Not exactly the Internet of Things for Outdoor Lighting: Requirements Document

Oregon State University CS Senior Capstone Group 22
Malcolm Diller, Sean Rettig, Evan Steele
Client: Victor Hsu

June 10, 2016

Contents

1 Team

1.1 Team Name

Cupcake Warriors

1.2 Team Members

Malcolm Diller	dillerm@oregonstate.edu
Sean Rettig	rettigs@oregonstate.edu
Evan Steele	steelee@oregonstate.edu

1.3 Client

Victor Hsu
Oregon State University
Phone: 541-737-4398
Email: hsuv@onid.orst.edu

2 Introduction

2.1 Problem Statement

Outdoor lighting seems like a simple problem to solve, but the solutions on the market today are less than ideal. The standard transformer/timer combos available in the big box stores are rudimentary and clunky at best (constantly needing to be adjusted for the changing sunset and sunrise times), but are reasonably priced. The new-generation smart apps for home automation are flexible and fancier, but are quite spendy and lock you into a specific protocol. So why not use an open platform running on commodity hardware? Easy to use, reasonably priced, and highly customizable–that is our goal.

2.2 Project Description

Our system will consist of a wireless network of tiny "client" computers that each control up to 4 sets of lights and are controlled by a central "server" computer, which will automatically send out commands to the clients when it's time to turn on or off. The central node will run a control program that can be easily accessed via a touch screen, a web browser, or a mobile device, where the user can locally or remotely control each light individually. Want your lights to turn on at sunset and then dim gradually as the sun rises? Simple. Want your lights to flash when you're throwing a party? Just press a button. The control interface will allow users to easily set "rules" for what their lights do and when, depending on the time of day, the sun/moon position, and potentially even triggers such as weather conditions or calendar dates. This system will be easily extensible to potentially control other devices as well, such as garage doors, sound systems, and more.

2.3 Design

Specifically, we plan to use a Raspberry Pi running Linux as the central server computer. The Raspberry Pi will run a web server program that will allow nearby devices (such as laptops, phones,

or tablets) to control it over a wireless network, or if connected to the home's internet connection, from practically anywhere in the world. The Pi will also have a small touchscreen connected to it with the website open in a browser, so the user has a dedicated interface for the device. The Pi will connect wirelessly to small wifi-enabled microcontrollers placed throughout the home, each of which are connected to up to 4 relays to control 4 sets of lights.

3 Requirements

3.1 Critical Requirements

The following requirements are critical to the basic functionality of the system and must be functional prior to the expo in spring term.

1. Device control
   (a) Server is loaded with Linux and is able to boot up to a graphical interface that can be viewed and controlled by the device's touchscreen.
   (b) Server is able to start the server control program automatically when powered on.
   (c) Clients are loaded with the client control program, which runs automatically when the device is powered on.
   (d) Client control program is able to automatically discover relays/lights that are connected to it.
   (e) Client control program can control the relays to power the lights on/off individually.
   (f) Server is able to act as a wireless access point that each client can connect to.
   (g) Clients can be paired with a server by putting the server into "pairing mode" (by pressing a button either on the user interface or on the Pi itself) that will last for a short time (just a few seconds). The server will then wirelessly broadcast UDP packets that nearby unpaired clients will see and use to connect to the server. This system will allow multiple servers to be used in close proximity with separate sets of clients.
   (h) Clients are able to communicate to the server which lights are available for control.
   (i) Server is able to send light toggle instructions wirelessly to clients.
   (j) Clients are able to read light toggle instructions wirelessly from the server.
   (k) Clients are able to perform light toggle instructions received from the server.
2. User interface
   (a) Main control program on server serves a web site with a control interface for the user.
   (b) The web interface is accessible via the system's built-in touchscreen.
   (c) The web interface is accessible to other devices like laptops and phones via a local wireless network.
   (d) The web interface displays a list of connected clients.
   (e) The web interface allows users to give each client a "nickname" for easy identification.
   (f) The web interface displays a list of connected lights.

(g) The web interface allows users to give each light a "nickname" for easy identification.

(h) The web interface contains buttons for each light that can be used to toggle lights individually.

(i) The web interface contains the ability to put lights into "groups" that can be controlled together all at once.

(j) The web interface displays a list of light groups.

(k) The web interface allows users to give each light group a "nickname" for easy identification.

(l) Groups can also be nested into other groups to create hierarchies of lights. For example, there can be two groups called "Front Porch" and "Back Porch" that control the front and back porch lights, respectively. Both groups can then be inside another group called "House". If the "House" group is toggled on, then both the front and back porch lights will be toggled on. If just the front porch lights are then toggled off, the back lights will stay on.

(m) The web interface provides the user with options to toggle lights/groups based on rules, as described in the below "Rules" section.

3. Rules

(a) Lights can be toggled manually, overriding any rules (as described in the previous section). The manual setting will stay in effect until another rule is triggered. For example, if your lights are set to turn off every morning at 6am, but you manually turn them on at 7am, they will stay on all day until the next morning at 6am.

(b) Rules can be toggled manually to temporarily disable them. For example, if you have your lights set to turn on every night at 8pm, but are going on vacation for a week and don't need them, you can turn that rule off and it will stop triggering until you turn it on again, keeping the lights off until after you get back. This way, the rule doesn't need to be deleted and completely recreated from scratch.

(c) Lights can be toggled based on time of day (e.g. "turn on at 8pm and turn off at 6am").

(d) Lights can be toggled based on day of week (e.g. "turn on during Wednesdays").

(e) Lights can be toggled based on day of month (e.g. "turn on every 1st of the month").

(f) Lights can be toggled based on day of year (e.g. "turn on every January 1st").

(g) Lights can be toggled based on specific dates (e.g. "turn on from Dec. 24th at 2am to Dec 27th at 8pm").

(h) Lights can be toggled based on sunrise/sunset times (using sunset/sunrise times either preloaded onto the Pi or retrieved from an Internet service, and using a geograpic location either entered by the user or determined through an Internet service)

(i) Multiple rules can be applied to a single light/group using an AND/OR system to combine the rules (e.g. "turn on between 8pm and 6am AND turn on on Wednesday" will turn the lights on between 8pm and 6am on Wednesdays, and will be off the rest of the time, whereas using an OR instead of an AND would cause the lights to be on every day from 8pm to 6am and also on all day during Wednesday).

(j) Lights can be toggled based on the toggle state of its parent group (e.g. by default, a light will turn on if its parent group turns on, but it can also be set to turn off if the parent group turns on).

(k) Rules can be set to toggle early/late by a constant or random period of time. For example, lights can be configured to turn on exactly 30 minutes after sunset, or perhaps randomly between 6pm and 6:30pm.

(l) Lights within a group can be set to individually toggle early/late by a random period of time so that they toggle in a staggered fashion. For example, if there are 5 lights in a group and are set to toggle off at 8pm, you can have the first light randomly toggle a few seconds before 8pm, then the next a few seconds later, etc. in a random order and with randomized times.

3.2 Stretch Goals

The following requirements are optional; they are not critical to the basic functionality of the system and may be implemented only if time permits.

4. Device control

   (a) Lights can be toggled over a user-specified period of time, e.g. a light can gradually turn on and grow brighter over the course of 30 seconds.

   (b) System also controls devices other than lights

      i. Garage door openers (e.g. users can program their garage door to automatically close at night).

      ii. Music players (e.g. users can set music to automatically play in the evenings from an attached music player or from the Internet).

      iii. Holiday decorations (e.g. users can set snowglobes or animatronic deer to automatically turn on at night)

      iv. Sprinkler systems (e.g. users can set their sprinklers to turn on from 4am to 5am).

5. User interface

   (a) The web interface contains sliders for each light that can be used to change the intensity of lights individually.

   (b) The web interface is accessible over the Internet (i.e. the user can control the system from anywhere in the world with an Internet connection, not just at their home)

      i. The web interface is accessible only to the homeowner or other authorized individuals (to prevent the general public from being able to view/control the user's lights). This could be implemented by asking the user to provide a password when logging into the system for the first time from a particular device. The device could then stay logged in via a cookie until the user logs out.

      ii. The web interface is hosted on a remote server for high availablity and the lack of need for the user to perform port forwarding on their router. If the web server is running on the user's existing home network, the site won't be available unless the user manually opens their router's interface and creates a port forwarding rule, and also won't be available in the case that the user's router/modem loses power or

connection to the Internet. If the site was hosted externally, the user would still be able to view how the system is currently set up and make changes that will apply once the user's power and/or Internet connection are restored.

6. Rules

   (a) Lights can be toggled based on weather conditions (e.g. lights can turn on automatically if it starts raining). This would require an Internet connection or light/moisture sensors.

   (b) Lights can be toggled based on a schedule in the user's calendar (e.g. a user could create a lighting schedule on their Google calendar and it would "sync" with the lighting system). This would require Internet connection.

   (c) Lights can be toggled based on moon position or other celestial data (e.g. lights could turn off during a full moon or solar eclipse for better viewing). This would likely require either an Internet connection or a cache of preloaded data and the user's location.

   (d) Lights can be toggled based on input from attached sensors

      i. Motion sensors (e.g. lights turn on when someone walks up to the front door)

      ii. Light sensors (e.g. lights turn off at a certain brightness level, regardless of time of day, weather, moon phase, etc. This would also account for an area's brightness changing with the seasons and tree cover).

      iii. Moisture sensors (e.g. lights turn on when it's wet outside, negating the need for an Internet connection to determine if it's raining andpossibly providing more accurate results. Could also potentially be used in combination with sprinkler integration to stop watering once a certain moisture level has been reached).

      iv. Sound sensors (e.g. lights turn on after a loud noise, or can strobe according to the beat of music that is playing).

   (e) Lights/groups can be set to gradually dim/brighten over a set period of time. For example, lights can turn on and gradually brighten from sunset to 30 minutes after sunset, after which they are at full brightness.

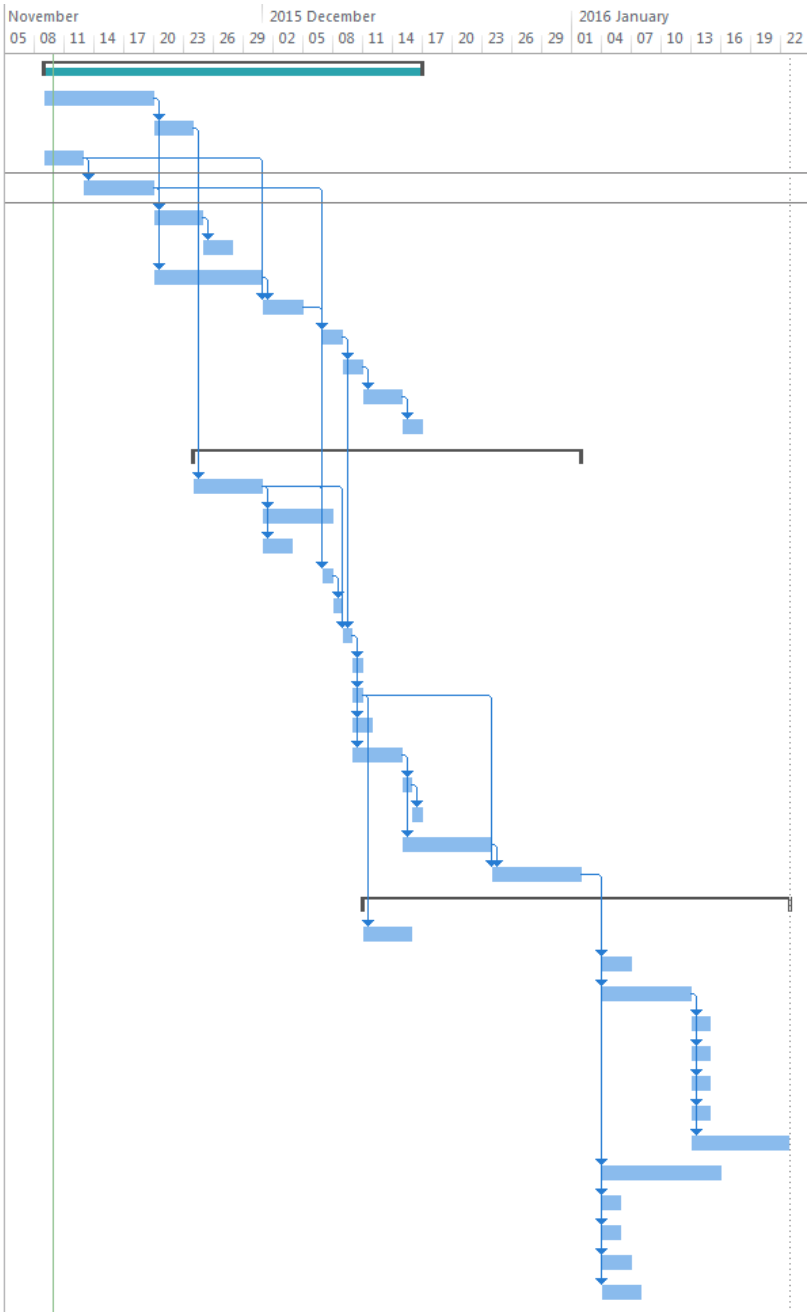
4 Preliminary Timetable

By the end of fall term, we plan to have a working proof-of-concept, using the Raspberry Pi and a basic control program to toggle lights on and off. By the end of winter term, we plan to have all required features at least partially implemented, at which our project enters the beta phase. As we finish up by ironing out bugs and perhaps completing stretch goals, we will release version 1.0 prior to the expo.

4.1 Gantt Chart

Gantt chart available at our project SharePoint website: `https://oregonstateuniversity-my.sharepoint.com/personal/rettigs_oregonstate_edu/capstone22_project/_layouts/15/start.aspx#/Lists/Tasks/gantt.aspx`

Task Name	Duration	Start	Finish	Predecessors
1. Device control	**28 days**	**Mon 15-11-09**	**Wed 15-12-16**	
(a) Server can boot with GUI and touchscreen support	9 days	Mon 15-11-09	Thu 15-11-19	
(b) Server starts control program on power on	2 days	Fri 15-11-20	Mon 15-11-23	2
(c) Clients are loaded with client control program	4 days	Mon 15-11-09	Thu 15-11-12	
(d) Clients discover connected relays/lights	5 days	Fri 15-11-13	Thu 15-11-19	4
(e) Clients can toggle lights individually	3 days	Fri 15-11-20	Tue 15-11-24	5
(f) Lights can be toggled over time	3 days	Wed 15-11-25	Fri 15-11-27	6
(g) Server acts as wireless access point for clients	7 days	Fri 15-11-20	Mon 15-11-30	2
(h) Clients can be paired with servers	4 days	Tue 15-12-01	Fri 15-12-04	8,4
(i) Clients tell server which lights are available	2 days	Mon 15-12-07	Tue 15-12-08	9,5
(j) Server can send light instructions to clients	2 days	Wed 15-12-09	Thu 15-12-10	10
(k) Clients can receive light instructions from server	2 days	Fri 15-12-11	Mon 15-12-14	11
(l) Clients can perform light instructions from server	2 days	Tue 15-12-15	Wed 15-12-16	12
2. User interface	**29 days**	**Tue 15-11-24**	**Fri 16-01-01**	
(a) Server serves web site with control interface for user	5 days	Tue 15-11-24	Mon 15-11-30	3
(b) Accessible via the system's built-in touchscreen	5 days	Tue 15-12-01	Mon 15-12-07	15
(c) Accessible to other devices like phones	3 days	Tue 15-12-01	Thu 15-12-03	15
(d) Displays list of connected clients	1 day	Mon 15-12-07	Mon 15-12-07	15,9
(e) Can nickname clients	1 day	Tue 15-12-08	Tue 15-12-08	18
(f) Displays list of connected lights	1 day	Wed 15-12-09	Wed 15-12-09	15,10
(g) Can nickname lights	1 day	Thu 15-12-10	Thu 15-12-10	20
(h) Buttons to toggle lights individually	1 day	Thu 15-12-10	Thu 15-12-10	20
(i) Sliders to change light intensity	2 days	Thu 15-12-10	Fri 15-12-11	20
(j) Can put lights into groups	3 days	Thu 15-12-10	Mon 15-12-14	20
(k) Displays list of groups	1 day	Tue 15-12-15	Tue 15-12-15	24
(l) Can nickname groups	1 day	Wed 15-12-16	Wed 15-12-16	25
(m) Groups can be nested	7 days	Tue 15-12-15	Wed 15-12-23	24
(n) Can toggle lights/groups based on rules	7 days	Thu 15-12-24	Fri 16-01-01	27,22
3. Rules	**31 days**	**Fri 15-12-11**	**Fri 16-01-22**	
(a) Lights can be toggled manually, overriding any rules	3 days	Fri 15-12-11	Tue 15-12-15	22
(b) Rules can be temporarily disabled	3 days	Mon 16-01-04	Wed 16-01-06	28
(c) Lights can be toggled based on time of day	7 days	Mon 16-01-04	Tue 16-01-12	28
(d) Lights can be toggled based on day of week	2 days	Wed 16-01-13	Thu 16-01-14	32
(e) Lights can be toggled based on day of month	2 days	Wed 16-01-13	Thu 16-01-14	32
(f) Lights can be toggled based on day of year	2 days	Wed 16-01-13	Thu 16-01-14	32
(g) Lights can be toggled based on specific dates	2 days	Wed 16-01-13	Thu 16-01-14	32
(h) Lights can be toggled based on sunrise/sunset times	8 days	Wed 16-01-13	Fri 16-01-22	32
(i) Multiple rules per light/group	10 days	Mon 16-01-04	Fri 16-01-15	28
(j) Lights can be toggled based on parent group	2 days	Mon 16-01-04	Tue 16-01-05	28
(k) Rules can toggle early/late by random time	2 days	Mon 16-01-04	Tue 16-01-05	28
(l) Lights can toggle early/late randomly within groups	3 days	Mon 16-01-04	Wed 16-01-06	28
(m) Lights can gradually toggle over time	4 days	Mon 16-01-04	Thu 16-01-07	28



5 Signatures

Malcolm Diller _____ Date _____
Sean Rettig _____ Date _____
Evan Steele _____ Date _____
Victor Hsu _____ Date _____