# Modular Autonomous Light

Team 11 - Brandon Noble, Songtao He, Zihong Zeng
ECE445 Project 2 Proposal - Spring 2020
Original Project: Spring 2020 #52
TA - Madison Hedlund

# Table of Contents

# 1. Introduction

## 1.1 Problem and Description

**Problem:**

Some modern buildings have motion detectors installed which are connected to a room's circuitry and can shut off the lights and power in a room when no one is occupying it. People can sometimes forget to turn off lights when they leave a room and this wastes a lot of power. However, currently, there is no modular solution that can be used with older buildings without having to open up the walls and rewire the internal circuitry. To make the action autonomous, we want to design a Modular Autonomous Light. It monitors the number of people/crowdedness in an area, usually classrooms/shop malls and it applies IoT without changing circuitry in the building to turn off lights/change brightness.

**Solution:**

We use WiFi MAC Address Tracking by counting the number of devices connected to the Internet. We will write a sniffer program on ESP 8266 Wi-Fi module to fetch information of devices (MAC address and connected AP) and APs (SSID and MAC address). Then this information will be sent to the microcontroller chip. If there's more than zero devices connected to the AP, the microcontroller chip will send a signal to activate a motor to switch on the light. If the number of devices drop to zero as people may leave the classroom and the connection to the AP is lost, the microcontroller chip will send another signal to activate the motor to switch off the light. The number of people in a room is approximately equal to the number of devices tracked. A person is recognized as a unique MAC address of his or her phone.

## 1.2 Visual Aid

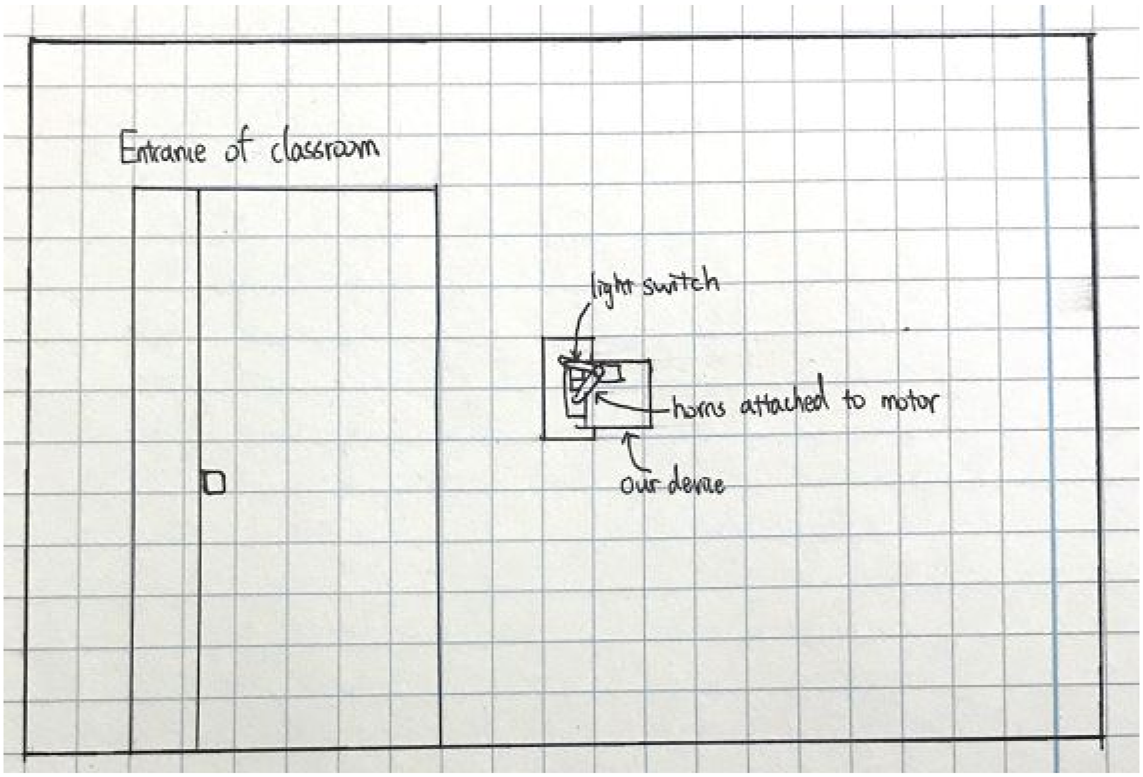



Figure 1. Visual Aid of the product on the wall

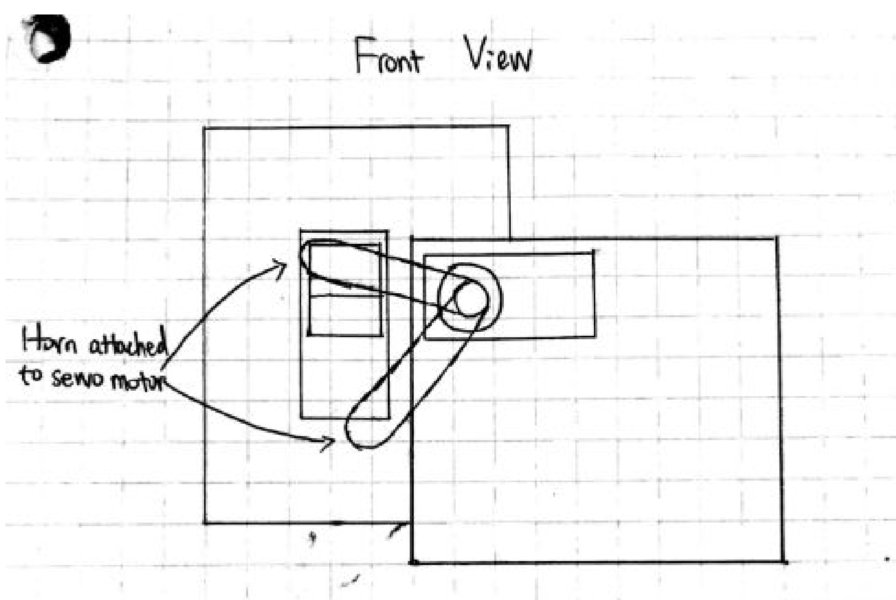



Figure 2. Physical Design(Front view of device)

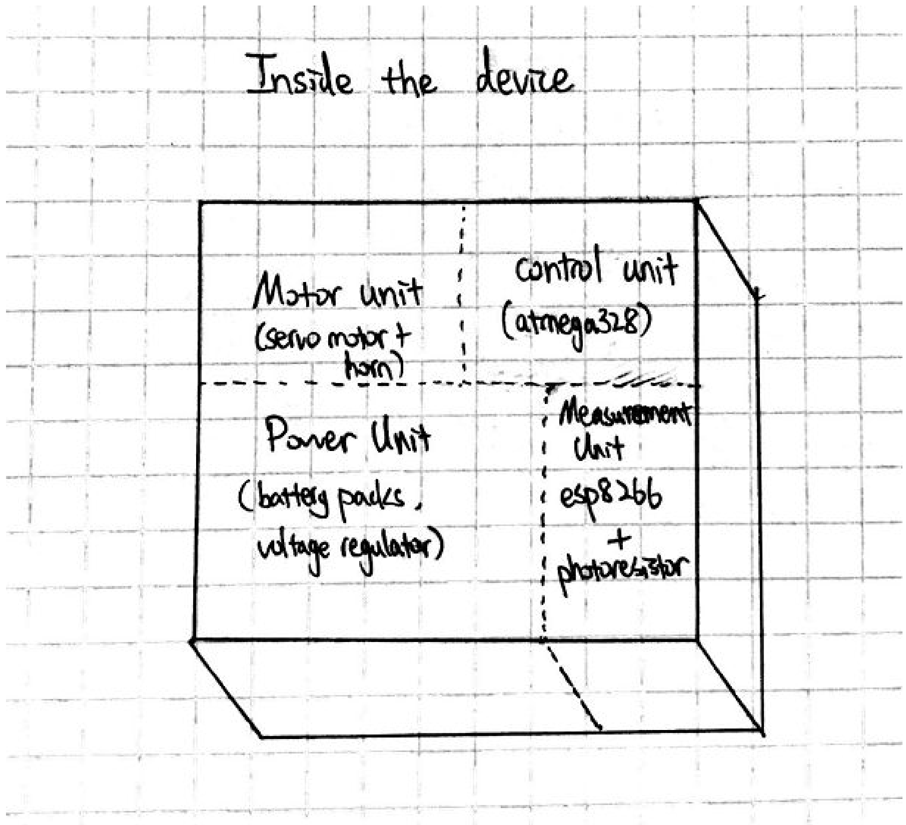


Figure 3. Physical Design(structure inside the device)

## 1.3 High-Level Requirements

- Has an overall accuracy of 80% to turn on/off light(s). The accuracy is measured by the number of times the switch is turned by the motor divided by the total number of times the switch is flipped, both mechanically and manually.
- The light must be switched within 1 minute depending on occupancy of the room, and so long as detected illuminance is below 250 lux
- The device can last for at least 24 hours without need to switch batteries.

# 2. Design

## 2.1 Block Diagram

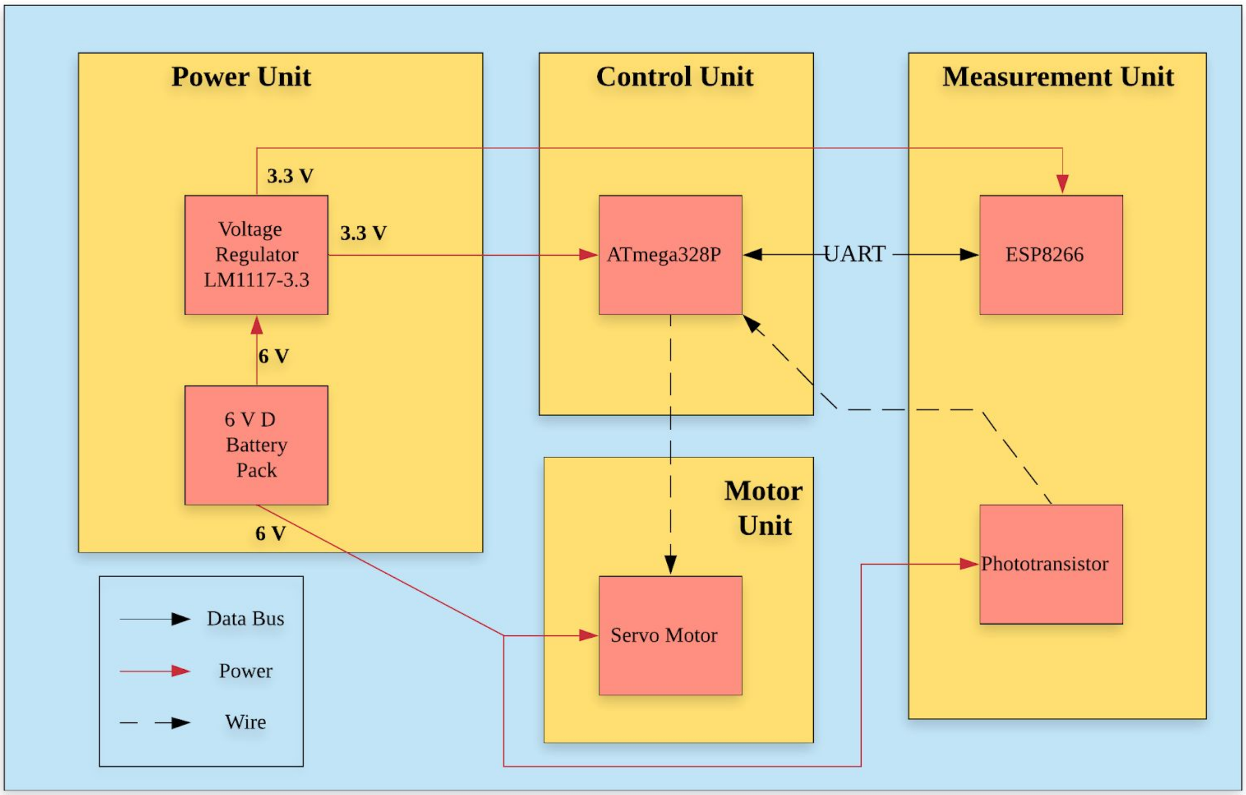



Figure 4. Block Diagram

## 2.3 Functional Overview and Block Requirements

## 2.3.1 Power Unit

**Batteries**

We are using 5 1.2 V D batteries with a max capacity of 10000 mAh in series to provide 6 V To power up our motor unit and photoresistor. We choose D batteries because of its high max capacity and it can provide enough power for our design to last for several days.

Table 1: Requirements and Verification for Batteries

| Requirement | Verification |
| --- | --- |
| 1. Stores >10000 mAh of charge | A. Connect 5 fully charged D batteries in series in a battery case<br>B. Discharge the batteries at 400 mA for 25 hours<br>C. We use a voltmeter to ensure that the battery voltage remains above 6 V |

**Voltage Regulator**

We are using LM1117 voltage regulator to regulate 3.3 V from the batteries and provide power supply for the Atmega328p microcontroller chip and esp8266 wifi module. We chose LM1117 as our voltage regulator because it provides enough current output to our units. It also has a low dropout voltage since we are only using 6V as our power supply. It's also less complicated for us to implement compared to a switch regulator.

Table 2: Requirements and Verification for Voltage Regulator

| Requirement | Verification |
| --- | --- |
| 1. Provides 3.3 V +/- 1.5% from a 6 V source.<br>2. Can draw 800 mA output.<br>3. Maintains thermal stability below 125 degree Celsius. | 1-2.<br>A. Using the circuit schematic in Figure 5., we provide 6 V power supply to the input and regulate 3.3 V.<br>B. Measure the output voltage VREG_3.3V using an oscilloscope, ensuring that the voltage stays within 1.5% of 3.3 V and the current supply is at least 800 mA.<br>3.<br>A. Use an IR thermometer to ensure that the temperature of ICs stays below |

| | |
|---|---|
| | 125 degree celsius during the above steps. |

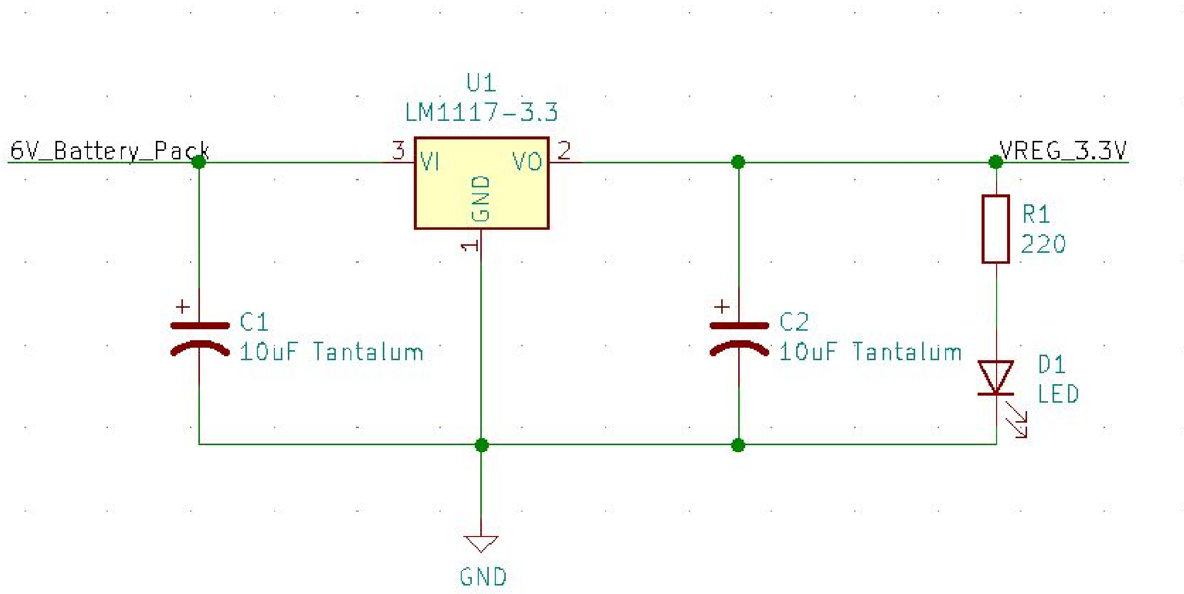



Figure 5. LM1117-3.3 Circuit Schematics

### 2.3.2 Motor Unit

We chose to use a servo motor because it fits the best in the scenario of our design. Compared to dc motor, servo motor has less noise and it's easier to implement [1] because we don't need extra regulators to control the motor which would add more complexity to our circuit. Since we only need to turn the light switch on and off, we don't require a very precise rotation that we only need two angles to achieve our goal so we are not using step motors. In addition, a light-weight servo motor provides enough torque to flip the switch. In the current market, the weight of a toggle light switch is about 10 oz which is 0.2kg. Our device would be placed at a distance within 5cm so it requires 9.8 [m/s^2] * 0.2 [kg] * 0.05 [m] = 0.098 [Nm] = 1 [kg-cm] torque from the motor to flip the light switch. In addition, we want to flip the switch on and off so we will program the motor to turn to two desired angles. To control the angular position of the motor, we would need to vary pulses width between 1ms to 2 ms as shown in Figure 6.

Table 3: Requirements and Verification for Servo Motor

| Requirement | Verification |
|---|---|

| | |
|---|---|
| 1. Motor must be able to provide a torque of at least 0.1 Nm in order to flip the switch.<br>2. Motor must turn to two angles (90 degrees (+/- 15 degrees) and 180 degrees (+/- 15 degrees)) and stop within the targeted range. | 1.<br>A. We would place a motor near to a toggle light switch.<br>B. The motor will be connected to an Arduino’s 5V power supply pin and ground pin<br>C. We will program the Arduino with an online script[2] and test the motor to see if it can flip the toggle light switch.<br>2.<br>A. The motor will be connected to an Arduino’s 5 V power supply pin and ground pin<br>B. We will program the Arduino with an online script[3] and test the motor to see if it can turn to 90 and 180 degrees.<br>C. We will measure the starting position and ending position of the horn attached to the motor with a protractor. |

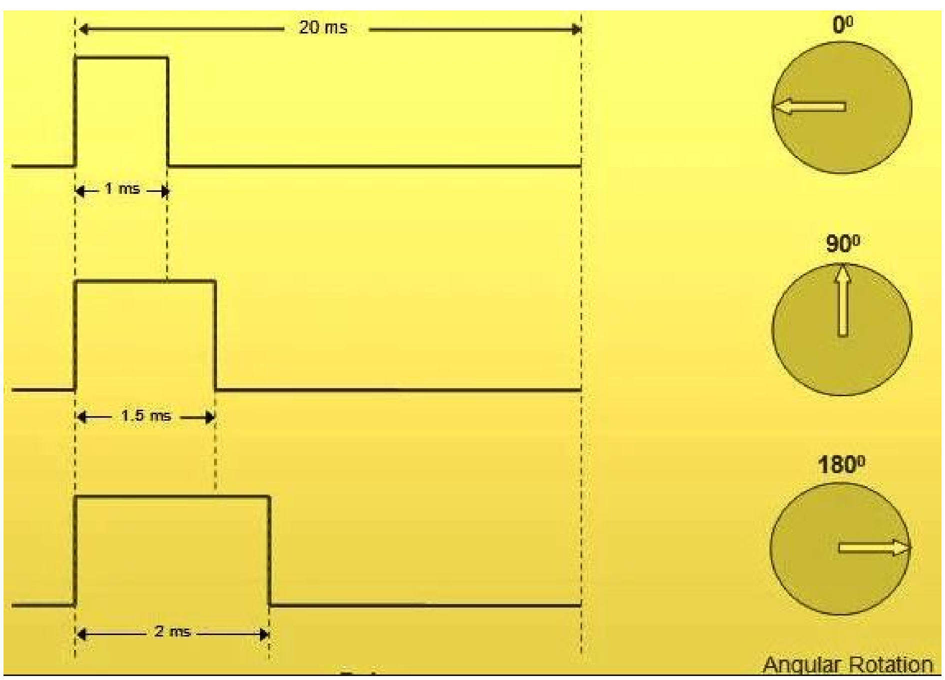



Figure 6. Pulse Width Timing Diagram

### 2.3.3 Light Sensor

**KDT00030TR Phototransistor**

Phototransistors are cheap and sensitive, able to provide a differing current across a range of potential lux values. The drawback of using a phototransistor is its differing current output depending on the type of light used (fluorescent/incandescent). The range in lux advisable for a classroom/office setting is 250-500 lux [4]. The peak wavelength of Sunlight is at 635nm, which is similar to that of fluorescent lighting, and thus the fluorescent light line of Figure 7 will be used for the desired output current of 0.1mA. According to Figure 8, this 0.1mA will result in a desired output voltage of 1V. The phototransistor will be mounted to the front of the case, to ensure the most exposure.

Table 4: Requirements and Verification for Phototransistor

| Requirement | Verification |
| --- | --- |
| 1. The KDT0030TR phototransistor outputs the desired 0.1mA when above 250 lux. | 1.<br>A. Using a fluorescent light source, increase brightness until KDT0030TR outputs 0.1mA.<br>B. Measure the lux using a simple Arduino program and check readings in the serial monitor.<br>C. If it is below 250 lux, increase lux until 250 lux is reached. Note the new current, and use as the benchmark for activation.<br>D. If it is above 1000 lux, decrease lux until 500 lux is reached. Note the new current, and use as the benchmark for activation. |

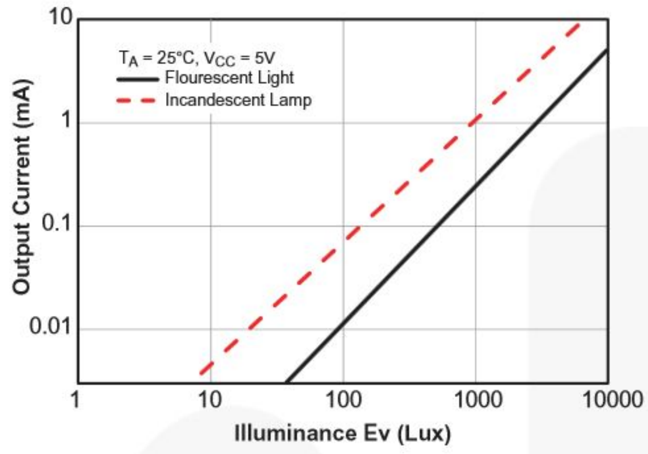



Figure 7. Output current vs. Lux for KDT00030TR

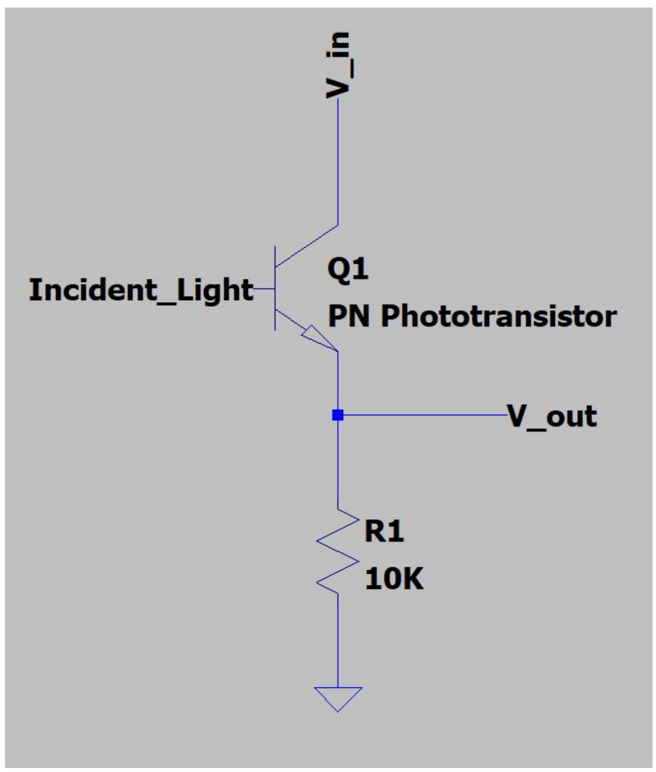



Figure 8. KDT0030TR photoresistor to detect ambient lux

### 2.3.4 Wi-Fi mini sniffer

**ESP-8266 Wi-Fi Module**

ESP-8266 can perform as a standalone system to interface with other systems to provide Wi-Fi through SPI/SDIO[4]. We picked a Wi-Fi module over bluetooth/XBee because it works well as a standalone system. It is efficient, easy to program and cheap. We will write a sniffer program on ESP-8266 to scan a list of devices and the respective AP one is connected to. ESP8266 is required to run in promiscuous mode which will display devices and Access Point MAC and SSID. ESP-8266 is required to run in promiscuous mode, which enables nodes (i.e. device->AP) to share with peers to maximize packet distribution.

Table 5: Requirements and Verification for Wi-Fi Module

| Requirement | Verification |
|---|---|
| 1. ESP-8266 must be able to communicate over IEEE 802.11b/g/n at 4.5Mbps with a 50Ω nominal RF connecton.<br>2. It must be able to communicate over both SPI and UART.<br>3. ESP-8266 must run in promiscuous mode. | 1.<br>A. Assemble WiFi IC on PCB as specified in the datasheet as the basic application schematic.<br>B. Note default WiFi network on a mobile device<br>C. The Serial monitor on Arduino IDE is able to get readings.<br>2.<br>A. Connect to the ESP8266's UART port with an FT232 UART bridge, as per the FT232 reference diagram, and a computer. This can be done on a through-hole breadboard with an ESP-8266 module.<br>3.<br>A. ESP-8266's blue LED flashes. |

**Serial flash**

The serial flash is connected to ESP-8266 by SPI, and it stores the program used by ESP-8266 WiFi module. The memory size is 1 MB and is subject to change as program size varies.

Table 6: Requirements and Verification for Serial Flash

| Requirement | Verification |
|---|---|
| 1. Operates consistently at 80 MHz | 1. |

| | |
|---|---|
| 2. Size must be 1 MB to store program | A. Operates the serial flash at 80 MHz, and WiFi module is able to read a simple program (send strings)<br>B. Receive correct strings as expected<br>2.<br>A. Store 1 MB memory of program/real-time data<br>B. Read expected data from serial monitor in Arduino IDE |

### 2.3.5 Control Unit

**ATmega328**

ATmega328 is used to build a standalone Arduino on PCB[5][6]. It is used for the purpose of turning on the servo-motor to flip the switch based on data collected from esp8266 and voltage value from the photoresistor. We also need a breakout board (SparkFun USB to Serial Breakout - FT232RL) to program ATmega328. The circuit schematics of ATmega328P is shown in Figure 9.

Table 7: Requirements and Verification for Microcontroller

| Requirement | Verification |
|---|---|
| 1. Operation voltage is 3.3 V +/- 1.5%.<br>2. The output to the photosensor should be logical high after ATmega328 receives a logical high input from ESP-8266<br>3. The output to the servo motor should be logical high after ATmega328 receives a 1V or higher reading from the photosensor circuit | 1.<br>A. Use the constant-current circuit in the Fig.8 connecting the output of the voltage regulator to “VCC” in the image<br>B. Measure the input voltage using an oscilloscope, ensure input voltage stays within 1.5% of 3.3 V<br>2.<br>A. Connect the photosensor circuit (Figure 8) input to a digital output pin, and the photosensor circuit output to an analog input pin.<br>B. Confirm to read and send signals with ESP_RX and ESP_TX to receive switch on/off signal from esp8266<br>C. Confirm power to photosensor circuit is cut off when a read from the circuit |

| | |
|---|---|
| | is above the trigger voltage<br>D. Confirm power to photosensor circuit is cut off when a logical low is received from ESP-8266<br>3.<br>A. Connect the servo motor to ATmega328P using circuit schematics from Figure 9.<br>B. Confirm power to servo motor is set after the photosensor detects insufficient luminosity<br>C. Follow the code snippet[7] to set values of registers which controls the high PWM periods.<br>D. Make sure that the motor can turn to 90 degrees and 180 degrees based on values of the positive pulses in a pre-defined period. |

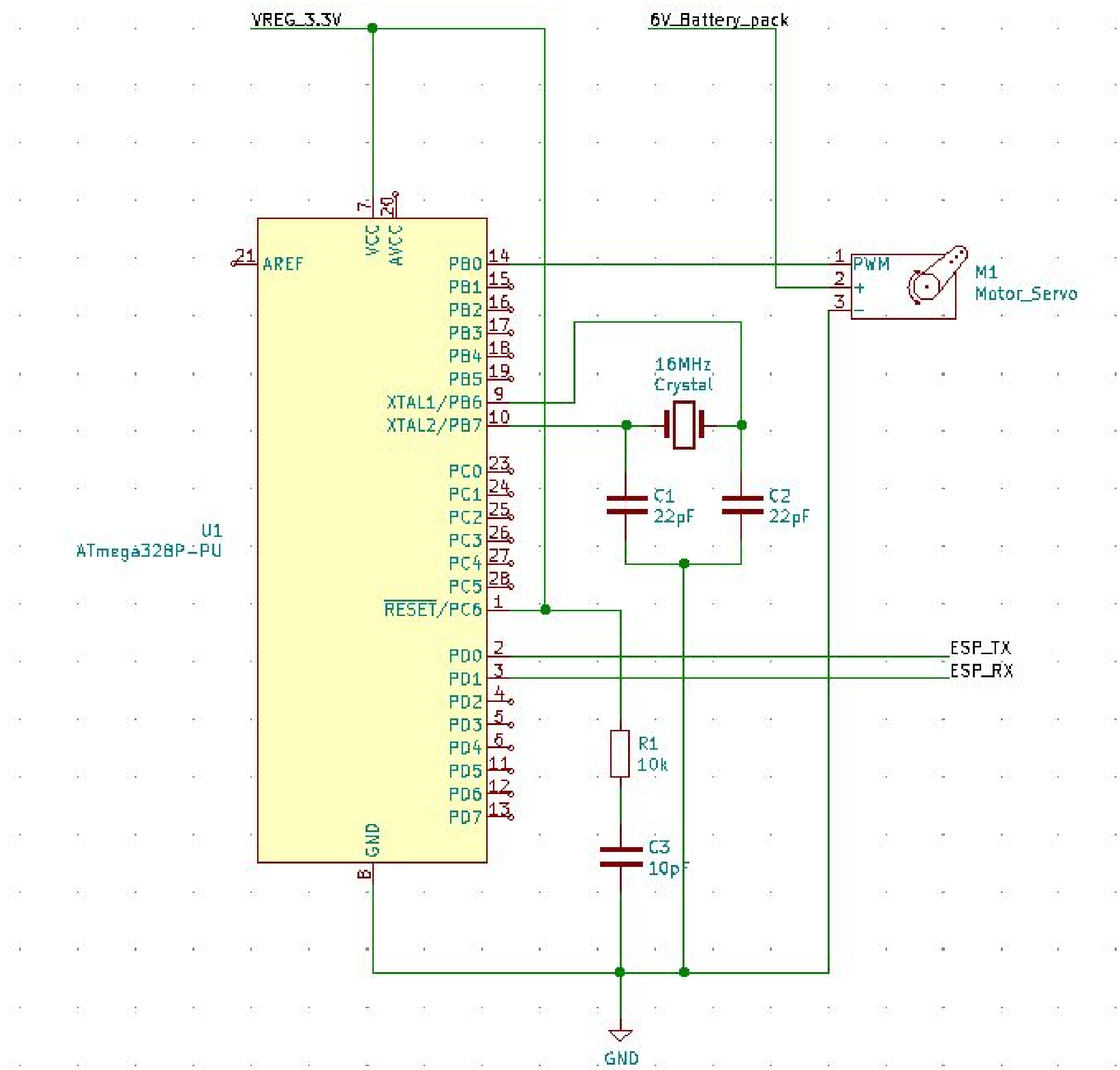



Figure 9. Circuit Schematics of Atmega328P and servo motor

## 2.4 Mini Sniffer Pseudocode [8]

A sniffer program is used in our design to track devices and their connections to APs. The Wi-Fi sniffer scans a list of devices and AP MAC addresses. The occupancy in a room is thus able to be determined by the number of known clients connected to a specific access point. Since people may carry more than one device, we do not count the actual number of people in a room.

Below is the pseudocode:

Setup() {

    Set operation mode to station mode // promiscuous only works under station mode

Set channel
Promiscuous mode disable
Set promiscuous receiver callback
Promiscuous mode enable
}

loop() {
Set channel
Find AP through algorithm described in section 3.3 (read in values from a CSV file containing predicted AP locations)
Determine the specific AP in the room
Check if any device is connected to the AP and is new to the list
If not, increment channel
If yes, register client in a list and increment the number of known clients
Count the number of known clients
}

First we need to set ESP-8266 on operation mode to connect to a Wi-Fi network. Then select a WiFi channel (initial value at 1) to improve WiFi coverage. Disable promiscuous mode to set promiscuous callback (which is setup in another header file) and re-enable promiscuous mode.
After finishing setup we can check if any device scanned is new to our list. If it is registered then we increment its channel (reset to 1 at 14) and count the number of clients known connected to an AP. If the device is new then we register the client and increment the number of clients known. In addition to the number of clients known, we have a list of APs and through the localization algorithm in section 3.3 to locate the AP (readings from a CSV file) and count the number of clients known to find out actual occupancy.

## 2.5 Flowchart

The flowchart shown in Figure 10. demonstrates the data flow of the sniffer program and how we determine whether to turn on/off the motors depending on the number of devices scanned and illuminance of the surrounding environment.

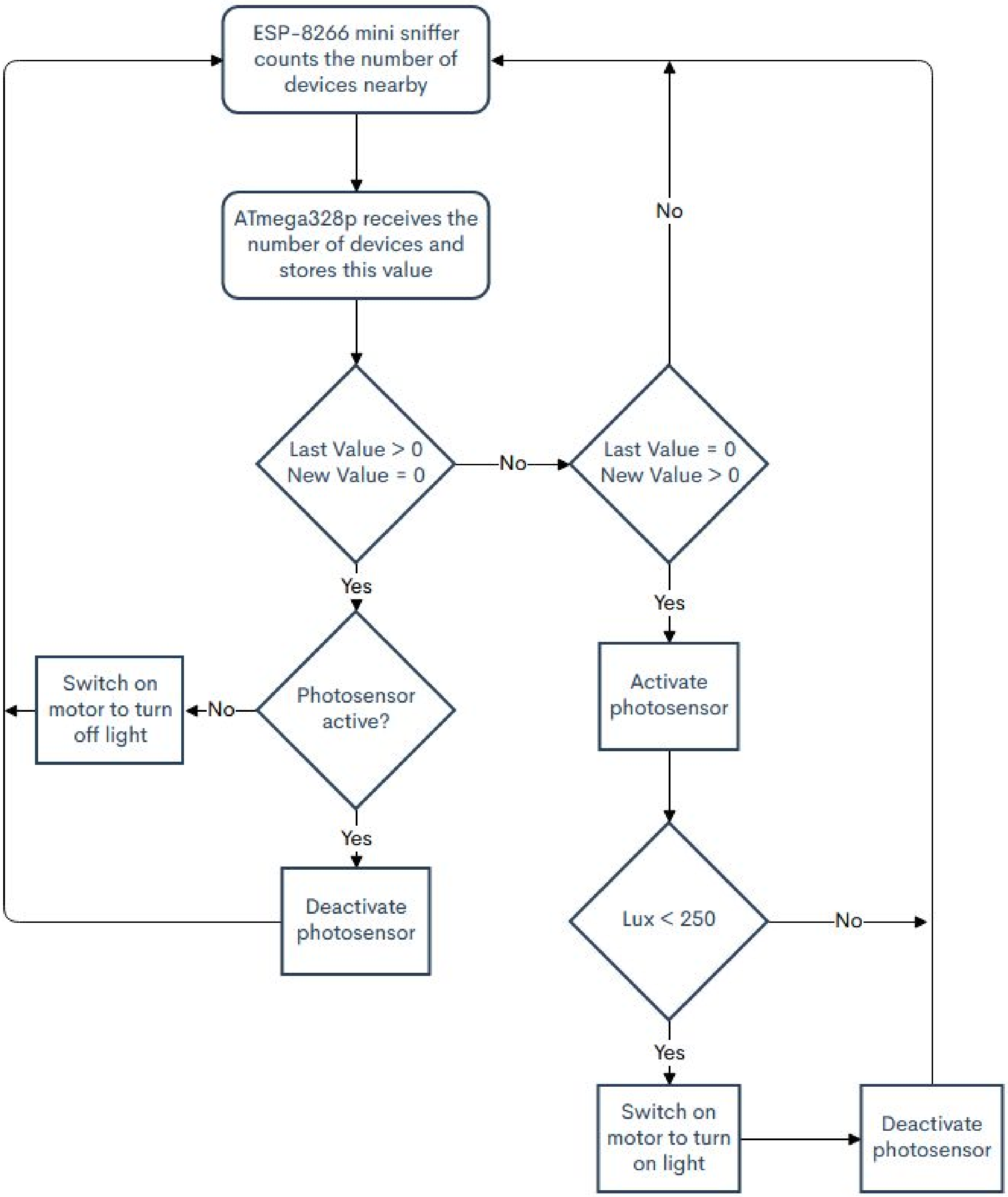



Figure 10. Data flow

## 2.6 Protocol

We use UDP protocol to transmit data. Since we have verification mechanisms within our design, it is suitable to use UDP as error checking and correction are not necessary. UDP effectively reduces the complexity of routing protocol.

## 2.7 Tolerance Analysis

**Analysis of Battery life**

Since the primary goal of our project is to save power, it is essential the batteries do not use too much power to a point that the design consumes more power than it saves from monitoring and switching the light. Battery life is estimated at users' convenience to make sure they do not have to change batteries frequently. We are using 5 EBL rechargeable D batteries as our power supply and each D battery has a capacity of 10000mAh. In our design, we try to avoid recharging or switching batteries of the device too frequently because it may add more labor costs to the project and it is very inconvenient. Therefore, we want to limit the frequency of switching batteries to once per day. In addition, we are using rechargeable batteries to reduce costs in purchasing batteries. Detailed analysis of battery usage is listed in Table 1.

Table 8. Power Analysis

| Components | Current Consumption |
|---|---|
| ATmega328P | Uses 200 mA * 1 h = 200 mAh |
| ESP8266 | Uses 300 mA * 1 h = 300 mAh |
| Serial Flash | Uses 30 mA * 1 h = 30 mAh |
| SG90 Servo Motor | Uses 250 mA * 1 h = 250 mAh(in movement)<br>Uses 10 mA * 1 h = 10 mAh(idle) |
| KDT00030TR Semiconductor | Uses 1 mA * 1 h = 1 mAh |

Considering that our ESP8266 will be run in deep sleep mode for one minute in every two minutes, the current consumption of esp8266 is actually 300 mA * 0.5 h = 150 mAh.

With the same reason, the servo motor will only be in movement to turn on/off the switch after esp8266 scans devices connected to the access point and determines to switch the light. In addition, the servo motor will only run for a short period(less than 10s in every 2 minutes) to finish the turn and stays in idle mode for the rest of the time. Therefore, the current consumption of servo motor is (250 mA* 1/12 h + 10 mA * 11/12 h) = 30 mAh.

Total current consumption of the device = 200 mAh + 150 mAh + 30 mAh + 30 mAh +1 mAh
= 411 mAh

Battery life = 10000 mAh / 411 mAh = 24.33 hrs

The calculation shows that our device lasts for about at least 24 hours and it satisfies our high level requirement.

# 3. Project Differences

## 3.1 Overview

We are unable to find any product in the market that is similar to our design. There is an approach to install a wireless occupancy sensing system in large rooms[9] but it costs about $400/room for installation. Other solutions may rely on motions in the room and it may be inaccurate when people are staying still in a room.

In the original solution, they used two PIR sensors at the entrance to detect whether people are entering or exiting the room and count the number of people depending on the number of times the sensor gets triggered. Then they will send the data to the control unit through bluetooth communication to activate the motor to switch the light. In addition, the data will also be sent to their phone app via bluetooth communication and the users are able to see the usage of lights in a room and command the switch module to turn lights on or off on the app[10]. In the original solution, the design is targeting small conference rooms and study rooms. The major difference between our design and the original solution is that our design is targeting rooms with a capacity of around 40 people and larger classrooms, and we are not using any PIR sensors in our project.

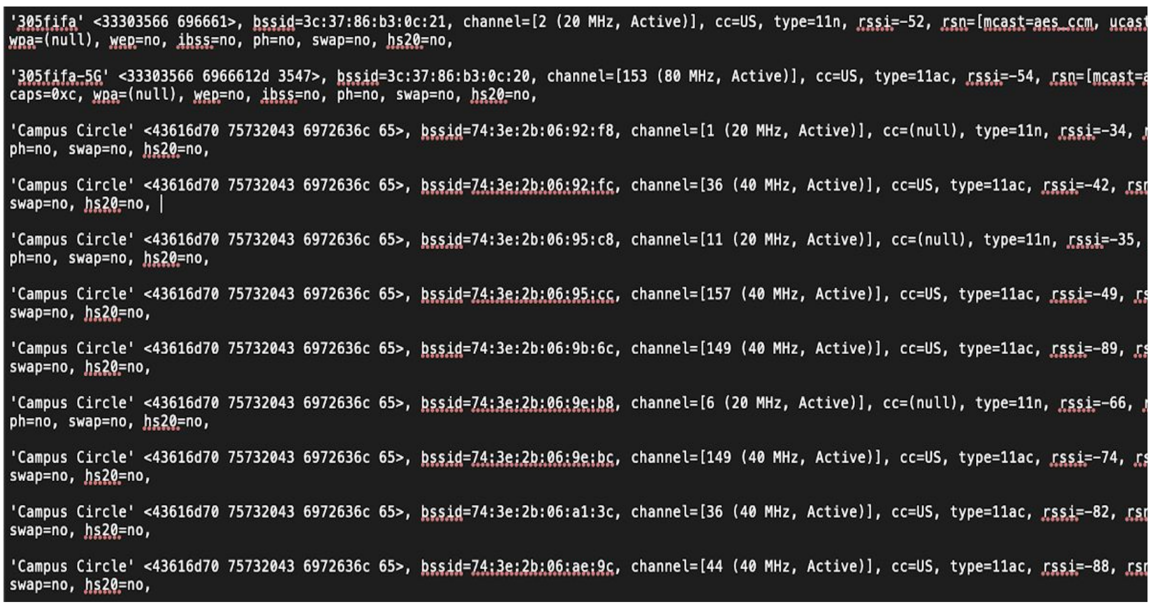
## 3.2 Analysis

As a reference[11], conference rooms in the ECE Building at UIUC have a capacity of 12 - 24 people, lab rooms have a capacity of more than 40 people and classrooms have a capacity of more than 70 people. Compared to the original solution where it's hard to synchronize the counting of the number of people entering/exiting at different entrances, our solution wouldn't have such concern because we determine the occupancy of the room by counting the number of devices connected to the internet with WiFi MAC Address Tracking. However, our device may not be very accurate in small study rooms or conference rooms. Therefore we may consider adding features to enhance this functionality of our design to fit better in smaller rooms. Our design would be more accurate in counting people in the scenario when many people are entering or exiting the room at the same time especially when multiple people are gathered at the entrance of the classroom. Furthermore, a photosensor will determine the general brightness of the room to decide whether lights are needed in the first place so it conserves energy by keeping lights off when occupants are detected if ambient light provided through other sources is sufficient.

## 3.3 Localization Algorithm

Of our blocks, the piece that will pose the greatest problem will be the algorithm locating AP in the device installed room. This block will be used to improve sniffer's accuracy detecting occupancy in smaller rooms. However, the correlation between signal strength and distance is affected by a large number of factors: non-uniform radiation pattern of the antenna, multi-path propagation (highly significant due to the short wavelength), scattering and path loss due to obstacles[12], etc. Finding and estimating the weight of each factor in our calculation is quite difficult and challenging. Accuracy estimation can really go wrong if the scenario has a unique setup we have not counted since scratch.

To estimate the location of an AP, we need first to measure RSS (received signal strength). We decided to measure RSSI on MAC OS X by using a built-in tool and localize each access point (x, y) based on its respective signal strength[13].



```
'305fifa' <33303566 696661>, bssid=3c:37:86:b3:0c:21, channel=[2 (20 MHz, Active)], cc=US, type=11n, rssi=-52, rsn=[mcast=aes_ccm, ucast
wpa=(null), wep=no, ibss=no, ph=no, swap=no, hs20=no,

'305fifa-5G' <33303566 6966612d 3547>, bssid=3c:37:86:b3:0c:20, channel=[153 (80 MHz, Active)], cc=US, type=11ac, rssi=-54, rsn=[mcast=a
caps=0xc, wpa=(null), wep=no, ibss=no, ph=no, swap=no, hs20=no,

'Campus Circle' <43616d70 75732043 6972636c 65>, bssid=74:3e:2b:06:92:f8, channel=[1 (20 MHz, Active)], cc=(null), type=11n, rssi=-34,
ph=no, swap=no, hs20=no,

'Campus Circle' <43616d70 75732043 6972636c 65>, bssid=74:3e:2b:06:92:fc, channel=[36 (40 MHz, Active)], cc=US, type=11ac, rssi=-42, rsn
swap=no, hs20=no, |

'Campus Circle' <43616d70 75732043 6972636c 65>, bssid=74:3e:2b:06:95:c8, channel=[11 (20 MHz, Active)], cc=(null), type=11n, rssi=-35,
ph=no, swap=no, hs20=no,

'Campus Circle' <43616d70 75732043 6972636c 65>, bssid=74:3e:2b:06:95:cc, channel=[157 (40 MHz, Active)], cc=US, type=11ac, rssi=-49, rs
swap=no, hs20=no,

'Campus Circle' <43616d70 75732043 6972636c 65>, bssid=74:3e:2b:06:9b:6c, channel=[149 (40 MHz, Active)], cc=US, type=11ac, rssi=-89, rs
swap=no, hs20=no,

'Campus Circle' <43616d70 75732043 6972636c 65>, bssid=74:3e:2b:06:9e:b8, channel=[6 (20 MHz, Active)], cc=(null), type=11n, rssi=-66,
ph=no, swap=no, hs20=no,

'Campus Circle' <43616d70 75732043 6972636c 65>, bssid=74:3e:2b:06:9e:bc, channel=[149 (40 MHz, Active)], cc=US, type=11ac, rssi=-74, rs
swap=no, hs20=no,

'Campus Circle' <43616d70 75732043 6972636c 65>, bssid=74:3e:2b:06:a1:3c, channel=[36 (40 MHz, Active)], cc=US, type=11ac, rssi=-82, rsn
swap=no, hs20=no,

'Campus Circle' <43616d70 75732043 6972636c 65>, bssid=74:3e:2b:06:ae:9c, channel=[44 (40 MHz, Active)], cc=US, type=11ac, rssi=-88, rsn
swap=no, hs20=no,
```

Figure 11. RSSI scan on MAC OS X

Now knowing this data we can predict its location by localization algorithms (Equation 1.1) with centroid/weighted centroid (Equation 1.2, 1.3). To locate the AP at the averaged location, multiply locations (x, y) with centroid/weighted centroid[14].

$$\text{Average location: } \left( \sum_{i=1}^{N} w_i x_i, \sum_{i=1}^{N} w_i y_i \right) \quad \text{(Equation 1.1)}$$

$$\text{Centroid: } w_i = \frac{1}{N} \quad \text{(Equation 1.2)}$$

$$\text{Weighted centroid: } w_i = \frac{SNR_i}{\sum_{j=1}^{N} SNR_j} \quad \text{(Equation 1.3)}$$

We will collect data and calculate accuracy of the algorithm after finishing ESP-8266 setup. Despite its simplicity, this algorithm has a relatively high accuracy. If we have access to school buildings we can collect data and hopefully derive a correlation with better overall accuracy, but for now we stick to the localization algorithm.

# 4. Cost and Schedule

## 4.1 Cost Analysis

Fix cost is estimated to be $40/hr, 10hr/week for three people.

$$3 * 40 \frac{dollars}{hour} * 10 \frac{hours}{week} * 8 \text{ weeks} * 2.5 = \$24000 \qquad \text{(Equation 2)}$$

Table 9. Cost of Parts

| Part | Cost (Prototype) |
|---|---|
| Wifi-IC (Expressif; ESP8266) | $6.95 |
| PCBs (PCBWay) | $5.0 |
| Microcontroller (Digikey; ATMEGA328P-PU ) | $2.08 |
| 1.2 V Battery * 5(Amazon; Tenergy 10000mAh NiMH D Battery) | $26 |
| Voltage Regulator (Digikey; LM1117MPX-3.3/NOPB) | $0.44 |
| Serial flash (Mouser;TC58CVG1S3HRAIG ) | $4 |
| Phototransistor | $0.62 |
| Resistors, capacitors, LEDs, sockets | $5 |
| Total | $50.09 |

The labor cost in Equation 2 and the cost of  parts yield a total cost of $24050.09.

## 4.2 Schedule

Table 10. Schedule

| Week | Brandon | Songtao | Zihong |
|---|---|---|---|
| Week 1 | Assemble phototransistor circuit | Finish setting up ESP-8266 | Begin on version1 voltage regulator schematics |
| Week 2 | Test phototransistor's | Finish coding wifi sniffer | Test battery life |

| | | | |
|---|---|---|---|
| | sensitivity, ensure acceptable readings | and verifying correctness by printout | |
| Week 3 | Revise phototransistor circuit components/Design switch mount | Develop an algorithm to estimate distances of APs from the sniffer | Test functionality of voltage regulators |
| Week 4 | Assemble switch mount | Develop an algorithm to estimate distances of APs from the sniffer | Version 1 PCB Design |
| Week 5 | Incorporate finished PCBs and servo motor into switch mount | Test accuracy of algorithm | Test functionality of servo motor |
| Week 6 | Switch mount revisions and longevity testing | Collect real time data from wifi sniffer | Test interface between servo motor and ATmega328P |
| Week 7 | Incorporate remaining components into switch mount | Collect real time data from wifi sniffer | Bugfix on problems occurred during previous tests |
| Week 8 | Prepare Demo | Prepare Demo | Prepare Demo |

# 5. Ethics and Safety

Our design follows the IEEE code of ethics Rule 1: “to hold paramount the safety, health, and welfare of the public, to strive to comply with ethical design and sustainable development practices, and to disclose promptly factors that might endanger the public or the environment”[15]. A number of ethical issues arise from Wi-Fi tracking. Privacy, property and accuracy are three ethical issues we try to solve in the design of our project. The biggest concern of our design is definitely privacy issues. You can know someone’s location based on his/her device(s) as MAC addresses are scanned and picked up by the mini sniffer. This collection of individuals’ information can be used illegally, such as predicting his/her behavior to track individual daily movements. To follow the ACM Code of Ethics principle 1.6[16], we must respect the privacy of these data and reduce vulnerability risk. Therefore, we will use an SHA-256 hashing function from the SHA-2 family[17] to pseudonymised MAC addresses of personal devices and delete part of the digits of the hashed MAC address[18]. To avoid storing and holding user’s information, the sniffer deletes data entry (MAC address, channel, etc.) once a device disconnects from the AP. In addition, we will try to get consent from school/organization to access device information. As a promise, we will not provide data to any third parties to violate ownership of personal information.

Accuracy is another concern that arises from Wi-Fi tracking. To ensure authenticity, fidelity and accuracy of data collected and processed, we will test our programs to minimize errors before releasing the product (refer to section 1.3 High-Level Requirements and 3.3 Localization Algorithm)[19].

Our testing and debugging techniques follow the IEEE code of ethics Rule 9, “to avoid injuring others, their property, reputation, or employment by false or malicious action”[20]. Working in an electronic lab is very challenging and dangerous, so we will always work in pairs and take care of each other in the lab to avoid possible false actions. We also finish lab safety training and understand the lab manual before we attend the labs and use the equipment. Before we test the motor to switch the light, we would use self adhesive bandage wrap to wrap the horn attached to the motor to avoid hurting someones’ finger while the horn is rotating. In addition, during our testing, we will use a thermometer to ensure different electronic parts in our device stay below safe temperature to prevent possible overheating. We will also connect resistors in parallel with certain ICs or voltage regulators when we do unit testing to avoid shorting the circuits and burning the ICs.

For better collaboration during the testing phase, our team will follow the IEEE code of ethics Rule 7[21], we adhere to “seek, accept, and offer honest criticism of technical work, to acknowledge and correct errors, and to credit properly the contributions of others”. We will improve our project and fix possible errors with TAs’ and professor’s feedback and suggestions.

The primary goal of our project aligns with ACM Code of Ethics principle 1.1[22], which is “contribute to society and to human well-being”.  We save power in commercial buildings or teaching buildings by keeping the lights off when people forget to turn the lights off when they leave the rooms or when the luminance in the room is enough without lights on.

# 6. References

[1] Helen. 2019. Choosing The Right Motor For Your Project – DC Vs Stepper Vs Servo Motors. [online]. Available: https://www.seeedstudio.com/blog/2019/04/01/choosing-the-right-motor-for-your-project-dc-vs-stepper-vs-servo-motors/. [Accessed: Apr. 2, 2020].

[2] ProtoSupplies. n.d. Servo Motor Micro SG90 - Protosupplies. [online]. Available: https://protosupplies.com/product/servo-motor-micro-sg90/. [Accessed: Apr. 5, 2020].

[3] George, L., 2014. Interfacing Servo Motor With Atmega32 Atmel AVR Microcontroller. [online]. electroSome. Available: https://electrosome.com/interfacing-servo-motor-with-atmega32-microcontroller/. [Accessed: Apr. 1, 2020].

[4] REcommended Light Levels (Illuminance) for Outdoor and Indoor Venues. [online]. Available: https://www.noao.edu/education/QLTkit/ACTIVITY_Documents/Safety/LightLevels_outdoor+indoor.pdf [Accessed: Apr 12, 2020]

[5] Mellis, D., Maw, C. and Nugent, R., 2008. Arduino - Setting Up An Arduino On A Breadboard. [online]. Arduino.cc. Available: https://www.arduino.cc/en/Main/Standalone [Accessed: Mar. 4, 2020].

[6] Heylen, T., 2017. $2 Arduino. The ATMEGA328 As A Stand-Alone. Easy, Cheap And Very Small. A Complete Guide.. [online]. Instructables. Available : https://www.instructables.com/id/2-Arduino-the-ATMEGA328-As-a-Stand-alone-Easy-Chea/. [Accessed Mar. 4, 2020].

[7] eXtreme Electronics. 2010. Servo Motor Control By Using AVR Atmega32 Microcontroller. [online]。 Available: https://extremeelectronics.co.in/avr-tutorials/servo-motor-control-by-using-avr-atmega32-microcontroller/. [Accessed: Apr. 15, 2020].

[8] Burnette, R. 2016. *ESP8266 Mini Sniff*. [online] hackster.io. Available: https://www.hackster.io/rayburne/esp8266-mini-sniff-f6b93a. [Accessed: Apr. 12, 2020]

[9] Enocean-alliance.org. 2016. *Wireless Lighting Controls: A Total Cost Analysis*. [online]. Available: https://www.enocean-alliance.org/wp-content/uploads/2016/11/Whitepaper__wireless_lighting_controls_payback.pdf. [Accessed: Apr. 1, 2020].

[10] Chai, C., Tizora, M. and Darmamulia, S., 2020. *MODULAR AUTONOMOUS HOME LIGHT*. [online]. https://courses.engr.illinois.edu. Available: https://courses.engr.illinois.edu/

ece445/getfile.asp?id=16880. [Accessed: Mar. 18, 2020].

[11] ECE ILLINOIS Room Reservations. [online]. Available at: https://reservations.ece.illinois.edu/ECE/BrowseFacilities.aspx. [Accessed: Apr. 2, 2020].

[12] Rajaraman, R. 2010. *Antennas & Propagation*. [online]. Northeastern University. Available: http://www.ccs.neu.edu/home/rraj/Courses/6710/S10/Lectures/AntennasPropagation.pdf. [Accessed: April 15, 2020]

[13] Zhan, Y. 2016. *WiFi Localization*. [Online]. Github. Available: https://https://github.com/aaaahern/WiFiLocalization. [Accessed: April 16, 2020]

[14] Han, D., Andersen, D.G., Kaminsky, M., Papagiannaki, K., and Seshan, S. 2009. *Access Point Localization Using Local Signal Strength Gradient*. [online]. Carnegie Mellon University. Available: https://www.cs.cmu.edu/~dga/papers/han-pam2009.pdf. [Accessed: April 16, 2020]

[15] ieee.org. “IEEE Code of Ethics”, 2020. [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html. [Accessed: Feb. 12, 2020].

[16] Acm.org. 2018. The Code Affirms An Obligation Of Computing Professionals To Use Their Skills For The Benefit Of Society. [online]. Available: https://www.acm.org/code-of-ethics [Accessed: Apr. 14, 2020].

[17] En.wikipedia.org. 2020. SHA-2. [online]. Available: https://en.wikipedia.org/wiki/SHA-2. [Accessed: Apr. 16, 2020].

[18] Runia, L. 2019. *Every step you take: three privacy challenges surrounding Wi-Fi tracking*. [online]. Available: https://www2.deloitte.com/nl/nl/pages/risk/articles/three-privacy-challenges-surrounding-wifi-tracking.html. [Accessed: Apr. 3, 2020].

[19] Mcnamee, A. 2005. *Ethical Issues arising from the Real Time Tracking and Monitoring of People Using GPS-based Location Services*. [Online]. University of Wollongong. Available: https://ro.uow.edu.au/cgi/viewcontent.cgi?article=1003&context=thesesinfo. [Accessed: Apr. 10, 2020].

[20] ieee.org. “IEEE Code of Ethics”, 2020. [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html. [Accessed: Feb. 12, 2020].

[21] ieee.org. “IEEE Code of Ethics”, 2020. [Online]. Available: https://www.ieee.org/about/

corporate/governance/p7-8.html. [Accessed: Feb. 12, 2020].

[22] Acm.org. 2018. The Code Affirms An Obligation Of Computing Professionals To Use Their Skills For The Benefit Of Society. [online]. Available: https://www.acm.org/code-of-ethics [Accessed: Apr. 14, 2020].