOT core and funneling it into a datastream to our ec2 instances. The ec2 instance running our server code will listen and send data back to the led board or vice versa if a switch is triggered on the led board. This will involve sending data back to the specific microcontroller back through iot core and down to the microcontroller. Finally, It is unfeasible to actually send data to the aws ecosystem because of limitations detailed above, however this model hopes to show a detailed example of how our project aims to accomplish one of the most vital software functions to our project. This was done by Raj.

3.1.2 Power Supply

Due to the covid-19 pandemic, we were unable to use the lab equipment required for the verification of the voltage regulators and battery sources. However, the schematics shown in section 2 allow us to create power requirements and verifications shown in the appendix. A functioning power supply is crucial to the success of our design. The Power Supply and Tolerance Analysis was done by Julia.

3.1.3 Light Module

The calculations in section 2 are critical to our design. The OPT3001 must be able to determine the room lux levels and communicate that information to it’s microcontroller in order for our design to work. Implementation steps taken to achieve this goal include providing the optical sensor with peripheral circuits that optimize its performance. Also, the tolerance analysis in section 2 outlines the device requirements we need in order to make sure that the subsystem can detect light accurately in varying conditions, as specified in our high-level requirements. The analysis for this module was done by Julia.

3.1.4 Relay

The calculations in section 2 showed that our relay’s power usage is not very high and that it can handle 200,000 operations. However the analysis also displayed that the relay will get relatively hot when switched on. The solution to that is to make the housing for the relay as perforated as possible so that heat can dissipate better. The analysis demonstrated that the tolako 5v relay can handle the loads that would be in use for this project. Due to its low power output it can be connected to the same power source as the light module and it would not have a major impact. Also the large amount of operations is also good because the relay can handle an appropriate amount of use. The temperature increase can be easily mitigated by a properly designed housing. The analysis for this module was done by Amrit.

3.2 Unknowns, Uncertainties, Testing Needed

3.2.1 Ambient Light Sensor Verification

Without lab access we can not properly test the ambient light sensors that are used in the light modules. Our main analysis of the sensor has been purely theoretical and based on the technical specifications of the device. If we had lab access we could use a power source to power a sensor and a voltmeter to verify if it operates as desired. To verify if the sensor works we would utilize a voltmeter and a lamp. The sensor would be powered by the supply and placed near the lamp's light bulb. We would then connect a voltmeter to the digital output pin of the device. We can then turn on the lamp and see if the sensor's digital output pin has a high value, indicating that it detects the light. Along with this we would perform a test on if nearby light sources can trigger the sensor. This test would be conducted by placing the sensor at distances away from the test lamp and checking if the voltmeter reads a large value. Without a voltmeter and power supply we could not test this device and verify that it completely works in the scenarios needed for the project's success.

3.3 Ethics and Safety

3.3.1 Ethics

Our design is in compliance with the IEEE Code of Ethics [2] and the ACM Code of Ethics and Professional Conduct [3]. To prevent any harm done to the user and the environment, as stated in ACM, code 1.2, and in IEEE code of Ethics, code 1, we will make our design process and the final product as sade as we can. Additionally we will strive to make the highest quality product that we can make which is in accordance with ACM code 2.1.

3.3.2 Safety

One issue that could occur would be if the LED board, light sensor module, or relay modules short. This would be a major issue since many of these devices handle high voltage and can potentially harm the user if they fail. For example if the light sensing module or the relay module short they could create a fire. As the number of sensors increases the potential for failure also will increase, so a major concern that we will work on is to make the modules as safe as possible so that this will not be a major issue. We will work on adding fuses wherever possible in the modules to avoid this problem.

The user is also responsible for installing the relay to their lighting fixture so there is an inherent danger where the user did not properly set up the relay module and this system may shock the user. Since the relay can control a 120 Volt device this shock can be deadly. For this reason the product should only be installed by people that have experience with installing electrical systems.

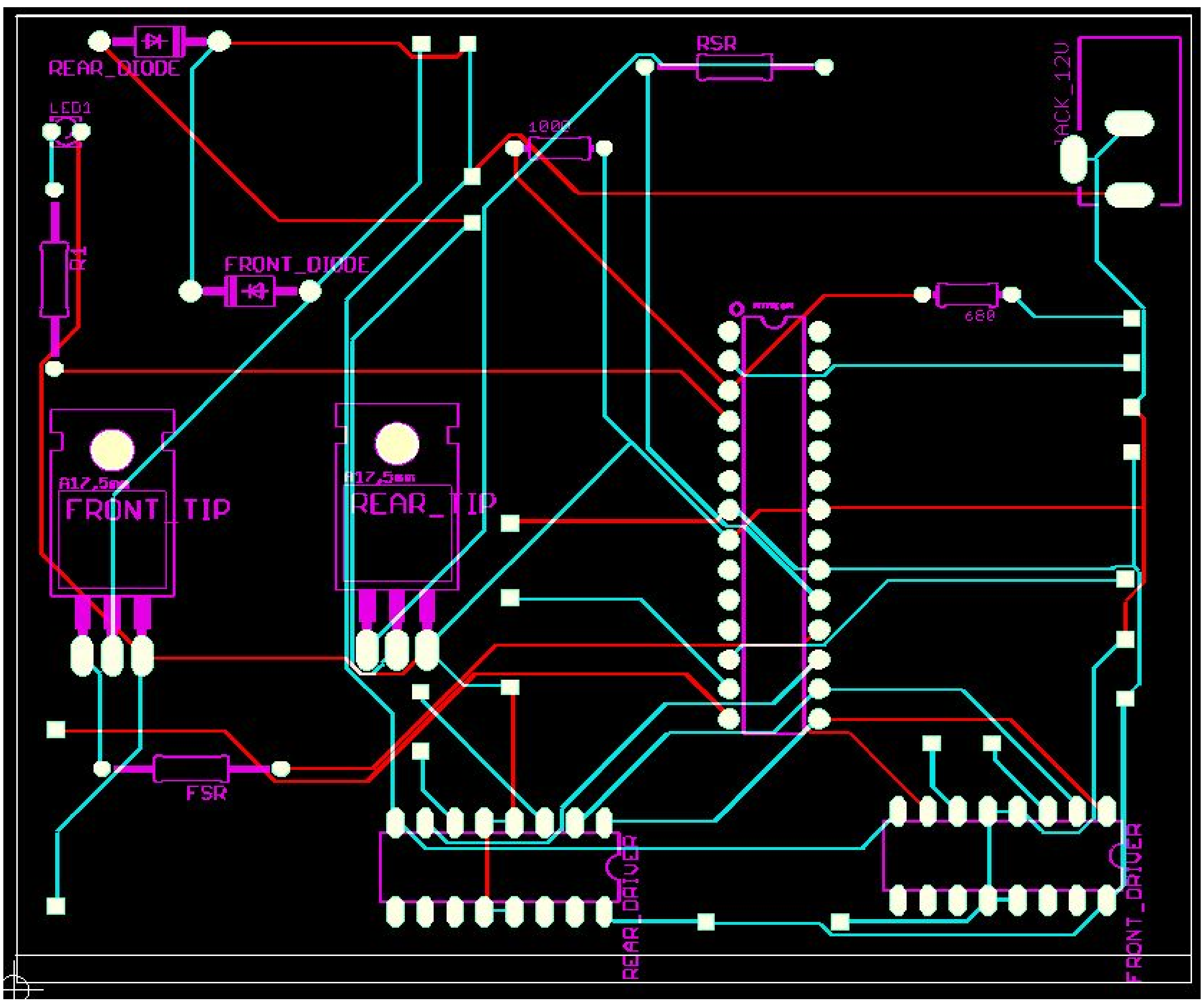
3.4 Project Improvements

- Since ac power is readily available at each light fixture, batteries are not necessary. Therefore, it would be more feasible to implement the Light Module Power System using the provided 120 V at each light fixture instead of an outside 5 V battery supply. This modification should significantly reduce design costs as well as prevent unnecessary maintenance since the user doesn't need to monitor or change any batteries.
- The OPT3001 optical sensor may not be necessary. Instead, we can connect current and voltage monitors to the wiring of each light fixture. Current and voltage monitors would reduce design cost. Additionally, they provide a simpler implementation and verification than the optical sensor which has specific device requirements and testing detailed in section 2.
- The LED Board user interface may not be necessary. Instead, since AWS is used, users could use an app instead. An additional alternative to the LED board is to make the design compatible with current smart home systems.

4 Progress on First Project

4.1 Main Circuit PCB

After the first design review we designed the main PCB for our three piece bike lock. This circuit holds each of the modules necessary for operation and it would be installed in the housing of the middle lock. We made other optimizations with this PCB such as no longer using IR sensors for the front and rear locks so that we can maximize the amount of atmega328p pins that are available to us and so we can reduce power consumption. Each of the motors and actuators that were used in this project would be soldered to the board and then placed in their desired location. The reason why we decided to make one main circuit instead of breaking up the circuits into three parts was so that we could make the front and rear locks more compact as their mounting does not give us as much room as the middle lock. This circuit would be powered by connecting our 12v battery to the barrel jack on the PCB.



5 References

[1] “Philips HueWCA 9.5W A19 E26 4 set US BLE,” *Hue*. [Online]. Available: https://www2.meethue.com/en-us/p/hue-white-and-color-ambiance-starter-kit-e26/046677548544. [Accessed: 02-Apr-2020].

[2] “IEEE Code of Ethics,” IEEE, Jun-2019. [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html. [Accessed: 03-Feb-2020].

[3] “The Code affirms an obligation of computing professionals to use their skills for the benefit of society.,” Code of Ethics, 22-Jun-2018. [Online]. Available: https://www.acm.org/code-of-ethics. [Accessed: 03-Feb-2020].

[4] “Tolako 5v Relay Module for Arduino ARM,” Amazon. [Online]. Available: https://www.amazon.com/Tolako-Arduino-Indicator-Channel-Official/dp/B00VRUAHLE.

[10] “OPT3001: Ambient Light Sensor Application Guide,” ti.com, Oct-2014. [Online]. Available: https://www.ti.com/lit/an/sbea002a/sbea002a.pdf. [Accessed: 16-Apr-2020].

[11] “Illuminance - Recommended Light Level,” Engineering ToolBox. [Online]. Available: https://www.engineeringtoolbox.com/light-level-rooms-d_708.html. [Accessed: 16-Apr-2020].

[12] “OPT3001 datasheet,” OPT3001 datasheet | TI.com, 2018. [Online]. Available: https://www.ti.com/document-viewer/OPT3001/datasheet/features-sbos6819829#SBOS6819829. [Accessed: 16-Apr-2020].

[13] Iotdesignpro.com. 2020. *Getting Started With Amazon AWS Iot And ESP8266*. [online] Available: https://iotdesignpro.com/projects/getting-started-with-esp8266-and-amazon-aws-iot> [Accessed 8 May 2020].

[14] D. Fernandes, L. I. Berry, R. Santos, L. W. Cherk, S. Santos, Scott, A. Baillie, Drex, M. Azhar, V. Nanda, C. Parsons, Omkar, Y. Y. Hao, Raymond, Greg, Kyle, Martin, Rodolfo, and Jeff, “Guide for Relay Module with Arduino,” Random Nerd Tutorials, 31-Oct-2019. [Online]. Available: https://randomnerdtutorials.com/guide-for-relay-module-with-arduino/. [Accessed: 08-May-2020].

[15] S. Relay, “pdf.” 08-May-2020.

[16] “TC1262-3.3VDBTR,” *DigiKey*. [Online]. Available: https://www.digikey.com/product-detail/en/microchip-technology/TC1262-3.3VDBTR/TC1262-3.3VDBCT-ND/669517. [Accessed: 08-May-2020].

Appendix A

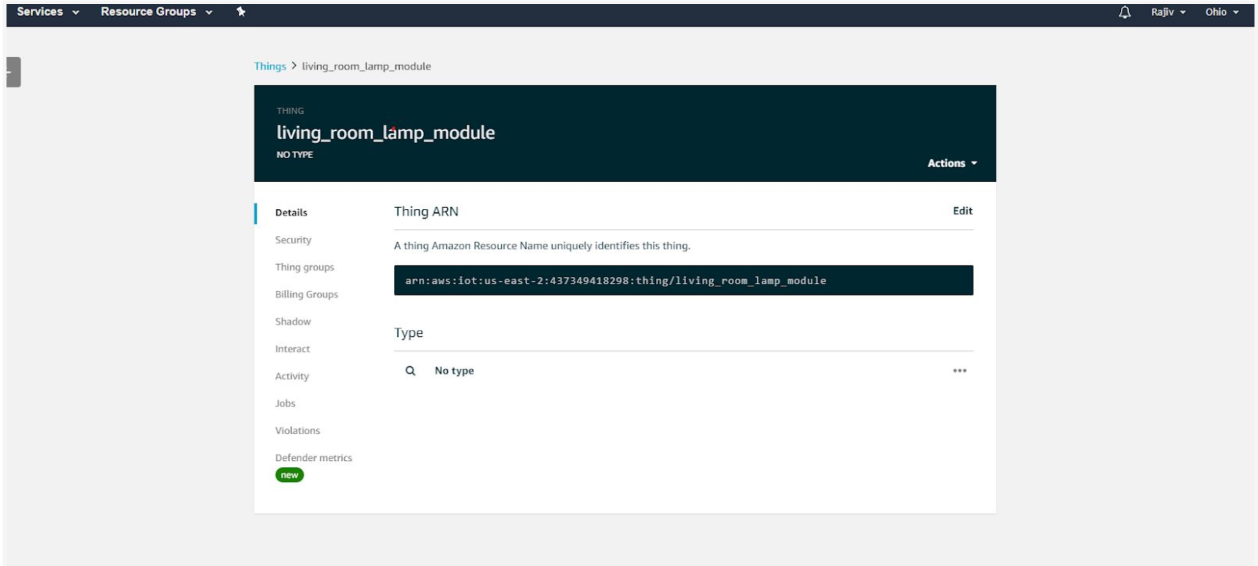


Figure 1 Implementation of AWS IOT core setup

```
#include <ESP8266WiFi.h>
#include <AmazonIOTClient.h>
#include "ESP8266AWSImplementations.h"

Esp8266HttpClient httpClient;
Esp8266DateTimeProvider dateTimeProvider;
AmazonIOTClient iotClient;
ActionError actionError;

char *ssid="netgear42";        // This is your wifi network
char *password="";       // wifi network password of your home network
int relay = 6;
void setup() {
  // This makes wifi connection
  delay(10);
  Serial.println(ssid);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  // This connects onto iot core.
  iotClient.setAWSRegion("us-east-2");
  iotClient.setAWSEndpoint("amazonaws.com");
  iotClient.setAWSDomain("iot.us-east-2.amazonaws.com");
  iotClient.setAWSPath("/things/living_room_lamp_module/shadow");
  iotClient.setAWSKeyID("");
  iotClient.setAWSSecretKey("");
  iotClient.setHttpClient(&httpClient);
  iotClient.setDateTimeProvider(&dateTimeProvider);
  // setup the relay
  pinMode(relay,OUTPUT)
  digitalWrite(relay,LOW)
  Serial.begin(9600);
}
```

Figure 2 Implementation of Light Module Setup code

```
void loop() {
  // First initalize the light sensor and obtain data from light sensor module on the status of the light.
  // We will consider an arbitrary light status since we are unable to produce real data due to limitations.
    int light_status = ?;
    int relay_signal = ?;
  // Consider the case of the relay signal being delivered to the light module so
  // the light must be switched on and the relay activated here
  if(relay_signal == 1){
    digitalWrite(relay,HIGH)
  }
  else{
    digitalWrite(relay,LOW)
    delay(100)
  }
  // Send this light status to the cloud through http request.
  char* light_sensor_data = "{\"state\":{\"living_room\": {light_status }}}";
  char* data = iotClient.update_shadow(shadow, actionError);
  Serial.print(data);
  delay(50000);
}
```

Figure 3 Implementation of Light Module Loop function

R&V Tables

Voltage and Regulator Requirements and Verification	
Requirments	Verification
1. Supplies 3.3 V +/- 5%	1. Connect the input voltage to a 5 V power supply.
	2. Measure Vout using an oscilloscope to see if it
2. Supplies 5 V +/- 5%	1. Connect the input voltage to a 12 V power supply.
	2. Measure Vout using an oscilloscope to see if it meets the requirement