March, 2014

Dr. Andrew Rawicz
School of Engineering Science
Simon Fraser University
Burnaby, British Columbia
V5A 1S6

Re: ENSC 440 Design Specifications for the BikeSmart System

Dear Dr. Rawicz:

This enclosed is a design specification for the BikeSmart System. This document provides a set of technical guidelines for design of BikeSmart System. The BikeSmart System is a smart bike safety system that allows cyclists to aware other road users before their next movement, without using the existing method of hand signal. In this project, we will equip the bicycle with a controller and a voice recognition system on the helmet to control signal lights.

The design specifications demonstrate the conceptual model of our design. To justify our design, the design specifications of each part are divided into two categories of hardware and software implementations. The hardware section will emphasize on design and assembly of the model; however, the software section will describe the coding and graphing design on any CAD software.

Please feel free to contact our CCO via phone: 778-889-2677 or email: ntehranc@sfu.ca, if you have any question or comment regarding to our design specification.

Sincerely,

Jason Coo

Jason Coo
President and CEO
DreamRide Corporation

Enclosure: Functional Specification for a BikeSmart System

Design Specification for the BikeSmart System

A smart and safe bicycle system

PROJECT TEAM
Stan Yang
Nadia Tehranchi
Jason Coo
Paul Chen
Conrad Wang

CONTACT PERSON
Nadia Tehranchi (CCO)
ntehranc@sfu.ca

SUBMITTED TO
Dr. Andrew Rawicz – ENSC 440W
Steve Whitmore – ENSC 305W
School of Engineering Science

ISSUED DATE
March 10, 2014

Executive Summary

Old method of hand signalling will put cyclists in more danger and risk of death while sharing the road with other road users. Many cyclists are injured or killed every year especially during the rainy season due to losing control. When it comes to cycling safety, visibility is the key. However, the DreamRide Corporation does not think those methods are safe to use while riding a bicycle. Therefore, the DreamRide team came up with an innovative technology for bicyclists in order to feel safer while riding. The BikeSmart System will provide user the safety, comfort, and quality they deserve.

The BikeSmart system will have two modes, as discussed in the *Functional Specification* [19] document: slide switch control and voice recognition. In the slide switch control mode, the user will be able to control the turn signal and any other added features to the system according to customer's needs. In addition to the tow modes, a speed indicator will be implemented on a control switch box. In this mode, the slide switch control will be wired. In the voice recognition mode, the turning signal will be controlled by a wireless microphone that is integrated in the helmet. The users will be able to communicate with the system via voice. For instance, by saying "right" or "left" into the microphone, the turning signals will be activated accordingly. Furthermore, our team will confirm that the system meets all the standards and safety Guidelines as discussed in previous document.

The team will follow all the design requirements in this document. This document encloses the design on BikeSmart and provides more detailed of design specification for LED, slide switch control, voice recognition, and hardware packaging. In addition, software and hardware part of each design section will be described in more details. Finally, the test plans are designed to ensure the quality and safety of our model.

The project will be a span of 4-month of challenging work. The prototype design with its specific features will be ready by April 1, 2014. We believe that our system with its affordable price range and incredible quality can accommodate many customers in the market.

Contents

List of Figures

List of Tables

Acronyms

ABS	Arylonitrile-butadiene-styrene copolymer
A/D	Analog to Digital
AMP	Amplifier
DH	Destination Address High
DI	Data In
DL	Destination Address Low
DO	Data Out
ISM	Industrial scientific and medical (radio bands)
IEEE	Institude of Electrical and Electronics Engineers
LED	Light-emitting Diode
NI	Node identifier
PLA	Polylactic acid which is a bio-degradable polymer
PWN	Pulse Width Modulation
RGB	Red Green Blue
SH	Serial number high
SL	Serial number low
USB	Universal Serial Bus
VIN	Voltage input

Glossary

Accelerometer	A device that measures proper acceleration
Arduino	An open source development board
Atmel	Atmel Corporation is an American-based designer and manufacturer of semiconductors, founded in 1984
ATMEGA328	A microcontroller created by Atmel Corporation
CTS	Hardware flow Control for data in buffer
Colourduino	An Arduino shield for LED matrices
dBm	An abbreviation for the power ratio in decibel(dB of the measured poer reference to one milliwatt(mW)
DTR	Hardware flow control line
FT232R	the latest device to be added to FTDI's range of USB UART interface integrated Circuit Devices
GND	The ground pin on the Arduino Uno board
Li-Ion Battery	Lithium Ion Polymer Battery
MakerBot	3D printer device
MakerWare	A free and powerful software that works in tandem with MakerBot 3D printer
mcd	Milli-candela; The unit to measure LED intensity
MCU	Multipoint control unit
RTS	Software flow Control for data in buffer
RXD	Arduino digital pin 0, Digital receiver
RST	The reset pin on Arduino
SketchUp	3D modeling software
TXD	Arduino digital pin 1, Digital transmitter
UART	Universal Asynchronous Reveicer/Transmitter
	Bee Socket - An adpter of Xbee serials which can connect wireless mudules with Arduino, such as WIFI Bee, etc.
X-CTU	Configuration and testing software to test XBee in a mash/network
Xbee	The brand name from Digi international for a family of form factor compatible radio modules
µSpeech	A voice recognition library for Arduino

1.0 Introduction

The BikeSmart system is a new technology for cyclists who risk losing control while using the traditional hand signaling method. The BikeSmart system contains a signal light device which can be controlled by voice recognition system on the helmet and a slide switch control system on handle bar. The objective of DreamRide Corporation is to provide cyclists a safe and comfortable riding experience as our famous quotes is that “visibility is a key”. This design specification document describes the detailed descriptions and requirement of the design and development of our proof-of-concept model.

1.1 Scope

This document details the necessary design requirement of the BikeSmart system according to what was described before in *Functional Specification for a BikeSmart System* [19]. The design specification will describe the requirements for the proof-of-concept model and the design techniques used. It should be noted that the design specification will not contain all functional details in the functional specification document.

1.2 Intended Audience

The design specification is proposed to guide the engineers of DreamRide. The document will be used as a guidebook for the design details of the BikeSmart system. The team members shall refer to the design specification to test the final product and verify the BikeSmart system’s safety and quality.

2.0 System Overview

The BikeSmart system is consisted of two primary features which are designed to control the LED signal lights installed under the bicycle seat. The first primary function is the voice recognition system that picks up keywords such as the world left, right, or stop from the cyclists. These analog inputs are converted into digital inputs allowing the microcontroller to send the correct signal to the LED panel. Instead of detecting keywords from cyclists’, the second primary feature consisted of a switch mounted on the handlebar, such that the cyclists’ is able to select the corresponding LED signal lights by simply sliding the switch. In addition to the primary features, the BikeSmart will have a monitoring system which is used to display information like calories, slope, and heart rate to the cyclists. Combining the primary and secondary features, the BikeSmart will provide cyclists a safe and functional riding condition.

The following open loop flow chart shows how the BikeSmart system will respond to the cyclist.

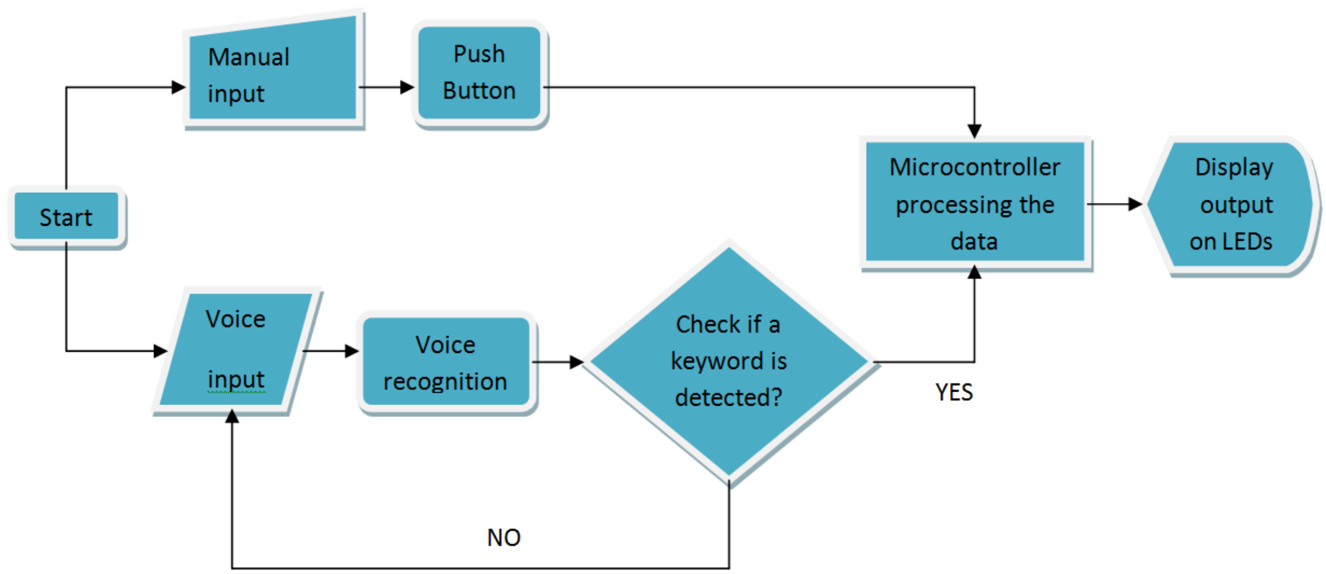


Figure 1: Flowchart of BikeSmart System

3.0 LED Panel Design

3.1 Overview

As the main purpose of the BikeSmart System is to allow better communication between road users and cyclists, the LED panel plays the most important role. Initially, we have two deigns of the LED panel. The first one consists of using two 8x8 LED matrices with the left panel to display left turn signal and vice versa. For brake signals, the 4 column on the right of the left panel and the 4 column on the left of the right panel will be used. Another design uses only one 8x8 LED matrix. As shown in Figure 2, the middle 16 LED are used to display brake signal and the side LEDs are used to display the turn signals.

Following the guidelines on the functional specification document, as shown below, the design of the LED panel must fulfill each requirement.

[R52-A] The LEDs shall be bright enough to see from a distance of at least 30 meters away.
[R53-A] The direction signal shall be yellow arrows, and brake signal shall be red.
[R54-A] The signals shall be easy to understand by other road users.
[R55-A] The length of the LED panel shall not exceed the width of the seat.
[R56-A] The LED panels shall be waterproofed in case of rainy seasons.
[R57-A] The LED panels shall be operate in the temperature range of -40~85 degrees.

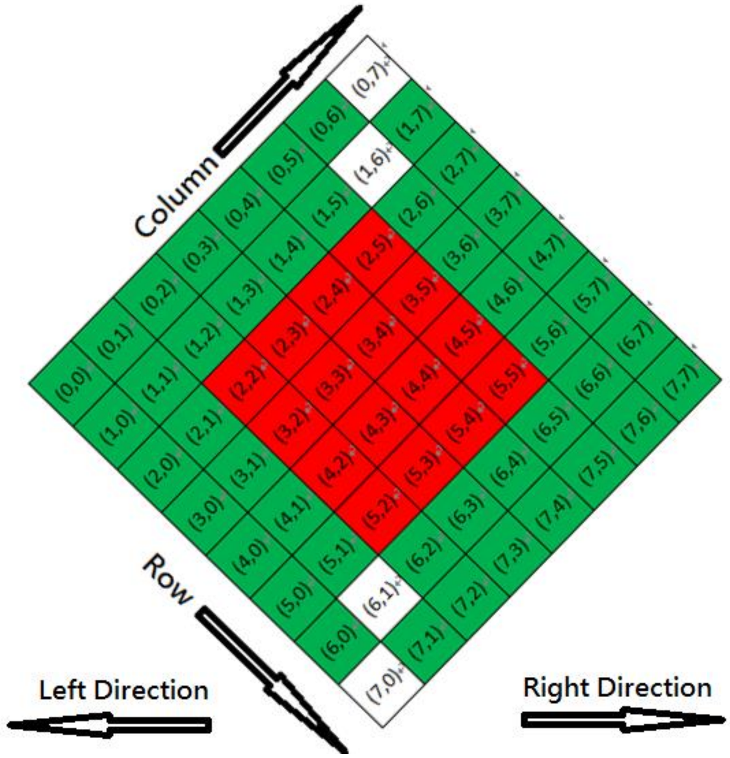


Figure 2: LED Matrix Display Design

While there is only limited space on the back of any bicycle, we decide to use the later design where only one LED matrix is used to minimize the use of space, allowing the BIkeSmart system to be installed on all computers.

According to the functional specification document, the following requirements must be met.

[R5-A]	The LEDs shall be bright enough to see from a distance of at least 30 meters away.
[R6-A]	The direction signal shall be yellow arrows, and brake signal shall be red.
[R7-A]	The signals shall be easy to understand by other road users.
[R7-A]	The length of the LED panel shall not exceed the width of the seat.
[R8-A]	The LED panels shall be waterproofed in case of rainy seasons.
[R9-A]	The LED panels shall be operate in the temperature range of -40~85 degrees Celsius.

3.1 Hardware

The 8x8 LED matrix with 60mm square RGB LEDs, as shown in figure 3, is decide to used, while it offers a very efficient light density.

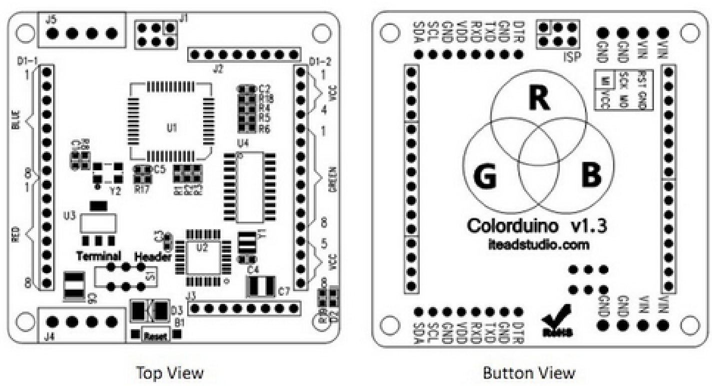
Figure 3: 8x8 RGB LED Matrix

As the size of this LED matrix is suitable for the BikeSmart system, however, it requires a LED matrix shell to configure and test. While the LED can be configured into 3 different colours of LED, our system requires only two colours to display directions and brake signal, yellow and red respectively. As the LEDs need to be programmed, a Colourduino V1.3 LED Matrix driver is acquired. It has three 8+6-bit channels of hardware PWM control of the LEDs.

Dot Size	5.0mm
Pixel Array	8×8
Luminous Intensity	40mcd
Package Dimension	60mm×60mm
Reverse Voltage(Max)	5V
Forward Current(Max)	25Ma
Peak Forward Current(Max)	100Ma
Power Dissipation(Max)	100Mw
Operating Temperature(Max)	-35~+85℃
Storage Temperature(Max)	-35~+85℃
Lead Solder Temperature(Max)	260℃ for 5 seconds

Table 1: RGB LED Matrix Specification

Figure 4: Colourduino shell



Pad Name	Type	Description
SDA	I/O	Data wire of IIC Bus
SCL	I/O	Clock wire of IIC Bus
GND	G	Ground plane
VDD	P	Power wire for all digital components
RXD	I/O	Data wire of UART Bus
TXD	I/O	Data wire of UART Bus
DTR	I	Special reset wire of Arduino program
VIN	P	Power wire for all LEDs and Super current driver
MI	I/O	Data input of ISP
MO	I/O	Data output of ISP
SCK	I/O	Clock input of ISP
RST	I	Reset input of ISP

Figure 5: Colourduino Specification

3.2 Software

In figure 2, the red area of the matrix will display a steady light when the 3-axis accelerometer senses a decrease in acceleration of the bicycle. The right half of the green area of the matrix presents the right direction when it receives the "Right" command from the voice control. The other half of the green area of the matrix presents the left direction when it receives the "Left" command from the voice control. The left upper corner of the matrix is (0,0). We are using a "for loop" to create left and right arrows to represent their respective directions. We also came up with an algorithm to show red steady lights in the middle for the "stop" signal.

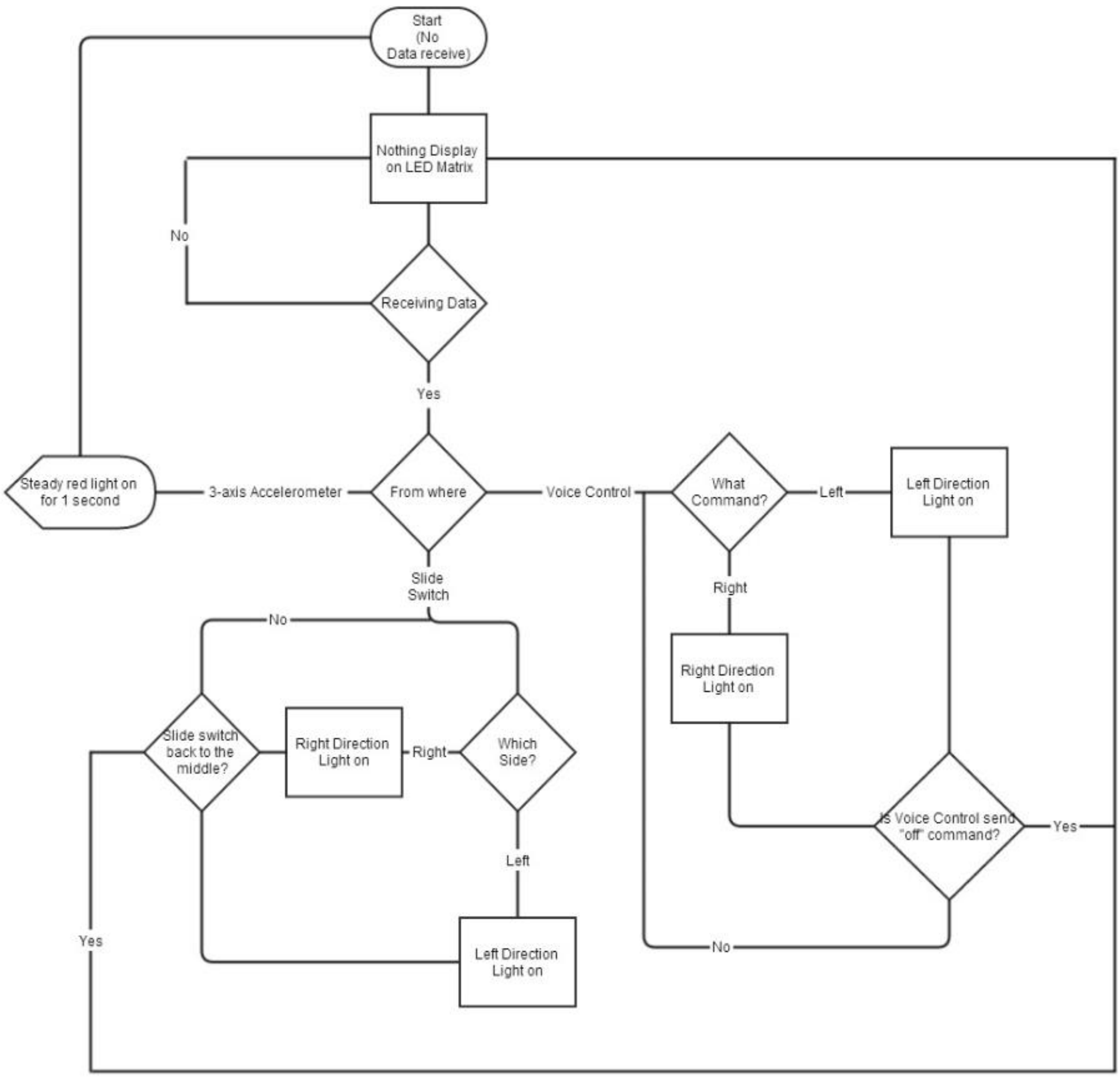


Figure 6: LED Matrix Display Design Flowchart

3.3 Three-axis accelerometer ADXL335

Figure 7: ADXL335: 3-Axis Accelerometer

The ADXL335 is a complete 3-axis acceleration measurement system. The ADXL335 has a measurement range of ±3g minimum. The output signals are analog voltages that are proportional to acceleration. It is capable to measure the static acceleration of gravity of many applications or even the dynamic acceleration of motion, shock or vibration. We will use this component to help sensing when the bicycle comes to a stop or even decelerate and display the red steady light on our matrix. We did not yet to know how to integrate it into our system, as it involves many math and coding difficulties.

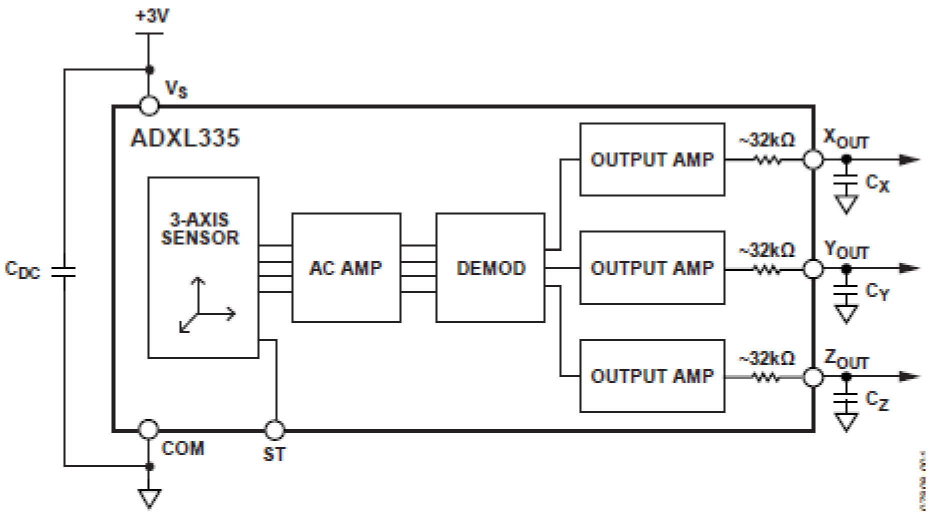


Figure 8: ADXL335 Functional Block Diagram

According to the functional block diagram shown in figure 8, we need to configure the bandwidth of the accelerometer by adjusting Cx, C*y*, C*z* capacitors at the X_{out}, Y_{out} and Z_{out} pins to suitable values. We also need to select the right bandwidth to match our product. The typical change in output is -1.08g (corresponding to -325 mV) in the X-axis, +1.08g (+325 mV) on the Y-axis, and +1.83g (or +550mV) on the Z-axis. However, as the supply voltage increases, voltage on those axes (X, Y, Z) will also increase. As a result, if we provide more supply voltage to the 3-axis accelerometer then we can collect more data from it. Figure 9 below clearly shows that the 3 axis of acceleration sensitivity. As the accelerated along the sensitive axis, the corresponding output voltage increase.

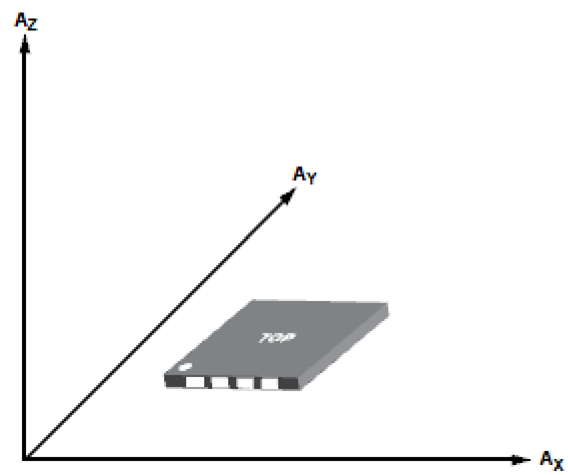


Figure 9: 3-Axis Accelerometer Sensitivity

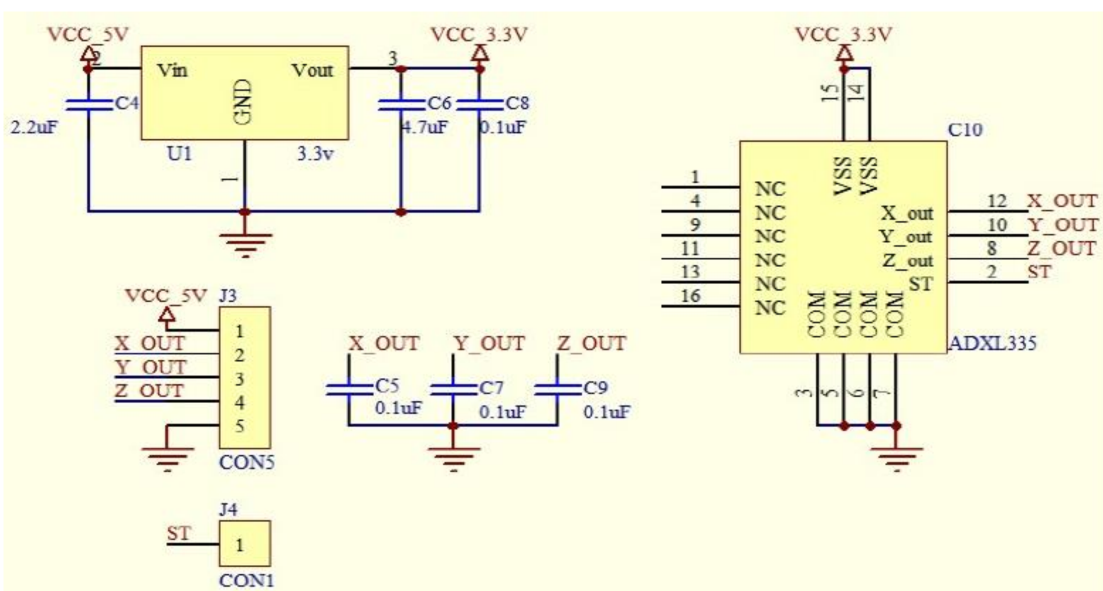


Figure 10: 3-Axis Accelerometer Schematic

4.0 Slide Control Design

4.1 Overview

As one of the major features, the slide control design is crucial to the BikeSmart system. We choose to base the system on the Arduino Uno. Even though it is more expensive than other development boards on the market, there are many Arduino libraries available on the internet, including the Colourduino and the speech recognition library, which allow us to focus the time and effort on more difficult tasks. In addition, while all members are familiar with C/C++, Arduino coding is based on C/C++. We feel that such benefit could further save us on time and justify the extra cost.

4.2 Electronic Design

As shown in our functional specification document, the following requirement must be met for the proof of concept system.

[R49-A] Sliding left on the switch shall turn on the left turn signal.
[R50-A] Sliding right on the switch shall turn on the right turn signal.
[R51-A] The sliding switch shall provide feedback in the controller box.

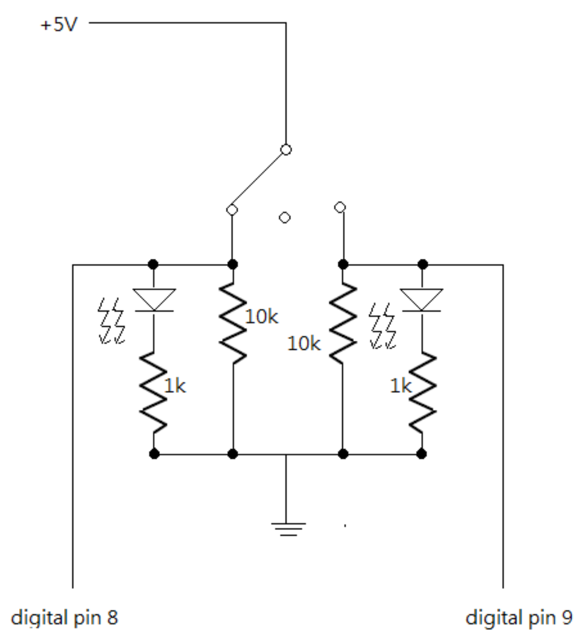


Figure 11: Slide Switch Schematic

Figure 12: On-off-on Slide switch

As the Colourduino LED matrix shield and Arduino Uno board required for the proof of concept model, a 5V input voltage is required. When the switch is slided to the either the left of right side, the circuit will pass signal to pin 8 or 9 respectively on the Arduino Uno board as a digital input. The two LED will provide feedback and will be used as user feedback. Sliding the switch to the middle will not output anything at all, as it is connected to ground.

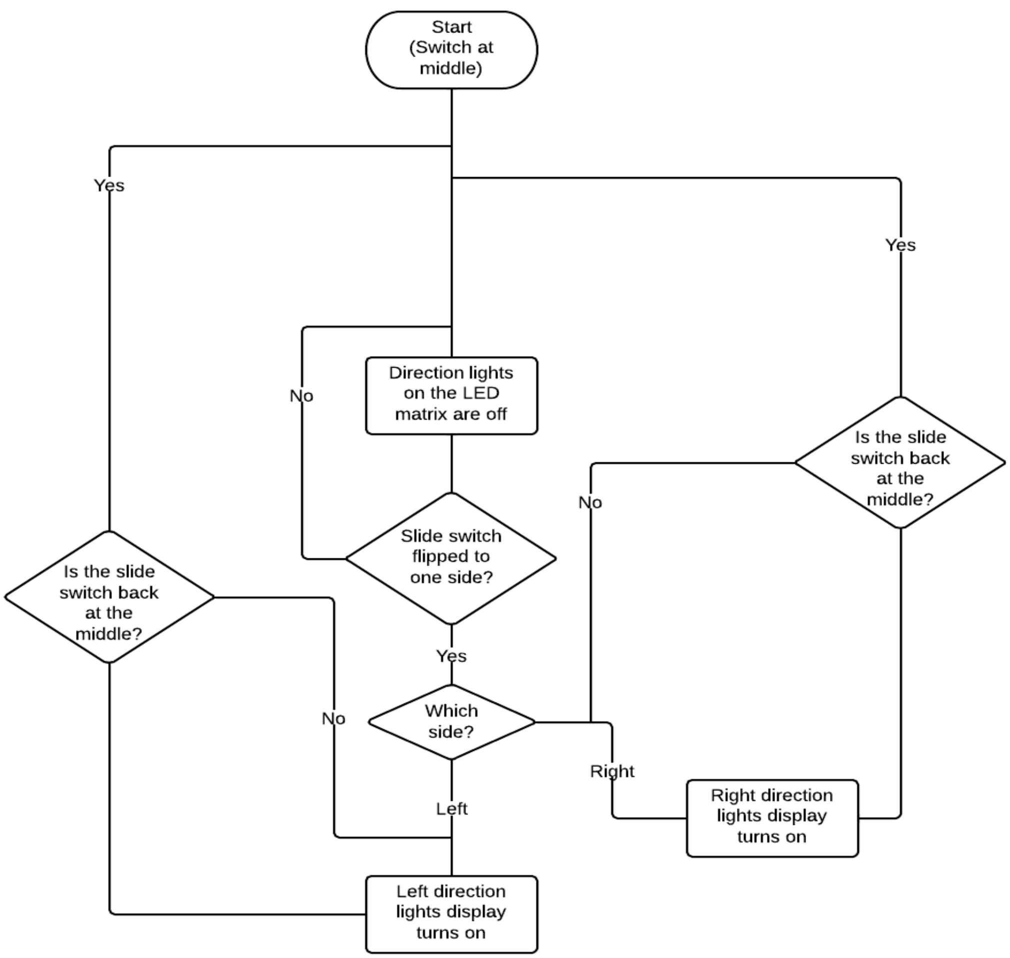


Figure 13: Slide Switch Flowchart

5.0 Voice Recognition System

5.1 Overview

Voice recognition is one of our major features which will be established on helmet with a microphone. It will detect keywords from the users and then it will send the signal to LED light. According to the functional specification document, the following are the requirements that the BikeSmart should meet.

[R32-A] BikeSmart system's signalling control function shall be activated by voice recognition.
[R35-A] BikeSmart system's speed counter shall be activated by voice recognition.
[R39-A] The voice recognition microphone shall have noise cancelation.
[R41-A] The voice recognition microphone shall pick up user command in both quiet and noisy environment.
[R42-A] The voice recognition system shall recognize any person's voice.
[R43-A] The voice recognition system shall recognize any person's accent.
[R44-A] The voice recognition system shall not recognize words that are not in the command library.

5.2 Software Implementation

The Arduino library we used to implement the voice recognition design is called µSpeech. µSpeech is simple and limited, yet it provides all the necessary features that are needed for the BikeSmart system. The basic functions of µSpeech allow users to design a set of vocabularies that can be recognized by voice. We can assign these vocabularies to certain actions. It does not require training to recognize words.

5.2.1 Noise Cancellation

Before we start saying any words into the microphone to perform actions, we must filter out the noise. Noise can be cumbersome because it can interfere with the words that are spoken into the microphone. As a result, the wrong sounds are picked up by the microcontroller and the correct actions will never be called. Luckily, µSpeech has a feature that can set a base volume and suppress most of the noise.

By sending a calibration command to µSpeech, the microphone will start listening and a series of numbers will be output to the screen. These numbers are the current voice level. For example, if the numbers are around 1000, then we can set the volume to 1000 as the minimum volume that will be picked up. This way, everything under 1000 will be considered as noise and will not affect the words we speak into the microphone.

For this project, we set the volume to 1500. This value is rather high because we only want the microphone to pick up sound that is very close to it which is our speaking voice. We want to make sure everything else from further away is suppressed.

5.2.2 Phoneme Calibration

The next step is to calibrate the sounds of the syllables also known as phonemes. These phonemes are the sounds that will be picked up by the microcontroller. However, not every sound can be recognized. In the uSpeech library, the following sounds can be recognized:

Phoneme	Sound
f	/f sound
e	/e and /i sound
a	/a, /o, /r, /l sound
v	/z, /v, /w sound
sh	/sh, /ch, /s sound
‘ ‘	No sound

Table 2: Phoneme Chart

For each of the sound to be recognized, we have to calibrate accordingly. Similar to the calibrating the volume level, we send a calibration command to uSpeech and then we speak the phonemes into the microphone. Making a sound will return a certain voice level and the numbers will be output to the screen. For example, saying ‘sh’ may return consistent values from 0 to 2, and saying ‘f’ may return values from 5 to 7. In this case, we set the value of ‘sh’ to 3 and the value of ‘f’ to 8 (the highest returned number + 1). There might be inconsistency in the calibration and random spike number may occur. We do not have worry about those since those numbers are not returned consistently.

For this project, we have ‘f’ set to 400, ‘e’ set to 1, ‘a’ set to 2, ‘v’ set to 3, and ‘sh’ set to 4. These numbers may change later for more accurate calibration.

5.2.3 Voice Recognition

After the calibration is completed, we can start writing the code to generate the desired actions. There are 3 key words that we are looking for: left, right, and stop. As mentioned above, uSpeech only recognize a few sounds. Therefore, to craft these words, we must compare the sounds that are available to the sounds of these words.

For left, we will try to listen for the ‘e’ sound first. Once ‘e’ is detected, then we move on to see if ‘f’ is heard next. If the sequence is correct, we will assume left is said.

For right, we don't have many choices here since 'r' and 'ght' sounds are not in the uSpeech library. In this case, we will simply detect the 'i' sound which in phoneme is 'e'.

For stop, first we will look at 's' sound then follow by 'o' sound which in phoneme translates into 'sh' and 'a'.

This algorithm will be put into a loop such that uSpeech is always listening. Once the action words are detected, the microcontroller will send the message out to turn on the LED signals.

5.3 Hardware Implementation

Figure 1 shows a basic Arduino microcontroller implementation of our voice control system. The whole circuit contains a breakout board for electret microphone, a Arduino UNO microcontroller, three 220Ω along with a 570nm LED (Left Signal), a 633nm LED (Stop Signal), and a 525nm LED (Right Signal).

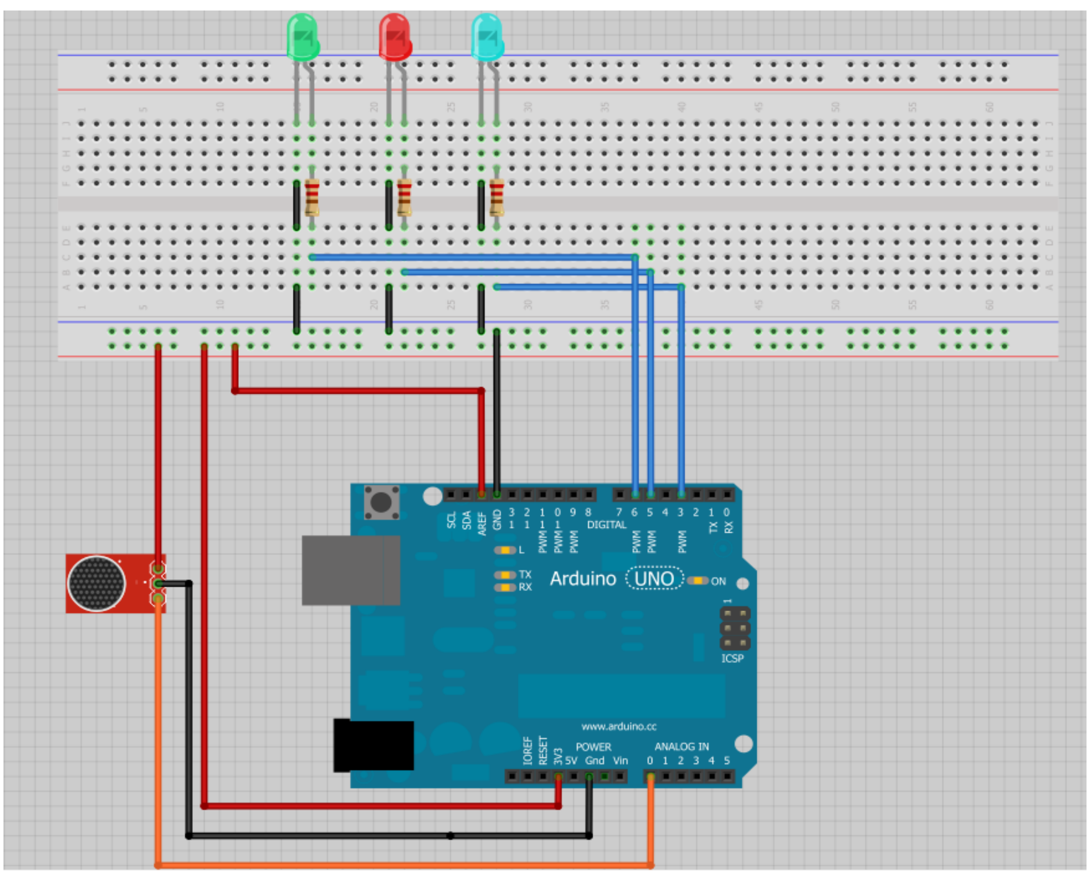


Figure 14: Breadboard diagram of the basic Voice Recognition System

5.3.1 Signal Lights

The three signal LEDs are connected to pulse-width modulation pins 3, 5 and 6 respectively in order to perform fading and flashing effect under proper programming. By using these pins, the digital control will create a series of square wave, in other words, a signal switched between on and off. This on and off pattern will then simulate voltage in between full on, which in our case is 3.3 volts, and off by changing the portion of the time the signal spends on versus the time that the signal spends off. If the on and off pattern is repeated fast enough with our signal LEDs, the result is as if the signal is a steady voltage between 0 volts and 3.3 volts controlling the brightness of the LEDs. This effect can be used on bicycle brake. When pressing the brake handle on the bicycle, the red stop signal LED will turn brighter, which simulates the brake of car.

5.3.2 Battery

The microcontroller board contains a Atmel 8-bit AVR RISC-based microcontroller element (ATMEGA328), a voltage regulator and a Reset button. The Li-on battery voltage is fed through the power pin in the Arduino and regulated inside the board. Note that all the circuit elements are powered using this regulated voltage. Otherwise the analog readings from the breakout board for Electret microphone would be unpredictable as the battery is drained through time.

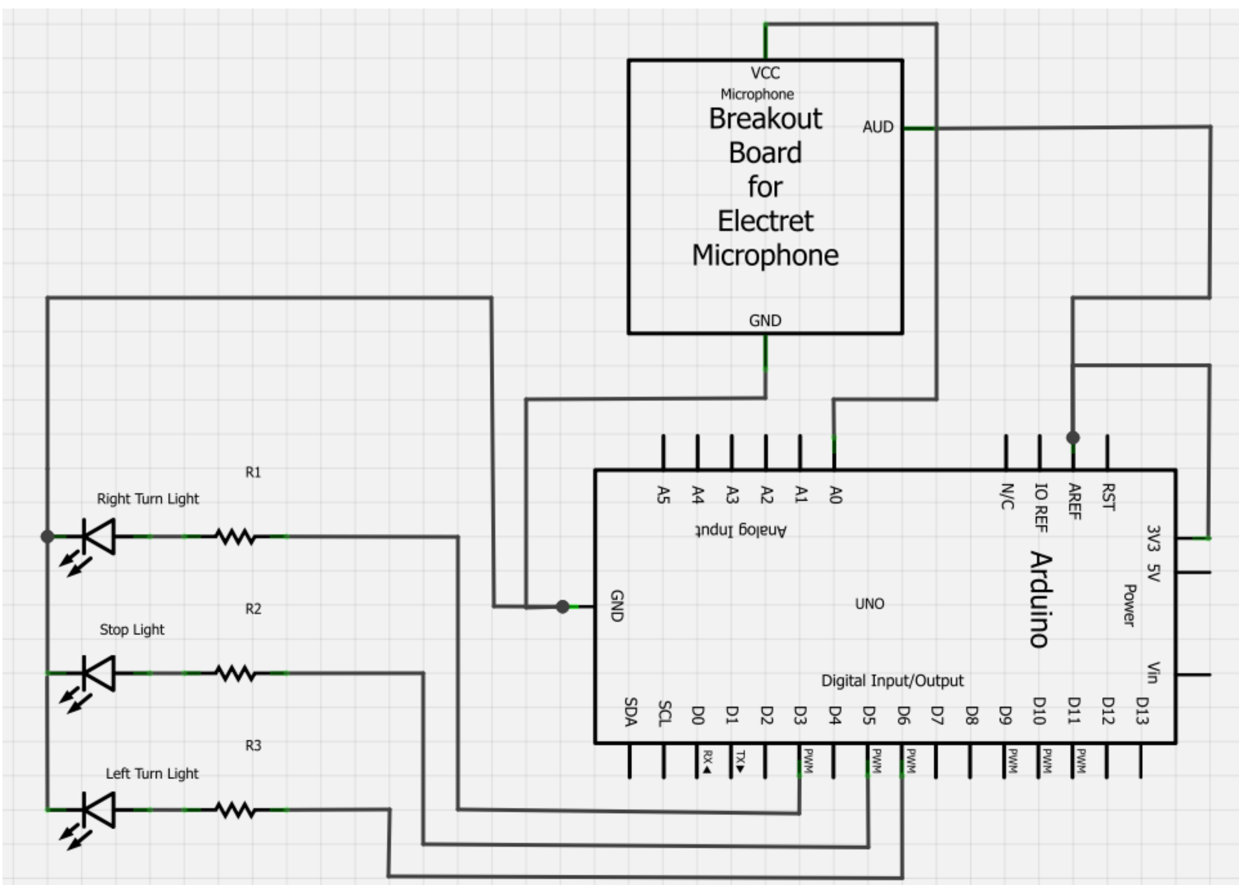


Figure 15: Schematic diagram of basic Voice Recognition System

5.3.3 Microphone

Initially we were using a bare Electret microphone which required us to construct an amplification circuit. By doing so, it could take up some space on our prototyping breadboard, so we have decided to use the breakout board, shown in Figure 7, instead of a bare microphone. This breakout board couples a small Electret microphone with a 100 times op-amp to amplify the sounds of the voice loud enough to be picked up by the microcontroller's analog to digital converter, which in our case is pin A0, shown in figure 5 and figure 6.

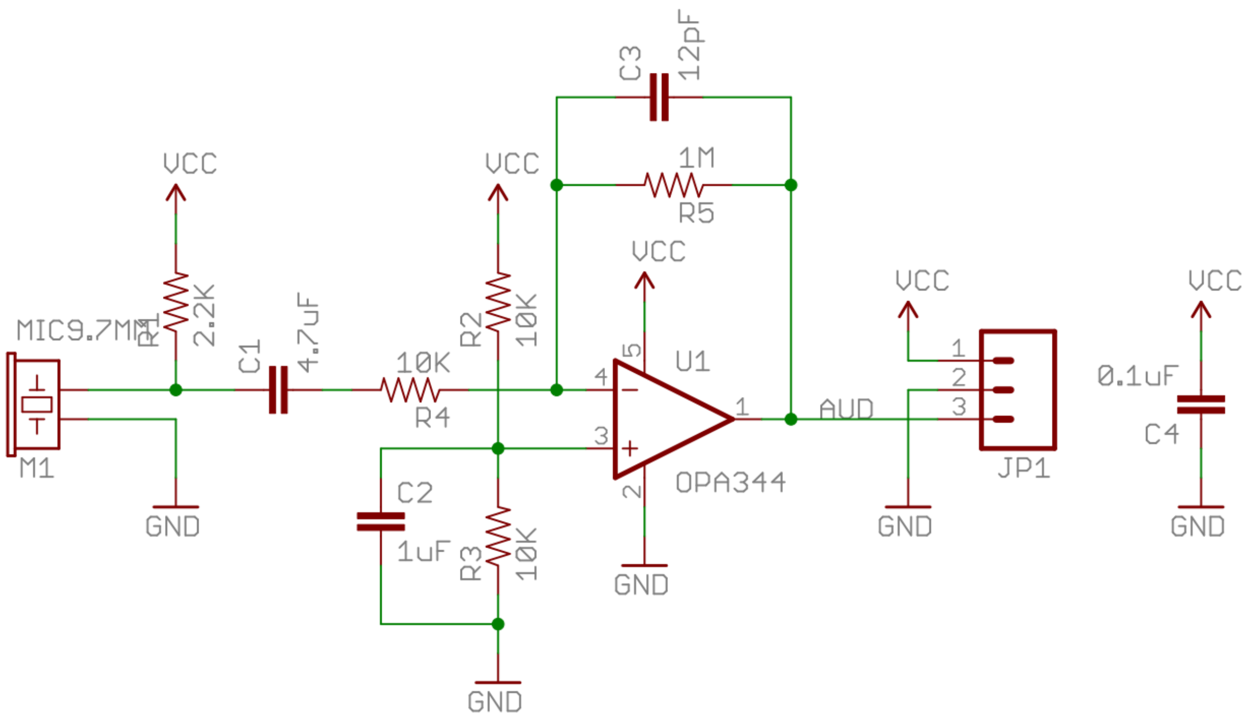


Figure 16: Schematic diagram of breakout board for Electret Microphone

5.3.4 A/D converter

The ATMEGA328 used in the project contains 6 channel A/D converter and in our basic voice control system, we are using pin zero converter. The converter has 10 bit resolution, returning integers from 0 to 1023. While the main function of the analog pins is to read analog sensor, the analog pins also have all the functionality of general purpose input and output pins, which functions same as digital pins 0 to 13.

6.0 Wireless Communication Design

6.1 Overview

Considering our cyclist customers are very motivate and energetic, we want them to enjoy our products through wireless communication design. We are going to add XBee S2 2mW Zigbee models in our product in order to achieve this goal.

Figure 17: XBee Sender Unit

The XBee is engineered to meet IEEE 802.15.4 standards and support the unique needs of low-cost, low-power wireless sensor networks. These modules also operate within the ISM 2.4 GHz frequency band and are pin-for-pin compatible with each other.

6.2 Specification

Indoor/Urban	40m
Outdoor line-of-sight	120m
Transmit Power	2mW (+3dBm)
Receiver Sensitivity	-93 dBm
TX peak Current	40 mA(@3.3V)
RX Current	40 mA(@3.3V)
Power-down Current	< 1 µA
Operating Frequency	ISM 2.4GHz
Supported Network Topologies	Point-to-point, Point-to-multipoint, Peer-to-peer, and Mesh
Addressing Options	PAN ID and Addresses, Cluster IDs and Endpoints (Optional)

Table 3: XBee Specification

As the XBee module cannot be directly connect with Arduino Uno R3, we need to attach our XBee module on the UartSbee V4.

Figure 18: UartSbee Module

UartSbee V4 is a compact USB to serial adapter equipped with BEE socket (20 pin 2.0mm). The integrated FT232RL can be used for programming or communicating with MCUs.

FT232RL Breakout Connector	3-pin (6x1), 2.54mm(0.1”) pitch, right-angle male header
USB Connector	5-pin Mini-B USB
USB 5V Resettable Fuse (PPTC)	Yes
BEE Connector	20-pins (10x2), 2.00mm pitch, vertical female header
BEE Breakout	20-pin (10x2), 2.54mm(0.1”) pitch
BEE Reset Button	Yes, right-angle

Table 4: UartSbee Specifications

6.3 Hardware

Here is the system data flow diagram in a UART-interfaced environment. The figure 19 clearly explains how one XBee module communicates with other XBee module. Usually the microcontroller (Arduino Uno R3) will send the data (commands) through DI (data in) and CTS is its buffer. On the other hand the XBee module will translate or transfer the commands from another XBee module to Microcontroller and RTS is its buffer.

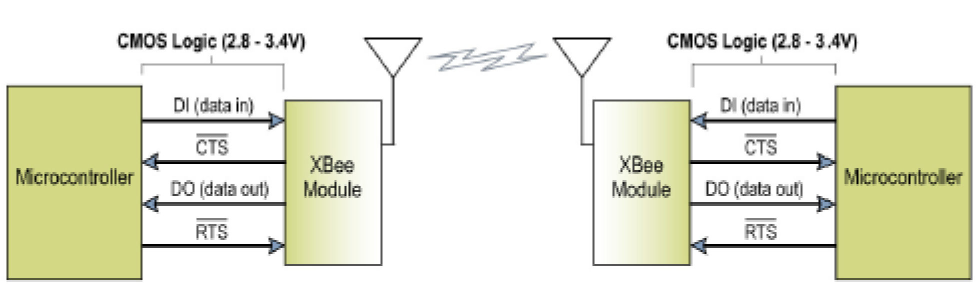


Figure 19: System data flow diagram in a UART-interfaced environment

In the hardware design, the first thing we need to do is how many XBee modules we need for our project. Here is the way how we determine how many XBee modules we want. As XBee module has Peer-to-Peer mode, we decrease the numbers of XBee modules from six to four.

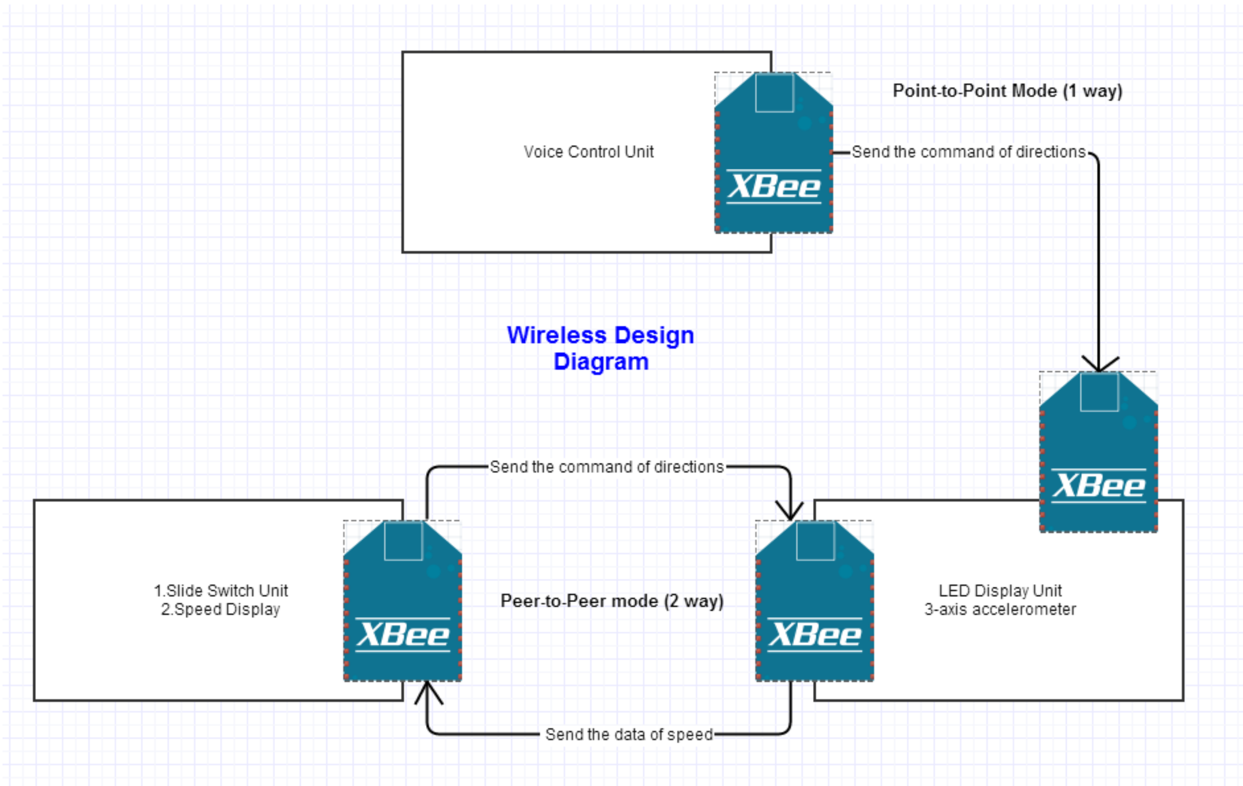


Figure 20: Wireless Design Diagram

6.4 Software

As we mention before, we need to attach the XBee module to UartSbee V4 for programming it configuration. We use the software, X-CTU, to program our XBee module. First we need to find the Serial Number High and Serial Number Low at the back of the module, record them down for late use. We also want to edit the ID-PAD and Node Identifier of those modules then we can recognize them easier. For example, in our Point-to-Point configuration, we set one of the XBee module as Coordinator (A) and the other one is Router (B). And we set Coordinator (A)'s DH and DL equals to Router (B)'s SH and SL via versa. The figure below it will make it clearer.

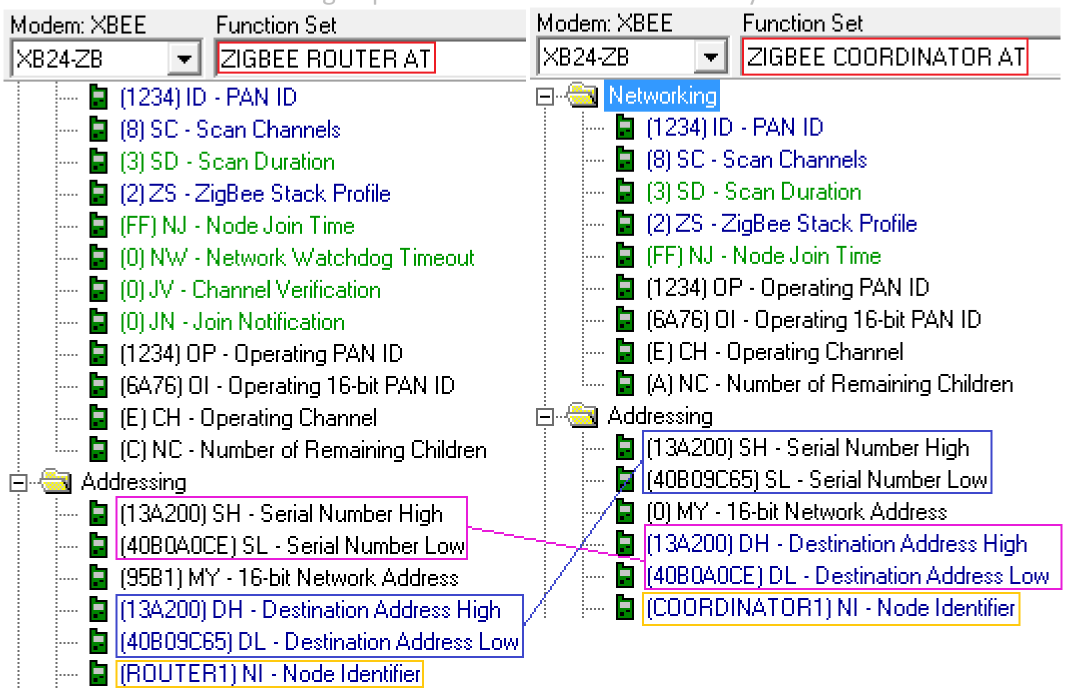


Figure 21: Serial Number and Destination Address Pairs

After those setup, we now can test those two modules can actually communicate each other.

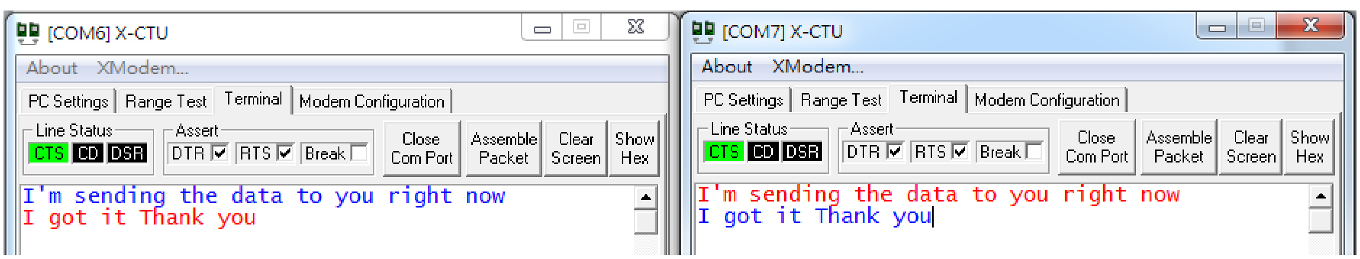


Figure 22: Demostration of the Communcation between Modules

The blue command is the sender and the read command is the receiver.

7.0 Power Supply

7.1 Overview

As the Arduino Uno board is used in the proof of concept model, in order to power all the individual subsystem of the BikeSmart system, a constant 5V input voltage must be supplied either via USB connector or an external power supply (through the V_{in} pin, or a 2.1mm center-positive plug).

7.2 Battery

The E585460 Li-Ion rechargeable battery is used to power the system. According to table 1, while the battery only provides a 3.7V voltage, a step-up regulator is needed to match the required input voltage of the Arduino Uno. We decided to use the LiPower Shield, shown in Figure 2. It connects a 3.7V Li-Ion battery through a JST connector and boost the voltage up to 5V which then connects to the Arduino board's 5V pin. The on-board MAX17043G+U IC is connected to the I2C lines (pin A4 and A5 on Arduino Uno) so that the BikeSmart System can monitor its own power supply. It is also able to charge the battery, as the board has the MCP73831/2 IC. The MCP73831/2 IC is essentially a battery charger that provides a 500mA charging current. The on-board mini-USB port allows a USB power source to charge the battery. With a 500 mA charging current and a 2000 mAh battery capacity, it is clear that a 4 hour of charging time is required to fully charge the battery.

Battery Model	Nominal Capacity	Cell Voltage	Discharge cut-off Voltage	Cycle Life
E585460	2000 mAh	3.7V	2.75V	300-500

Table 5: E585460 Li-Ion Battery Specification

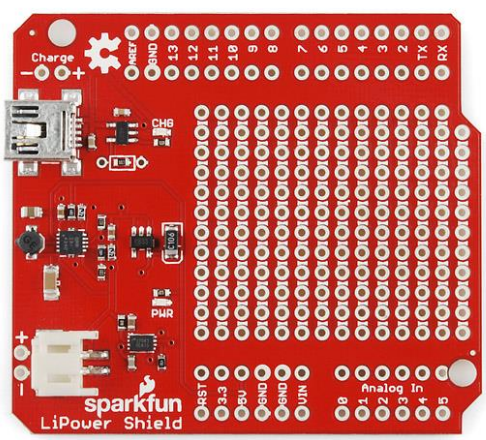


Figure 23: LiPower Shield

8.0 Hardware Packaging

8.1 Overview

Any prototype should come with a good packaging design. For our product we decided to use the most innovative technique of packaging, 3D printer. The most famous of 3D printers nowadays in the industry is MakerBot printer [2]. They are many benefits over using MakerBot compare to other 3D printers such as its low cost material that it is using over other printers. In this project, we will use MakerBot Replicator (model 2X) for fabrication of all designs in creating case for LED lights and slide switch control. The 2X model is intended to work in a build envelope (285 mm x 153 mm) and has a printing rate of 100 µm per layer. It has an enclosed build area, dual-extruder support, and the ability to print with ABS plastic, PLA, Flexible, and Dissolvable filaments [2]. The most common filaments are ABS plastic or PLA natural filaments. A PLA filament is a renewable bio-plastic made from corn. ABS filament (acrylonitrile butadiene styrene) is a common thermoplastic. The ABS filament will be used to obtain all results presented in this paper [2]. The figure below shows the MakerBot that we are using for our designs.

According to the functional specification requirements, the design of the hardware case must follow each requirement.

[R8-A] The design of controller box on the handle bar must be small in size.
[R9-A] ABS filament shall be used via 3D printer for the whole system casing.

Figure 24: MakerBot Replicator 2X that is used in Micro-instrumentation Laboratory at SFU

8.2 Software

In order to create designs using MakerBot, we have to use a CAD software that is able to accommodate with MakerBot. The designs of the case for slide switch control unit on the handle bar and case for LED panel under the seat of bicycle will be created in using a CAD program called SketchUp [1]. SketchUp is a 3D modeling program for architects, civil and mechanical engineers, film and video game makers, but has also been adapted for use in 3D printing via, eg, MakerBot. After we created the design on SketchUp, the SketchUp file in .skp format will be converted into 3D models in format of .STL or .obj. After designing each component in SketchUp, the design was exported to the visual platform of MakerBot: the MakerWare BETA software [3], in file format of ".thing".

The scaling and positioning of the design was can be configured in MakerWare software before printing. Figure 23 shows the MakerWare environment which will allow users to easily move, turn, and scale the object on MakerBot printer.

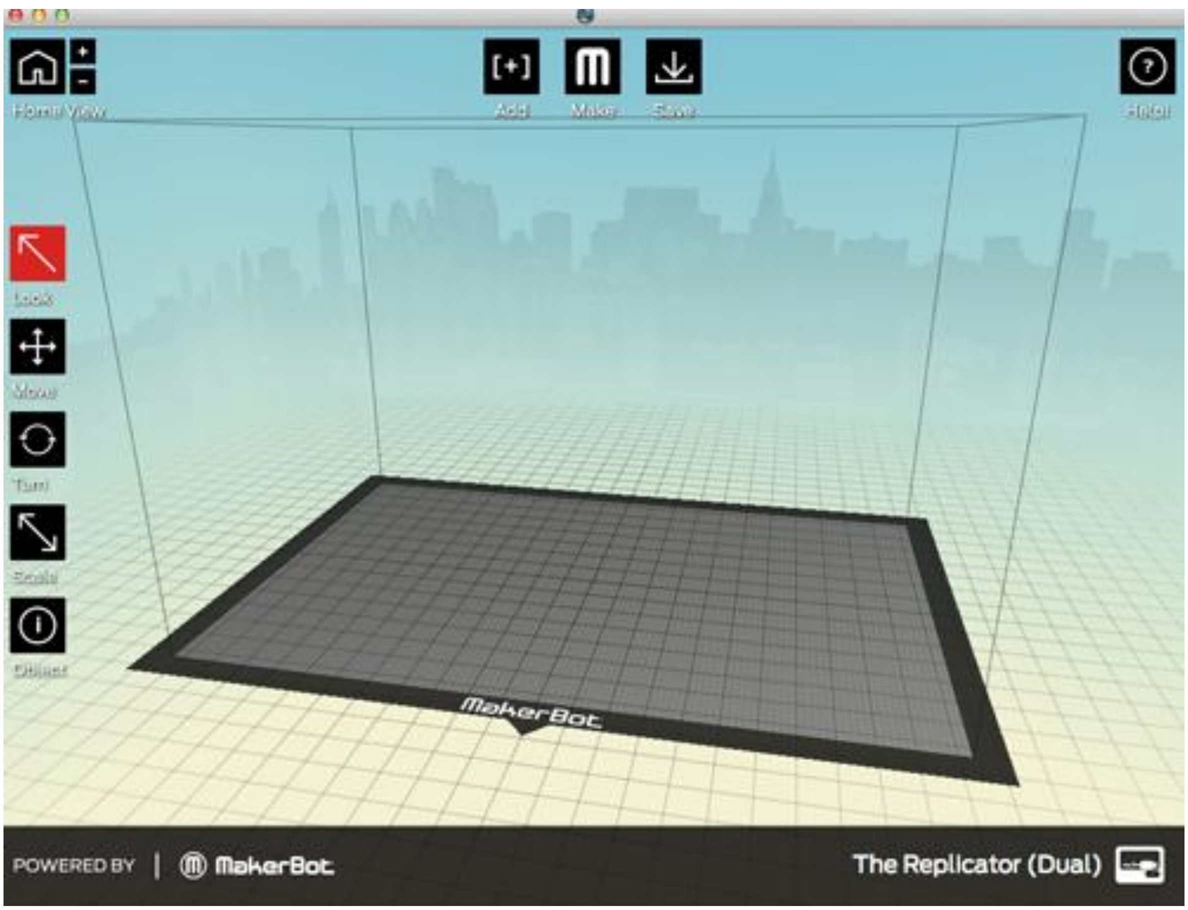


Figure 25: MakerWare software environment

8.3 Case Design of Slide switch Control

According to our design of slide switch control, we design a rectangle with a length of 10cm, width 6cm, and height 5cm on SketchUp software. Our design on SketchUp is shown in Figure 24; please note that the measurement and final packaging might be different. On the slide switch control box, we will have the slide switch control unit and the speed display 7-segment.

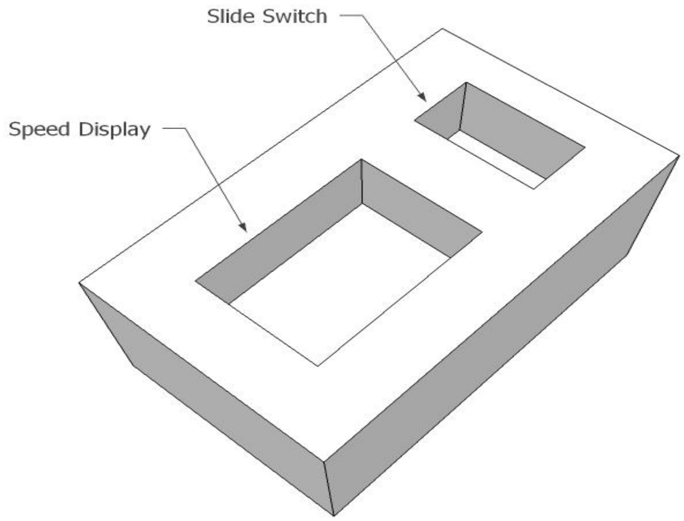


Figure 26: SketchUp design of slide switch control box

8.4 Case Design of LED light

For the case design of LED light, we have two different designs: one is with only one 8x8 LED matrix and another with two 8x8 LED matrix. Figure 25 shows the design of LED light with only one matrix (8x8). This design will have length and width of 9.50cm and height of 6cm. Figure 22 shows the same design but with two LED light being used. The two 8x8 matrices will be placed next to each other. Half of each matrix will be used for the brake signal and the other half of them will be used to show arrows to Left or Right side. This design will have length of 12 cm, width 8cm, and height of 6cm. The LED light will be placed on the empty spot shown in Figure 25 and 26, and Arduino with other related hardware parts will be placed inside the cases. Please note that final packaging might have different measurement and these are just estimations.

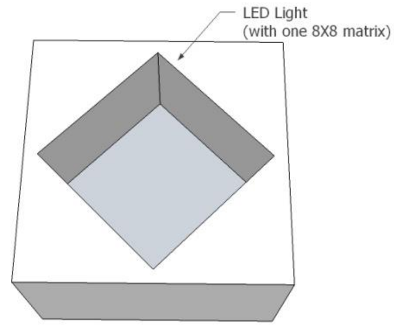


Figure 27: SketchUp design of the case for LED light with only one 8x8 matrix

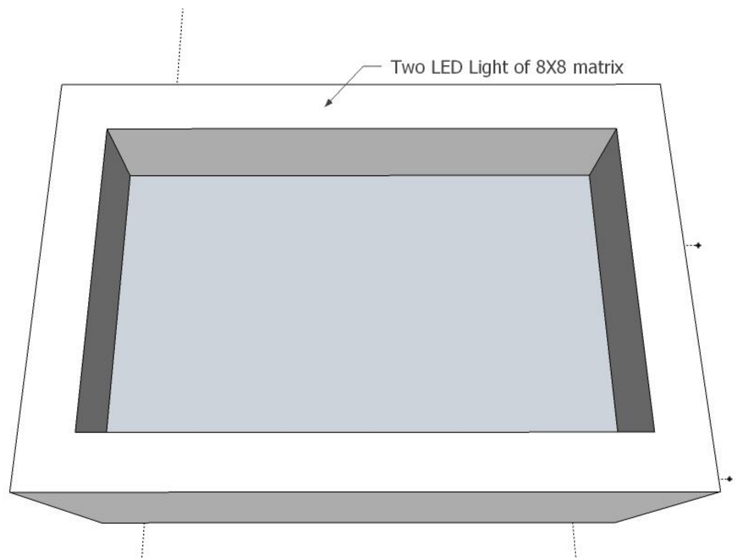


Figure 28: SketchUp design of the case for LED light with two 8x8 matrices

9.0 System Test Plan

The LED light, slide switch control, and voice recognition will be tested after all individual part have been successfully tested, the hardware packaging which consist of case for LED and slide switch control will also be tested.

Test 1: LED visibility

User input: User turns on the matrix of LEDs on the bicycle.
Conditions: User displays the bicycle in a 100 meter of distance while the LEDs are flashing.
Expected observations: Under any kind of weather, the LEDs should be visible from other road user 100 meter behind the bicycle.

Test 2: Noise consideration

User input: Power up the voice recognition system to detect the key words from cyclist.

Conditions: Let the system to detect the key words while riding down hill with certain speed to allow wind to create some noise.

Expected observations: The system should be able to detect the key words from user without any problem.

Test 3: Waterproof

User input: Power up the voice recognition system to detect the key words from cyclist.

Conditions: Rainy Day.

Expected observations: The system should be able to run properly under rainy condition with rain drops falling on sealed devices.

Test 4: Communication

User input: User says keyword “left”, “right” and also pressing the brake handle.

Conditions: Any.

Expected observations: The system should be display the correct signal light corresponding to the input keyword and movement.

Test 5: Shock absorbing

User input: Power up the voice recognition system to detect the key words from cyclist.

Conditions: User riding the bicycle on the rough road in order to create some shocks.

Expected observations: The wire connecting the device should be firmly attached and no signal should be lost.

Test 6: Case Strength

Task: case strength measurement

1. The casing is water-resistant
2. Switches are comfortable to use
3. Case is small and not space consuming
4. Visible color to use it at night

Expected Observation: The case of slide switch control and LED lights do not get wet under a rainy weather. The color of case can be chose according to customer desire.

10.0 Conclusion

The design specifications outlined in this document clearly define the technical requirements and preliminary design standards toward the implementation of the BikeSmart System proof of concept model. Based on the functional specification document, the design specifications document will serve as a guideline when implementing all function requirements. The test plan will help to ensure that the design specifications are met and that the model is in working order.

At the end of the proof of concept development scheduled for April 2014, we hope that the model will serve as an active tool to improve cyclists' safety and riding experience.

11.0 Acknowledgement

We would like to express our deepest appreciation to Dr. Bonnie Gray, who gives us permission to work in Micro-instrumentation Laboratory at SFU and use the MakerBot 3D printer.

12.0 References

[1]. “SketchUp: SketchUp Pro”. Last updated: March 30, 2014. [Online].
Available: http://www.sketchup.com/[Accessed on March. 01, 2014].

[2]. “MakerBot: Replicator 2X”. [Online]. Available: http://www.makerbot.com/. [Accessed on March.03, 2014].

[3]. “MakerBot-MakerWare”. [Online]. Available: http://www.makerbot.com/makerware. [Accessed on March. 03, 2014].

[4]. "Cycling accidents". Retrieved January, 2013 Available: http://www.rospa.com/roadsafety/advice/cycling/info/cycling_accidents_factsheet.pdf

[5]. "Reported road casualties Great Britain: annual report 2012". Retrieved January, 2014 Available: https://www.gov.uk/government/publications/reported-road-casualties-great-britain-annual-report-2012

[6]. "Lithium-Ion Battery Maintenance Guidelines". Retrieved February, 2014 Available: http://canada.newark.com/pdfs/techarticles/tektronix/LIBMG.pdf

[7]. "MOTOR VEHICLE ACT ". Retrieved February, 2013 Available: http://www.bclaws.ca/EPLibraries/bclaws_new/document/LOC/freeside/--%20M%20--/46_Motor%20Vehicle%20Act%20RSBC%201996%20c.%20318/00_Act/96318_05.xml#section184

[8]. "STREET AND TRAFFIC BY-LAW NO. 2849". Retrieved February, 2014 Available: http://former.vancouver.ca/bylaws/2849c.PDF#page=34

[9]. Loewen , S , "A Refresher on Cycling Laws". Retrieved January , 2013 Available: http://www.travelsmart.ca/en/Life-and-Home/Bike-More/A-Refresher-on-Cycling-Laws.aspx

[10]. "Preliminary Technical Data ADXL345". Retrieved March , 2014 Available: http://students.cs.byu.edu/~cs124ta/references/RBX430-1/peripherals/Accelerometer%20-%20ADXL345%20Rev.%20PrA.pdf

[11]. "Miniature Single-Cell, Fully Integrated Li-Ion, Li-Polymer Charge Management Controllers". Retrieved March , 2014 Available: https://www.sparkfun.com/datasheets/Prototyping/Batteries/MCP73831T.pdf

[12]. Katherine. "How to Plug LED Matrix into Colorduino / Colors Shield". Retrieved March, 2014
Available: http://blog.iteadstudio.com/how-to-plug-led-matrix-into-colorduino-colors-shield/

[13]. Anian Brosig. "Colorduino". Retrieved February, 2014
Available: http://www.katalytischerkonverter.com/hello/electronic/arduino-projects/colorduino.html

[14]. Lincomatic. "Colorduino Library for Arduino". Retrieved February, 2014
Available:
http://blog.lincomatic.com/?p=148

[15]. *ANALOG DEVICES*. "Colorduino Library for Arduino". Retrieved February, 2014
Available: www.analog.com

[16]. Digi. "Xbee/Xbee-PRO ZB RF Modules". Retrieved February, 2014
Available: www.digi.com

[17].Cooper Maa. "Arduino and Xbee Series2". Retrieved February, 2014
Available:
http://coopermaa2nd.blogspot.tw/2011/08/arduino-and-xbee-series-2.html

[18]. Ober. "Xbee Series 2 Point to Point Communication". Retrieved February, 2014
Available:
http://tutorial.cytron.com.my/2012/03/08/xbee-series-2-point-to-point-communication/

[19]. DreamRide Corporation , " Function Specification for the BikeSmart System,".

Appendix A

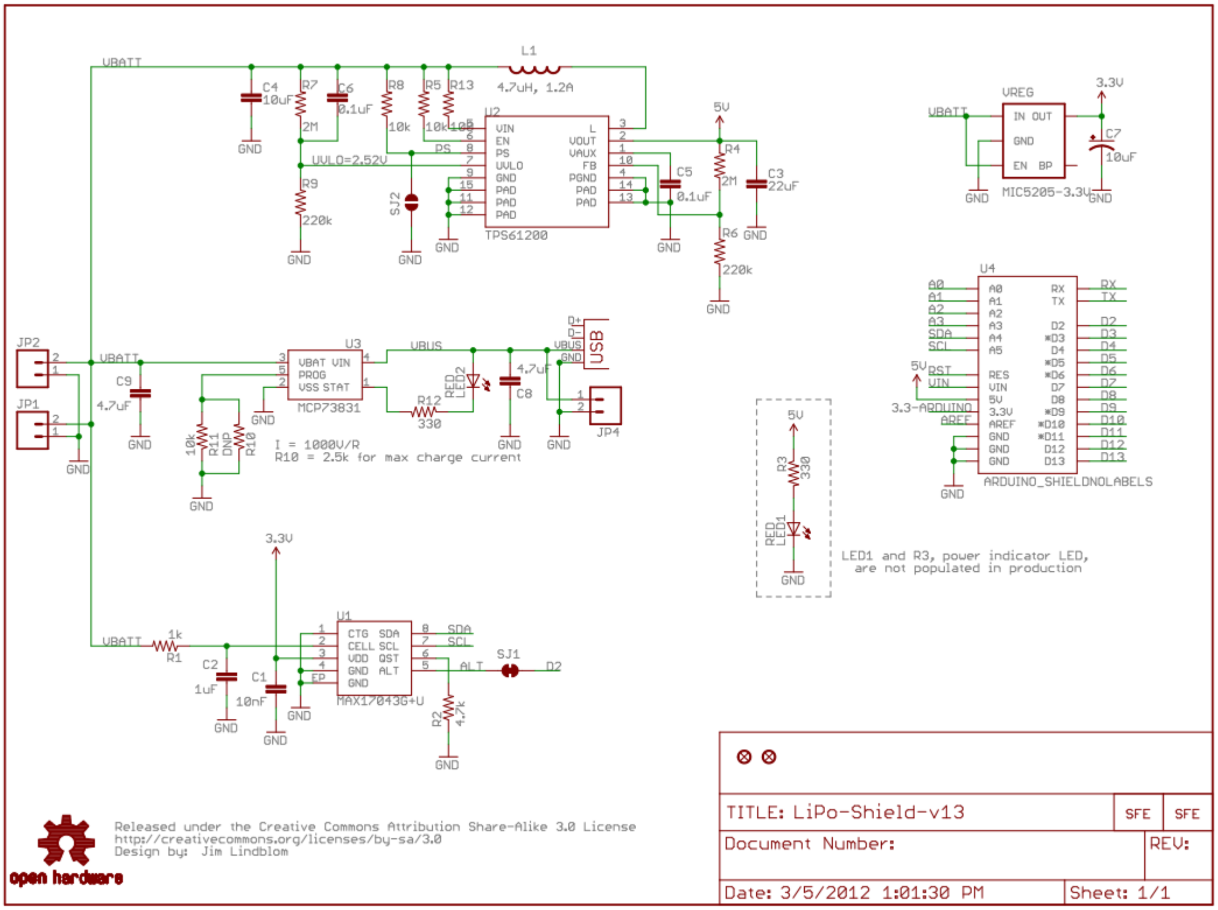


Figure 29: LiPower Shield Schematic