Twilight

Rauhul Varma, Naren Sivagnanadasan

1 Introduction

1.1 Objective

Smart lighting systems do not work how people ex[1]. Some light bulbs forget their state when the power is cycled (e.g. I turn the switch on or off). Almost all require an app on your phone to configure the actual "smart" functionality, which is a user interaction that is really unnatural to most people. More intelligent behaviors typically require even more apps on your phone. Installation is a long process of pairing with a hub of some sort or connecting to wifi networks. The system also does not know how to run itself either and requires the user to program it. And if there system needs to be moved or a bulb needs to be changed out, then the system has to be reprogrammed. It is clear that the interaction models currently implemented in smart lighting systems are not user friendly and do not enable functionality that would improve outcomes for users and that is mainly what we are looking to address and we are looking to do it in the place where people spend almost 50% of their waking time at a cost competitive to current solutions in the workplace.

We think that by using ideas inspired by biology in self organizing systems, we can create a reconfigurable lighting system for workplaces that is easy to control in an intuitive fashion, easy to maintain both from a technical development standpoint and a repairs standpoint and also be a good platform for people to experiment with controlling distributed peer to peer systems.

We think we can do all of this for significantly less than the current lighting systems being deployed in office buildings today. Lighting is one of the subtle factors that affect our mood and productivity and building this system will bring us closer to the grand vision of our environment modifying itself to maximize comfort for its inhabitants at any given time.

1.2 Background

The environment people live and work in has a deep impact on many things including mood, psychological health and productivity.[2] Simple changes like having the correct color temperature at different times of the day or or having the environment handle simple background tasks to reduce cognitive load may help people live happier and healthier lives.[3] However, the main blockers to having these intelligent environments widely deployed include cost and rigidity of the system (i.e. the system cannot be torn down and rebuilt easily or parts are hard to replace). Twilight aims to rectify some of these problems.

Typically there are a couple issues that can cause systems to be rigid and hard to use. Dependence on user interface flows that are not intuitive or do not leverage the habits people are used to (e.g. using a light switch to control the lights) means that whatever baked in smart functionality doesn’t get used.[1] These systems when not used in the design user interaction scheme often fall back to some sort of default functionality that is not much better than the standard light fixture (e.g. turns on as a white light by default). These systems are rigid as well because they typically have some sort of external relyance like a hub or cloud service in order to coordinate the system. The hub is a not so great solution as it is a bottleneck, typically there is are limit to how many nodes a hub can support (usually far lower than the number of bulbs needed to light a house). For cloud services there is a direct dependence on an internet connection to have a functional intelligent lighting system.

The solution we are interested in leverages the same ideas expressed in bee hives and ant colonies, where nodes of a system work together to accomplish higher goals all while self organizing. Using ideas in distributed systems to propagate instructions throughout the network, we can bring a new unit into a system without halting the rest of the system. The system is also robust to topology since it all that matters is connection to the network. This means no external controller is necessary to get the same behavior as a smart hub or cloud based lighting system.

1.3 High-Level Requirements

1.3.1 Requirement 1

Almost all existing smart lighting solutions require an app on your phone to configure the actual "smart" functionality, which is a user interaction that is really unnatural to most people. More intelligent behaviors typically require even more apps on your phone. Installation is a long process of pairing with a hub of some sort or connecting to wifi networks. The system also does not know how to run itself either and require the user to program it. Twilight will not require any significant external control for its operations and will make programming the system should be easy for application developers and not required by the end user.

1.3.2 Requirement 2

Installation and maintenance of Twilight should be trivial to the point where a student can put a fixture up with no help. As a result Twilight should be compatible with standard ceiling tiles and be robust to individual Twilight units failing. Twilight will detect and localize all units in the system and automatically reconfigure if connection to one is lost. Twilight units will run off standard AC wall power and additionally will daisy chain power to one another to reduce the need for cabling.

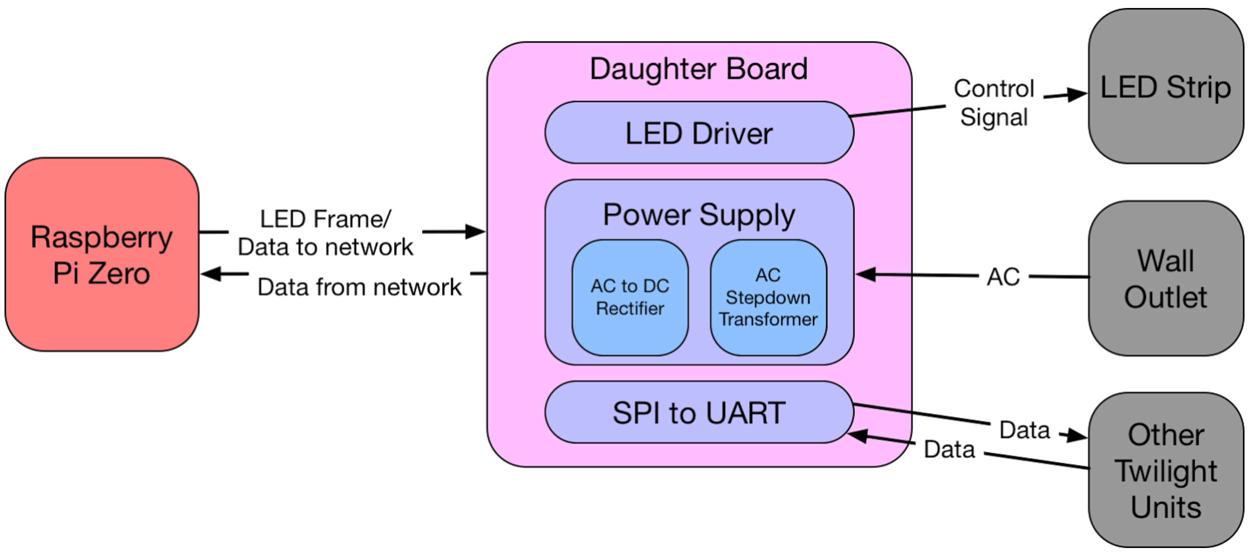
1.3.3 Requirement 3

Twilight is an education tool; ACM@UIUC is an organization that exists to help students explore the world of computing, mainly through experimentation and project building. This is one of many platforms ACM@UIUC is bringing up to provide the members opportunities to work with complex technologies in a tangible way. Therefore Twilight must be a system that lets students express their creativity on this platform with a simple user interface.

2 Design

2.1 Block Diagram

Figure 1.



The backbone of each light is a Raspberry Pi Zero responsible for communicating with the other lights and determining what ‘state’ the light should be in. Though the self organizing functionality of the project can be accomplished on a microcontroller, we choose the Pi because it provides a full networking stack and provides developers using the system a comfortable environment to develop applications. The Raspberry Pi includes a full networking stack built, which is significant hardware component that we don't have the time to design/implement in a single semester. Additionally, we also want this to be a programmable easy-to-access platform that should last past our graduation date, using a Pi means that we have access to any number programming languages making writing applications easier for developers.

We will be designing a custom daughter board for the Pi; it will include an LED driver to control an RGB LED strip within each Twilight unit. We decided to use an RGB strip because they allow for more complex and dynamic animations and smart effects than a purely warm LED strip (i.e. can change color temperature between a typical range of 2000K and 6300K). The daughter board will also include a power supply for the Pi and LEDs to run off of standard AC power using an AC step down transformer, a full bridge rectifier, and a buck transformer. Lastly, the daughter board will include an SPI to 4-port UART breakout to allow for inter-Twilight unit communication. We initially planned on using ethernet between each unit however the decided against it due to the technical complexity of creating/housing the corresponding required networking switch.

2.2 Physical Design

Figure 2.

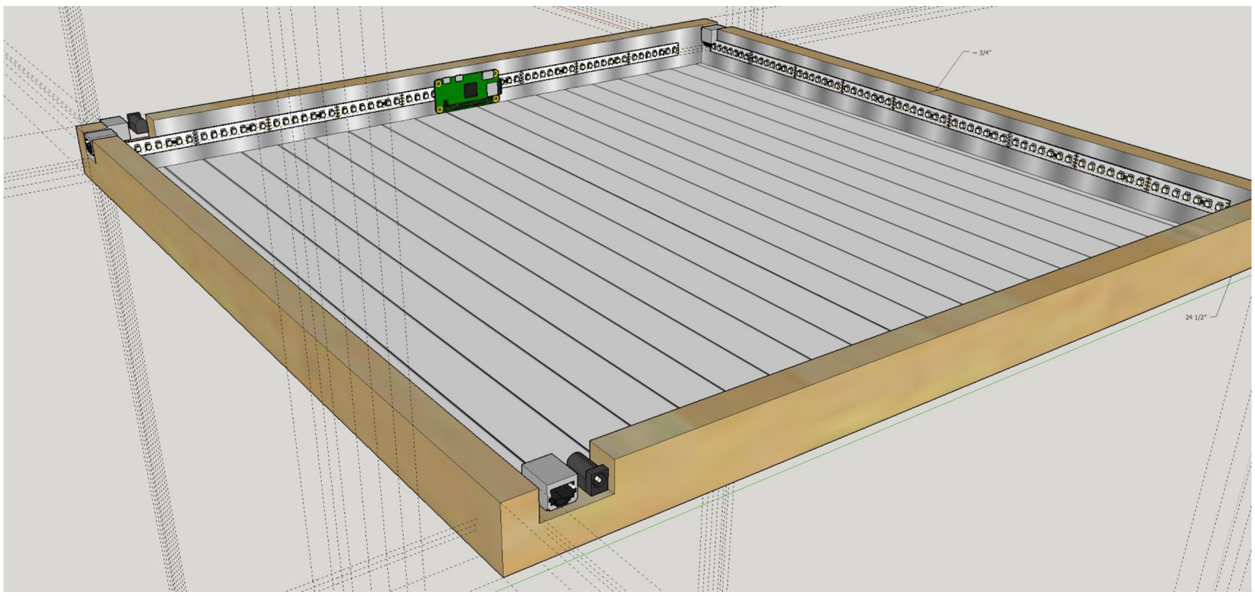
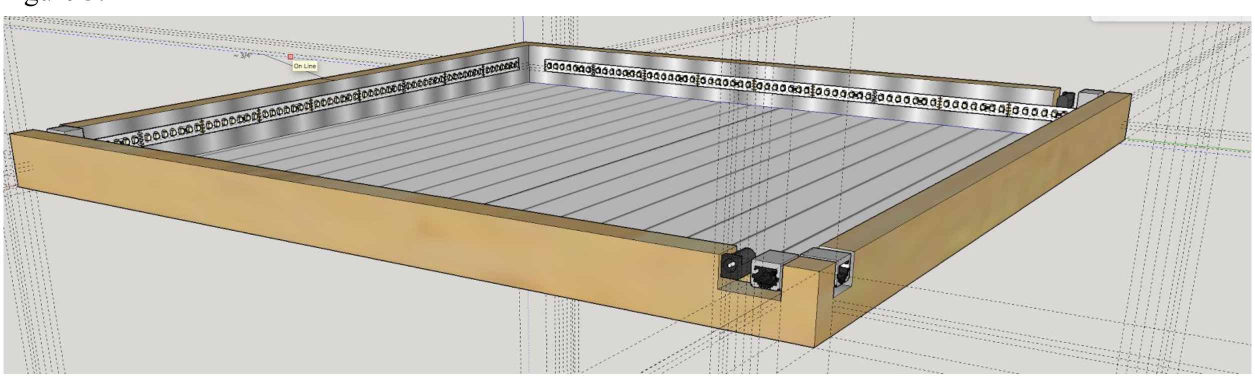


Figure 3.



.

The physical design of the system is as follows; the entire system consists of a collection of units (we are specing to make 15). Each unit is a magnetically suspended wooden box, the same dimensions as a standard ceiling tile (so it is square with external width 24.5” and internal width 23”). On the inside of the box we coat the sides with aluminum tape to reflect the LEDs. We use 92” of LEDs (roughly 140 LEDs). We defuse the light using a canvas layer mounted to the inside of the frame. We then mount a Raspberry

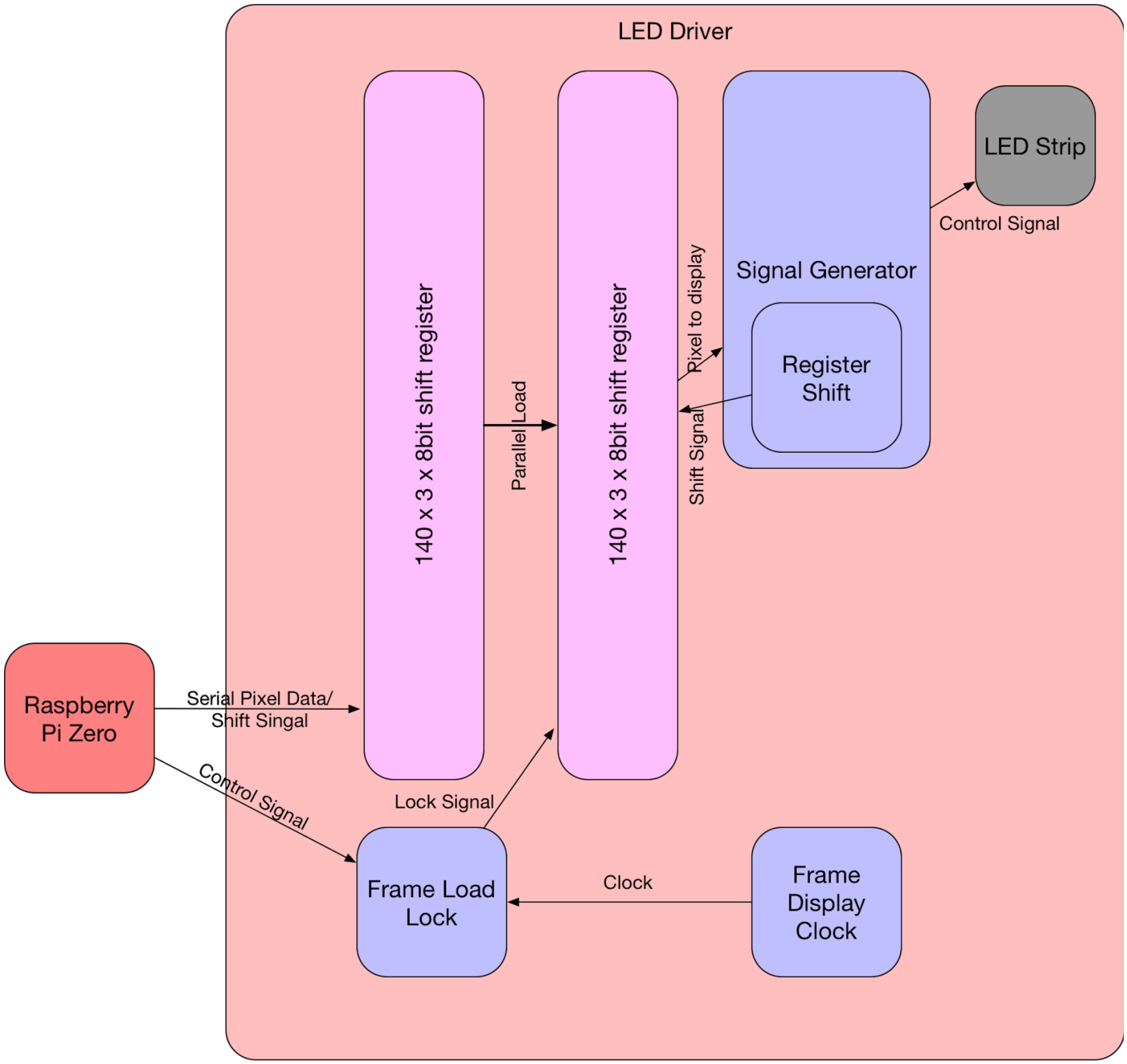
Pi Zero with a custom Pi hat with our communications, control and power systems on board. On two opposing sides we place a barrel jack to provide and distribute power. And on all 3 sides there will be a RJ45 jack to carry the serial communication between units.

2.3 Functional Overview

Each block in the diagram is essentially independent of all other blocks other than the fact that they live on the same pcb. The only interdependence is between the Power Supply and the other two as the Power Supply is how the other blocks are able to function.

The LED Driver is responsible for taking frames (a description of the color of each LED in a Twilight unit) from the Raspberry Pi and displaying them. It works with a via a Ping-Pong of shift register loading. The driver first loads a frame into a storage buffer. The contents of this buffer are then loaded into another shift register whose values are shifted out into block that actually generates the signal required for the LED Strip the change to the desired color. It uses a couple clock signals to manage the timing of each operation in order to meet the specifications laid out below. The proposed design can be found in a Figure 2.

Figure 4.



The Power block has two major components, a step down in voltage 120VAC to 12VAC and then the conversion from AC to DC 5v to power the Pi and the LEDs. The step down will be performed with a standard iron core transformer. The rectification will occur in a two step process, we will first use a full bridge rectifier to convert the AC wave to a DC wave followed by a capacitor to smooth this output. The smoothed rectified output will then be fed into a buck transformer, further reducing the voltage down to 5V which will also be smoothed by an accompanying capacitor. We are using 2m of LEDs with 60 LEDs / m which draws 3A. The Raspberry Pi Zero draws 120mA.

The SPI to UART block manages the inter unit communication. Raspberry Pi Zero comes with a single dedicated UART handler exposed on GPIO. But each block could have to support up to 4. We therefore use SPI as a way to multiplex a single connection to a PI to the four serial connections it needs to support.

The design we are using for this is based off of a PIC24FJ64GA306, which implements are protocol translating each incoming/outgoing UART connection into a SPI slave.

Figure 5.

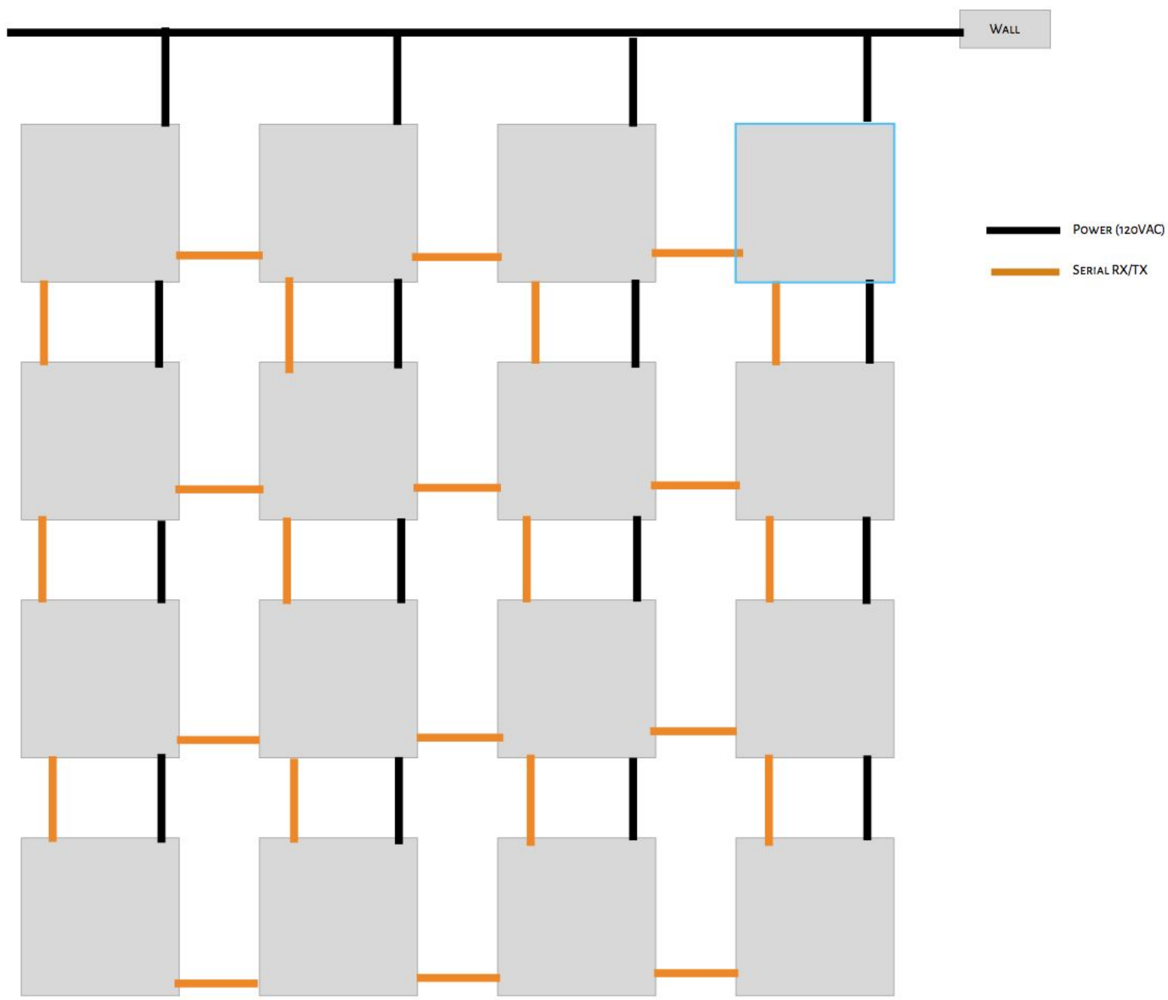


Figure 5 is the proposed default topology. Each block is connected to its neighbors via a serial connection and power is distributed via 120VAC rails extending from the wall, into each block and out the other side, which powers the Pi and LEDs. While other configurations of the system will be supported this will be the default implementation as it most closely matches the space we are going to deploy the system in.

On system initialization (i.e. first installation) the system will begin with a localization procedure where starting from the blocks with only 2 connections, a localization message will propagate through the system assigning each block a location. Then the system will enter the application space and begin to execute its current application (in this case the color temperature application). Periodically clocks will be synced using Network Time Protocol (NTP) and there will be a nodes in the system that will act as a border routers, providing a network connection, so updates can be applied.

2.4 Block Requirements

2.4.1 LED Driver

The LED Driver must be able to display 30 frames on each Twilight unit each second. Additionally, it must not cause flickering or other undefined behavior on startup and shutdown; strange behavior like this must not occur as it is a significant detriment to a user's experience of the system. Additionally, the LED driver must be able to load a new frame to be displayed from the Raspberry Pi under 20ms. Moreover if a new frame is not available from the Raspberry Pi the LED driver must display the old frame and not display a black frame.

2.4.2 Power Supply

The Power Supply must be able to take 120V AC and convert it to 5V DC. Additionally the power supply must be able to provide 4A continuous current. In the event more than 20 watts are being drawn the power supply the unit should turn off. The circuit must be relatively light weight as each Twilight unit will be attached to the ceiling.

2.4.3 SPI to UART

The SPI to UART block must breakout the SPI connection on the Raspberry Pi to allow each Twilight unit to communicate with up to 4 other units. It also must allow for packets between each of the 4 legs to be distinguished so that Twilight can locate each unit without user input. It must also be able to handle connections being added/removed dynamically while the system is on, to allow for simple reconfiguration. The connection will be at 9600 baud and since the connection is multiplexed, there needs a handshake before a message is sent so it does not get clobbered by other nodes.

2.4.4 Risk Analysis

The Power Supply block appears to be the most difficult to implement and biggest risk to a successful completion of Twilight. The main reason for this is an overall lack of strong knowledge with power systems (beyond that in ECE 330) between the group members. Moreover, on top of the non-trivial electrical requirements, this block has significant weight restrictions that make the design more difficult. This being said, we found a lot of instructional material and guides on the subject that have help restore confidence in completing this block.

2.5 Ethics and Safety

The biggest safety issue we face is working with wall power since it is 120V AC. Since every unit in the system will be working off wall AC power, verification of the power supply for the Pi is crucial. We also have to deal with concerns about issues like epilepsy when dealing with fast animations. We can address this by limiting the frequency of the LEDs, or by reviewing software deployed on the system. There are potential environmental concerns that may arise when sourcing LEDs as some have been found to contain lead, arsenic and other dangerous substances. Proper care in sourcing RoHS compliant components must be taken.

We also explicitly layout in this proposal the potential for this platform to be used to improve people's mood through the use of color temperature modulation as an example. But just as you can use this platform to positively affect people you could potentially use the system to decrease people's quality of life. People have complained about short wave heavy white LEDs used in streetlights preventing people from sleeping for instance. Extended viewing of LEDs directly could also cause retina damage. The system also could potentially contribute to the growing issue of light pollution (though this is mostly an issue with outdoor lighting).

2.6 Citations

[1]C. Rowland, "What’s different about user experience design for the Internet of Things?", *O'Reilly Media*, 2017. [Online]. Available: https://www.oreilly.com/learning/whats-different-about-user-experience-design-for-the-internet-of-things. [Accessed: 03- Oct- 2017]

[2]W. van Bommel and G. van den Beld, "Lighting for work: a review of visual and biological effects", *Lighting Research & Technology*, vol. 36, no. 4, pp. 255-266, 2004.

[3]I. Knez, "Effects of indoor lighting on mood and cognition", *Journal of Environmental Psychology*, vol. 15, no. 1, pp. 39-51, 1995.

[4]H. Ishii, H. Kanagawa, Y. Shimamura, K. Uchiyama, K. Miyagi, F. Obayashi and H. Shimoda, "Intellectual productivity under task ambient lighting", *Lighting Research and Technology*, 2016.

[5]R. Küller, S. Ballal, T. Laike, B. Mikellides and G. Tonello, "The impact of light and colour on psychological mood: a cross-cultural study of indoor work environments", *Ergonomics*, vol. 49, no. 14, pp. 1496-1507, 2006.

[6]J. Kim, J. Ko and M. Cho, "A Study of Integrated Evaluation of System Lighting and User Centered Guideline Development - Focused on the Lighting Design Method for Office Space -", *Korean Institute of Interior Design Journal*, vol. 23, no. 6, pp. 78-86, 2014.

[7]D. Park, Y. Lee, M. Yun, S. Song, I. Rhiu, S. Kwon and Y. An, "User centered gesture development for smart lighting", *HCI Korea 2016*, 2016.

[8]F. Tan, "User-in-the-loop smart lighting control system", Masters, Delft University of Technology, 2016.

[9]D. Burmeister, A. Schrader and B. Altakrouri, "Reflective Interaction Capabilities by Use of Ambient Manuals for an Ambient Light-Control", *HCI International 2016 – Posters' Extended Abstracts*, pp. 409-415, 2016.

[10]A. Lucero, J. Mason, A. Wiethoff, B. Meerbeek, H. Pihlajaniemi and D. Aliakseyeu, "Rethinking our interactions with light", *interactions*, vol. 23, no. 6, pp. 54-59, 2016.