ClearNav

March 11th, 2022

Dr. Mike Hegedus
Simon Fraser University
8888 University Drive
Burnaby, British Columbia, V5A 1S6

RE: ENSC 405W/ENSC 440 Design Specification – Motorcycle HUD System

Dear Dr. Hegedus,

The following document describes the Design Specifications for our product EZhud. Our goal is to give motorcycle riders a convenient and safe option for viewing navigation and speed control information while on the road. By using a heads up display (HUD), our solution guides riders in unknown urban environments while allowing them to keep their eyes towards the road, thus improving their situational awareness.

This document contains detailed specifications for all aspects of our system, describing the individual subsystems and how they are connected. Many components will be discussed, including how the device is mounted onto the helmet, the software running on the microprocessor as well as our connected web server. Our design choices will be made to meet the requirements listed in the requirements document, and each design item will be targeted for the three phases of our project.

Our team consists of a diverse set of engineers from the computer, electronic and biomedical concentrations. Our members include Taimoor Ahmed, Muhammad Ahmed Athar, Namsakhi Kumar, Spencer Lall, William Huong and William Xue. Our multifaceted skill sets will allow us to engineer reliable systems and deliver clear documentation.

ClearNav would like to thank you for taking the time to read our design specification document. If there are any questions or concerns, please do not hesitate to contact me at zjxue@sfu.ca.

Sincerely,

William Xue

William Xue
Chief Executive Officer
ClearNav

ClearNav

Design Specification

EZhud

Partners: Spencer Lall
William Xue
William Huong
Namsakhi Kumar
Muhammad Ahmed Athar
Taimoor Ahmed

Contact: William Xue
zjxue@sfu.ca

Submitted To: Dr. Mike Hegedus
ENSC 405W
School of Engineering Science
Simon Fraser University

Issue Date: March 11, 2022

Abstract

This document outlines the design specifications for each subsystem and component of EZhud: a smart HUD for motorcyclists. Alternative design options are explored and justifications for the final design decisions are provided with reference to requirements in the requirements specification document. Appendices for appearance and user interfaces and detailed test plans are also included in this specification.

List of Figures

List of Tables

Table of Contents

Version History

Table 0.1: Version History

Date	Version	Notes
2022-03-13	1.0.0	Published

Glossary

Acronym	Definition
ABS	Acrylonitrile butadiene styrene
API	Application programming interface
CSI	Camera Serial Interface
FPS	Frames Per Second
GPIO	GPIO (General Purpose Input Output) pins enable the microcontroller to communicate with different components
GPS	Global Positioning System
Highway Speeds	Highway speeds in British Columbia – ranging from 80 km/h to 120 km/h [1]
HUD	Heads-up Display
LCD	Liquid crystal display
LCOS	Liquid crystal on silicon
Li-ion	Lithium-ion
MCU	Microcontroller unit
OLED	Organic light emitting diode
PETG	Polyethylene terephthalate glycol
PLA	Polylactic acid
REST	Representational state transfer
Rider	Motorcycle Rider in British Columbia
RPi 4	Raspberry Pi 4B
TPU	Tensor Processing Unit
TPU (3D Print)	Thermoplastic Polyurethane
VHB Tape	3M Very High Bond Tape

1 Introduction

Motorcycles lack the convenience of easily visible navigation and speed information that most car drivers are very much used to. Firstly, unlike driving a modern car, very few motorcycles possess a built-in infotainment system that supports turn-by-turn navigation. Secondly, glancing down at the speedometer on a motorcycle is much more dangerous than in a car and is a common factor for motorcycle accidents [2]. Lastly, many drivers are often unaware of the speed limit on the current road, especially because motorcycles lack a comprehensive infotainment system.

In response to these problems, ClearNav has designed EZhud: a smart heads-up display that attaches to most existing motorcycle helmets to display turn-by-turn navigation, current speed, and speed limit within your peripheral vision. The product will tackle the challenge of combining a powerful microcontroller, an in-helmet display, a front-facing camera module, and a quality battery into a single device with a small, lightweight form factor communicable by smartphone that provides the motorcyclist with useful information safely. A general overview of EZhud system is outlined in Figure 1.1

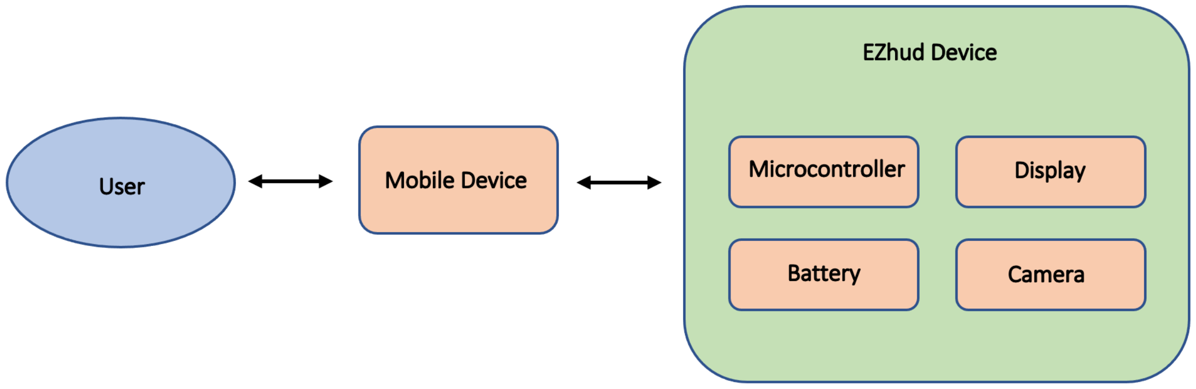


Figure 1.1: General Overview of the EZhud System

1.1 Scope

This document will outline the full design specification of the EZhud system including design choices, challenges, alternative design options, justifications for design, and any relevant models or predictions for each sub system of the product.

Design for the following subsystems of EZhud are considered

1. System Hardware Design
2. Back-End Software Design

More elaborate details are provided in Appendices

1. **Appendix A**: User interface and Appearance

        - Visual mock-ups of the EZhud device and it's user interfaces
2. **Appendix B**: Design alternative Appendix
    - Elaborated details on all design alternatives considered
3. **Appendix C**: Test Plan Appendix
    - Detailed test plan

1.2 Intended Audience

This document serves as EZhud's design requirements guide for the Company 11 team, its potential clients and partners, Dr. Mike Hegedus, Dr. Andrew Rawicz, and ENSC 405W / 440 teaching assistants. Any future design revisions or modifications will draw from the requirements detailed in this document.

1.3 Design Classification

The design requirements in this document will be categorized using the following convention:

Des <Section>.<Number> - <Design Phase Acronym>

The three product design phases are as follows

Table 1.4: Product Design Phases

Acronym	**Product Design Phase**	**Description**
A	(Alpha Phase) Proof-Of-Concept	Requirements that will be presented during the ENSC 405W demo
B	(Beta Phase) Engineering Prototype	All core components of the product are integrated together and function as a single unit.
C	(Production Phase) Final Product	System meets safety and sustainability standards and fulfills all requirements specified.

2 System Overview

EZhud is a helmet mounted HUD composed of the following major components: the display, a microcontroller, a battery, a camera, along with an accompanying smartphone app. These components provide real time graphical navigation and computer vision extracted speed limits to the rider in an unobtrusive manner. Figure 2.1 shows a high level overview of the full system. To avoid distracting the rider, the Sony ECX336CN OLED display was chosen for its diminutive size while still having resolution to be easily legible. Controlling the display will be a Raspberry Pi 4B, running Crankshaft, a head unit emulator built on OpenAuto [3], to receive Google Maps projections from a smartphone via Android Auto. In addition to navigation, speed limits will be displayed on the screen as well. Using video footage from a Pi Camera NoIR, speed limit signs will be processed by OpenCV and Tensorflow Lite, accelerated by a Coral TPU, and the speed limits extracted will be displayed alongside the maps and navigation on the screen. Powering all of this will be two 18650 Li-ion battery cells managed by a PowerBoost 1000C, which will be responsible for safely charging the 18650's and maintaining a stable 5V for the Raspberry Pi.

Off device an accompanying smartphone app will be provided to the rider. This app will provide two functions: allowing the rider to set the navigation route, and allowing the user to change EZhud system settings. For route setting, most of the work will be done through the Google Maps API. When the user searches for a location, the results from Google Maps will be returned, and selecting a location will start navigation on EZhud's screen via the Android Auto protocol. For system settings, communication between EZhud and the app will be through HTTP request to a Lighttpd web server running on the Raspberry Pi. The web server will be able to use Crankshaft's "Crankshaft Management Tool" to get and set system settings. These settings can be sent to the device when requested, or changed based on requests from the app.

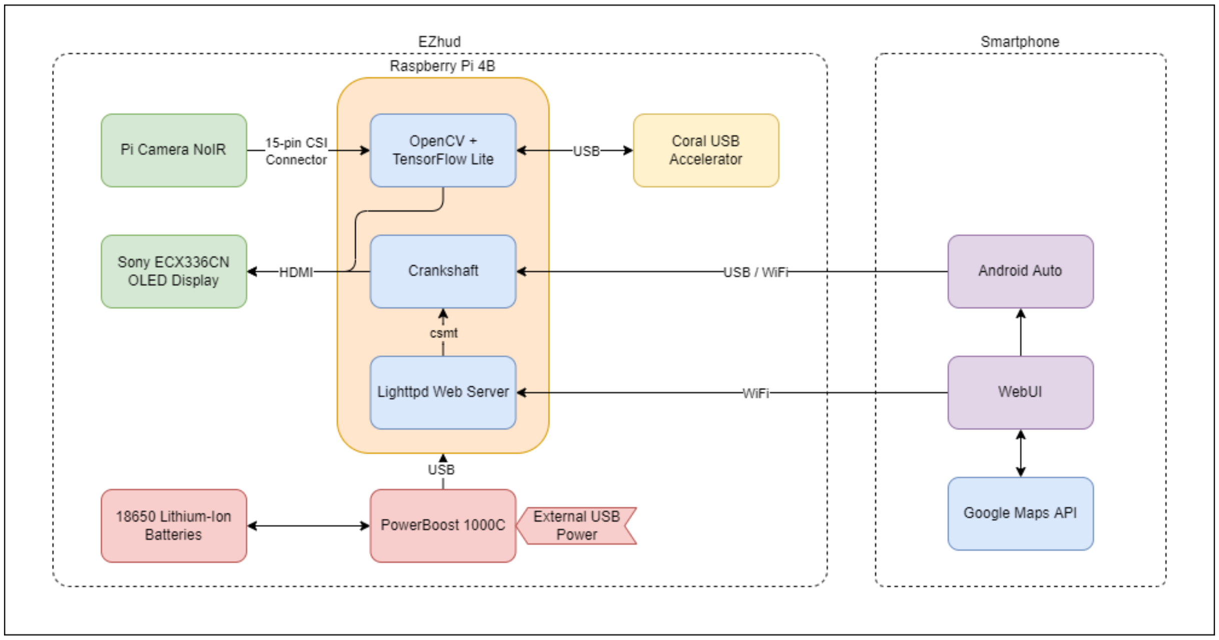


Figure 2.1: High Level System Diagram

3 Hardware Design

3.1 General Design Requirements

Table 3.1.1: General Design Requirements

Design ID	Design Description	Corresponding Requirements
Des3.1.1 - A	Power button to turn off/on the device and display	Req4.5 - A Req3.1.5 - C
Des3.1.2 - B	Mode button when pressed will switch between navigation, current speed and speed limit	Req5.1.3 - B
Des3.1.3 - C	Brightness button when pressed increases brightness by 20%	Req 6.3.3 - C
Des3.1.4 - C	LEDs indicating battery status	Req4.4a - C
Des3.1.5 - C	Device will cost between $500-$600	Req 4.9-C

To facilitate some of the above requirements for the user we have included three buttons on EZhud i.e. power button, brightness button and mode button to power on/off, toggle between display modes and brightness adjustment respectively.

For Indicating the battery status we plan on using the LED's in Adafruit Powerboost 1000C. Table 3.1.2 shows the LED,s and their corresponding status.

Table 3.1.2: Battery LED Status

LED	Status
Red	Low battery needs to charge
Yellow	Charging in process
Green	Battery is fully charged

The estimated price breakdown for each individual components of EZhud is listed in table 3.1.3 below:

Table 3.1.3: Estimated Cost Breakdown

Component	Price
Raspberry Pi 4B	$75
Coral USB Accelerator	$75
Sony ECX336 OLED	$320
Pi Camera Noir	$35
18650 batteries	$20
Adafruit Powerboost 1000C	$28
Total Estimated Cost	$553

3.2 System Design Requirements

The following main aspects will be considered in the system design requirements: protection of sensitive electronic components against wind and rain, the mounting of the main body and display to the helmet and the structure and shape of the main body.

To protect the electronic components within the main body, consisting of a Raspberry Pi 4B board, Raspberry Pi Camera NoIR, a Coral TPU, and the two 18650 Li-ion battery cells, a 3D printed housing will be designed to protect against environmental factors. The main 3D printed filaments considered include, PLA, ABS, PETG, and TPU [4].

When considering the EZhud, the chosen material must be strong, nontoxic, and suitable for extended outdoor use. The following details are taken into account when selecting a material:

Table 3.2.1: 3D Printed Material Considerations

ABS	For ABS filament, even though the material is strong and nontoxic, ABS is easily broken down by UV radiation, often indicated by becoming brittle and losing colour [5]. Thus ABS is not suited for our application.
PLA	PLA is a material made from biological materials and while tough, it begins to deform at temperatures higher than 60°C [5]. PLA is also not water resistant, thus unsuitable for EZhud.
PETG	PETG is known for being weather resistant, it is insensitive to changing temperatures (up to 80°C [6]) and UV rays [7]. The material does not deform in windy or rainy conditions and it is approved for food use (non-toxic)[5]. Thus very suitable for EZhud.
TPU	While TPU is abrasion resistant, and can withstand temperatures of up to 80°C [8],

	TPU, the material is difficult to print due to its high flexibility. Since EZhud requires a rigid body for protection, TPU's flexibility is not required in our design.

After taking the above comparisons into consideration, PETG is the most suitable material for our design. By using PETG as the enclosure material, the design requirements below can be fulfilled.

Table 3.2.2: Enclosure Design Requirements

Design ID	Design Description	Corresponding Requirements
Des3.2.1 - C	The enclosure shall not allow water into the device.	Req4.2a - C
Des3.2.2 - C	The enclosure shall not deform within an operating temperature range of -15°C to 80°C.	Req4.2b - C

When considering the placement of the device on the helmet, the main design choices included the placement of the front-facing camera, the aerodynamic properties of the device, and the center of mass of the helmet. Taking these into consideration, by placing the enclosure on the front right hand side of the full face helmet, the design can keep the camera forward facing, reduce aerodynamic drag and keep the center of mass at the bottom of the helmet.

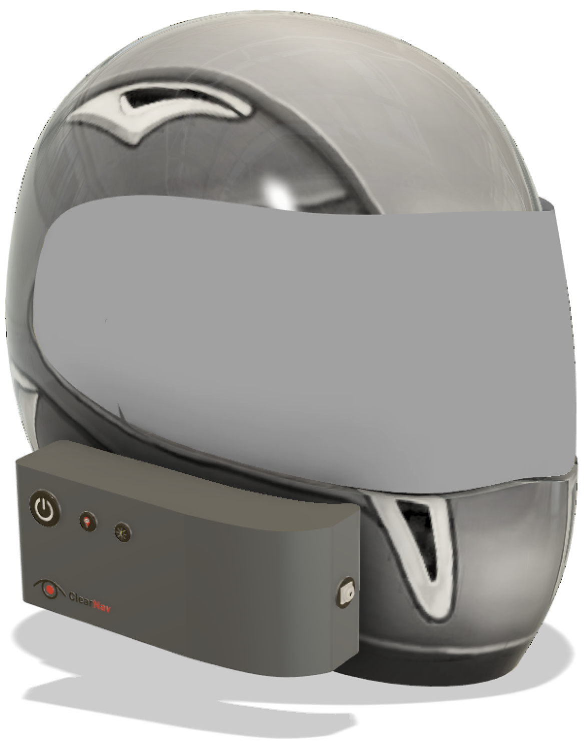


Figure 3.2.1 - EZhud Mounted on Helmet (Helmet From [9])

When considering mounting the device onto the helmet, VHB tape is a strong contender for such an application. Not only is 3M VHB tape relatively inexpensive, the adhesive is designed to permanently bond to any surface, including metal, wood, glass and plastics etc [10]. VHB tape also maintains its adhesive properties between the temperature of -40°C and 90°C and long term studies have shown that UV exposure does not cause the tape to lose bond strength [11].

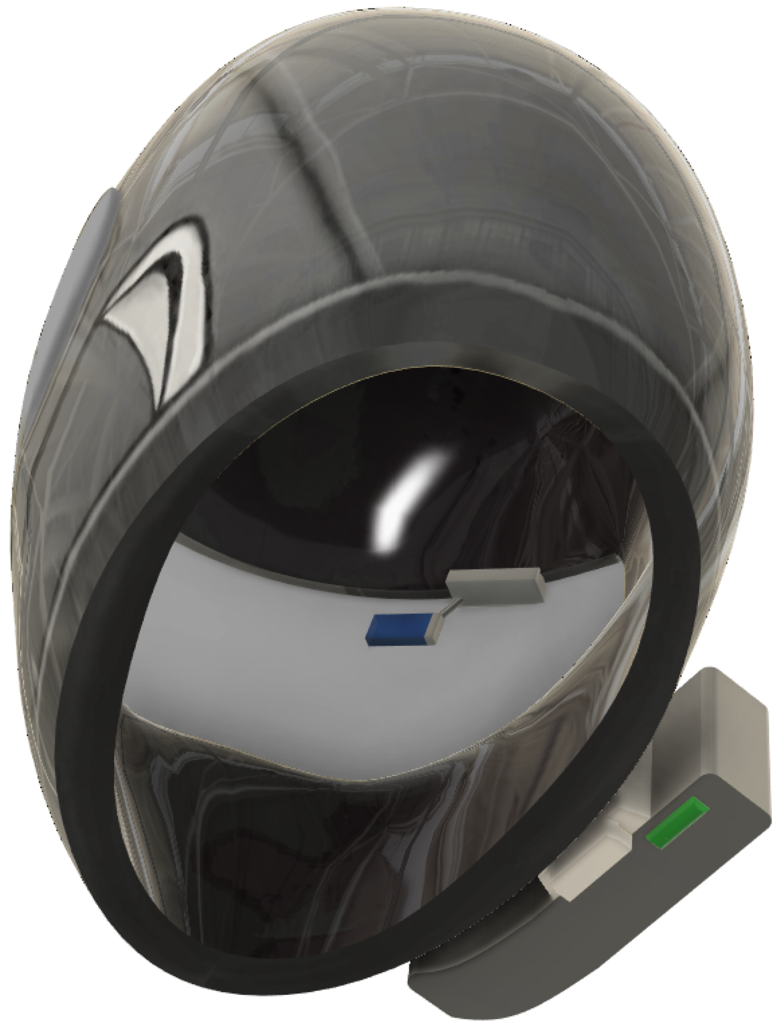


Figure 3.2.2 - Helmet Underside With Display (Helmet From [9])

Whilst the device is attached to the helmet via VHB tape, the device may be detached by depressing a mechanical button. Leaving only the clip permanently attached to the helmet. Thus, the user may be able to use the helmet without the EZhud device.

As shown in Figure 3.2.2, the display shall be mounted on the upper right-hand edge of the visor. The display is connected to a display body which will be attached via VHB tape to the inner edge of the visor. By using a micro-display, such as the Sony ECX336, the display shall not block much of the rider's peripheral view. A micro-HDMI cable shall connect the main microprocessor on the main body, to the display within the helmet.

Table 3.2.3: General System Design Requirements

Design ID	Design Description	Corresponding Requirements
Des3.2.3 - B	The PETG 3D printed enclosure shall house the microprocessor, camera, external TPU, and battery packs.	Req4.1a - B
Des3.2.4 - B	The display shall be mounted on the upper right-hand edge of the visor.	Req4.1b - B Req3.1.1a - B
Des3.2.5 - B	The enclosure shall be placed on the front right hand side of the full face helmet. The shape of the enclosure shall hug the outer wall of the helmet.	Req4.1c - B Req3.1.3 - C
Des3.2.6 - C	The main body of the device shall be attached to the helmet via VHB tape.	Req4.2c - C
Des3.2.7 - B	The main body shall be detachable from the helmet via pressing a mechanical button.	Req4.1c - B

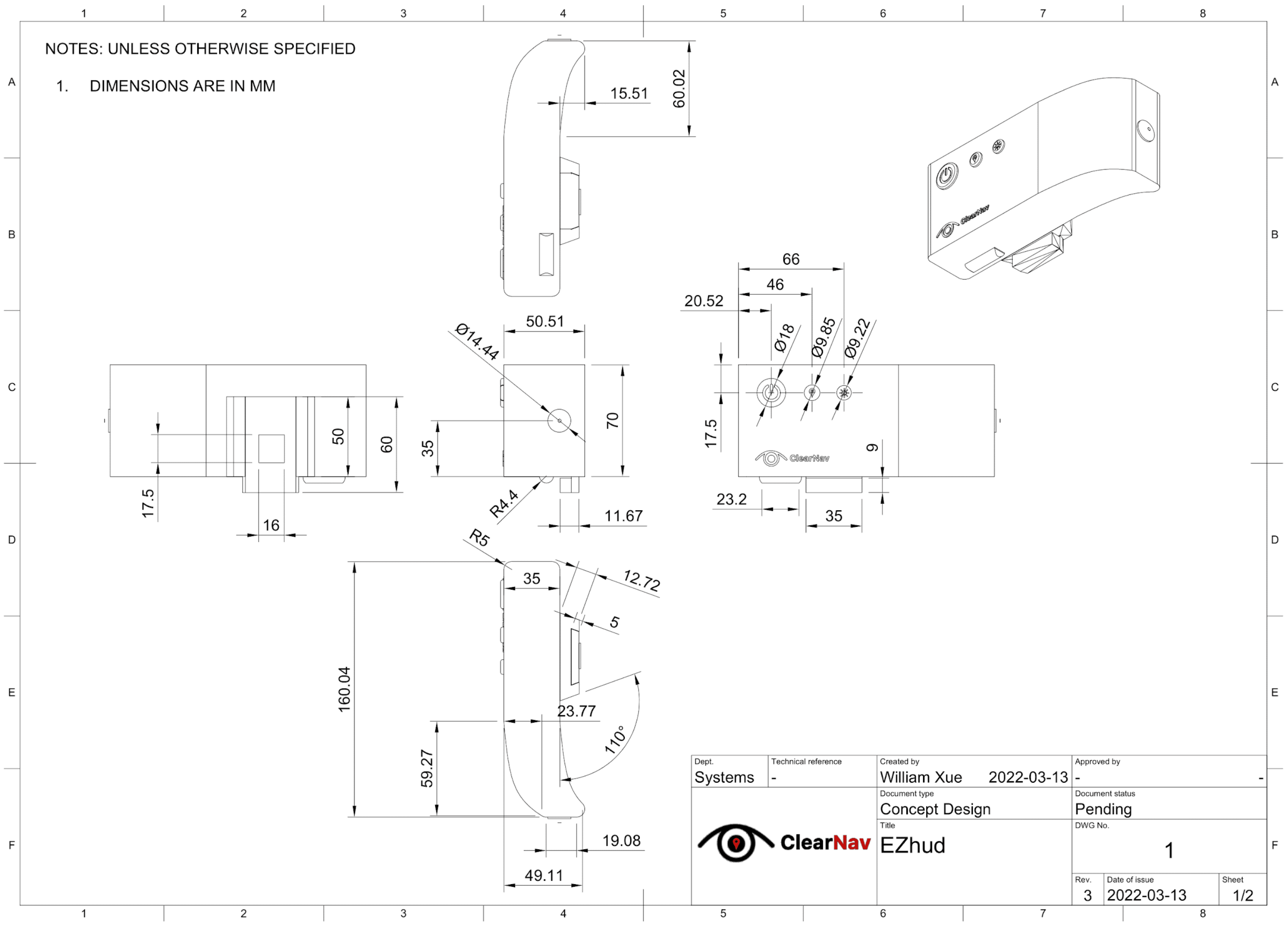


Figure 3.2.3 - EZhud Schematic

The design schematic above lists the dimensions for the main body of the EZhud device. The main considerations to take into account is the ability of the components to fit in the main body, as well as the fit on the helmet itself. Among the components, the dimensions are as follows:

Table 3.2.4: Main Component Dimensions

Component	**Dimension (L * W * H) mm**
Raspberry Pi 4B	85 * 56 * 17
Coral TPU	30 * 65 * 8
Powerboost board (Adafruit)	45 * 23 * 10
Li-Ion battery	37 * 65

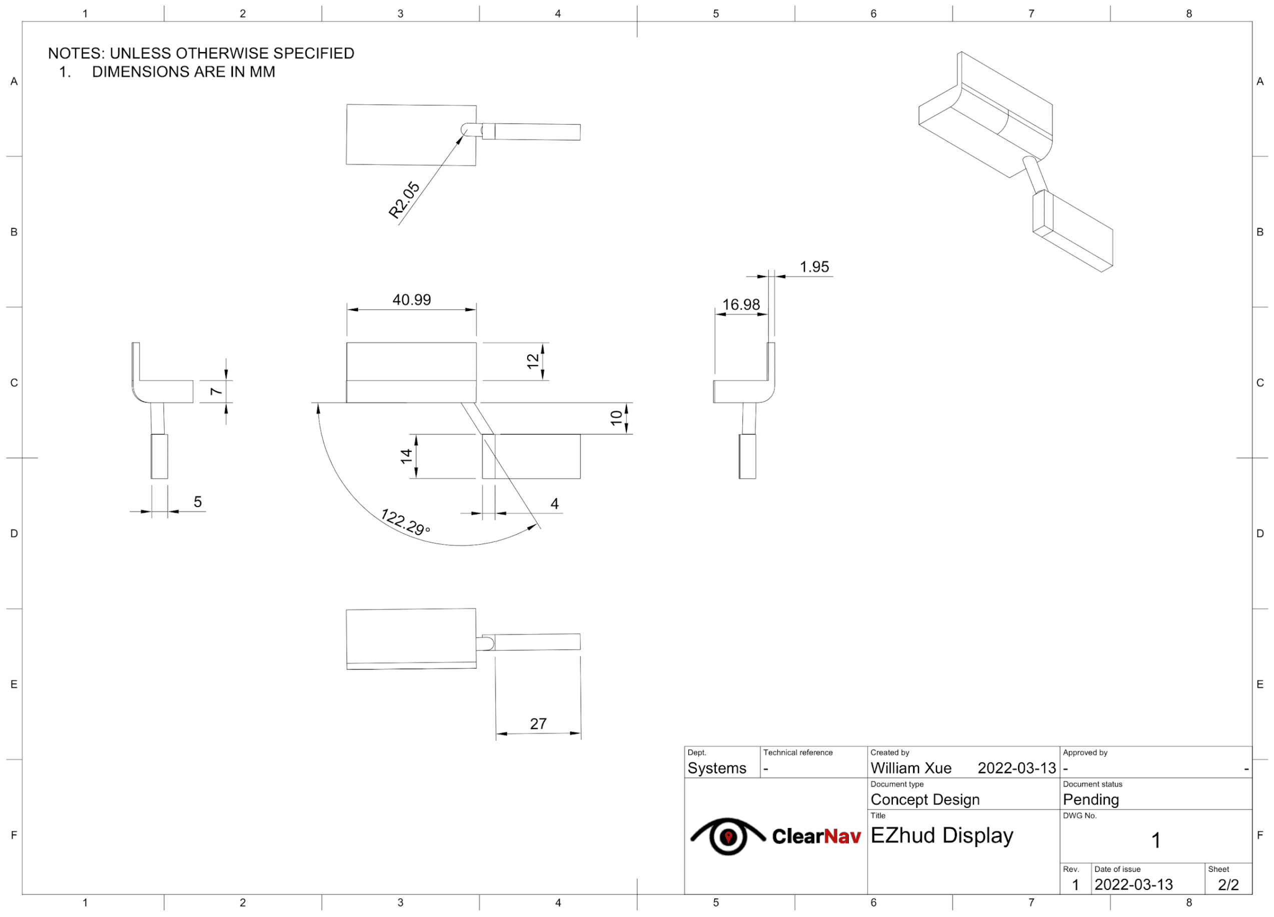


Figure 3.2.4 - EZhud Display Schematic

The dimensions of the display are also listed in Figure 3.2.4. As mentioned above, this display will sit within the helmet and will be attached to the upper visor.

Table 3.2.5: General System Design Requirements

Design ID	Design Description	Corresponding Requirements
Des3.2.8 - C	The design will have a base area no larger than 16 cm by 5 cm.	Req4.7 - C
Des3.2.9 - C	The main body will not protrude more than 5 cm from the surface of the helmet.	Req4.6 - C
Des3.2.10 - C	When attached, the main body will not freely move with respect to the helmet.	Req4.2 - C

3.3 Display Design Requirements

For our display, we require a display that is both small enough such that it can sit close to the rider's face while obstructing as minimal of their field of view as possible. Additionally, the display must be high enough resolution and full color to display clear and legible maps and navigation information. To that end, the display we plan to use will be the Sony ECX336CN OLED display. The Sony ECX336CN measures in at 0.6cm (0.23inch) diagonal, with a resolution of 640 x 400 pixels[12], covered by an optical piece to help increase legibility and clarity [13]. While the display itself requires a 51-pin ribbon connector [12], we will be pairing it with a breakout board that only requires an HDMI connection and USB power for full operation [13]. This will simplify the connection to our Raspberry Pi microcontroller.

Table 3.3.1: Display Design Requirements

Design ID	Design Description	Corresponding Requirements
Des3.3.1 - A	The device will use a Sony ECX336CN OLED display for visual navigation and maps	Req3.1.1 - A Req3.1.1a - B Req4.1b - B Req6.1.6 - B Req6.3.1 - B Req6.3.2 - B Req6.3.3 - C

3.4 Camera Design Requirements

The camera we plan to use for our heads up display is Raspberry Pi Camera Module 2 NoIR. This Camera module has an 8-megapixel sensor from Sony (IMX219) which does not have an infrared filter but instead it has a built-in infrared light which will make the picture better in dark light [14]. This camera has a small footprint with a ¼" lens able to produce a picture quality of 3280 X 2464 pixels. It is also able to record at 30 frames per second at 1080p that will suffice for speed limit sign inferencing. The Raspberry Pi NoIR camera uses the standard CSI 15 pin connector to communicate with Raspberry Pi.

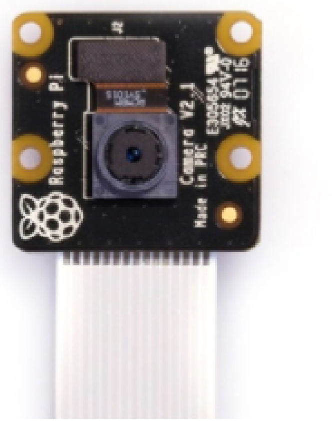


Figure 3.4.1 - Raspberry Pi Camera Module 2 NoIR [14]

Table 3.4.1: Camera Design Requirements

Design ID	Design Description	Corresponding Requirements
Des 3.4 - B	The camera will be able connect with the MCU	Req6.1.7 - B

Considering the maximum speed limit on BC highways of 120 km / h [1], and the Raspberry Pi camera's max capture rate of 30 FPS, we can calculate the frames in terms of the distance traveled.

Conversion to m/s.

$$120\ \frac{km}{hour} \times 1000\frac{meter}{km} \times \frac{1}{3600}\frac{hour}{second} = 33.33\ \frac{meter}{second}$$

Conversion to Frames Per Meter

$$30\ \frac{Frames}{second} \times \frac{1}{33.33}\frac{seconds}{meter} = 0.9\frac{Frames}{meter}$$

Following the rate of 0.9 frames / meter, and assuming each image will give a clear frame every ~1 meter, the Raspberry Pi NoIR camera meets the framerate requirements for capturing speed limits at the highest BC highway speed.

3.5 Weight Design Requirements

Table 3.5.1: Weight Design Requirements

Design ID	Design Description	Corresponding Requirements
Des 3.5 - C	EZ hud will not be bulky and will weight less than 500g	Req4.8 - C

In order for EZhud to be portable and lightweight we need to consider the weight of each component. Table 3.5.2 shows the weight of all the components. Note that the weight of the display and housing is estimated.

Table 3.5.2: Weight of all the components

Item	Component	Weight (g)
1	Raspberry Pi 4	46 [15]
2	TPU	30
3	Camera	3
4	Battery	120
5	Battery Shield Circuit	6
6	Housing	100
7	Display	40
8	Miscellaneous (wiring, button board)	10
Total Weight		**355g**

3.6 Microcontroller Design Requirements

The microcontroller unit (MCU) is a core requirement of EZhud, it allows the display, camera and all the other components to communicate with each other. It is the main processing unit of EZhud that allows navigation and speed limit capture. The design requirements that our MCU should meet are shown in the table below.

Table 3.6.1: Microcontroller Design Requirements

Design ID	Design Description	Corresponding Requirements
Des3.6.1 - A	Will support Bluetooth connectivity	Req6.1.1 - A
Des3.6.2 - A	Will support Wifi connectivity	Req6.1.2 - A
Des3.6.3 - B	Will be powerful enough to run inference and maps	Req5.1.2b-C Req6.1.4 - B
Des3.6.4 - A	Will support usb data transfer	Req6.1.5 - A
Des3.6.5 - A	Will connect to LCD/OLED display	Req6.1.6 - B
Des3.6.6 - A	Will communicate with a camera	Req6.1.7 - B

For Phase 1 of our project we aim to run a working real time navigation and have decided to work with Raspberry Pi 4B since it meets our requirements above. Raspberry Pi is an open source development platform with easy access to their updated schematics and code. In addition, they have a huge online forum to provide support and documentation throughout the development process.

Raspberry Pi 4B supports Wifi and Bluetooth connectivity. It consists of standard 40 pin GPIO header and 2 micro-HDMI ports supporting multiple options for LCD/OLED display connectivity. It includes 2 USB 3.0 ports and 2 USB 2.0 ports to support usb data transfer. This powerful MCU runs at a speed of 1.5GHz and has an option of upto 8GB of RAM [16]. The small footprint of RPi 4 makes it an ideal candidate for EZhud.

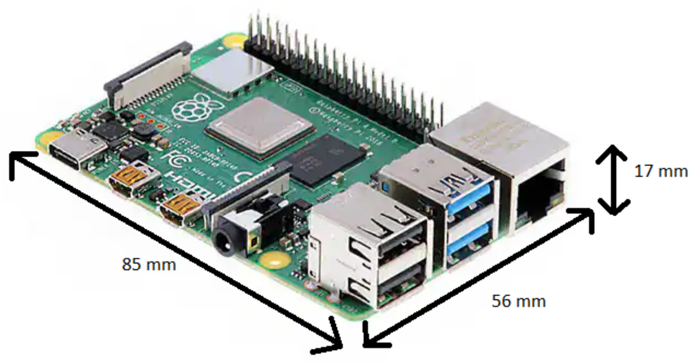


Figure 3.6.1: Dimensions of Raspberry Pi 4B [16]

RPi 4 requires 5V, 3A provided either via a USB-C connector or the 5V GPIO pin to be powered. If downstream USB peripherals consume less than 500mA then a 2.5A power supply can be used [16]. Out of 40 GPIO pins there are two 5V and 3.3V power pins and 8 ground pins. The GPIO header pinout of RPi 4 is shown in figure below.

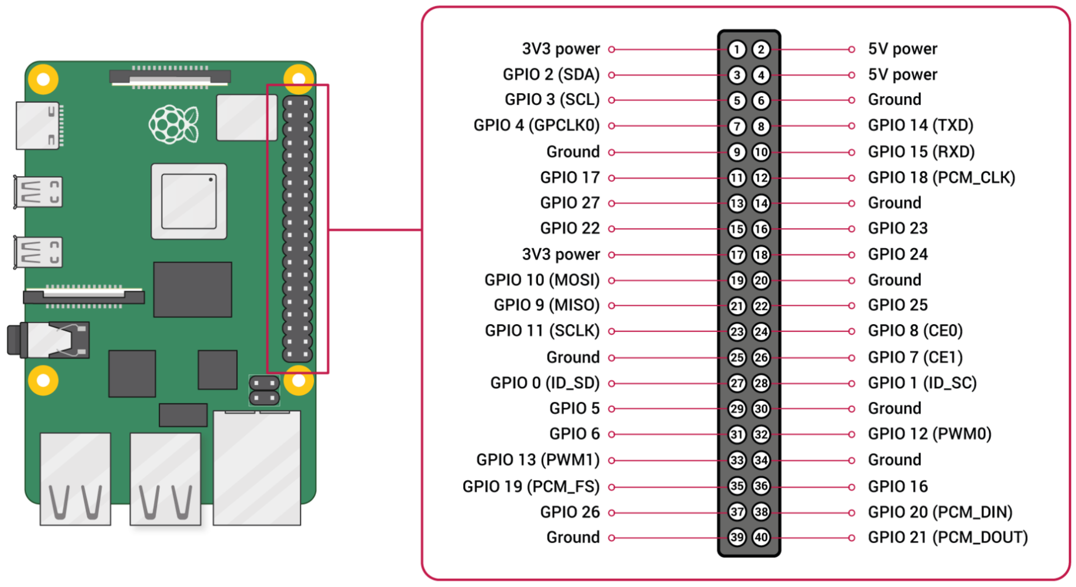


Figure 3.6.2: GPIO Connector Pinout of Raspberry Pi 4B [16]

For Phase 2 and Phase 3 of our project, we aim to achieve real time navigation along with speed limit signs inference. We would require the microcontroller to perform object detection and image processing. Based on initial testing Raspberry Pi 4 supports tensorflow, tensorflow lite and OpenCV and we were able to get both maps and inference running with 3.5-4 FPS. However, to boost our performance we plan to use the Coral USB Accelerator. Coral USB Accelerator adds an edge TPU coprocessor to our RPi to boost inferencing capabilities of our microcontroller by accelerating TensorFlow Lite models in a power efficient manner. It has a smaller footprint than a Pi 4 and can be connected to the Pi via a USB 3.0 to USB-C connector [17].

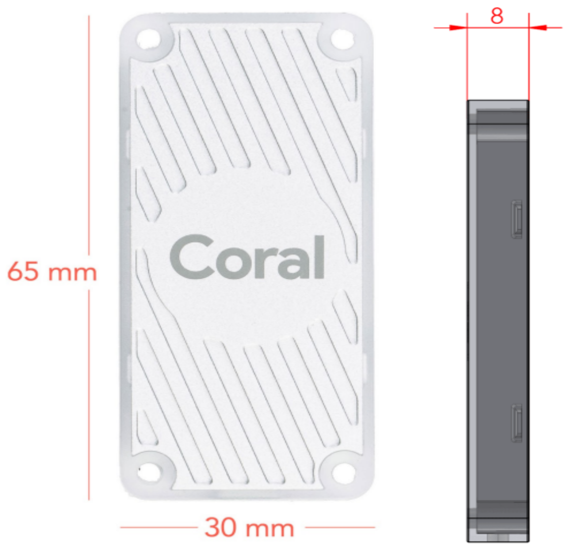


Figure 3.6.3: Coral USB Accelerator [17]

Coral TPU is powered by 5V from the USB interface and requires 500mA to operate. However, if running at maximum speed it can draw upto 900mA [17].

4 Back-End Software Design

4.1 Computer Vision Requirements

Speed limit sign recognition consists of two main stages i.e. detection and recognition of signs[18]**.** In the detection stage, the image is pre-processed and segmented according to sign properties such as color, shape, and dimension. In the recognition stage, the pre-processed and segmented sign image is classified according to its respective class which is achieved using deep learning models. For BC speed limit sign recognition in real time using OpenCV libraries and tensorflow, the model creation includes the following steps.

- Dataset creation
  - Loading the data
  - Data preprocessing
  - Data augmentation
- Training the model
  - Selection of model architecture
  - Model training via transfer learning
  - Model testing
- Testing new images on the model
  - Running model inferences on Raspberry Pi.

Considering the limiting computing power of Raspberry Pi, we prefer to use the SSD MobileNet model, designed for mobile and embedded vision applications, considering restricted resources for an on-device or embedded application. Unlike other models such as LeNet, VGGNet and AlexNet, MobileNet significantly reduces the number of parameters to 13 million parameters [19].

This results in lightweight deep neural networks. MobileNets are small, low-latency, low-power models parameterized to meet the resource constraints of a variety of use cases. We prefer to use a quantized version of this architecture because edge TPU hardware only runs quantized models [20].

The key preprocessing techniques that our model uses include shuffling, grayscaling, histogram equalization and normalization [21]. The model will shuffle the training data to increase randomness and new possibilities in the training dataset. Conversion of RGB images to grayscale improves convolution network's accuracy. Since our dataset consists of real images and many of them have low contrast, our computer vision model will use histogram equalization to enhance the images with low contrast.

For image data augmentation, the model will apply various transformations (horizontal, vertical and rotational) on existing training dataset to create new training data [22]. This will expand the training dataset, making the model robust. Furthermore, to make computation efficient, our model will normalize the images by reducing values between 0 to 1.

Since the shape of the BC speed limit sign board is rectangular with round edges, our model will use an edge detection algorithm to detect rectangles in each of the frames as they may correspond to speed limit sign boards. Also, to prevent the noise being mistakenly considered as edges, the noise must be reduced to a certain level. Thus, the images will be smoothed by applying a gaussian filter.

Table 4.1.1: Computer Vision Design Requirements

Design ID	**Description**	**Corresponding Requirements**
Des4.1.1 - B	Computer vision model will preprocess the input images via shuffling and normalization	Req5.3.1 - B
Des4.1.2 - B	Computer vision model will use vertical, horizontal and rotation augmentation for image augmentation	Req5.3.1 - B
Des4.1.3 - B	Computer vision model will use gaussian filter to minimize noise in the image	Req5.3.1 - B
Des4.1.4 - B	Computer vision model will use edge detection algorithm to detect the shape of speed limit sign board	Req5.3.1 - B
Des4.1.5 - B	Computer vision model will enhance and segment sign properties such as color, shape, and dimension	Req5.3.1 - B
Des4.1.6 - C	The design model architecture will use MobileNet SSD Deep Learning Model	Req5.3.2 - C
Des4.1.7 - C	Computer vision model will use quantized model of model architecture	Req5.3.2 - C
Des4.1.8 - C	Computer vision model will modify the speed limit sign once a new speed limit sign is recognized	Req5.3.4 - C

4.2 Navigation / Maps Design Requirements

For navigation, we will be using Google Maps over the Android Auto protocol. Being a product maintained by Google and in wide-spread deployment for the last 17 years, the performance and polish would be difficult to beat. Through Google Maps, users will be able to search for locations and start turn-by-turn navigation to destinations. Google Maps will make use of GPS to provide an accurate current location, provide timely turn-by-turn directions to the destination, update directions should a turn be missed, provide current speed and speed limit, displaying this information and updates in a neat graphical interface.

To project this information from the user's smartphone to EZhud, we will make use of the Android Auto protocol [23]. Android Auto is well supported by the Android operating system, even coming pre-installed and integrated into newer versions of the OS [24]. On the Raspberry Pi, we will make use of Crankshaft, a head unit emulator built on OpenAuto and aasdk, to provide a projection target for Android Auto on the user's smartphone [3].

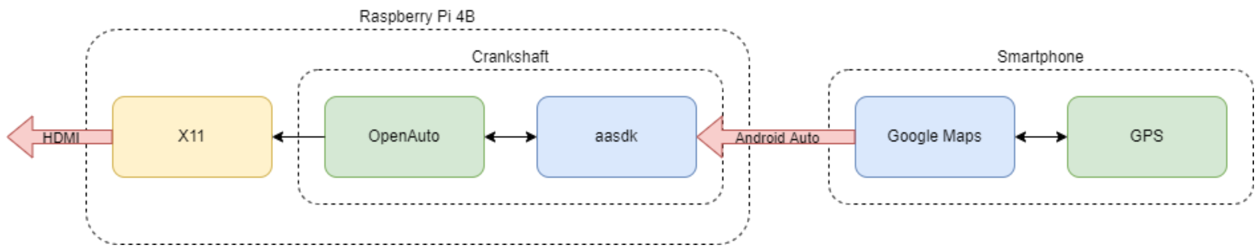


Figure 4.2.1: Overview of Navigation Data Flow

Table 4.2.1: Navigation / Maps Design Requirements

Design ID	**Description**	**Corresponding Requirement**
Des4.2.1 - A	The device will use Google Maps for maps and navigation	Req3.1.1 - A Req3.1.2 - B Req5.1.1 - A Req5.1.2 - C Req5.1.2a - B Req5.1.2b - C Req.5.1.4 - B Req.5.1.5 - C Req.5.2.1 - A Req.5.2.2 - B Req.5.2.3 - B Req.5.2.4 - C Req.5.2.5 - C Req.5.2.6 - C
Des4.2.2 - A	The device will use the Android Auto protocol to	Req3.1.7 - B Req5.1.3 - B

	stream maps and navigation information	
Des4.2.3 - A	The device will use Crankshaft to provide an Android Auto projection target for Google Maps	Req3.1.1 - A Req3.1.2 - A Req3.1.7 - B Req5.1.1 - A Req5.1.2 - C Req5.1.2a - B Req5.1.2b - C Req5.1.3 - B Req.5.2.1 - A Req.5.2.2 - B Req.5.2.3 - B Req.5.2.4 - C Req.5.2.5 - C Req.5.2.6 - C

4.3 Front-End User Application Requirements

The front-end user application for the EZhud system will take the form of a WebUI which will communicate with a web-server hosted on the device's raspberry pi. A WebUI is chosen because it allows for the most flexibility with the best accessibility. A WebUI is flexible because a cross-platform mobile app may be developed easily in the future using the React Native framework for example. A WebUI is accessible because any mobile device or desktop computer with internet access may send and receive information from the device, thus satisfying the requirement of Android compatibility.

The front-end user application will carry out three main functions

1. Let the user program navigation route into EZhud before riding
2. Let the user view and change general settings
3. Let the user view battery charge status of the EZhud device

Table 4.3.1: Front-End User Application Design Requirements

Design ID	Design Description	Corresponding Requirements
Des4.3.1 - B	The user application will be accessible from an Android mobile phone	Req5.4.1 - B
Des4.3.2 - B	The user application will display the battery status of the device	Req4.4a - C

Des4.3.3 - C	The user application will allow the user to change general settings	Req5.4.3 - C
Des4.3.4 - B	The user application will communicate with the Raspberry Pi through REST API calls over Wi-Fi	Req5.4.2 - B Req5.4.3 - C
Des4.3.5 - C	The user application will have a maximum of three screens to carry out the three main functions to minimize complexity for the user	Req5.4.4 - C

4.4 Back-End System Communication

Communication will be established between the front-end user application and the EZhud device using client-server model. The Raspberry Pi inside the EZhud device will host a Lighttpd web server, and the front-end WebUI will send and retrieve information from the device using REST API calls over Wi-Fi. A Lighttpd server is preferred over a more popular option, such as Node.js, because it is optimized for high-performance environments and prioritizes memory and CPU efficiency [25]. Furthermore, by hosting a web server on the Raspberry Pi, any device with internet access may communicate, regardless of their proximity, with the EZhud device.

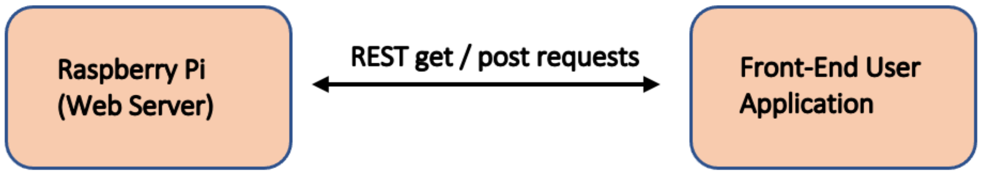


Figure 4.4.1: Back-End System Communication

Table 4.4.1: Back-End System Communication Design Requirements

Design ID	Design Description	Corresponding Requirements
Des4.4.1 - B	The device will communicate with the front-end user application using a client-server model and REST API calls.	Req5.1.1 - A Req5.4.2 - B Req5.4.3 - C
Des4.4.2 - B	A lightweight web server will be hosted on the Raspberry Pi	Req5.1.1 - A

5 Electrical Design

5.1 MCU Power consumptions calculation:

Table 5.1.1: Battery and Power Supply Design Requirements

Design ID	Design Description	Corresponding Requirements
Des 5.1.1 - A	Power supply will provide enough power for camera and GPIO current draw	Req6.2.3 - B
Des 5.1.2 - B	Battery will provide minimum 2 hours of run time	Req4.4 - B
Des 5.1.3 - A	Will include circuit protection for the product	Req6.2.4 - B

Raspberry pi 4B requires a power supply with 5V and minimum 2.5A current. Maximum current draw from one GPIO pin is 17mA and 50mA is the maximum current you can draw from all the GPIO pins [16]. The RPi camera module adds up to 200-250mA to the power requirement [26]. Raspberry pi power consumption when Idle is around 540mA (2.7 W) and it consumes a maximum of 1280mA (6.4 W) when running on full capacity (all 4 cores) [27].

Maximum current draw = 50mA (GPIO) + 250mA (camera) + 1280mA = 1580mA

Therefore, we can calculate the power consumption by the microcontroller:

P_{max} (consumed by Raspberry pi 4) = V * I = 5 * 1580 = 7900 mW or 7.9W

For this project we will be using a 3.7V 7000mAh Li-ion battery. We will be using Adafruit Powerboost 1000C which has a 90% power transfer efficiency. The power can be calculated by the equation shown below:

Power = $P_{battery}$ * hours (provided by the battery) * 0.90 = 3.7V * 7000mAh = 23.30 Wh

Thus, we can estimate the battery life when the microcontroller is at its maximum use, as shown below:

$$\text{Battery life (Minimum run time)} = \frac{Power_{battery}}{P_{max}} = \frac{23.30}{7.9} = 2.95\ hrs$$

This calculation reflects Raspberry Pi's all 4 cores are processing at 100%, on average the consumed power will be lower, giving us a better battery time. Using the formula above we can conclude that with 100% power consumption the choice of this battery can last ~ 3 hours.

5.2 Power Supply and battery

To power up the Raspberry Pi we will be using a lithium-ion (Li-ion) battery shown below in figure 5.2.1. Two 3.7V 3500mAh Li-ion batteries in parallel will be sufficient for the power requirements for Raspberry Pi. Li-ion batteries are leading the industry because of their small size, high output currents and low discharge rates. Unlike Ni-Cd or Ni-MH batteries Li-ion batteries do not have memory effect, which means that if you discharge them in partial cycles the battery has a tendency to remember a lower capacity [28].

Raspberry pi needs a steady 5V dc power supply, we will be using Adafruit Powerboost 1000C shown in figure 5.2.2. This PCB circuit performs two key roles, it has a DC-DC boost control chip (TPS61090) and a Li-ion charger chip (MCP73871T-2CCI). The power boost control chip provides a 5.2V instead of 5V output to cover up the internal resistance of the wires and other elements. It also has a low battery indicator, a red Led will light up when the output goes below 3.2V. The Li-ion charger chip will efficiently charge the two Li-ion batteries and also has two Led indicators, yellow light indicates the battery is being charged and a green light indicates when the battery is charged [29].

Figure 5.2.1: Li-ion Battery

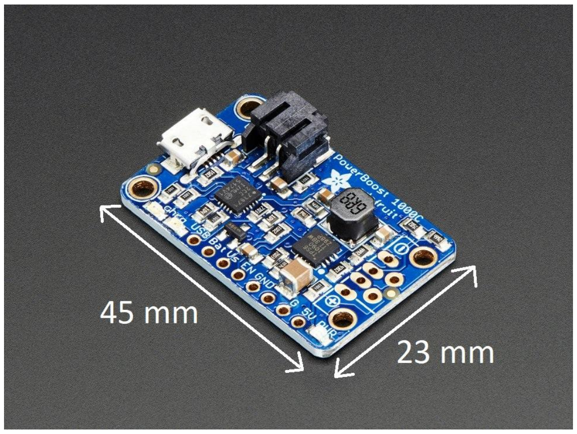


Figure 5.2.2: Powerboost 1000C [A10]

6 Conclusion

The goal of EZhud is to engineer a lightweight, reliable, and practical HUD system that can provide motorcycle riders with navigation and speed information in a safe and convenient way. The requirements document combined with this design specification ensures that this goal will be met, and the user will enjoy the product. This document focuses on the proof-of-concept and prototype phase of the product and thoroughly walks through the details regarding the design choice, design alternatives considered, and justifications for the selected design are mentioned for each subsystem and component of the product. This is a living document that serves as a reference for the design and implementation for EZhud and its subsystems.

7 References

[1]"Road and Driving Information | Transportation | Travel Resources", Travel British Columbia, 2022. [Online]. Available: https://www.travel-british-columbia.com/travel-resources/transportation/road-driving-info/. [Accessed: 14-Feb- 2022]

[2]Hg.org, 2022. [Online]. Available: https://www.hg.org/legal-articles/little-known-facts-about-motorcycle-accidents-31124. [Accessed: 14- Feb- 2022]

[3]“Crankshaft - GNU/Linux for your car with Raspberry Pi and Android Auto,” *getcrankshaft.com*. [Online]. Available: https://getcrankshaft.com/ [Accessed 12 March 2022].

[4]"WHAT ARE 3D PRINTED MATERIALS AND HOW ARE THEY USED?," [Online]. Available: https://www.makerbot.com/stories/design/3d-printing-materials/. [Accessed 13 3 2022].

[5]“P. Gharge, "Best Filament for Ender 3 (Pro/V2) - FIlament Guide," 1 1 2022. [Online]. Available: https://all3dp.com/2/ender-3-filament-guide-materials-you-can-3d-print/. [Accessed 13 3 2022].

[6]R. Rahmati, “Will Pla melt in a car? ways to avoid it,” *3dprintem*. [Online]. Available: https://www.3dprintem.com/will-pla-melt-in-a-car/. [Accessed: 13-Mar-2022].

[7]"TRACTUS3D," 2020. [Online]. Available: https://tractus3d.com/materials/tpu/. [Accessed 13 3 2022].

[8]"TRACTUS3D," 2020. [Online]. Available: https://tractus3d.com/knowledge/learn-3d-printing/filaments-for-outdoor-use/. [Accessed 13 3 2022].

[9] “Motorcycle helmet V2 3D model,” *Free3D*. [Online]. Available: https://free3d.com/3d-model/motorcycle-helmet-v2--886241.html. [Accessed: 13-Mar-2022].

[10] S. Chambers, “Your guide to 3M VHB tape,” *Strouse*. [Online]. Available: https://www.strouse.com/blog/3m-vhb-tape-guide. [Accessed: 13-Mar-2022].

[11] “VHB tapes.” [Online]. Available: https://multimedia.3m.com/mws/media/764998O/iatd-product-info.pdf. [Accessed: 14-Mar-2022].

[12] “ECX336CN 0.6 cm (Type 0.23) Active Matrix Color OLED Panel Module .” Sony Semiconductor Solutions Corporation, Atsugi, Kanagawa, Japan, 2019.

[13] Ecx336 Sony Oled 0.23inch Microdisplay With Driver Board And Freeform Optics For Ar Vr,Smart Helmet Etc. - Buy Oled 0.23,Sony Oled 0.23,Sony Oled 0.23 Ecx336 Product on Alibaba.com. [Online]. Available: https://www.alibaba.com/product-detail/ECX336-Sony-OLED-0-23inch-microdisplay_1600371987005.html. [Accessed: 13-Mar-2022].

[14]Raspberry Pi, “Buy A raspberry pi camera module 2 noir,” *Raspberry Pi*. [Online]. Available: https://www.raspberrypi.com/products/pi-noir-camera-v2/. [Accessed: 13-Mar-2022].

[15]B. Westover, “Raspberry pi 4 model B review,” *Tom's Guide*, 27-May-2021. [Online]. Available: https://www.tomsguide.com/reviews/raspberry-pi-4-model-b. [Accessed: 13-Mar-2022].

[16]“DATASHEET.” [Online]. Available: https://datasheets.raspberrypi.com/rpi4/raspberry-pi-4-datasheet.pdf. [Accessed: 14-Mar-2022].

[17]“USB accelerator datasheet,” *Coral*. [Online]. Available: https://coral.ai/docs/accelerator/datasheet/. [Accessed: 13-Mar-2022].

[18]“Traffic signs detection and recognition system using deep ...” [Online]. Available: https://arxiv.org/pdf/2003.03256. [Accessed: 14-Mar-2022].

[19] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, “MobileNets: Efficient convolutional neural networks for Mobile Vision Applications,” – *arXiv Vanity*. [Online]. Available: https://www.arxiv-vanity.com/papers/1704.04861/#:~:text=The%20MobileNet%20model%20has%20only,and%200.58%20Million%20mult%2Dadds. [Accessed: 13-Mar-2022].

[20] D. Tian, “DeepPiCar - part 6: Traffic sign and pedestrian detection and handling,” *Medium*, 21-May-2019. [Online]. Available: https://towardsdatascience.com/deeppicar-part-6-963334b2abe0. [Accessed: 13-Mar-2022].

[21] Ameen, M., 2019. *German Traffic Sign Classification Using TensorFlow*. [online] GitHub. Available at: <https://github.com/mohamedameen93/German-Traffic-Sign-Classification-Using-TensorFlow> [Accessed 14 March 2022].

[22] J. Brownlee, “How to configure image data augmentation in Keras,” *Machine Learning Mastery*, 05-Jul-2019. [Online]. Available: https://machinelearningmastery.com/how-to-configure-image-data-augmentation-when-training-deep-learning-neural-networks/. [Accessed: 13-Mar-2022].

[23]“Android Auto,” Android, 2016. [Online]. Available: https://www.android.com/auto/. [Accessed 13 March 2022]

[24]“Android auto not showing up in app drawer - Android Auto Community,” support.google.com. [Online]. Available: https://support.google.com/androidauto/thread/49061726/android-auto-not-showing-up-in-app-drawer?hl=en [Accessed 13 March 2022].

[25] "Lighttpd wiki and documentation," 19 1 2022. [Online]. Available: https://www.lighttpd.net. [Accessed 13 3 2022].

[26] “Raspberry pi documentation,” *Camera*. [Online]. Available: https://www.raspberrypi.com/documentation/accessories/camera.html. [Accessed: 13-Mar-2022].

[27] “Power consumption benchmarks | raspberry pi dramble.” [Online]. Available: https://www.pidramble.com/wiki/benchmarks/power-consumption. [Accessed: 14-Mar-2022].

[28] “Lithium-Ion Battery,” *Clean Energy Institute*, 25-Sep-2020. [Online]. Available: https://www.cei.washington.edu/education/science-of-solar/battery-technology/. [Accessed: 13-Mar-2022].

[29] A. Industries, “PowerBoost 1000 Charger - rechargeable 5V lipo USB boost @ 1A,” *adafruit industries blog RSS*. [Online]. Available: https://www.adafruit.com/product/2465. [Accessed: 13-Mar-2022].

[30] *ICBC*. [Online]. Available: https://www.icbc.com/driver-licensing/types-licences/Pages/licence-restrictions.aspx. [Accessed: 04-Mar-2022].

[31] D. A. Norman, *The Design of Everyday Things*. Cambridge, MA: The MIT Press, 2013.

[32] S. E Haggar, “Sustainable Industrial Design and Waste Management: Cradle-to-Cradle for Sustainable Development”. Amsterdam, Netherlands: Elsevier Science & Technology, 2007. [Online]. Available: https://ebookcentral-proquest-com.proxy.lib.sfu.ca/lib/sfu-ebooks/reader.action?docID=307125&ppg=37. [accessed 03-03-2022].

[33] "Cradle to Cradle," Interreg Baltic Sea Region, [Online]. Available: https://sustainabilityguide.eu/methods/cradle-to-cradle/. [accessed 03-03-2022].

[34] T. Klosowski. “What You Should Know About Right to Repair”. The New York Times. [Online]. Available: https://www.nytimes.com/wirecutter/blog/what-is-right-to-repair/. [accessed 03-03-2022].

[35] *Use of electronic devices while driving regulation*. [Online]. Available: https://www.bclaws.gov.bc.ca/civix/document/id/complete/statreg/308_2009. [Accessed: 04-Mar-2022].

[36] "Information technology — Real time locating systems — Test and evaluation of localization and tracking systems," 2016. [Online]. Available: https://www.iso.org/standard/62090.html. [Accessed 2022].

[37] "IEEE Approved Draft Standard for Wearable Consumer Electronic Devices - Overview and Architecture," 2022. [Online]. Available: https://standards.ieee.org/ieee/360/6244/. [Accessed 03-03-2022].

[38] "Software and systems engineering — Software testing — Part 11: Guidelines on the testing of AI-based systems," 2020. [Online]. Available: https://www.iso.org/standard/79016.html. [Accessed 03-03-2022].

[39] "Enclosures for Electrical Equipment, Environmental Considerations," 2012. [Online]. Available: https://www.scc.ca/en/standardsdb/standards/23524. [Accessed 03-03-2022].

[40] “ISO 31022:2020,” *ISO*, 26-May-2020. [Online]. Available: https://www.iso.org/standard/69295.html. [Accessed: 03-03-2022].

[41] “Core app quality,” *Android Developers*. [Online]. Available: https://developer.android.com/docs/quality-guidelines/core-app-quality. [Accessed: 03-03-2022].

[42] "Ergonomics of human-system interaction — Part 161: Guidance on visual user-interface elements," 2016. [Online]. Available: https://www.iso.org/standard/60476.html. [Accessed 03-03-2022].

[43] "Information technology — User interface component accessibility — Part 15: Guidance on scanning visual information for presentation as text in various modalities," 2017. [Online]. Available: https://www.iso.org/standard/71904.html. [Accessed 03-03-2022].

[44] "Information technology — User interface icons — Part 1: Introduction to and overview of icon standards," 2011. [Online]. Available: https://www.iso.org/standard/46444.html. [Accessed 03-03-2022].

[45] “Ease of operation of everyday products — Part 1: Design requirements for context of use and user characteristics,” *ISO*, 09-Dec-2020. [Online]. Available: https://www.iso.org/standard/34122.html. [Accessed: 03-03-2022].

[46] E. Forson, “Understanding SSD multibox - real-time object detection in deep learning,” Medium, 13-Mar-2022. [Online]. Available:

https://towardsdatascience.com/understanding-ssd-multibox-real-time-object-detection-in-deep-learning-495ef744fab. [Accessed: 13-Mar-2022].

[47] Prabhu, “CNN architectures - lenet, alexnet, VGG, googlenet and ResNet,” *Medium*, 15-Mar-2018. [Online]. Available: https://medium.com/@RaghavPrabhu/cnn-architectures-lenet-alexnet-vgg-googlenet-and-resnet-7c81c017b848. [Accessed: 13-Mar-2022].

[48] S. Das, “CNN architectures: Lenet, alexnet, VGG, googlenet, ResNet and more,” *Medium*, 17-Sep-2019. [Online]. Available: https://medium.com/analytics-vidhya/cnns-architectures-lenet-alexnet-vgg-googlenet-resnet-and-more-666091488df5. [Accessed: 13-Mar-2022].

[49] A. Pujara, “Image classification with MobileNet,” *Medium*, 15-Jul-2020. [Online]. Available: https://medium.com/analytics-vidhya/image-classification-with-mobilenet-cc6fbb2cd470. [Accessed: 13-Mar-2022].

[50]“OpenDash” *GitHub*, Mar. 05, 2022. https://github.com/openDsh/dash [Accessed 13 March 2022].

[51]“OpenAuto Pro,” BlueWave Studio. [Online]. Available: https://bluewavestudio.io/shop/openauto-pro-car-head-unit-solution/ [Accessed 13 March 2022].

[52]M. Szwaj, “OpenAuto,” GitHub, Mar. 13, 2022. [Online]. Available: https://github.com/f1xpl/openauto [Accessed 13 March 2022].

[53]M. Szwaj, “aasdk,” GitHub, Mar. 09, 2022. [Online]. Available: https://github.com/f1xpl/aasdk [Accessed 13 March 2022].

[54]“Crankshaft Management Tool · opencardev/crankshaft Wiki,” GitHub. [Online]. Available: https://github.com/opencardev/crankshaft/wiki/Crankshaft-Management-Tool [Accessed 13 March 2022].

[55]“Jetson Nano with Embedded vision camera - Benchmarks,” *APIs - APIs - ximea support*. [Online]. Available: https://www.ximea.com/support/projects/apis/wiki. [Accessed: 13-Mar-2022].

[56] “Nvidia Jetson Nano,” *NVIDIA*. [Online]. Available: https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/jetson-nano/product-development/. [Accessed: 13-Mar-2022].

[57] DFRobot, “Lattepanda 3 delta – a pocket-sized hackable computer for mega creativity,” *DFRobot*, 02-Nov-2021. [Online]. Available: https://www.dfrobot.com/blog-1592.html. [Accessed: 13-Mar-2022].

Appendix A: User Interface and Appearance Appendix

A1 Introduction

EZhud targets the motorcyclist and aims to provide them with an efficient solution for displaying navigation and speed limit. Unlike modern cars, motorcycles do not provide a built-in turn-by-turn navigation system. EZhud is a smart heads-up display that any user can attach to their existing helmet. After programming the route via the smartphone mobile app, the user is immediately provided with turn-by-turn navigation, current speed, and speed limit within your peripheral vision.

Engineers at Clear Nav, aim to design a product that offers a simple and efficient user interface and ensures the safety of our users. We want to make sure our device is easy to adapt and the technology is familiar to our users.

A1.1 Purpose

The purpose of this document is to illustrate the design of the EZhud user interface and provide an overview of the functionality. The document emphasizes on our design considerations based on “Seven elements of UI Interaction” from The Design of Everyday Things.

A1.2 Scope

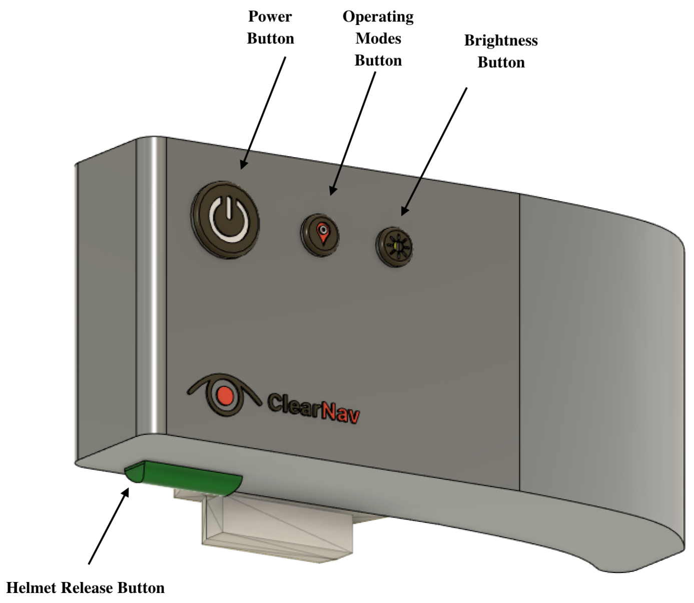
This document outlines and discusses User and Technical Analysis, Engineering Standards, Sustainability and Safety of the product. It also describes the testing procedures carried out by the engineers at ClearNav in Analytical Usability Testing and Empirical Usability Testing sections.

A2 Graphical Presentation

For the EZhud device, the user will mainly control the device through the onboard physical buttons and the included mobile application. The following sections give graphical views of both these interfaces.

A2.1 CAD Model

Shown below are the user interfaces present on the EZhud product. A 3D model is presented to explain the main buttons. As we can see in Figure A1 below, the EZhud device contains 3 buttons to interact with the microprocessor and 1 physical button to release the device from the helmet. These buttons will be further discussed in the technical analysis section of this document (Section A4).



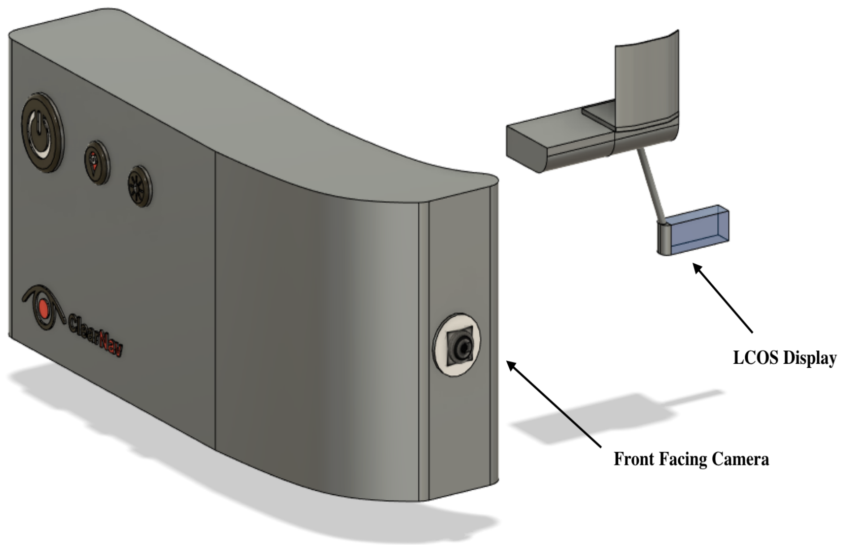


Figure A2.1: EZhud 3D Design - buttons

Figure A2.2: EZhud 3D Design - Front View

A2.2 Mobile App UI design

A graphical representation of our mobile app is shown below. The app has three pages:

1. **Main page:** allows the user to set up their route
2. **Settings page:** allows the user to change some general settings
3. **Battery percentage page:** allows the user to view the battery percentage of the device

Note: the battery percentage page is not shown in the below figures.

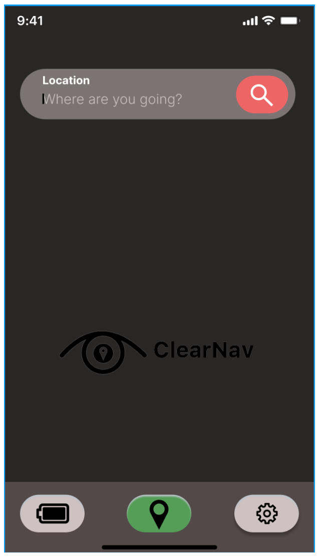


Figure A2.3: Mobile App UI - Main Page

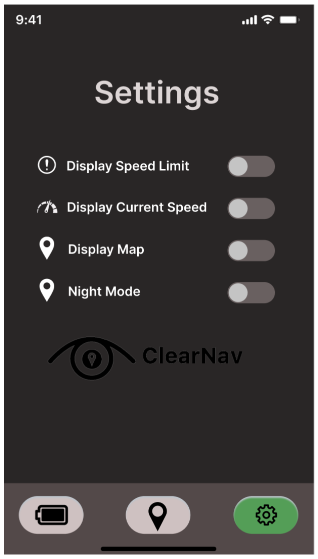


Figure A2.4: Mobile App UI - Settings Page

A3 User Analysis

EZhud is a heads up display designed for motorcycle riders in British Columbia. The rider will be able to attach our device on any full face helmet using very high bond (VHB) tape. The rider using EZhud should be able to follow turn-by-turn navigation from google maps application. New drivers that possess only a learners (L) license are prohibited from using electronic devices while riding according to BC laws which can restrict them from using EZhud [30].

The use of smartphones is very common in adults, most of our target audience will be using a smartphone. We aim to keep the setup of our device simple so that any person with a smartphone can easily connect to our device using our android application. We will be developing an application that has a simple user interface and will be used to connect to the phone. The rider can press the brightness button on the device to adjust brightness. The rider is expected to know how to set up directions in google maps and will use their knowledge to set navigation on their phone prior to boarding. After successful connection to EZhud, it will then take over the display of the maps.

A4 Technical Analysis

This section covers ClearNav's technical aspects based on seven elements of UI interaction from Don Norman's “The Design of Everyday Things”. This includes discoverability, feedback, conceptual models, affordances, signifiers, mappings, and constraints [31]

Discoverability

Discoverability in the context of interface design is a measure of ease of the user interaction when the user accesses the device or system for the first time and wants to know about essential components and setup connection for the system to work properly.

To make the user's interaction with the system effective and avoid any confusion, ClearNav will include an installation guide and usb-c cable to recharge the device. The installation guide will tell the user how to attach EZhud to the helmet and what app needs to be downloaded. The EZhud will be attached to the ride side of the helmet, in line with the right eye.The hinges in the EZhud will help the user to adjust the alignment of the display screen in accordance with his view such that it does not block the front view. The app will prompt the user to pair the device with the smartphone. Once the connection is made, the next time a user goes to the ClearNav app, the helmet will connect automatically. The app will be straightforward. Everytime, the user opens the app, navigation maps will pop up asking for the destination. At the bottom of the app there will be three icons, one for battery status, one for map navigation and one for general settings.

EZhud will have one power button which when pressed once, will turn on the device. Second button that will allow the user to toggle the modes of the display. Third button is a brightness dimmer that will allow the user to manually adjust the brightness of the head up display.

Feedback

Feedback is defined as the response from the system to notify the user about real time changes in the system. For our product, the feedback will mainly be through visualizations on the display screen as well as the app. If the initial connection is set up properly, maps will be launched. The battery indicator next to the charging port as well as the battery bar on the screen will indicate the percentage left. The battery bar will turn green when fully charged, yellow when 75% battery is remaining and red when 15% battery is remaining. Current speed and speed limit will be displayed in large font and vibrant colors. If the user is exceeding speed limit, a red outline will surround the speed limit, giving an instant visual to alert the user.

Conceptual Model

A conceptual model is an abstract representation of a system which conveys the basic functionality including the features of the system to the user [31].

EZhud's conceptual model is comprised of hardware systems and software systems. The hardware system consists of a display screen which will show real time data and maps, a camera to capture and process speed limit signs in real time using computer vision and power management system which comprise battery and a battery pack. The software system is the smartphone app which is designed in a way that will allow the user to select the real time parameters he wants to see on the display screen before starting the ride. Once the user downloads the app to pair the device with their smartphone and presses the power button to turn on the display, only then will they be able to see the real time data. This includes maps, current speed and speed limit, all of which will be displayed on their screen, showing them the significance of our hardware interface with our custom app.

Affordances

Affordances define what actions are possible by a user when he interacts with the system. To minimize the difference between user's perceived affordance and actual affordance, our app will provide strong clues in the form of icons for smooth operation of the system. Furthermore, our device includes only three buttons (power button, button to toggle screen modes and brightness dimmer) that afford specific tasks corresponding to the specific icon. Since the head up display is not designed to be directly in the rider's field of view, the user is also afforded the option to mechanically adjust the mounted head up display as per his viewing preference.

Signifiers

Signifiers refer to signals in the form of signs, symbols or any perceivable indicator that directs where the action should take place. The signifiers for EZhud are to some extent similar to those mentioned in the feedback section. Also, the three visual icons in the app (map navigation, battery life and general settings) act as signifiers, indicating the user when each of the icons needs to be selected. Besides that, when the device is turned on, it will indicate a solid blue LED. While when the device is turned off, no LED will be displayed. Pressing the second button will allow the user to toggle between two modes of display, indicated by yellow LED and third button for adjusting brightness levels.

Mappings

Mappings are defined as links between elements with certain control and their functionality that assists the user to interact with the system with ease. Our app enhances the mapping of icons to corresponding actions via spatial layout. Map icon maps to map navigation, gear icon corresponds to general settings and battery logo maps to battery status. The three buttons on EZhud i.e. power button, second button and third button map to power on/off, toggle between display modes and brightness adjustment respectively.

Constraints

Constraints are limitations that force a desired behavior by limiting a set of possible actions. EZhud uses a lockout mechanism to implement a forcing function. It will have a dedicated mechanical button that must be held down in order for the mounted head up display to be detached. This ensures safety for the user while riding. Besides this, our decision to implement projection of navigation maps, speed limit and current speed on the top right corner of the field of view, to avoid blockage of front view is a logical constraint.

A5 Sustainability / Safety

Sustainability

To meet sustainability requirements, the development and design of the product shall follow two major tenets: cradle-to-cradle and right to repair.

Cradle-to-cradle extends upon the traditional cradle-to-grave concept by viewing the product's life as a cycle [32]; once a device has reached the end of it's useful life, the materials constituting the device should be either returned to the natural environment, or reclaimed and used to create new products at the same or better quality [33]. In this way, little to no new raw resources are needed.

To abide by cradle-to-cradle framework, EZhud shall be made of either biodegradable materials (e.g. biodegradable plastics) or materials that can be easily recycled and reused without any loss in quality (e.g. metals) wherever possible.

Right to repair on the other hand focuses on extending a product's useful lifespan, thereby reducing the number of devices that need to be manufactured and then disposed of or recycled in the first place. Right to repair focuses on a few core principles: designing products to be repairable to begin with; and making parts and information to perform repairs available [34].
To abide by the right to repair principles, EZhud shall be designed to be easily repaired and having major components be modular, such that components can be replaced individually. Specific examples include favoring reusable fasteners such as screws over adhesives, having the charging port on its own daughter board instead of integrated into the main board, using standard, commonly available lithium battery cells, and having replacement available for external facing components such as the camera and display.

Safety

Safety is one of the most important considerations to take into account when designing EZhud. Many aspects of safety will be considered during the process of design, including driver distractions, exposed electrical components and material safety.

ClearNav will follow the Motor Vehicle Act and ensure that the device is securely fastened and that it will not obstruct the driver's view of the front or sides of the motorcycle [35]. The device will not interfere with the safe operation of the vehicle. Our device can also be removed when it is not in use, and the display can be turned off by the user, giving the user the choice to turn off any possible distractions.

The design of our device must also ensure that the outer shell is waterproof, and no contaminants can interfere with the regular operation of the electronic components. By protecting sensitive parts, we can assure the safe usage of EZhud on road networks. Our materials will be chosen so that they are not toxic, or corrosive during the normal use of our device.

A6 Empirical Usability Testing

Empirical usability testing will be performed by end users who have no prior knowledge or experience with EZhud or HUDs in general. Before testing the users will be provided with a EZhud unit and all accompanying documentation and user manuals. After the tests, user feedback will be collected and analyzed to verify the system is safe, reliable, and able to recover from errors in a timely manner. This information will also be used to optimize the user interface and functionality for future iterations of the device.

Table A6: Empirical Usability Tests

Category	User Activity	Questions
Out of Box Experience / Initial Setup	Attach device to existing helmet	1. Was the device easy to set up and attach to your existing helmet? 2. Was it difficult to position the HUD tape correctly? 3. Do you have confidence the device will not detach on its own?
	Pair with phone for the first time	1. Were you able to pair your phone with the device? 2. Did the system prompt you to retry pairing if the pairing failed?
Programming Route / Configuration	Program route into HUD using phone	1. Were you able to program a route using a smartphone before a ride? 2. Was it clear that the device was ready with a route?
Ride Experience	Start riding with navigation active	**1.** Is the HUD readable and non-distracting? 2. Is the placement of the HUD blocking too much of your field of view? 3. Did the speed limit update correctly and in a timely fashion? 4. Were the navigation directions clear and timely? 5. Do the navigation directions reroute when riding off route? 6. Was the device comfortable in weight, balance, and size?
Final Thoughts / Retrospective	Post Ride / Testing Interview	1. Do you feel it was beneficial to have the device? 2. Do the benefits outweigh costs? 3. Would you recommend our product to others?

A7 Analytical Usability Testing

Analytical usability testing will be performed by the engineers at ClearNav to verify the working functionality of the EZhud product. In depth testing will help to identify any inconsistencies or flaws in the UI design that would negatively affect the user's experience of the device. The following tests are designed to test the main components of EZhud’s user interface to provide a product with minimal bugs.

Physical Device Buttons

1. The buttons are easy to press while using a glove
2. The power button will turn on the device once you press and hold it for 3 seconds
3. The mode button should turn the map off and only display speed limit and current speed
4. The brightness button when pressed increases the brightness of the display

Device Camera

1. The camera should be able to detect the speed limit

Device Display

1. The display should not obstruct the rider’s field of view
2. The display should be securely attached so that it does not move around while driving

Navigation Maps Software

1. The navigation should work properly and update the route in case of a missed turn

LEDs

1. The red LED should light up when the device is put on charge
2. The blue LED should light up when the device is turned on

Helmet Attachment

1. The attachment of the device should be easy to mount and secure

Mobile Application

1. Various settings on the app reflect changes shown in the EZhud device
2. Programming the route on the app is straightforward and the route is display on the device

A8 Engineering Standards

ClearNav will adhere to the following engineering standards to ensure the project’s success from both the engineer’s point of view and the customer’s point of view.

Table A8.1: Engineering Standards Key to the Project’s Success

Standard	Description
ISO/IEC 18305:2016	Information technology — Real time locating systems — Test and evaluation of localization and tracking systems [36]
IEEE 360-2022	IEEE Standard for Wearable Consumer Electronic Devices – Overview and Architecture [37]

ISO/IEC TR 29119-11:2020	Software and systems engineering — Software testing — Part 11: Guidelines on the testing of AI-based systems [38]
CAN/CSA-C22. 2 No. 94.2-07 (R2012)	Enclosures for Electrical Equipment, Environmental Considerations [39]
ISO 31022:2020	Risk Management - Guidelines for the management of legal risk[40]

Table A8.2: Engineering Standards for EZhud User Interface

Standard	Description
Android Core App Quality	Core quality criteria and associated tests to assess the quality of an Android app [41]
ISO 9241-161:2016	Ergonomics of human-system interaction — Part 161: Guidance on visual user-interface elements [42]
ISO/IEC TS 20071-15:2017	Information technology — User interface component accessibility — Part 15: Guidance on scanning visual information for presentation as text in various modalities [43]
ISO/IEC TR 11581-1:2011	Information technology — User interface icons — Part 1: Introduction to and overview of icon standards [44]
ISO 20282-1:2006	Ease of operation of everyday products — Part 1: Design requirements for context of use and user characteristics [45]

A9 Conclusion

A well designed user interface can make or break the user experience while using a product. The ClearNav team has designed the user interfaces of EZhud with considerations from Don Norman's "The Design of Everyday Things" [31], allowing the team to reflect on key aspects such as the discoverability of components and the mappings of functionalities. The main interfaces discussed in this document include the mobile application as well as the hardware buttons in the design. By using large buttons suited for motorcycle gloves, and a well-refined application, we hope to bring to users an intuitive and easy-to-use user interface.

For the proof-of-concept and appearance prototypes, the main UI work required would be to implement buttons related to power on/off as well as implement interfaces required to program map information to the HUD device. Non-UI work would be to implement speed limit detection and turn-by-turn directions on the Raspberry Pi itself.

Appendix B: Design Alternative Appendix

This appendix outlines the alternative design options that were considered for each component of the product, and the justification for why the final design decision was made.

B1 Computer Vision Neural Network Alternatives

Table B1.1 lists the computer vision requirements. Table B1.1 and B1.2 lists the design options considered for base networks and detection networks respectively.

Table B1.1: Computer Vision Requirements

Requirement ID	Description
Req5.3.1 - B	Computer vision system shall detect speed limit sign in front of rider
Req5.3.2 - C	Computer vision model shall correctly identify speed limit sign with 80% success rate
Req5.3.3 - C	Speed limit feature shall use an algorithm to solve conflict in the case of a conflict between ride reading and dataset
Req5.3.4 - C	The HUD shall notify the user when the current speed exceeds the speed limit.

Table B1.2: Comparison of popular Detection networks [46]

Design Options	Specifications
R-CNN (Region with CNN)	● Training the data is unwieldy and too long ● Training happens in multiple phases ● Network is too slow at inference time
SSD (Single Shot Multibox Detector) **(Preferred)**	● tasks of object localization and classification are done in a single forward pass of the network ● build on the VGG-16 architecture, but discards the fully connected layers. ● Able to extract features at multiple scales ● decrease the size of the input to each layer due to additional auxiliary convolutional layers

Table B1.3: Comparison of popular Base networks [47] [48] [49]

Design Options	Specifications
LeNet-5	● This architecture is intended to classify hand-written digits. ● Expected high accuracy when dealing with speed signs, given that both hand-written digits and traffic signs are given to the computer in the form of pixel images.

	● It uses Sigmoid or Tanh nonlinearity functions. ● It consists of 60 thousand parameters.
AlexNet	● It uses ReLU activation function instead of Sigmoid or Tanh function which increase speed by 5 times with the same accuracy. ● It uses dropout technique to avoid overfitting but training time is doubled with increase in dropout ratio. ● Bigger model with 7 hidden layers, 650K units and 60M parameters.
GoogLeNet	● This architecture consists of 22 layers deep. ● It reduces the number of parameters from 60 million (AlexNet) to 4 million
VGGNet	● VGGNet consists of 16 convolutional layers ● Very appealing because of its very uniform architecture. ● VGGNet consists of 138 million parameters, which can be a bit challenging to handle.
MobileNet **(Preferred)**	● Designed for mobile and embedded vision applications, considering restricted resources for an on-device or embedded application. ● It significantly reduces the number of parameters to 13 million parameters ● This results in lightweight deep neural networks. ● MobileNets are small, low-latency, low-power models built considering computational resource constraints..

B2 Navigations / Maps Design Alternatives

Table B2.1 lists the navigations and maps requirements. Table B3.2 Lists the design options considered.

Table B2.1: Navigation and Maps Requirements

Requirement ID	Description
Req3.1.1 - A	When prompted, EZhud shall display turn-by-turn navigation instructions to the rider.
Req3.1.2 - B	When on, EZhud shall display the current speed and detect the speed limit.
Req3.1.7 - B	The device shall be android friendly.
Req5.1.1 - A	The HUD shall support programmability with map information by the user
Req5.1.2 - C	The software shall be responsive enough to display information to the user while driving.
Req5.1.2a - B	The software shall be responsive enough on city roads.

Req5.1.2b - C	The software shall be responsive enough at highway speeds.
Req5.1.3 - B	The software shall provide a way for the user to exit navigation.
Req5.1.4 - B	The software shall automatically update navigation directions in case of missed turn or user has generally gone off route.
Req5.1.5 - C	The software shall not instruct the user to make unsafe or illegal actions.
Req5.2.1 - A	The HUD shall correctly show map with current location
Req5.2.2 - B	The HUD shall correctly show point-to-point navigation directions after the user programs route
Req5.2.3 - B	The HUD navigation directions shall display the type of action (i.e., “left turn”) and distance until action (i.e., “left turn in 100m”)
Req5.2.4 - C	The HUD shall display current speed within a margin of +- 5km/h
Req5.2.5 - C	The HUD shall update navigation, current speed, and speed limit information at a minimum of 5Hz refresh-rate.
Req5.2.6 - C	The HUD navigation notification shall appear in front of the user at a safe time and distance in advance

Table B2.2: Design Options for Navigation and Maps

Design Options	Specification
Crankshaft [3]	● Available as a source code and a pre-built image ● Uses X11 for graphical elements ● Does not require desktop environment ● Exposes settings in the csmt CLI tool
OpenDash [50]	● Available as source only, must be compiled ● QT-based graphical elements ● Must be launched from a desktop environment
OpenAuto Pro [51]	● Available as a pre-built image ● Paid software ($30 USD)

All three of the above options are built upon OpenAuto, an open source Android Auto head unit emulator [52], which in turn is built on the aask, a C++ library implementing core Android Auto functionality [53]. Therefore, core Android Auto functionality is equivalent between them. Instead, differences lie in additional features and resource consumption.

Crankshaft uses significantly less CPU time due to the fact that it does not need a full desktop environment. Additionally, the software is designed to be tinkered with, exposing nearly all configurations and settings through the csmt CLI tool [54].

In contrast, OpenDash has a more pleasing UI and support for additional features such as AM/FM radio control, OBDII integration, etc., but requires nearly all the CPU time available on the Raspberry Pi 4B due to the QT-based graphical interface [50]. While the OpenDash source code is available, there is a minimal number of settings available via GUI only. OpenAuto Pro is similar to OpenDash; a more polished UI and more additional features outside of Android Auto are supported in comparison to Crankshaft.

OpenAuto pro is the most locked down option; the source code is not available, and you must pay $30 for a unique, pre-built image [51].

Since we need to distribute CPU time between computer vision and the web server on top of Android Auto, are unable to take advantage of the extra features offered by OpenDash and OpenAuto Pro, and having CLI access to system settings would greatly simplify mobile app/web server implementation, we opt to use Crankshaft.

B3 Back-end System Communication Alternatives

Table B3.1 lists the mobile application and microcontroller requirements. Table B3.2 lists the design options considered.

Table B3.1: Mobile Application and Microcontroller Requirements

Requirement ID	Description
Req5.4.1 - B	The mobile application shall be supported on android
Req5.4.2 - B	User shall be able to program navigation route into EZhud via the mobile application over bluetooth before riding
Req5.4.3 - C	User shall be able to adjust which information (navigation, current speed, speed limit) is displayed on HUD
Req5.4.4 - C	A user experienced with smartphone operation shall be able to program their route into EZhud using the mobile application in less than 15 minutes.
Req6.1.4 - B	The processor of the MCU shall be powerful enough to run image and navigation at the same time

Table B3.2: Design Options for Back-End System Communication

Design Options	Specification
Lighttpd Web Server	● Simple, open source HTTP server optimized for high performance environments ● Very small memory footprint compared to Node.js and Nginx ● Supports URL rewriting
Node.js Web Server	● Full-fledged web server ● Scales well as server load increases due to it's asynchronous nature
Nginx Web Server	● lightweight, open source HTTP server ● Good job of taking advantage of machine's multiple cores
Bluetooth	● Requires proximity considerations ● More complicated to implement

All design options fulfill the mobile application and microcontroller requirements for the EZhud product, however, Lighttpd was chosen for the web server because of it's optimization and efficiency specifically for environments requiring high performance with limited memory.

B4 Front-end User Application Alternatives

Table B4.1 lists the mobile application requirements. Table B4.2 lists the design options considered.

Table B4.1: Mobile Application Requirements

Requirement ID	Description
Req5.4.1 - B	The mobile application shall be supported on android
Req5.4.2 - B	User shall be able to program navigation route into EZhud via the mobile application over bluetooth before riding
Req5.4.3 - C	User shall be able to adjust which information (navigation, current speed, speed limit) is displayed on HUD
Req5.4.4 - C	A user experienced with smartphone operation shall be able to program their route into EZhud using the mobile application in less than 15 minutes.

Table B4.2: Design Options For Front-End User Application

Design Options	Specification
WebUI	● Accessible from any device with internet connection ● Extremely easy to setup and get running ● May not be as convenient for the user dedicated phone app
Android Phone App	● Only accessible from Android phone ● Not crossplatform

WebUI was chosen because of its simple setup and accessibility.

B5 Camera Module Alternatives

The table B5.1 shows the camera requirements and table B5.2 shows the design option that we considered.

Table B5.1: Camera Requirements

Requirement ID	Description
Req6.1.7 - B	MCU shall be able to communicate with the camera

Table B5.2: Design Options for Camera

Design Options	Specification
Raspberry Pi Camera Module 2 NoIR	● 8 Megapixel, 30 FPS at 1080p ● Resolution 3280 x 2464 ● Footprint 25 x 24 x 9 (mm) ● Used dedicated infrared light for low light
Raspberry Pi V2 Camera	● 8 Megapixel, 30 FPS at 1080p ● Resolution 3280 x 2464 ● Footprint 25 x 24 x 9 (mm) ● Uses infrared filter for low light
Raspberry Pi High Quality Camera	● 12.3 Megapixel, 30 FPS at 1080p ● Resolution 4056 x 3040 ● Footprint 38 x 38 x 18.4 (mm) ● Uses infrared filter for low light

We chose Raspberry Pi Camera Module 2 NoIR as it has a built-in infrared light which will enhance the low light vision and have better low light resolution. Raspberry Pi V2 Camera has the same footprint and resolution but it uses an infrared filter for low light, having a dedicated infrared light would be much better for our product. Raspberry Pi High Quality Camera does have a better resolution and megapixel but has a big footprint which will not be suitable for our design.

B6 Microcontroller Alternatives

Table B6.1 lists the microcontroller requirements. Table B6.2 lists the design options considered.

Table B6.1: Microcontroller Requirements

Requirement ID	Description
Req6.1.1 - A	MCU shall support Bluetooth connectivity
Req6.1.2 - A	MCU shall support Wi-Fi connectivity
Req6.1.3 - A	MCU shall work with a 5V/2.5A DC power supply
Req6.1.4 - B	The processor of the MCU shall be powerful enough to run image processing and navigation at the same time
Req6.1.5 - A	MCU shall support power input and data transfer over USB
Req6.1.6 - B	MCU shall be able to easily connect with a LCD/OLED display
Req6.1.7 - B	MCU shall be able to communicate with the camera

Table B6.2: Design Options for Microcontroller

Design Options	Specification
Raspberry Pi 4B	● Requires 5V DC power supply with 2.5-3A current ● Wifi and Bluetooth connectivity ● 40 pin GPIO header ● Quad core Cortex-A72 (ARM v8) with max speed of 1.5GHz ● Small footprint (85mm x 56mm x 17mm) ● 2 USB 2.0 and 2 USB 3.0 ports ● Support Tensorflow, Tensorflow lite and OpenCV
Coral USB Accelerator	● Works with 5V DC Power supply ● Adds an edge TPU coprocessor to any MCU to boost inferencing capabilities ● Small footprint (65mm x 30mm x 10mm)
Jetson Nano	● Requires 5V DC power supply delivering 4A for optimal performance [55] ● 40 pin GPIO header [56] ● Quad core Cortex-A57 with max speed of 1.43GHz ● Comparatively larger footprint (100mm x 80mm x 29mm)
LattePanda	● Requires 5V 2A DC power supply [57]

	• Quad Core Intel Z8350 with max speed of 1.8GHz • Large footprint (88mm x 70mm)

After carefully looking at all our MCU design options we decided to work with Raspberry Pi 4B and add Coral USB Accelerator for Phase 2 to boost performance. Although Jetson Nano has a more powerful GPU and LattePanda has better CPU, they both are more expensive ($140-$160 CAD) and have larger footprint. RPi 4 is comparatively cheaper, has a smaller footprint, and good power consumption. Moreover, Raspberry Pi has a huge online forum with lots of support available.

B7 Power Management Alternatives

Table B.7.1 lists the requirement for power management and table B.7.2 shows the design options.

Table B7.1: Power Management Requirements

Requirement ID	Description
Req4.4 - B	The HUD device shall maintain a battery charge of minimum 2 hours under normal operating conditions.
Req 6.2.3 - B	Battery shall provide sufficient power to run the camera and support maximum GPIO current draw on the microcontroller
Req 6.2.4 - B	Supply circuit needs to have circuit protection for microcontroller and other devices

Table B7.2: Design Options for Power management

Design Options	Specification
Adafruit Powerboost 1000C	• Option for power on/off button • We are choosing a 3.7V 7000mAh battery • Indicator for low battery, full battery and battery charging
PiSugar3	• Needs to turn on manually after battery drain or power loss • 5000mAh battery • No in-bulit battery indicator
MakerHawk Raspberry Pi UPS Power Supply	• Power switch to power on/off • Uses 18650 batteries, capacity can vary depending on battery being used • Programmable LED to indicate power

For our product we will go with Adafruit Powerboost 1000C. With this option we have the ability to keep the powerboost board and the battery separate which would give us the ability to optimize space in the housing. With Powerboost we also have the freedom to change our battery and would be able to use any 3.7V Li-ion battery. For PiSugar3 we have a fixed battery capacity and there is no option for powering RaspberryPi on after the battery is discharged. MakerHawk Power Supply can have variable battery capacity but the device comes as one unit which would restrict our ability to move the battery from the PCB.

B8 Display Alternatives

Table B8.1 lists display requirements. Table B8.2 lists the design options considered

Table B8.1: Display Requirements

Requirement ID	Description
Req3.1.1 - A Req3.1.1a - B	When prompted, EZhud shall display turn-by-turn navigation instructions to the rider. The display shall not obstruct the riders view.
Req4.1b - B	The display shall be mounted external to the device such that it is visible to the rider.
Req6.1.6 - B	MCU shall be able to easily connect with a LCD/OLED display
Req 6.3.1 - B	Resolution shall be sharp enough to read the map and speed limit
Req 6.3.2 - B	Size of the display shall be small enough to not obstruct the riders view
Req 6.3.3 - C	Display shall have an option to adjust the brightness

Table B8.2: Design Options for Display

Design ID	Specification
Sony ECX336CN OLED Display	● 600 x 400 @60Hz OLED display ● 0.6cm (0.23inch) diagonal ● 51-pin ribbon connector (HDMI with breakout board) ● Optical Unit ● ~$320 USD / unit from Alibaba ● Microdisplay
Vufine+ Wearable Display	● 1280x720p @60Hz LCOS display ● HDMI Compatible ● $270 CAD / unit from Amazon Prime ● Microdisplay
4” Resistive Touch IPS Display	● 800 x 480 @60Hz IPS TFT display ● 4 inches diagonal ● HDMI + micro USB power (or Raspberry Pi GPIO power)

	● ~$50 CAD / unit ● Sits very close to face, difficult to focus on

After carefully considering our options for our display, the main contenders are the Vufine+ display as well as the Sony ECX336CN display. The 4” Resistive touch display was rejected since in our testing the screen was difficult to focus on at such a close range. The Sony ECX336 display won over the Vufine+ display because of two reasons, the Vufine+ would require us to disassemble the existing housing to fit the display in the helmet whilst the Sony display would come in a configurable manner. The Vufine+ also contains an internal battery which would be independent of our main battery. Thus the Sony ECX336 was chosen because it would be easier to integrate into the helmet.

Appendix C: Test Plan Appendix

This appendix outlines the full test plan for testing the subsystems and components of the Proof-of-Concept and prototype phase of the EZhud product. The test plan will detail objectives, resources, and processes for a specific test, and is designed to verify functionality against requirements.

C1 Test coverage

The following test plan will verify requirements outlined in the design specification and requirements specification for only the Proof-of-Concept and prototype phase of the EZhud product. All subsystems will be tested, including hardware, software, and electronics. All components will be tested, including display, camera, microcontroller, battery, back-end software, and mobile application

C2 Test methods

All tests outlined below will be verified using visual inspection. Only a 5V, 3A power supply with micro-usb connection is required for test equipment to verify EZhud function.

C3 Test responsibilities

The test plan will be carried out by all engineers from the ClearNav team.

C4 Hardware Testing

Test Name: Turn on / off Display	**Subsystem/Component Tested:** Hardware button / Display
Test Description: Press and hold to turn on display. Press once to turn off the display. **Corresponding requirement:** Req4.5 - A , Req3.1.5 - C	
Expected outcome: Display LED light turns blue when display is turned on and LED is off when display is off.	
Observed outcome:	
Performed by:	

Test Name: Toggle display mode button	**Subsystem/Component Tested:** Hardware buttons / Display
Test Description: While the device is on, press the mode button **Corresponding requirement:** Req5.1.3 - B	
Expected outcome: The default display should show maps, current speed, and speed limit. Pressing the mode button once will show only the current speed and speed limit on the HUD.	
Observed outcome:	
Performed by:	

Test Name: Physical brightness dimmer	**Subsystem/Component Tested:** Hardware button / Display
Test Description: While the device is on, press the brightness button several times **Corresponding requirement:** Req 6.3.3 - C	
Expected outcome: HUD brightness increases by 20% on each press and wraps around to 0% after reaching 100% brightness	
Observed outcome:	
Performed by:	

Test Name: Power connection	**Subsystem/Component Tested:** Microcontroller
Test Description: Connect the microcontroller to the 5V DC power supply through type-C USB cable **Corresponding requirement:** Req6.1.3 - A	
Expected outcome: Microcontroller power LED indicator will turn red, indicating the microcontroller is powered on.	
Observed outcome:	
Performed by:	

C5 Software Testing

Test Name: Program route	**Subsystem/Component Tested:** Mobile application / System communication
Test Description: Program and start a route onto EZhud device using WebUI **Corresponding requirement:** Req 5.4.2 - B	
Expected outcome: EZhud device starts navigation and displays turn-by-turn directions on display	
Observed outcome:	
Performed by:	

Test Name: Speed limit sign detection	**Subsystem/Component Tested:** Camera / Computer Vision Software
Test Description: Camera detects speed limit while riding **Corresponding requirement:** Req5.3.1 - B	
Expected outcome: Camera detects speed limit in front of rider and is displayed on HUD	
Observed outcome:	
Performed by:	

Test Name: Display readability	**Subsystem/Component Tested:** Display
Test Description: Connect the phone to the device, program navigation route, and turn the display on **Corresponding requirement:** Req3.1.2 - B	
Expected outcome: While riding, navigation maps, current speed, and speed limit is readable day and night	
Observed outcome:	
Performed by:	

Test Name: Change general settings	**Subsystem/Component Tested:** Mobile application / System communication
Test Description: Change a system setting using WebUI **Corresponding requirement:** Req5.4.3 - C	
Expected outcome: Change is reflected on EZhud device when turned on	
Observed outcome:	
Performed by:	

C6 Electronics Testing

Test Name: Battery charging	**Subsystem/Component Tested:** Battery
Test Description: Connect the battery power board to 5V DC power supply **Corresponding requirement:** Req4.4a - C	
Expected outcome: Battery LED indicator turns yellow to indicate it is charging	
Observed outcome:	
Performed by:	

Test Name: Low Battery Indicator	**Subsystem/Component Tested:** Battery
Test Description: Drain battery by running the device for 2+ hours **Corresponding requirement:** Req4.4a - C	
Expected outcome: Battery LED indicator turns red when battery is low	
Observed outcome:	
Performed by:	