Christmas Lights Animation

System Design Document | Current Version 1.1.0

Prepared By:

Austin Wentz

Jordan Doell

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Author** | **Version** | **Comments** |
| 9/12 | Austin Wentz | 1.0.0 | Initial version |
| 12/11/12 | Austin Wentz | 1.1.0 | Fleshed out content |
| 4/20/13 | Jordan Doell | 1.2.0 | Added iOS app information |

Table of Contents

[1.0 Overview 6](#_Toc343107502)

[1.1 Scope 6](#_Toc343107503)

[1.2 Purpose 6](#_Toc343107504)

[1.2.1 Hardware 6](#_Toc343107505)

[1.2.2 iOS App and Device 6](#_Toc343107506)

[1.2.3 Miscellaneous 6](#_Toc343107507)

[1.3 Systems Goals 6](#_Toc343107508)

[1.4 System Overview and Diagram 6](#_Toc343107509)

[1.5 Technologies Overview 7](#_Toc343107510)

[2.0 Project Overview 7](#_Toc343107511)

[2.1 Team Members and Roles 7](#_Toc343107512)

[2.2 Project Management Approach 8](#_Toc343107513)

[2.3 Phase Overview 8](#_Toc343107514)

[2.4 Terminology and Acronyms 8](#_Toc343107515)

[3.0 Requirements 8](#_Toc343107516)

[4.0 Design and Implementation 8](#_Toc343107517)

[4.1 Hardware 8](#_Toc343107518)

[4.1.1 Technologies Used 8](#_Toc343107519)

[4.1.2 Component Overview 8](#_Toc343107520)

[4.1.3 Phase Overview 8](#_Toc343107521)

[4.1.4 Architecture Diagram 8](#_Toc343107522)

[4.1.5 Data Flow Diagram 8](#_Toc343107523)

[4.1.6 Design Details 9](#_Toc343107524)

[4.2 iOS app and Device 9](#_Toc343107525)

[4.2.1 Technologies Used 9](#_Toc343107526)

[4.2.2 Component Overview 9](#_Toc343107527)

[4.2.3 Phase Overview 9](#_Toc343107528)

[4.2.4 Architecture Diagram 9](#_Toc343107529)

[4.2.5 Data | Logic Flow Diagram 9](#_Toc343107530)

[4.2.6 Design Details 9](#_Toc343107531)

[4.3 Miscellaneous 9](#_Toc343107532)

[4.3.1 Technologies Used 9](#_Toc343107533)

[4.3.2 Component Overview 9](#_Toc343107534)

[4.3.3 Phase Overview 9](#_Toc343107535)

[4.3.4 Architecture Diagram 10](#_Toc343107536)

[4.3.5 Data Flow Diagram 10](#_Toc343107537)

[4.3.6 Design Details 10](#_Toc343107538)

[5.0 System and Unit Testing 10](#_Toc343107539)

[5.1 Overview 10](#_Toc343107540)

[5.2 Dependencies 10](#_Toc343107541)

[5.3 Test Setup and Execution 10](#_Toc343107542)

[5.3.1 Solid State Relay Testing 10](#_Toc343107543)

[5.3.2 Renard Controller Testing 10](#_Toc343107544)

[5.3.3 Christmas Light and Extension Cord Testing 11](#_Toc343107545)

[6.0 Development Environment 11](#_Toc343107546)

[6.1 Development IDE and Tools 11](#_Toc343107547)

[6.2 Source Control 11](#_Toc343107548)

[6.3 Dependencies 11](#_Toc343107549)

[6.4 Build Environment 11](#_Toc343107550)

[6.5 Development Machine Setup 11](#_Toc343107551)

[7.0 Release | Setup | Deployment 11](#_Toc343107552)

[7.1 Deployment Information and Dependencies 11](#_Toc343107553)

[7.2 Setup Information 11](#_Toc343107554)

[7.3 System Versioning Information 11](#_Toc343107555)

[8.0 End User Documentation 12](#_Toc343107556)

[Appendix I: List of Figures 13](#_Toc343107557)

[Appendix II: Supporting Information and Details 14](#_Toc343107558)

[Appendix III: Progress | Sprint Reports 15](#_Toc343107559)

[III.1 Sprint 1 Progress Report 15](#_Toc343107560)

[**Sponsor Description:** 15](#_Toc343107561)

[**Sponsor’s Problem/Goal:** 15](#_Toc343107562)

[**Customer Needs:** 15](#_Toc343107563)

[**Project Environment** 15](#_Toc343107564)

[**Project Boundaries** 15](#_Toc343107565)

[**Project Context** 16](#_Toc343107566)

[**Technical Environment** 16](#_Toc343107567)

[**Current Systems Overview** 17](#_Toc343107568)

[**Product Deliverables** 17](#_Toc343107569)

[**Future Product Deliverables** 18](#_Toc343107570)

[**Backlog** 18](#_Toc343107571)

[**Completed** 18](#_Toc343107572)

[**Remaining** 18](#_Toc343107573)

[**Potential Issues** 18](#_Toc343107574)

[III.2 Sprint 2 Progress Report 18](#_Toc343107575)

[**Backlog** 19](#_Toc343107576)

[**Completed** 19](#_Toc343107577)

[**Remaining** 19](#_Toc343107578)

[**iOS Application progress:** 19](#_Toc343107579)

[**App Prototype:** 19](#_Toc343107580)

[**Christmas Light Controller Progress** 21](#_Toc343107581)

[**Display Case** 21](#_Toc343107582)

[III.3 Sprint 3 Progress Report 21](#_Toc343107583)

[**Backlog** 21](#_Toc343107584)

[**Completed** 21](#_Toc343107585)

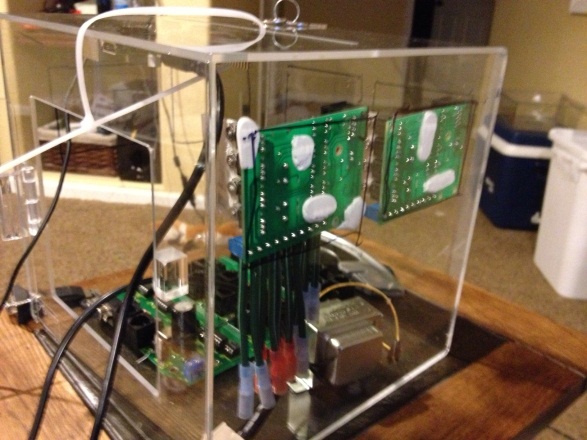
[**Remaining** 22](#_Toc343107586)

[**Christmas Light Controller Progress** 22](#_Toc343107587)

[**Display Case** 22](#_Toc343107588)

[**Before** 22](#_Toc343107589)

[**After** 23](#_Toc343107590)

[** ** 23](#_Toc343107591)

[**Christmas Lights** 23](#_Toc343107592)

# Overview

## Scope

This document covers the design specifications of the XMASLA project.

## Purpose

To make an interactive Christmas lights animation product. It will be controllable from an iOS device.

### Hardware

* Raspberry Pi
* Renard 64XC
* SSRez solid state relays

### iOS App and Device

The iPhone app will be able to send various commands to the server.

### Miscellaneous

* Christmas lights
* Extension cords
* Display case

## Systems Goals

* Choose a song from the iPhone app
* Play music and have the Christmas lights synced to the music
* Control each channel’s brightness from the iPhone app to have a fully interactive experience

## System Overview and Diagram



Figure 1 System Diagram

## Technologies Overview

# Project Overview

## Team Members and Roles

* Jordan Doell – iOS development / front end
* Austin Wentz – Hardware / back end

## Project Management Approach

The project is managed using the Agile methodology Scrum. The Scrum Master is Dr. Jeff McGough. Sprints are 3 weeks in length and weekly meetings are held. Trello is used for managing the backlog.

## Phase Overview

## Terminology and Acronyms

* SSR – Solid State Relay
* Renard – PIC-based dimming controller used to animate Christmas lights

# Requirements

See the System Requirements Document.

# Design and Implementation

This section is used to describe the design details for each of the major components in the system. This section is not brief and requires the necessary detail that can be used by the reader to truly understand the architecture and implementation details without having to dig into the code.

## Hardware

### Technologies Used

This section provides a list of technologies used for this component. The details for the technologies have already been provided in the Overview section.

### Component Overview

This section can take the form of a list of features.

### Phase Overview

This is an extension of the Phase Overview above, but specific to this component. It is meant to be basically a brief list with space for marking the phase status.

### Architecture Diagram

It is important to build and maintain an architecture diagram. However, it may be that a component is best described visually with a data flow diagram.

### Data Flow Diagram

It is important to build and maintain a data flow diagram. However, it may be that a component is best described visually with an architecture diagram.

### Design Details

This is where the details are presented and may contain subsections.

## iOS app and Device

### Technologies Used

* Macbook Pro for iOS development
* XCode IDE
* iPhone simulator for testing

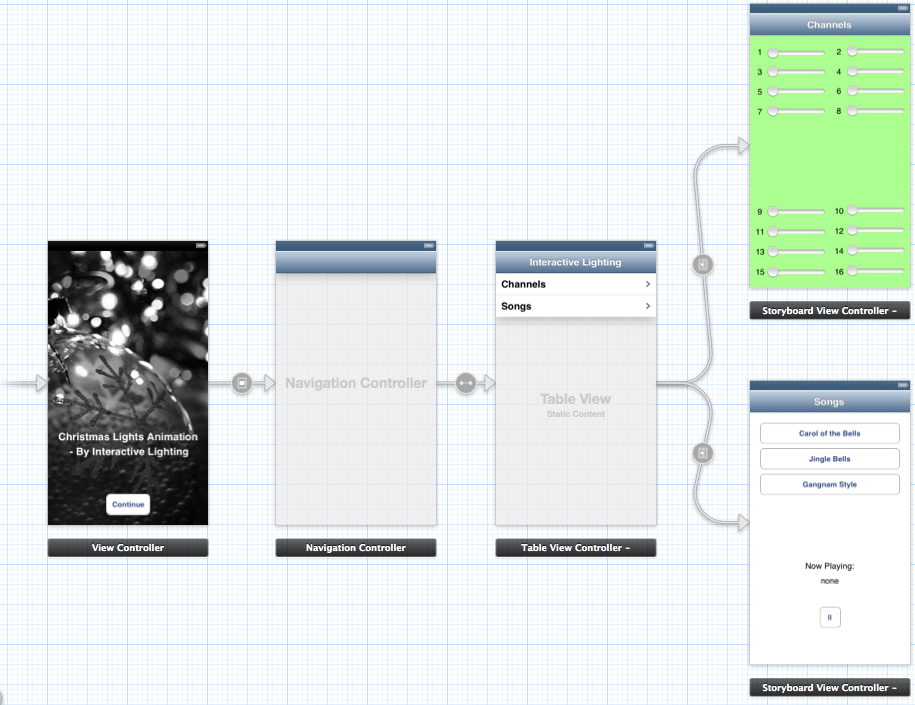
### Component Overview

* Send commands to the server
  + Play a song
  + Pause a song
* Change each channel’s brightness

### Phase Overview

This is an extension of the Phase Overview above, but specific to this component. It is meant to be basically a brief list with space for marking the phase status.

### Architecture Diagram



### Design Details

When the app is started up, there is an initial view that has a continue button. After tapping continue, it goes into a table view with a section for channels and another section for songs. The channel section allows the user to change each individual channel’s brightness. For the songs section, there is a button for each song. Tapping on a song button sends a command to play that song. Tapping the pause button pauses the song.

## Miscellaneous

### Technologies Used

This section provides a list of technologies used for this component. The details for the technologies have already been provided in the Overview section.

### Component Overview

This section can take the form of a list of features.

### Phase Overview

This is an extension of the Phase Overview above, but specific to this component. It is meant to be basically a brief list with space for marking the phase status.

### Architecture Diagram

It is important to build and maintain an architecture diagram. However, it may be that a component is best described visually with a data flow diagram.

### Data Flow Diagram

It is important to build and maintain a data flow diagram. However, it may be that a component is best described visually with an architecture diagram.

### Design Details

This is where the details are presented and may contain subsections.

# System and Unit Testing

This section describes the approach taken with regard to system and unit testing.

## Overview

Currently, the only testing implemented is for the hardware.

## Dependencies

Describe the basic dependencies which should include unit testing frameworks and reference material.

## Test Setup and Execution

### Solid State Relay Testing

Testing was done on each of the eight SSRez. The SSR’s were connected to the Renard controller and test lights were attached to the SSR’s and then a jumper wire was used to test each channel according to the following table.

|  |  |  |  |
| --- | --- | --- | --- |
| Renard Channel | Optoisolator | SSR Channel | Connect Jumper Wires between |
| 1 | U1 | 1 | U6 IC socket pin 3 and pin 14 |
| 2 | U2 | 2 | U6 IC socket pin 13 and 14 |
| 3 | U3 | 3 | U6 IC socket pin 12 and 14 |
| 4 | U4 | 4 | U6 IC socket pin 11 and 14 |

### Renard Controller Testing

The Renard 64XC was tested independently from the SSR’s by using the onboard LED lights.

### Christmas Light and Extension Cord Testing

The Christmas lights and the extension cords were each individually tested by plugging them in.

# Development Environment

The basic purpose for this section is to give a developer all of the necessary information to setup their development environment to run, test, and/or develop.

## Development IDE and Tools

Describe which IDE and provide links to installs and/or reference material.

## Source Control

Github is used for source control.

## Dependencies

Describe all dependencies associated with developing the system.

## Build Environment

How are the packages built? Are there build scripts?

## Development Machine Setup

If warranted, provide a list of steps and details associated with setting up a machine for use by a developer.

# Release | Setup | Deployment

This section should contain any specific subsection regarding specifics in releasing, setup, and/or deployment of the system.

## Deployment Information and Dependencies

Are there dependencies that are not embedded into the system install?

## Setup Information

How is a setup/install built?

## System Versioning Information

How is the system versioned?

# End User Documentation

This section should contain the basis for any end user documentation for the system. End user documentation would cover the basic steps for setup and use of the system. It is likely that the majority of this section would be present in its own document to be delivered to the end user. However, it is recommended the origin is contained and maintained in this document.

* + - 1. List of Figures

[Figure 1 System Diagram 7](#_Toc328651691)

* + - 1. Supporting Information and Details

This document will contain several appendices used as a way to separate out major component details, logic details, or tables of information. Use of this structure will help keep the document clean, readable, and organized.

* + - 1. Progress | Sprint Reports

This section will contain a complete list of all of the period progress and/or sprint reports which are deliverables for the phases and versions of the system.

* + - * 1. Sprint 1 Progress Report

Sprint Report 1

**Team Members:** Austin Wentz and Jordan Doell

**Date:** October 5, 2012

**Class:**  Senior Design

**Subject:** Sprint 1 Report

**Sponsor:** L-3: June Alexander-Knight

**Sponsor Description:**

L-3 Communications is a world class defense contractor. They play a huge role in the defense industry for the United States government. June Alexander-Knight graduated from SDSMT and since then, works for L-3. She has also been a strong supporter of SDSMT students and graduates.

**Sponsor’s Problem/Goal:**

Sync Christmas lights to music using a Linux board and controller, and control the system using an iPhone app.

**Customer Needs:**

* Linux board to control lights
* SSR’s to power on and off the strands of lights
* iPhone app to do sequences or play music
* Use sequencer software to program light show

**Project Environment**

**Project Boundaries**

* The project will have two separate environments: mobile device environment and Christmas lights controller environment
* The project’s mobile environment will be focused on iOS devices
* The project’s controller environment consists of Raspberry Pi, PIC microcontrollers, and additional circuitry to control the lights
* Communication between environments will be done over TCP/IP via JSON
* The mobile environment will be developed in Objective-C
* The controller environment will be developed in Python, and also in Clojure
* No code will need to be written for the PIC microcontrollers

**Project Context**

**Technical Environment**

The technical environment can be split into three parts: mobile device, high-level controller, and low-low-level controller.

***Mobile Device***

The iPhone is used as the mobile device. Development will be done on a Mac mini.

***High-Level Controller***

The Raspberry Pi is used as a high-level controller. It will receive commands from the mobile device, perform any required processing on command data, and send the commands to the low-level controller. The Raspberry Pi uses a Debian-like flavor of Linux. Development will be done in Linux and Windows.

***Low-Level Controller***

To directly control the Christmas lights, we are using a popular do-it-yourself light dimmer scheme called Renard. In particular we are using the Renard 64 XC design. No development needs to be done on the low-level controller.

**Current Systems Overview**



**Figure 2: System Overview**

**Product Deliverables**

No product deliverables at this point.

**Future Product Deliverables**

* Functional prototype
* Source code
* User manual / documentation
* Requirements document
* Design document

**Backlog**

**Completed**

* Purchase and configure single board computer (SBC) to act as high –level controller
* Purchase SSR pcb kit, SSR heat sinks, and Renard microcontroller pcb kit
* Analysis and research for design and requirements for project
* Start learning iOS development

**Remaining**

* Develop interface between Raspberry Pi and Renard 64XC
* Implement Renard serial protocol
* Develop prototype which switches lights on and off using predefined sequence
* Assemble additional circuitry (SSR and Renard kits)
* Purchase Christmas lights
* Purchase extension cords
* Program and configure Raspberry Pi to act as midi sequencer for lights
* Develop and implement iPhone app which controls the Christmas lights

**Potential Issues**

* Difficulty in assembling additional circuitry correctly
* Troubleshooting issues with SSRs and Renard microcontroller
* Safety issues when dealing with high voltage power sources
* Possible issues with iOS development
  + - * 1. Sprint 2 Progress Report

Sprint 2 Report

**Team Members:** Austin Wentz and Jordan Doell

**Date:** November 1, 2012

**Class:**  Senior Design

**Subject:** Sprint 2 Report

**Sponsor:** L-3: June Alexander-Knight

**Backlog**

**Completed**

* Purchase and configure single board computer (SBC) to act as high –level controller
* Purchase SSR pcb kit, SSR heat sinks, and Renard microcontroller pcb kit
* Analysis and research for design and requirements for project
* Start learning iOS development
* Assemble additional circuitry (SSR and Renard kits)
* Implement Renard serial protocol
* Purchase Christmas lights
* Purchase extension cords
* Develop prototype which switches lights on and off using predefined sequence
* iPhone app prototype

**Remaining**

* Design display case for electronic components
* Have the display case made and assembled.
* Program and configure Raspberry Pi to act as midi sequencer for lights
* Develop and implement iPhone app which controls the Christmas lights

**iOS Application progress:**

*Jordan Doell*

During Sprint 2, I have been continuing to learn Objective-C and iOS application development. I found and have been watching a podcast that covers iOS development and Objective-C. Also, James has been lecturing to me and Josh about iOS and some of the components we will need for the project. We still have a few more lectures to go, but we are making progress.

**App Prototype:**

I have gained enough knowledge of iOS so far to make a simple prototype. It is nonfunctional so far, but gives a little direction to where we are headed with the app. Below are some screenshots of the different views in the app.

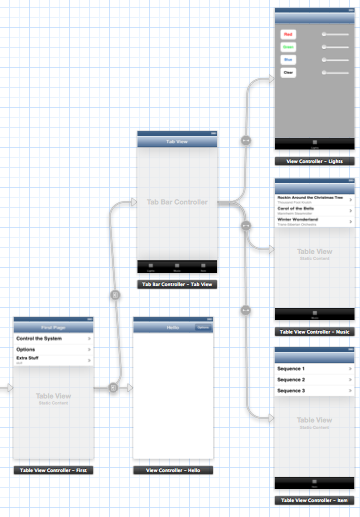
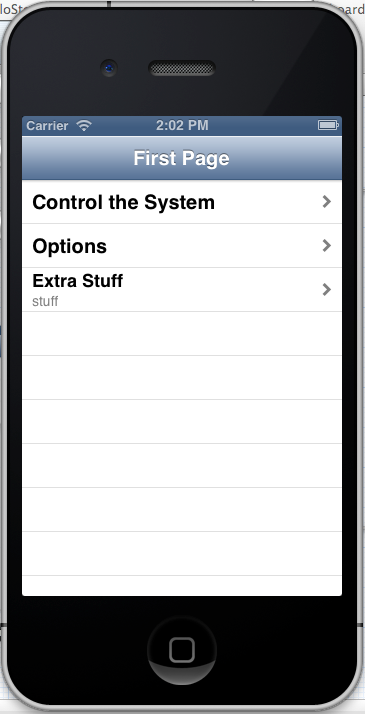


Fig. 1: Overall storyboard for the prototype Fig. 2: Main page of the app

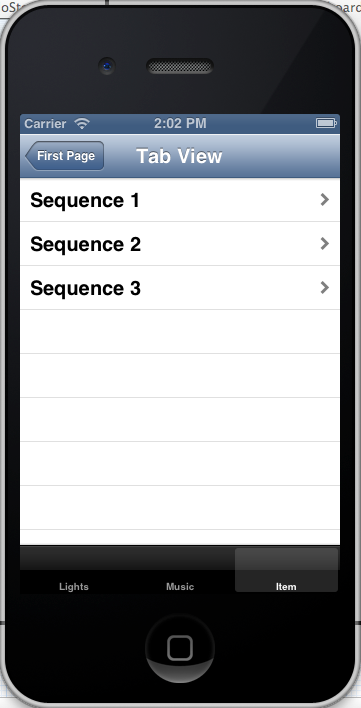
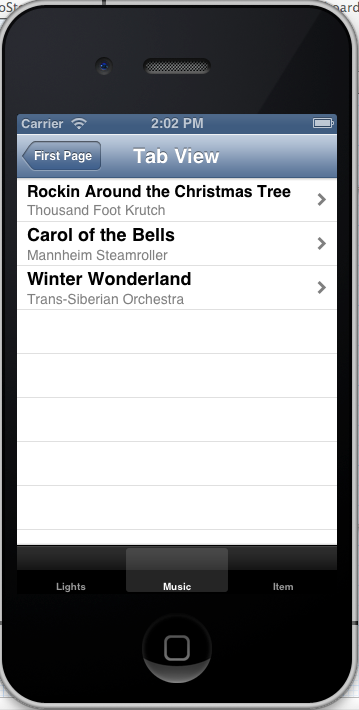
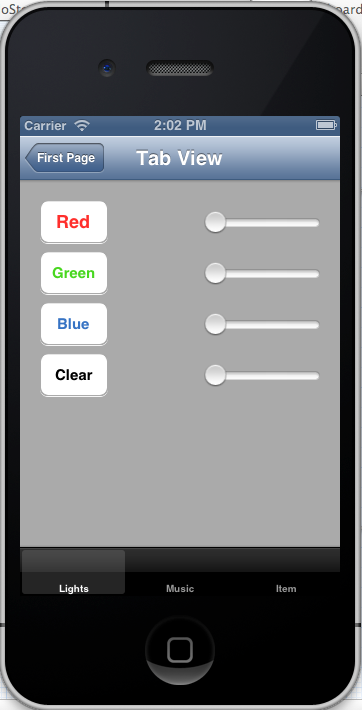


Fig. 3,4,5: Lights tab, Music tab, and Sequences tab

**Christmas Light Controller Progress**

*Austin Wentz*

Considerable progress has been made on the hardware front. The Renard 64XC and the 8 SSRez’s are now soldered and thoroughly tested. In total, the soldering took 20-25 hours. Testing took another 5 hours to complete. With the hardware assembled, I put together a simple prototype which turns lights on and off using a predefined sequence. Several short videos are available to demo the prototype.

**Display Case**

I have also been working on a design for a display case which houses the hardware. The dimensions of the case will be 16.5 inches x 16.5 inches x 12 inches. Here are some initial requirements for the case:

* Safety features – Renard 64XC and SSRez’s will only be powered when lid is closed.
* Locking mechanism to prevent theft
* Made of acrylic
* Fan for keeping SSR’s cool
* Cord management
  + - * 1. Sprint 3 Progress Report

Sprint 3 Report

**Team Members:** Austin Wentz and Jordan Doell

**Date:** December 7, 2012

**Class:**  Senior Design

**Subject:** Sprint 3 Report

**Sponsor:** L-3: June Alexander-Knight

**Backlog**

**Completed**

* Put Christmas lights on house for demo
* Film demo of Christmas lights blinking in sync to music
* Design display case for electronic components
* Have the display case made and assembled (went with pre-assembled case)

**Remaining**

* Program and configure Raspberry Pi to act as midi sequencer for lights
* Develop and implement iPhone app which controls the Christmas lights

**Christmas Light Controller Progress**

*Austin Wentz*

We made substantial progress on the Christmas light controller during Sprint 3. A display case was purchased to house the embedded hardware, I put up Christmas lights on my house, and a demo was filmed of the Christmas lights blinking in sync with music.

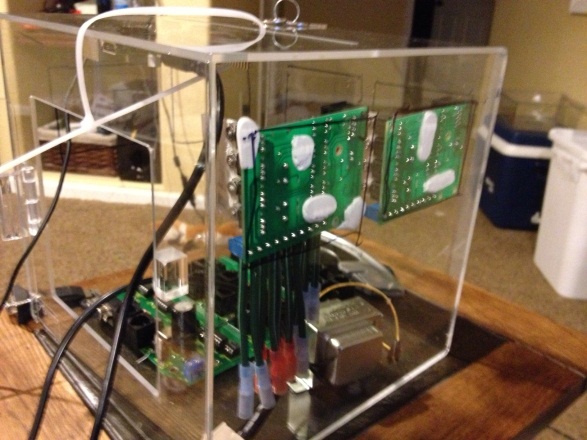
**Display Case**

Designing a display case and having it assembled was taking more time than originally anticipated, so we went with a premade temporary solution. An 8” x 8” x 8” acrylic display case, originally designed to be a ballot box, was purchased and modified. Before and after photos are shown below:

**Before**



**After**

** **

**Christmas Lights**



**iPhone App**

*Jordan Doell*

**Prototype**

The prototype app GUI was primarily made during sprint 2, but some of sprint 3 was spent researching some more of how we want the app to look. Hopefully during Christmas break, some more progress will be made.

Also, more time was spent learning a little more about iOS. I’m still getting through the podcast class, so hopefully the app will begin gaining some functionality soon. James is putting the final touches on his framework, so as soon as he gets that finished, that will be put into the app as well. The communication between the iPhone and base station will probably be the biggest challenge to get working.