ECE 445

SENIOR DESIGN LABORATORY

DESIGN DOCUMENT

Cycling Assist System with Rear Camera Detection

Team #12

JACOB BETZ

(jmbetz2@illinois.edu)

TRISHA YADAV

(tyadav4@illinois.edu)

JINGDI LIU

(jingdil2@illinois.edu)

TA: Kai Chieh Chang

September 26, 2022

Contents

1 Introduction **1**

1.1 Problem 1

1.2 Solution 1

1.3 Visual Aid 2

1.4 High-Level Requirements List 2

2 Design and Requirement **3**

2.1 Block Diagram 3

2.2 Physical Design 3

2.3 Functional Overview & Block Diagram Requirements 5

2.3.1 Power Subsystem 5

2.3.2 Rear View Camera + Object Detection 6

2.3.3 Dashboard Warning System 8

2.4 Hardware Design 10

2.4.1 Circuit Schematic 10

2.4.2 PCB Design 11

2.5 Software Design 12

2.5.1 Raspberry Pi Object Detection 12

2.5.2 STM32 Software 13

2.6 Tolerance Analysis 14

2.6.1 Operating Voltage and Current Calculations 14

3 Cost and Schedule **17**

3.1 Cost Analysis 17

3.2 Schedule 19

4 Ethics and Safety **20**

References **21**

1 Introduction

1.1 Problem

Cycling is a very popular activity among many different age groups. With new eco-conscious initiatives launching across the United States, more and more people are opting for cycling as a mode of transportation. However, as common of an activity as it is, biking can also be extremely dangerous. Over 130,000 people are injured while biking in the United States every year [1]. Because more people are using cycling for transportation, there are even more bicycles riding on main roads. While new monitors and tracking systems for bicycles appear on the market all the time, there are a few safety features that have yet to be made.

A necessary feature for frequent cyclers is rear view detection. Whether cycling on the main road or a busy trail, blind spot detection could be imperative to the safety of cyclers. Along with this, while there are many different detection systems available, many require you to order separate sensors to monitor different features. There are very few systems that encompass different cyclist safety features in one compact dashboard. For example, the Garmin Varia RTL515 uses RADAR blindspot detection [2]. However, in order to detect other features, another sensor must be bought and installed in parallel to this sensor.

1.2 Solution

In order to solve this issue, a sensor system paired with a user display system could be made to assist cyclists. The sensor system would be placed so that it faced behind the bicycle. This system would contain a rear camera made to show the cyclist the rear view on their dashboard. This will be fed into OpenCV with object detection so that the system can detect other vehicles and bikes. This camera will be a small 1080p USB camera. This sensor would be held by a 3D-printed enclosure. The camera enclosure will then mount to the back of the bicycle seat. Along with this system, a custom dashboard will be created to display the aforementioned cycle rear view. This dashboard would be a 3.5-inch LCD display mounted on the handlebars of the bike. To give warnings regarding imminent danger, there will be an array of LED indicator lights to warn cyclists how close an object may be approaching. The dashboard would include an STM32 chip mounted directly on a PCB.

1.3 Visual Aid

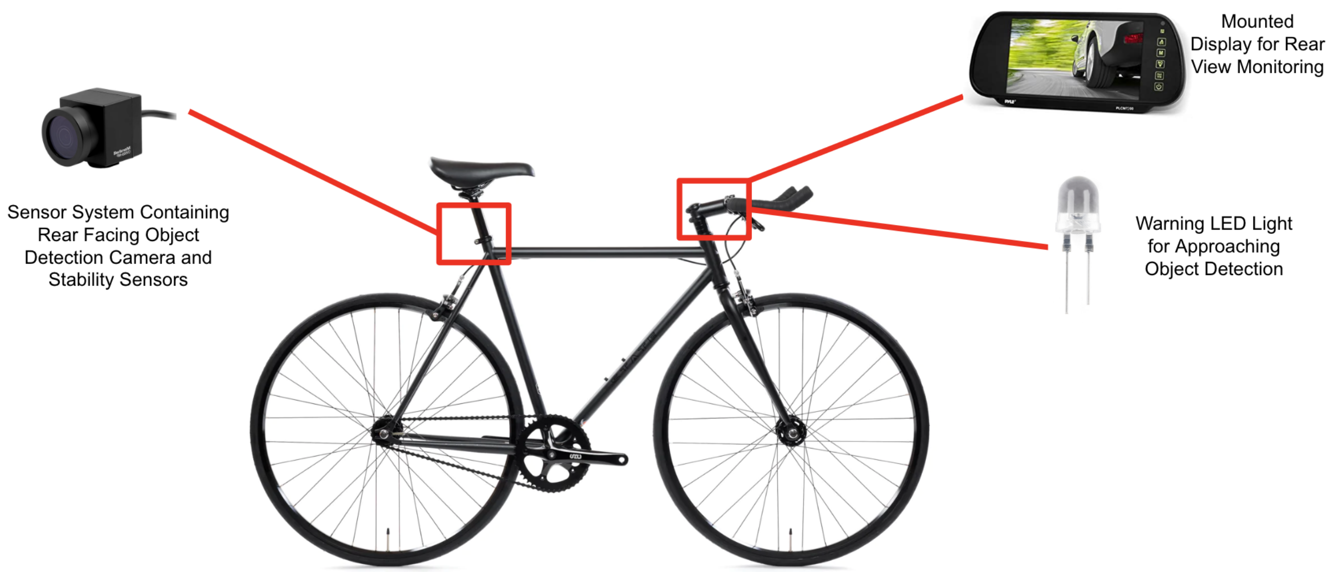

Figure 1: Visual Aid

1.4 High-Level Requirements List

Three quantitative characteristics that this project must exhibit in order to solve the problem mentioned include:

1. The system needs to be able to use a camera and display combination to show a live rear view of the bicycle while riding.
2. Device needs to detect rear approaching objects within at least 10 meters using object detection libraries and algorithms.
3. Device needs to be able to inform the user using LEDs and a buzzer how close a vehicle is to the cyclist and if an object was detected in the bicycle's blind spot.

2 Design and Requirement

2.1 Block Diagram

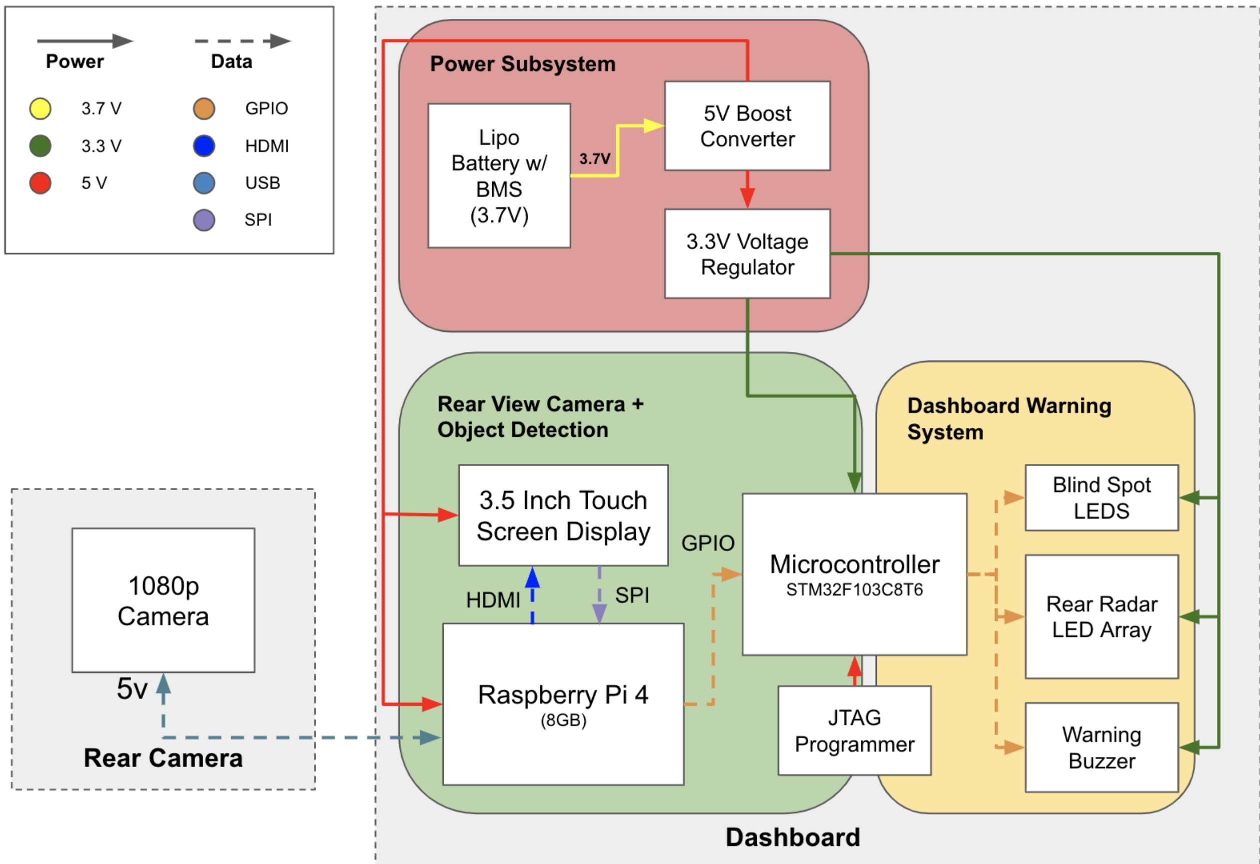

Figure 2: Block Diagram

2.2 Physical Design

The physical design on the dashboard includes an entirely custom 3D printed enclosure, made out of PLA, that will hold the Raspberry Pi 4, Amp Ripper 3000 boost converter, custom PCB and battery securely in place on the bicycle. The dashboard will be mounted around the front handle bars of the bicycle and its total dimensions are 8x5x2.7 inches (108 cubic inches).

To secure the dashboard to the bicycle, hose clamps will wrap through the open loop on the underside of the dashboard, attaching to the cylindrical structure of the bicycle below it. Also note the square shield surrounding the USB ports of the Raspberry Pi 4. This will be used to cover the Raspberry Pi's outputs as well as route the USB camera cable to the rear of the bicycle.

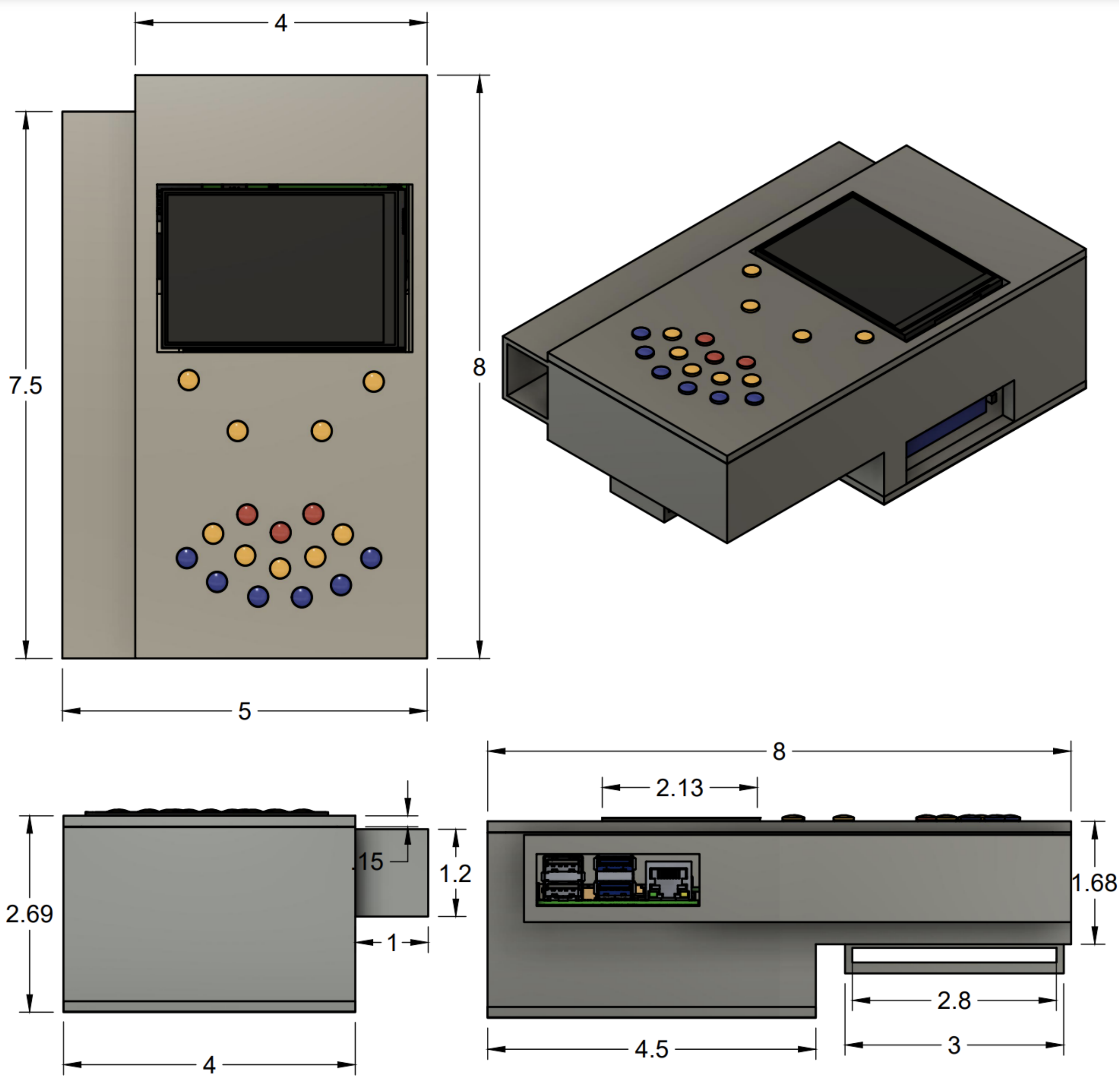

Figure 3: Physical Design

2.3 Functional Overview & Block Diagram Requirements

2.3.1 Power Subsystem

The power subsystem is responsible for supplying power to our entire project. Due to the nature of our project, various voltages are needed to power our entire system. The power subsystem will feature a 10Ah Lithium Polymer battery, a unique 5V Boost Converter built specifically for Raspberry Pi 4 based projects, and a 3.3V linear voltage regulator. The Lithium Polymer battery will provide the power at 3.7V and with a high continuous discharge rate of at least 2C. This will connect through a JST connector to the Amp Ripper 3000 Boost Converter that will supply a continuous 2A current through the USB-C output. Using the onboard pins, it will also be the 5V supply to the PCB, providing a 5V input to the 3.3V linear voltage regulator (LD1117S33CTR) to power our LEDs, buzzer and microcontroller.

Requirements	Verification
• The power subsystem must be able to supply at least a 2.5A continuous current output and up to 3A total at peak loads.	• Apply a constant 2.5A load for 30 minutes (time of average bike ride). Monitor that the current stays at a constant 2.5A for all 30 minutes of operation. • Apply a constant 3A load for 10s. Monitor that the current stays at a constant 3A for all 10s of operation.
• The power subsystem must be able to supply both a 5V and 3.3V output to the system at a tolerance of ±10% each.	• Connect battery and boost converter to the system. Check voltage using multimeter that no output voltage falls outside of 5V±10% and 3.3V±10%.
• The battery must be able to be easily recharged via USB input and will stop when the battery reaches 4.2V. The battery must be able to last at full charge for at least 3 hours.	• Start with battery voltage around 3.7V, connect USB cable to charging input of boost converter, monitor current delivered to battery and monitor battery voltage. Verify that the charging stops when the battery reaches 4.2V. • Apply constant 2.5A load and record time it takes for battery voltage to drop under 3.7V. Verify that this time is at least 3 hours.

2.3.2 Rear View Camera + Object Detection

The Rear View Camera and Object Detection subsystem is the core sensing and processing part of our project. This subsystem is responsible for capturing the live video from the rear of the bicycle, displaying the live video on our 3.5 inch display and processing the video using complex computer vision algorithms and libraries. The subsystem will feature a unique USB camera capable of filming in both day and night conditions using an IR system. The camera will connect directly to the Raspberry Pi 4 (8gb model) to be displayed and processed. The 3.5 inch display should show the cyclist clearly what is behind him.

The Raspberry Pi 4 will feature both custom algorithms for processing the video and computer vision libraries, namely OpenCV [3] and TensorFlow Lite, to detect the objects behind the cyclist. The entire process will use TensorFlow Lite to detect the vehicles, pedestrians and other cyclists behind the cyclist and OpenCV to determine distance and location of those objects behind the cyclist. Once the video has been processed, the Raspberry Pi 4 will send flags to the warning system to warn the cyclist when rear objects could be in a dangerous position.

Requirements	Verification
• The Raspberry Pi 4 must be able to drive the 3.5 inch display at a frame rate of 10fps or higher with the live video from the rear camera.	• Connect the rearview camera to the Raspberry Pi 4 via USB connection and connect the 3.5in display to the Raspberry Pi. Verify using the Raspberry Pi's onboard fps counter that the display consistently displays at least 10fps when viewing the camera image.
• The Raspberry Pi must be able to use object detection libraries and algorithms to detect moving objects behind a cyclist and shine the correct LED array depending at 2m, 5m and 10m ±20% each.	• Have a pedestrian carefully run behind the cyclist at 2m, 5m and 10m. Verify using a multimeter, that each of the GPIO pins for the 2m, 5m, and 10m flags send a high signal depending on the pedestrian's location.
• The Raspberry Pi must be able to send flagged objects to the STM32 microcontroller through GPIO pins.	• Have a pedestrian stand behind the cyclist at 5m. Use a multimeter to verify that the GPIO pin for a 5m warning is sent.
• The rearview camera must be able to capture video at a resolution of at least 480 x 380 in both day and night conditions.	• Connect the rearview camera to the Raspberry Pi in both day and night conditions. Verify that the pixel count in OpenCV for the video is at least 480 x 380.

2.3.3 Dashboard Warning System

The dashboard warning system will be built on a custom PCB and will feature multiple LEDs, an STM32 microcontroller and a buzzer to warn the cyclist effectively. The STM32F103 series microcontroller has the necessary power to drive all our LEDs and buzzer as well as connect to the Raspberry Pi through GPIO pins. The STM32F103 series can supply up to 25mA per GPIO pin and runs at a 3.3V input. Since the project will feature a total of 16 Super Bright White LEDs (20ma), MOSFETs will be used to power each set of LEDs at one time. Along with the warning

LEDs, a piezoelectric buzzer will be used when objects are dangerously close to the cyclist. The microcontroller will control which set of LEDs and when the buzzer powers by using flags sent through the GPIO connection to the Raspberry Pi. Finally, to quickly program the STM32F1032 microcontroller, a tag connect programmer and ST LINK-V2 will be used to quickly reprogram the processor. All of these connections can be seen in the circuit schematic below.

Requirements	**Verification**
• The STM32 must be able to source each gate of the MOSFET to turn on each set of LEDs: L Blind Spot, R Blind Spot, 2m LEDs, 5m LEDs, and 10m LEDs within 500ms when conditions are met.	• Using a voltage supply, supply a 3.3V input to one of the GPIO flags from the Raspberry Pi to the STM32. Using a camera with high capture rate, verify that as the input is set the LEDs turn on within 500ms.
• Verify that when conditions are met for the Piezoelectric buzzer to sound that the buzzer is at least 50dB. Verify that the buzzer is under the safe level of 120dB.	• Apply a 3.3V input to the buzzer inside the dashboard enclosure. Using a decibel meter, verify that the sound at the average cyclist head level is within the bounds of 50dB and 120dB.
• Verify that when the LED warning indicators are turned on, it can be seen in both day and night conditions.	• Have a pedestrian stand behind the cyclist at 5m in both day and night, verify that the LED light is visible even in sunny conditions.

2.4 Hardware Design

2.4.1 Circuit Schematic

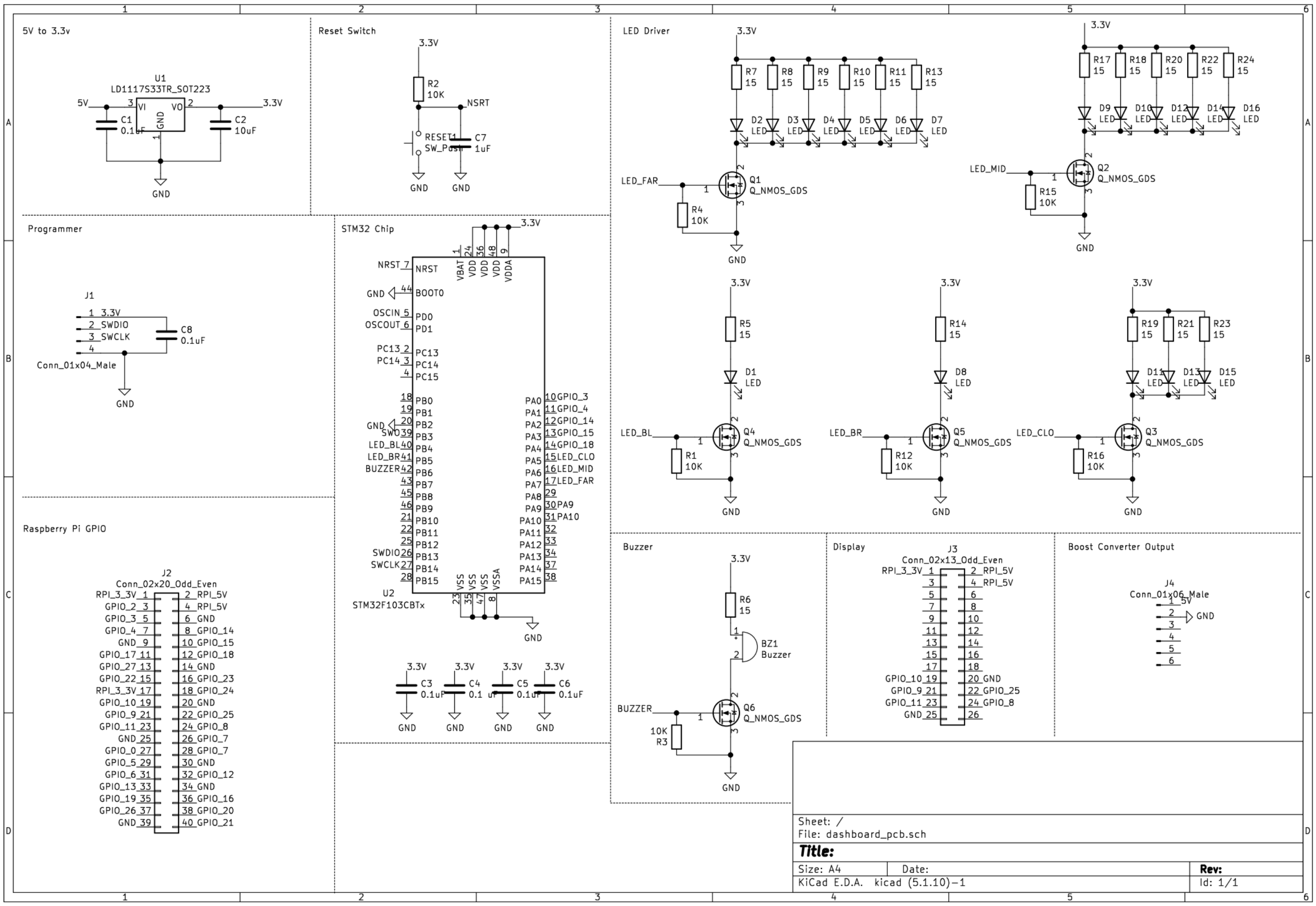

Figure 4: Circuit Schematic

2.4.2 PCB Design

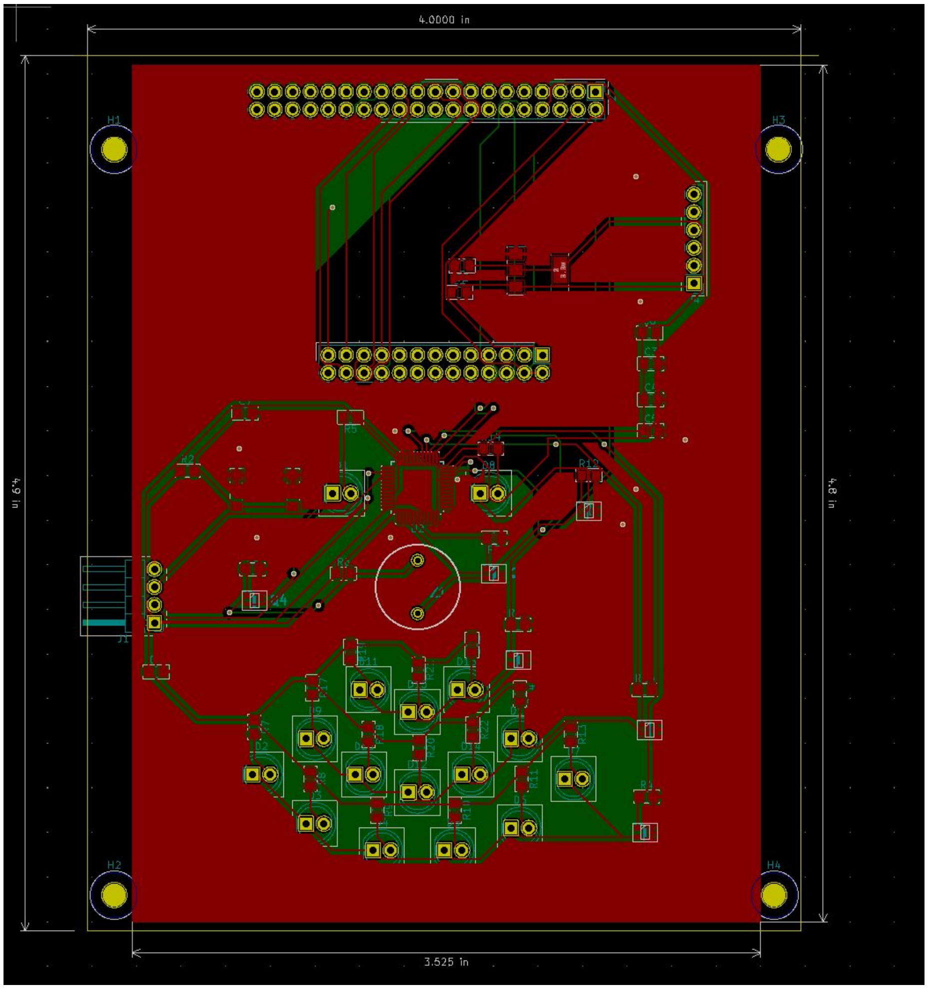

Figure 5: PCB Design

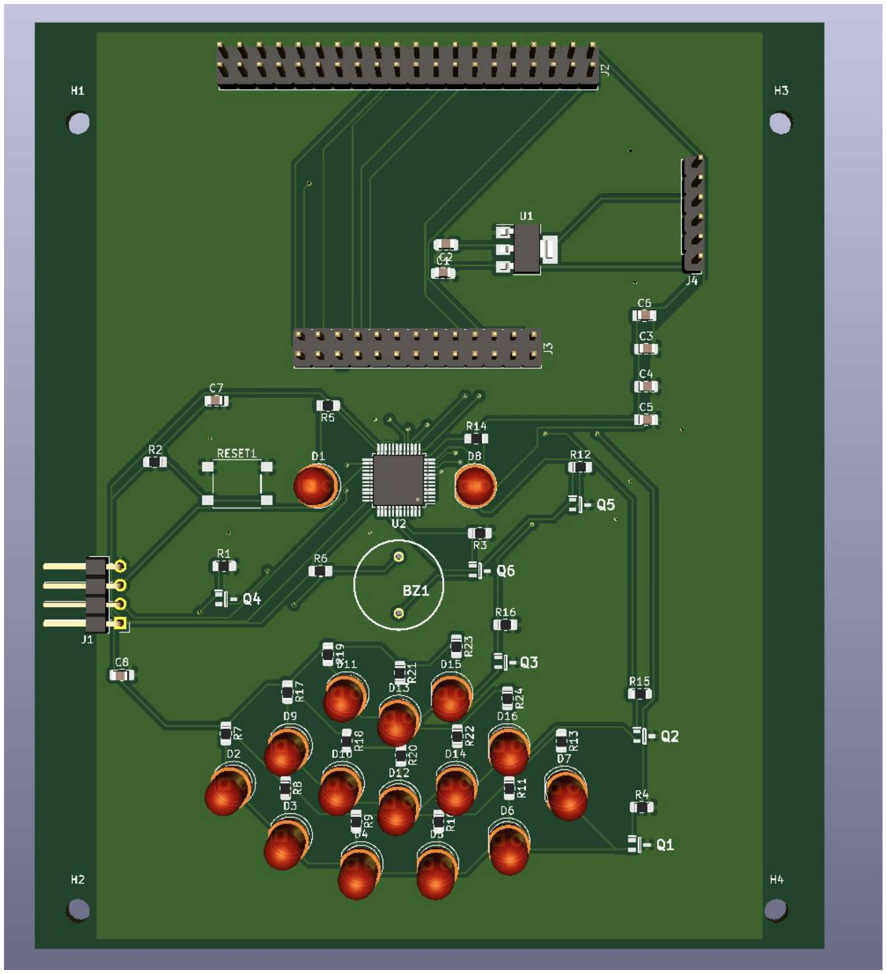

Figure 6: PCB Design

2.5 Software Design

2.5.1 Raspberry Pi Object Detection

The Raspberry Pi will take input from the ArduCam USB camera and perform object detection while estimating the distance to the object. The image will be preprocessed first by OpenCV [3], and will then be the input for the TensorFlow Lite model. The model will be trained to detect objects behind the cyclist such as vehicles, pedestrians, and other cyclists with more than 45% accuracy while keeping the video output on display to be more than 10fps. Then the OpenCV will utilize triangle similarity to estimate the distance between cyclist and detected objects. The triangle similarity method for distance estimation works as follow: We will need an object (paper) with known width W. We then place this marker at some distance D from the camera. We will take a picture of the object using the camera and measure the apparent pixels width P. The perceived focal length F of the camera can be calculated as $F = (P \times D) / W$ Use the calculated focal length F to estimate the distance to the detected objects with $D' = (F \times W) / P$ Once the distance has been calculated, warning signals will be generated by Raspberry Pi 4 to the STM32 IO based on the distance estimated to be less than 2m, 5m, and 10m.

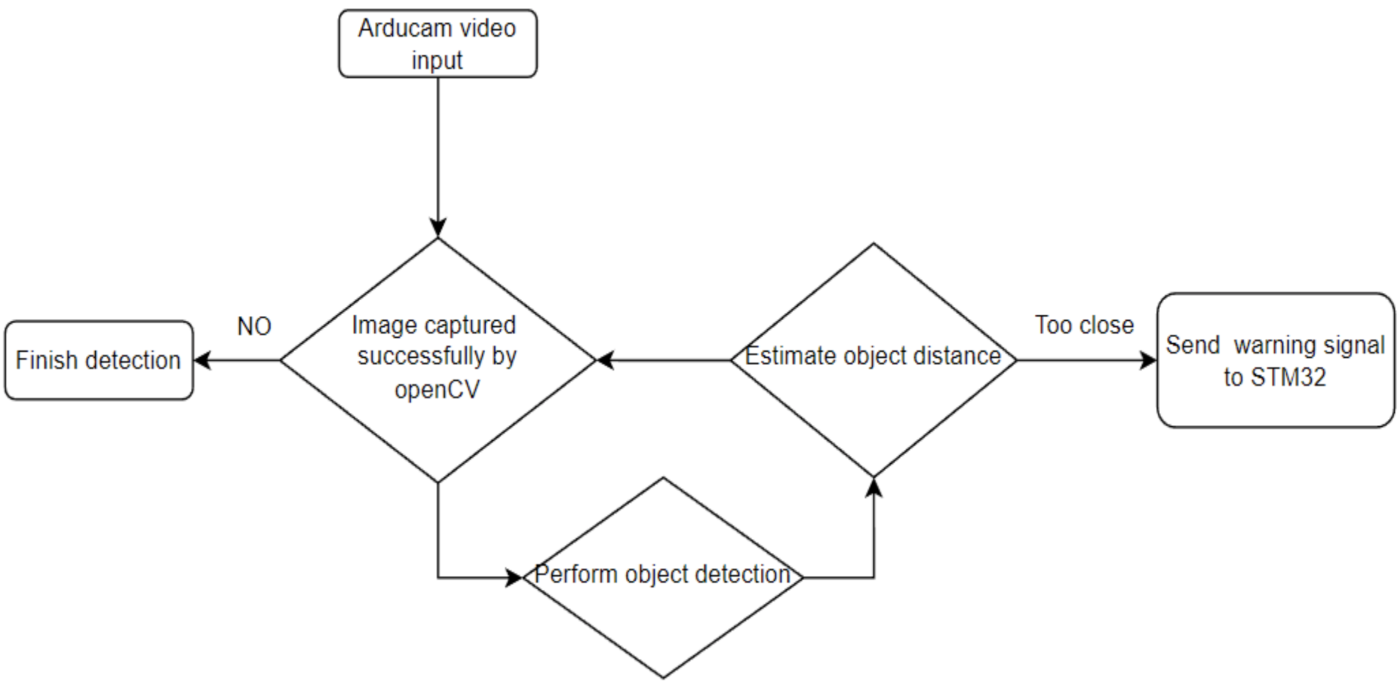

Figure 7: Raspberry Pi Object Detection Algorithm Flow Chart

2.5.2 STM32 Software

The STM32F103C8T6 is a low cost, low power microcontroller that will be used to drive the warning system LEDs and buzzer on the dashboard. The STM32 HAL integrated library will make it possible to drive the GPIO pins to connect to the various warning devices. A custom loop will be made when the system is powered on. The loop will monitor all input IO from the Raspberry Pi GPIO and will decide using the loop below when to power the MOSFET gates for the warning devices.

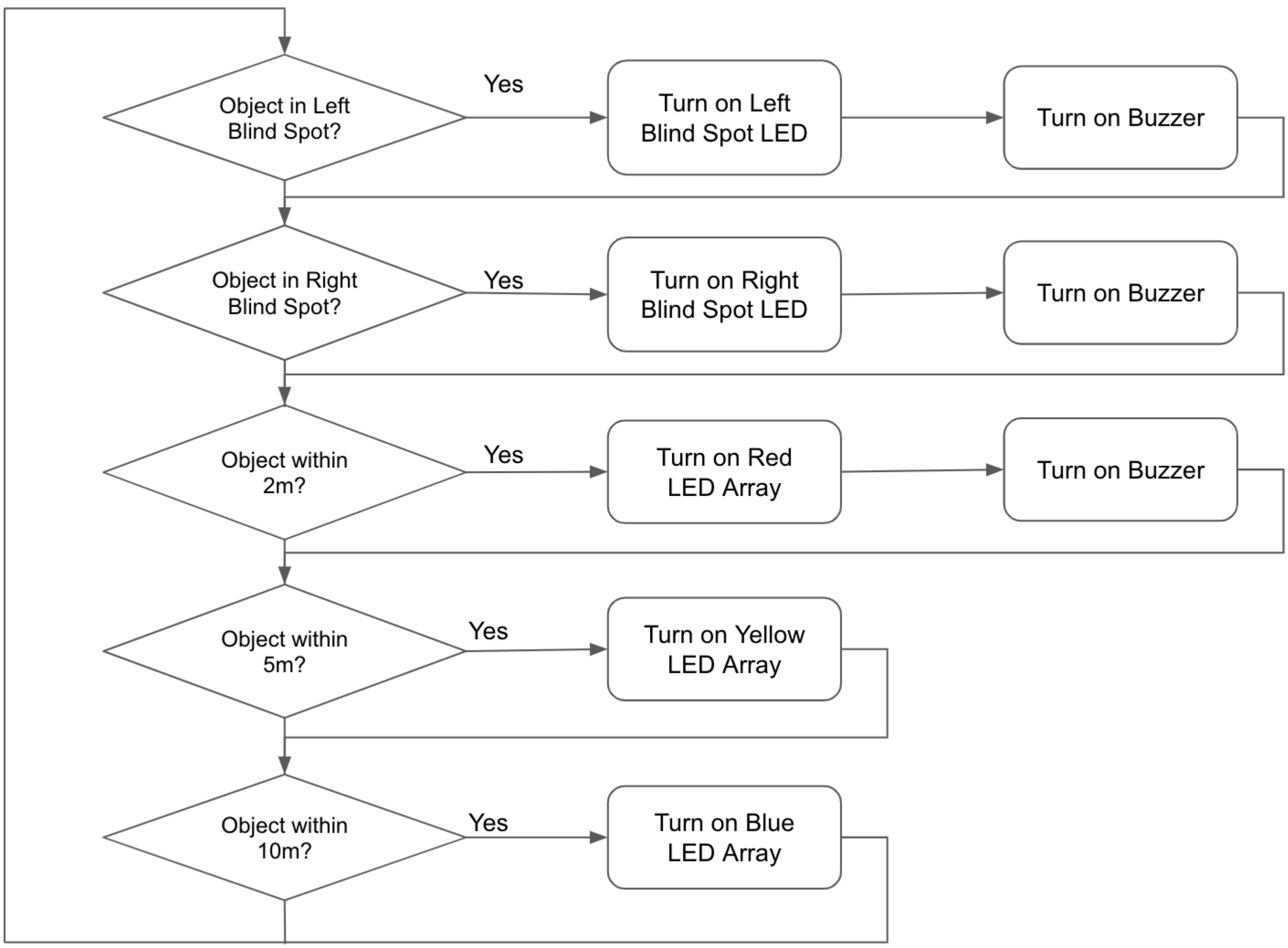

Figure 8: STM32 Microcontroller Software Flow Chart

2.6 Tolerance Analysis

2.6.1 Operating Voltage and Current Calculations

One of our most challenging portions of this project is the power requirement. We have a unique system that has a high power computer processing video therefore the Raspberry Pi 4 requires a lot of power. We also have an entire other low power microcontroller that is responsible to controlling many different warning devices. Therefore a complex analysis of our power requirements is crucial for our success in this project.

The Raspberry Pi 4 requires at least 5v at 2.5A [4] with a USB device less than 500mA. In our case, our Arducam USB camera has a maximum current of 370mA and our 3.5 inch Touch Display has a maximum current of 120mA [5], meaning that our Raspberry Pi 4 will need a 5V power supply at least 2.5A of continuous current.

Next our STM32F103 microcontroller requires a 0.3-4V max main supply voltage at a maximum current of 150mA [6]. Therefore a 3.3V supply up to 150mA would be sufficient to power our STM32F103. Each Super Bright LED has a current consumption of 20mA a piece [7]. With 16 total LEDs, that is 320mA of total current consumption. Finally our piezoelectric buzzer has a maximum working current of 10mA [8]. This brings our total Dashboard Warning Subsystem current consumption to 480mA.

Another component to include in our operating voltage and current calculation is the 3.3V linear voltage regulator. Since our requirement for our Dashboard Warning subsystem is 480mA, the LD1117-3.3 works perfectly [9]. It has an input voltage range of 4.3V-15V and a maximum current of 800mA. It has a maximum current consumption of 10mA, therefore this regulator will fit our requirements for the warning system.

Therefore our total current consumption for our entire project is 2985mA or roughly 3A maximum current consumption.

Subsystem	**Component**	**Input Voltage**	**Maximum Current Consumption**
Object Detection	Raspberry Pi 4 (8GB)	5V	2500mA
Object Detection	3.5 Inch Display	5V	120mA
Object Detection	ArduCam USB Camera	5V	370mA
Dashboard Warning	STM32F103C8T6	3.3V	150mA
Dashboard Warning	Super Bright LEDs	3.3V	16x20mA = 320mA
Dashboard Warning	Piezo Buzzer	3.3V	10mA
Power Subsystem	LD1117-3.3	5V	5mA

Total Maximum Current Consumption: 2985mA

Total Maximum Power Consumption: 14.925W

The Amp Killer 3000 Boost converter can supply 5V at a continuous 3A (with sufficient cooling). Therefore even at peak loads, the Raspberry Pi 4 will not be throttled. And it can supply the entire project system with the necessary power.

The final component to include is the battery. The Amp Killer 3000 requires a battery that has a 3.7V-4.2V input and at least 3000mAh capacity. We have chosen a 3.7V Lithium Polymer with 10000mAh capacity. This will be able to supply the boost converter with the correct voltage and will have enough capacity for several hours of use, even at peak consumption.

From this analysis we learned that our power requirements are met in our system, even when every single device is active and under peak load (a situation that should theoretically never happen). However, it is easy to see how not computing the requirements and analysing the outcome could have had consequences to the performance of our project. For example, if the Raspberry Pi 4 does not meet its power requirements the performance will be throttled. This could cause the object detection algorithm's accuracy to drop and miss a moving object.

Benchmarks and power measurements results [10] can be helpful for us to estimate the power consumption for our Raspberry Pi 4 performing object detection. Under an idling condition, the Raspberry Pi consumes 575mA with a 5V input voltage. And the consumption peaked at roughly 885mA when loading IXDE. The 1080p video shooting power consumption was measured to be 640mA under required input voltage. Since our Raspberry Pi will be required to perform at least 480p video shooting, object detection, and video display with 10+ fps, an estimated power consumption for these tasks should be about 2500mA.

3 Cost and Schedule

3.1 Cost Analysis

The first calculation needed in order to determine the total cost of this project is the cost of all the components needed to complete the project. The sum of all the components listed in the figure below is $248.02. After this, we must calculate the labor costs associated with the project as well. The average starting salary for a computer engineering graduate is $105,352 [2]. With a 40 hour work week, and 52 weeks in a year, this salary equates to about $50.65/hr. The time from project approval to the end of this course is about 12 weeks, and the team will work about 8 hours a week. There are three members in our team. Therefore, the cost of the labor will be: $50.65/hr x 2.5 x 60 hours * 3 team members = $22792.50 This makes the total the sum of the component costs and the labor costs: $22792.5 + $248.02. The total cost of this project is $23040.52.

Description	Manufacturer	Quantity	Cost	Link
1080P Day and Night Vision USB Camera	Arducam	1	34.99	Link
3.5″ LCD Touch Screen Display	Waveshare	1	19.80	Link
Break-away 0.1″ 2x20-pin Strip Dual Male Header	Adafruit	1	0.95	Link
DIY HDMI Cable Parts - Right Angle (R bend) Mini HDMI Plug	Adafruit	1	6.50	Link
DIY USB or HDMI Cable Parts - 10 cm Ribbon Cable	Adafruit	1	1.50	Link
DIY HDMI Cable Parts - Straight Mini HDMI Plug Adapter	Adafruit	1	6.50	Link
USB 3.0 Adapter 90 Degree Male to Female Coupler Connector Plug Left Angle and Right Angle	Oxsubor	1	6.99	Link
Raspberry Pi 4 Model B - 8 GB RAM	Adafruit	1	75.00	Link
Super Bright White 5mm LED (25 pack)	Adafruit	1	6.95	Link
CLB 300 Series Round Lens for 5mm LED, 0.250-Inch/6.5mm Diameter, Yellow	VCC	1	5.14	Link
CLB 280 Series Round Lens for 5mm LED, 0.250-Inch/6.5mm Diameter, Amber	VCC	1	10.06	Link
CLB 300 Series Round Lens for 5mm LED, 0.250-Inch/6.5mm Diameter, Blue	VCC	1	11.49	Link
3.7V 10000mAh 1165114 Lipo Battery Rechargeable Lithium Polymer ion Battery with JST Connector	AKZYTUE	1	25.99	Link
AmpRipper 3000	Kickstart Design	1	24.99	Link
STM32F103CBT6TR Microcontrollers	Arrow	1	9.19	Link
BUZZER PIEZO 1.5V 22MM TH	Murata Electronics	1	0.73	Link
3.3V 800mA Linear Voltage Regulator - LD1117-3.3 TO-220	Adafruit	1	1.25	Link

Figure 9: List of Parts and Costs Needed for Project

3.2 Schedule

Date	Task	Person
10/3	Design Review, order PCB	Everyone
10/10	Buy all needed parts	Everyone
	Continue research on object detection software	Trisha and Jingdi
10/17	Receive PCB and start building	Jacob
	Begin STM setup	Jingdi
	Begin working on object detection prototype	Trisha
10/24	Finalize 3D printed enclosure	Jacob
10/31	First prototype object detection	Trisha
	Continue working on STM software	Jingdi
11/7	Test object detection	Trisha and Jingdi
11/14	Mock Demo	Everyone
	Finalize STM software and object detection	Trisha and Jingdi
	Work on presentation and report	Everyone
11/21	Fall break	-
11/28	Final demo, work on presentation and report	Everyone
12/5	Final presentation	Everyone

Figure 10: Schedule for Project

4 Ethics and Safety

There are some ethics and safety policies that should be considered carefully. The purpose of this project is to assist cyclists using a sensor system and a user display system to keep track of the rearview and stability of a bike. This purpose falls under safety standards by IEEE's Code of Ethics Section I.1, which is "to hold paramount the safety, health, and welfare of the public... and to promptly disclose factors that might endanger the public or the environment" [11]. This project aims to assist cyclists by giving effective warnings and informing them of their status while they are cycling. However, the system will not be able to physically assist cyclists by preventing accidents, which means the risk of cycling will not be eliminated. Therefore, group members will explicitly mention this to users before giving any further instruction, and this follows the IEEE's Code of Ethics Section I.5, which is to "acknowledge and correct errors, to be honest, and realistic in stating claims or estimates based on available data, and to credit properly the contributions of others"[11].

Furthermore, group members working together will respect each other and treat others fairly through frequent and effective communication either in person or online. This practice follows IEEE's Code of Ethics Section II.2, "to not engage in discrimination based on characteristics such as race... gender identity, or gender expression" [11]. Along with this, the team will make sure to follow all of the ECE lab safety rules, as stated in the university's Laboratory Safety Training by the Division of Research Safety.

Finally, the team will follow the COVID-19 CDC Guideline when planning to meet in person to work on the project. This fall under the COVID-19 CDC Guideline, "Reiterating that regardless of vaccination status, you should isolate from others when you have COVID-19" [12] and "Recommending that if you test positive for COVID-19, you stay home for at least 5 days and isolate... Wear a high-quality mask when you must be around others at home and in public" [12]. The team will also follow the Lab Safety Guidelines when working on PCB and circuits.

References

[1] "Bicycle Safety." (2022), [Online]. Available: https://www.cdc.gov/transportationsafety/bicycle/ (visited on 09/14/2022).

[2] "Garmin Varia™ RCT715." (), [Online]. Available: https://www.garmin.com/en-US/p/721258 (visited on 09/14/2022).

[3] "Open CV." (), [Online]. Available: https://opencv.org/ (visited on 09/28/2022).

[4] "Raspberry Pi 4 Model B." (2019), [Online]. Available: https://datasheets.raspberrypi.com/rpi4/raspberry-pi-4-datasheet.pdf (visited on 09/20/2022).

[5] "3.5inch HDMI LCD." (2022), [Online]. Available: https://www.waveshare.com/3.5inch-hdmi-lcd.htm (visited on 09/18/2022).

[6] "STM32F103x8." (2022), [Online]. Available: https://www.st.com/resource/en/datasheet/stm32f103c8.pdf (visited on 09/20/2022).

[7] "Super Bright White 5mm LED." (2022), [Online]. Available: https://www.adafruit.com/product/754?gclid=CjwKCAjwhNWZBhB_EiwAPzlhNmcI7ga2VqS-8DLy1G5JT_2It4aqkaQh0yR22mEY0RDuhuZCPVFu5xoCieIQAvD_BwE (visited on 09/20/2022).

[8] "PKM22EPPH2001-B0." (2022), [Online]. Available: https://www.digikey.com/en/products/detail/murata-electronics/PKM22EPPH2001-B0/1219322 (visited on 09/20/2022).

[9] "LD1117S33CTR." (2022), [Online]. Available: https://www.digikey.com/en/products/detail/stmicroelectronics/LD1117S33CTR/585765 (visited on 09/24/2022).

[10] "How much power does the Pi4B use? Power Measurements." (2019), [Online]. Available: https://raspi.tv/2019/how-much-power-does-the-pi4b-use-power-measurements (visited on 09/29/2022).

[11] "IEEE Code of Ethics." (2016), [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html (visited on 09/14/2022).

[12] "CDC streamlines COVID-19 guidance to help the public better protect themselves and understand their risk." (2022), [Online]. Available: https://www.cdc.gov/media/releases/2022/p0811-covid-guidance.html (visited on 09/14/2022).