# Wireless-Enabled Smart-Lights Hub Prototype

**Final Project Plan**

Senior Design December 2019 Team 16

Dr. Ravikumar Gelli
Dr. Manimaran Govindarasu
Alexander Beaver
John Ryan Bush
Aaron Ramsey
Logan Zasada

# Table of Contents

## List of Figures



## List of Tables



## List of Definitions

Please include any definitions and/or acronyms the readers would like to know.

**IOT - Internet of things:** the interconnection via the internet of computing devices embedded in everyday objects, enabling them to send and receive data

**TJB – Transmitter Junction Box:** A box we will design that takes the inputs from the relay boxes and creates a wireless packet to send to the controlling computer

# 1 Introductory Material

## 1.1 Acknowledgement

Our faculty adviser is Dr. Manimaran Govindarasu, who has been a mentor for the project by helping us manage our time and progress, provide funding, and giving guidance for the direction of the project and any problems we faced.

Our client is Dr. Ravikumar Gelli, who has helped make large decisions in regards to the direction of the project, as well as give feedback on ways we can improve the project as it is designed.

A few extra students working with Ravikumar Gelli and Manimaran, namely Jiztom Kavalakkatt Francis and Alexander Nicklaus, were good resources for wireless communication.

## 1.2 Problem Statement

The current system in the PowerCyber lab is not a great visual representation of the power stations being simulated. Our senior design project implements a much more visually appealing version of the status light representing a power station and allows the users to see multiple different locations, for the power stations in a geographical area, all at once. The overall purpose is to visually show, using the light units for a specific geographical location, the status of each power plant.

## 1.3 Operating Environment

The light modules will operate while connected to a magnetic board, most likely indoors. Since the operating environment is indoors, the temperatures are expected to be room temperature and well within the light's temperature specifications. It is possible that dust may accumulate on the lights if they are kept in one place or not cleaned on a regular basis. The transmitter junction box is largely stationary, so it will be exposed to dust and potentially higher temperatures. The docking station will be exposed to largely the same conditions as the light modules. The Server will be run on a Linux server, and will be accessed by users using any operating system.

## 1.4 Intended Users and Intended Uses

For the proposed system there will be two groups of users. The primary user group are the members of the PowerCyber lab. The primary users will be the ones that configure, maintain, and move the system. System configuration will consist of mapping automation controller relay outputs to specific light units. System configuration will also consist of routine setup tasks such as placing light units on to the magnetic mounting board. To maintain the system, the primary users will need to make sure the light units are charged before the system is deployed and verify the mapping between the relay outputs and the light units. To move the system, the primary users will install the magnetic mounting board at whatever location the system is to be used.

The secondary users of the system will be industry partners and the general public. The primary interaction of the secondary users will be observing the system and presentations from the PowerCyber lab to learn about cyber-security in a power systems context. Industry partners will also learn how cyber-security principles and intelligent grid management can be used to improve the quality of their operation. Because the secondary users will only be observing the system it's important that the system will have pleasing aesthetics to make sure they're satisfied.

The following case diagram simply illustrates how each user is intended to use the system.

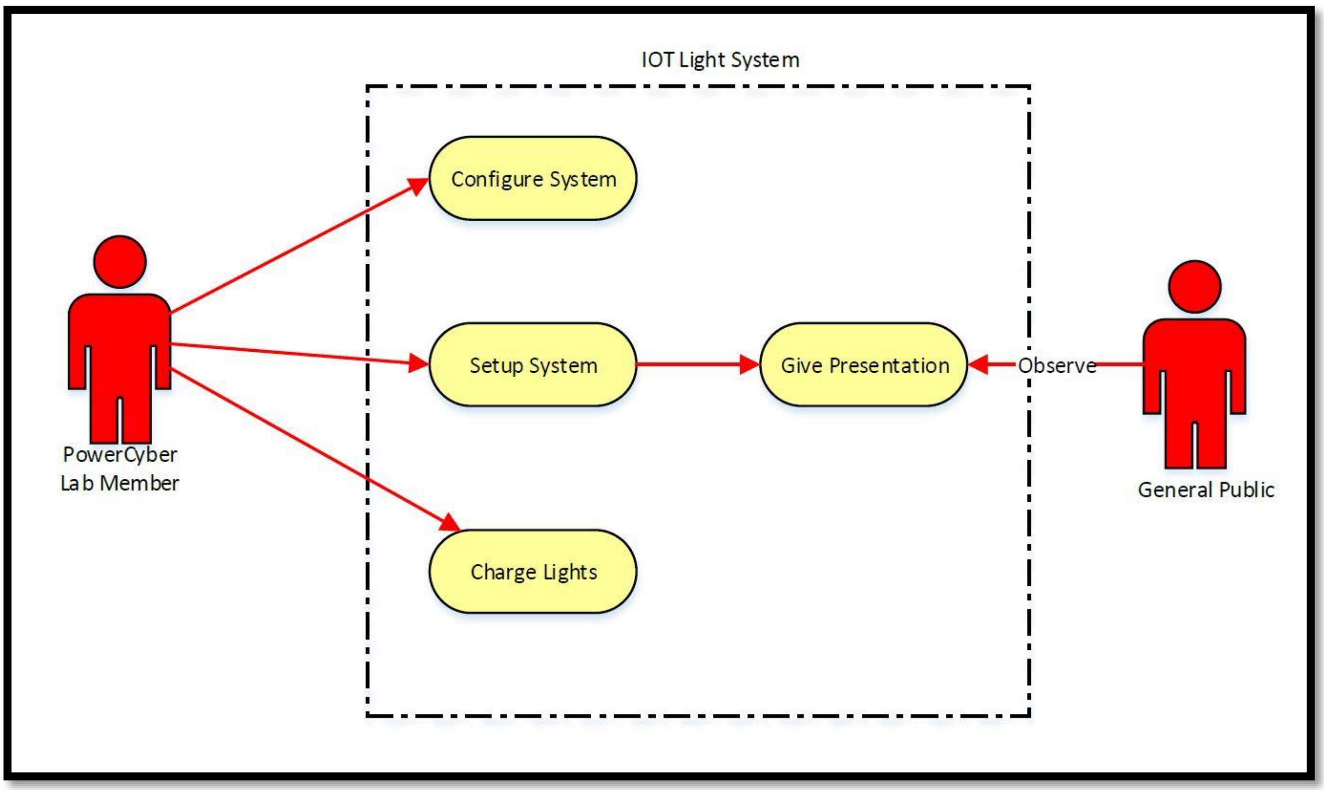


*Figure 1: Case Diagram of Light System*

### 1.5 Assumptions and Limitations

A few things can be assumed about this project both during design and the implementation. The project is funded and supplied explicitly by Dr. Manimaran and ETG. The only people handling the system are PowerCyber Lab members. When the system is in use, internet connectivity will always be present along with a power outlet for the docking station, when needed. Upon completion of the project, maintenance and care of system will transfer to Dr. Manimaran Govindarasu.

Along with assumptions, there are also limitations to the project. One major limitation for the project is time, since we only have two semesters to complete it. The part of the project connected to the relays is limited in that it must be a direct wire connection. Individuals outside of those authorized, will be unable to initiate any wireless communication. Another major limitation for the project is only having one software proficient team member.

### 1.6 Expected End Product and Other Deliverables

At the end of the Fall 2019 semester, we will deliver the following items to the client:

- Transmitter Junction Box (TJB) that takes port status from each relay and updates the server periodically
- Server containing port status, port-to-light mapping info, and other info
- Hub that can connect via wi-fi or ethernet to our server, and send the data in the server to the light modules
- Approximately 100 light modules controlled by the aforementioned hub
- Multiple charging stations capable of charging stacks of 8 lights at a time

The purpose of the TJB is to poll the status of each port of multiple relay boxes, packetize the data, and update our server. This will be done with a microcontroller and some digital circuitry, as well as an ethernet communication module. The TJB will be permanently stored in the PowerCyber lab, and as such will always be connected to 120V AC power and plugged into the PowerCyber lab's network.

The server works as a central database, and keeps a live table based on what's updated by the TJB. The database holds a port-to-light mapping that corresponds a specific hardware port in a specific relay box to an individual light module. The users can access the server to view statuses of ports and lights, and also change the port-to-light mapping. This allows users to, when they're doing a demonstration using our system, grab the number of fully charged lights they need, decide which light corresponds to which port in the relay, and then any update to the ports they selected will be reflected on the light they mapped. Also, if during demonstration, one of the lights should run out of battery, they can easily swap out the light by changing the mapping from the port to a different, more charged light.

As users travel to different locations with our system, sometimes outside of the reach of the PowerCyber network, they will need a way to connect to it so they can grab the light statuses from the server. The hub works as this bridge, and will consist of a microprocessor, such as a Raspberry Pi, connected via ethernet to our server, and connected via some other connection method to each light module the users bring with them. The connection method from the hub to the light modules is still undefined at the time of writing this, and is discussed later on in this report. The hub will be connected to a 120V AC source as well as either Wi-fi or ethernet for the duration of the user's demonstration.

Each light module will contain a wireless communication module of some sort, as well as some circuitry to convert the wireless data into an actual signal that tells it whether its light should be on. Therefore, the light module will contain an LED, with some sort of diffusion to make it obvious when the LED is illuminated. These lights modules will be powered by rechargeable batteries, and the outside of the module's housing will have a magnet so users can stick the light module on a large magnetic board for easy demonstration.

Finally, the charging stations will be permanently housed in the PowerCyber lab, and be able to set up to 8 light modules inside and charge them, with some sort of indication of charge status.

# 2 Proposed Approach and Statement of Work

### 2.1 Objective of the Task

We are to design three (possibly four) hardware products and two software products. The first hardware product, a variable number of IOT lights, should communicate and receive communications from a transmitter junction box (TJB). We must also design the TJB. The TJB is a device that will read the status of the relays and upload that information to the light status database. The last hardware product we must design is the charging station for light units. If we cannot find one that suits our needs, we will also have to design a light transmitting hub.

For software products we will need to design a server/database solution and a user interface. The server/database solution will be the heart of the IOT system and hold all the information that the IOT connected devices need to function. The user interface will give the users a way to reconfigure the server/database to their liking.

### 2.2 Functional Requirements

The project has two categories of requirements. Base requirements and optional requirements. Base requirements are what define the project as a success, and optional requirements are things we can add in if we think they will benefit the project in any way, but aren't critical.

- Base Requirements
  - Transmitter Junction Box (TJB)
    - Take input from multiple (approximately 10) relay boxes. Each relay box has 8 output contacts
    - Must be scalable to allow flexibility of connecting multiple relay boxes to each TJB with relative ease
    - Sends relay statuses to light status database via ethernet connection
  - Light Status Transmitter
    - Receive relevant light status info from light status database
    - Transmits messages to lights to turn them on and off
    - Should be able to communicate with 100+ light modules
  - Light Status Database
    - Should be able to receive messages from TJB
    - Should be able to receive request from Light Status Transmitter
    - Should be able to store relay status indefinitely
    - Should be interfaceable via User Interface for Programming lights
  - Individual Light Modules
    - Approximately 100 standalone light modules
    - Have a magnetic backing to stick to a magnetic board
    - Have an ID associated with each light so a single port of the relay box can somehow be associated with a single light module
    - Have wireless receiver in light grab data packet from Light Status transmitter

        - Battery is rechargeable and lasts for 8-12 hours
        - Physical size is about the same as existing lights (approximately 2” in diameter and 4” long)
        - Compatible with docking station
    - Light Docking Station
        - Each docking station can charge 8 lights at a time
        - Have multiple docking stations
    - User Interface for Programming Lights
        - Have some way to program lights if we wanted to lay all lights out in a certain pattern (such as spell out a name) and have the lights flash/change color
        - Lights can be accessed from anywhere with a remote connection to the PowerCyber lab
        - Needs to (re)configure mapping between relay and light statuses
    - Other
        - Magnetic mounting board must be able to move easily in and out of PowerCyber lab
        - Maps should be able to easily projected onto magnetic mounting board
        - In the event of a failure, the system should fail safely
- Optional Requirements
    - TJB
        - Have status lights to show what state each of its input ports are at for quick debugging
    - Light Modules
        - LED’s to show whether the light is ready or not
        - Battery level indicator
        - Dimmable light
        - Color changeable
    - Docking Station
        - Status LED’s for battery level/charging/done
    - User Interface
        - Device built into TJB or standalone that allows for programming lights with GUI

### 2.3 Constraints Considerations

The major constraints with the project are as follows:

- Time
- Space
- Technical Knowledge

Time is a major constraint for the project because the successful completion of the project will require much design time as well as a large amount of manufacturing time. Because the successful completion of the project requires the making of 100+ light units and the appropriate system peripherals (such as wiring harnesses) it's possible that the manufacturing could take more time than the design and testing.

Space must be taken into consideration because the system consists of 100+ units, a charging stations, and a magnetic mounting board. However, all of the system components must easily be stored in the PowerCyber lab.

Technical knowledge represents a serious challenge because our team consists mostly of electrical engineers with limited software and IOT experience. Because of this, it will take a non-trivial amount of time to develop a functional system architecture which will take away time from design, testing, and manufacturing.

The Major non-technical requirements for the system are as follows:

- The light board with mounted light units should be aesthetically pleasing
- The system should be easy to configure
- The system should be easy to transport
- The system should function safely and reliably

Since the light system will be used for outreach purposes, it's important that the observers aren't put off by the aesthetics of the system. The system must be easy to configure so that the PowerCyber lab doesn't have to waste excessive amounts of time configuring the system before each use. Because the system will be taken to different locations, it's important to design it in such a way that will make it easy to physically transport from the 3rd floor of Coover hall to any off-site location. Lastly, the project group will most likely not touch the system once the project has been completed, so it's important that the system is designed with reliability and safety in mind.

Because the project team is so small, there's no need to adhere to a standard protocol for designing or documenting. However, this means that the project members must also make a conscious effort to maintain clear communication of workflows, expectations, and results. Otherwise the project could fall into unethical states and disappoint the client.

### 2.4 Previous Work And Literature

The client is looking for a wireless-controlled device with up to 8 hours of battery life, that can be controlled automatically (to automatically update the status of the lights based on relay boxes). To satisfy the client, we had two options; find something on the market that was close, or build everything ourselves. After lots of research, we came upon the following devices.

#### Amazon "VegaHome Smart Light Switch"

The VegaHome is wireless, Wi-Fi controlled, and it controls a light bulb. One difference that can be seen initially, our modules input, to tell the device to be on or off, is coming from a system already in use. The VegaHome's input is the user pressing the light switch button or through an Amazon Alexa or the google home using voice directions to control the light output. Another difference is the variability of the VegaHome's light brightness. Our light will be at a specific unchangeable brightness.

The shortcomings of this already existing system for our specified project would come down to the input of the VegaHome versus the input of ours. Since our input comes in data packets from the Transmitter junction box, to adjust this already existing system and make it work with our project, may be more work and harder to accomplish than finding the smaller sub-components and designing them to work together.

**Alibaba "Wi-fi LED Battery Light Source"**

This light module could have been the ultimate solution. It came with a wireless-controlled, battery powered LED light that is programmable and hackable. The university had trouble ordering parts from Alibaba, however, and we couldn't get it. As we got farther into the project, we realized that this light being controlled by Wi-fi may have been a challenge, however, so it's a good thing we couldn't get these, as we would have discovered they wouldn't work.

**Amazon "Zigbee A19 LED Bulb" (AC)**

As another option, we decided to try AC lights. Due to the "smart-home" revolution happening right now, there are TONS of AC lights that can be programmed to do whatever you want, and plenty of software environments to program whatever you want. Important to us was the ability to read data from a server and update the lights based on the server, which an open-source software called OpenHAB would allow us to do. After working with OpenHAB and trying other devices, like a Samsung SmartThings hub, we decided it would be too complex to go this route, especially considering we'd have to power an AC light with a DC battery, and make it last 8 hours.

After looking through all of these options, we started to realize that this type of light we were looking for was pretty niche. Most devices that have wireless-communication abilities, unless they're BLE, can't sustain 8 hours of battery life, simply because wireless communication takes so much power. As a result, we can't find exactly what we want on the market, because nobody can make it effectively.

## 2.5 Proposed Design

### 2.5.1 Relay – to – Server Transmitter Junction Box

Because our system will be controlling IOT light units based on the status of relay boxes in the PowerCyber lab; we will need device that can interface with relay boxes and is IOT connected. In the proposed design, we suggest the implementation of a transmitter junction box (TJB) that would handle this function for the system. The figure below shows a block diagram of how the transmitter junction box is supposed to integrate with the system.

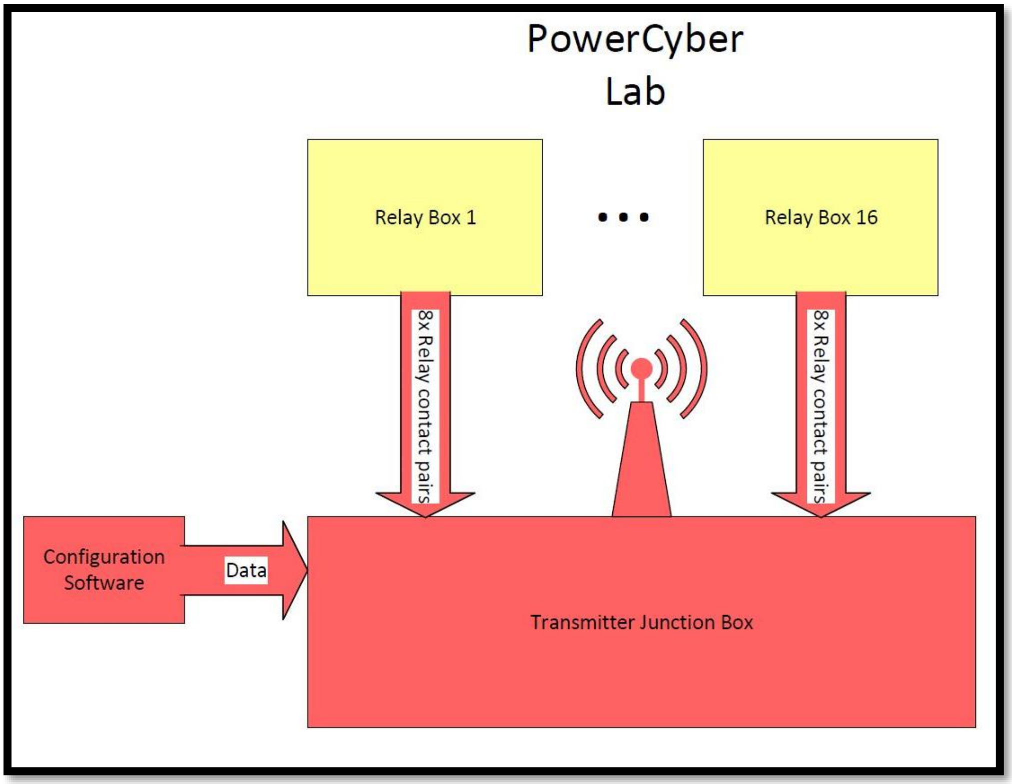


***Figure 2: TJB Block Diagram***

As for the detailed design of the transmitter junction box, the following schematic has been proposed

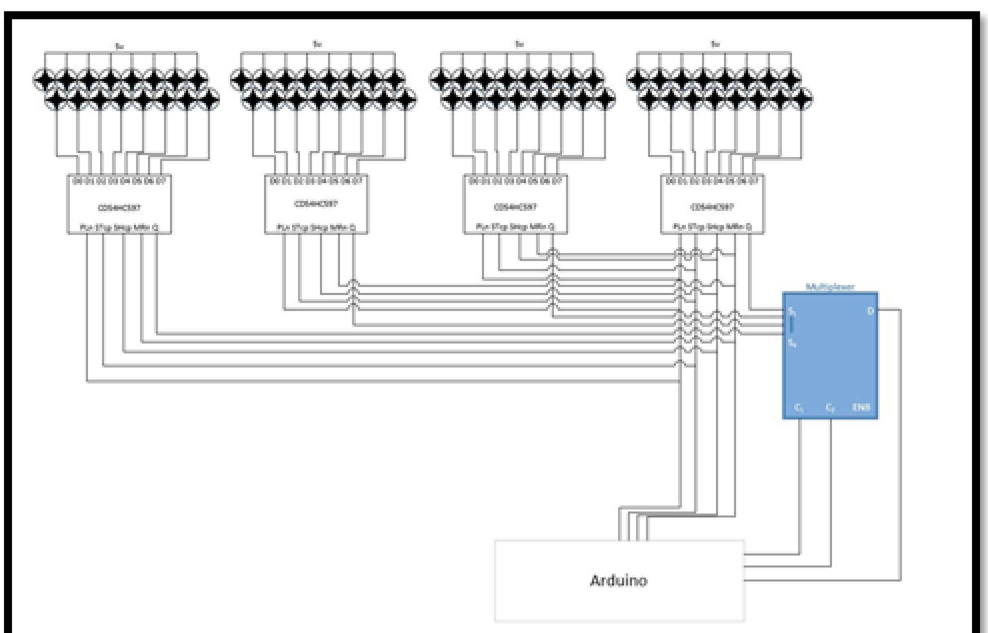


***Figure 3: TJB Circuit Diagram***

The proposed design uses parallel to serial shift registers and a multiplexer to serialize the data from the relay boxes and allow the Arduino to read all of the relay statuses by sending the proper control systems and changing the multiplexer select bits. The Arduino will then use the connected ethernet shield to transmit this data to the light status database. This design was chosen because It's modular and easily scalable. This design is also a good choice because it relies entirely on digital signals which are more immune to noise to analog alternatives.

### 2.5.2 Server

In order for the system to function, a server must be set up to receive requests from the relay microcontroller. In the lab, there is an Ubuntu VM running on a Windows desktop that can be remoted into.

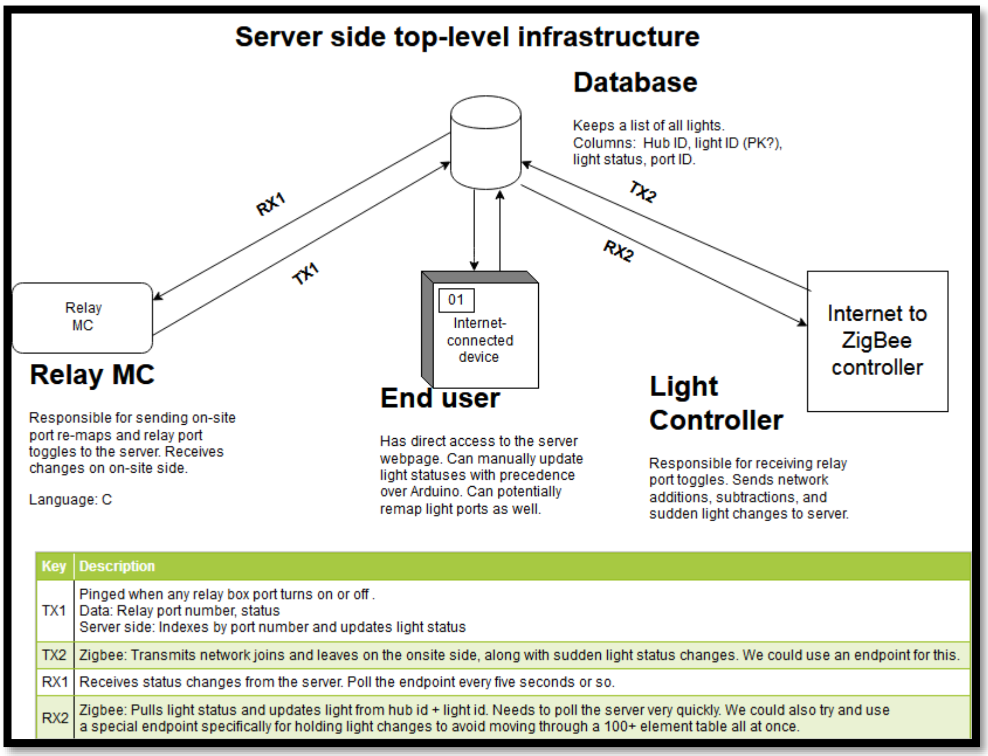


***Figure 4: Server communications with either end point***

The figure above details the various communications the server would need to be a part of. The most important of which is light status communication from the in-lab microcontroller to the onsite controller. To accomplish this, the server would need to have a site that would be public-facing as it might not be possible to mask the communications behind the Iowa State VPN. As such, the server is split up into three parts: The database, the front-end, and the API endpoints.

| **Column Name** | **Column Type** | **Visible to…** |
|---|---|---|
| Light_ID | int | On-site hub |

| | | |
|---|---|---|
| Hub_ID | int | On-site hub |
| Light_Status | int | In-lab microcontroller<br>On-site hub |
| Relay_ID | varchar | In-lab microcontroller |

***Table 1: Proposed database table***

| **Endpoint name** | **URL scheme** | **Response** |
|---|---|---|
| Toggle light | toggle?ID=x&status=x | 200 if successful<br>520 if not in DB<br>400 if wrong syntax |
| Add light | add?ID=x | 200 if successful<br>520 if in DB<br>400 if wrong syntax |
| Remove light | rem?ID=x | 200 if successful<br>520 if not in DB |

***Table 2: Proposed API endpoints***

According to the RFC 2616 specification, custom HTML status codes can be defined, but must be treated as if they are of the type defined as the most significant digit. The 520 errors would be processed as server errors as they are within the 500 class. The endpoints will consume the key value pairs and query the database based on what endpoint the microcontrollers hit.

The main front end view contains the database table itself. A user should be able to turn a light off from the database and the light itself turns off. Remapping ports should ideally be done through the front end as well.

### 2.5.3 Server – to – Light hub & Light Modules

To transmit the data from the server to individual lights, we will have a local hub connected to our server via internet, and it will communicate wirelessly to local light modules. The local wireless communication method is still undecided, we're deciding between using Zigbee and RF. Below are the schematics for each design. You'll notice that the designs are very similar, with the only difference being the communication modules and the communication method.

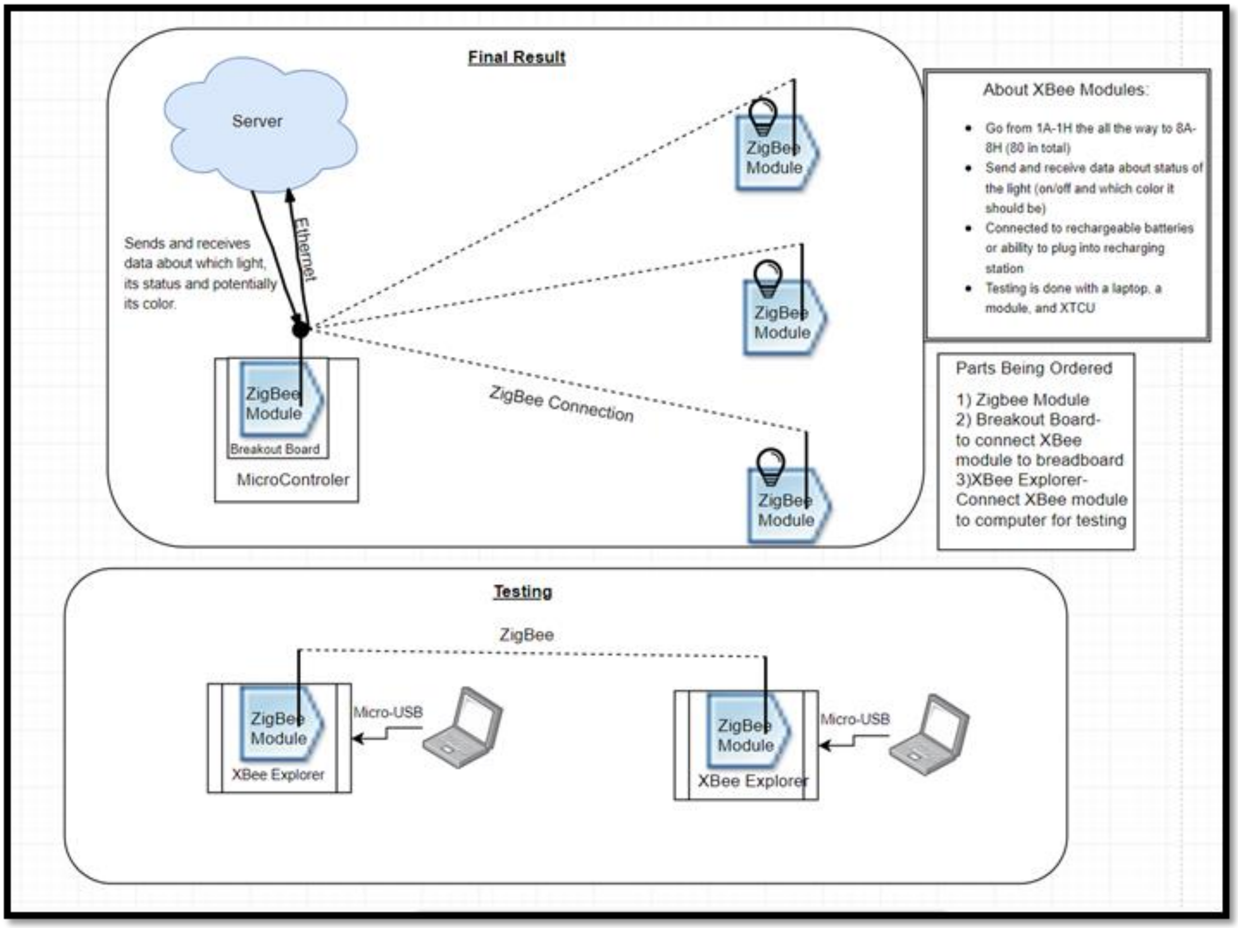


***Figure 5. Server-to-light module using Zigbee Communication***

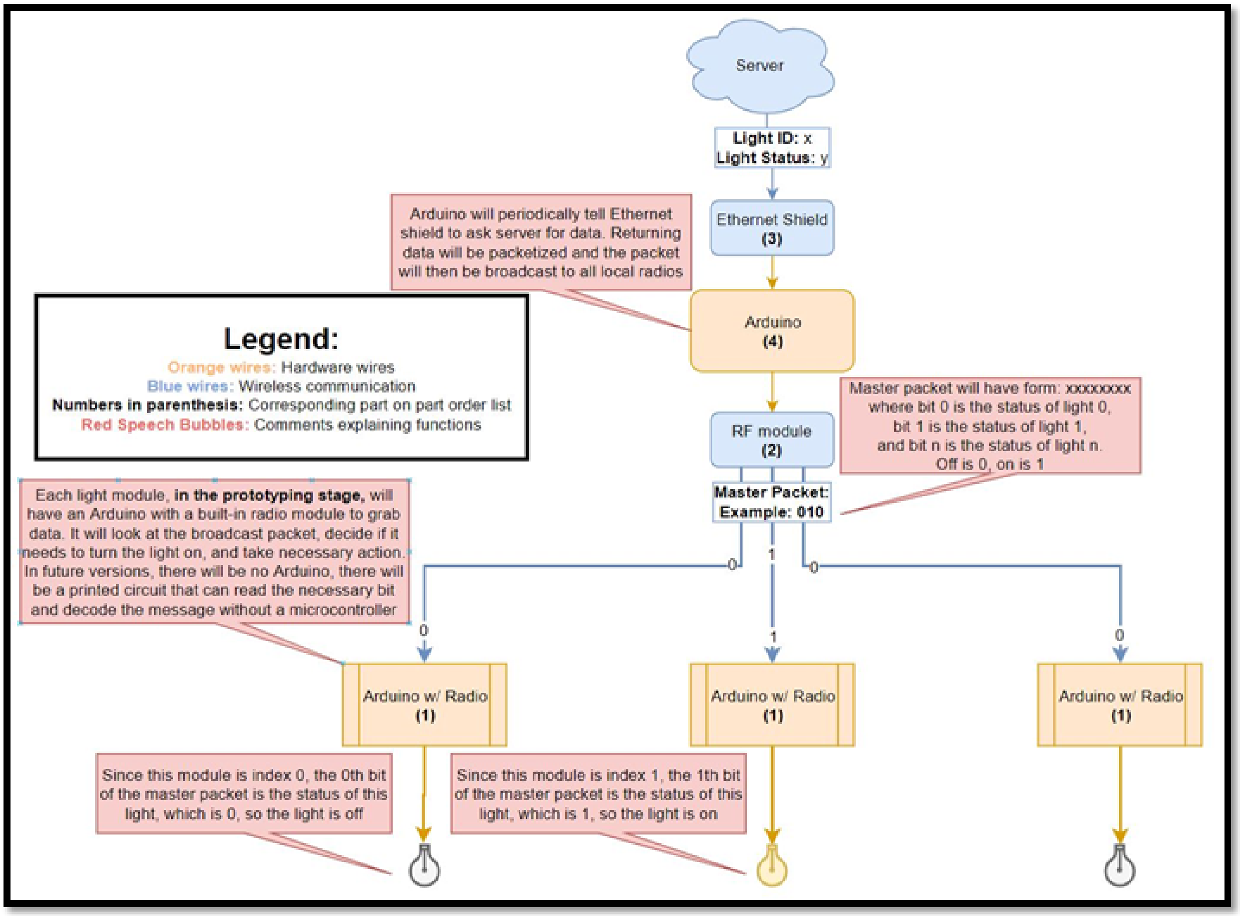


***Figure 6. Server-to-light module using RF Communication***

In both above figures, there is a microcontroller acting as a hub to transfer data from the server to each light module. This may be replaced with a Raspberry Pi due to the ease of this packetization using python versus doing everything on a lower level with a microcontroller. Also, currently the individual light modules have Arduinos to decode the message received and decide whether the light is on. In future designs, we hope to remove the microcontroller entirely and have a digital circuit decode the data without using a microcontroller. This will reduce costs significantly, since the microcontroller would be the most expensive part of the light module and we're planning to make about 100 light modules.

### 2.5.4 Light Charging Stations

The Light Charging stations will consist of a charging circuit board that takes 120V AC power and converts it into a voltage that is capable of safely charging the batteries. This circuit board will be housed in a box whose only interface is a plugin for 120V AC and the plugins for each light module to allow it to charge. Each charging station will be capable of charging 8 lights at a time, and have status indicators of the charge of each battery charging port.

### 2.6 Technology Considerations

IOT lights are an already available technology we could utilize. Instead of designing our own light product, we could repurpose a product that already exists. This cuts down on project complexity, but also means that there is a purchasing cost. Time would also need to be spent modifying each light. The lights also may have functions that the client does not need and may interfere with normal operations.

However, finding IOT lights that are compatible with the projects system requirments has proven near impossible. Because of this we will also need to develop our own light modules in parallel with searching for IOT lights. Developing these light modules will require evaluating different wireless communication technologies as well as battery technologies.

This project also requires the extensive research of IOT technology in general. From Blue-tooth LE to WiFi there are a lot of different avenues to explore for interfacing between different parts of the system. There's also the question of what database technology we should use and how that can be integrated with the IOT technology. If any of these considerations are neglected, it's highly probable that the project will result in failure.

### 2.7 Safety Considerations

Our major safety concern is the safety of the light units because these units will likely use lithium batteries. If these units are mistreated, it's possible for them to catch fire in such a way that could catastrophically damage their surroundings. This danger is compounded by the fact that there will be a large quantity of these units in one place and one unit catching fire could theoretically catch the other units on fire. Not only will safety of the end product need to be considered, but the project team must also be conscious of the dangers of lithium ion batteries while they are developing the project.

The other major safety concern for this project will be working with 120VAC. It's likely that the transmitter junction box and charging station will be powered by 120VAC so it's important that the team members working on these components are educated about the best practices for working with 120VAC. The final system will need to be designed in such a way that the 120VAC circuits will fail safely (by using fuses and other protection devices). The corresponding enclosure should also be designed with safety in mind to protect the system users from electrical shock hazards.

### 2.8 Task Approach

#### 2.8.1 Relay – to – Server Transmitter Junction Box

The Relay-to-Server Transmitter Junction Box (TJB) is a data handling device that will digitize the status of the relays in the power-cyber lab and upload this information to the Light Status Database on the server via an Ethernet connection.

Because of the complexity of the ethernet protocol, this device will require the implementation of a microcontroller that's connected to ethernet-enabled peripherals. The other major consideration that adds to the device's complexity is the number of relays it will have to accept input from. Because the device may have to accept input from upwards of 100 relays, it's important that the relay interface is designed to be modular and scalable. The sheer number of relays the device must connect to necessitates a fairly large number of I/O on the microcontroller regardless of design hierarchy.

It's because of these designed requirements we've chosen to develop the device based on the Arduino platform. The Arduino platform has an established ethernet communication peripheral circuit with plenty of reference libraries. If more I/O is required, it's extremely easy to modify code for the smaller microcontroller to run on a larger one. These factors combined will expedite the development of the device and help create a better end product.

### 2.8.2 Server

As the server side mostly deals with software transactions, prototyping largely deals with the software running in a controlled offline environment. Before any physical prototypes are constructed, a web framework and language driving it will be chosen and finalized. An SQL schema can be implemented either locally or on a remote VM as easy changes could be made in software to reflect database location adjustments.

In a controlled development environment, an implementation of both the front and the back ends will be done. A major objective for this area is that any changes in the front-end webpage would reflect changes in the back-end database and vice versa. From there, a web server will be installed on the VM the team is provided. The application will be deployed on the web server, and the endpoints listed in the above table will be tested with the microcontroller. If one light is able to be turned on, and the respective change is seen in the database, then the integration test will be successful. Further scaling tests will be done afterwards to test for any growing issues on the server.

### 2.8.3 Server – to – Light hub & Light Modules

As discussed earlier, the server-to-light hub will have to take in data from the server via ethernet, turn that into a message to communicate to all lights in the local area, and communicate it. We also haven't determined whether to use RF or Zigbee communication in the local area. To determine which is the better way to go, we're developing a prototype using each communication method, and at the end of the Spring semester, we'll discuss with our team and advisers the strengths and weaknesses of each method and decide which way to go.

Once we decide which communication is better, we will continue improving on that prototype, reducing complexities and adding functionalities until we have a final project. At this point, we hope to design printed circuit boards to have approximately 3 hubs and 100 light modules. We will get these professionally printed, construct the rest of the light modules, and deliver the final product after heavy testing.

### 2.8.4 Light Charging Stations

Once we determine the best battery to use for the lights, we will know what kind of circuitry we need to charge that type of battery. We can then begin designing the actual battery charging circuit and prototype it when it is done. We can then test it by discharging a battery and adding it to the charging circuit to test how quickly it charges and when the charge cuts off. If the test goes well, we can test out how well the circuit stacks and how many batteries we can charge at once. After testing, we will construct the station casing and run a field test using the lights we have designed.

## 2.9 Possible Risks And Risk Management

This project will require the use of lithium ion batteries. These batteries are known to catch fire in a way that is extremely catastrophic. To mitigate the risk batteries, pose, the project members should educate themselves on the best practices for handling these batteries and verify their designs will fail safely in case of a battery failure.

For communication, this project will require the use of either Zigbee, a low-power networking protocol, or another communication method such as RF. We also need a wireless transmit-receive circuit on every device. The team working on this project does not have experience with IOT circuits or protocols so their implementation will require a significant amount of time. To mitigate this risk, the team will split off in their focus areas and develop a working prototype as quickly as possible. They will experiment with different technologies only as necessary for the development of this prototype to the final product.

To complete all the optional requirements for the project, the team will need extensive knowledge of internet protocols. Presently this knowledge is absent from the team, which poses a significant technical challenge to the team. It's possible that by the time the team is comfortable with internet protocols there will not be enough time to implement internet functionality into the project. To mitigate this risk posed by this technical challenge, the team will start with a working prototype that's not internet connected while researching internet protocols.

## 2.10 Project Proposed Milestones and Evaluation Criteria

To ensure that our project meets the requirements, one of our biggest goals is to always have a working prototype so we can get immediate feedback on how the system works. As a result, the first milestone was to have a first version of the working prototype by mid-March 2019. After struggling with Zigbee communication setting us back, we now hope to get a prototype of the server-to-light module with each communication method by the end of the Spring semester.

By April 2019, we had wanted the final design for the TJB drawn out, knowing what parts will connect to what, and how everything is enclosed, which we met. The hope now is to have a working prototype of the TJB module that communicates to the server by the end of the semester.

In May 2019, before the semester ends, we'd like to have a plan for the charging stations and a rough idea of what the final software will look like.

For the fall semester, by September, we'd like to have a semi-final prototype built and available for demonstration. Then in September/October we can conduct final tests. Final tests we'd like to conduct are:

- Battery life within 8-12 hours
- Charges within reasonable time
- System can handle reasonable number of lights updated at the same time
- System functions in real-time
- System has reasonable range
- Software is easy to use

### 2.11 Project Tracking Procedures

Communication from each member of the group was the first step to properly track the progress of the project. In order to communicate properly, each member now has access to GroupMe (a group messaging app), as well as a CyBox, where the meeting minutes, weekly reports, part orders, diagrams, and any other type of documentation are stored. We also have weekly meetings with our advisor/client, Dr. Manimaran Govindarasu, as well as weekly meetings with just the four members of the group.

In these weekly meetings, we discuss the progress of the project in the previous week, as well as any current difficulties or problems. We then lay out our goals we want to achieve in the following week, so everyone knows what each person is working on and can assist when needed.

### 2.12 Expected Results and Validation

At the end of the project, we should be able to turn on the lights and then individually turn them on and off wirelessly through the relay boxes. To do this, we will use the PowerCyber lab's software that controls each relay box, turn on and off certain ports in the box, and examine the results on the lights. This will verify that the lights are able to turn on, the lights are connected to the TJB, and the lights are wirelessly communicating to the TJB. We then plan to leave them on for an extended period of time to test their battery life. When the battery life is expended, we will test the docking station by charging them up. Finally, to test scalability, we will add more lights to the system.

### 2.13 Test Plan

The most important tests are named in section 2.10, and this section will go a little more in depth as to what each test is looking to accomplish and how we plan to run them.

- Battery life within 8-12 hours
  - Set up approximately 3 lights in 1 of 3 states:
    - On 100% of the time
    - On 50% of the time (on for certain number of minutes, off for same amount of time)

        - Light is ready to be turned on and connected to laptop, but the actual light is not on
    - Leave these lights in these states and, every 15 minutes or so, check if any of the lights have run out of battery, and mark which ones are out
- Charges within reasonable time
    - Run multiple lights until the batteries are completely empty, time how long each takes to recharge fully, and discuss with client to see if we have a reasonable charge time
- System can handle reasonable number of lights updated at the same time
    - Connect as many lights as the PowerCyber lab currently has of ports and see if the system still runs in real-time
- System functions in real-time
    - The time between tripping a contact and the physical light turning on should be within a second
- System has reasonable range
    - Set up the system in a large lecture hall, with the computer at one end of the lecture hall, and try turning on lights at the other end
    - This is farther than we need, but it may help alleviate some noise concerns
- Software is easy to use
    - Give the software to the client and get their opinions on how easy it is to program lights

# 3 Project Timeline, Estimated Resources, and Challenges

## 3.1 Project Timeline

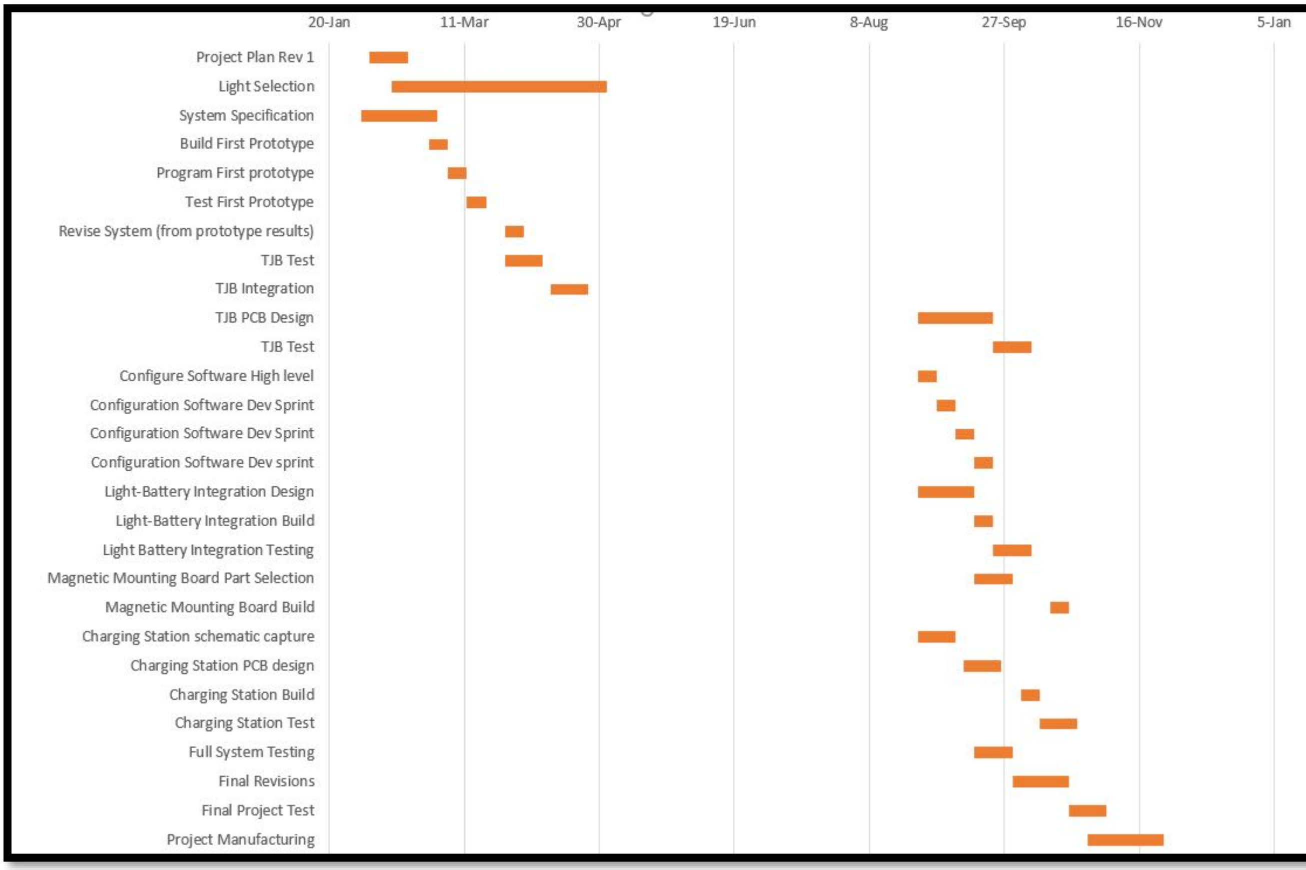


***Figure 7: Gantt chart demonstrating proposed work schedules for the entire year.***

The preceding Gantt chart shows the timeline we plan to follow for this project. It aims to identify all the designing, building and testing that must be done to complete the project. The timeline demonstrates how the system planning will happen all throughout the first semester and the rest of the project will be primarily development and testing. The chart also shows how the first semester will involve creating a lot of documentation, while the second semester will involve much more building and testing. Another thing that can be observed here is that there are multiple work flows at most times.

This is because the timeline was designed with teamwork in mind. The tasks are separated so that one or two people could reasonably complete them within the given time frame with a commitment of 5-7 hours a week. This seemed to be the most ethical way to break down the tasks as it would be unfair if there was one task or task series that required a higher hour commitment per week than the others. However, due to difficulties selecting an IOT light solution, much of the charging station and battery-light integration tasks have been pushed to the second semester. Meaning that the second semester will now require more work from everybody as a team.

There's a reasonable amount of time between a lot of the designing and the building because it will take several weeks for parts to come. There's a lot of EE 492 assignments that are missing from this chart, but they will be worked on in parallel with the tasks seen.

The last thing to note about the timeline is that there's a significant amount of time that's dedicated to system level and component level revision. It's extremely rare that things work out perfectly the first time so planning for this in the project and implementing it into our timeline will be a major asset for communicating project expectations and goals.

### 3.2 Feasibility Assessment

The project involves the design of multiple modules, each with its own challenges. The most challenging aspect of the TJB will be connecting it to the server. Currently, none of our team members have programmed an IOT connected device before so creating the server connection will take significant research. However, we will use well established platforms will high level API's like Arduino. This makes the development of the TJB feasible even though the team lacks experience.

The light modules would be ideally purchased from a manufacturer with the functionality in them, but the tasks we're trying to accomplish are very niche and it will be difficult to find the functionality we're looking for without spending too much and getting more than we need. It's possible we'll have to create the lights ourselves; which will be a costly, and time-consuming. Endeavor. To create the light modules on our own, we will need to either mod an existing light with a communication module and a communication module, or we will have to create an IOT light (including the physical housing) from scratch. Manufacturing 100 light units will take up a large portion of our time if we do end up having to make the lights ourselves. If we must go this route, it will negatively impact the feasibility of project. It's feasible that we could manufacture 100 light units, however it's more probable that we will manufacture somewhere around 30 lights and the PowerCyber team will have to find someone to make the rest.

The charging station will have to be aesthetically pleasing as well as physically sturdy, and since all of us are from an electrical and computer engineering background, we may struggle with designing the housing. The technology required to make a safe charging station is readily available and easy to implement. Because of this, developing the charging station in house is extremely feasible endeavor that can be completed in little time.

The user interface will have to be very intuitive and easy to use. None of us don't have a strong user interface background, so this task will take lots of research and time to develop to make sure that people who see our project being used in presentations are impressed with it. However, the development of this interface remains feasible as we have information resources we can utilize and there are no extensive lead times to worry about.

### 3.3 Personnel Effort Requirements

Below is a table which shows the main contributions each member should have, on a weekly basis, to the group. Each week we have at least two meetings, the first is on Mondays with Dr. Gelli Ravikumar who is our client and works closely with our advisor, the second meeting is with only group members. Attendance of these meetings is critical to keeping the project group well informed and the meeting quality will suffer if even one member doesn't attend. Many hours of work are required to build a functional system that meets all requirements; therefore, it is essential each member puts in at least 5 hours a week. Because some parts of the projects have gotten behind in the first semester, it will be crucial to step up our involvement in the next semester to the point where each member is putting in about 7 hours a week. At each meeting, we set some sort of goal by the end of the week; which makes it easier to track our progress and monitor the project. Documentation is critical to our success; that's why each member is responsible for reporting their achievements and roadblocks in the weekly report. The last column refers to how each of the members in the group was assigned a role at the beginning, and on a weekly bases it is important to be included to hold each member accountable.

| Week X | Attended Meetings | Put at least 3 hours of work this week on project | Accomplished goals set and discussed by the group for this week | Finished portion of weekly report | Has been doing the duties of assigned role |
|---|---|---|---|---|---|
| Aaron | | | | | |
| Alex | | | | | |
| Logan | | | | | |
| Ryan | | | | | |

***Table 3: Example Weekly Contributions Checklist***

### 3.4 Other Resource Requirements

The financial resources will largely depend on if we are going to manufacture our lights, or buy them off the market and modify them. It will also depend on how we are going to communicate with the lights. For prototyping, at minimum we will require one microcontroller for transmitting signals, and three or four lights with wireless receivers for receiving the signals. The transmitter junction box will require a microcontroller with an ethernet port and an outer casing that supports the contacts from the relay boxes. We will require 100 lights, each with a simple microcontroller containing an onboard Zigbee or RF transceiver, a battery, an external LED status light, and a magnetic backing. An Internet-connected hub is also required for communicating with the lights. This can be bought off the internet.

## 3.5 Financial Requirements

The financial requirements will depend largely on whether we decide to use ZigBee to send the data to the lights or if we use an RF Radio. The below tables cover these two cases and are subject to change as the project progresses.

| ***Item*** | ***Qty*** | ***Price per item*** | ***Total Price*** |
|---|---|---|---|
| *XBee s2c Radio Module** | *101* | *$17.50* | ***$1787.50*** |
| *XBee Explorer USB** | *2* | *$25.95* | ***$51.90*** |
| *Breakout Board** | *100* | *$2.95* | ***$295.00*** |
| *Arduino Uno (Microcontroller)* | *2* | *$20.90* | *$41.80* |
| *Ethernet Shield* | *2* | *$15.99* | *$31.98* |
| *Shift Register (Parallel to serial)* | *3* | *$1.57* | *$4.71* |
| *8:1 Demux* | *4* | *$0.57* | *$2.28* |
| *Raspberry Pi* | *1* | *$40.00* | *$40.00* |
| *LED* | *100* | *~$1.00 (estimated)* | *$100.00* |
| *RF Radio Hub*** | *1* | *$9.95* | *$9.95* |
| *Arduino with Radio*** | *100* | *$24.95* | ***$2495.00*** |
| *Miscellaneous Costs* | *N/A* | *$200.00* | ***$200.00*** |

Table 2: Cost list of components

The "*" symbol in the above table refers to the ZigBee module communication path and "**" refers to the RF module communication. Early on in the project, there was much difficulty figuring out how to interface the different ZigBee components. Due to this, we decided to design both the ZigBee path and Radio path in parallel. The total costs for most paths are compared below.

| Pathway | Total Cost ($220.77 + (ZigBee or Radio)) |
|---|---|
| ZigBee Path Implemented | $2348.17 |
| Radio Path Implemented | $2915.77 |

Table 3: Total Cost

(Note these values are subject to be changed or adjusted as project continues.)

# 4 Closure Materials

## 4.1 Conclusion

With the current setup in the PowerCyber lab, the existing lights are inadequate in number and in representation. Relay boxes output 8 signals and the current setup directly hard-wired to the light units only. It is difficult to manage and operate this hard-wired setup as it scales up. Also, it limits aesthetic and informative visualization of these light units with reference to a power grid map as they are not flexible to move as well as improper space to project a power grid map underneath the light units.

Our project, in an effort to fix the two aforementioned problems, will be a new system where there is a large magnetic board with a geographical map projected onto it. Users will place a wireless-enabled light at a point on the map that they want to represent a certain power station. They will take the ID of that light and tell our system which power station (meaning which port in a relay box) that light corresponds to. Our system will take in the current status of that power station from the relay boxes in the PowerCyber lab, send it wirelessly to a local hub at the presentation site, and update the status of that power station with the corresponding light. This will allow presentations to simulate a hack in one power station, and show the effects of that power station being compromised on other power stations.

### 4.2 References


Amazon, "VegaHome Smart Light Switch, Remote Control Wi-Fi Light Switch with Timer, 15A Wireless In-wall Light Switch, Works with Alexa/Google Assistant/IFTTT, No Hub Required, ETL & FCC Listed, White, 2 Pack." . https://www.amazon.com/VegaHome-Control-Wireless-Assistant-Required/dp/B07HF8VY7V

WIFI Remote Control LED Battery Light Source RGB Multi Colors Changing Rechargeable Light For Outdoor https://www.alibaba.com/product-detail/WIFI-Remote-Control-LED-Battery-Light_60264914786.html?spm=a2700.7724857.normalList.2.6747e0feUNzzUH

remote/wifi/DMX control battery operated led light for costume decoration https://www.alibaba.com/product-detail/remote-wifi-DMX-control-battery-operated_60287138403.html?spm=a2700.7724857.normalList.12.6747e0feUNzzUH

SYLVANIA LIGHTIFY ZigBee Adjustable White A19 LED Bulb, Works with SmartThings, Wink, and Amazon Echo Plus, Hub needed for Amazon Alexa and the Google Assistant, Basic Box https://www.amazon.com/SYLVANIA-LIGHTIFY-Adjustable-SmartThings-Assistant/dp/B01HGKSXDC/ref=sr_1_13?keywords=sylvania+smart%2B+zigbee&qid=1550593762&s=gateway&sr=8-13%20

ZigBee / IEEE 802.15.4 - https://fenix.tecnico.ulisboa.pt/downloadFile/3779579626799/Zigbee-sintese.pdf

Overview - Bluetooth, ZigBee, and WiFi - https://www.slideshare.net/MostafaAli39/understanding-differences-between-bluetooth-zigbee-and-wifi

ZigBee - https://www.zigbee.org/

Zigbee Light Link - https://www.zigbee.org/zigbee-for-developers/applicationstandards/zigbee-light-link/ & https://www.zigbee.org/what-is-zigbee/connectedlighting/

ZigBee 3.0 - https://www.zigbee.org/zigbee-for-developers/zigbee-3-0/

RFC 2616 - https://tools.ietf.org/html/rfc2616#section-6.1.1


### 4.3 Appendices

Figure 1: Case Diagram of Light System

Figure 2: TJB Block Diagram

Figure 3: TJB Circuit Diagram

Figure 5. Server-to-light module using Zigbee Communication