Lights Out

Authors: Diva-Oriane Marty, Malavika Krishnamurthy, Ryan Gess: Electrical and Computer Engineering, Carnegie Mellon University

***Abstract*—Lights Out is a system capable of reducing energy consumption due to light usage in shared work spaces and labs. It does so by dividing up large work spaces into individually lit work areas and automatically turning lights on and off based on human activity. It is a more advanced motion detection automatic light system. Our system uses sound and motion sensors to detect human presence. Other affordable automatic light systems rely solely on a few motion detectors which cause them to consume extra energy by turning all lights on in a room and leave them on even if all people exit and fail to stay on in common cases, such as when there is a group of people talking, but not moving.**

***Index Terms*—Energy Consumption, Lighting, Microphone, Motion Detection, Occupation Sensing, Open Work Space, Passive Infared Sensor**

1 INTRODUCTION

According to the Commercial Building Energy Consumption Survey, commercial buildings consume 2.5 kWh per square foot [1]. This can be reduced by decreasing the consumption of light energy. Often students working in classrooms and labs don't need the whole room to be lit up to comfortably complete their work. The spaces they work in are broken up into cubicles, desks, or lab benches. When only a specific delimited space(s) is being used, turning on the whole room is a waste of energy. According a Meta-Analysis conducted by the Lawrence Berkeley National Laboratory [2] the average energy saving potential of occupancy detection controlled lighting systems is 24% for Commercial Buildings. Motion detection lights are often used to ensure that only the necessary lights stay on. However, a primary problem with current motion detection lights is that they require movement and thus shut off when people are working and not actively moving around the room to trigger the motion detectors. This can cause the lights to turn off at inappropriate times. The most simple solution is just to keep the lights on for a relatively long period of time, (i.e. 20 minutes), however this can be very wasteful and is not an efficient solution.

Lights Out is a solution that aims to reduce energy consumption due to active lights in open work spaces such as labs or open floor offices. Lights Out is an enhanced automatic lights system that uses motion detection sensors and microphones to ensure lights are only on in specific occupied section(s). This solution targets classrooms, labs, and office spaces as it caters to spaces with predefined and delimited work areas. This solution uses only motion detection sensors and microphones, which are relatively low cost sensors when compared to alternatives such as thermal cameras. These sensors also collect relatively autonomous data when compared to cameras. Another goal of this product is to provide an inexpensive solution that allows for relative anonymity to be maintained.

2 DESIGN REQUIREMENTS

Our requirements are broken into several parts to best fit our use cases, namely: correctness, accuracy, latency, scalability, cost effectiveness, and privacy. Since one of the primary goals is to improve the correctness of a motion detection light system, i.e. more correctly identify human presence, it is imperative that we attain better correctness than current motion detection light systems. Since we are breaking up the room into work areas it is not only necessary to correctly detect presence, but also accurately locate which work area(s) people are occupying. In order for our solution to be useful and user friendly, the latency for human presence to be detect or not detected must be reasonable. A goal of our product is to cater to various open work spaces. Each work space has slightly different needs so it is imperative that our product is scalable. To make this solution attractive for these kinds of spaces it must be cost effective, not much more expensive that current motion detection light systems. Lastly, since we are recording human behavior it is important to keep in mind privacy and ethical concerns.

A *Correctness*

The goal is to reduce energy consumption in proportion to the unoccupied space in a room (50% of room is occupied = 50% energy consumption). In order to achieve this goal we must ensure a certain level of correctness. Perfect correctness would mean that the lights are only on and remain on when a given area is being occupied. We want to achieve better correctness than what is already attained with simple motion detection lights. For this we must achieve correctness in the simplest case: anytime there is continued noticeable movement. In addition to this basic case, we need to achieve correctness when there is noticeable sound. Noticeable sound would be approximately 60db, which is the average noise level of a conversation. [3] However, it is import to note that in these spaces it is common to have people working at desks, so we want to be able to detect common work noises such as typing. We want to be able to detect these work noises the majority of the time. Hence, for this low movement and low noise case we want at least 50% accuracy. Low movement is anything less than walking or waving arms, it could be scratching your nose or turning to look at a phone. Low noise is anything around 40db, such as moving papers or typing. In these work environments it is relatively rare for a person to be completely still and making no noise for extended periods of time. Hence, we acknowledge that we may not always be able to account for the no noise no movement case, but don't feel like it is detrimental for our use case. We are most concerned with cases where lack of correctness causes users to be interrupted from their tasks and have to manually turn back on the lights.

B *Accuracy*
We want to be able to detect individual within a 1 meter radius and turn on the lights around them. A 1 meter radius is reasonable for our use test case where each work area is 4 meters wide. Also based on the placement of the lights in our lab it is a reasonable assumption that this requirement would provide adequate lighting to work. Accuracy is important in order to correctly locate human presence and not turn on unnecessary lights. Correctness and accuracy differ in that correctness is related to when the lights are on and accuracy is related to which lights are on.

C *Latency*
Latency is in regards to how fast the system recognizes that a person has entered an area and turns on the lights. We want to at least maintain the latency of a simple motion detection light system. From our own use cases, 2 seconds is a reasonable time expectation to locate and communicate that a person has entered an area, and to turn on the lights accordingly. In addition it is reasonable, because it would take a person at least two seconds to find a light switch and turn on the lights manually.

D *Scalability*
The solution should be easily adaptable for different room sizes and different number of work areas. The solution should be modular and the sensor setup should be specific to a certain square footage.

E *Cost Effectiveness*
We cannot directly compare the cost of our prototype to that of motion detection light system as we are using low cost LEDs and very simple fixtures to mimic the lighting setup of the lab. However, since our primary additional costs come from the additional sensors and the Raspeberry Pi, we can determine our cost effectiveness by insure that they remain under the a certain threshold. With the cost of microphones being under $7 for our MVP, and potentially cheaper microphones could be used, we find that our additional sensors produce negligible additional cost. A motion detection light switch alone seems to go for anywhere between $11 to $50. The goal would be for our prototype to cost close to the lower end of this range.

F *Privacy*
In order to ensure privacy we want to keep our data offline. The system should not store raw sensor data, but rather it should store the resulting behavior based on a given type of feedback from the sensors. Feedback from the microphones should not get saved as sound waves, but rather a binary signal, that shows whether or not it is meeting the sound threshold to detect potential human presence.

3 ARCHITECTURE OVERVIEW

3.1 Sensor Setup: Guide

The system uses two types of sensors: PIRs (Passive Infrared Sensors), which are standard in motion detection light systems, and microphones. The setup is depended on the shape and size of the room. Figure 1.b shows an example setup of a lab. The following properties must be taken into account.

- No/Minimum overlap between PIRs
  (Each PIR can be set to have a 120 degree cone shaped detection area with a radius set to between 1 meter and 6 meters. Set the PIR radius and position appropriately to attain these conditions for detection area.)
- All desk areas are covered by a PIR
- Each desk area is covered by a microphone

Each work area is associated to a minimum of 1 PIR and 1 microphone, more if the size/ formation of the space requires it. Each sensor communicates back to the Raspberry Pi base station through a Node MCU micro controller, which sends data over WiFi. In locations such as open floor offices and labs, plugs are available so we are generally not concerned with finding a power source or battery life.

3.2 Sensor Setup: Lab Test Case

The particular lab we are using for our test case has 4 general work areas that cover the two sides of the room. The room is 8x4 meters. Each work area is about about 4 meters long, and < 2 meters wide. The width is not a precise measurement as people move around when in the work area, i.e. chairs can be pushed further out. In order to ensure the work areas are covered, we positioned 2 PIR sensors per side of the room. Each work area has 1 PIR sensor that is unique to them. We want to ensure that we cover the 4 meter diameter along the sides, without interacting with the PIR sensors on the opposite side. In order to do so we attached our PIR sensors above the desks at a 60 degree angle from the horizon. Our PIR's sensing radius is approximately 2 meters. We also ensure that there is one microphone positioned in the middle of each work area. Each work area has its own LED strip. See figure 1 for a birds eye view of our lab test case.

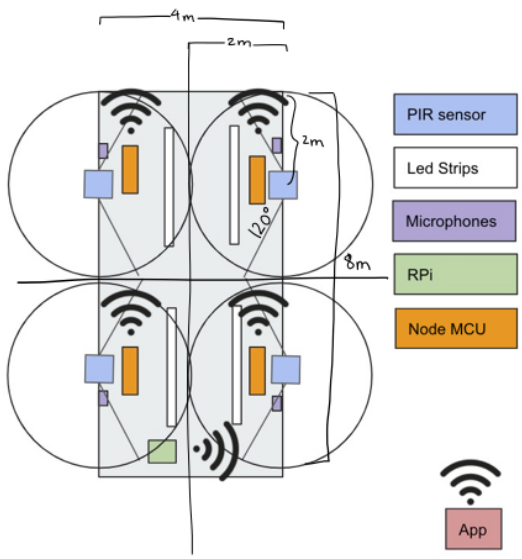


Figure 1: Sensor Setup in Lab Test Case with 4 benches, room size: 8 m x 4 m

3.3 Overall System Interaction

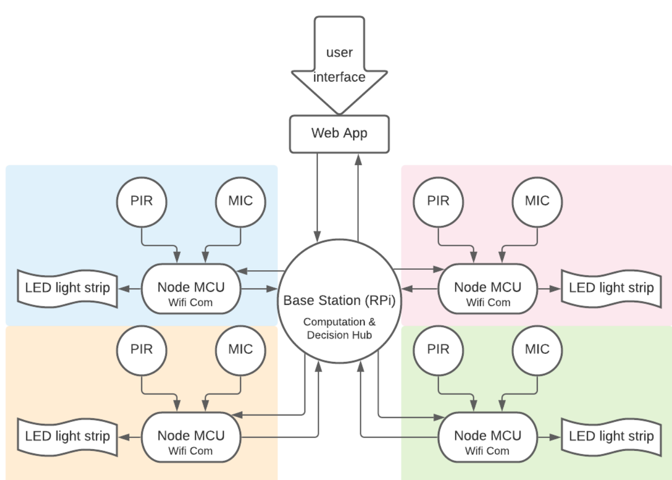


Figure 2: System picture: Overall system

The base station, i.e. Raspberry Pi, is the computation and decision making hub of the system. The NodeMCU (micro controller) at each workstation receives a binary signal from both its associated PIR and microphone. The NodeMCU transmits these binary signals to the base station over WiFi communication. The base station recognizes what workstation sent them a given incoming sensor signal(s) and takes it into account when computing whether or not that workstation's light should be on or off. The NodeMCU receives the decision from the base station and sends a corresponding high or low signal to its associated LED. The base station also receives user input through the web application that impacts its decisions. For example, based on a user's specific needs, the user can block the use of certain a sensor type of data, or overwrite the program and force the lights on. The web application also reflects the current state of the system. The overall system interaction described in this section is further depicted in Figure 2. In the System Description Section (Section 5) the base station's computation method and the overall WiFi communication method are further explained.

3.4 Work Area

A work area is defined as a desk or lab bench and its surrounding typical use space. When defining a work area, we try to ensure that its area does not overlap with another work area. However, in the case were sensor data is unclear, the light in multiple work areas could be turned on. The algorithm detects human presence in a given work areas as an independent process. Previous work shows that detecting a person's path with PIR sensors become highly unreliable once there are 4 people in a room [5]. In addition, in our particular case we are not concerned with how many people are at each work area, nor do we care exactly how they moved around the room. We are just concerned with whether or not at least one person is in a given work space. Additionally, human presence in a given work space is often independent from human presence in another work space. Only case were path tracking could be potentially useful is when a person moves from one space to another, or leaves all together and we can guarantee that no other people entered the space as they were leaving. With the types of sensors and the data we are collecting from them, we cannot tell if one or more people are triggering them. Hence, path tracking is not beneficial.

3.5 Web Application

Sometimes the needs of a work area may vary from the default system, so the web application allows users to modify the lighting system based on their own preferences as opposed to allowing the system to automatically determine which work stations should remain on. Moreover, over 90.6% of respondents from our user survey mentioned that they would like to have the option to remotely control the motion detecting light system and manually override the automation.

The web application provides users with three general functionalities when controlling the overall system: a) forcing on a specific (or multiple) light(s) at a work station even if the system does not detect a presence at the specified location, b) preventing the system from using either PIR sensors or microphones when localizing signals to determine the location of a person in the room, and c) specifying the respective weights of the PIR sensor and microphone the user wants the system to use when computing the individual work station values. In addition to this, the web server displays a graphical user interface of an abstracted version of every work zone as a box. This box is black when the respective light is inactive, and becomes highlighted as yellow when the work station light is turned on.

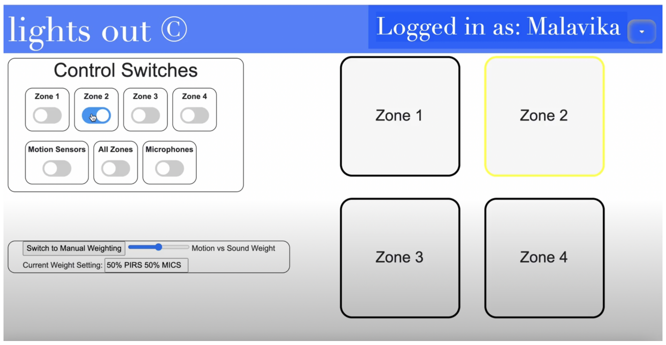


Figure 3: The main controls page of the web application. The right half is the GUI, zone 2 is yellow as it is currently being forced on by the website using the control panel on the top left. Bottom left panel shows weight setting controls.

The web application was created using Django, a python based web framework to develop websites and was hosted locally. The application uses a combination of Python, JavaScript, HTML, and CSS to develop all the functionalities and aesthetics. The server uses the MQTT protocol to establish a bidirectional or two way communication channel with the Raspberry Pi to implement all the features. It does so by subscribing to the Raspberry Pi as a web client, for which the Pi acts as a broker. By subscribing to a specific topic, namely "pir/data/baseStation", the web server is able to read all messages published to that topic by the Raspberry Pi and parse them. Similarly, when the web server needs to communicate to the RPi, it will simply publish messages to a separate topic named "pir/data/web" from which the RPi reads and to which it is subscribed. The connection between the central hub and web client is established on a specified port using the websockets listening protocol.

Upon accessing the web server, new users will be directed to a Log In page where existing account holders can sign into their work space and control the system installed in their own area. The web server uses a Google API, namely the OAuth 2.0 protocol, to authenticate and validate users. User input is also sanitized to ensure a basic level of web security and to defend users from malicious attackers. New users that do not already have an account can navigate to the Registration page where they can input their names, email address, and passwords to instantiate a new account and proceed to log into the web server and access the main controls page.

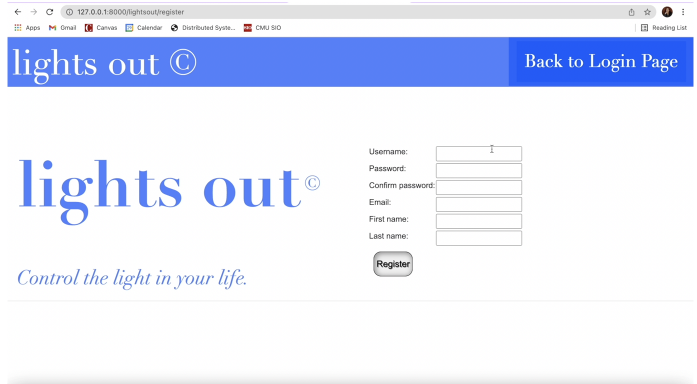


Figure 4: The registration page of the Lights Out web application where new users can make an account

An authentication service on the web server is necessary to safeguard user privacy and ensure that outside third parties do not gain control of another user's lights and system settings.

A *Manual Override of Lights*

The motivation behind allowing users to force on certain work zones was meeting the market demand and preferences for such a feature and allowing a more personalized system setting. This feature will also mitigate the use case scenario where an individual near a light fixture is neither moving nor making noise, thus not triggering the sensors of the system. Since this is a difficult situation to localize and individual, a manual override feature will ensure that a light fixture of the user's choosing will remain on regardless of the system's failure to detect the presence of an individual.

The functionality works by sending a message to the Raspberry Pi using the MQTT communication protocol upon the triggering of a button for a specific work zone. The message "STATION:2:ON" to the Raspberry Pi will notify the central hub that the user wants work zone 2 to turn on. The RPi will then send a message to the appropriate thread for a work station and set a boolean Force On flag to true, which is periodically checked and the corresponding LED strip will be turned on if this value is set to true. Similarly, users can toggle the switch back to the inactive mode and the system will revert to using its own sensors to determine whether or not to turn on the fixture. There is also a separate button to force on all of the lights at once.

B *Sensor Enabling and Disabling*

The motivation behind including this functionality is mainly for protecting the privacy of users, as microphones can be used to track sensitive information, especially in an individual's personal home. According to the results from our user survey, 37.8% of respondents answered "Yes" when asked if they would have concerns when microphones are used to monitor activity in the room. Allowing users to specify that they do not want the microphone being used through remote control of the web application helps mitigate this risk. The web application sends a message to the Raspberry Pi once again when the button is triggered, notifying it to disregard microphone inputs when walking through the decision making process for turning on a light. While it is not feasible to completely disconnect and turn off a microphone within the scope of our demo, this implementation serves as a proof of concept for mitigating some privacy concerns of potential customers.

C Sensor Weight Setting
Customers will also be able to personalize and tailor the light system to their needs according to their preferences. For instance, users can override the automatic weight detecting mechanism and specify their own weights for the motion sensors and microphones using a slider tool on the web server. These weights are interpreted as percentages (always adding up to one hundred percent), and are published as messages to the "pir/data/web" topic to which the RPi is subscribed. This allows users to effectively make the system entirely sound based, motion based, or any combination thereof.

4 DESIGN TRADE STUDIES

A few options were considered for our design.

A *Communication - Wifi*
For communication between sensors and the Raspberry Pi computation hub, three frameworks were considered: bluetooth, wifi and wires. Because of their ease of setup and all of our application areas will be rooms with a wifi network, it was decided to use Node MCU D1 mini's equipped with an ESP8266 microcontroller which can connect and send messages over a wifi network. While wires are also simple to setup and cheap, their usability is not great as long wires typically get tangled up and it is best to prevent our users from dealing with this. While we can also accomplish our wireless requirement with Bluetooth modules, wifi is more cost effective and the simplicity of setup is comparable. It is important to note however, there are some current complications connecting to CMU's wifi network in the lab so this design may need to be adjusted.

B *Computation - Raspberry Pi*
The Raspberry Pi was chosen as the computation hub because of its CPU specifications and its compatibility with our wifi communication modules. Some NVDIA Jetson models were also considered however our algorithm does not require a GPU, the 4 core Raspberry Pi will suffice.

C *Infrared Sensors*
Passive Infared Sensors (PIR) were chosen as they are the existing technology and the goal of this project is to improve the existing technology. The number of sensors in the room is being increase to allow a greater coverage of the room and these sensors are cheap so this fits under the design requirements. While a video camera could meet the accuracy requirements of the system in terms of localizing people, video cameras are more expensive and added hardware complexity. PIR is a simple hardware and allows a novel algorithm to be developed for human localization.

D *Microphones*
Microphones were added as a supplement to PIR sensors. They are cost effective. Microphones also allow for more privacy and less computation power than video cameras. The microphones can be used in such a way that they only provide a binary signal based on if the sound level is above or below a certain threshold. This allows for a higher level of anonymity and lower computing latency. If we were to use video footage we would have to at least detect figures, which results in less anonymity and higher computing latency. Microphones were added to increase functionality of the system in addition to the accuracy. A requirement for our improved motion detection light system is that we can localize individuals in a low movement environment. It is expected that in these situations there may be some sound (conversation, typing at a desk, etc). Hence with the microphones even when individuals are exhibiting little movement, their relative location can still be determined.

E *Lights - LED Strips*
LED Light strips were chosen to simulate classroom/lab or conference room lights. These allow for prototyping and customization of different room configurations without being constrained to existing classrooms and modifying the existing circuitry or dealing with high amperage.

F *Power*
Since the use case is a classroom/lab or conference room there is typically access to wall outlets throughout the room so it was decided to plug components into the wall instead of using some other power source such as a battery.

G *Human Localization*
Various localization techniques were iterated, including multiple human tracking using PIR, audio beamforming, and the chosen novel solution which combines principles from each of the former techniques and in addition, includes principles from statistics and sensor fusion.

Audio Beamforming
Beamforming is a technique that uses filtering of mi-

crophone signals and constructively combining them together in order to determine the spatial location. Often, delay of signal input is used as an influential feature in determining location [4]. Microphone arrays can be purchased for beamforming and localization however these are much more expensive than setting up our own around the room. We can still achieve our required accuracy at a lower cost. Further, since this technique would perform better with a constant stream of audio input, we are hesitant to use this method for privacy concerns. Upon some initial testing and gathering output from the microphones we decided to use microphones that detect when sound exceeds a loudness threshold instead of recording voices and noises for localization.

Human Tracking using PIR
This approach uses many PIR sensors to track paths of multiple humans in a room. The algorithm relies on two main processing steps and location estimation for path estimation. It also maintains path history. The algorithm performed very well for tracking one persons path in a room. Unfortunately, as the number of individuals increases in the room, the accuracy greatly decreased [5]. Therefore, it was decided to not rely fully on path tracking and this technique to localize in a room as we would like to scale to multiple people in the room.

Lights Out
Our approach combines a few principles from microphone beamforming, path tracking as well as signal weighting to localize humans in the room. This is algorithmically complex and requires multiple threads however it can be performed on a Raspberry Pi. In addition, we are not constrained to determining location by path, however we can use the last location detected as a signal and weight this in our combination of signals.

5 SYSTEM DESCRIPTION

5.1 Sensor and Software Interface: General Setup

As previously mentioned in the section 3.4, for our use case determining human presence per work area is the best fitted solution. Due to the independent nature of the work areas' states, our design has one thread per work area. There is a queue that takes in the sensor data, and sends it to the thread(s) (work area(s)) concerned. The thread structure of the program running on the Raspberry Pi (base station) is depicted in Figure 5.

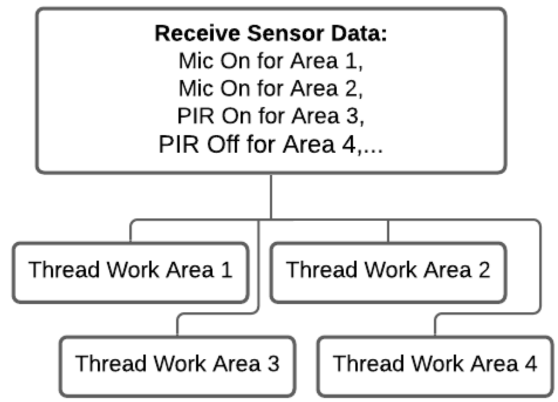


Figure 5: Sensor/Thread Interaction

5.2 Work Area Thread

Each thread handles calculates its own score using the logic depicted in Figure 6. Every second the thread checks if it needs to increase or decrease its score based on incoming sensor data. It then checks if it should turn on or keep the light(s) on, or turn off or keep the lights(s) off based on its score.

5.3 PIR Sensor Detection

A PIR sensor is either on or off. If it is on, then we determine which work area it is associated to. PIRs are unique to a workstation. The PIR sensors are very reliable for detecting any movement. Based on our testing, false positives for PIRs are relatively rare, while false negatives are more common. False positives are due to detecting movement outside of the intended area, while false negatives are due to being outside of the expected desk space, or having too little movement. The PIR's short comings are that they don't detect when a person is not moving, producing false negatives. The microphones seek to help in these cases.

5.4 Microphone Sensor Detection

The microphones seek to reduce the false negatives detected by PIRs due to there being limited to no movement at a workstation. Microphones don't generate these false negatives, but they do produces a lot of false positives. Standalone microphone feedback is less reliable, due to false positive caused by it being hard to distinguish between soft noises near the microphone and stronger noises far away from the microphone. Hence, the microphone sensors primary purpose is to detect continued presence of a person in a workstation when they have stopped moving. This is to avoid common cases such as when the lights turning off when the person is talking, but not moving in an area. A microphone sensor is not used to detect initial presence in an area. It is used to determine if a person is still there.

Microphones are always unique to a work area. The microphone sensor sends a 1 when the incoming sound level is about the sound threshold, and 0 otherwise. The sound threshold is determined by tuning the ohmmeter on the microphone. It is calibrated by hooking up the microphone to the computer and twisting the ohmmeter till we get reasonable sound input, i.e. it is not constantly detecting sound when there is relative silence and is not never detecting when we create noise.

5.5 PIR and Microphone Signal Coupling

When entering a work area, you are sure to set a PIR sensor on. We only consider microphone sensor inputs in the case were a PIR has recently detected movement. This is in order to avoid the case where there is a loud noise in one work area that is detected by the microphones in other non-occupied work areas. We are using this coupling of information to minimize expected false positives from microphone sensors and false negatives from the PIR sensors.

When the PIR is detecting we increment the score by PIRvalue. Since microphones are needed when there is limited motion to no motion, we increment the score by MICvalue when PIR is not detecting.

5.6 Communicating System Decisions

Our communication system uses mosquitto protocol which allows two way communication between all of the components of our system: our base station, web application interface and work stations. Mosquitto is a publish and subscribe protocol meaning each client can publish or send messages to a topic and clients can also subscribe to these topics to receive the messages sent on specific channels. Our base station acts as the broker which means it handles all distribution of messages instead of using a server. This protects our latency requirements by keeping communication local. It also makes scaling up or down the number of stations easy, because the messages are not directly delivered to the devices they are published to topics meaning when another device is added it simply needs to subscribe to the topic instead of directly sending a message to the device. Connections are made between client and broker instead of between client and server or client to client.

For example, in our system, the ESP8266 at work station 1 detects a high signal from a PIR. It then publishes this value to topic A. The Raspberry Pi is subscribed to this topic and can read this message. Based on this value as an input to our algorithm, it is decided that the lights at workstation 1 should be turned on. The Raspberry Pi then publishes a message to topic B and topic C to turn the light stations on. Work station 1 is subscribed to topic B and will output a high value to turn on the lights. Our web application is also subscribed to topic C and thus will reflect work station 1 is on in the web application.

The workstations publish and subscribe to topics that include their unique IP address, this allows the base station to identify a given workstation. This system is easy to scale up and scale down, the Raspberry PI takes care of identifying the workstations as they connect and creating a thread that will control them. The ability to scale up and down was important for this project as it allows the system to be usable in different work spaces with little setup overhead.

Figure 6 color codes the different topics, and shows which elements are subscribed or publishing to which topics.

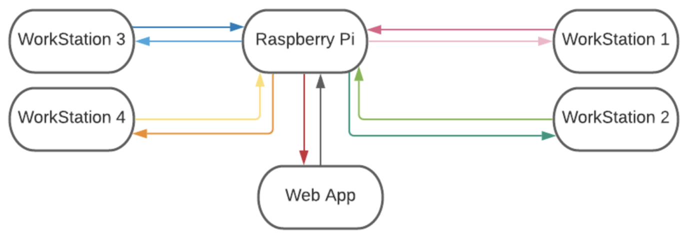


Figure 6: Mosquitto Protocol and Sending Messages: *The colors represent different topic channels, workstation topics contain their unique IP address*

5.7 Calculating the Score

There are two cases in which a workstation thread increases its is score. In all other cases it decreases its score. As a result of the explanation presented in sections 5.3 and 5.4, we first consider the PIR sensor values. Keeping in mind that the Web Application allows users to stop the use of PIRs or/and Microphones, the first step is to check if the PIR can be used. If yes, the second step, is to check if the PIR is on. If the PIR is on, the thread is to increase the score by PIRvalue. Otherwise, if the PIR is off, we must check if the PIR was recently on, if so we check if we can use the microphones, and then if the microphone is on we increase the score by MICvalue. If the PIR was not recently on, we may assume that the microphone value could potentially be a false positive. This is a reasonable assumption, because based on our testing we never found ourselves completely still when talking or working over longer intervals of time. In all other cases we decrease the score. The one exception case is if the PIRs are not allowed to be used, but the microphones are then we ignore all the PIR related information when deciding to increase and decrease the score. The score has an upper bound and lower bound in order to not have cases where the score increases or decreases so much that it is impossible for the system to eventually turn on or turn off in a reasonable amount of time. Please refer to Figure 7 for a visual representation of the score modification logic.

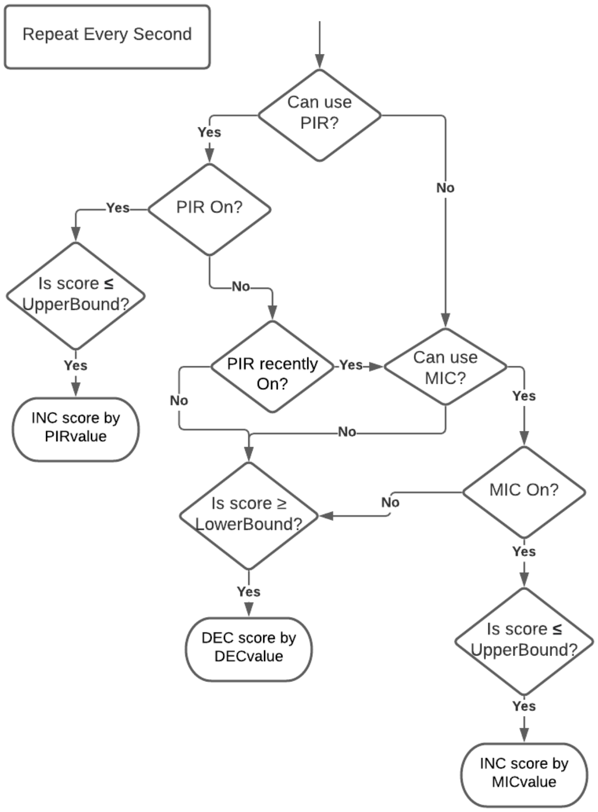


Figure 7: Modify Score

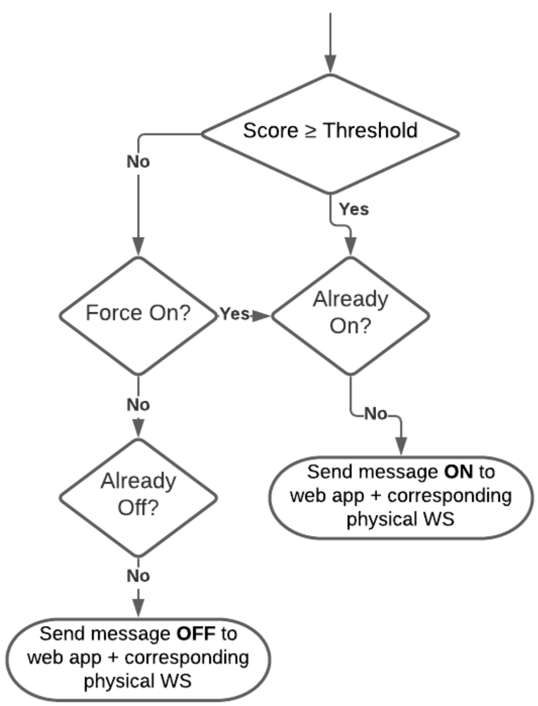


Figure 8: Send On or Off message

5.8 Determining Workstation States

Every time the score is updated, the thread checks if it needs to send a message to turn on or off the workstation. If the score is above the threshold or if the user forced on the workstation through the web application, and the workstation is not already on, the Raspberry Pi sends a message to the web application and a message to the corresponding physical workstation through its NodeMCU. Otherwise if the score is not above the threshold, it is not forced on, and it is not already off, then it sends an off message to the web application and the physical workstation. This is depicted in Figure 8.

The longer a person stays in an area the more confident the algorithm is that they are in the area, and the longer it will take to turn off once the person has left the area. This behavior is limited by the upper bound score in order to avoid extreme cases like the score getting too big it would take forever to decrease to below the threshold and turn off. If a person just passes by a station and sets off the sensors, the score will only increase by a little bite above the threshold and hence it will take less time for it to decrease below the threshold and turn off the lights. However, if a person stays at the station the score could reach the upper bound score and take the maximum amount of time to turn off once the person has left the station.

5.9 Defining Constants

The previously mentioned constants (PIRvalue, MICvalue, DECvalue, threshold score, upper bound score, lower bound score) are defined and explained in this section.

Based on our requirements we want our system to take a maximum of 2 seconds to turn on. Since the score is changed every 1 second. We want the difference between the lower bound score and the threshold score to be 2 * PIRvalue/sec, i.e. worst case scenario would only take two rounds (2 seconds) to turn on once triggered. In order to run multiple tests, but still demonstrate the system's full behavior we decided that we would run 2 minute tests. Within those two minutes we want our system to be able to turn on and off, so we decided that our maximum turn off time would be a quarter of our testing time, and hence 30 seconds. We want the difference between the upper bound score and the threshold score to be (30 - $PIR\_recently\_on\_time\_threshold$) * DECvalue/sec, i.e. worst case scenario would only take 30 - $PIR\_recently\_on\_time\_threshold$ rounds (seconds) to turn off once PIR and microphone are off. These were our guidelines for determining the rest of the constants. In addition, we decided that we would set our lower bound to zero.

The decrease value (DECvalue) is always equal to the changeValue, while the increasing value is equal to the changeValue, but is split between the PIRvalue and MICvalue based on the set weights. The weights give user's control on how sensitive the system is to PIR sensing and microphone sensing based on their use case. The default is that the system is equally sensitive to PIR detection as it is to microphone detection and hence the weights are automatically set to 0.5/0.5 (PIR weight and microphone weight respectively). In order for the default increment for both PIRvalue and MICvalue to be 1, we set the total changeValue to two. Change value is the total potential score change per second.

Since the default PIRvalue is 1 and the max turn on time is 2 seconds, the threshold must be 2. Since the total turn off time is 30 seconds, we decided that the upper bound should be 32, i.e. 30 points above the threshold. We want the microphone sensing to take up at least half the time 15 seconds, i.e. $PIR\_recently\_on\_time\_threshold =$ 15 seconds. Assuming the score value is at the upper bound, in the 15 following seconds the DECvalue should be able to bring the score to be below the threshold and turn off the system. Hence, the DECvalue being 2 is reasonable. It will take 15 seconds of continuously decreasing to bring the score from 32 to 2. The calculations are summarized in the box below.

$$thresh - lbound = 2sec * PIRvalue/sec$$
$$ubound - thresh = (30sec - PIR\_recently\_on\_time\_threshold) * DECvalue/sec$$
$$changeValue = 2$$
$$PIRweight + MICweight = 1$$
$$PIRvalue = PIRweight * changeValue$$
$$MICvalue = MICweight * changeValue$$
$$DECvalue = changeValue$$

$$auto : PIRweight = MICweight = 0.5$$
$$thresh - lbound = 2sec * (0.5 * 2)$$
$$ubound - thresh = (30sec - 15sec) * (2)$$

`if lbound = 0, then ubound = 32 and thresh = 2`

Summary of Constants

A `max turn on time` = 2 seconds

B `max turn off time` = 30 seconds

C `ubound` = 32 (the upper bound score that a client can have)

D `lbound` = 32 (the lower bound score that a client can have)

E `tresh` = 2 (the threshold score at which the light switches between on and off)

F PIRvalue/sec = `PIRweight * changeValue` (increase score by this value while a PIR that is specific to a work area detects motion)

G `MICvalue/sec` = MICweight * changeValue (increase score by this value when a microphone that is specific to a work area detects sound)

H auto `PIRweight` = 0.5 (out of a 100%)

I auto `MICweight` = 0.5 (out of a 100%)

J `changeValue` = 2 (total increase or decrease score by increment)

6 TEST & VALIDATION

A *Latency*

In order to test the latency of our system, we measured the time it took for the lights to turn on at a station when a person entered the monitored area of 1m radius. We filmed a group member enter the area and use the timestamps in the video to determine how many seconds it took for the lights to turn on which is the expected behavior in this case. We hoped to see less than a 2 second latency to maintain the latency of current motion detected systems and this is the approximate amount of time it would take for an individual to physically turn on the lights when they enter the room or station.

Trial	Time (s)
1	1.2
2	0.81
3	1.11
4	0.97
5	1.05
Average	**1.03**

Our results show an average latency less than 2 seconds and each trial less than 2 seconds. We consider this a success as this confirms that the latency of communication and computation of a signal from the sensor output, to the computation hub, back to the

ESP8266 to turn the lights on is approximately 1 second.

B *System Logic and Behavior*
For our system logic and behavior, we tested 4 cases High Motion and High Sound, High Motion and Low Sound, Low Motion and High Sound, and Low Motion and Low Sound. We also tested No Sound and No Motion meaning there is nobody in this work area. These cases were tested with infared and microphones sensors as well as just infared sensors. A success is defined as the system exhibiting correct behavior (on when a person is there, or turns off when a person isn't) for 60 seconds. noS+ noM = no sound + no motion : no activity

	HM & HS	HM & LS	LM & HS
PIRonly	5/5	5/5	3/5
PIR+MIC	5/5	5/5	5/5

	LM & LS	no M& no S
PIRonly	3/5	5/5
PIR+MIC	4/5	5/5

Key
HS = high sound : talking or above
HM = high motion : moving around
LS = low sound : no talking, equivalent to keyboard noises
LM = low motion : no big movements, equivalent level of motion of someone typing

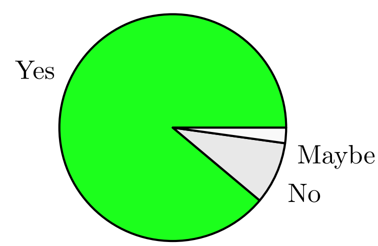
Overall this is a success. For the majority of cases our system exhibited correct behavior. Due to the multiple infared sensors, we are able to pick up movements at a high level of granularity. If a person is sitting and quietly typing, they are still exhibiting some level of motion (taking a drink or getting a book from their bag). Our system is able to pick these motions up that most other systems cannot. Many systems have a sensor at the door that trigger on the lights for a set amount of time and will turn off after this period is over. In addition, even in cases of low sound and low movement our microphones still pick up typing noises and the aforementioned movements so we still have a somewhat high level of accuracy (4/5). In all cases that only infared sensor (PIR) was not fully accurate, our microphones increased the accuracy. We experienced a few more issues with scaling than we expected. Occasionally, when increasing the number of stations, our system gets stuck and we need to reset it. We tried a few things to mitigate this such as waiting for the number of expected work stations to connect before starting. We weren't able to make this perfectly robust however this delay was able to decrease many of the occurrences.

C *Web Application*
In order to test the web application, we tested three main functionalities. We tested forcing the lights on at each station, adjusting sensor weights, and the reflection of the status of the lights (ON/OFF). For each of these cases, we tested for correct functionality in addition to latency. Since there are not many edge cases here, we did 5 trials to ensure these features are functional.

(a) Test 1: Web app toggles lights on at each station, and all lights simultaneously.
 i. 5 successful trials were recorded.
(b) Test 2: Web app adjusts sensor weights to be more and less sensitive. This was validated by checking the scores were reflected on the Pi as well as checking that when sensors values were completely ignored, movement or sound was not detected.
 i. 5 successful trials were recorded.
(c) Test 3: Web app reflects status of lights (ON/OFF). We validated this when we were performing Test 1 (when lights were toggled by the web app) as well triggering the lights by walking around the room. A success would mean the status on the web app reflects the status of the system (i.e. stations 1,2,3 lights on, then stations 1,2,3 are highlighted in web app)
 i. 10 successful trials were recorded.

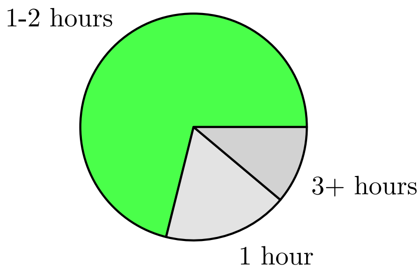
D *User Survey* To inform some of our design decisions, we sent out a survey to students at Carnegie Mellon University to gain a better understanding of their experience with existing motion detection light systems and their concerns about privacy and working behaviors. We surveyed students since our main use case is within classrooms and lab spaces. We asked the following questions and received 45 responses:

(a) Have you ever experienced lights controlled by motion detection turning off while you were still in the room?
 i. **Yes, 40 (88.9%)**
 ii. No, 4 (8.9%)
 iii. Maybe, 1 (2.2%)



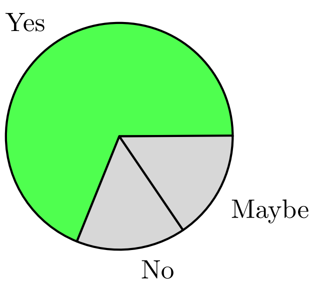
(b) How long would you like them to stay on for when you trigger them? (How long is your average work session?) [If you leave the room and come back that would be two work sessions the lights turn off in between]

i. less than 1 hour, 8 (17.8%)
ii. **1-2 hours, 32 (71.1%)**
iii. 3+ hours, 5 (11.1%)



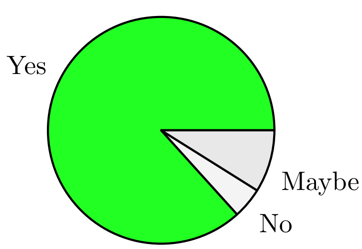
(c) In an effort to conserve energy, would you agree it's a good idea to only turn on lights in occupied sections of a room?

i. **Yes, 31 (68.9%)**
ii. No, 7 (15.6%)
iii. Maybe, 7 (15.6%)



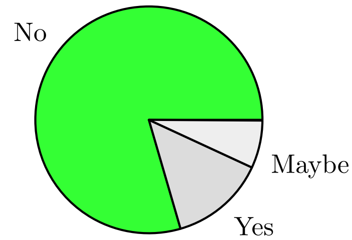
(d) In an effort to conserve energy, would you agree it's a good idea for automated light systems to shut off in a short period of time after a person leaves an occupied section?

i. **Yes, 39 (86.7%)**
ii. No, 2 (4.4%)
iii. Maybe, 4 (8.9%)



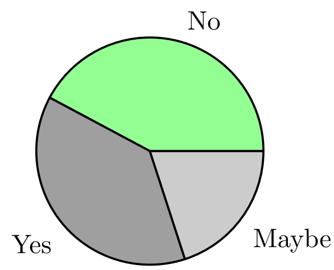
(e) Would you have concerns with **Infared Motion Detectors** monitoring your activity to localize your position in the room?

i. Yes, 6 (13.6%)
ii. **No, 35 (79.5%)**
iii. Maybe, 3 (6.8%)



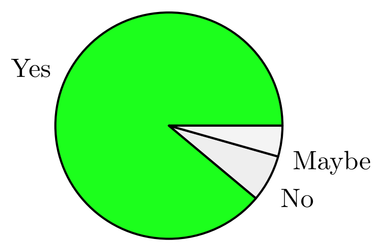
(f) Would you have concerns with **Microphones** monitoring your activity to localize your position in the room? The microphones only capture when noise level exceeds a threshold, they are not recording.

i. Yes, 17 (37.8%)
ii. **No, 19 (42.2%)**
iii. Maybe, 9 (20%)



(g) Would you prefer to have the option to remotely control motion sensing lights to manually override the system at times?

i. **Yes, 40 (88.9%)**
ii. No, 3 (6.7%)
iii. Maybe, 2 (4.4%)



We conclude from the survey results that

(a) Many students have experienced lights turning off on them while they are working. It is very important to make the system accurate in determining whether there is a person in the room. In general, we error on the side of false positives, if we are less confident there is a person in an area, we keep the lights on to error on the side of keeping the lights on when a person is not in the room instead of turning them off when the person is still there. We also incorporate more Infared Sensors and include Microphones to pick up more human activity than current systems in order to localize people in the room.

(b) We also know that the average work session is roughly 1-2 hours. We designed our system to turn off within a specified amount of time when it becomes confident there is nobody in the area. This can be increased decreased based on user needs. For our demo, we set this at 30 seconds to show the lights turning off when a person leaves the room however this can be increased to 60 minutes or longer if needed.

(c) However, students also agree that it is a good idea to turn the lights off in a short amount of time if a person leaves an unoccupied section. Therefore, our system increases confidence proportional to the movement and sound in an area. The more confidence (movement and sound) that is detected, the longer the lights will stay on, otherwise for low movements and low sound the confidence score will stay low. For example, a person entering an area and making a lot of movements and sound then being quite will trigger the lights on longer than a person simply walking through the work space.

(d) Students are less comfortable around microphones than Infared Motion Detectors. We considered this and thus incorporated microphones with a hardware filter to detect when loudness exceeds a threshold as opposed to recording all sounds. We realize we are asking a relatively educated group on the vulnerabilities of IoT systems and microphones at Carnegie Mellon University and expect with more education about our system we can alleviate these concerns. We feel our system is secure and protects privacy more than a system that uses microphones that record all sounds. In hindsight, an additional question: Would you feel more comfortable with microphones that detect sound surpassing a threshold than microphones that record all sounds? would have been more informative.

(e) Most students also enjoy the option to remotely control the lights and manually override the system so we included functionality within our web application to adjust privacy settings and manually turn lights on or off.

7 PROJECT MANAGEMENT

7.1 Schedule

Our schedule and timeline is reflected in figure X at the end of the document on page X in the form of a Gantt chart. The first couple of weeks of our semester were devoted to brainstorming and formalizing a project idea in the form of a proposal presentation. After meetings and discussions with our faculty advisers and staff, we settled on the proposal for Lights Out and presented this to our peers. The next week consisted of ordering parts and formalizing our design report to plan the technical details of implementing all our features.

Tasks were divided up according to team members' strengths, described in more detail in the following section. The final schedule presented in this document has been slightly modified from our initial design review schedule. Changes include allotted time for reconsidering the weights of microphones and motion sensors after unit testing, establishing bidirectional communication between the web application and Raspberry Pi, and sending out user survey, all completed in the last remaining weeks. The schedule in the beginning accounted for an extra two weeks at the end of the semester to allow for dealing with unforeseen circumstances and/or setbacks.

7.2 Team Member Responsibilities

The entire project was divided into three main parts: the logic behind the computation of threshold values for each primary work station, communication between the sensors, micro controllers, and RPi, and the web application. Other parts including the circuitry and hardware assembly of the stations were group efforts and contributed to by everyone.

The main three goals were assigned to different team members based on each member's area of expertise and comfort. For instance, Malavika designed the web server portion of the project due to her prior experience from having taken Web Applications and Development in a prior semester at Carnegie Mellon University. Ryan and Diva have strong technical skills from having taken Electronic Devices and Analog Circuits, as well as Distributed Systems for dealing with concurrency and multi threading.

7.3 Budget

A thorough breakdown of our budget and which parts to which different portions of spending were devoted is illustrated in the appendix at the end of this report on Table 1. Though we were never concerned with exceeding our budget, we encountered two main setbacks. The first was that the initial microphones we had ordered from Knowles were not functional nor sensitive enough. We found that after unit testing, the microphone was not detecting noises from typing or speaking at work stations. We order another set of microphones to resolve this issue. Secondly, the pack of 12 motion sensors we had initially ordered never arrived, so we placed an order for another set of 12 motion sensors from the same manufacturer. However, these sensors never arrived. Items that were used in our entire system design but did not need to be purchased (as we already owned them or were provided them at no cost) include an ARM Cortex-A53 Processor Raspberry Pi, and four breadboards.

7.4 AWS Credit Usage

We would like to take this opportunity to thank Amazon for gifting us with AWS credits to use for hosting our web application. Though we did not use this credit as we hosted the website locally and did not host the server on the Cloud, we are grateful for the credits extended to us.

7.5 Risk Management

Throughout the semester there were numerous risks to be taking into consideration and accounted for in the proposal process, design review, and the implementation of the final product. The first major risk we had to take into account was finding a microphone sensitive enough to localize an individual in the room. After meeting with our assigned faculty advisor and teaching assistant in the beginning of the semester, our mentors warned us that it would be difficult to find microphones that sample at high enough frequencies to detect which microphones detected a sound before the other one. In order to localize someone and identify the microphone nearest to them, we would need to compare the recorded times of noise, and our teaching assistant mentioned that it was a difficult task, as he also was faced with a similar issue the previous semester for their group's own capstone project.

In order to mitigate this risk, we were advised to consider using a device with a pre assembled microphone array that was capable of determining which microphone sensed a noise first as opposed to filtering signals on our own end. However, our final solution outlined in this report met our design and system requirements.

We did not face many risks on the budgeting end, as most of our materials were inexpensive or readily available to us at no cost. To ensure that we would not exceed our budget, we made sure to submit a request for the RPis owned by the ECE Department early on, as these were being given out on a first come first served basis. The PIR sensors, LED lights, and Node MCU micro controllers were also relatively inexpensive.

We did face some pressure under time when the first set of microphones we ordered from the manufacturer Knowles was not meeting design requirements. In order to mitigate this we ordered a different set of microphones and proceeded to develop the logic of the system and the PIR integration and communication.

The new set of microphones were also difficult to integrate into the system, as it was difficult to identify an appropriate threshold for the microphone to detect and send digital outputs to the RPi. The design trade-offs discussed earlier regarding this threshold made settling on an appropriate threshold to detect soft enough but ignore distant sounds a precarious task. In order to preserve the functionality of the entire system we decided to prioritize PIR sensor data first, and only increment a workstation's overall value using microphone data if the motion sensors do not currently detect movement but recently did.

To mitigate the risk of running out of time, we allotted multiple weeks towards the end of the semester as a fail safe to account for unforeseen setbacks and to perform last minute testing and to make minor design changes. This is reflected in our Gantt chart at the end of the report.

8 ETHICAL ISSUES

Our primary ethical issue is privacy, which we addressed in the design requirements section. Privacy is an ever present issue when dealing with IoT devices. However, this product is catered towards work environments so privacy issues could be seen as less sever than if this was a Home IoT device. We primarily address this issue by not collecting any identifiable data on the people using spaces in which our product is installed. We do acknowledge however that the time at which the lights are turned on and off could be used to keep track of general patterns that could be associated to all people that often use the space. if they have any concerns with specific types of sensors or if they want to manually control the lights, we allow users to overwrite the system using the web application. We don't believe our system significantly increases privacy concerns for people using a space. However, we tried to better understand user concerns by sending out the survey referenced in section 6. Only 13.6% of those surveyed said they have concerns about PIRs monitoring their activity, while 37.8% said they have concerns about microphones monitoring their activity. 88.9% liked the idea of being able to overwrite the system remotely (i.e. using the web application). Hence, we believe that the control that users have through the web application takes care of a lot of their remaining privacy concerns.

9 RELATED WORK

In the Spring 2021 semester, a group of students developed a product called "StenoPhone" for their capstone project which uses a microphone array to determine which individual is speaking in a room and also to transcribe their speech. The microphone component of this project is similar to "Lights Out" in that it involves the concept of localizing individuals in a room based off of sound. "StenoPhone" employs the technique of beamforming, or spatial filtering of audio input on the mic array's firmware. While we do not perform any filtering of digital signals, we localize individuals based off of which separate microphones they trigger instead of a central microphone array.

Similarly, in the Fall 2020 semester, a group of students (including our teaching assistant) developed a product called "iContact", also using a microphone array for localization of humans. The project aimed to create a more immersive viewing experience for users on video sharing platforms such as Zoom and Skype by automatically focusing the camera on the speaker in the room. In the same spirit of the previously mentioned project, "Lights Out" also uses sound to localize and identify where in the room an individual is located, but uses a dispersed network of microphones and sensors as opposed to an array with the built in functionality to do so.

10 SUMMARY

Overall our system met our design requirements. It met our correctness requirement by detecting human presence with more granularity than current motion detection light systems, particularly in the case where people are not moving but are talking (low movement, high noise), in addition to the case of low movement and low noise. Our system required less than 2 seconds to turn on once a person entered a workstation, and correctly distinguish between activity at different work stations. It met our scalability requirement by not having to make changes to the program or existing setup when adding or removing workstations (the base station can identify workstations as they connect). For the most part, latency remained low with the addition of work stations however sometimes our threads would get stuck and take longer to turn on or off. Typically this was resolved with a complete initialization and reset. It approximately met our cost requirements. Table 2, lays out the overall cost of a single workstation. Without including the light strip itself (which is only needed for prototyping) a workstation costs $13.76. This is close to the lower bound of current motion detection light switches ($11-$50). In addition, we assume the cost of our system could easily be reduced by finding cheaper equivalents of the sensors we used. The sensors we used are typically marketed towards hobby builders, so it is reasonable to assume that they are a little overpriced.

One of the issues we experienced this semester was testing sensors in an inefficient manner. This was particularly an issue with our microphone sensors. On a few occasions we ordered a single type of microphone, test it, and realize it did not meet our exact needs. We would then have to reorder and wait for a whole new set or new model to arrive. This delayed our initial building. Eventually we started placing orders for a couple of different types of microphone at once, so we could test multiple types at once. If future semester students have the extra budget, we would recommend that they start by ordering multiple types of a given part, especially if the spec sheet does not provide enough information. Something we did that we recommend future semesters to do is to leave at least the last couple of weeks to testing, because a lot of small issues reveal themselves during that time. There needs to be sufficient time to fix the issues that may appear during testing. Testing can sometimes also be done periodically, which also insures that the project is on the right track.

References

[1] Independent Statistics Analysis U.S. Energy Information Administration. "Table E6. Electricity consumption intensities (kWh) by end use, 2012". In: *COMMERCIAL BUILDINGS ENERGY CONSUMPTION SURVEY (CBECS)* (May 2016).

[2] Karina Garbesi Alison Williams Barbara Atkinson and Francis Rubinstein. "A Meta-Analysis of Energy Savings from Lighting Controls in Commercial Buildings". In: *Energy Analysis Department Lawrence Berkeley National Laboratory* (Sept. 2011), p. 13.

[3] Centers for Disease Control and Prevention. *What Noises Cause Hearing Loss?* URL: `https://www.cdc.gov/nceh/hearing_loss/what_noises_cause_hearing_loss.html`.

[4] Abdulqadir Alaqeeli Amiri Hamid Hidri Adel Meddeb Souad. "Beamforming Techniques for Multichannel Audio Signal Separation". In: ().

[5] Yuki Kasama Toshiaki Miyazaki. "Multiple Human Tracking Using Binary Infrared Sensors". In: *Sensors (Basel, Switzerland)* (June 2015).

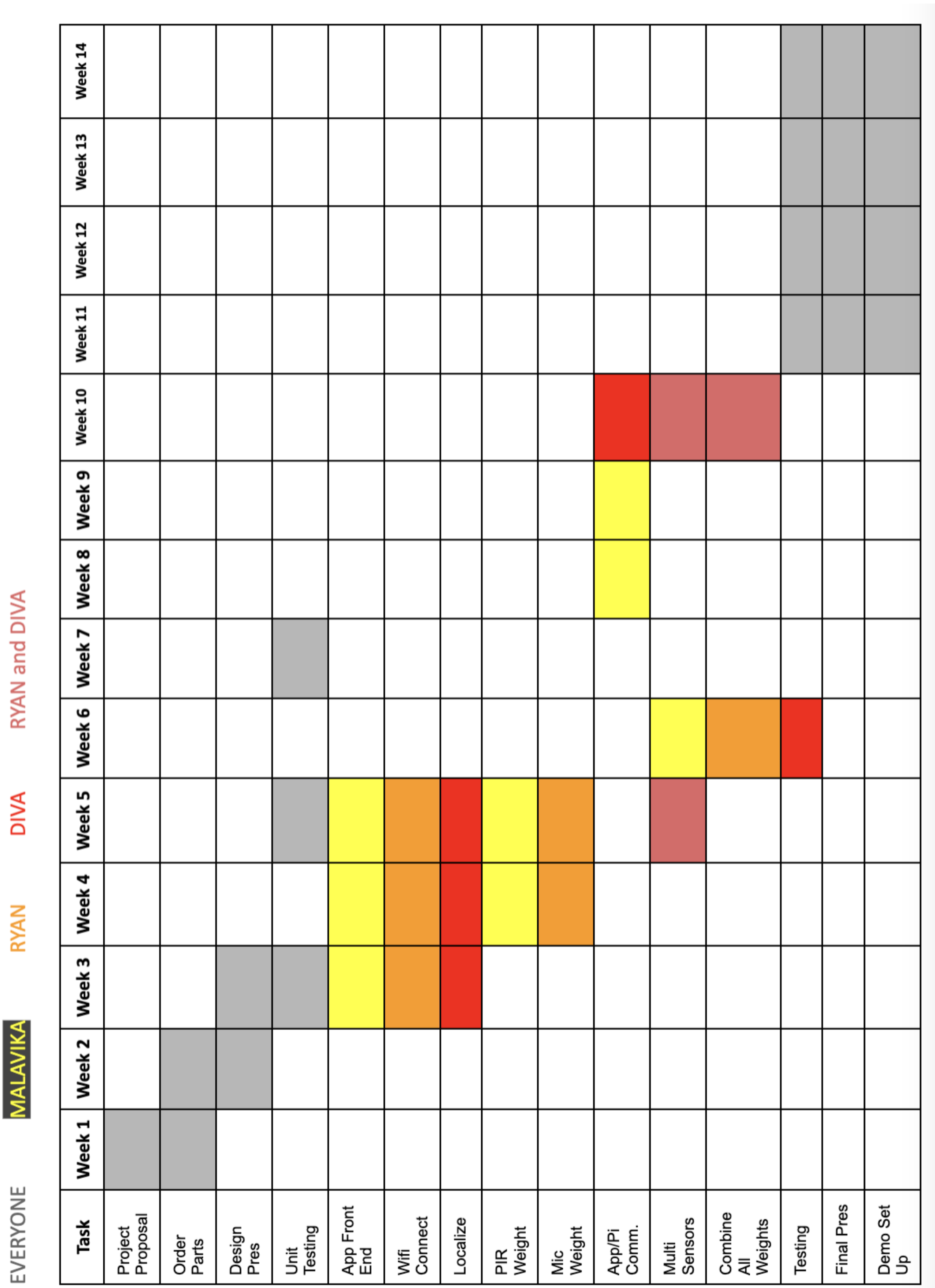
Table 1: Bill of materials

Description	Model #	Manufacturer	Quantity	Cost @	Total
PIR Sensor	1528-1991	AdaFruit	10	$9.95	$99.50
Microphone	SPH0641LU4H-1	Knowles	8	$2.14	$17.12
Raspberry Pi	1690-1028	Raspberry Pi	1	$25.00	$0
LED Lights		KWZM	4	$11.99/2	$23.98
Node MCU	ESP8266	AITRIP	6	$2.66	$15.99
Microphone	SPW2430	AdaFruit	1	$8.58	$8.58
Microphone	TS-US-115-CA	DAOKI	5	$1.14	$5.69
Breadboard	64	Adafruit	4	$5.00	$20.00
					$190.86

Table 2: Cost of 1 Workstation

Description	Model #	Manufacturer	Quantity	Cost
PIR Sensor	1528-1991	AdaFruit	1	$9.95
LED Lights		KWZM	1	$5.995
Node MCU	ESP8266	AITRIP	1	$2.66
Microphone	TS-US-115-CA	DAOKI	1	$1.14
Breadboard	64	Adafruit	1	$5.00
				$24.75

The cost without the lights is $13.76, which is close to the lower bound of the current cost of a motion detection light ($11).


EVERYONE
MALAVIKA
RYAN
DIVA
RYAN and DIVA
Task
Week 1
Week 2
Week 3
Week 4
Week 5
Week 6
Week 7
Week 8
Week 9
Week 10
Week 11
Week 12
Week 13
Week 14
Project Proposal
Order Parts
Design Pres
Unit Testing
App Front End
Wifi Connect
Localize
PIR Weight
Mic Weight
App/Pi Comm.
Multi Sensors
Combine All Weights
Testing
Final Pres
Demo Set Up