

Nathan Moore
CS4670 - Professor David Wagstaff
Utah Valley University
Spring 2017
1066378

Streaming Christmas Light Show

Senior Project Report

Abstract

When christmas time comes around, christmas lights come out and are put just about anywhere you look. With computers becoming more accessible and more advanced, it is natural to apply a computer to the christmas lights show we see as we go down our neighborhood street. We see the standard lights you get at any store that can flash on and off and look like a bell is ringing, but what if we can do more than that. With a few electrical components and some great software, we can change our lights from boring on and off lights, into a real show for all to enjoy. Because many of these shows are controlled by a computer anyway, other services can be added to the show to give an even better experience for viewers. As more devices are connected to the internet, sharing the experience of a christmas light show can be seen from anywhere in the world. Along with seeing the show around the world, the internet gives us the capabilities to incorporate user's input into the show and change the display for viewers to feel like they are part of the experience as well.

Table of Contents

Abstract	1
Table of contents	2
Deliverables	3
Project Overview and Goals	4
Problem Being Solved	5
Project Time	6
Research Methodology and Findings	7
Problems encountered	11
Overview of Design	13
Appendices	14
• Original Project Proposal	14
• How To	17

Deliverables

- Java source code
- Web source code
- Final project report

Project Overview and Goals

From the outset, my main goal is to make a system that allows me to run a christmas light show to music. After seeing many of them on youtube, I wanted to create my own with a raspberry pi. I have seen many raspberry pi projects as well, and I wanted to make sure to get my own and make it do something. This was the prime opportunity to put these two together. I also wanted to make a system that would allow visitors to interact with the display as well as show their social media posts at the show and on a website. Hosting a website on the raspberry pi itself will allow the reading of the show to be passed to those who are visiting the website.

Problem being solved

Light shows are becoming more and more popular and even more extravagant each and every year, especially around the holidays. These shows give great entertainment to the viewers that are at the location of the show, but are not as easily enjoyed by others until the youtube video is posted with the show after the holiday season.

However with websites even easier to create and deploy to those around the world to visit, why not show the light show in real time to users who would like to see the show but are not able to travel to the home. To make the website not only for those who are not able to go to the show, the internet allows those at the show to interact with the show. By visiting the same website users can turn part of the show on and off and also use social media to display their own message at the show and on the website. With this project, I want to be able to allow those who make a christmas light show to be able to bring more viewer interaction to their display.

Project Timeline

Oct 1, 2016	Order and received Raspberry pi to test and create test system to make sure this project is able to be done in applicable time
Dec 30, 2016	Simple website hosted on pi to turn off and on gpios.
Jan 15, 2017	Testing electrical outlets working with gpio.
Jan 30, 2017	Twitter4j set up to be listening for hashtag search
Feb 18, 2017	@moore_lights account set up for its own workings.
Feb 23, 2017	First build of electric components completed. Can now test all outlets at once.
Feb 27, 2017	LightShowPi installed and able to run songs on Pi.
Mar 10, 2017	Front end website setup to use a gif of the show
Mar 12, 2017	Second build of electric components completed. Much better layout and easier to manage.
Mar 20, 2017	Front end website setup to use sockets using a chat room
Apr 6, 2017	Front end website using sockets to show and follow the Pi show.
Apr 17, 2017	Final light show display video filmed and uploaded to youtube.
Apr 21, 2017	Final presentation

Research Methodology and Findings

All of this project was done out of curiosity as to answer the question of “How do they do that” as well as, “How can I add a few features”. I started out with what the compantants I thought would be best to use and would allow me to do what I want. I was able to very quickly set up an apache server running on my home network. I knew that would be just about the easiest part to get working. However as i went along, the more I found that the apache server didn't have much of a way to allow me to work with gpio (General Purpose Input/Output) pins. Through this time and process, I made three different basic applications to control the gpio pins. This allowed me to see what different languages and frameworks were available to me to use for accessing and controlling different components of a light show. A friend of mine was able to point me to the direction of a framework called Pi4j that is Springboot based that has great access to the gpio pins and work really well with hosting websites. Java based with many features built in and plenty of documentation, this framework let me get off the ground rather quickly.

I was first able to get the a website that was just for accepting input from the site and turning the gpios on and off. This was an extremely satisfying and happy event. With this I felt comfortable enough to give it a break and for about a week. With that long of a break I kicked it into high gear trying to get some more of the project done. I next focused on how social media. I really want a website that fans of the show can share their ideas of the show and have that come up on the website. I also wanted that to be displayed on a tv screen at the site so those who come and see the show can

have their thoughts up on the big screen for all to see. My first social media I wanted to use, was Twitter. This was a challenge to work with. With my resolve to kick it into high gear I worked through the night to get some kind of info coming through twitter.

Using the framework Twitter4j, this was one of the most fun parts of this project. In the beginning it was slightly awkward as it always is learning a new toy. The developers for Twitter4j have some great examples of how to use the code and how to implement the features. My first set up with this did well to be able to get tweets a single time from one user. I of course made this for “@moore_lights”. However, after the initial grab of tweets that @moore_lights was tagged in, I could only listen for tweets with one specific “#” hashtag. With that achievement, I was beyond excited to be able to get that info from twitter and be able to pass that to my raspberry pi and user website. At that time I moved on to to be able to add a better user experience to the website I had set up. However after that easy step was done, I quickly came back to my twitter to see about adding better listening. I was quickly able to find how to set up twitter listeners that allowed me to grab the home timeline of a specific user as well as listen for any tweets from any number of specific users using their unique twitter account number which can be found at <http://gettwitterid.com/>. Along with listening to a specific account for tweets from them, I was also able to listen to any number to hashtag searches for when those hashtag terms are used in tweets. This has given me so many more options to try out and see what I can pull from twitter. What is also nice is the user website is stand alone and will simply request tweets from what is in the app. No need to go back

and rework the user interface to ask for a specific hashtag. However that would be a great feature to implement in the future for an admin page.

With that installed and ready to have fun with, I need to be able push play for my music and read in the information. This got me reworking my user interface to only leave at this time one user input for them to turn lights on and off again. The original light show app I thought would be great for this was Vixen 3 software. I first started playing around with this software and found that, yes it can be run on a raspberry pi, however, there is steep, and I mean steep, learning curve in learning the different parts and lights that are available to use in the software. That is a software I plan to try to work with in the future because of the customizability it gives you. However, there is a great app that I was directed to that is called LightShowPi. This is a very simple app that works specifically for the raspberry pi and does all the work for you. It does not have a user or web interface at this time, however the development notes state that the developers are working on it. This is a simple python project that controls the gpios you list out for it to use in the settings. Once those are set, the software analyzes the music file you provide and turns the gpios on and off according to the music. Extremely simple and easy to implement. The one downside is that there is no customization you can do. Again I see in the development notes that that is a feature they want to add in the future. This is the one I went on to use. The python music app runs great and uses the gpios as out, while my java application is able to use the same gpios as input.

After setting all the gpios for input in my java application, I then needed to work to get the show displayed on line. I wanted the online parts to match up with the show that

is actually being shown at the site. To do this, I turned to Sockets. I have always struggled in implementing sockets in my projects previously so to say I was nervous for this part was an understatement. Working with Springboot, I was able to find that they have java sockets implemented into the Springboot framework. That was a huge relief. But that was only half the battle. The documentation definitely helped a lot in getting my application to work with sockets and I was quickly able to get the basis chat room conversation up and running. Once that was working, I was able to change the chat room to not accept any info back from clients but to send info from the server when the gpio's state changes. This allowed my user's front end websites to turn on or off a picture of lights that would make their end look like the lights are flashing. With that completed that brought the show was ready to go.

The last thing I needed to do was to set up electrical sockets for the raspberry pi's gpio's to control. With some youtube explanation and several trips to Home Depot, these quickly came together for eight sockets to get christmas lights connected to. With the hardware set, and the software running, the show is ready to dazzle at the next holiday show.

Problems encountered

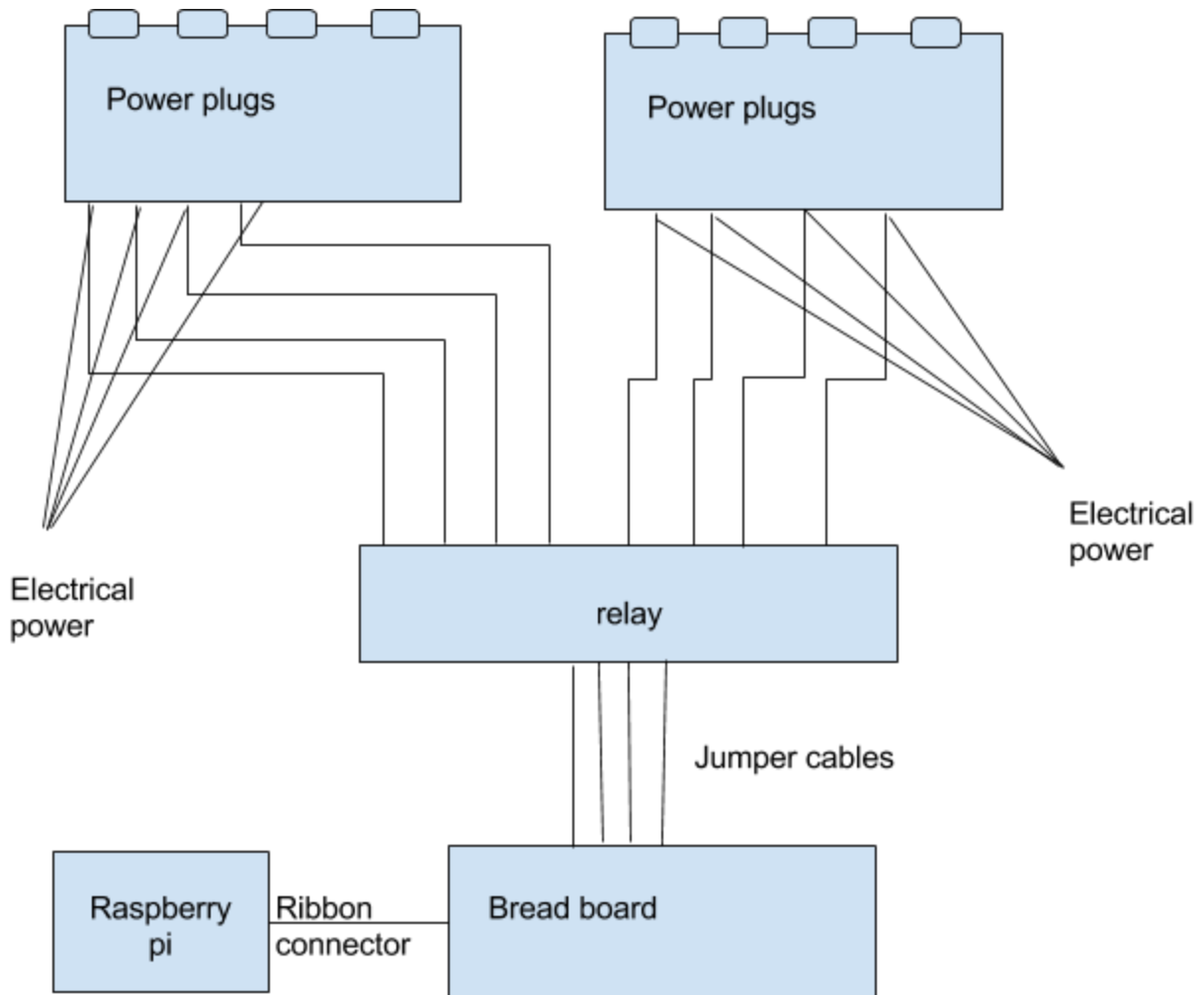
There were several issues I came across while working on this project. The first, at the very beginning of the project, when I was working on making test outlets to use, was not having any idea as to how to put a socket together. Watching plenty of youtube videos of other people's projects were great, except they never really explain how they did the sockets. They cut that part out and they just magically worked. So constantly having the fear that I had put these together wrong and was going to be electrocuted or catch something on fire was always there.

Another part that was all hardware that I did not understand how big of a deal it would be is the relay. My first relay I got and used was a mechanical relay. Now when you are simply turning a light switch on and off every once in awhile, these mechanical relays are great. They are cheap and do the job perfectly. However with a christmas light show that lights are flashing on and off several times a second, these are loud, annoying, and will wear out very fast. After my system had been set up to use the mechanical relays and was playing songs and lights were flashing on and off, I quickly started looking for a different solution. I was able to find a solid state relay that allows for the same workings as the mechanical, but without the annoying clicking and will last much longer.

The learning curve for using Springboot also threw me for a loop. It was not the most intuitive development. The system also "takes care of a lot for you". This is really nice that it does all of this on its own, however, some of the errors you get do not

describe the real issue to be looking for and it is up to you to chase down the issue you are having through a lot more searching than really should be.

Overview of Design



Appendices

Original Project Proposal

Description

Using a Raspberry Pi I will create an interactive light show for christmas light viewers so they can enjoy and interact with the display.

More Info

For the senior project that I am looking into doing I will be using a raspberry pi to create an automated system that allows christmas lights to be controlled from an admin's mobile device and also allows lights to be matched with music that will play. I am wanting to make an automated system that would provide a fun environment for viewers to see and interact with a christmas light display. I would like to incorporate motion detection to allow some part of the display to do something when viewers walk through the display as well as make something that will be accessible to viewers to open on their cell phones to interact with the display. I would also like to interact with twitter and instagram to display pictures of viewers that post with a specific hashtag for them to feel even more involved in the display and also display this same screen on the website app.

The primary goal for this project is for me to be able to learn more about hosting websites on my own device, as well as work with user feedback from mobile devices that users will be controlling. I am also very interested in applying a lot of the front end design techniques that I have been learning in my digital media minor as well. Making a mobile app/web app accessible to others to control devices on the back end will grant me special experience of interaction between outside and inside the system.

I want to dive deeper into understanding how hosting your own website and application works as well as designing front end technologies that allow for a nice user experience. For the front end developing I will be taking advantage of several frameworks that allow to the site to be accessed and nicely viewed from any device but will be geared mostly toward the mobile side. For the backend and connection from frontend to backend is where I will be needing the most help and will be doing the most research in. I will also be using a dynamic dns service to allow for my RP to host the website and application. I have heard of Pi4j that is a java implementation that allows for easier access from frontend to backend but I am not sure if that is the best way to go about running this application. To run this application I will be using the default raspbian operating system that is based on linux.

End requirements:

1. Must be able to host own website using dynamic dns service that allows for Admin login service and non login use.
2. Have at least one song programmed to work with Christmas lights
3. Be able to accept input from users mobile device to interact with the display.

Tools

- Raspberry pi
- Linux
- Apache server
 - Bootstrap
 - Php
- Pi4j - uncertain but some system to allow access
- Dynamic DNS
- Multi channel relays
- Christmas lights
- Twitter
- Instagram
- Vixen lights
- <http://www.johnsonlightshow.com/>

Licencing

- Dynamic dns/domain name registration

Deliverable

- Active website
- Final project report
- Oral presentation
- Weekly progress reports

Schedule

Dec 9, 2016	Have RP setup to accept request to server from In. RP able to accept requests turn off and on lights,(not connected to power outlet, just simple LED)
Jan 6, 2017	Build first single outlet that is able to be controlled from RP
Jan 13, 2017	Basic web page that can control lights on and off from admin screen (requires login)
Jan 20, 2017	Domain registered and active dyn dns service running to allow access to site outside of LAN, all sockets made and able to be controlled.
Jan 23, 2017	Be able to make timed programs that match with music and can run music

	and lights on and off at same time
Jan 27, 2017	Have site setup that does not require login that can change lights on and off have instagram and twitter accounts services ready to be able to be displayed
Feb 3, 2017	Improve site to make it more user friendly and user centered and visibly pleasing, add to admin site to make that more visually pleasing
Feb 10, 2017	Create more tracks to run music and lights together
Feb 17, 2017	Be able get only pictures from twitter and instagram and only display the picture on screen.
Feb 24, 2017	Add design elements to the website/work on back end design and connection
Mar 3, 2016	Practice running songs with lights
Mar 10, 2016	Put christmas lights up
Mar 17, 2016	Connect christmas lights to raspberry pi controler
Mar 24, 2016	Work on write up report
Mar 31, 2016	Polish spoken presentation
Apr 7, 2017	Present to committee!!!
Apr 14, 2017	
Apr 21, 2017	
Apr 28, 2017	

How To

LightShowPi

For getting light show pi running, it is simple to download that to your pi using apt-get.

Their website explains how to do everything very simply.

<http://lightshowpi.org/>

Twitter

You will need to set up an app for your twitter account and enter the information into the twitterAccount.properties file for your twitter service. You can do that at:

<https://apps.twitter.com/>

MooreLights

Once you have downloaded the entire project you will will need to enter the web4Lights directory. From there you must first package the software using maven. From within the web4Lights directory run the following:

`mvn package`

Once that is done you will have to start the jar that is created. From that same directory type the following:

`sudo java -jar target/pi4led-1.0-SNAPSHOT.jar`

You will see the Spring logo displayed as the server and services are loaded.

Once that is done your server and website are good to go. You just need to start playing songs using LightShowPi. HAVE FUN!