

Light Control by LytCtrl

Arduino based remote for lights/outlets controlled by Android App

9/12/2016 Version 1.2

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 ©2016-2017 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.

# Signatures

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

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

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

# Change Log

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Author** | **Company** | **Version** | **Date** | **File Name** | **Comments** |
| Lydia Doza | LytCtrl | 1.00 | 5/14/2016 | Senior Project Proposal | First draft of proposal |
| Lydia Doza | LytCtrl | 1.10 | 6/7/2016 | Senior Project Proposal.docx | 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 |
|  |  |  |  |  |  |

Contents

[Legal Notice i](#_Toc461457249)

[Copyright Notice i](#_Toc461457250)

[Signatures ii](#_Toc461457251)

[Change Log ii](#_Toc461457252)

[1 Introduction A](#_Toc461457253)

[1.1 Project Goal Statement B](#_Toc461457254)

[1.2 Major Features C](#_Toc461457255)

[2 Customers D](#_Toc461457256)

[3 Project Success E](#_Toc461457257)

[4 Risk Management E](#_Toc461457258)

[4.1 Risk Mitigation Plan E](#_Toc461457259)

[5 Technical Environment E](#_Toc461457260)

[8.2.2 Major Subsystems E](#_Toc461457261)

[8.1.4 Hardware Platform Description F](#_Toc461457262)

[8.2.5 Software Platform Description F](#_Toc461457263)

[6 Summer Work Plan G](#_Toc461457264)

[Appendix A Glossary G](#_Toc461457265)

# Introduction

Home automation systems are still new to the market, which means the cost of buying one is high. *LytCtrl* (pronounced light control) aims to provide an economical light automation system with android app controlled outlets.

*Light Control* will provide an inexpensive option to users who wish to replicate the setup. This project will facilitate a “do it yourself” approach to adding android app controlled lights in the home.

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 will work),
3. the outlet assembly,
4. and an android app.

*LytCtrl* focuses on the user’s perspective by making installation intuitive. The software package for the home computer will lead the user through a step-by-step wizard for installation and security setup. Each outlet cover has a Bluetooth chip, allowing communication to the home computer. When a new Bluetooth device is discovered, the home computer notifies the administrative Android user. The Android user can setup a new outlet and assign it at product type (light, alarm clock, tv, etc) and a nickname. Later, they can check to see which outlets are on or off. With a quick tap, they can turn off their bedside lamp they forgot to turn off.

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, outlets, and android devices. *LytCtrl* will be on the leading edge of this new, IoT industry.

## Project Goal Statement

LytCtrl will deliver an android app, server software package, and an outlet cover to turn an outlet on and off by June 2017.

## Major Features

1. **Android App**
   1. Ability to sign up for account
   2. Ability to sign into account
      1. Automatic if signed 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
         1. Send IP address of server
         2. Update MySQL Database with user credentials
            1. First user paired is Admin
      2. Android Client communication
         1. Receive instruction
            1. Send Bluetooth packet to specified OutletID
         2. 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
      1. Set Bluetooth Name (20 character limit)
      2. Pair Bluetooth with server
      3. Change Bluetooth pair password
      4. Test Bluetooth communication
      5. Send state of relay (on/off position) over Bluetooth
      6. Receive instruction from Bluetooth server
   3. Microcontroller
      1. Save paired Bluetooth info to server
         1. BluetoothName
         2. BluetoothPairingPassword
      2. 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)

# Customers

The main customers for this project are the type of users who like “Do It Yourself” (DIY) projects. All of the design work created for this project will be published online. There will be step by step instructions to re-create this project for those who wish to teach themselves about writing code, basic electronics, and networking.

# Project Success

Response time from toggling a light on/off should be within one second.

# 4 Risk Management

Project risks include the security of the wireless router for each house, the secure Bluetooth connection between the Raspberry Pi and each outlet, and the reliability of the setup for each outlet (the connections between outlet and the relay, the relay and the microcontroller, the microcontroller and the Bluetooth module).

Risks

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Risk ID** | **Description** | **Impact (1-10)** | **Likelihood (1-10)** | **Priority (Impact \* Likelihood)** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

## 4.1 Risk Mitigation Plan

# 5 Technical Environment

### 8.2.2 Major Subsystems

*Light Control* will be broken up into three major subsystems: the Raspberry Pi, the outlet assembly and the Android app. The Raspberry Pi will be host the server, which will remain on at all times, connected to a wireless router either via Ethernet or WiFi. The outlet assembly will house the microcontroller assembly. The server will have a primitive GUI available in order to access connected devices (Android devices and outlets) and the database. The Android app will provide a GUI for outlet control.

### Hardware Platform Description

* Android Phone (Samsung Galaxy S6 for this project)
* Raspberry Pi 3
  + Ubuntu Server
* Wireless Router
* Outlets
  + ATTiny85 microcontroller development board
  + 220VAC @ 20A Relay
  + JBtek HC-06 Bluetooth to UART converter

### 8.2.5 Software Platform Description

* Android Version 6.0 (Marshmallow)
* Android API Version 23
* Ubuntu Server (v16.04)
* Embedded C for microcontroller communications

|  |  |  |
| --- | --- | --- |
| **Technical Area** | **Technical Tool Used** | **Where You Learned the Technology** |
| Programming Language(s) |  |  |
| Database |  |  |
| Middle Tier |  |  |
| Client Tier / User Interface |  |  |
| Networking |  |  |
|  |  |  |

# 6 Summer Work Plan

# Appendix A Glossary

*[This section is included for future use]*