

University of Waterloo  
Faculty of Engineering  
Department of Electrical and Computer Engineering

StreetSmart: Rider's Enhanced Awareness Device  
Final Report

Group 2024.65

Yixing Qie (20828359)  
Patricia Liu (20824476)  
Anees Aissaoui (20828221)  
Kenneth Hung (20836537)

Consultant: Prof. William D. Bishop

February 26, 2024

# Abstract

According to the CAA, road safety rules are disrespected in approximately 1 of 3 cycling deaths, with over 70% of incidents involving motorized vehicles<sup>[1]</sup>. These numbers are expected to rise as the popularity of electric bicycles continues to grow. We aim to design a safety assistance system that increases a rider's awareness and protects them from preventable accidents. Additionally, this device raises a motorized vehicle driver's awareness of the cyclists that share the road. StreetSmart is a two-device safety system that cyclists mount onto their bicycle. Equipped with an accelerometer to detect deceleration, StreetSmart automatically flashes its brake lights to alert drivers of a rider's intentions to brake. The central device uses Light Detection and Ranging (LIDAR) for vehicle detection and alerts riders of rear collision risks through haptic feedback on a second, smaller device. The second device also features a switch for the rider to indicate turn signals, and communicates to the central device wirelessly. A significant advantage of StreetSmart over other alternatives is its ability to alert everyone on the road, including the user, of potential dangers.

# Acknowledgements

We want to take a moment to thank our consultant Dr. William Bishop, Director of Admissions and Continuing Lecturer at the University of Waterloo. Dr. Bishop provided invaluable insights throughout all stages of the design of StreetSmart, and also served as our professor for ECE 224, where we learned the foundations of embedded systems used in this project.

# Table of Contents

<b>Abstract</b>	<b>1</b>
<b>Acknowledgements</b>	<b>1</b>
<b>Table of Contents</b>	<b>2</b>
<b>List of Figures</b>	<b>4</b>
<b>List of Tables</b>	<b>5</b>
<b>1. High-Level Description of the Project</b>	<b>6</b>
1.1 Motivation	6
1.2 Project Objective	6
1.3 Block Diagram	6
<b>2. Project Specifications</b>	<b>8</b>
2.1 Functional Specifications	8
2.2 Non-Functional Specifications	9
<b>3. Detailed Design</b>	<b>10</b>
3.1 Central Controller	10
3.1.1 Mechanical Enclosure	10
3.1.2 Central Controller PCB	10
3.1.3 Software Subsystem Design	11
3.1.3.1 Braking Detection Algorithm	11
3.1.3.2 Distance Detection Algorithm	12
3.1.3.3 LED Pattern Algorithm	13
3.1.3.4 Wireless Communication Algorithm	13
3.1.4 Controller Subsystem	15
3.1.5 Power Subsystem	15
3.1.6 Lighting Subsystem	17
3.1.7 Sensor Subsystem	17
3.1.7.1 Lidar	17
3.1.7.2 Accelerometer	17
3.1.8 Communication Subsystem	18
3.2 Wireless Haptic Feedback Controller	18
3.2.1 Mechanical Enclosure	18
3.2.2 Wireless Haptic Feedback Controller PCB	19
3.2.3 Software Subsystem Design	19
3.2.3.1 Haptic Algorithm	19
3.2.3.2 Communication Algorithm	19

3.2.4 Controller Subsystem	20
3.2.5 Power Subsystem	20
3.2.4 Haptic Subsystem	22
3.2.5 Communication Subsystem	22
<b>4. Prototype Data</b>	<b>23</b>
4.1 Central Controller	23
4.1.1 Lidar Accuracy	23
4.1.2 Braking Detection Accuracy	23
4.1.3 Mechanical Enclosure	25
4.2 Wireless Haptic Feedback Controller	26
4.2.1 Mechanical Enclosure	26
4.2.2 Haptic Tuning	27
4.3 RF Communication Reliability	28
4.4 PCB Design	28
<b>5. Discussion and Conclusions</b>	<b>30</b>
5.1 Evaluation of Final Design	30
5.2 Use of Advanced Knowledge	30
5.3 Creativity, Novelty, Elegance	31
5.4 Quality of Risk Assessment	31
5.5 Student Workload	31
<b>References</b>	<b>32</b>
<b>Appendices</b>	<b>35</b>
Appendix A - Distance Detection Algorithm Alternative	35
Appendix B - Cameras	36
Appendix C - Buck Converter	37
Appendix D - Boost Converter	37
Appendix E - Previous Mechanical Designs	38

# List of Figures

Figure 1.3.1: Block diagram showing the Interconnectedness of Subsystems	7
Figure 3.1.1.1: Isometric View of Central Enclosure (with Top View) and Lid (with Bottom View)	10
Figure 3.1.3.1.1: Sample Python Code <sup>[12]</sup> Accessing Raw Measurements from the MPU-6050	
11	
Figure 3.1.3.2.1: Format of a Measurement Data Packet <sup>[24]</sup>	12
Figure 3.1.3.3.1: Sample Code <sup>[15]</sup> Showing how to Configure LEDs on the Raspberry Pi 4	13
Figure 3.1.3.4.1: Pseudocode <sup>[22]</sup> Outlining the Setup and Algorithm Required for the Host Side of the Wireless Communication	15
Figure 3.2.1.1: Isometric View of Haptic Enclosure (with Top View) and Lid (with Bottom View)	19
Figure 3.2.3.2.1: Pseudocode <sup>[22]</sup> for the algorithm on the Guest Side of the Wireless Communication	20
Figure 4.1.1.1: Lidar Distance Detection with Object at 2.9 m	23
Figure 4.1.2.1: 3-Plane Acceleration Measured while a Bicycle Brakes Regularly	24
Figure 4.1.2.2: Condition to Turn on Brake Lights	24
Figure 4.1.3.1: Prototype of Central Controller	25
Figure 4.1.3.2: Mechanical Mount in Different Pivot Positions	25
Figure 4.2.1.1: Prototype of Haptic Controller	26
Figure 4.2.1.2: Top and Bottom View of Handlebar Mount for Haptic Controller	27
Figure 4.2.1.3: Phone Mount Bracket on Haptic Controller with Various Test Devices	27
Figure 4.3.1: NRF24L01 Packet Loss over Channel Frequency	28
Figure 4.4.1: Initial Prototype of StreetSmart	29
Figure 4.4.2: Latest PCB Design for Electrical Components	29
Figure A: The relationship between depth and disparity <sup>[5]</sup>	36
Figure C: 5V Output Buck Converter using the LM2596S Switching Regulator	37
Figure D: 5V Output Boost Converter using the XL6009 Switching Regulator	38
Figure E.1: External Schematic of Central Controller	38
Figure E.2: Internal Schematic with Hardware Components	38
Figure E.3: External Schematic of Haptic Controller	38
Figure E.4: Internal Schematic with Hardware Components	38

# List of Tables

Table 2.1.1: List of Essential Functional Specifications for StreetSmart	8
Table 2.1.2: List of Non-Essential Functional Specifications for StreetSmart	8
Table 2.2.1: List of Essential Non-Functional Specifications for StreetSmart	9
Table 2.2.2: List of Non-Essential Non-Functional Specifications for StreetSmart	9
Table 3.1.5.1: Total Power Consumption Analysis for the Central Controller's Power Subsystem	16
Table 3.1.5.2: Battery Requirement Analysis for the Central Controller's Power Subsystem	16
Table 3.2.5.1: Total Power Consumption Analysis for the Haptic Controller's Power Subsystem	21
Table 3.2.5.2: Battery Requirement Analysis for the Haptic Controller's Power Subsystem	21
Table 4.1.3.3: Central and Haptic Controller Water Testing Results	26
Table 5.5.1: Student Contribution to StreetSmart as a Percentage	32

# 1. High-Level Description of the Project

This section describes the motivation and objectives of our project, as well as provides a block diagram of StreetSmart's components.

## 1.1 Motivation

Bicycles have been a common mode of transportation for decades, offering a convenient, low-cost, and environmentally friendly alternative. However, cyclists are often at risk due to the lack of consistency and enforcement of bicycle laws, along with their smaller frame. Unfortunately, the most common hazards they encounter are motorized vehicles. On average, over 70% of cyclist fatalities involve a collision with a motor vehicle<sup>[1]</sup>. Furthermore, numerous studies have shown that motorists are more likely to be at fault in a bike-car incident, with the most common accident being a driver striking a cyclist from behind<sup>[2]</sup>.

One of the main factors for these incidents is a lack of awareness. Researchers have discovered that drivers frequently fail to notice cyclists due to “inattentional blindness”<sup>[4]</sup>. When drivers see cyclists, they don’t register their presence and behave recklessly<sup>[4]</sup>. In fact, according to the CAA, road safety rules are disrespected in approximately 1 of 3 cycling deaths<sup>[1]</sup>. Furthermore, cyclists themselves lack the devices to provide a consistent flow of environmental information. As a result, there is a need for a device that increases awareness to reduce the risk of incidents and provide peace of mind.

## 1.2 Project Objective

The objective of this project is to design a safety assistance system that increases rider awareness and prevents avoidable accidents. Additionally, the system aims to increase the awareness of motorized vehicle drivers who share the road with bicycle riders.

## 1.3 Block Diagram

This section depicts StreetSmart’s system design consisting of 2 devices: the Central Controller and the Wireless Haptic Feedback Controller. Each device can be broken down into subsystems and visualized in Figure 1.3.1. Items indicated with **[D]** will be fully designed in-house, and parts indicated with **[ND]** will not require designing.

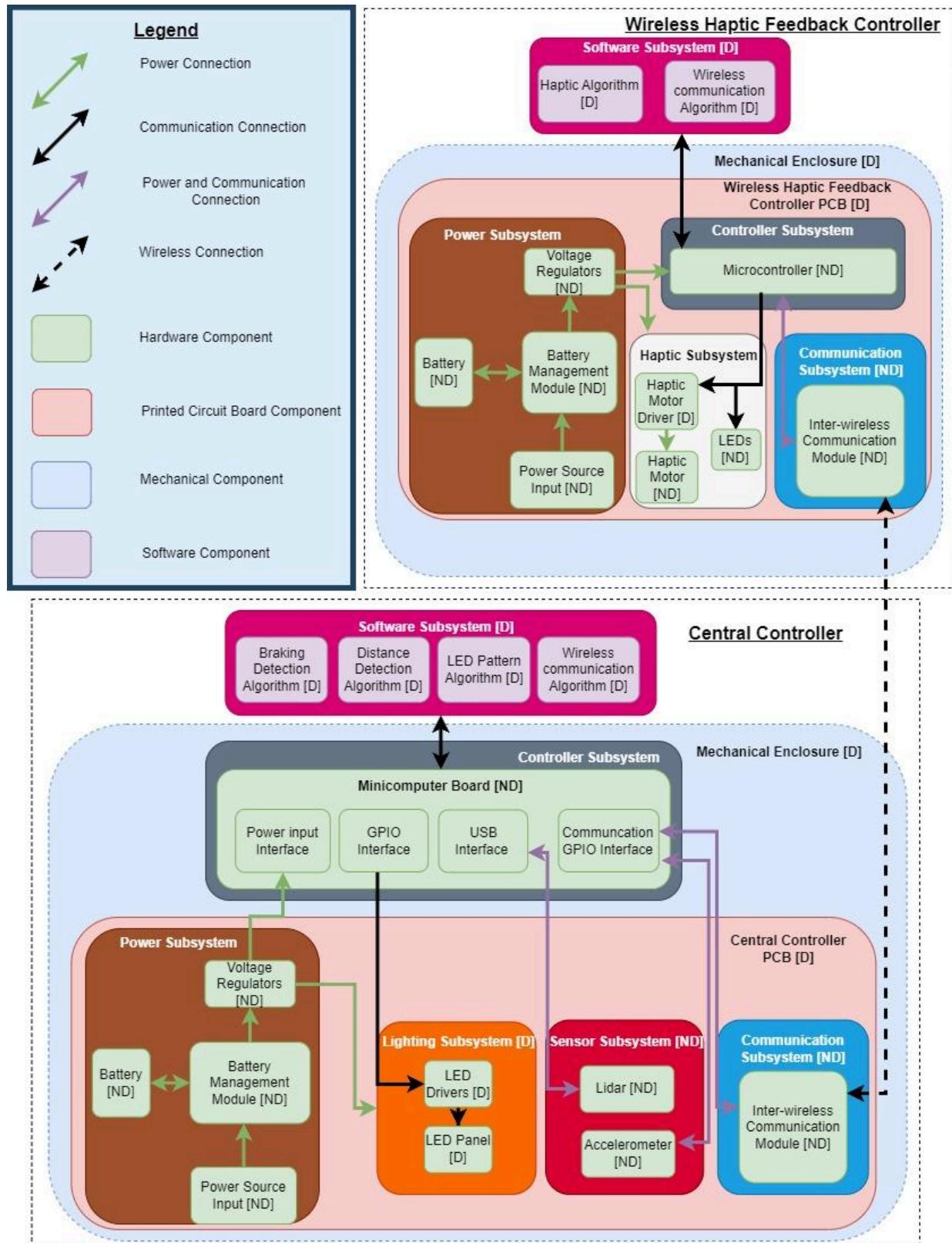


Figure 1.3.1: Block diagram showing the Interconnectedness of Subsystems

## 2. Project Specifications

This section describes the functional and non-functional specifications of the product in tables 2.1.1, 2.1.2, 2.2.1, and 2.2.2. Each table denotes whether the specifications are essential or non-essential in terms of delivery. A functional specification is defined as a feature the system must have, while a non-functional specification is defined as a general property of the system.

### 2.1 Functional Specifications

Table 2.1.1: List of Essential Functional Specifications for StreetSmart

Specification	Subsystem	Description
Acceleration/Deceleration Detection	Sensor	The device must be able to perform acceleration detection precisely for movement of up to $\pm 30 \text{ m/s}^2$ .
Rechargeable Power System	Power	Devices must have a built-in rechargeable power system.
Blindspot and Rear Distance Detection	Sensor	The device must be able to detect objects in the user's blind spot and rear with a field of view up to 90°.
Wireless Haptic Feedback Device	Haptic	The system must have a haptic feedback device for object detection with a power rating of $>=0.5 \text{ W}$ .
Braking/Turning Indicator Lights	Lighting	The device must have red LEDs to indicate turns and braking.
Inter-device Wireless Connectivity	Wireless	The system must be able to communicate with all of its subcomponents.
Turn Signal Buttons	Wireless User Haptic Feedback	The system should have a user input mechanism to indicate turn signals.

Table 2.1.2: List of Non-Essential Functional Specifications for StreetSmart

Specification	Subsystem	Description
Smartphone User Interface	Software	The system should be controllable with a smartphone user interface on Android.
Smartphone Wireless Connectivity	Communication	The system should be able to connect to a wireless interface such as a smartphone within a 4 m range.
Close Collision Automatic Video Recording	Software	The device should be able to record videos with lengths under 1 Gigabyte each in the event of a close collision.
GPS Path Mapping	Sensor	The device should have GPS tracking allowing for path mapping, and have a positional accuracy within 15 m of the user's location on a path.
Speed Tracking	Sensor	The device should have speed tracking for speeds up to 50 km/h with an accuracy of $\pm 5 \text{ km/h}$ .

## 2.2 Non-Functional Specifications

Table 2.2.1: List of Essential Non-Functional Specifications for StreetSmart

Specification	Subsystem	Description
Water Resistance	Mechanical Enclosure	External components should be water-resistant to rain/splash for 30 minutes.
Indicator Light	Lighting	Indicator lights on the device must be at least 50 lumens.
Battery Life	Power	The system should be operable for up to 2 hours with the built-in rechargeable power system.
Distance Detection	Sensor	The blind spot/rear view component must be able to detect objects within 3 m from the device.

Table 2.2.2: List of Non-Essential Non-Functional Specifications for StreetSmart

Specification	Subsystem	Description
Weight Limit	Overall	The main device should be less than 0.5 kg.
Cost	Overall	The bill of material cost for the prototype must be under \$600.
System Inter-Wireless Distance Range	Communication	The system can communicate with its subcomponent's wireless interfaces from a distance of up to 2 m.
Wireless Latency	Communication	The inter-wireless connection latency on the device should be under 250 ms.

## 3. Detailed Design

### 3.1 Central Controller

The central controller consists of the core functionalities of StreetSmart, along with the main processing unit. It is the more complex of the two controllers. See a breakdown of each of its subcomponents below.

#### 3.1.1 Mechanical Enclosure

The central controller mechanical enclosure holds all electronic devices used in Section 3.1. The enclosure contains the minicomputer, power, lighting, and sensor hardware in a compact space. In this design iteration, the focus was on optimization. The cameras were replaced with lidar, and a dedicated PCB was printed to save space, resulting in a smaller form factor with a lidar cutout. The new enclosure's volume is around  $720\text{cm}^3$ , while the previous volume was  $1584\text{cm}^3$ , resulting in a 55% volume reduction. **Previous iterations of the prototype can be found in Appendix E.** Grooves (highlighted in orange) were added for watertight sealing which was achieved by filling the gap with water-resistant sealing. The central controller is mounted on the bike via a mechanical mounting arm, which is demonstrated in section 4.1. The figure below illustrates the current design.

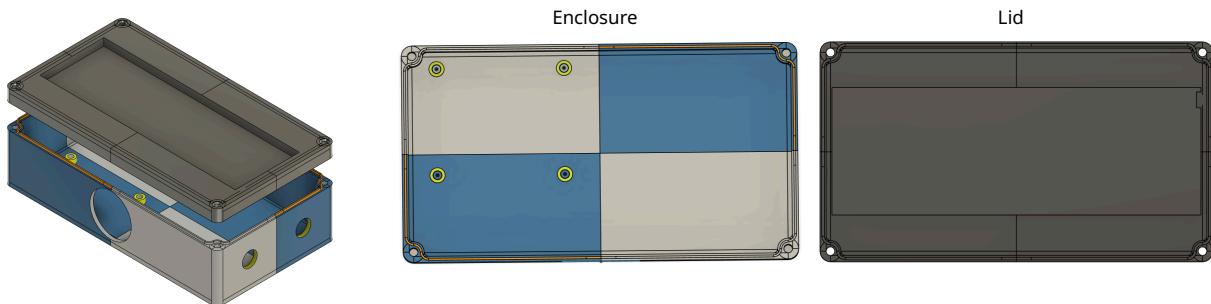


Figure 3.1.1.1: Isometric View of Central Enclosure (with Top View) and Lid (with Bottom View)

#### 3.1.2 Central Controller PCB

The Central Controller printed circuit board (PCB) is responsible for connecting the communication and power between the subsystems and the minicomputer. By using low-speed communication lines and power traces, this PCB board can be accomplished with a 2 layer board. Although the PCB routing is straightforward, caution must be demonstrated when routing power-switching components as traces need to be appropriately sized to meet high power requirements. Power switching components need

to be placed in a location that is far from communication lines, the communication module, and the accelerometer, to prevent interference.

### 3.1.3 Software Subsystem Design

The software subsystem consists of all the code developed by the team to leverage the features brought by the hardware in the central controller. This section explains the design for each algorithm used within the product.

#### 3.1.3.1 Braking Detection Algorithm

StreetSmart uses an MPU-6050<sup>[11]</sup> accelerometer to measure a user's acceleration, letting the device know when to trigger the brake lights. This subcomponent highlights the software design required to read from the accelerometer, clean the data with a filter, and then decide whether or not to send a trigger to the brake lights. The inputs to this subcomponent are the raw acceleration ( $\text{m/s}^2$ ) and gyro data ( $^\circ/\text{s}$ ) in the X, Y, and Z directions. The output is a potential trigger of the brake light subcomponent.

The code snippet below shows a simple Python program which prints the raw data from the MPU-6050 using its Arduino library.

```
import time
import board
import adafruit_mpu6050

i2c = board.I2C()
mpu = adafruit_mpu6050.MPU6050(i2c)

while True:
    print("Acceleration: X:%.2f, Y: %.2f, Z: %.2f m/s^2" % (mpu.acceleration))
    print("Gyro X: %.2f, Y: %.2f, Z: %.2f rad/s" % (mpu.gyro))
    print("")
    time.sleep(1)
```

Figure 3.1.3.1.1: Sample Python Code<sup>[12]</sup> Accessing Raw Measurements from the MPU-6050

The algorithm continuously reads the X, Y, and Z acceleration values, as well as the gyro data from the MPU-6050 sensor at regular intervals. This data is used to determine the deceleration of the vehicle. To remove noise, experimenting with filters is necessary. Implementing a Kalman filter<sup>[13]</sup> is likely the ideal method to smooth out the data. This increases the accuracy of deceleration values.

With the acceleration data in all 3 directions, the resultant acceleration is computed using the formula in Equation (1).

$$\text{resultant\_acceleration} = \sqrt{x^2 + y^2 + z^2} \quad (1)$$

The filtered acceleration values are compared against an experimentally determined threshold that implies the user is slowing down enough to warrant triggering the brake lights. To avoid continuous triggering of the brake lights due to small fluctuations in acceleration, the acceleration readings must satisfy the threshold condition for a minimum amount of time, or reach a minimum value before triggering the brake lights. These values can also be determined experimentally during implementation.

The gyro inputs can be optionally used to validate deceleration detection. When a bike decelerates, it is expected to lean forward due to the weight shift. This forward tilt indicates braking. Like acceleration, the pitch angle obtained from the gyro data can be monitored to confirm if it aligns with the expected behaviour during braking. The expected range is found experimentally by monitoring real gyro data. This additional validation helps reduce false positives. Using a boolean value to control the state of the brake lights: if the brake lights are off and the threshold conditions are met, the algorithm triggers the boolean value (equal to 1). If the conditions are not met, regardless of the current state of the brake lights, they should be turned off (equal to 0).

### 3.1.3.2 Distance Detection Algorithm

This distance detection algorithm leverages measurement data from the LD19 Lidar (refer to section 3.1.7.1 for specifications). The LD19 Lidar continuously sends measurement data packets serially, and this data is eventually parsed into a more readable format. Each measurement data packet contains multiple measurement data points that consist of a 2 byte distance value (in millimeters) and a 1 byte confidence value <sup>[24]</sup>.

Header	VerLen	Speed		Start angle		Data	End angle		Timestamp		CRC check
54H	2CH	LSB	MSB	LSB	MSB	.....	LSB	MSB	LSB	MSB	1Byte
↓											
Measuring point 1			Measuring point 2			...	Measuring point n				
distance		intensity	distance		intensity		distance		intensity		
LSB	MSB	1 Byte	LSB	MSB	1 Byte	...	LSB	MSB	1 Byte		

Figure 3.1.3.2.1: Format of a Measurement Data Packet<sup>[24]</sup>

The angle of each measured data point can be obtained using linear interpolation between the starting and ending angle.

Given that the step is  $step = (end\ angle - start\ angle) / (number\ of\ data\ points - 1)$ <sup>[24]</sup>

The angle of a certain data point is

$angle = start\ angle + step * i \quad where\ i \in [0, number\ of\ data\ points]$ <sup>[24]</sup>.

If the angle of a measured data point is less than or equal to 50° or is greater than or equal to 310° (100° field of view), the distance and angle information is stored. Refer to Figure 4.1.1.1 to understand how the lidar interprets angles.

Once all the data points from the measurement data packet have been parsed, any decently large objects breaching the distance threshold can be detected and used to trigger the haptic feedback controller. **Refer to Appendix A for the initial design of the distance detection algorithm.**

### 3.1.3.3 LED Pattern Algorithm

The LED Pattern Algorithm sends LED patterns to the WS2812B<sup>[14]</sup> light strip based on the current light state. The input to the module is a light state (LEFT, RIGHT, BRAKE, or OFF), which toggles the LED to the desired pattern. The output of this module is a light signal array that is sent to the GPIO pin connected to the WS2812B light strip.

The following code exhibits how the LED can be configured on a Raspberry Pi:

```
import board
import neopixel

GPIO_PIN = board.D18 # Example usage of GPIO pin 18
NUM_OF_LEDS = 60

# Initial pixels array setup
pixels = neopixel.NeoPixel(GPIO_PIN, NUM_OF_LEDS)

# Example configuration to make first LED red based on RGB values
pixels[0] = (255, 0, 0)
```

Figure 3.1.3.3.1: Sample Code<sup>[15]</sup> Showing how to Configure LEDs on the Raspberry Pi 4

### 3.1.3.4 Wireless Communication Algorithm

The communication algorithm on the central controller is responsible for sending and receiving instructions to and from the NRF24L01<sup>[16]</sup> module on the haptic controller. The central controller is the host device in this wireless network (main sender), and the haptic controller is the guest device (main receiver). Two-way communication is still possible

since the haptic controller can send messages in the form of acknowledgements. For information on the haptic controller's wireless algorithm, refer to Section 3.2.3.2.

The algorithm communicates with the controller subsystems of each device via the SPI protocol, which is accessible using built-in libraries. The format of the messages is a custom data structure which contains a variety of metadata as well as a boolean data type indicating whether to trigger the haptic engine, as shown in Figure 3.1.3.4.1. Also in the figure, note that *myRadio* has methods like *setChannel()*, *setPAlevel()*, and *setDataRate()* which determine the communication parameters. The values are determined experimentally based on factors like physical distance and data transmission size.

```
#include <SPI.h>
#include "RF24.h"

RF24 myRadio (9, 10);
byte addresses[][6] = {"1Node", "2Node"};

struct package
{
    int id = 1;
    bool enHaptic = true;
    char text[100] = "Some kind of message";
};

typedef struct package Package;

void setupComm()
{
    myRadio.begin();
    myRadio.setChannel(115);
    myRadio.setPAlevel(RF24_PA_MAX);
    myRadio.setDataRate(RF24_250KBPS);
    myRadio.openWritingPipe(addresses[0]);
    myRadio.openReadingPipe(1, addresses[0]);
    myRadio.startListening();
}

void loop_host() {
    unsigned char recv_data;
    unsigned char trans_data;
    radio.stopListening();
    if (!radio.write(&trans_data, sizeof(unsigned char))) { //Sends data
        Serial.println("No acknowledgement");
    }
    radio.startListening();
    unsigned long started_waiting_at = millis();
    while (!radio.available()) {
        if (millis() - started_waiting_at > 200) {

```

```

        Serial.println("timeout!");
        return;
    }
    radio.read(&recv_data, sizeof(unsigned char)); //Reads the Acknowledgement data
}

```

Figure 3.1.3.4.1: Pseudocode<sup>[22]</sup> Outlining the Setup and Algorithm Required for the Host Side of the Wireless Communication

Upon receiving a message from the haptic controller to trigger the turn signal lights, it will notify the lighting subsystem to do so. Similarly, if the central controller wants to trigger the haptic engine, this algorithm sends a message to the haptic controller to do so.

### 3.1.4 Controller Subsystem

StreetSmart uses a Raspberry PI 4 Model B with 4 GB RAM as the controller because it has a good price for the reliability and number of features provided. The RPI4 has a 4-bit quad-core processor<sup>[8]</sup> which means tasks are easily parallelizable for speed and power efficiency. This model has a plethora of connectivity functionalities such as True Gigabit Ethernet, WIFI, Bluetooth, 40-pin GPIO connector, USB ports, etc<sup>[8]</sup>. The availability of Bluetooth allows for future proofing in case of the addition of an Android app. Furthermore, StreetSmart connects many sensors to the controller, so having many connectivity options, such as USB, I2C, and SPI<sup>[8]</sup> are very useful.

Another appealing feature of the RPI4 is the availability of hardware and software support and the RPI community (RPI Forums, StackExchange, etc.). These resources are typically harder to find for other board brands and are vital for effective troubleshooting.

### 3.1.5 Power Subsystem

The power subsystem is responsible for providing stable output voltages, powering the system off of the battery, and recharging the battery. The battery runtime for the system must last at least 2 hours under reasonable assumptions. The required battery capacity is determined by performing a power consumption analysis as shown in Table 3.1.5.1. Since components of the system draw current at different voltages and for different durations, the required capacity for each component is calculated with Equation (2).

$$\text{Capacity(Wh)} = \text{Voltage(V)} \times \text{Current(A)} \times \text{Time(hr)} \quad (2)$$

In Table 3.1.5.1, it is assumed that a typical user triggers the brake lights or turn signals for a max duration of 1 out of 2 hours. All other components run for the full 2 hours and are treated as running at a worse-case full power draw.

Table 3.1.5.1: Total Power Consumption Analysis for the Central Controller's Power Subsystem

Load	Voltage (V)	Max Current (A)	Number of Loads	Max Power (W)	% of Runtime at Max Power	Total Watt-hour per Load (Wh)	Notes
Raspberry PI 4 Model B with 4 GB RAM	5.00	1.01	1.00	5.05	100.00%	10.10	1
WS2812B LED	5.00	0.02	60.00	6.00	50.00%	6.00	2
FHL-LD19 Lidar	5.00	0.18	1.00	0.90	100.00%	1.80	3
NRF24L01 Communication Module	3.30	0.01	1.00	0.04	100.00%	0.08	4
MPU6050 Accelerometer Module	5.00	0.00	1.00	0.02	100.00%	0.04	5
<b>Peak Power Consumption (W) =</b>				12.01	<b>Minimum Battery Watt-hour (Wh) =</b>	18.02	

#### Notes

1. Based on benchmarked consumptions at 100% CPU Load <sup>[17]</sup>
2. Based on WS2812B single colour max current consumption <sup>[15]</sup>
3. Based on USB Meter Measurements
4. Based on NRF24L01 max receiving current @ 250 Kbps <sup>[16]</sup>
5. Based on MPU6050 average current consumption for the Accelerometer and Gyroscope enabled <sup>[12]</sup>

Based on the power consumption analysis, the battery capacity must be at least 18Wh. To select a battery pack for this system, the nominal battery voltage must be selected as it will determine if a buck or boost converter is required to provide the 5 V supply voltage. Observing Table 3.1.5.2, different nominal battery voltages are considered, where a higher nominal voltage requires a lower max battery current. Taking into consideration safety and required current handling, a 7.4 V lithium-ion battery with a capacity of 5.2 Ah is selected. This battery pack has lower power losses due to the higher voltage as well as provides more than enough energy to run for 2 hours.

Table 3.1.5.2: Battery Requirement Analysis for the Central Controller's Power Subsystem

Nominal Battery Voltage (V)	Peak Power Consumption (W)	Max Battery Current (A)	Min Battery Watt-hour (Wh)	Min Battery Capacity(Ah)	Step Up or Step Down?
7.40	12.01	1.62	18.02	2.44	Step Down
3.70	12.01	3.25	18.02	4.87	Step Up

To supply the 5 V rail for the Raspberry PI and other components that run on 5 V, a buck converter is required to step down the 7.4 V to 5 V. The LM2596S buck converter is selected as it has an input voltage range of 3.2 V to 35 V and an output voltage range of 1.25 V to 30 V. Additionally, the regulator has a max continuous output current of 3 A which is required for the system's peak current draw. In the latest iteration, a prebuilt module was selected to reduce prototyping costs. **Refer to Appendix C for the previously customized buck converter design.**

To supply the 3.3 V for the NRF24L01 Module, that rail can be extracted from the Raspberry PI's GPIO pins. Onboard the Raspberry PI will have a 3.3 V rail which can supply a maximum of 50 mA<sup>[9]</sup>.

### 3.1.6 Lighting Subsystem

The lighting subsystem used is a WS2812B addressable RGB lighting strip. The individually addressable configuration allows specific patterns to be created, such as the turn signals and brake lights. The patterns are controlled by the LED Pattern Algorithm. Refer to Section 3.1.3.3 for more details.

### 3.1.7 Sensor Subsystem

#### 3.1.7.1 Lidar

StreetSmart uses a DTOF Lidar LD19 for distance detection. The LD19 Lidar has a 360° scanning radius, as well as a measurement radius of 12 m<sup>[23]</sup>. It also has a 4500 Hz measurement frequency and boasts a service life greater than 10 000 hours<sup>[23]</sup>. The LD19's wide field of view and frequent polling will provide accurate and up-to-date environmental data that our distance detection algorithm can use to accurately identify objects of interest/danger. Furthermore, having a field of view of 90° and a detection range of 3 m are functional and non-functional specifications, respectively, and these criteria are fulfilled by this lidar. **Refer to Appendix B for information about the sensors that were used in the initial iteration of the product for distance detection.**

#### 3.1.7.2 Accelerometer

StreetSmart uses an MPU-6050<sup>[12]</sup> accelerometer to monitor when the user may be braking to trigger brake lights. The accelerometer can provide measurements of acceleration and rotational motion in three dimensions (X, Y, and Z). It can measure acceleration with a sensitivity range of up to ±16 G, allowing it to capture the range of accelerations specified in the essential functional specification ( $\pm 30 \text{ m/s}^2$ ). The gyroscope offers angular rate

measurements with a range of up to  $\pm 2000^\circ/\text{s}$ , enabling analysis of vehicle orientation. The MPU-6050 is compact, low-power, and offers the I2C communication interface, making it easy to create software using its readings. This makes it a reliable choice for implementing brake light triggering systems.

### 3.1.8 Communication Subsystem

The communication subsystem is led by the NRF24L01<sup>[16]</sup> wireless module. It operates in the 2.4 GHz frequency band, which is within Canadian regulations<sup>[10]</sup> and less prone to interference. It also offers a low power consumption mode, making it suitable for the battery life specification of 2 hours. The NRF24L01 offers a maximum data transmission rate of 2 Mbps, which is far more than needed for data transmission. The size of the part is also helpful for keeping the weight of StreetSmart below 0.5 kg. Paired with the Arduino NRF24L01 code module, this device ensures easy communication between the two controllers. The central controller's communication subsystem acts as a host to the haptic system's communication subsystem, which is the guest.

## 3.2 Wireless Haptic Feedback Controller

### 3.2.1 Mechanical Enclosure

The wireless controller mechanical enclosure holds all electronic devices used in Section 3.2. The enclosure contains the microcontroller, power, haptic, and communication hardware in a compact space, with a turn signal switch. The focus of this design iteration for the haptic controller was optimization and functionality improvement. The haptic controller is mounted to the handle of the bike via a standard handlebar bracket, demonstrated in section 4.1. To add additional functionality for the controller, the enclosure also acts as a phone holder with adjustable clamping. With the new PCB and fewer motors, the new enclosure's volume is 156 cm<sup>3</sup> compared to the previous volume of 195 cm<sup>3</sup>, this results in a 20% volume reduction, allowing the phone holder functionality to be added. **Previous iterations of the prototype can be found in Appendix E.** Grooves (highlighted in orange) were added for watertight sealing which was achieved by filling the gap with water-resistant sealing. The central controller is mounted on the bike via a mechanical mounting arm. The figure below illustrates the current design.

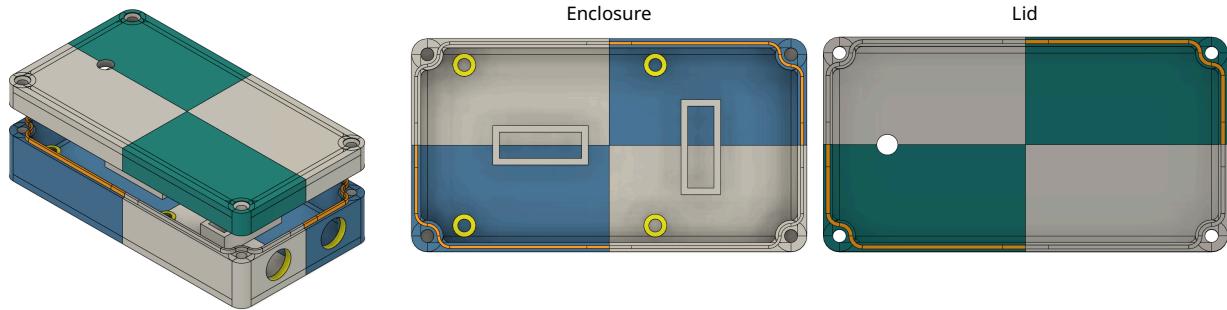


Figure 3.2.1.1: Isometric View of Haptic Enclosure (with Top View) and Lid (with Bottom View)

### 3.2.2 Wireless Haptic Feedback Controller PCB

The wireless haptic feedback controller printed circuit board (PCB) is responsible for neatly connecting all the communication and power lines between the subsystems and MCU. This board can be accomplished with a 2-layer board since it is a low-speed and low-power system. Power switching parts need to be placed far from communication lines, the communication module, and the accelerometer to prevent interference.

### 3.2.3 Software Subsystem Design

The software subsystem consists of all the code developed by the team to leverage the features brought by the hardware in the haptic controller. This section explains the design for each algorithm used within this device.

#### 3.2.3.1 Haptic Algorithm

The haptic algorithm controls the haptic motor to its desired state (on or off) based on data from the communication algorithm. The input to this algorithm is the requested state of the haptic motor, which determines the state of the motor. The output of this module turns on the correct GPIO pin to activate or deactivate the haptic motors. The algorithm activates the haptic motors with a vibration pattern for maximum alertness and optimal battery usage. Additionally, the new haptic button provides a mode to toggle haptic vibration. By default, the haptics will remain on and with an input of a long hold of the button (2 seconds) the haptics will be toggled off, indicated with a flash on the button. All functionalities remain the same except the haptic motors are off in this mode.

#### 3.2.3.2 Communication Algorithm

Referring to Section 3.1.3.4 and Figure 3.1.3.4.1, the communication algorithm on the haptic controller uses the same setup code.

```

void loop_guest() {
    unsigned char recv_data;
    unsigned char trans_data;

    if (radio.available()) {
        while (radio.available()) {
            radio.read(&recv_data, sizeof(char)); // Reads the incoming data
        }
        radio.stopListening();
        radio.write(&trans_data, sizeof(char)); // Acknowledges with data
        radio.startListening();
    }
}

```

Figure 3.2.3.2.1: Pseudocode<sup>[22]</sup> for the algorithm on the Guest Side of the Wireless Communication

The algorithm in Figure 3.2.3.2.1 shows the guest behavior of the haptic controller. It continuously receives payload from the central controller and responds with acknowledgements. Upon receiving a message from the central controller to trigger the haptic engine, it notifies the haptic controller to do so. Similarly, if the haptic controller wants to trigger the brake lights, this algorithm will send a message to the central controller through the acknowledgement. This is how two-way communication is achieved through the host-guest model. Additionally, the haptic mode button serves as a connection indicator. A long flash will indicate that the haptic controller is attempting to communicate with the main controller, and a solid light indicates that the controllers are connected and communicating.

### 3.2.4 Controller Subsystem

The controller subsystem of the haptic controller is the brain of this device. StreetSmart uses an Arduino Nano that contains the ATmega328P<sup>[19]</sup> chip controlled by low-power logic. The ATmega328P provides sufficient processing power, 32 KB of in-system programmable flash memory, and 23 GPIO pins to handle controller functions effectively. The Arduino ecosystem offers libraries for any software and simple integration with other Arduino-compatible devices like the RPI4 Model B in the central controller. The chip also has SPI and I2C communication protocols to gather data from the sensors connected to it.

### 3.2.5 Power Subsystem

Like the power subsystem for the central controller, this power subsystem is responsible for providing stable output voltages, powering the system off of the battery for at least 2 hours, and being able to recharge the battery. The power consumption analysis is shown in Table 3.2.5.1. It is assumed that a typical user receives haptic feedback for a maximum

duration of 1 out of 2 hours. All other components are running for the full 2 hours and are treated as running at a worse-case full power draw.

Table 3.2.5.1: Total Power Consumption Analysis for the Haptic Controller's Power Subsystem

Load	Voltage (V)	Max Current (A)	Number of Loads	Max Power (W)	% of Runtime at Max Power	Total Watt-hour per Load (Wh)	Notes
Haptic Motor	3.30	0.10	4.00	1.32	50.00%	1.32	1
Atmega328P	5.00	0.02	1.00	0.10	100.00%	0.20	2
NRF24L01 Communication Module	3.30	0.01	1.00	0.04	100.00%	0.08	3
<b>Peak Power Consumption (W) =</b>				1.46	<b>Minimum Battery Watt-hour (Wh) =</b>	1.60	

#### Notes

1. Based on manual power measurements
2. Based on the max current draw of the ATmega328P at 5 V @16 MHz <sup>[19]</sup>
3. Based on NRF24L01 max receiving current @ 250 Kbps <sup>[16]</sup>

Based on the power consumption analysis, the battery capacity must be at least 2 Wh. Due to the low current draw observed in Table 3.2.5.2, a 3.7 V lithium-ion battery with a capacity of 2 Ah is selected. This battery will be more than sufficient to power the haptic controller for 2 hours. Choosing a larger capacity battery provides convenience to the user as they do not have to recharge it as often as the central controller.

Table 3.2.5.2: Battery Requirement Analysis for the Haptic Controller's Power Subsystem

Nominal Battery Voltage (V)	Peak Power Consumption (W)	Max Battery Current	Min Battery Watt-hour (Wh)	Min Battery Capacity(Ah)	Step Up or Step Down?
3.70	1.46	0.40	1.60	0.43	Step Up

To supply the 5 V rail for the ATmega328P, a boost converter is required to step up the battery's nominal 3.7 V to 5 V. The XL6009 boost converter is selected as it has an input voltage range of 3 V to 32 V and an output voltage range of 5 V to 35 V. **Refer to Appendix D for the previously customized boost converter design.**

To supply the 3.3 V for the NRF24L01 Communication Module and Haptic Motors, another regulator needs to supply 3.3 V. This can be taken from the onboard 3.3 V regulator of the Arduino Nano.

### 3.2.4 Haptic Subsystem

The Haptic Subsystem consists of two 3 V 16000 RPM coreless motors which allow for variable vibration intensity based on the voltage input. These voltages to the motors will be controlled via a PWM-driven mosfet. The design of this subsystem in conjunction with the haptic algorithm outlined in Section 3.2.3.1 vibrates the entire wireless controller device in a pulsing pattern to alert the user.

In the latest iteration, LEDs were added to the haptic subsystem to allow for visual indications of the current turn signal state and/or blindspot status. The visual indication provides more reliable feedback to the user when riding on rougher terrain where vibrations may not be as prominent. An additional push-button provides customizability where detection distance can be varied and vibrations can be turned off.

### 3.2.5 Communication Subsystem

Refer to Section 3.1.8. The communication subsystem uses the NRF24L01<sup>[16]</sup> wireless module for the same reasons presented in that section. The haptic controller's communication subsystem acts as a guest to the central controller's communication subsystem, which is the host.

## 4. Prototype Data

### 4.1 Central Controller

#### 4.1.1 Lidar Accuracy

In the latest design, the lidar was used due to its accurate measurements and low power consumption. The lidar provides data that can be visualized as distance in a radial pattern from the user. The data was processed so that it only considers the 100-degree view from the back of the user instead of the full 360-degree view of the lidar. The distance detection algorithm forms a 3 meter radial bubble behind the user to check if objects are within it. When an object is present in the user's blindspot, the lidar can detect these variations in distances from the user. In Figure 4.1.1.1, an object is placed 2.9 m from the user and it is accurately picked up by the lidar. This provides accurate data to the system compared to the previous iteration using stereo camera vision. The lidar can reject environmental noise and sunlight, which the cameras cannot.

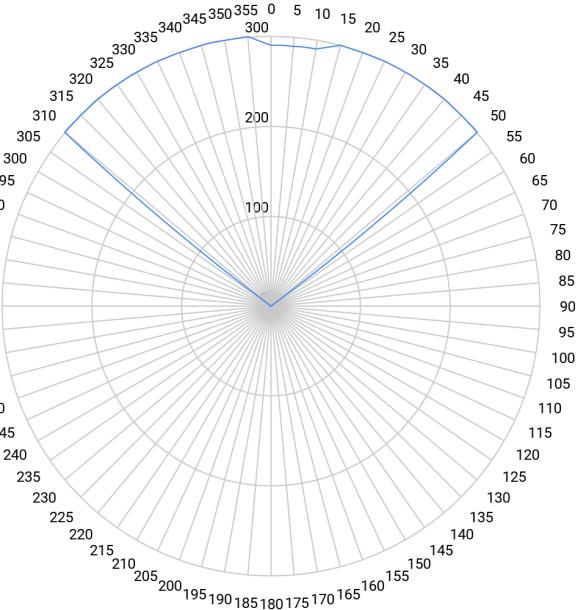


Figure 4.1.1.1: Lidar Distance Detection with Object at 2.9 m

#### 4.1.2 Braking Detection Accuracy

The accelerometer measures acceleration in the X, Y, and Z planes. In the first iteration of StreetSmart, the accelerometer was placed in the device so that the Z axis represents the

forward-facing direction of the bicycle. The Y axis is on the horizontal plane going through the bike, and the X axis is on the vertical plane. To properly detect when the rider brakes, the current accelerations and moving averages of acceleration values in each direction are compared against experimentally determined thresholds. To find these threshold values, real data was collected while the bike slowed down at different rates.

As seen in Figure 4.1.2.1, the general trend is that the Y acceleration is almost always around 0. However, if the user brakes suddenly, the accelerometer may detect some horizontal deceleration. Similarly, the X acceleration typically stays around 1 G, which represents gravity. If the user slows down, it can drop slightly. The biggest indicator of the rider braking is if the forward acceleration drops significantly, meaning accuracy is most important in this direction. For this reason, the Z acceleration value compared against the threshold is actually the delta between the instantaneous Z acceleration and a moving average of Z accelerations. The data also shows that as the user comes to a stop, the device shakes, causing a sudden spike in acceleration.

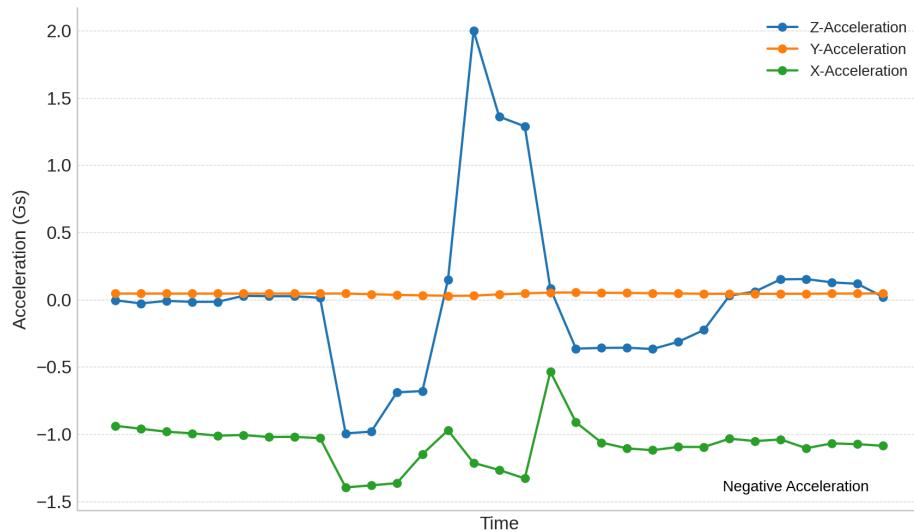


Figure 4.1.2.1: 3-Plane Acceleration Measured while a Bicycle Brakes Regularly

With these observations in mind, the brake lights will only turn on if the condition in Figure 4.1.2.2. The directions that each plane represents have changed in the final prototype, but the algorithm uses the same thresholds, just in different directions.

```
z_delta > 0.3 and z_delta < 0.8 and Ay > -1.25 and Ay < -0.75 and Ax < 0.3 and Ax > -0.3
```

Figure 4.1.2.2: Condition to Turn on Brake Lights

### 4.1.3 Mechanical Enclosure

The enclosure was tested in multiple aspects to meet the original criteria devised in the plan. First, the mounting system was revisited to use a mechanical arm with a pivot joint. This allowed flexibility in controller tilt, while ensuring the stability of mounting the controller, as visualized in Figure 4.1.3.2. Bike ride testing was done in practical scenarios such as on paved roads, and in extreme scenarios such as off-road dirt terrain, grass, and hills. Under various conditions, the mount upheld the controller with no issues.



Figure 4.1.3.1: Prototype of Central Controller



Figure 4.1.3.2: Mechanical Mount in Different Pivot Positions

Additionally, waterproofing was tested with the main controller. The enclosure was sealed with perfect-fitting grooves filled with rubber adhesives for maximum sealing. The acrylic light panel was also sealed with glue to avoid water leakage. The controller was tested under various water splash scenarios to mimic rainy scenarios and wheel splash on rainy terrain for over 30 minutes, meeting the non-functional specification. Overall, there was no water damage after various tests, as shown in Table 4.1.3.3.

Table 4.1.3.3: Central and Haptic Controller Water Testing Results

Water Testing Scenario	Description & Discussion
No Water	Device is fully operational under dry conditions.
Light Splash	Testing was performed with mist sprayer. Water resided on surface of enclosure. No water seeped through inside when probed open for analysis.
Medium Splash	Testing was performed by splashing water over the device multiple times manually. Water resided on surface of enclosure, with some water along the grooves. No water seeped through inside when probed open for analysis.
Heavy Splash	Testing was performed by placing device under shower head. Water resided on surface of enclosure, with water along the grooves. No water seeped through inside when probed open for analysis.

## 4.2 Wireless Haptic Feedback Controller

### 4.2.1 Mechanical Enclosure

The enclosure was tested in multiple aspects to meet the original criteria devised in the plan. First, the mounting system was revisited to use a mounting bracket for bike handlebars, as shown in Figure 4.2.1.2. The mount clamp was fitted with additional plastic to ensure a tight grip and avoid tilting or shifting. This allowed the wireless controller to clamp safely, and it successfully withstood multiple ride and water tests that the central controller also undertook as described in Section 4.1.3. The haptic controller enclosure also underwent water testing and results are described in Table 4.1.3.3 above.



Figure 4.2.1.1: Prototype of Haptic Controller

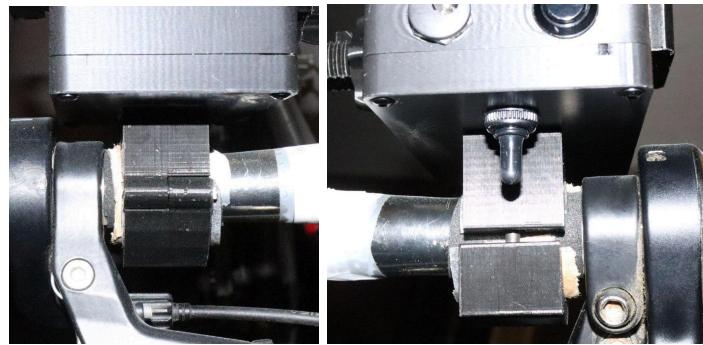


Figure 4.2.1.2: Top and Bottom View of Handlebar Mount for Haptic Controller

For additional functionality, a phone mount bracket was added onto the haptic controller enclosure. Since the controller occupies a significant portion of the handlebar typically reserved for a phone mount, users can utilize StreetSmart without sacrificing any potential functionality. The phone mount with various test devices can be seen in Figure 4.2.1.3.



Figure 4.2.1.3: Phone Mount Bracket on Haptic Controller with Various Test Devices

A haptic mode button on the left side of the device was implemented for users to toggle the vibration on or off by holding the button for 2 seconds. Additionally, turn signal indicators have been incorporated onto the front of the controller. These lights mirror the rear indicators, enabling users to determine whether their signal lights are activated.

## 4.2.2 Haptic Tuning

To ensure the rider is aware when the lidar detects activity in a blind spot, the haptic feedback vibrates at an intensity that was experimentally determined. To find this value, the StreetSmart group members rode several bicycles in different environments. The team reached a consensus on an intensity level that provided noticeable feedback through the handlebars without causing alarm to the rider. Because all members of StreetSmart are of different heights and weights, this method is sufficient for tuning the haptic intensity.

### 4.3 RF Communication Reliability

To provide reliable communication between the central controller and haptic controller, the NRF24L01 communication modules' frequency needs to be tuned based on experimental data. This module communicated in the 2.4 Ghz frequency band which can overlap with Wifi which can result in interference and packet losses. Visualizing the packet losses in Figure 4.3.1, the packet losses are higher in the 2.4 Ghz to 2.5 Ghz range, which can be due to interference from Wi-Fi or surrounding electronics. With this information, the NRF24L01 had its parameters tuned to the highest supported value of 2.524 GHz channel to increase reliability and range. Testing with this parameter resulted in max 100 ms latency and a max line of sight range of 10 meters.

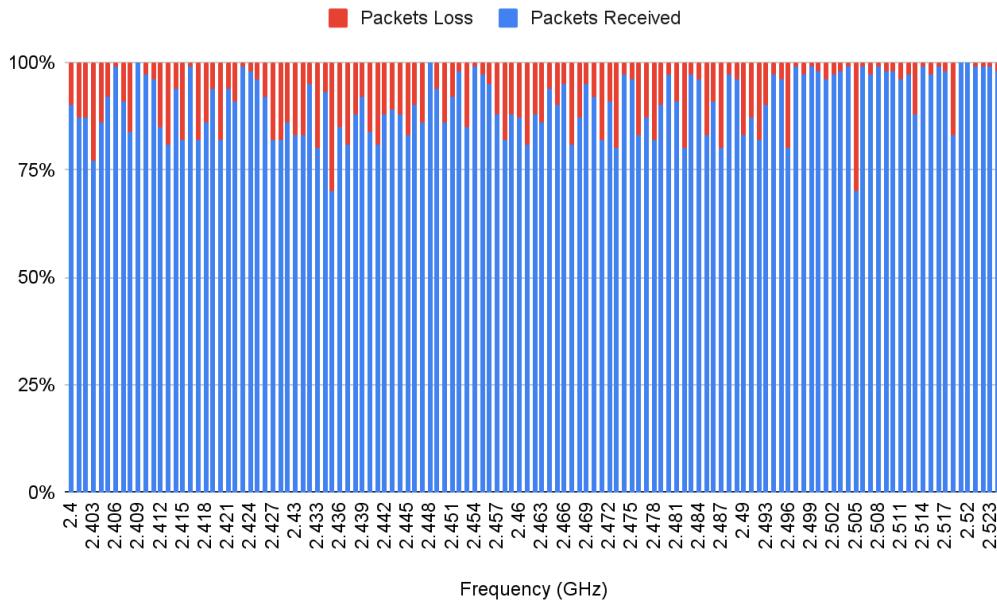


Figure 4.3.1: NRF24L01 Packet Loss over Channel Frequency

### 4.4 PCB Design

In the first iteration of this project in ECE498A, the main objective was to construct the prototype as fast as possible to validate functionality. The initial prototype was on a perforated board which allowed for flexibility to create reliable solder connections between components while also allowing for small size and moderate customization, unlike the bulky breadboards. However, this prototype seen in Figure 4.4.1 required many wires to connect between components and there were no mounting features resulting in the inability to create a small form factor casing.

In the latest iteration, the PCB was designed to take into consideration being a small form factor while also reducing costs. Instead of designing 2 separate PCBs, one for the central controller and one for the haptic controller. In Figures 4.4.2, the two circuits were merged into one board since most of the circuit repeats between both systems. It was designed to fit as a module on top of the Raspberry Pi, but also be a self-sufficient board for the haptic controller seen in Figure 4.4.3. Different components would be populated depending on which board is needed.

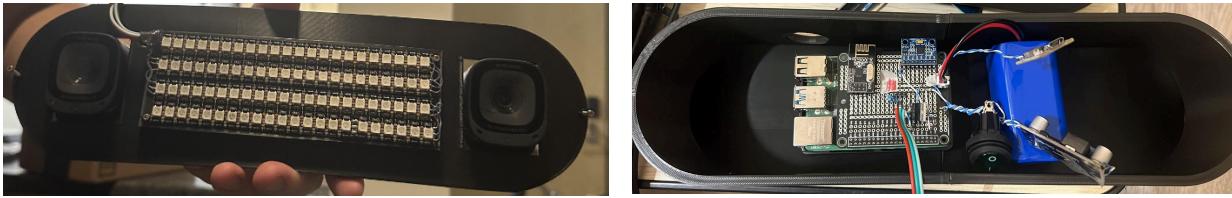


Figure 4.4.1: Initial Prototype of StreetSmart

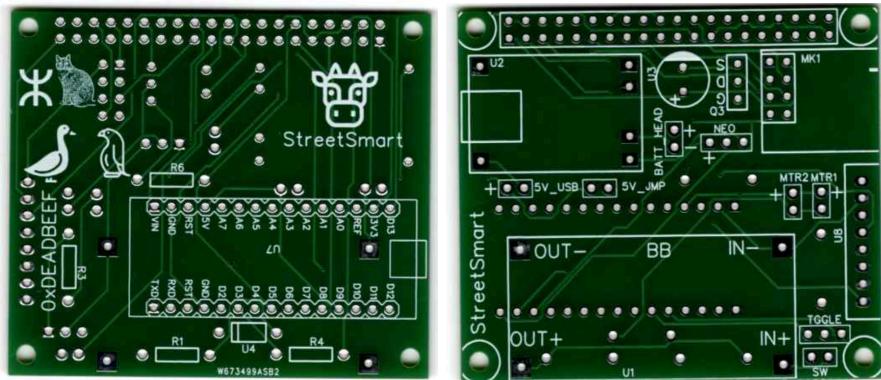


Figure 4.4.2: Latest PCB Design for Electrical Components



Figure 4.4.3: PCB Fitted into Central Controller Housing

## 5. Discussion and Conclusions

### 5.1 Evaluation of Final Design

The objective of this project is to create a portable and easy-to-use device to increase the awareness of cyclists and motorized vehicle drivers. This design meets the objective by being lightweight (less than 0.5 kg) and designed to be used with no consumer setup necessary. The initial prototype from ECE498A met the original design specifications specified in Section 2 in terms of functionality, water resistance, and battery life. Compared to the initial prototype, the final prototype boasts a lighter weight, more compact mechanical enclosures, mounts for both devices, enhanced haptic and acceleration algorithms, and user-friendly updates like a phone mount, better buttons, and turn indicator lights on the haptic controller. Overall, the final design still meets all the original requirements and the StreetSmart is a much more polished product.

### 5.2 Use of Advanced Knowledge

Technical knowledge from many upper-year courses is being applied in the design of StreetSmart. ECE 423 (Embedded Computer Systems) provides essential upper-year knowledge on the embedded system aspects of StreetSmart. The specification and design techniques covered to assist in defining the hardware and software requirements of the device. The hardware-software integration topics help ensure the device's hardware and software components work together seamlessly. In the testing stages of the project, verification methods can be applied to ensure the correctness and reliability of our embedded systems.

ECE 452 (Software Design and Architectures) makes it possible to apply design patterns and iteration strategies when implementing the algorithms of our software subsystems. This helps develop a well-structured and modular software architecture for the device. This also complements the design and integration methods taught in ECE 423.

ECE 463 (Design & Applications of Power Electronic Converters) gives invaluable insight into the power elements of StreetSmart. Analysis and design techniques for power converters assist in implementing the appropriate power supply for the device. Principles of power conditioning and power converters can also be applied to the power-related aspects of our device.

## 5.3 Creativity, Novelty, Elegance

StreetSmart is an innovative device designed to provide an all-in-one solution to the on-road challenges experienced by cyclists. The elegance of this solution is highlighted by its ease of use, versatility, and user-safety-oriented design. StreetSmart is a one-size-fits-all device that can be easily mounted to the rear of any type of bicycle, enabling StreetSmart to ensure a safer journey for all cyclists. With this solution, users also have the flexibility to customize their cycling experience with the click of a button. They are able to enable/disable haptic feedback, as well as increase or decrease the rear distance detection radius (1 m or 3 m). Finally, StreetSmart's wireless haptic feedback controller provides a convenient and safe place for cyclists to store their phones during their ride. Using this built-in phone stand, users can use their phones to navigate safely.

## 5.4 Quality of Risk Assessment

Although StreetSmart is designed to keep users safer on the road, there may be potential safety hazards with the device itself. The central controller and wireless haptic controller both contain lithium batteries, which can be hazardous if not handled properly (e.g. short circuits and operation outside of recommended temperatures). These scenarios could physically harm users in the form of electric shocks or burns.

In ECE 498A, the group identified these risks and took several precautions to prevent safety issues. To alleviate and avoid the hazards, the batteries were connected to a battery controller for recharge and regulation purposes, meaning that they were regularly monitored and protected properly. The circuits were handled delicately via ESD straps and proper protective gear. A fume extractor and goggles were used during the soldering and connection of circuits. Despite meticulous risk assessment and prevention, the group still faced a very small amount of risky situations. For example, while testing haptic feedback, the motor driver blew due to avoidable mistakes in the circuit. To mitigate future risks like that, the group installed protection diodes and continued to test extensively. Overall, the group did a good job assessing and mitigating risks.

## 5.5 Student Workload

In Table 5.5.1 below, each student's percentage contribution to StreetSmart is shown. The justification for Yixing contributing slightly more than the rest of the group members is his expertise in hardware design. Yixing was the primary contributor to areas like the power subsystem where other group members lack experience. Since StreetSmart is an

embedded system, Yixing also contributed to the design of other components. The remaining group members focused more on software algorithms and mechanical enclosures, as well as testing.

Table 5.5.1: Student Contribution to StreetSmart as a Percentage

Student	Percentage Contribution (%)
Yixing Qie	31
Kenneth Hung	23
Patricia Liu	23
Anees Aissaoui	23
<b>TOTAL</b>	<b>100</b>

## References

- [1] "Bike Safety Statistics," CAA National, <https://www.caa.ca/driving-safely/cycling/bike-statistics/> (accessed May. 17, 2023).
- [2] E. Barclay, "When bikes and cars collide, who's more likely to be at fault?," NPR, <https://www.npr.org/sections/health-shots/2011/05/20/136462246/when-bikes-and-cars-collide-whos-more-likely-to-be-at-fault> (accessed Jun. 4, 2023).
- [4] S. Yeager, "Why drivers look straight at cyclists and still don't 'see' them," Bicycling, <https://www.bicycling.com/news/a20043758/drivers-dont-see-cyclists-inattentional-blindness/> (accessed Jun. 4, 2023).
- [5] K. Sadekar, "Stereo camera depth estimation with opencv (python/C++)," LearnOpenCV, <https://learnopencv.com/depth-perception-using-stereo-camera-python-c/> (accessed Jun. 25, 2023).
- [6] K. Sadekar, "Making a low-cost stereo camera using opencv," LearnOpenCV, <https://learnopencv.com/making-a-low-cost-stereo-camera-using-opencv/> (accessed Jun. 25, 2023).
- [7] "Anker PowerConf C200 USER MANUAL," [https://cdn.shopify.com/s/files/1/0580/2262/5458/files/A3369\\_WEB\\_Manual\\_V01\\_202111220.pdf](https://cdn.shopify.com/s/files/1/0580/2262/5458/files/A3369_WEB_Manual_V01_202111220.pdf) (accessed Jun. 23, 2023).

- [8] Raspberry Pi 4 Computer Model B,  
<https://static.raspberrypi.org/files/product-briefs/Raspberry-Pi-4-Product-Brief.pdf> (accessed Jun. 23, 2023).
- [9] P. Clifford, "GPIO electrical specifications, Raspberry Pi input and output pin voltage and current capability," Mosaic Industries,  
<http://www.mosaic-industries.com/embedded-systems/microcontroller-projects/raspberry-pi/gpio-pin-electrical-specifications> (accessed Jun. 28, 2023).
- [10] F. Vergès, "5GHz Regulations in Canada (2018 update)," SemFio Networks,  
<https://semfionetworks.com/blog/5ghz-regulations-in-canada-2018-update/> (accessed Jun. 29, 2023).
- [11] "MPU-6050," TDK InvenSense,  
<https://invensense.tdk.com/products/motion-tracking/6-axis/mpu-6050/> (accessed Jun. 29, 2023).
- [12] B. Siepert, "MPU6050 6-DOF accelerometer and Gyro," Adafruit Learning System,  
<https://learn.adafruit.com/mpu6050-6-dof-accelerometer-and-gyro/python-and-circuitpython> (accessed Jun. 29, 2023).
- [13] M. Laaraiedh , "ArXiv.org e-print archive," Implementation of Kalman Filter with Python Language, <https://arxiv.org/ftp/arxiv/papers/1204/1204.0375.pdf> (accessed Jun. 30, 2023).
- [14] "Feature," WorldSemi, <https://cdn-shop.adafruit.com/datasheets/WS2812B.pdf> (accessed Jun. 30, 2023).
- [15] M. Murray, "Controlling WS2812B leds with a Raspberry Pi," The Geek Pub,  
<https://www.thegeekpub.com/16187/controlling-ws2812b-leds-with-a-raspberry-pi/> (accessed Jun. 29, 2023).
- [16] "Preliminary product specification V1 - Sparkfun electronics," Nordic Semiconductor,  
[https://www.sparkfun.com/datasheets/Components/SMD/nRF24L01Pluss\\_Preliminary\\_Product\\_Specification\\_v1\\_0.pdf](https://www.sparkfun.com/datasheets/Components/SMD/nRF24L01Pluss_Preliminary_Product_Specification_v1_0.pdf) (accessed Jun. 30, 2023).
- [17] "Power consumption benchmarks," Power Consumption Benchmarks | Raspberry Pi Dramble, <https://www.pidramble.com/wiki/benchmarks/power-consumption> (accessed Jun. 29, 2023).
- [18] "LM2596 simple Switcher® power converter 150-KHz 3-a step-down voltage ...," Texas Instrument, <https://www.ti.com/lit/ds/symlink/lm2596.pdf> (accessed Jun. 30, 2023).
- [19] "ATMEGA328P - Microchip Technology," ATmel,  
[https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P\\_Datasheet.pdf](https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf) (accessed Jun. 30, 2023).

[20] "Switching Current Boost/Buck Boost/Inverting DC/DC Converter," Kylinchip, <https://www.haoyuelectronics.com/Attachment/XL6009/XL6009-DC-DC-Converter-Datasheet.pdf> (accessed Jun. 30, 2023).

[21] "800 ma fixed low dropout positive regulator," Microchip Technology, <https://ww1.microchip.com/downloads/aemDocuments/documents/APID/ProductDocuments/DataSheets/21665D.pdf> (accessed Jun. 30, 2023).

[22] R. Bacon, "NRF24L01-transceiver," GitHub, <https://github.com/RalphBacon/nRF24L01-transceiver> (accessed Jun. 30, 2023).

[23] "DTOF LIDAR LD19," DTOF LIDAR LD19 - Waveshare Wiki, [https://www.waveshare.com/wiki/DTOF\\_LIDAR\\_LD19](https://www.waveshare.com/wiki/DTOF_LIDAR_LD19) (accessed Feb. 23, 2024).

[24] "LiDAR Sensor LD19 Development Manual," Elecrow, [https://www.elecrow.com/download/product/SLD06360F/LD19\\_Development%20Manual\\_V2.3.pdf](https://www.elecrow.com/download/product/SLD06360F/LD19_Development%20Manual_V2.3.pdf) (accessed Feb. 23, 2024).

# Appendices

## Appendix A - Distance Detection Algorithm Alternative

In the initial iteration of StreetSmart, the product used a distance detection algorithm that employed a stereo camera system – two USB cameras are placed in parallel, at a fixed distance apart<sup>[6]</sup>.

To set up the stereo camera system, the cameras are first calibrated using the following steps:

1. Calibrate each camera individually with the help of a Checkerboard. The goal of this process is to estimate the internal and external parameters of a camera and determine the relationship between a pixel from a 2D image to a point in 3D space.<sup>[6]</sup>
  - a. Firstly, use the checkerboard to define coordinates in 3D space. Next, take images of the checkerboard and use the function `cv2.findCheckerboardCorners()` to determine the pixel coordinates of the checkerboard corners within the images. Finally, use `cv2.calibrateCamera()` to determine the transformation between pixel coordinates and 3D coordinates<sup>[6]</sup>.
2. Determine the transformation between the two USB cameras<sup>[6]</sup>
  - a. Pass the output from camera calibration(step 1) into `cv2.stereoCalibrate()`. This function computes the rotation and translation between the two cameras, as well as the essential and fundamental matrices.
3. Apply stereo rectification by using the function `cv2.stereoRectify()`<sup>[6]</sup>.
4. Determine the mappings used to generate an undistorted rectified stereo image pair from a stereo image pair<sup>[6]</sup>.

Since the cameras are at a fixed position, mappings calculations only need to occur once. To use the mappings, the writer can apply them to a raw stereo image pair by using the function `cv2.remap()` to get the corresponding undistorted, rectified stereo image pair<sup>[6]</sup>.

As for the distance detection algorithm, it accepts undistorted rectified image pairs as input and outputs the distance to an object.

The steps in the algorithm are as follows:

1. Given a pair of undistorted rectified stereo images, calculate the disparity map using the Block Matching Algorithm<sup>[5]</sup>.

- a. OpenCV provides an implementation of this algorithm in a class called StereoBM. When using this class, users need to tune parameters such as numDisparities and blockSize to achieve an optimal disparity map.
- 2. Given the equation  $disparity = x - x' = \frac{B*f}{Z}$  (where Z = depth, and  $x - x'$  = disparity between two images, B = baseline of stereo camera setup, and f = focal length), derive B and f [5].

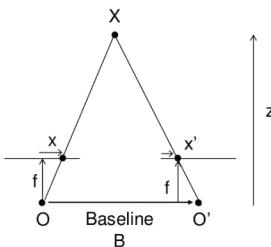


Figure A: The relationship between depth and disparity<sup>[5]</sup>

- 3. Convert the disparity map to a depth map using the equation  $depthmap = \frac{B*f}{disparity map}$ . This depth map will be used to help depth perception<sup>[5]</sup>.
- 4. Given a minimum depth value (i.e. threshold for the safe distance behind the micro-mobility user), determine regions in the depth map where the depth value is too low<sup>[5]</sup>.
- 5. Find the largest contour within the defined region(s), and calculate the average depth of the object.
- 6. Return the depth value in cm and use this value to determine whether to trigger haptic feedback.

In the final iteration of the product, this distance detection algorithm was replaced by the one described in section 3.1.3.2.

## Appendix B - Cameras

In the first iteration of this product, StreetSmart used Anker PowerConf C200 2K HD cameras for distance detection. These cameras boast features such as 2K resolution at 30 fps, a 95° field of view, and a focus distance of up to 3 m<sup>[7]</sup>. The high resolution allows the camera to capture more detail, which helps the distance detection algorithm accurately find the objects of interest/danger. Furthermore, having a field of view of 90° and a visual range of 3 m are functional and non-functional specifications respectively, and these criteria are fulfilled by this camera.

## Appendix C - Buck Converter

Designing the LM2596S buck converter would provide the option to tune the output voltage ripples and efficiently organize components within the enclosure and PCB. Following the manufacturer's notes, the capacitors and inductor are selected to mitigate the output voltage ripple to 1% <sup>[18]</sup>, where Figure C illustrates the designed buck converter circuit. The output voltage is selected via the voltage divide of R1 and R2 and can be calculated using Equation (3).

$$V_o = 1.23 * \left(1 + \frac{R_2}{R_1}\right) \quad (3)$$

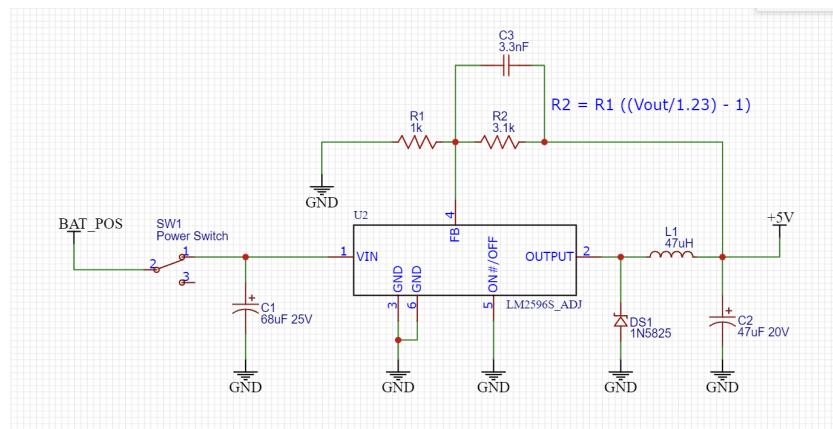


Figure C: 5V Output Buck Converter using the LM2596S Switching Regulator

## Appendix D - Boost Converter

Designing the XL009 boost converter allows for a custom layout to fit in a small form factor on the PCB. Following the manufacturer's recommended capacitor and inductor selection<sup>[20]</sup>, Figure D illustrates the designed boost converter circuit. The output voltage is selected via the voltage divide of R1 and R2 and can be calculated using Equation (4).

$$V_o = 1.25 * \left(1 + \frac{R_2}{R_1}\right) \quad (4)$$

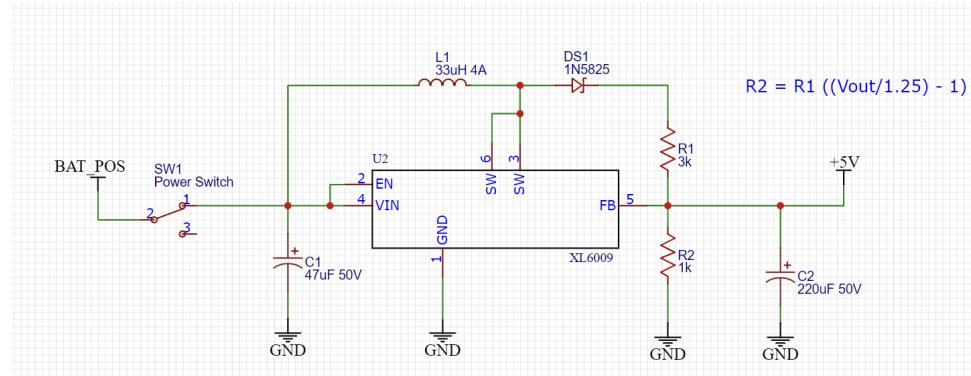


Figure D: 5V Output Boost Converter using the XL6009 Switching Regulator

## Appendix E - Previous Mechanical Designs

Below, diagrams describing previous mechanical encasing designs are shown for reference of the iterative process.

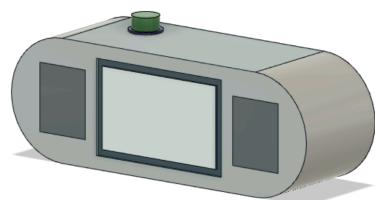
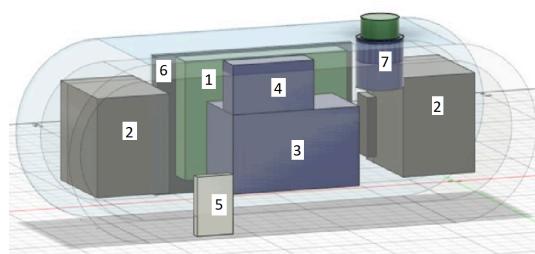


Figure E.1: External Schematic of Central Controller



- Legend**
1. Raspberry Pi
  2. Cameras
  3. Battery
  4. Buck Converter
  5. Charging circuit
  6. LEDs
  7. Toggle switch

Figure E.2: Internal Schematic with Hardware Components

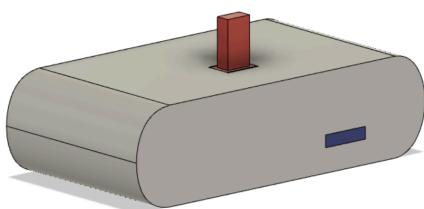


Figure E.3: External Schematic of Haptic Controller

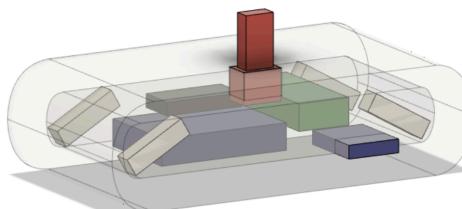


Figure E.4: Internal Schematic with Hardware Components

- Legend**
- Red: Toggle switch
  - Green: Arduino
  - Blue: Battery + Charging Circuit
  - Black: Wireless module
  - Grey: Haptic Motors