

IoT Smart Light

DESIGN DOCUMENT

Sr. Design Team 21-20
Client: Govindarasu Manimaran
Adviser: Gelli Ravikumar

Team Members and Roles:
Hamza Mostafa Sayed Mahmoud - Hardware
William Gavins - Hardware
Xinlei Yu- Software
Nathan Orts - Software

Team Email: sddec21-20@iastate.edu
Team Website: <https://sddec21-20.sd.ece.iastate.edu>

First edition: March 9 /Version 1.0

Executive Summary

Development Standards & Practices Used

This system will use the Zigbee wireless communication protocol for communication between the smart lights and the Zigbee controller.

Summary of Requirements

- Lights must be able to operate continuously for 8 hours.
- Lights must be able to change to at least 4 different colors.
- Lights must operate wirelessly while being controlled from a host computer connected to the PowerCyberLab simulation.
- Latency between a relay status change and light status update must be less than one second.
- Light enclosures must be entirely self-contained.
- Software interface between the user and the light system must be easy to use and understand.

Applicable Courses from Iowa State University Curriculum

- CPRE 281 & 288 & 430 & 450 & 489
- COM S 309 & 311
- EE 201 & 230

New Skills/Knowledge acquired that was not taught in courses

- PCB design is by far the most important skill we are to acquire outside of class.
- Embedded IoT communication protocol (i.e Zigbee) usage is also another skill we are to learn

Table of Contents

1	Introduction	4
1.1	Acknowledgement	4
1.2	Problem and Project Statement	4
1.3	Operational Environment	4
1.4	Requirements	4
1.5	Intended Users and Uses	4
1.6	Assumptions and Limitations	5
1.7	Expected End Product and Deliverables	5
2	Project Plan	5
2.1	Task Decomposition	5
2.2	Risks And Risk Management/Mitigation	6
2.3	Project Proposed Milestones, Metrics, and Evaluation Criteria	6
2.4	Project Timeline/Schedule	6
2.5	Project Tracking Procedures	6
2.6	Personnel Effort Requirements	7
2.7	Other Resource Requirements	7
2.8	Financial Requirements	7
3	Design	7
3.1	Previous Work And Literature	7
3.2	Design Thinking	7
3.3	Proposed Design	7
3.4	Technology Considerations	8
3.5	Design Analysis	8
3.6	Development Process	8
3.7	Design Plan	8
4	Testing	9
4.1	Unit Testing	9
4.2	Interface Testing	9
4.3	Acceptance Testing	9
4.4	Results	9

5	Implementation	10
6	Closing Material	10
6.1	Conclusion	10
6.2	References	10
6.3	Appendices	10

List of figures/tables/symbols/definitions (This should be the similar to the project plan)

1 Introduction

1.1 ACKNOWLEDGEMENT

We would like to express our sincere gratitude to Dr. Gelli Ravikumar as our advisor and Dr. Govindarasu Manimaran as our client and secondary advisor. They both provided invaluable guidance, feedback, and resources to ensure the successful completion of this project.

1.2 PROBLEM AND PROJECT STATEMENT

- Problem Statement

Currently, when running power grid simulations in the PowerCyberLab in Coover Hall, the status of relays within the simulated power grid is shown by twelve lights mounted on the hardware running the simulations. In the past, when doing outreach demonstrations or presentations, there would be a livestream of the lights via a video camera in the lab. Though this worked, it is hard for viewers to visualize the simulated power grid and the status of the relays when the status lights are fixed to the hardware, rather than being arrayed over a map of the simulated power grid.

- Solution Approach

Our goal is to provide a way to more easily demonstrate the power grid simulations in outreach presentations. We would like to use wireless, magnetic lights in an internet of things environment, such that the simulated power grid can be projected onto a surface and our lights can be placed where the relays are located in the grid, providing a better visual aid for the demonstrations.

The deliverables for this project are a wireless, magnetically mountable LED light, a Zigbee coordinator that controls the lights, and software that configures the lights to fit the current simulation and transmits the relay statuses to the lights to update the demonstration in real time.

1.3 OPERATIONAL ENVIRONMENT

Our wireless light system will be used in indoor presentation venues, such as classrooms, lecture halls, or electronics labs. Our system must have access to the internet so that it can communicate with a server on campus that serves as the intermediary between the light system and the power grid simulation in Coover hall.

1.4 REQUIREMENTS

Functional requirements:

- Lights must be magnetically mountable.
- Lights must be able to operate continuously for 8 hours.
- Lights must be able to change to at least 4 different colors.
- Lights must operate wirelessly while being controlled from a host computer connected to the PowerCyberLab simulation.
- Lights must be operable while charging.
- System must accept input from up to 100 Simulation Components.
- System must be capable of controlling up to 100 lights.
- System must be capable of mapping relay inputs to light outputs via a user interface.
- System must be capable of charging at least 8 lights at once.

Non-Functional Requirements:

- All system components together must be easy to move through standard door frames.
- Magnetic mounting boards should not obscure any image projected upon it.
- Latency between a relay status change and light status update must be less than one second
- Relay interface components must be easy to connect.
- Light enclosures must be entirely self-contained.
- Software interface between the user and the light system must be easy to use and understand.

1.5 INTENDED USERS AND USES

This system is intended for use by members of the PowerCyber lab at Iowa State University to visually represent the status of relays in power grid simulations, though the system could be easily adapted to visually represent the status of nodes in any graph-type system, such as city streets, wireless networks, or supply chains.

1.6 ASSUMPTIONS AND LIMITATIONS

Assumptions:

- Users can read and understand english.
- Users have their own computer(s) that can run our software.
- Users are in an environment with an internet connection.
- If outside of the Iowa State University campus, users will have access to the Iowa State University VPN.
- Users have access to electricity when needing to charge the wireless lights.
- The system will be used indoors and will not be exposed to the elements.
- The system will be kept dry at all times.
- The colors for the following statuses will be: blue for functional, yellow for <intermediate>, and red for non-functional or non-responsive.

Limitations:

- System must operate at 120 volts and 60 Hertz.
- System in its entirety must fit through a standard door.

1.7 EXPECTED END PRODUCT AND DELIVERABLES

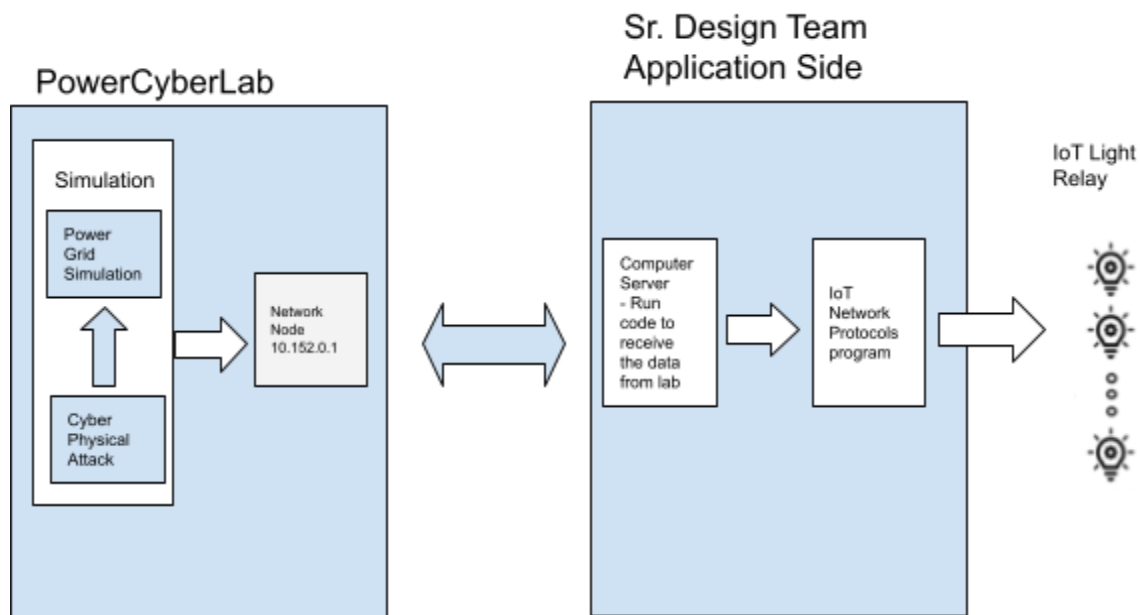
The primary deliverables for this project are the wireless LED lights, the software that runs on the user's computer which is used to configure the system, and the Zigbee coordinator that acts as the intermediary between the software and the wireless lights. We will also include a way to charge multiple lights at the same time, in the form of a charger hub.

2 Early Project Plan

2.1 TASK DECOMPOSITION

The technical parts will be divided into two parts: hardware and software. Hardware components are PCB board design, power charger system, and circuits design. The software components are GUI design, IoT network protocols, and software control units. Both hardware and software are subjective to update if there is a more efficient or useful case to come up.

The plan and design phase is divided into three parts. Because our project is adopted from the senior design team 19-16, our first part of the senior design is to understand the previous design and implementation. Once we comprehend the previous team's accomplishments, our second task is to iterate our own design upon the previous or even rebuild it more efficiently, securely, and better in performance. Lastly, after we finalize the our prototype implementation plan for the design, we will need approval from both the client and advisor. This will require the team to revise our design and ensure it meets our expected standards.



2.2 RISKS AND RISK MANAGEMENT/MITIGATION

In general, we have decided there are three levels of risks we may have in the way of completing the project. Critical level risk means a red flag, all members and the advisor/client should be notified at the first moment. An immediate meeting between the team and the advisor need be scheduled as soon as possible to discuss mitigation strategy. Moderate level risk means a warning sign, all group members should schedule a meeting to discuss the issue and give a mitigation plan for the risk. Depending on how the relativity of the issue and flexibility of the advisor, the advisor may not need to join the meeting. Small fix means some level of attention is needed. This isn't an urgent issue or something off the track. Team members can help each other on this level of risk, able to resolve the risk within the team, and no need to raise the advisor's attention. Below are some examples of each level of risk.

- Critical: Project/Mission fail due to design issue, Milestone overdue, final deliverable overdue, financial resources shortage, personnel out of duty or injury, and a new pandemic.
- Moderate: Weekly plan off track, significant technical issue.
- Small fix: Excused absence of weekly meeting etc.

2.3 PROJECT PROPOSED MILESTONES, METRICS, AND EVALUATION CRITERIA

Milestone

To guarantee that our team's project satisfies all requirements, we keep track of our progress against the schedule, keep the advisor and client updated on progress, and revise and reflect our work. Since the spring 2021 is focused on the design and plan of the project. By the end of the next design document version due, we will have more detail and specification added to the design document. This also includes any new idea we would like to implement for the project. By the end of final

design documentation, we would have an achievable design document to carry on to the implementation phase in fall 2021.

For fall 2021, we will start implementing the project as we planned. We plan to have a prototype built before the middle of the semester. We will let the advisor and client give us feedback on the prototype.

Metrics

Due to the limitation of time and the COVID-19, the team wasn't able to play with the existing software and hardware by the time of this first version of the design document. Numeric specification of metrics and evaluation criteria may be added in the next version of this document. The metrics are measured in four main aspects: power, protocol, GUI, and PCB board. Detail shows below.

Power:

1. Faster Charging

Protocol:

1. Less power consumption
2. Longer range
3. More secure

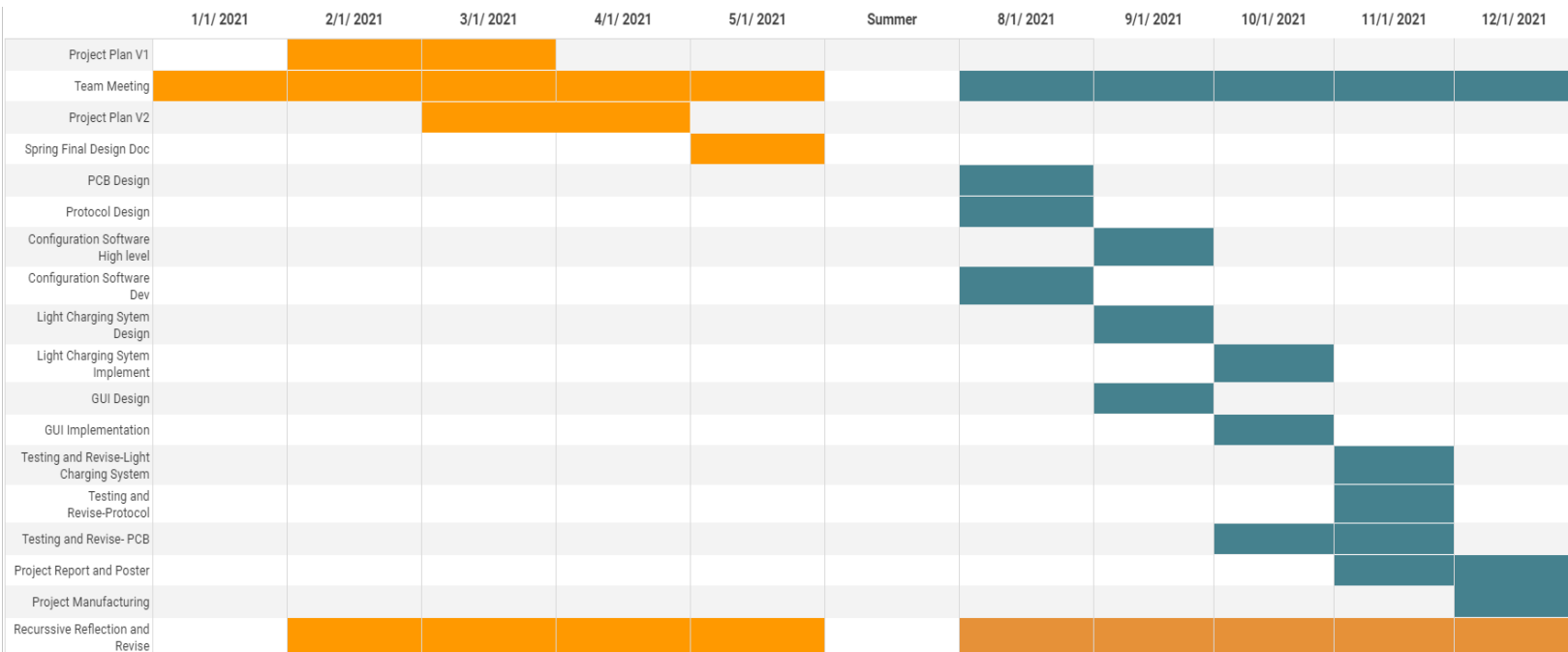
GUI

1. UI/UX easy to use and make significant updates.

PCB Board

1. Efficiency
2. Newer and more compacted design

2.4 PROJECT TIMELINE/SCHEDULE.



The Gantt chart above shows the timeline we plan to follow for this project. This chart identifies all the designing, building and testing/revising that need to be done for the project. The chart demonstrates how the system planning will happen as we plan in this first version of the design document.

This is because the timeline is designed with teamwork in mind. The tasks are separated, so that one or two people can complete the tasks reasonably within a given time frame, dedicating 5 or more hours a week. This seems to be the most effective way to break down tasks, because it would be unfair if a task or series of tasks requires more hours of investment per week than other tasks.

2.5 PROJECT TRACKING PROCEDURES

To track our team's progress, one helpful tool we utilize is the Trello board. We have our all team members and the advisor added to the Trello board. In the Trello board, we set up sections such as development modules, To Do list, Done list, meeting notes, resources, and archives. Each section keeps track of one aspect of the project. Once we move to the implementation phase, we will add a specific task section to track the progress. Overall, by using the Trello board, it minimizes miscommunication around the project with simple tools to organize tasks and track progress.

Github will be set up and configured to store the software coding. The Github issues and milestone is also a good tool to track our software development progress.

Another tool we utilize is discord. Discord is mainly for our day-to-day communication tool. We can easily set up text and voice channels. This can supplement the Trello board because the Trello board doesn't support real time communication. Any notification and risk happened, team members can set out a text in the discord channel and other team members can be notified immediately. This helps all group members stay tuned.

2.6 PERSONNEL EFFORT REQUIREMENTS

Group Member (first name)	Technical Role	Efforts in term of hours per week
Hamza	Hardware Developer	<ul style="list-style-type: none">• 2 hours team meeting, twice a week. (The frequency and length of meetings may increase when project load increases.• 1 hour meeting with our client and advisor• 5 hours of effort in design, plan, and develop the IoT smart light project. (The load may be lighter in the first week or heavier when we face some technical issue.)
William	Hardware Developer	
Xinlei	Software Developer	
Nathan	Software Developer	

2.7 OTHER RESOURCE REQUIREMENTS

As far as we have decided in our project, there isn't any other resource requirement needed. Although we may add items once we start implementing the project, most parts can be collected from the previous team and existing resources of PowerCyberLab.

2.8 FINANCIAL REQUIREMENTS

From the project proposal, there is a financial budget for this project. By now there aren't any financial requirements to be declared. Once we kick off the development of the project or realize resources from outside of the PowerCyberLab, we will promote it to our advisor and client and revise this section. The department/board will decide whether we have any financial resources.

3 Design

3.1 PREVIOUS WORK AND LITERATURE

Our project is an enhancement on previous team work (<https://sddec19-16.sd.ece.iastate.edu>). Our purpose is making the design more usable by adding more features and enhancing communication

protocol and battery time. The current product's battery can only last for 8 hours, and only has one RGB light for presenting data. However, the previous team successfully created a functional communication using the Zigbee protocol.



3.2 DESIGN THINKING

Design is currently using the Zigbee communication protocol for simulating processes running in different locations. In our design thinking, group members suggested using different protocols, i.e. LoRa and Z-Wave. Another proposed idea was using bluetooth instead of IoT. We found ZigBee however to be the best protocol to be used as it consumes the least amount of energy and allows a wider range of devices to be connected to and communicate with.

	Owner	Frequency (MHz)	Range	Power requirement	Security	Compatibility
Zigbee	Zigbee Alliance	868 - 868.6 (Europe) 902 - 928 (US)	10–100 meters line-of-sight	Low-Power, Potential Batteryless	Low, basic encryption	Compatible across Zigbee devices. DotDot OS.
Lo-RaWan	LoRa Alliance	169, 433, 868 (Europe) 915 (US)	Up to 6.2 miles or 10 km.	Low-Power	Basic 64-128 bit encryption	Depends on OEM
LTE-M	GSMA - Cellular Carriers	LTE Bands: 450-2350 (uplink)	Global	Band dependant	NSA AES-256	Application dependant
IEEE 802.11af (White-Fi)	Open - IEEE Certified	470 - 710 (Digital Dividend)	Short, up to 100m	Low	WPA	Application dependant
IEEE 802.11ah (HaLow)	Open - IEEE Certified	850 (Europe) 900 (US) 700 (China)	Up to 13 miles or 20 km.	Medium	WPA	Application dependant

3.3 PROPOSED DESIGN

- Enhancing the IoT smart Light product in order to have a two way communication method for sending and receiving data
- Increasing the features by adding new buttons and RGB light or LCD display
- Using ZigBee communication protocol over others for its low consumption of energy and ability to connect to a wide range of devices.
- Editing PCB design to allow charging the battery and using the device at the same time

3.4 TECHNOLOGY CONSIDERATIONS

The proposed design will be able to simulate a running process in a different location using multiple smart lights. The design still lacks the ability to run for longer than 8 hours, however with adding the charging and using at the same time feature, we will be able to overcome this issue. Some trade-off has to be made as the PCB is very limited in size as per the client's requirements, and therefore limited amount of features can be added. Such an issue will be overcome by getting more creative with different ways to present data using only an LCD screen or multiple RGB lights. The design will have the ability to perform two way communication which will allow us to add more features.

3.5 DESIGN ANALYSIS

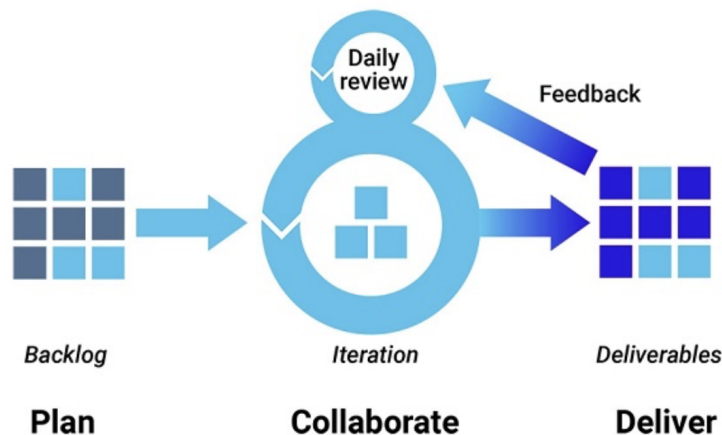
Thus far, most of our efforts have been focused towards planning, communication of different roles, learning about how to interface the different parts of the system, and writing the associated documentation and making the diagrams.

Our design is currently under development and implementation. Further tests on both hardware and software ends will be performed and therefore we will be able to test the functionality.

TBD

3.6 DEVELOPMENT PROCESS

Our team is following the Agile development process. We are using this process as our project needs to always have more features added to it, and therefore we will be planning which features we will be working on, work and collaborate on that feature till delivering it, and then back to considering new features.



3.7 DESIGN PLAN

Describe a design plan with respect to use-cases within the context of requirements, modules in your design (dependency/concurrency of modules through a module diagram, interfaces, architectural overview), module constraints tied to requirements.

The final design will be composed of 4 modules: The Transmitter Junction Box (TJB), individual light modules, a charging station for the lights, and a software user interface.

Functionally, the TJB will give power to each port of the relay boxes it is connected to. For an individual port, if the contact in that relay is closed, the power will be returned to a corresponding port in the TJB. This will tell the TJB that the corresponding port is closed, updating the TJB's memory map of ports. This memory map has 2 inputs, the direct ports that correspond directly to a light, as well as the configuration software, discussed later, that allows a user to override the port map and program their own function for the lights, in the event of a demonstration.

4 Testing

Testing is an **extremely** important component of most projects, whether it involves a circuit, a process, or software.

The project will be divided into phases, such that each phase will have time to debug and catch any large time consuming errors that could hold the team back. Because of the nature of IOT there may be hardware design revisions. Getting some level of debugged hardware prototype functioning is the first step for testing on all levels.

Each phase of testing will be depending on what modules of software are complete, as well as what revision of hardware is currently functioning. This iterative process will allow changes to be made to hardware to prevent lockups in the development process.

4.1 UNIT TESTING

Tests are broken down into units to emphasize modular testing, this way the team can debug stages one level at a time to prevent confusion. This can allow software to be broken up for easy testing

Each module will be broken down and tested individually before combining into more complicated sets. Between each unit test there will be interface tests to ensure that two quantities are communicating correctly.

After a PCB Prototype has been fabricated, it is important to ensure that the printed and soldered prototypes are matching the circuit diagram completely, this will require testing each connection and component to the parts specifications and verify the design to the original specification.

4.1.1 IOT MODULES

Each revision of the module will need the following verified:

The module can send data to the host regarding its updated state based on a button press. This will send a packet to the server via zigbee. Testing is needed to ensure that ranges are clear and packet loss is minimized.

The module can receive data from the host based on its changing state from the backend simulation.

The module can correctly cycle through all the colors from each previous color. For example when the module is currently red, it can then change the color to any subsequent color.

In the early phases this will be done on a breadboard to make sure that hardware is both functional as well as modular for rapid design changes. Testing the functionality will need to be repeated for each iteration of the PCB. At this point an artificial system will need to be in place to simulate the communication between the IOT smart light and the back end server.

4.1.2 BROADCASTER MODULE

The broadcaster module designates the iot smart light modules to reflect the simulation that takes place within the PowerCyber Lab. In order to prove that the module works when a signal is received, a simulated signal will be sent to show that the backend updates based on the input. Ensuring that the module can trigger changes in the backend is the foundation of the two way communication needed for a successful deliverable. Simulated functions used to make it “appear” that a signal is received or sent will be used to mock test this functionality.

4.2 INTERFACE TESTING

For each level of software development, testing protocol connections will ensure that communication between the broadcaster and the endpoint avoids any regressions. This can be done by effectively flooding the interface with packets to measure when the signal finally breaks down.

To test this the Xbee broadcaster will start with connecting the broadcaster to a node, then sending packets to and forth across the connection to see if the node reflects the information sent out.

4.3 OVERALL SYSTEM TESTING

When other testing has finished, and it has been determined that the hardware, interfaces, and the software is functional, actual tests with the powerCyberLab will be needed to validate the overall functionality of the system.

This will be done with a full setup of the entire system, including the software and hardware all working together to reflect the specifications of the project. Each relay will be connected

4.4 ACCEPTANCE TESTING

The best way to ensure that the end product is to the specification of both our client and advisor, both a functioning deliverable and demonstration are in order according to the requirements. This would be both to clear up usability and to teach the client how to use the system. This will also ensure that non-functional requirements are clearly completed to the clients satisfaction.

4.5 RESULTS

TBD

5 Implementation

Describe any (preliminary) implementation plan for the next semester for your proposed design in 3.3.

6 Closing Material

6.1 CONCLUSION

Summarize the work you have done so far. Briefly re-iterate your goals. Then, re-iterate the best plan of action (or solution) to achieving your goals and indicate why this surpasses all other possible solutions tested.

6.2 REFERENCES

List technical references and related work / market survey references. Do professional citation style (ex. IEEE).

6.3 APPENDICES

Any additional information that would be helpful to the evaluation of your design document.

If you have any large graphs, tables, or similar data that does not directly pertain to the problem but helps support it, include it here. This would also be a good area to include hardware/software manuals used. May include CAD files, circuit schematics, layout etc., PCB testing issues etc., Software bugs etc.