

Light Control by LytCtrl

Arduino based remote for lights/outlets controlled by Android App

9/15/2016 Version 2.00

Submitted by: Lydia Doza [lydia.doza@oit.edu](mailto:lydia.doza@oit.edu)

Submitted to: Sherry Yang

# Legal Notice

LytCtrl makes no warranties or representations of any kind concerning the accuracy or suitability of the information contained in this document for any purpose. All such information is provided “as is” and with specific disclaimer of any warranties of merchantability, fitness for purpose, title and/or non-infringement. In no event shall LytCtrl, its employees or agents be liable for any direct, indirect, or consequential damages resulting from the information provided in this document. This exclusion and limitation only applies to the extent permitted by law and is without prejudice to any express provisions to the contrary in any written license or subscription agreement from LytCtrl in respect of this document.

# Copyright Notice

**Original Work Copyright ©2018-2019 LytCtrl.**

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

# Revision History

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Date** | **Version** | **Author** | **Company** | **Comments** |
| 5/14/2016 | 1.00 | Lydia Doza | LytCtrl | First draft of proposal |
| 6/7/2016 | 1.10 | Lydia Doza | LytCtrl | Proposal Revision  Numbered sections  Dotted line in Table of contents  Section 7.1.1  Included Calvin Caldwell’s name  Section 7.1.5  Clarified location of archive  Section 8.1  Extended section  Section 9.1  Added details to  1.1, 1.2, 1.3, 1.4.1  Added requirement to describe control system  Section 9.5  Added more details for WiFi info and Outlet info |
| 9/15/2016 | 2.00 | Lydia Doza | LytCtrl | Formatting changes to conform to Wilsonville campus senior project proposal template standards. |
| 10/11/2018 | 3.0 | Lydia Doza | LytCtrl | Formatting changes to conform to new proposal requirements. Refining scope of project. |

# Signatures

## Document Submitted by:

|  |  |
| --- | --- |
| **Lydia Doza** | **Date** |

## Document Accepted by:

|  |  |
| --- | --- |
| **Instructor, OIT** | **Date** |

|  |  |
| --- | --- |
| **Instructor, OIT** | **Date** |

1. Contents

[Legal Notice i](#_Toc461707876)

[Copyright Notice i](#_Toc461707877)

[Signatures ii](#_Toc461707878)

[Change Log ii](#_Toc461707879)

[1 Introduction 1](#_Toc461707880)

[1.1 Project Goal Statement 1](#_Toc461707881)

[1.2 Major Features 2](#_Toc461707882)

[2 Customers 3](#_Toc461707883)

[3 Project Success 4](#_Toc461707884)

[4 Risk Management 4](#_Toc461707885)

[5 Technical Environment 4](#_Toc461707886)

[6 Summer Work Plan 5](#_Toc461707887)

[Appendix A Glossary 5](#_Toc461707888)

[Appendix B Outlet Assembly Wiring Diagram 5](#_Toc461707889)

[Appendix C Microcontroller Data Sheet 6](#_Toc461707890)

# 

# Introduction

## Purpose

This document serves as a proposal for the senior project Android App, *LytCtrl*. There are many others like it, but this one describes one mere Home Automation app.

This proposal lays out the features and functionality to be implemented in the project, pending approval.

## Scope

This proposal provides a coarse overview of the product requirements

It describes:

* For whom the document is intended
* Project management
* A general description of the project
* Product requirements
* User profiles
* Glossary
* Additional documentation for the project

## Intended Audience

This proposal was written primarily for my Senior Project Advisor, Sherry Yang, and for those who happen to read my GitHub. I’m looking at you, recruiters and future employers.

# Project Management

## Change Management Procedure

The CAT team will handle change requests as the need arises.

### CAT Team

The CAT team is made up by the Senior Project Advisor, Sherry Yang, and the senior year student, Lydia Doza.

### Medium

Change requests will be managed through email within the CAT Team and managed in the scrum wiki

### Protocol Response Time

Response to change requests will be analyzed and responded to within 3 business days.

### Impact Analysis

Both members will evaluate the affects a change will have on the project regarding functional requirements, dependencies, and the impact on the timeline.

### Archive

Change requests will be archived in the scrum wiki for reference.

## Software Delivery, Installation, and Acceptance Criteria

The Android App and the Raspberry Pi software will have installation instructions and the source code available on GitHub.com at <https://github.com/LydiaDoza/LightControl>. The acceptance criteria will depend on the functional requirements being met and usability.

## Documentation and Online Help

Project management documentation and tracking is available at <https://tree.taiga.io/project/lydiasaurus-lytctrl-senior-project/>.

Online help is available in the GitHub repository at <https://github.com/LydiaDoza/LightControl>.

## Project Risks

Some of the risks this project make face include the communication between the server on the Raspberry Pi and the NanoLeaf Rhythm module, successfully setting up the server on the Raspberry Pi, and syncing information between the Android App users and the server.

## Customer Responsibilities

The customer will be responsible for setting up the server on the Raspberry Pi (including securing it) and providing their own NanoLeaf Light Panels with Rhythm Module.

## Status Reporting

Project status will be reported once a week to the senior project advisor and will include:

* Work completed this week
* Work to be completed next week
* Issues, their severity, and possible solutions

# System General Description

LytCtrl will deliver an android app and a server software package to control NanoLeaf Light Panels.

## Project Summary

Smart lighting is becoming more popular, but control of the lights has limited functionality. Typically, smart lights allow their color and brightness to be changed manually with limited automated color changing or turning on/off.

*LytCtrl* will give more control to owners of the NanoLeaf Light Panels with Rhythm modules by adding time scheduling and user profiles so that multiple users can control the same lights.

The system will have four major subsystems:

1. a WiFi router,
2. a server (in this project a Raspberry Pi 3 is used but any server should work),
3. the NanoLeaf Light Panels with Rhythm Module,
4. and an android app.

*LytCtrl* focuses on the user’s perspective by making installation intuitive. The software package for the Raspberry Pi server will lead the user through a step-by-step wizard for installation and security setup (assuming the Raspberry Pi is used in UI mode). Upon launching the Android app for the first time, the user will be prompted to log in. The Android app will then search for lights nearby. The user will then be able to see what orientation their lights are in and start programming their color and brightness. In a separate menu on the Android app, the user will be able to schedule when they want their lights to be turned on and which “light profile” (color/brightness/transition) to use.

The secondary focus of *LytCtrl* is to help develop home use of Internet of Things (IoT), the idea that everyday objects have some sort of network connectivity. In this case, it is the network of the computer, smart lights, and android devices. *LytCtrl* will be on the leading edge of this new, IoT industry.

## Perspective

### History/Prior Releases

No prior releases of this app have been made.

## Major Subsystems

* Android Phone (Samsung Galaxy S6 for this project)
* Raspberry Pi 3
* Wireless Router
* NanoLeaf Light Panels
* NanoLeaf Rhythm Module

|  |  |
| --- | --- |
| **Technical Area** | **Technical Tool Used** |
| Programming Language(s) | Java |
| Database | MySql for Linux platform and SQLite for Android |
| Middle Tier | Ubuntu server |
| Client Tier / User Interface | Android app |
| Networking | Bluetooth within the home for outlets to connect to server/gateway.  WiFi to connect Android device to setup server/remote access. |
|  |  |

I will be using online resources to learn about the tools as I build this project. I have limited to no exposure to any of the tools listed prior to building this project.

## Relation of System to Existing System(s)

The *LytCtrl* Android app will depend on the Android SDK. The Raspberry Pi server will use MySql and the NanoLeaf API in order to store user data and communicate with the NanoLeaf Light Panels.

## Hardware Platform Description

* Android phone (development and testing on Samsung S6)
* Raspberry Pi 3
* NanoLeaf Panels and Rhythm Module

## Software Platform Description

* Android version 7.0 (Nougat)
* MySQL Community Server version 8.0.12
* SQLite for Android
* NanoLeaf Open API

## Third Party Libraries

* NanoLeaf Open API

# Product Requirements

## Functional

* + 1. **Android App**
       1. Ability to sign up for account
       2. Ability to sign into account
          1. Allow to automatically sign into Android (via Gmail)
       3. Pair Android app with Ubuntu server over locally connected WiFi
          1. Save IP address of server
       4. Shows user outlets
          1. Name of outlet
          2. Organized by room
          3. Current on/off status
          4. Allows user to turn outlet on/off
       5. Allows user to set a timed action
          1. Outlet will be turned on or off at time chosen by user
    2. **Admin Access stored to “global” server**
       1. Add new outlets
       2. Remove outlets
       3. Edit outlet name
       4. Add rooms
       5. Edit rooms
       6. Edit outlet room
       7. Has access to every outlet in house
       8. Add/remove any outlet’s list of users
       9. Grant admin privileges to other users
    3. **Raspberry Pi 3**
       1. Hosts the home Ubuntu server
          1. Pair server with Android connected on local WiFi
       2. Send IP address of server
       3. Update MySQL Database with user credentials
          1. First user paired is Admin
          2. Android Client communication
       4. Receive instruction
          1. Send Bluetooth packet to specified OutletID
       5. Receive status request
          1. Send status request packets to specified OutletIDs
          2. Receive Bluetooth packets

Interpret and update database of outlet status

* + 1. **Outlets**
       1. Relay default position set to off
       2. Bluetooth
    2. Set Bluetooth Name (20 character limit)
    3. Pair Bluetooth with server
    4. Change Bluetooth pair password
    5. Test Bluetooth communication
    6. Send state of relay (on/off position) over Bluetooth
    7. Receive instruction from Bluetooth server
       1. Microcontroller
    8. Save paired Bluetooth info to server
       1. BluetoothName
       2. BluetoothPairingPassword
    9. Interpret incoming Bluetooth packet
       1. Instruction
          1. Set pin to output

Turn relay on

Turn relay off

* + - 1. Status Request
      2. Send Packet
         1. BluetoothName
         2. Relay Status (on/off)

## Performance

## Reliability

## Data Description

## Security/Safety

## Constraints

# User Profiles

# Glossary

# Appendices

## 