**Boston University**

**Electrical & Computer Engineering**

**EC 463 Senior Design Project**

First Semester Report



Submitted to

Thomas Little

Boston University

8 Saint Mary’s Street

Boston, MA

(617) 353-9877

tdcl@bu.edu

by

Team 1

Team Luminesense

Team Members

Sneha Pradhan [snehha@bu.edu](mailto:email1@bu.edu)

Zehua Zhao [zehuaz@bu.edu](mailto:email1@bu.edu)

Tanatsigwa Hungwe [hungweta@bu.edu](mailto:email1@bu.edu)

Caroline Jones [ctjones@bu.edu](mailto:email1@bu.edu)

Michael Haley [mhaley10@bu.edu](mailto:email3@bu.edu)

Submitted: 12/12/16

#### Table of Contents

Executive Summary 2

1.0 Introduction 3

2.0 Concept Development 5

3.0 System Description 8

4.0 First Semester Progress 10

5.0 Technical Plan 14

6.0 Budget Estimate 17

7.0 Attachments 19

7.1 Appendix 1 – Engineering Requirements 19

7.2 Appendix 2 – Gantt Chart 20

7.3 Appendix 3 – Other Appendices 21

# 

# 

# Executive Summary

Adaptive Gesture Based Lighting System

1 – Luminesense

Lighting systems have failed to take advantage of the wealth of opportunities presented by modern technology - to integrate into a world of intelligent, interconnected devices. Consumers need a lighting system that is more dynamic, intuitive and capable of conserving energy that is currently wasted.

Our final deliverable will be a system that consists of a wearable transceiver that commands the Luminesense system through an array of predefined gestures, and an “adaptive mode” which will automatically detect the presence of people in a room. Accompanying this wearable is control software existing in an IP enabled central hub that recognizes gestures sent from the wearable, analyzes information sent from various sensors, and appropriately updates the state of up to 16 lighting fixtures in a room. The performance and energy data of our system will be displayed on a web application to be provided.

Luminesense’s approach to solving this problem involves separating the modes of engagement into 2 distinct parts: “gesture” and “adaptive” mode. The former allows the user to explicitly interact with the system by invoking gestures through the wearable; the latter automatically controls the lights when the wearable is disconnected or a user leaves or events the room.

The Luminesense system features a host of capabilities: an accurate gesture recognition system sent through a comfortable, wearable device, a machine learning algorithm that empowers users to define their own gestures; an intuitive gesture library, and a state of the art motion capture system that allows us to adjust lights based on user presence in real time. The project is not exceptional by virtue of its objective, but by the execution of the objective: to create a coalesced system of communication amongst the various components.

# Introduction

The problem our client faces is the need for a dynamic, convenient lighting system that can intuitively respond to user preferences and commands using a wearable and that can be fully integrated with their existing Luminaire technology. Additionally, there is a need to improve the energy efficiency in current lighting systems. The proposed solution is to automatically adapt to energy-saving protocols based on the presence of occupants in the room.

As the “Internet of Things” phenomenon emerges and an increasing number of everyday objects have network connectivity, lighting systems continue to lag behind, failing to become “smarter”. The standard light switch has existed for over 100 years and is still highly limited in its functional capacity (typically just “off” and “on”). This is clearly a market that could reap great benefits by adapting IoT technology.

From an environmental perspective, humans are depleting electricity, a critical resource, at an alarming rate. By carelessly leaving the lights in a room and leaving, the average consumer produces an additional 0.15 pounds of greenhouse gas emissions per hour. These greenhouse gases trap heat in our atmosphere and create climate change that has the potential to compromise our way of life. This type of reckless behavior wastes money, energy, and has a negative impact on the environment.

Our project will provide the customer with a wearable with which a user can interact seamlessly with the customer’s existing Luminaire technology. The wearable will enable the user to use gestures to interact with the lighting, which provides convenience and more precise control over lighting than any other system. The project will also switch between gesture and adaptive modes automatically in order to save energy. For example, when a user leaves the room without gesturing to turn the light off, the adaptive mode will take over and switch the lights off in order to prevent energy waste.

The team’s focus is to provide convenience for the user, both by providing an innovative and customizable solution for controlling lights. The system must, in general, be intuitive and easy to interact with. It must offer features to provide the best possible user experience, while also meeting client expectations regarding the integration of their technology into the system and the overall innovation of the system.

The objective of this project is to develop an intelligent lighting system capable of dynamic, convenient, eco-friendly control over light fixtures in a room. This will involve a gesture-based mode, an adaptive mode, and a wearable device to support user interaction (gestures). The gesture mode will give users explicit control over lights through the wearable and command library. Consequently, adaptive mode will passively control the lights. Providing these two modes of interaction guarantees user energy savings and a more dynamic, user-friendly interaction between humans and light fixtures.

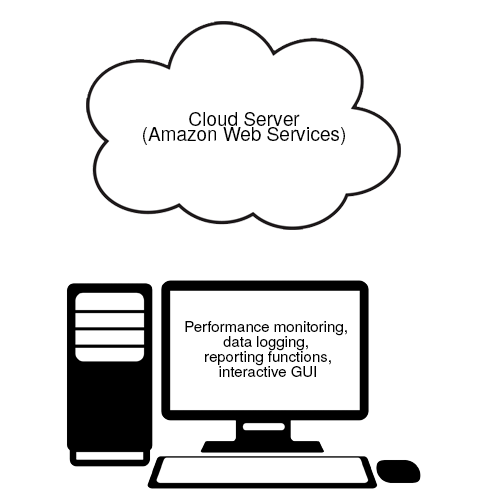
Special Features:

1. Wearable transceiver with Bluetooth capability (Figure 1.)

2. Comprehensive gesture library to support user control over lighting fixtures with accompanying gesture dictionary that explains all possible gestures

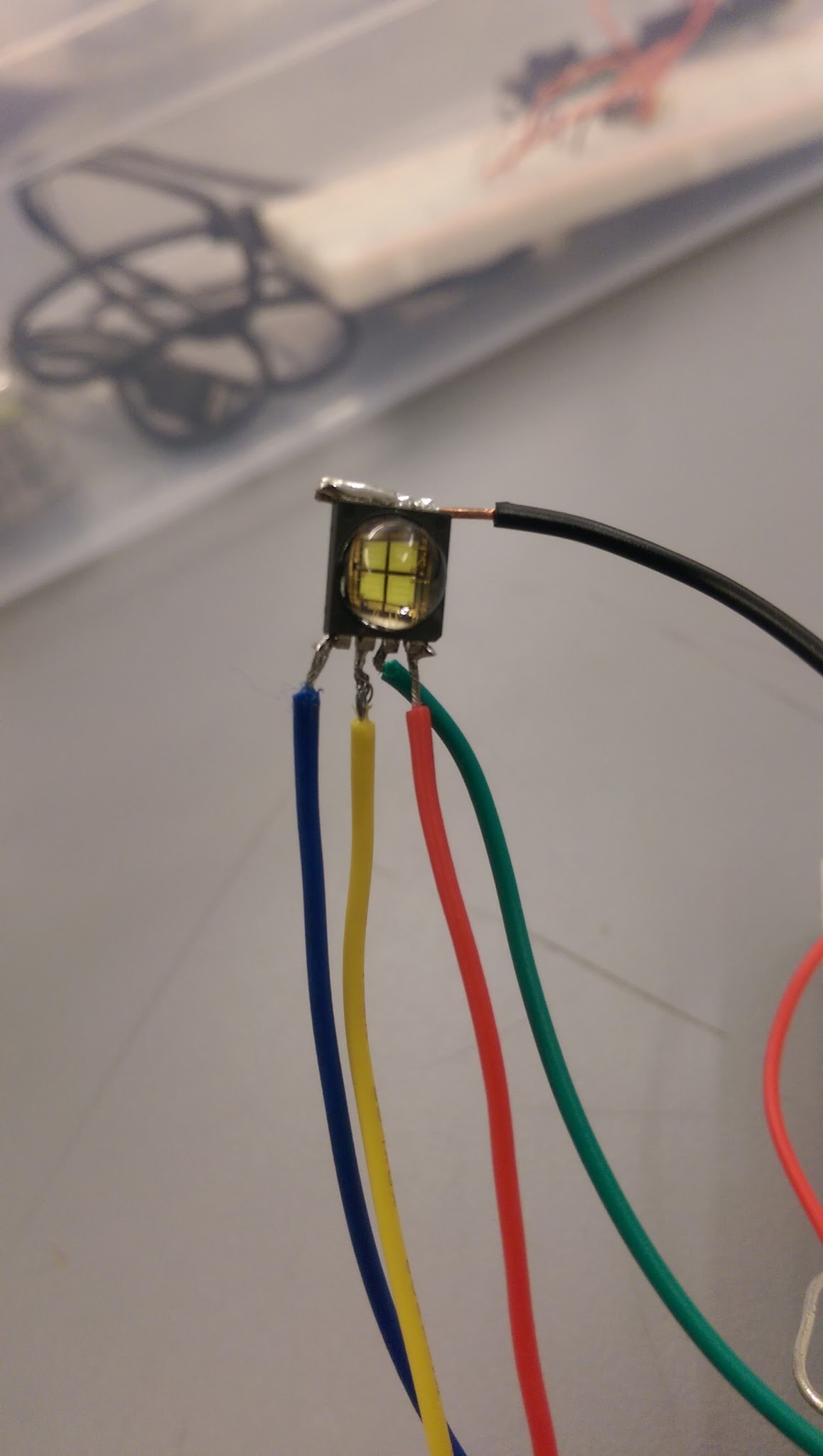
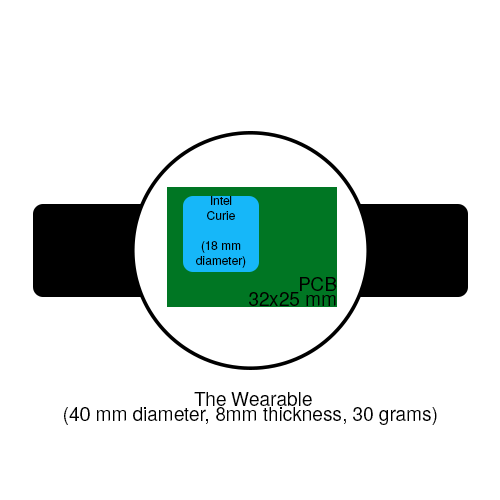
3 Web application monitoring performance data and energy savings (Figure 2.)

4. Control software designed to accomplish three tasks:

* Recognize the location of the user(s) utilizing the wearable device.
* Analyze the information being sent amongst the Raspberry Pi, the Luminaires and the user i.e the gesture and the light codes.
* Control the states of each individual Luminaire within the system.

5. Occupancy detection driving adaptive mode control

6. Multiple user support with conflict avoidance







# Concept Development

The proposed product is an intelligent gesture-based system which provides new ways for humans to interact with lighting. This product solves the problem of inefficient and archaic interactions with lighting systems. In today’s world, everything is becoming “smart” i.e. can facilitate more avenues of communication, thanks to the internet. The concept of the smart home and the Internet of Things has created a snowball effect of highly interesting and innovative projects. Unfortunately, there has not been any notable advancements in the realm of lighting systems.

In the average household, the same lighting system has been installed for decades. Companies such as Lutron and Ubiquilux are attempting to innovate lighting products. Unfortunately, the solutions they provide are too expensive and unattainable to much of the market. Additionally, the systems are not as convenient nor intuitive as desired. Within many sectors of life - homes, hospitals, labs and schools - the shortcomings of light switches have become a contentious issue regarding power, energy, monetary, and even health concerns. Currently, there is no comprehensive system that assists the user in confronting and solving these issues.

Luminesense seeks to solve the above problem via two ways: automatic/adaptive mode and explicit/gesture mode. The two modes will work on the same system and will cater to different scenarios. The adaptive mode is an automatic mode where the sensors detect the presence of occupants in the room. If none are present, the lights will automatically shut off. When an individual enters the room, the lights turn on to either a preset user preference or a default setting. Keeping lights off when nobody is present helps solve the customer’s problem of energy efficiency. The second mode, gesture mode, is a more active way to control the lights. The user will utilize a wearable that to control the lights through a variety of gestures that the user can design and implement. This is the solution to the customer’s problem of inconvenient interactions with lights. The first step to solving these large problem is by scaling back and starting small.

The adaptive mode will be supported by PIRs (Passive Infrared Sensors) as they are capable of detecting the presence of users (when indoors), and the customer already has a system for presence detection established. This technology will fill the client’s requirement that the system be compatible with research he has already produced and the lighting technology that he has already developed. Also, Infrared sensors are favorable because they are cheap and easy to implement.

The gesture mode will be extremely interactive and a vital asset to the system. This mode allows the user to interact with the system with a set of gestures defined in a “gesture library” . These gestures will be saved in the cloud where they can be edited or customized by users. This will allow users to define their gestures however they choose. These library commands will either be stored on the wearable itself or on a cloud server. Gesture mode is supported by the wearable device containing the Intel Curie Microcontroller, as well as the Photodiode and a battery to power the wearable. The Intel Curie chip has built in Bluetooth Low Energy as well as a 6-axis sensor that will detect the user’s gesture. The customer requires that gestures can cover a wide range of commands, therefore the sensing tools for the gestures must be able to differentiate between a wide variety of commands. This was the reason that the Intel Curie was chosen. The 6-axis sensors, gyroscope and accelerometer will enable precise differentiation between gestures. Also, this satisfies the customer’s request for accurate and fast gesture interpretation, which plays a large role in user convenience and the usability of the system.

The Luminesense system will be based on the cloud. Therefore, the users can perform actions such as adding their gestures, modifying them and deleting them using a web application. This web application will also allow the resident to add or delete users in their home lighting system, which allows the system administrator to have precise control over who interacts with the system. This is a security measure so that the administrator can limit the system to specific users. This satisfies the customer’s requirement of creating a secure system: one that prevents the wearable from being used for malicious activity within another system.

Moreover, the user’s preferences will be stored in a cloud database which can be accessed anytime and from anywhere with an internet connection. The cloud will also enable the user to track their energy savings resulting from the use of the system. The concept of tracking energy efficiency is a major point of interest to the client, who pitched the idea primarily for energy efficiency purposes. Showing the user the positive impact they are having on the planet validates the societal impact that users are having by choosing the Luminesense system.

An alternate solution that was considered contained a Particle Photon in the central base which was connected to the Raspberry Pi. It was thought that the communication between the Raspberry Pi and the Particle Photon would be difficult to accomplish, especially compared to Photon to Photon communication, so in this set-up the Raspberry Pi and Photon had a “master-slave” relationship, wherein the Raspberry Pi would push commands to the Photon, and the Photon would then send this command to the Photon on the specified Luminaire. This solution was abandoned because it ended up being feasible to implement Raspberry Pi to Photon communication over Wi-Fi using the Particle Photon Javascript API.

Another consideration involved using a pre-existing wearable device to communicate within the system. Tools such as Texas Instrument's WearIT and the MetaWatch were researched upon and considered. These products provided many of the desired communication modes required (Bluetooth, and Wi-Fi capabilities). However, they lacked the relevant embedded hardware required by the system - optical sensors, optical laser and photodiode. Additionally, the wearables were far too expensive (1 MetaWatch costs $199.00), making the technology less accessible to key target users.

This concept currently being implemented was favoured because of the need for a lighting system that provides information and awareness to the user about the energy he or she is consuming. With the help of the cloud, the user will always be able to check how much energy is being consumed as well as the amount of money they are saving using our system. Luminesense will provide a sleek interface to the users to control their lights. This system will make saving energy very convenient and users will never have to worry about their lighting anymore.

# 

# System Description

As mentioned earlier, solution proposed will be accomplished through 2 modes: the first one is adaptive mode, the second is gesture mode. The wearable, which is modeled after a smartwatch, is to be worn by the user and will be connected to the central base, the Raspberry Pi 3 and the router, via Bluetooth Low Energy. Once connected, the screen on the watch will display “Connected”.

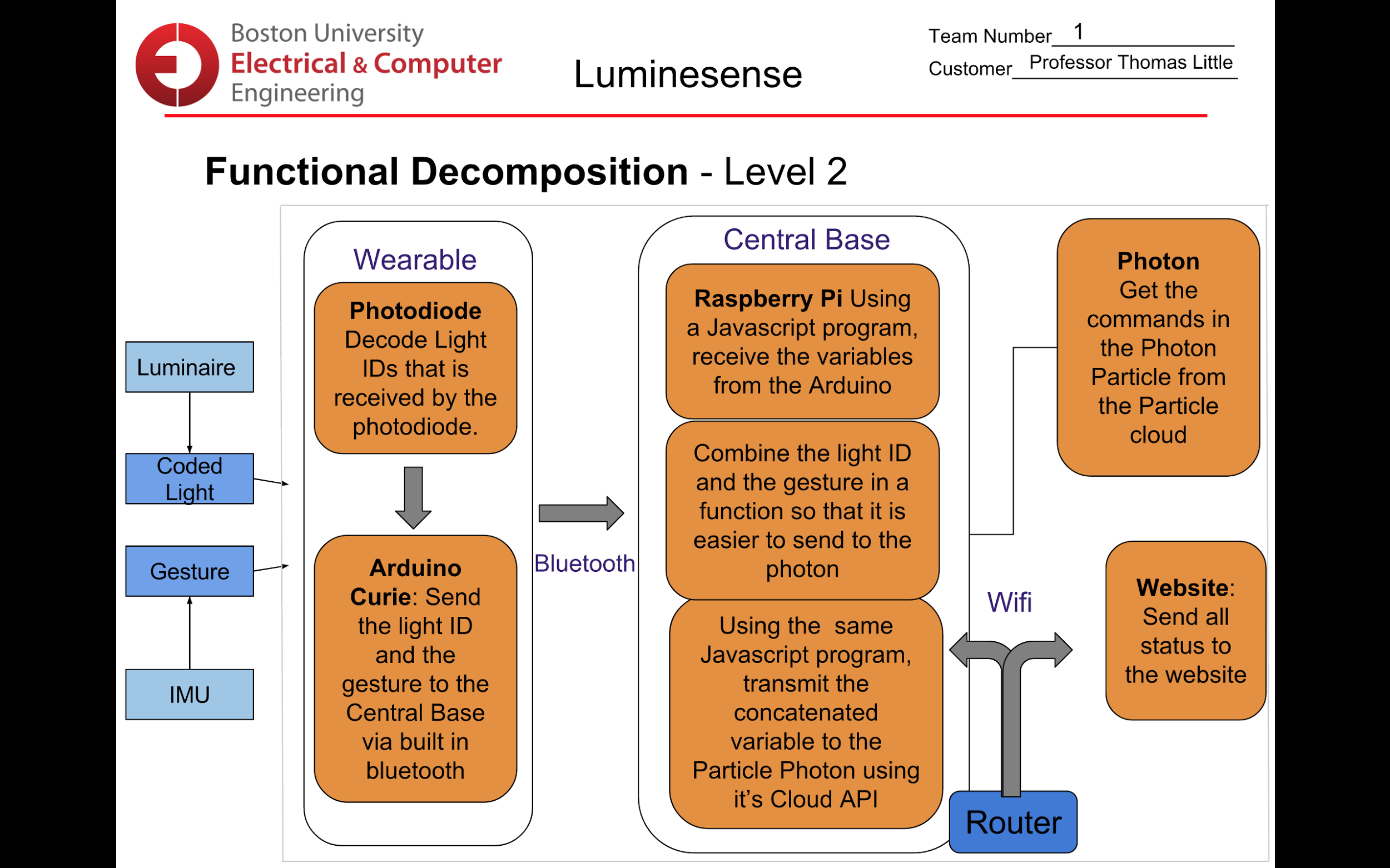
When the user enters the room with the wearable, the wearable will send its ID to the Raspberry Pi. This ID is then recognized by the cloud server, where the user stores all of their settings. Upon user recognition, the room will alter its lights to proper intensity and color according to the user’s settings that were stored in the cloud. If another user enters the room, the Raspberry Pi will receive the ID of the new user. Each Raspberry Pi will have an administrator ID that has administrative rights over the system. That is the ID number of the “owner” of the room who has the “authoritative” control privileges to the room. When the “owner” is connected to the central base, that user’s commands are given priority over any other user when interacting with the system. If there is no “owner” user currently in the room, then central base will prioritize the user who entered the room first.

Users may have different default settings according to different times of the day. For example, one can set the default setting for Luminaires to be fully operational during the evening time while having the midnight light intensity to be a quarter strong as compared to the evening. To change the preference settings, users can go to the web application, which is supported by the cloud server, and modify as needed. For users don’t have the wearable, the lighting system will turn on when they get observed by the single pixel camera installed in the area. When they leave the room, the Luminaires will shut off in a similar fashion. By sampling the time that the Luminaires have been off, the web application will be able to calculate the energy saved during that time.

In the gesture control mode, the user may modify either an individual or a group of Luminaires as needed. The user will be be provided a wearable which will capture one’s gestures to command the Luminaires. The little LCD screen on the wearable will show time during the day and show messages accordingly, such as when the battery is low or when the user is updating their gesture preferences. To control a single Luminaire, the user should press a button on the wearable in order to indicates the user’s intent to begin interacting with the system. Then, the user points the wearable to the desired Luminaire in order to get optical code with the photodiode built into the wearable. When the optical code is received, the user will then perform a gesture command to the Luminaire. The users will be provided with a default gesture library manual which includes preset gestures to turn Luminaires on/off, brighter/dimmer, and color changing. In order to let users be able to create their own gesture library, we will develop a machine learning algorithm to teach the wearable to remember new gestures to command Luminaires. The user can do this by interacting with the web app, and training the machine learning algorithm by performing their gesture 10 times, which are signaled by a button press at the beginning of the gesture and a button press at the end of the gesture.

When both the Luminaire’s ID and the gesture command are collected, the watch will send both sets of data to the central base via a Bluetooth connection that has been established the moment the user stepped into the room. The Raspberry Pi acts as the central base, collecting and aggregating the data received. The data is then parsed and forwarded as a command to the Particle Photon. Upon receiving this command, the Photon on the appropriate Luminaire will alter the its state accordingly. If the user wishes to command Luminaires within the entire room or in a certain range, he/she can get the optical code of the entire section of Luminaires and then perform gestures to command those Luminaires.

Shown below is the block diagram of the gesture-control mode. The inputs to the system are the user’s gesture, driven by an IMU on the wearable, and the coded light, driven by the customer provided Luminaire. These inputs go to the wearable, which determines the light ID and sends its gesture information over Bluetooth Low Energy to the Raspberry Pi. The Raspberry Pi determines the gesture with the machine learning library, and sends this and the light ID to the Photon over Wi-Fi, which changes the state of the selected Luminaire based on this information.



*Figure 5. Block diagram*

# 

# First Semester Progress

The work done this semester establishes a sound foundation which the team can continue to build in order to deliver an innovative design that exceeds the client’s expectations. The work done this semester focused on three key areas: definition of the problem and understanding the client’s needs, research into various means to solve this problem in order to define a preliminary design, and the construction and test of the first deliverable to show a proof of concept of key elements of our design.

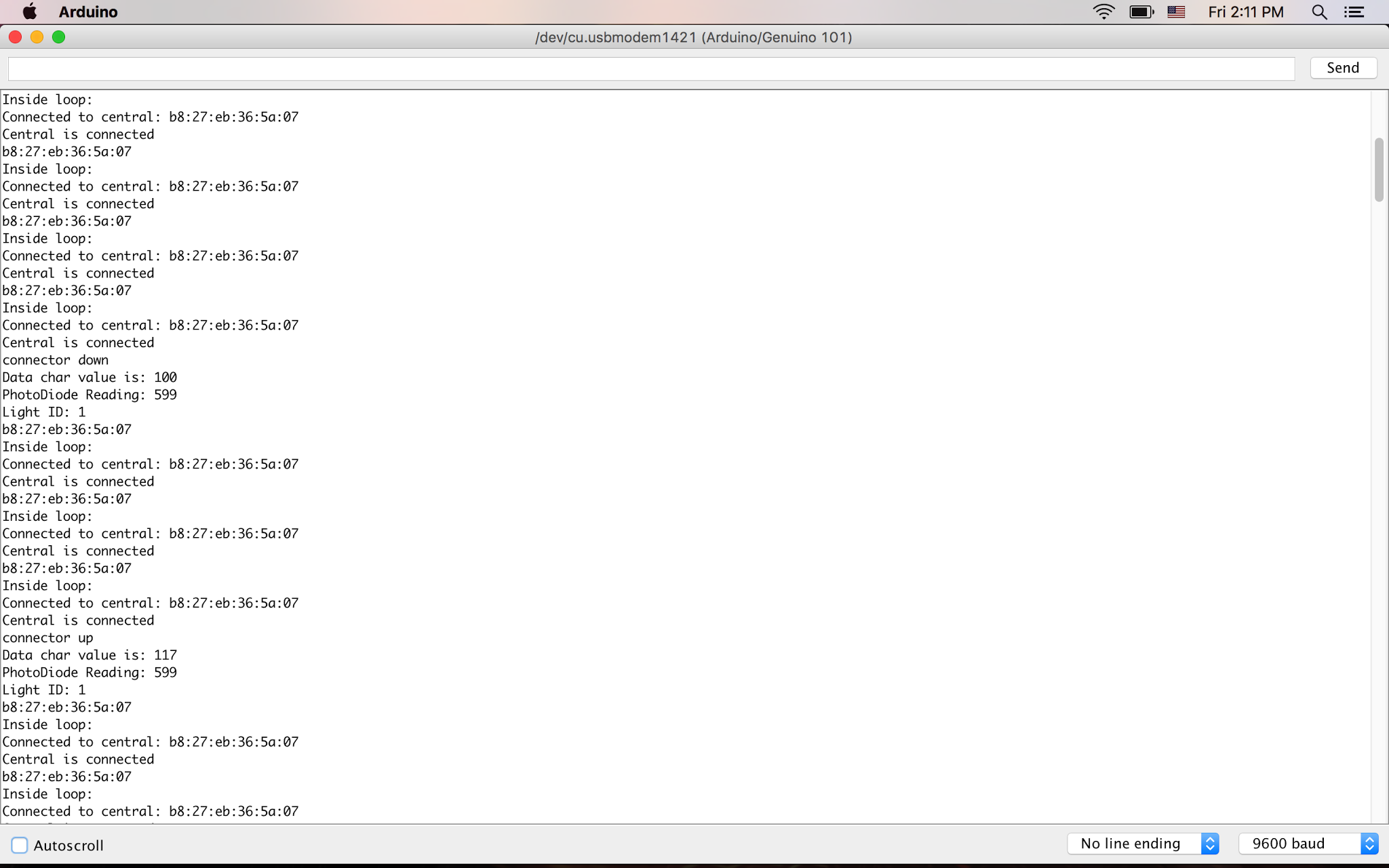
An important component of our project is the communication loop between all of the elements of the system. Without this loop, our project simply cannot function. Early on, the client stressed the importance of getting all of the components of the system working together first, as opposed to polishing individual components first and then attempting to have them interact later. Therefore, our first goal was to prove that the three main components of our project - the wearable, the central base, and the Luminaire - were able to send and act upon meaningful data in this communication loop.

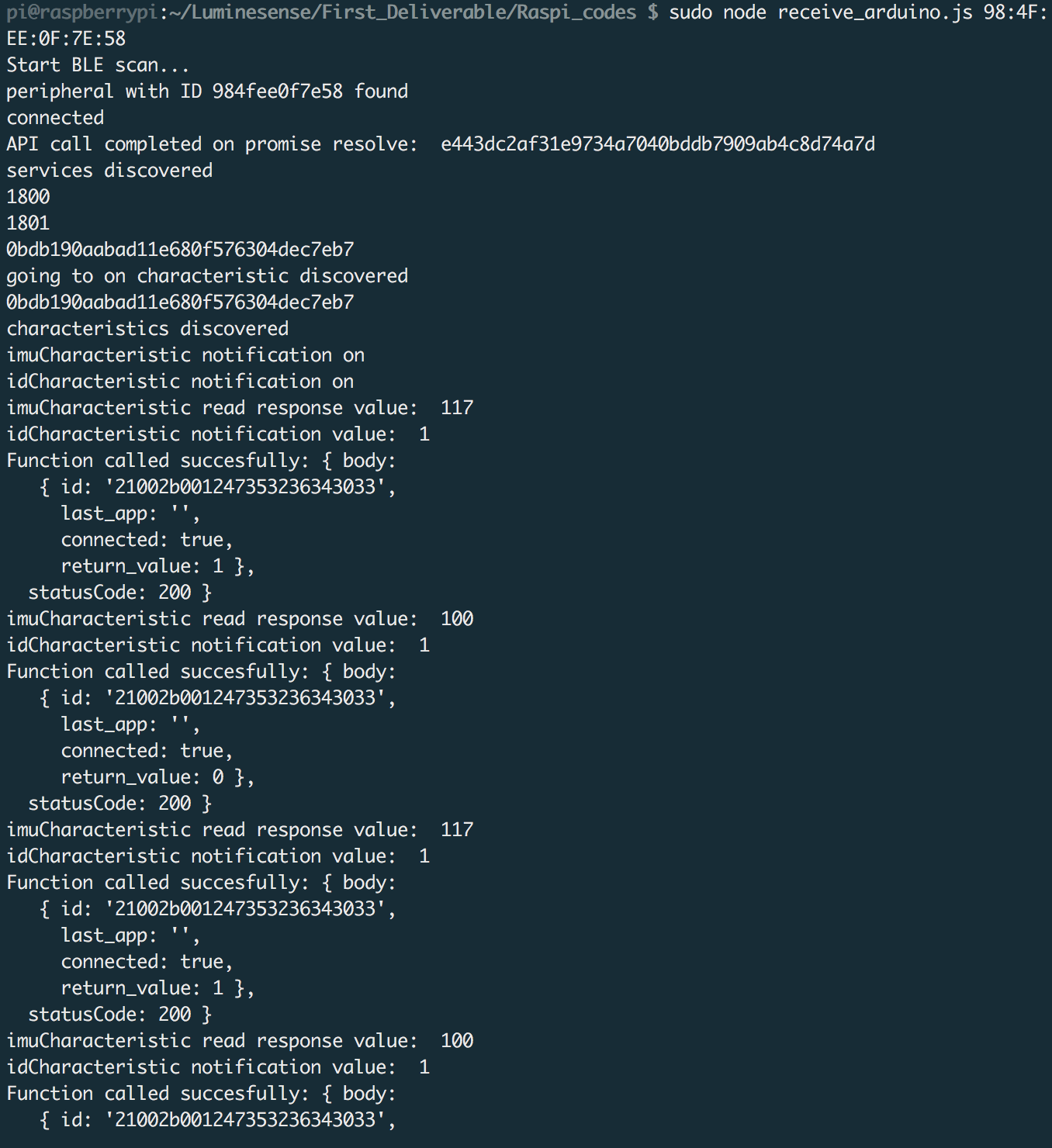
The wearable in this test is modeled by the Arduino 101 and a photodiode circuit. The Arduino 101 has the Intel Curie chip on board and has a 6-axis accelerometer and gyroscope, as well as Bluetooth Low Energy. The board and chip were designed for wearable and internet of things technology, which is why they are a good fit for modeling the wearable in our project. These sensors will be used to detect gestures that the users perform in order to command the Luminaires. For our first deliverable test, we decided to define two simple motions on the Arduino 101 to model an “On” command and an “Off” command for the light on the Luminaire.

The initial test was primarily to show a proof of concept of the communication loop and not the robustness of our gesture commands. The defined gestures for the commands are not realistic for the final product, and will obviously be improved upon for future tests. Currently, the open source Madgwick’s algorithm is used in order to determine the orientation of the Arduino board. Orientations described as “connector up” and “connector down” were used to model the “On” and “Off” command.

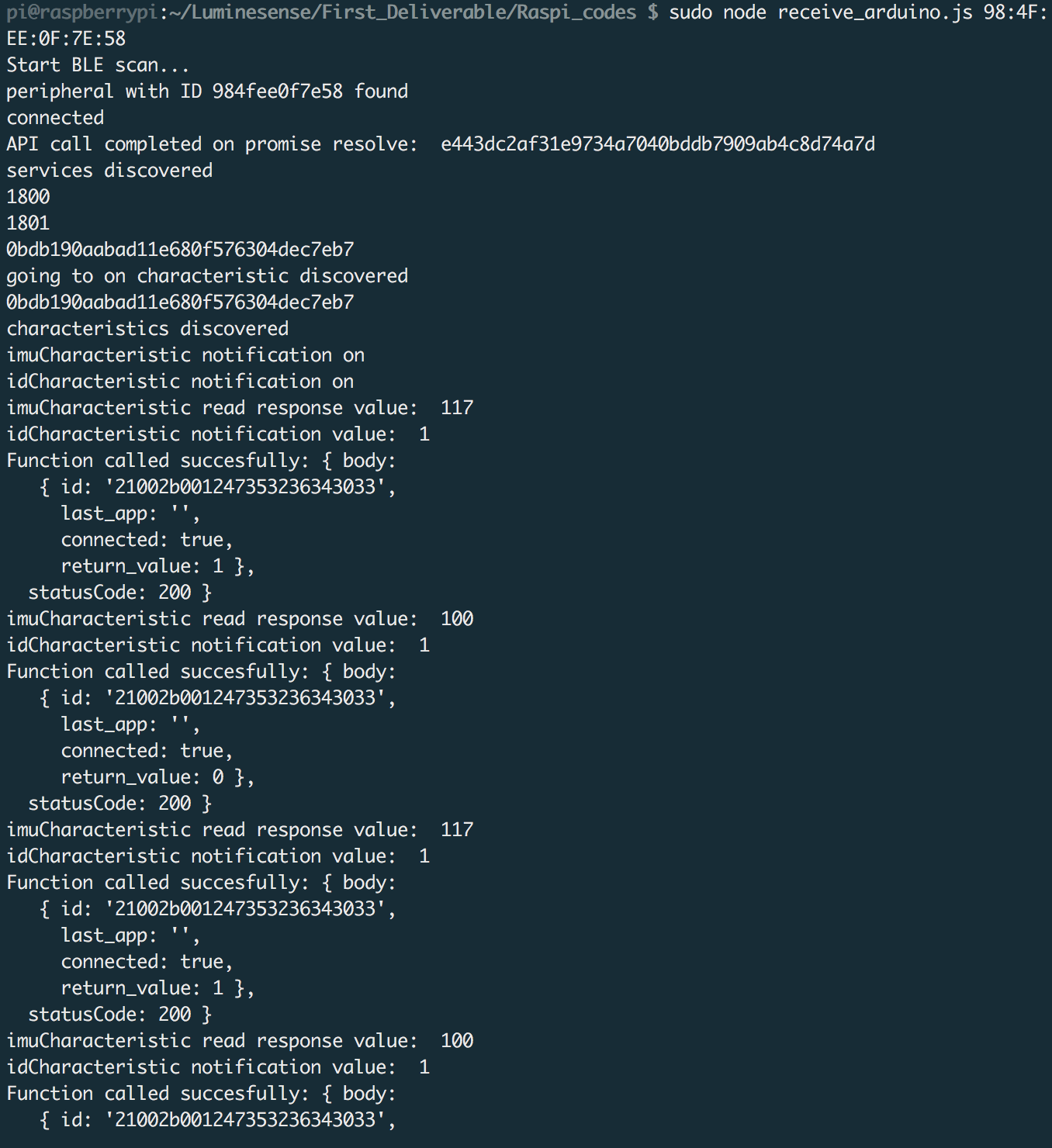
The other data that is send over the communication loop from the Arduino is the Luminaire ID, which is acquired using the photodiode circuit. Because we only have a prototype of the Luminaire from the client, this setup is only in the early stages of design, and we are actively working with the client in order to improve this aspect of the wearable. The photodiode senses light intensity using an amplifier circuit, which is translated to an ID based on this intensity. For this test there is only one Luminaire, so this ID is 1. This sensing shows a proof of concept of the communication between the Luminaire and the wearable, which is one of the components of the communication loop that we are testing in this deliverable.

Now that we have these two pieces of data, we must send them to the Raspberry Pi 3, or the “central base,” which also consists of a router. This is the second link in our communication loop, which is established using Bluetooth Low Energy. The Raspberry Pi runs the “Noble” open source ble library in node.js in order to connect with the Arduino and read the ble “characteristics” when they are updated with new data. The two ble characteristics are the command and the light ID. By showing that the Raspberry Pi is able to successfully read and understand these commands, we have shown proof of two links in our communication loop working as intended: the one between the Luminaire and the Arduino, and the one between the Arduino and the Raspberry Pi.



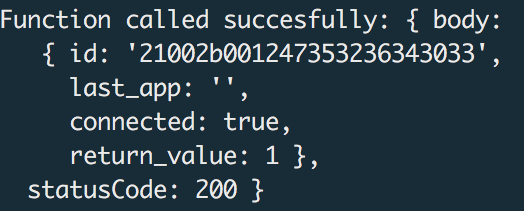
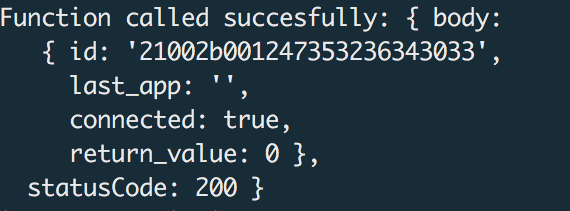
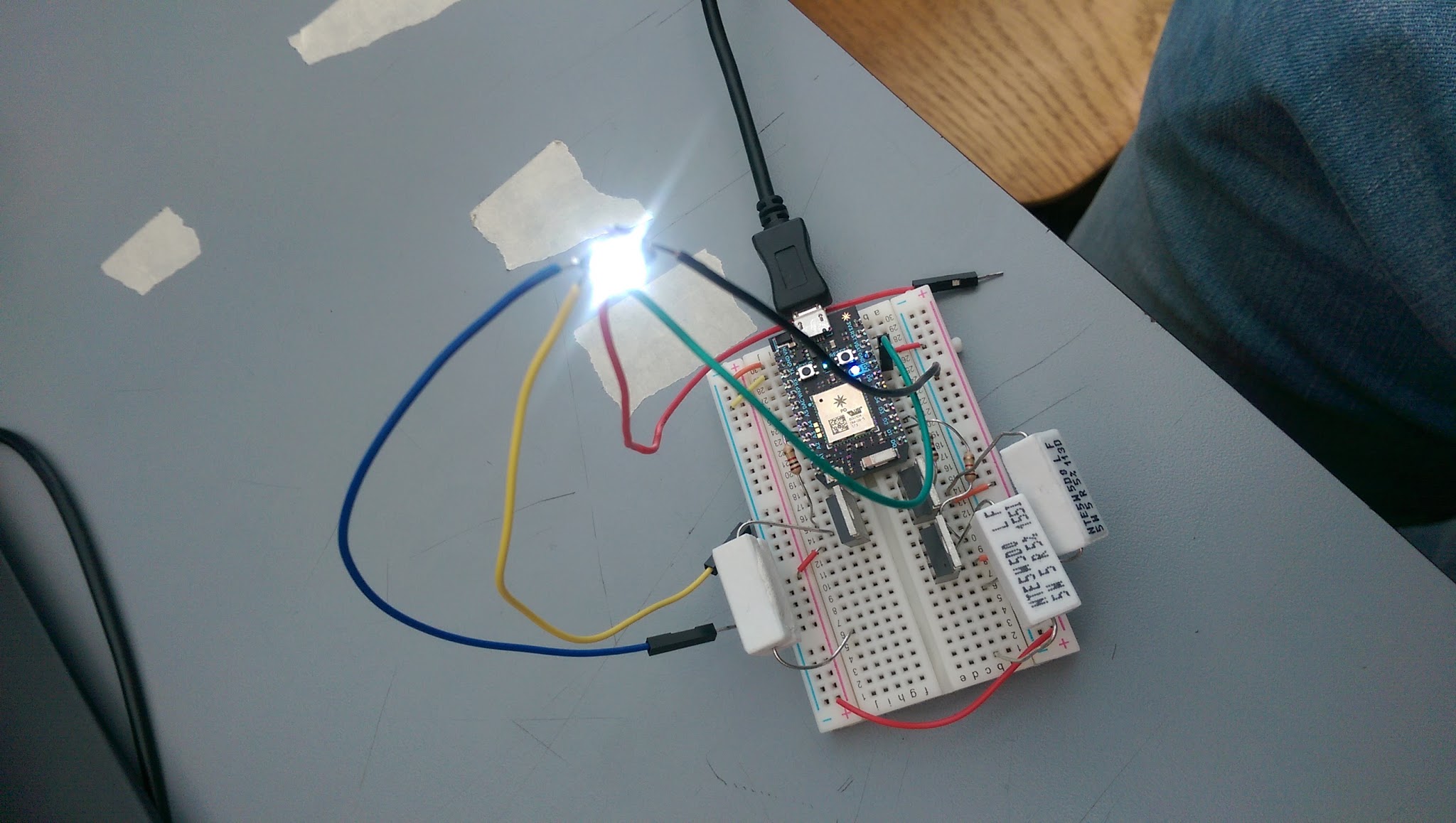
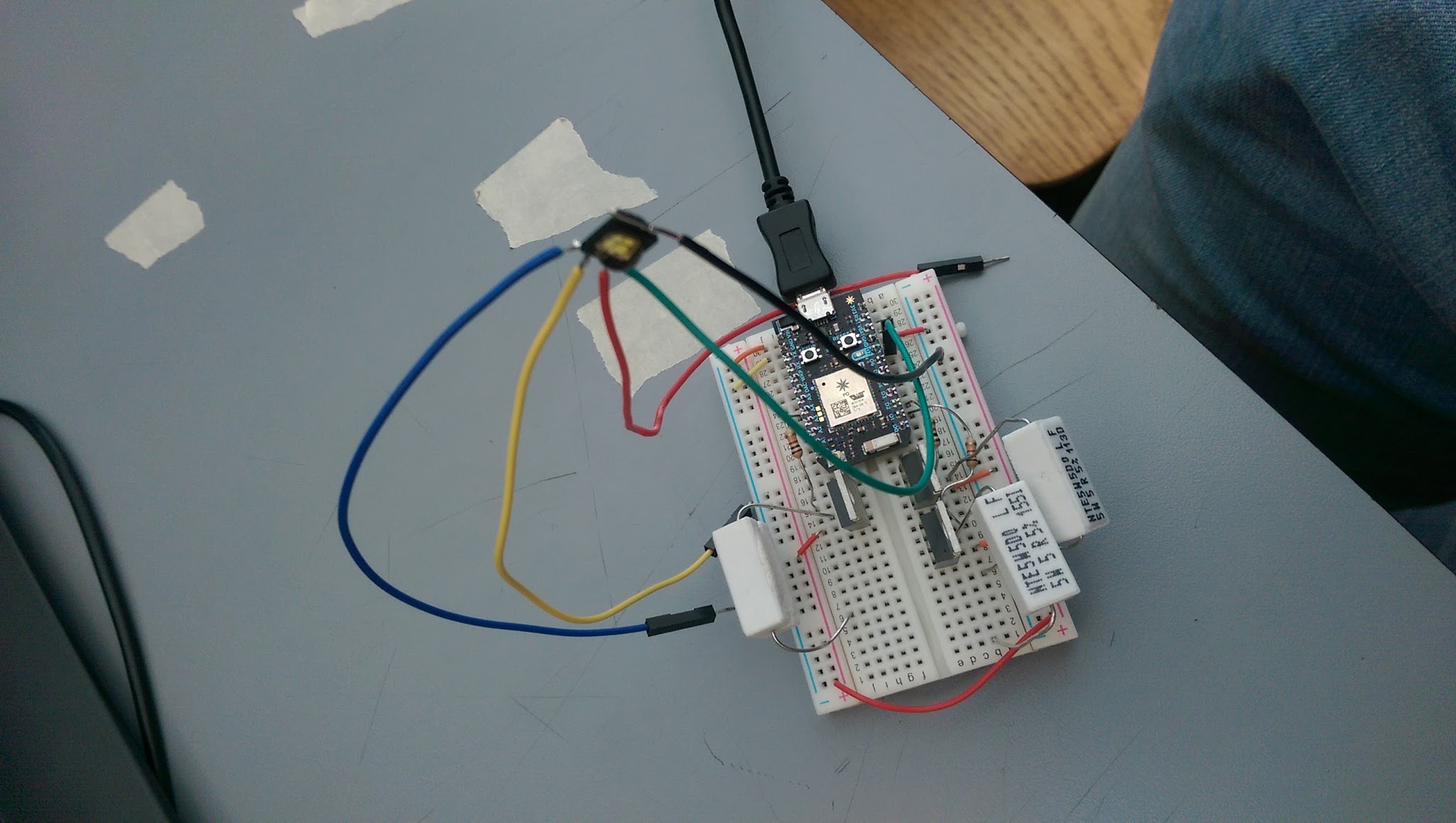
*Figure 6. The Serial Monitor on the Arduino outputting light, gesture, and bluetooth connection data*

*Figure 7. The Raspberry Pi CLI output showing a successful bluetooth connection to the Arduino*



*Figure 8. The Raspberry Pi CLI output showing gesture command (the value 117 corresponds to “u” in ASCII for the “on” command)*

The final portion of the communication loop is the Raspberry Pi to Luminaire connection. This is how the light actually receives commands. This is done by a call to the Particle Cloud API on the Raspberry Pi using node.js. The Raspberry Pi authenticates a connection to the Particle Photon on the Luminaire by providing login credentials. The command that the Raspberry Pi received from the wearable are parsed and then passed as an argument to a function on the Particle Photon. The Particle Photon then carries out the command, which in this test is to turn the light on or off. This is the ultimate test of our communication loop: executing a gesture on the wearable, and watching the light on the Luminaire react accordingly.



*Figure 9. Luminaire state compared with command outputs*

This proof of concept of our communication loop is an accomplishment as it shows that one of the most basic foundations of our system is functional. It validates our entire system design and allows us to flesh out each individual component with more features and depth, and allows us to test the functionality of each of these future features by integrating them into the comhas faith that the project will be successfully implemented.

# Technical Plan

Task 1. Battery power supply for Wearable

In order to reduce pollution and increase energy efficiency, a rechargeable lithium battery will be employed for the wearable. The current battery choice has dimensions of 60 mm x 36 mm x 7 mm which is about the size of a regular smartwatch. The output voltage is 3.7V. The pins on the Arduino boards runs voltage at 3.3V but will tolerate up to 5V, which is voltage of the USB port power of a laptop. The capacity of the lithium battery is 2000mAh, sufficient enough to power the wearable for 2 to 3 days. A more granular calculation of the duration of a fully charged watch will depend on the frequency of the wearable’s usage. With the battery safety issue faced by the Samsung Note 7 in mind, ensuring the battery does not endanger the user is paramount. The consumption will be tested and verified by timestamping sessions of usage and compare the battery levels. Lead: Harry.

Task 2. Light modulation

The Wearable device needs to detect and identify individual Luminaires from an array (section) of Luminaires. This will be solved by assigning PWM frequencies to each Luminaire (e.g. 1100Hz = ID 1, 1200 Hz = ID 2, etc.) and observing their frequencies using an optical receiver on the Wearable. The receiver determines frequency (ID) by latching onto rising edge of received PWM signal determining the time between pulses. The Luminaires will be initialized to these frequencies using the Particle Photon. This code will be developed and deployed using the Build IDE and Arduino IDE. The software created must account for 16 Luminaires i.e. 16 unique ID’s. The optimal frequencies will be calculated via testing in order to avoid adjacency issues (e.g. assigning differentiable values to neighbouring Luminaires). Additional testing will involve verifying assigned IDs by printing ID and frequency values to the CLI (Command Line Interface). Lead: Tanatsigwa.

Task 3. Luminaire intensity adjustment

The intensity levels of the Luminaires should be adjustable. This will be solved by utilizing the color properties (RGBW) of the light controlled by PWM. This function will depend on the gesture commands issued by the user. The Particle Photon will have a function that sets the duty cycles (collective RGB values) of the selected Luminaire to the desired levels. This code will be developed and deployed using the Build IDE. The design will be tested on a single Luminaire and verified using photodiode sensor readings and Photon function output values before and after dimming commands. Lead: Tanatsigwa.

Task 4. Luminaire color adjustment

The color levels of the Luminaires should be adjustable. This too will be solved by utilizing the color properties (RGBW) of the light controlled by PWM. This function will depend on the gesture commands issued by the user. The Particle Photon will have a function that sets the duty cycles (individual Red, Green, and Blue values) of the selected Luminaire to the desired levels. This code will be developed and deployed using the Build IDE. The design will be tested on a single Luminaire and verified using observed Luminaire color readings and Photon function output values before and after color-altering commands. Lead: Tanatsigwa.

Task 5. Gesture Library

The gesture library is the location where the user can add, design and modify all the gestures that they want. This is the best way to give the user’s complete control over their preferences. If the user does not want to modify or add any of their own gestures, they can just use the provided gestures that are built into the wearable. This gesture library will be written in C/C++ as all of our code relating to the Intel Curie is in C/C++, so it will be easier to implement. This will be implemented in the Intel Curie as in this platform, we will be figuring out the gesture provided by the user. Lead: Sneha

Task 6. Machine Learning

To learn the movements of the user and make the user experience even better, we will implementing a machine learning program. This program will learn the movements of the user and the gesture that they provide, so that the machine can figure out who the user is and make the actions more and more accurate. This will also be very helpful in terms of security as with the gesture library, we can know if anyone who is not supposed to use the wearable uses the wearable. This could come in very handy in the future if we plan to integrate this device with any existing wearables in the market. Lead: Sneha

Task 7. Cloud Integration

A cloud based solution to storing user preferences, gesture machine learning data, user identification information, and energy usage information. The Raspberry Pi shall interact with this cloud server in order to interact with and update this data. The information stored on the cloud must not pose a security risk for the user, such as revealing dates that the user is not present in their home. Lead: Caroline

Task 8. Web Application

The web application will enable the user to interact with the information stored on the cloud. They will be able to see their gesture preferences, add new ones, or modify the existing ones. Moreover, the users will be able to interact with and change the set of users allowed to interact with their system using the wearable. Lastly, the users will be able to access and interact with their energy usage data. Lead: Sneha

Task 9. Wearable Design

The Wearable device will be designed to be comfortable and easy to wear. A prototype will be fabricated and user tested to ensure that the wearable fulfills the customer’s requirements. The prototype will be designed in a CAD software such as SolidWorks or Creo, and the necessary materials will be 3D printed in the client’s personal lab. The prototype will be tested by user analysis and fitting on many subjects in order to establish an optimal wearable size. The wearable will also have an LCD screen to give a better user experience to the users. Leads: Sneha and Harry

Task 10. Apply PCB

We will design a PCB (Printed Circuit Board) for the wearable. The board will be used to connect and embed relevant hardware: the Intel Curie (with IMU), optical sensor, a photodiode, and the battery power supply. The PCB will go through various iterations, therefore many will be designed and tested in order to establish an optimal design aligned with system and client constraints (such as capacity to host all permanently connected/soldered electronic components) Tests will be done to achieve a balance amongst form factor and power/heat dissipation. Leads: Harry and Sneha

Task 11. Multiple Luminaire control

If they choose to do so; the user should be able to control the states of many Luminaires simultaneously (instead of menially selecting and gesturing to each individual Luminaire). This function will depend on the gesture commands issued by the user. The user will select any Luminaire in a given row and the appropriate gesture will affect all the Luminaires in the same row. This will be implemented by using the light modulation strategy described in Task 2. The assigned frequencies/IDs can be mapped into a matrix to represent the positions of the Luminaires in the room. This matrix will be accessed to select all the relevant Luminaires when the user performs a multiple Luminaire gesture. This code will be developed and deployed using the Build IDE. This will be evaluated and verified by comparing matrix index values, Luminaire state values printed to the CLI. Lead: Tanatsigwa Assist: Harry

Task 12. Adaptive Mode control

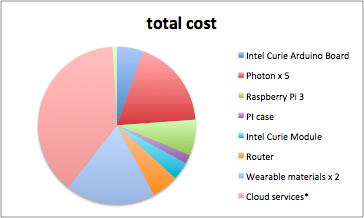
When no wearable is connected to the central hub, the system should react to user activity in a room through the adaptive mode. To implement this, a single pixel camera (provided by the client) will be sampling “occupancy data” at a fixed time interval (e.g. <1 s). This binary data will be communicated over a serial bus to the Raspberry Pi. An algorithm will then be developed on the RasPi to extract the “occupancy status” of the room, in other words indicate whether any users entered/exited. Control logic will be needed to update the Luminaires accordingly (e.g. turn off all lights when room has been empty for a certain amount of time). Lights will be updated according to the presence and location of users in the room. To test this design, have test subjects enter/leave the room in various time intervals from various entry points and monitor the occupancy status on the Raspberry Pi. Lead: Michael

Task 13. Multiple User control

When multiple users are interacting with the system (e.g. 2 or more wearables), there needs to be a clear protocol for avoiding conflicts. To ensure this, a lock system will be implemented on light resources to ensure the system is responding to only one command at a time. Users should release locks once their commands have been realized, at which point any users waiting to command can acquire the lock and update the Luminaires. There should be control logic to ensure lights are updated correctly and consistently in the situation with multiple users. Lead: Michael

# Budget Estimate

|  |  |  |  |
| --- | --- | --- | --- |
| **Item** | **Description** | **Quantity** | **Total Cost** |
| 1 | Intel Curie Arduino Board | 1 | $29.99 |
| 2 | Particle Photon | 5 | $99.95 |
| 3 | Raspberry Pi 3 | 1 | $39.99 |
| 4 | Raspberry Pi case | 1 | $9.99 |
| 5 | Intel Curie Module | 1 | $20.00 |
| 6 | Router | 1 | $30.00 |
| 7 | Wearable materials (includes PCB) | 2 | $100.00 |
| 8 | Cloud services | 3 months | $209.79 |
| 9 | Battery | 2 | $9.92 |
| 10 | Luminaire |  | Provided |
| 11 | Motion capture system |  | Provided |
|  | Total | **$549.63** | |



Budget Constraints:

The Luminaire and Motion capture system are both part of the customer’s research and will be provided to the team so that we are able to fully integrate our project with their existing systems. Cloud services cost calculated assuming use of Amazon RDS at 10% usage. This will differ depending on cloud technology choice (if changed) as well as usage rate. Five Photons are needed for testing of multiple Luminaires in a room. The amount of particle photons used (e.g. if 16 are ordered to couple with 16 Luminaires) will also affect the total cost. Cost of wearable materials was estimated by client to be $50. We have money to build two wearables in order to test our user command conflict resolution algorithm. This number could fluctuate once decisions are made on specific wearable materials.

# Attachments

# Appendix 1 – Engineering Requirements

Team 1 Team Name: Luminesense

Project Name: Adaptive, Gesture Based Lighting System

|  |  |
| --- | --- |
| **Requirement** | **Value, range, tolerance, units** |
| Wearable optical sensor range | ~ 40.0 ft;  the size of the client’s laboratory |
| Wearable Bluetooth Low Energy range | > 30.0 ft; |
| Bluetooth Low Energy data rate | ~ 1 Mbit/s;  < 1s to send gestures |
| Wearable dimensions | 68.0 mm x 38.0 mm x 16.0mm |
| Wearable weight | < 40.0g |
| Wearable power consumption | 3.7V Lithium battery;  2000mAh |
| Wearable cost | < $50.00 |
| Luminaire quantity | 16 |
| Photon data rate | ~ 65 Mbit/s;  < 1s to implement commands |
| Wi-Fi range | > 30.0 ft;  the size of the client’s laboratory |
| Wi-Fi configuration | Open, WEP, WAPI, WPA and WPA2-PSK |
| Raspberry Pi Bluetooth Low Energy range | > 30.0 ft |
| Raspberry Pi Bluetooth Low Energy data rate | ~ 1 Mbit/s |
| Raspberry Pi Wi-Fi data rate | > 65 Mbit/s;  < 1s to transmit commands |

# Appendix 2 – Gantt Chart

# 15419324_1890140464549979_1477525441_o.png

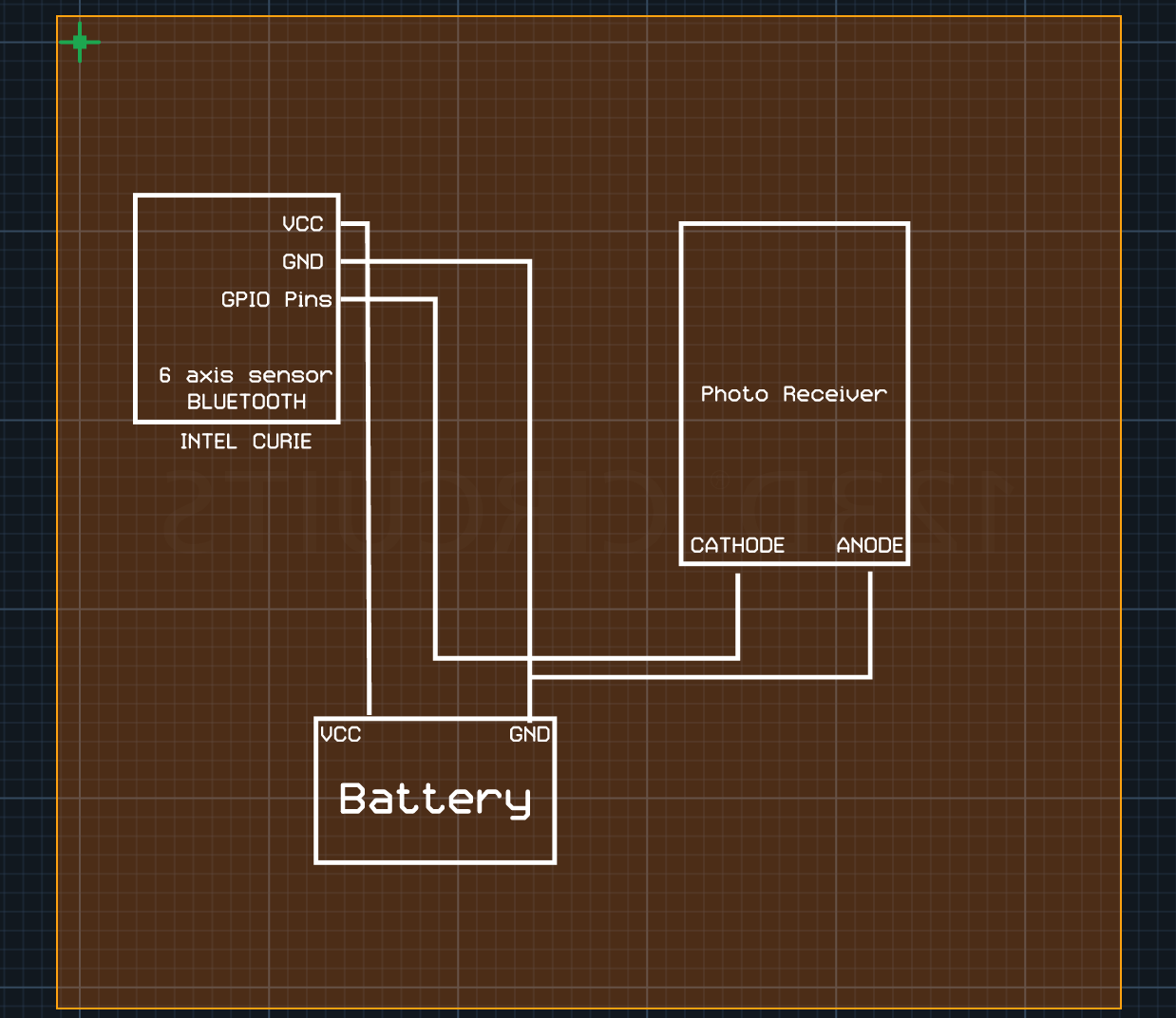
# Appendix 3 – Other Appendices

Technical Reference

* <https://www.arduino.cc/en/Reference/CurieIMU>
* <https://github.com/sandeepmistry/noble>
* <https://docs.particle.io/reference/firmware/photon/>

Drawings and Schematics

The wearable:



Biography of each member:

Sneha Pradhan is an undergraduate student pursuing a Bachelor’s Degree in Computer Engineering at Boston University. Born and raised in Kathmandu, Nepal, Pradhan strives to makes a difference in the third world countries with the knowledge of technology.

Phone Number: 857-272-1257

Email: [snehha@bu.edu](mailto:snehha@bu.edu)

Michael Haley is a senior at Boston University, pursuing a Bachelor’s Degree in Computer Engineering. He was born and raised in Quincy, MA and spent last semester at the University of Sydney in Australia.

Phone Number: 857-939-8554

Email: [mhaley10@bu.edu](mailto:mhaley10@bu.edu)

Caroline Jones is a senior studying Computer Engineering at Boston University. She is from Los Angeles, California and has interned at both the University of Southern California and The Aerospace Corporation. She will start as a full-time employee at The Aerospace Corporation in June 2017.

Phone number: (818) 919-5444

Email: [ctjones@bu.edu](mailto:ctjones@bu.edu)

Zehua (Harry) Zhao is a senior studying Computer Engineering at Boston University. He is from Henan, China. He transferred to BU from Pine Manor College after completing his first year there. He will be graduating in May 2017 after completing BU’s Computer Engineering course in 3 years.

Phone number: (818) 919-5444

Email: zehuaz@bu.edu

Tanatsigwa Hungwe is a product engineer for adapptED Co. Currently, Tanatsigwa is enrolled in Boston University’s awesome Computer Engineering undergraduate program. He is from Harare, Zimbabwe and pursuing a minor in Film and Television.

Phone Number: (617)-963-6108

Email: hungweta@bu.edu