



School of Engineering Science • Burnaby, BC • V5A 1S6
xmli@sfu.ca

March 7, 2016

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

Re: ENSC 305W/440W Design Specification for *PortableHUD*

Dear Dr. Rawicz:

The attached document from SafeVision is the Design Specification for a Safety System for Snowboarders/Cyclists/Motorcyclists, *PortableHUD*. We are designing and implementing a Heads Up Display (HUD) which shows GPS information, speed, temperature, and time. With the *PortableHUD* mounted on helmets, users could get the required information without losing their focus on their current activities. The *PortableHUD* could also communicate via RF radio within a group.

The design specification refers to the previous document "Functional Specification for *PortableHUD*" [1]. This document aims to justify the proof of concept model for *PortableHUD* and potential design improvements. Furthermore, the test plan and use cases are included for future reference when examining the functionality of *PortableHUD*.

SafeVision consists of six hard working, detail-oriented, and motivated fourth-fifth year engineering students: Pak Lun Hoi, Xueming Monica Li, Amina Qurban, Anastasia Suprun, Yifeng Xie, and Qing Zhuang. If you have any questions or concerns about the Design specification, please, feel free to contact our CEO by phone at (778) 828-6433 or e-mail at aqurban@sfu.ca.

Sincerely,

Amina Qurban
President and CEO
SafeVision

Enclosure: *Design Specification for PortableHUD*

Design Specification for PortableHUD

Project Team:

Amina Qurban
Anastasia Suprun
Pak Lun Hoi
Qing Zhuang
Yifeng Xie
Xuemeng Monica Li

Contact Person:

Xuemeng Monica Li
xmli@sfu.ca

Submitted to:

Dr. Andrew Rawicz – ENSC 440
Steve Whitmore – ENSC 305
School of Engineering Science
Simon Fraser University

Issue date:

March 7, 2016

Revision:

1.1

Abstract

The engineers at SafeVision believe that distractions are one of the most preventable causes of accidents. With *PortableHUD* distractions are minimized while performing sports activities. The functionalities of *PortableHUD* offer time, location, speed, and RF communication. *PortableHUD* can be used as versatile mount for different types of helmets for various activities such as snowboarding, motor biking and cycling.

The design specification refers to the previous document “Functional Specification for *PortableHUD*” [1]. It outlines the details of design specifications for *PortableHUD* including all the components that are used. The goal is to give the reader a detailed look at each portion of the proof-of-concept model, from hardware and software to mechanical design, and include specifications and justifications of the design approaches.

Overall, the proof-of-concept model of *PortableHUD* includes four major design processes:

- **Hardware design** provides an overview of each component being used, their circuit configuration, and the theory of operation. Hardware design, also, provides integration of the readings from the sensors into the *PortableHUD*.
- **Software design** describes how the information received from the modules is being processed and handled.
- **Mechanical design** provides an overall physical design of *PortableHUD*.

Also, the document provides an approach for mentioned in “Functional Specification for *PortableHUD*” requirements: individual requirements for each component as well as safety, sustainability, reliability, and durability [1].

Final section of the document provides a set of preliminary test procedures to examine the functionality of proof-of-concept model. The test plan is divided into three parts: individual component, integration, and system testing.

Table of Contents

Abstract.....	ii
List of figures.....	v
List of tables.....	v
Glossary.....	vi
1. Introduction	- 1 -
1.1 Scope.....	- 1 -
1.2 Intended Audience.....	- 1 -
1.3 Project Background.....	- 1 -
1.4 Requirements Classification.....	- 2 -
2. System Specification	- 3 -
2.1 Use Case	- 3 -
2.2 Top level Design	- 4 -
2.3 Functional Justifications.....	- 4 -
2.4 Safety and Sustainability.....	- 5 -
3. Hardware Design.....	- 6 -
3.1 GPS.....	- 6 -
3.1.1 Overview	- 6 -
3.1.2 Circuit Configuration	- 6 -
3.1.3 Theory of Operation.....	- 7 -
3.2 Temperature	- 7 -
3.2.1 Overview	- 7 -
3.2.2 Theory of Operation.....	- 8 -
3.3 RF Communication.....	- 9 -
3.3.1 Overview	- 9 -
3.3.2 Circuit Configuration	- 9 -
3.3.3 Transmitter and Receiver.....	- 10 -
3.4 Real Time Clock	- 10 -
3.4.1 Overview	- 10 -
3.4.2 Circuit Configuration	- 11 -
3.4.3 Theory of operation	- 11 -
3.5 Wireless Data Transmission	- 12 -
3.5.1 Overview	- 12 -

3.5.2	Circuit Configuration	- 12 -
3.5.3	Theory of Operation.....	- 13 -
3.6	Liquid Crystal Display	- 13 -
3.6.1	Overview	- 13 -
3.6.2	Circuit Configuration	- 13 -
3.6.3	Theory of Operation.....	- 14 -
4.	Software Design	- 15 -
4.1	GPS	- 15 -
4.2	Temperature	- 15 -
4.3	RF Communication.....	- 16 -
4.3.1	Channels.....	- 16 -
4.3.2	Microphone and Speaker.....	- 17 -
4.4	Real Time Clock	- 17 -
4.5	Wireless Data Transmission	- 18 -
4.5.1	Wireless Transmitter and Receiver	- 18 -
4.6	Liquid Crystal Design	- 18 -
5.	Mechanical Design	- 19 -
5.1	Master Control Panel	- 19 -
5.2	Slave Control Panel	- 20 -
5.3	Extendable Support.....	- 20 -
5.4	Reliability and Durability.....	- 20 -
6.	Test Plan.....	- 21 -
6.1	Individual Component Testing	- 21 -
6.1.1	GPS	- 21 -
6.1.2	Temperature	- 21 -
6.1.3	RF communication	- 21 -
6.1.4	Real Time Clock	- 21 -
6.1.5	Wireless communication	- 22 -
6.1.6	LCD	- 22 -
6.2	Integration Testing.....	- 22 -
6.3	System Testing	- 22 -
7.	Conclusion.....	- 23 -
	Appendix A: Detailed Test Plans	- 26 -

List of figures

Figure 1: Front view of <i>PortableHUD</i>	1 -
Figure 2: Rear view of <i>PortableHUD</i>	2 -
Figure 3: Front view when <i>PortableHUD</i> is attached to helmet.....	2 -
Figure 4: Back view when <i>PortableHUD</i> is attached to helmet.....	2 -
Figure 5: Use case diagram	3 -
Figure 6: Block diagram of <i>PortableHUD</i>	4 -
Figure 7 GPS Module used for <i>PortableHUD</i> [5]	6 -
Figure 8: Circuit Configuration of GPS Module [6].....	6 -
Figure 9: GPS Module connected to Arduino UNO Board [7].....	7 -
Figure 10: Temperature Module used for <i>PortableHUD</i> [8].....	8 -
Figure 11: Temperature Module connected to Arduino UNO Board [9]	8 -
Figure 12: The connection between RF Components	9 -
Figure 13: Circuit Configuration for RF Transmitter [10].....	9 -
Figure 14: Circuit Configuration for RF Receiver [11].....	10 -
Figure 15: Hardware of radio components used for transmitter (left) [12] and receiver (right) [13]	10 -
Figure 16: Schematic of RCT module [15].....	11 -
Figure 17: RTC Module connection to Arduino Uno Board [14].....	11 -
Figure 18: Wireless Module used for <i>PortableHUD</i> [16].....	12 -
Figure 19: Transmitter (left) and receiver (right) pair for the wireless communication [17].....	12 -
Figure 20: 2.8" TFT LCD used for <i>PortableHUD</i> [18].....	13 -
Figure 21: Connection of LCD with Arduino Uno [19] [20].....	14 -
Figure 22: Steps to synchronize the communication and channel of the transmitter and receiver.....	16 -
Figure 23: Microphone and speaker operation	17 -
Figure 24: Steps to send and receive data packages.....	18 -
Figure 25: SolidWorks model of MCP	19 -
Figure 26: Expanded view of SCP	20 -
Figure 27: Enclosed view of SCP	20 -
Figure 28: Extendable Support	20 -

List of tables

Table 1: Pinouts of GPS module [6]	7 -
Table 2: Pinouts for LM35 Temperature [8]	8 -
Table 3: Pinouts of RTC module [14]	11 -
Table 4: Pinout of LCD.....	14 -
Table 5: Functions provided by Adafruit Ultimate GPS module [21]	15 -
Table 6: Channels and Frequency used for the radio frequency component	16 -
Table 7: Function provided by RTC_DS1307	17 -
Table 8: Function provided by 2.8" Color TFT LCD.....	19 -

Glossary

ABS: Acrylonitrile butadiene styrene

CSA: Canadian Standards Association

dB: decibel

GPS: Global Positioning System

HUD: Heads Up Display

IEC: International Experience Canada

IEEE: The Institute of Electrical and Electronics Engineers

LCD: Liquid Crystal Display

MCP: Master Control Panel

PCB: Printed Circuit Board

RF: Radio Frequency

SCP: Slave Control Panel

SQ: Square wave

TFT: Thin Film Transistor

1. Introduction

1.1 Scope

This document provides design specifications of *PortableHUD*, which includes the technical details for the proof-of-concept and prototype. The requirements are labeled accordingly throughout the document as a reference for future development and documentation.

1.2 Intended Audience

The intended audience of this document includes all the engineers involved in the development of *PortableHUD*, stakeholders, and the board of the company. The senior engineers could use this document as reference for functional clarification; whereas, stakeholders can be informed of the risks and progress through this documentation. The board of the company has provided this information to gain an overview on how *PortableHUD* is integrated.

1.3 Project Background

Some of the most preventable accidents in sports are caused by distraction caused by looking at electronic devices. In addition, helmets are commonly used in many sport activities including snowboarding, motorcycling and cycling as it provides additional protection. The goal of the SafeVision team is to produce an attachable device for helmets that could provide GPS location, speed, temperature, and RF communication. *PortableHUD* allows providing information while causing minimum distractions to the user. Our product is, also, aimed to be cheaper comparing to the competitors, i.e. BMW, Recon, and Skully.

PortableHUD does not require any modifications of the helmet nor glasses by mounting the product to the side of any helmet; thus, it makes our product unique. By using a Heads Up Display (HUD) attached to a helmet, SafeVision team is designing and implementing a device that would be portable and easily mountable to any type of helmets utilized for sports. *PortableHUD* consists of Master and Slave Control Panels (used as MCP and SCP throughout the document), and extendable mount. The following Figure illustrates the CAD model of *PortableHUD* standalone:

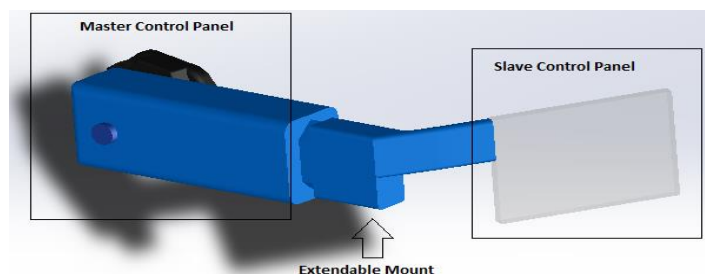


Figure 1: Front view of *PortableHUD*

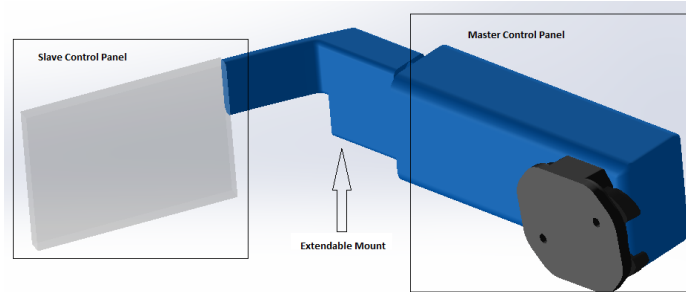


Figure 2: Rear view of *PortableHUD*

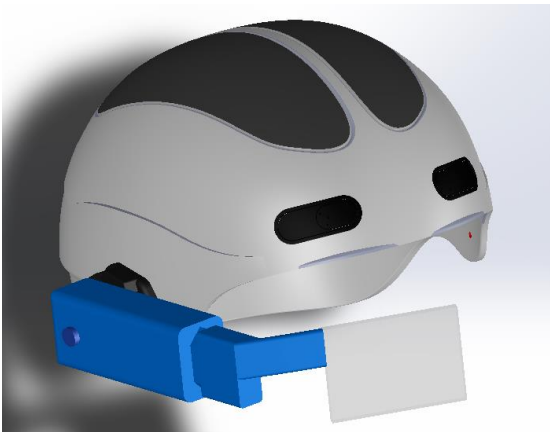


Figure 3: Front view when *PortableHUD* is attached to helmet

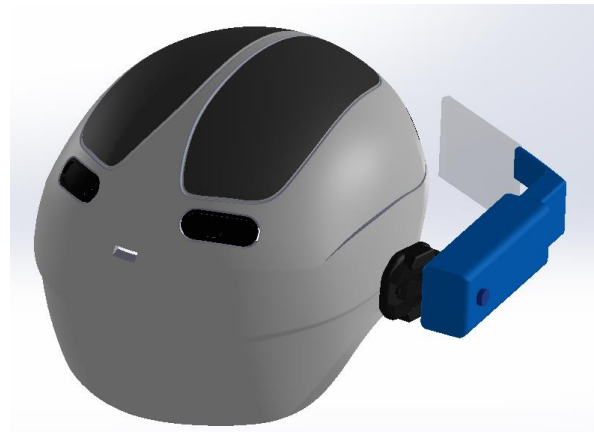


Figure 4: Back view when *PortableHUD* is attached to helmet

As we can see from Figures 1, 2, 3 and 4, the MCP and SCP are located on the side of and in front of the helmet respectively. The main circuitry is in the Master while LCD is on the Slave side.

1.4 Requirements Classification

The requirements used through this document are referenced from the functional specification [1]. Throughout the document, the following conventions are used to indicate the design requirements:

[ReqX-P]

where X is the Design requirement number and P is the priority of the design requirement denoted by one of the three values:

- I The requirement applies to proof-of-concept
- II The requirement applies to prototype revisions
- III The requirement applies to final product

2. System Specification

PortableHUD provides a way to combine some of the functionalities of the mobile device with a HUD to minimize distractions. Users would be able to access an information including time, temperature, and speed, or communicate with their team members using RF communication components without need to use cell phones. All the parts of *PortableHUD* are created to fit into a standard mount of a helmet to provide users with the flexibility of changing helmets for different types of sports. Top level design, functional justification, safety, and sustainability will be covered in the following sections.

2.1 Use Case

One of the goals of the SafeVision team is to provide a device that is easily used for variety sport activities. The device would be mounted on the helmet and adjusted for the user's best position using extendable mount of the mechanical arm. Then, by using ON/OFF button on MCP, users will be able to turn on the device and see needed information on LCD. When the device is not in use, the user can move the SCP up or to the side. A use case diagram for *PortableHUD* is shown in Figure 5:

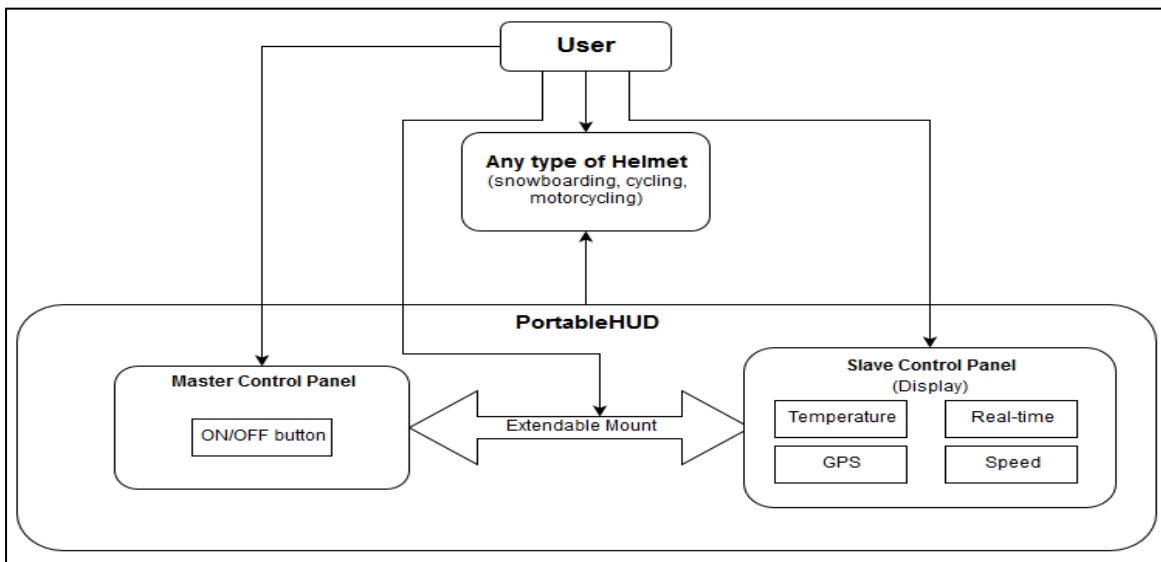


Figure 5: Use case diagram

The Figure 5 shows the interaction between the user and *PortableHUD*. Also, two or more operators of *PortableHUD* can connect to the same broadcast and they would be able to communicate using the RF component of the device for a maximum distance of 1km [Req3.3.1-I]. The detailed implementation of the requirement can be found the following sections.

Future models of *PortableHUD* would have a transparent screen with an option to display two screen configurations: first mode would display temperature, time, and speed; whereas, second mode would display location of the user using GPS.

2.2 Top level Design

PortableHUD consists of two Arduino Uno separate boards. One Arduino Uno is a part of the MCP for processing the majority of the information received from the sensors. The MCP performs the calculations from the latitude, longitude, and altitude from GPS module. The RF receiver is being handled by the MCP while the RF transmitter is being handled by the SCP. Second Arduino Uno is a part of the SCP used for displaying the information on the LCD and broadcasting the radio transmission through speakers.

The external power supply used for *PortableHUD* would be a 9 Volts rechargeable battery for the proof of concept stage. For future generations, we will be using an alternate power source: solar panels or lithium-ion battery. The block diagram of *PortableHUD* is demonstrated on Figure 6.

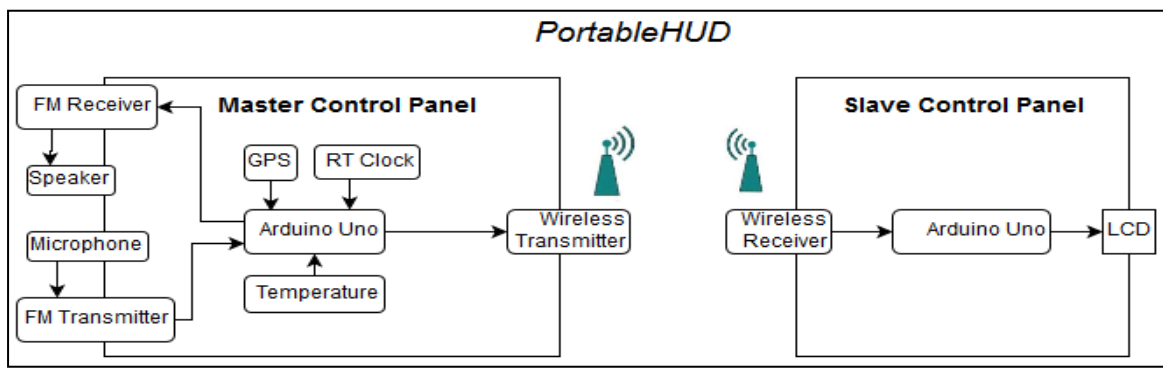


Figure 6: Block diagram of *PortableHUD*

2.3 Functional Justifications

By speaking to frequent snowboarders, cyclists, and motorcyclists, SafeVision came up with *PortableHUD* functionalities: GPS, temperature, time, speed, and group communication for team activities. To provide service for different sport users, SafeVision members decided to make their device portable so it can be easily mounted[Req3.3-II] on various types of helmets [Req3.2-I], similar to Go Pro camera mounts. The aim of *PortableHUD* is to have a lower cost [Req3.4-II] compared to the competitor's product with more sustainable and robust design. The team chose Arduino as a microcontroller due to the open source software, low power requirements, and long operation time [Req3.5-II]. To be sustainable, *PortableHUD* uses recyclable materials: ABS plastic for its enclosure, recyclable modules and microcontrollers. The extendable mount of *PortableHUD* is implemented so that the user can adjust the screen in front of their eyes for the best fit instead of having "a one size fits all". In order to eliminate the wires going through the extendable mount, wireless communication is used between two microcontrollers. In addition, by using hard impact plastic such as ABS, the interior components of *PortableHUD* are protected from the surrounding environment. Lastly, our design is implemented as simple as possible in terms of handling to allow user enjoying the functionality of *PortableHUD*.

2.4 Safety and Sustainability

The designers of SafeVision are focusing on building both a safe and sustainable product for potential customers. One of the most important safety requirements states that the integrity of the helmet shall not be compromised with the presence of the *PortableHUD* [Req5.1-I]. To ensure that, the future development of the device will include size and weight reduction to improve the safety and user experience.

The mechanical support is designed with round edges to eliminate the danger to the user [Req5.6-I] while maintaining an intuitive design. The device will be mounted on the side of the helmet to minimize the distraction caused by the information displayed on the screen [Req5.4-I]. Electrical components will be secured in the MCP and SCP to prevent movement and damage of the modules. Also, electronic components will be placed in the plastic shell to prevent direct contact burning due to heat dissipation [Req5.5-I]. All of the connectors will be designed such that all metal portions are isolated with non-conductive materials. Wireless communication is used in order to eliminate the extra wires between MCP and SCP. Finally, *PortableHUD* will be designed to keep noise level lower than 40dB to prevent negative hearing health effects [Req5.7-II].

In addition, the following safety requirements will have to be passed: all tests normally run on the helmet and safety check. For example, helmet with the device should stay on the headform for “rolloff” test when yanked aft or fore [2]. For “turn upside down and drop” [2] test, accelerometer inside the headform should register less than 300 g during the impact as it was mentioned in [Req5.3-I].

Another important aspect is the sustainability which follows the cradle-to-cradle concept. As a biomimetic approach, Cradle-to-Cradle design models human industry on nature’s nutrients circulating process [3]. SafeVision is focused on producing devices that has as little negative impact on the environment as possible. Restriction of Hazardous Substances (RoHS) compliant materials under category 4 will be selected to ensure that electrical hardware of *PortableHUD* will only contain lead free components [Req6.3-II] [4]. The electronics will be easily separable from the enclosure for the easier recycling procedures. By following the requirements mentioned above, SafeVision will keep the sustainability in mind during the design of *PortableHUD*.

Also, strategies for the electrical components redesign for the most efficient energy use, minimum size, and cost reduction are being discussed. Future development stages of SafeVision team will focus on the improvement of the safety and sustainability factors of *PortableHUD* to ensure safest device not only for the final user but also for the environment. Finally, certification of the device will be done when the final product is produced.

3. Hardware Design

The following section will discuss the hardware aspects of the design for the individual modules: GPS, temperature, RF communication, RTC, Wireless data transmission, and LCD.

3.1 GPS

GPS section will cover general overview of the module, circuit configuration, and theory of operation.

3.1.1 Overview

GPS module provides users with information including the current speed, altitude, longitude and latitude which are shown on LCD. Adafruit Ultimate GPS module can track up to 22 satellites on 66 channels with -165 dB tracking sensitivity [5]. The velocity accuracy can be 0.1 meters/s and the position accuracy can be 1.8 meters [5]. Since *PortableHUD* aims at our potential users who frequently go snowboarding, cycling or motorcycling, the tracking sensitivity and the tolerance of velocity and position are acceptable for the activities. Adafruit Ultimate GPS module has the size of 25.5mm x 35mm x 6.5mm and weighs 8.5g [5]. The module has a built-in ceramic patch antenna GTPA013 with the size of 15mm x 15mm x 4mm [5]:



Figure 7 GPS Module used for *PortableHUD* [5]

3.1.2 Circuit Configuration

Electronics circuit of the GPS module is presented on Figure 8.

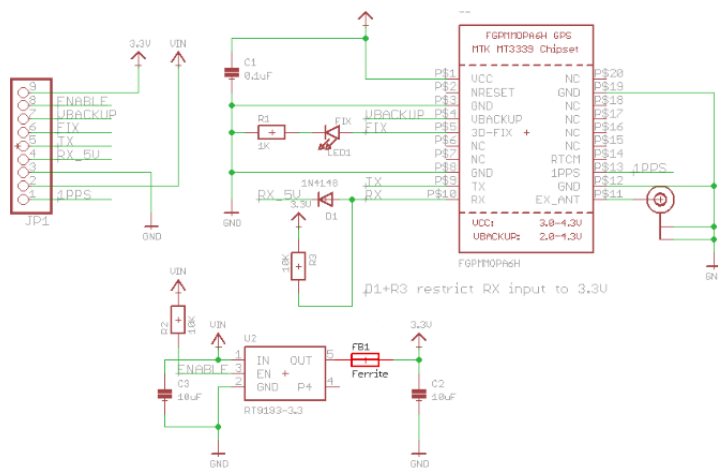


Figure 8: Circuit Configuration of GPS Module [6]

Pinouts of Adafruit Ultimate GPS module are presented in the Table 1:

Table 1: Pinouts of GPS module [6]

Pin	Description
VIN	5V supply to power the module
GND	Ground
RX	Data into GPS
TX	Data our from GPS
EN	Enable GPS by microcontroller

3.1.3 Theory of Operation

Adafruit Ultimate GPS module's connection to Arduino Uno Board is shown in Figure 9:

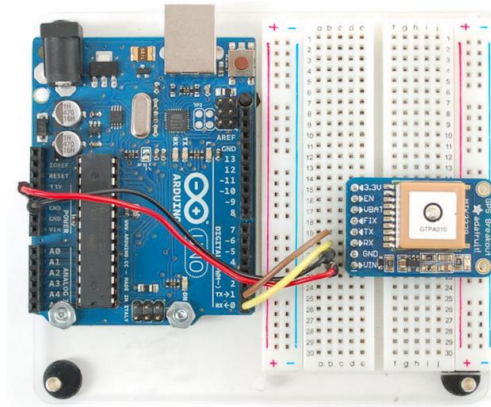


Figure 9: GPS Module connected to Arduino UNO Board [7]

Adafruit Ultimate GPS module is built around the MTK3339 chipset which can detect current speed, altitude, longitude and latitude of current location. Adafruit Ultimate GPS module is powered by pin VIN and grounded by pin GND from Arduino Uno microcontroller. A tiny red LED blinks at about 1Hz while it's searching for satellites. The module receives and transforms the data by digital pins of microcontroller. When the microcontroller sends the "Start Logging" command to the GPS module, the information of longitude, latitude, altitude and speed is logged every 15 seconds. By connecting Arduino Uno microcontroller to LCD display, the information can be read by users through the display.

3.2 Temperature

The following temperature section will cover general overview of the module and theory of operation.

3.2.1 Overview

PortableHUD provides users with the information about ambient temperature that is shown on LCD and measured by LM35 temperature sensor. LM35 is one of the cheapest temperature sensors and it is available in the market. LM35 has three pins and can be

powered by 5V from microcontroller. The comparison of the size of LM35 and Canadian penny is shown below:

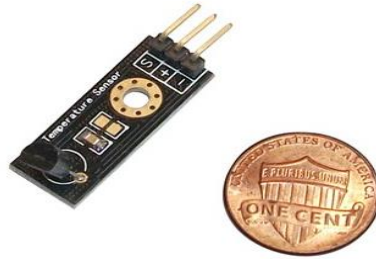


Figure 10: Temperature Module used for PortableHUD [8]

3.2.2 Theory of Operation

Pinouts of LM35 temperature sensor presented in the Table 2:

Table 2: Pinouts for LM35 Temperature [8]

Pin	Description
S	Analog Voltage Output
+	5V supply to power the module
-	Ground

LM35 temperature sensor's connection to Arduino Uno Board is shown in Figure 11:

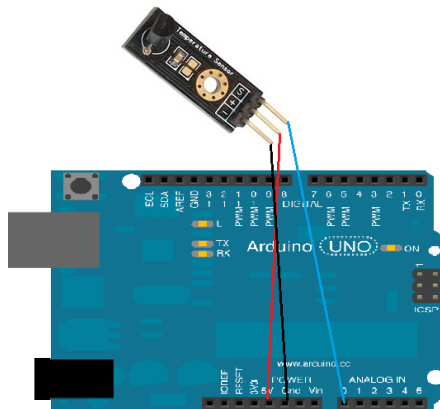


Figure 11: Temperature Module connected to Arduino UNO Board [9]

LM35 temperature sensor uses the LM35 integrated circuit which can detect current temperature. Sensor is powered by pin (+) and grounded by pin (-) from Arduino Uno microcontroller. LM35 transforms the analog voltage signal to microcontroller by pin S, the output pin, which is connected to one of analog pins of microcontroller.

3.3 RF Communication

Radio frequency communication part will cover general overview of the module, circuit configuration, and transmitter and receiver sections.

3.3.1 Overview

The radio frequency communication's primary function is to help group of users to communicate with each other when they are exercising. Users could choose one of seven radio frequencies. The RF components consist of a radio receiver, transmitter, microphone and speaker. With seven channels available, the user can ensure that he would not be interfering with other unrelated RF communication channels. To activate the microphone, the user would press the button at the side of the helmet. The speaker would be muted during the transmission to prevent noise feedback.

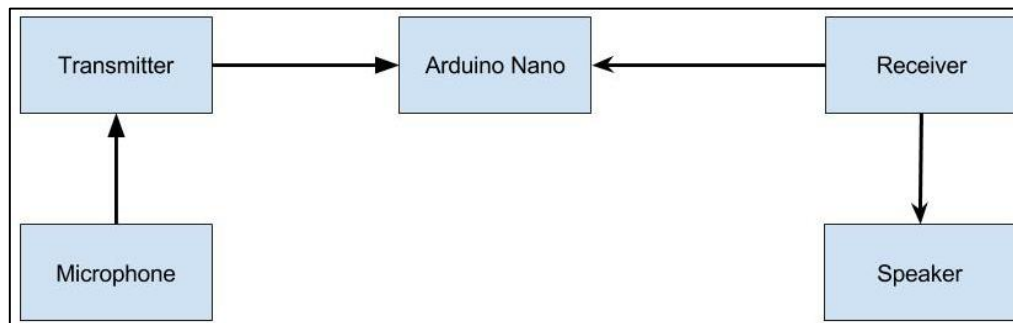


Figure 12: The connection between RF Components

3.3.2 Circuit Configuration

The following is the circuit configuration of the transmitter and receiver used for the RF components.

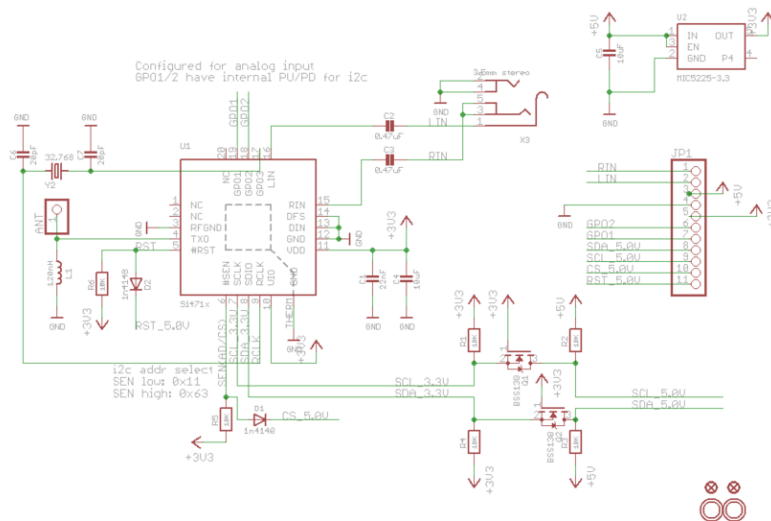


Figure 13: Circuit Configuration for RF Transmitter [10]

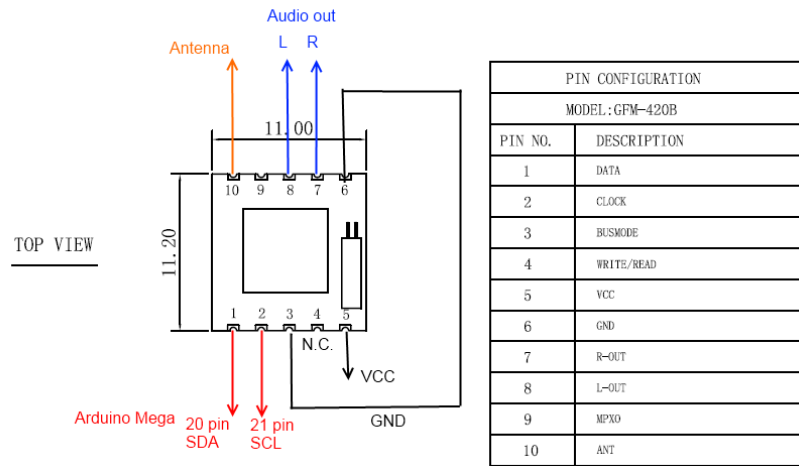


Figure 14: Circuit Configuration for RF Receiver [11]

3.3.3 Transmitter and Receiver

Based on the datasheet, the communication range of receiving the audio signal would be at least 10 meters [12]. To fulfil [Req3.3.3 I], we attached an external antenna to the radio module so that the communication range would reach at least 1km.



Figure 15: Hardware of radio components used for transmitter (left) [12] and receiver (right) [13]

3.4 Real Time Clock

Real time clock section will cover general overview of the module, circuit configuration, and theory of operation.

3.4.1 Overview

Arduino Tiny RTC I2C module provides real time information which to be shown on LCD display: current time, consisted of hour, minute and second, and date including year, month and day. The module comes in 25mm x 28mm x 8.4mm tiny packaging and is incorporated with DS1307 I2C real time clock integrated circuit and 24C32 32K I2C EEPROM storage. Module, also, consists of LIR2303 rechargeable lithium battery that can run for 1 year on a single charge. DS18B20 temperature sensor can be connected to the board; however, we are going to use LM35 temperature sensor [14].

3.4.2 Circuit Configuration

The schematic of RCT module is show on Figure 16:

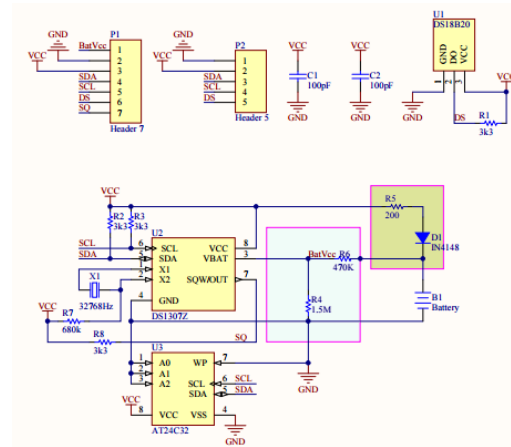


Figure 16: Schematic of RCT module [15]

Pinouts of the RTC module are presented in the Table 3:

Table 3: Pinouts of RTC module [14]

Pin	Description
GND	Ground
VCC	5V supply to power the module and charge the battery
BAT	To monitor battery voltage if desired
SDA	Data line for the RTC
SCL	Clock line for the RTC
DS	DS18B20 Temperature Sensor Output
SQ	Square wave, not used normally

3.4.3 Theory of operation

Figure 17 shows RTC module's connection to Arduino Uno Board:

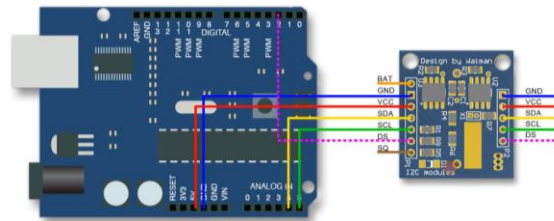


Figure 17: RTC Module connection to Arduino Uno Board [14]

Pins GND, VCC, SDA, SCL, and DS in Table 3 are duplicated from P1 to P2 in Figure 16 since they are the most useful. VCC pin should be connected to 5V to trickle charge the battery. I2C data SDA and clock SDA lines should be connected to (as an example) the analog 4 and 5 pins for Arduino Uno microcontroller. If desirable to monitor battery level, BAT can be fed to ADC pin [14].

3.5 Wireless Data Transmission

Wireless data transmission section will cover general overview of the module, circuit configuration, and theory of operation.

3.5.1 Overview

The wireless data transmission is used to establish a connection between MCP and SCP to prevent mechanical damage on wires. Since most of the computations performed and measurements taken from sensors are at the MCP, the weight and size of the SCP is reduced. The SCP consists of only an Arduino Uno and LCD receiving all of its data wirelessly from the MCP.

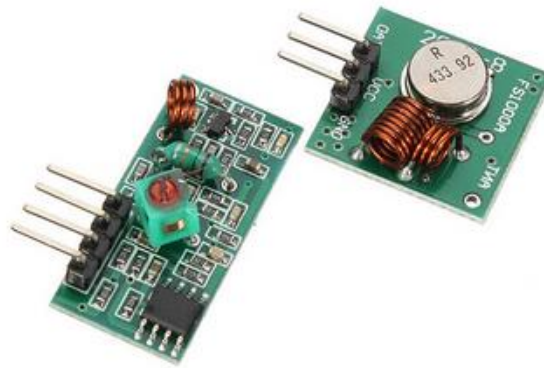


Figure 18: Wireless Module used for *PortableHUD* [16]

3.5.2 Circuit Configuration

The following is the circuit configuration of the transmitter and receiver used for the wireless components.

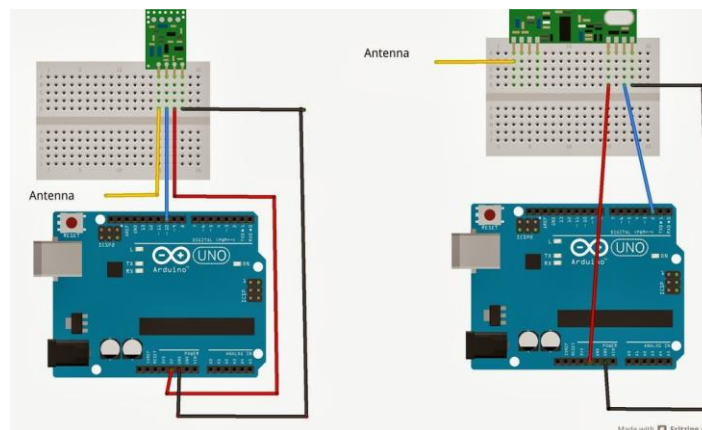


Figure 19: Transmitter (left) and receiver (right) pair for the wireless communication [17]

3.5.3 Theory of Operation

By sending the packages of data on the same radio frequency as the receiver, the SCP is able to display the information on the LCD. Since all the data from the sensors are being continuously updated, the SCP would have a way of parsing the information needed for displaying (since each LCD menu only consists of a few things to be displayed).

3.6 Liquid Crystal Display

Finally, LCD section will cover general overview of the module, circuit configuration, and theory of operation.

3.6.1 Overview

Originally, SafeVision aimed to have a transparent LCD in order not to block the users' view. However, the transparent LCDs are not on the market yet because they are just being implemented [Req3.5.1-I]. So 2.8" Color TFT LCD is used to display due to its size, color, price, and compatibility with Arduino microcontroller. In order for the screen to be less distractive, it is placed on the SCP with capability of adjusting it for the users' preference [Req3.5.4-II].

3.6.2 Circuit Configuration

2.8" Color TFT LCD has 320 x 240 resolution with 65,536 colors (16 Bit) and integrated SD card; an operating voltage is 5 Volts.

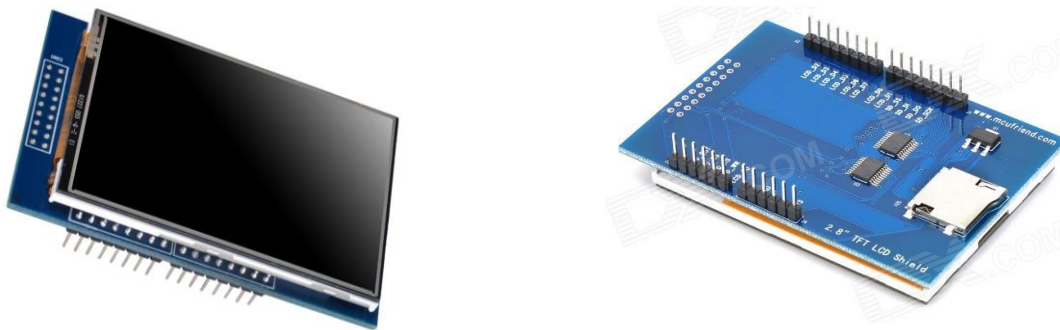


Figure 20: 2.8" TFT LCD used for PortableHUD [18]

3.6.3 Theory of Operation

2.8" TFT LCD shield can be clipped on the Arduino Uno.



Figure 21: Connection of LCD with Arduino Uno [19] [20]

Table 4 shows the connection of the pins of LCD to Arduino Uno and the reason behind it:

Table 4: Pinout of LCD

2.8" TFT screen	Arduino Uno	Description
VCC	5V	Used to power the LCD
GND	GND	ground
RSET	A4	Used to reset the LCD
CD	A3	Command/Data goes to Analog 2
CS	A2	Chip Select goes to Analog 3
WR	A1	LCD Write goes to Analog 1
RD	A0	LCD Read goes to Analog 0
TFT_CS	D5	Used for SD card chip select (CS) SPI pin to indicate the shield that the Arduino wants to send and receive data to and from the micro SD card
TFT_D/C	D6	Used for touch screen chip select (CS) SPI pin to indicate the shield that the Arduino wants to send and receive data to and from the display/screen.
SPI (MOSI)	D11	Used for TFT Data/Command control pin. The value of this pin (HIGH, or LOW) will the shield if the Arduino wants to send data or commands.
SPI (MISO)	D12	Used as data pin for SD card and screen. This pin is used by the Arduino to send data to the SD card or screen.

Since other modules are being connected on the microcontroller, SafeVision team is working on adding Arduino Nano to the Uno board for wireless data transmission, while LCD would be on Arduino Uno board.

4. Software Design

The following section will discuss the software aspects of the design for the individual modules: GPS, temperature, RF communication, RTC, Wireless data transmission, and LCD

4.1 GPS

The Adafruit Ultimate GPS module is using "Parser" library which is one of the libraries provided by Adafruit to be able to retrieve the GMT (Greenwich Mean Time), altitude, speed, longitude, and latitude; it also shows how many satellites it is currently connected to. The information provided has an error of a few meters from the current location that is due to the inaccuracy from the GPS module. For the information we need, the functions list in Table 5 of the GPS module will be called:

Table 5: Functions provided by Adafruit Ultimate GPS module [21]

Functions	Description
GPS.latitude	Provide latitude of current location
GPS.longitude	Provide longitude of current location
GPS.altitude	Provide altitude of current location
GPS.satellites	Provide number of satellites
GPS.speed	Provide current speed

4.2 Temperature

LM35 follows the scale factor of linear 10mV/°C with the accuracy of 0.5°C at +25°C. The output pin of LM35 is pin 5 which is connected to one of analog pins (temp Pin) of microcontroller. The temperature in degree Celsius can be converted from output voltage by the math formula [22]:

$$\frac{5 * analogRead(tempPin) * 100}{1024} \quad (1)$$

The "5" Volts in the Formula 1 is the power supply voltage provided for LM35 temperature sensor. The number varies on different power supplies so it would need to be adjusted according to the setup.

4.3 RF Communication

The radio frequency communication's primary function is to help group users to communicate with each other when they are exercising. By using the provided Adafruit HTU21DF test libraries to integrate the transmitter and receiver, we are able to provide a selection of 7 different FM channels for communication to be broadcasted [23]. This way, people using similar RF products are not interfered. According to [Req 3.3.5-II], the control button to set the channel would be located on the MCP of the *PortableHUD*.

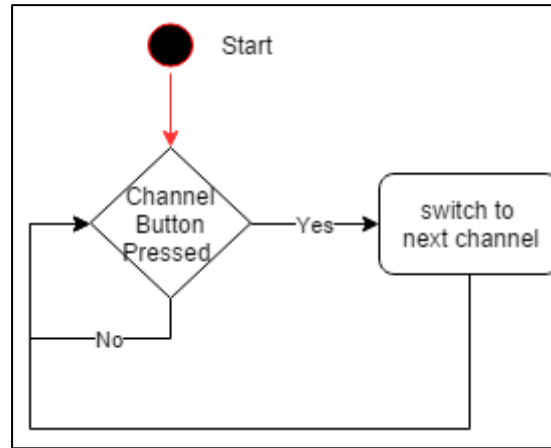


Figure 22: Steps to synchronize the communication and channel of the transmitter and receiver

4.3.1 Channels

Switching between different channels would be a fundamental function for the radio component as it allows a group to talk on the same channel. As mentioned in the requirement [Req 3.3.2-I], the system will have at least 7 different channels to choose from. In the proof-of-concept stage, the users will be able to choose between 2 channels. By pressing the button on the side of *PortableHUD*, device would be switched to the next channel. Table 6 shows the frequency SafeVision is planning to use.

Table 6: Channels and Frequency used for the radio frequency component

Channels	Frequency
1	102.30 MHz
2	102.70 MHz

4.3.2 Microphone and Speaker

When the radio transmission is in use, the speaker would be turned to avoid noise feedback. Similarly, when the speaker is turned on, the microphone would be disabled. As mentioned in requirement [Req 3.3.5-II], the control button for both the channel and microphone are located on the side of *PortableHUD*. Device will use the button to control the microphone and speaker. When the button is pressed, the microphone is enabled and the speaker is turned off. Similarly, when the button is released, the microphone is disabled and the speaker is turned on.

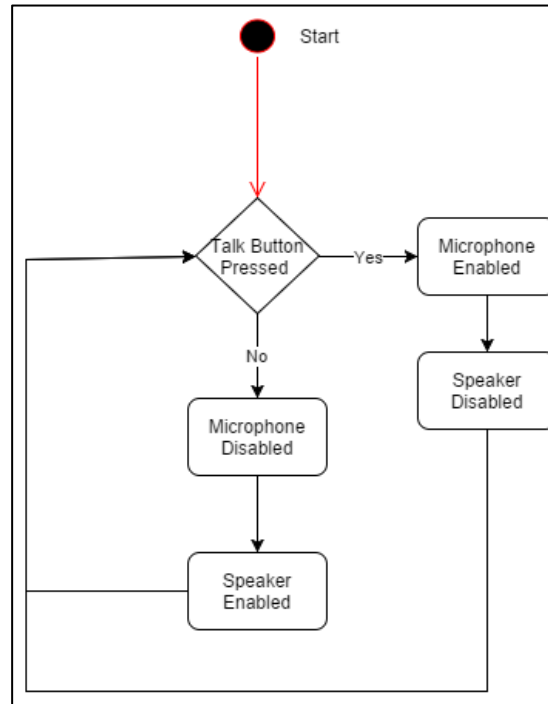


Figure 23: Microphone and speaker operation

4.4 Real Time Clock

Tiny Real Time Clock is using “Wire” and “RTClib” libraries to print out current time and date either on the LCD or console. The new time is printed with delay (1000). To display time, we need to create a new element of `RTC_DS1307` class and call the following functions:

Table 7: Function provided by `RTC_DS1307`

Functions
<code>year()</code>
<code>month()</code>
<code>day()</code>
<code>hour()</code>
<code>minute()</code>
<code>second()</code>

4.5 Wireless Data Transmission

The wireless data transmission is going to be use to integrate both the MCP and SCP by establishing a connection to sending packages of data. Using the RC Switch library provided by the Adafruit, we are able to send specific packages to the wireless receiver without disturbing nearby wireless devices by specifying the unique channel [17]. Since most of the computations are performed on the MCP, a parser would be used to send specific messages to the SCP depending on the menu selected. For example, if time and temperature were displayed on the LCD, information from GPS and current speed would not be sent to the SCP.

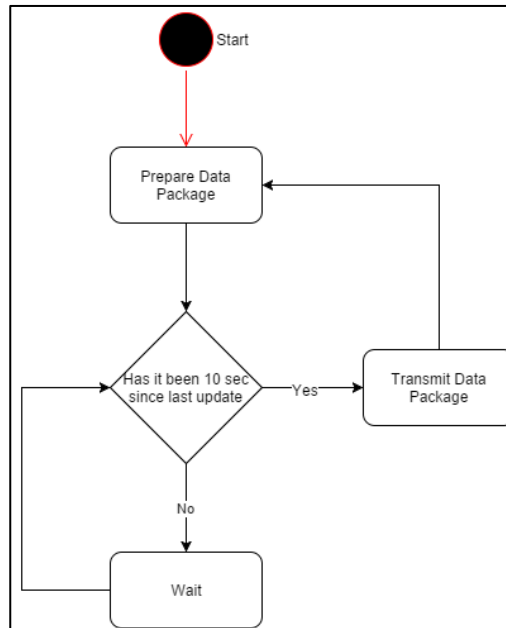


Figure 24: Steps to send and receive data packages

4.5.1 Wireless Transmitter and Receiver

The wireless transmitter would be in charge of transmitting data package including GPS data, temperature, and real time from the MCP to SCP. According to [Req3.4.2-II], the transmitter would be sending the packages to the SCP without causing any interference. To implement the functionality, SafeVision would pick a frequency that it's hardly used. The wireless receiver would be placed on the SCP to provide the LCD the information collected from the MCP according to [Req 3.4.3-II].

4.6 Liquid Crystal Design

2.8" Color TFT LCD uses "Adafruit_GFX" (Core graphics) and "Adafruit_TFTLCD" (Hardware-specific) libraries to send the information to LCD and display it according to the layout on a certain location of the display [24]. The main functions that are used are shown in Table 8.

Table 8: Function provided by 2.8" Color TFT LCD

Functions	Description
tft.setRotation();	Rotation of the object/text
tft.fillScreen();	Set a color of the display
tft.setCursor();	Navigate to the specific pixel
tft.setTextSize();	Size of the text to be displayed
tft.println();	Print text on the screen

In order to display SafeVision logo on the screen, the SD card that is attached to 2.8" Color TFT LCD shield is used along with "SPI" and "SD" libraries [24].

5. Mechanical Design

From mechanical aspect, PortableHUD consists of two parts: MCP and SCP. Most of the sensors and Uno microcontroller are located in the MCP with rigid enclosure around them; the battery is located on the back side of the MCP. SCP consists of LCD and second Uno microcontroller and can be extend from the MCP to fit user's preferences. For a proof-of-concept design, the mechanical system is large and heavy due to use of two Arduino UNOs. In the future, PortableHUD will be implemented using custom made PCB in order to reduce size, weight, and power consumption as it was discussed in "Safety and Sustainability" section.

5.1 Master Control Panel

The enclosure contains Arduino Uno microcontroller, temperature sensor, GPS module, wireless module, RF module (receiver and transmitter), and 9V battery. MCP can be mounted on the helmet using Go-pro mounting system. The size of the prototype is 100mm x 65mm x 40mm, and SafeVision team is working on reducing the size and weight of the system. Figure 25 shows the expended view of MCP which is additively manufactured from ABS plastic.

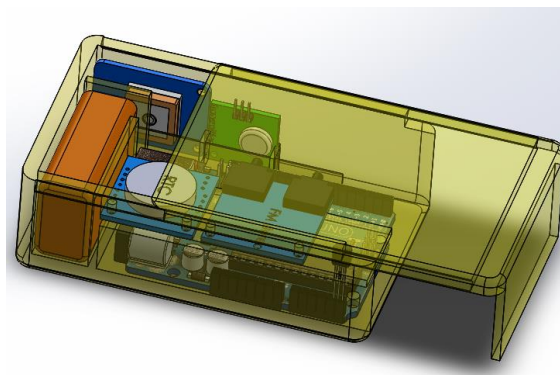


Figure 25: SolidWorks model of MCP

5.2 Slave Control Panel

SCP houses a 2.8" LCD for displaying information collected from MCP, an Arduino Uno microcontroller, and wireless data receiver. The slave control panel is using the same ABS material as the MCP. Figure 26-27 show the expanded and enclosed views of the SCP.

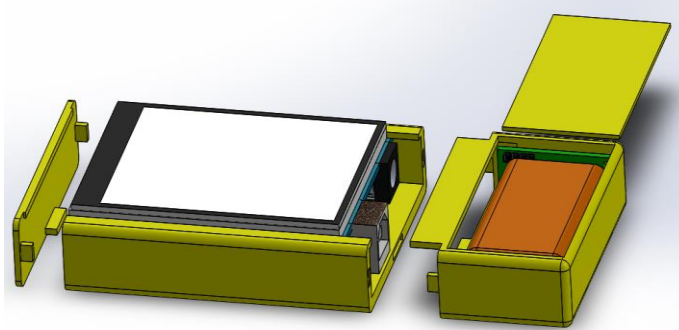


Figure 26: Expanded view of SCP

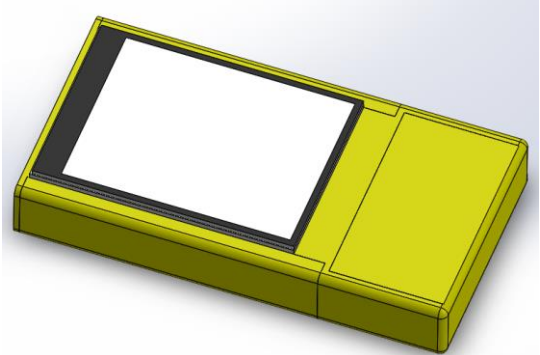


Figure 27: Enclosed view of SCP

5.3 Extendable Support

Extendable Support connects MCP and SCP to make the device more portable. The mechanism is designed in such a way that the SCP can slide on a rail to make adjustable depending on users preferences. Inside the sliding rail, there will be a “spring saw tooth” mechanism to allow the user to lock the extendable arm in place. The mechanism would unlock when the user tries to retract the extendable arm.

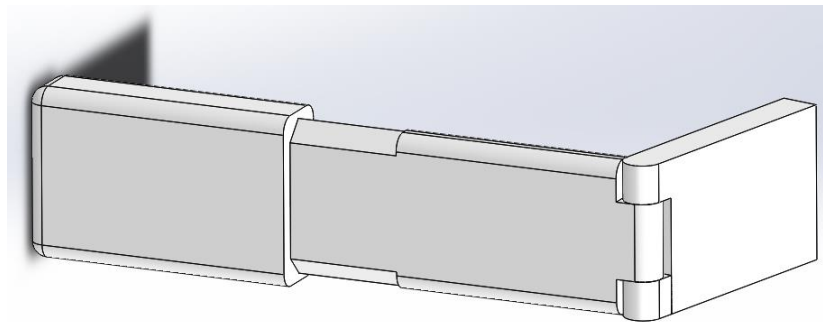


Figure 28: Extendable Support

5.4 Reliability and Durability

Reliability and durability of the device play important role in the mass production of the device. According to the requirement [Req4.3-III], testing of random samples in the independent laboratory will have to be passed. To pass the requirement, the team will need to simplify the design of the device and overdesign in order to prevent the errors caused by the electronics and the mechanical support.

Also, helmet strap will have to withstand 19.6J impact while dropping the helmet with the device on and not fail nor elongate more than 25mm [Req4.1-I]. The following requirement will be met by reducing size and weight of *PortableHUD* which can be achieved by assembling modules on the single PCB.

In order to ensure proper operation of the device, SafeVision team will have to make sure that device can properly operate at extreme temperature range [-40; 40] degrees as of [Req 3.7-II]. That can be achieved by isolating components so the ambient temperature does not affect the operation of the components.

Finally, ventilation inside the helmet should not be affected when the device is attached to the side. By ensuring that the ventilation of the helmet is not covered by *PortableHUD*, the requirement can be met.

6. Test Plan

In the following sections detailed individual component, integration, and system testing will be discussed. Individual component testing will outline main functionalities for the modules including GPS, temperature sensor, RF communication, Real Time Clock, wireless communication and LCD. In the integration testing section ordering of the modules integration and the required tests to check the correctness of the system at each stage will be discussed. Finally, system testing section will outline the tests required to pass on the system level. Detailed test plans can be found in Appendix A.

6.1 Individual Component Testing

The following tests will be required to pass before moving on to the integration of the system.

6.1.1 GPS

- Able to detect current speed of users when they are exercising
- Able to detect current location of users including altitude, latitude, longitude

6.1.2 Temperature

- Able to detect current ambient temperature

6.1.3 RF communication

- Able to receive signals within the distance of at least 10 meters
- Able to communicate on multiple channels
- Button switches between enabling the speaker and microphone

6.1.4 Real Time Clock

- Able to show current time as of HR:MIN format
- Able to show current date as of DD/MM/YYYY

6.1.5 Wireless communication

- Able to transmit wireless packages between the Arduino boards
- Able to select the frequency used to transmit the data across the board to avoid interference

6.1.6 LCD

- Able to display information that is being send
- Able to display SafeVision logo(image using SD card) on top right corner

6.2 Integration Testing

Once the individual components are implemented and tested, integration modules one-by-one followed by testing of the system can be done with the following requirements:

- Able to add temperature module to the system and display ambient temperature on LCD
- Able to add RTC module to the system and display current time and date on LCD
- Able to add GPS module to the system and display GPS coordinates on LCD
- Able to integrate RF communication and switch between two channels
- Able to transmit data wirelessly between two boards

6.3 System Testing

After the integration testing is done, the overall system should work with the following features implemented (Detailed test plans can be found in Appendix A):

- Current temperature is displayed on LCD
- Current location information is displayed on LCD
- Current time and date are displayed on LCD
- Group communication with other users who have *PortableHUDs*
- Steady placement to prevent *PortableHUD* falling from the helmet
- Extendable support stays locked while user is performing activity

7. Conclusion

The goal of SafeVision team is to build a HUD as an attachable device for various types of helmets. The document includes the design specifications, design approaches, and technical details that correspond to the functional specifications of the *PortableHUD*. It, also, presents an overview of the main functionalities and design process of *PortableHUD*. *PortableHUD* is designed based on the priorities and specific requirements from the Functional Specification document [1].

The document above describes the modules of the *PortableHUD* from the following three aspects:

- Hardware
- Software
- Mechanical

The hardware section includes the design and connections for *PortableHUD* and how each module physically integrates into the system and how the readings from the sensor are integrated into the *PortableHUD* structurally. The software section describes how the information received from the modules is processed and handled. Also, based on the inputs provided, the software describes the expected behavior of *PortableHUD*. The hardware, software, and mechanical designs mentioned above are chosen to make the product affordable, distraction minimized and compatible with helmets.

References

- [1] "PortableHUD Functional Specification," SafeVision, [Online]. Available: <http://www2.ensc.sfu.ca/~whitmore/courses/ensc305/projects/2016/1func.pdf>. [Accessed 6 March 2016].
- [2] "Helmets," Bicycle Helmet Standards, [Online]. Available: <http://www.bhsi.org/standard.htm>. [Accessed 6 March 2016].
- [3] W. McDonough and M. Braungart, *Cradle to Cradle*, North Point Press, 2002.
- [4] "RoHS Guide," Compliance, [Online]. Available: <http://www.rohsguide.com/rohs-categories.htm>. [Accessed 7 March 2016].
- [5] "GPS Module," AdaFruit, [Online]. Available: <https://learn.adafruit.com/adafruit-ultimate-gps/overview>. [Accessed 3 March 2016].
- [6] "Circuit Configuration of GPS Module," AdaFruit, [Online]. Available: <https://learn.adafruit.com/adafruit-ultimate-gps/downloads-and-resources>. [Accessed 6 March 2016].
- [7] "Direct Computer Wiring," Adafruit, [Online]. Available: <https://learn.adafruit.com/adafruit-ultimate-gps/direct-computer-wiring>. [Accessed 6 March 2016].
- [8] "Temperature Module," RP Electronics, [Online]. Available: <http://www.rpelectronics.com/temp-01.html>. [Accessed 6 March 2016].
- [9] "Interfacing PIR-Human Sensor with Arduino," Elementz Tech Blog, [Online]. Available: <https://elementztechblog.wordpress.com/2014/11/03/interfacing-pir-human-sensor-with-arduino/>. [Accessed 6 March 2016].
- [10] "Circuit Configuration RF Transmitter," Adafruit, [Online]. Available: <https://learn.adafruit.com/assets/17633>. [Accessed 6 March 2016].
- [11] "Arduino FM receiver with TEA5767," electronicsblog.net, [Online]. Available: <https://www.electronicsblog.net/arduino-fm-receiver-with-tea5767/>. [Accessed 6 March 2016].
- [12] "Si4713 FM Radio Transmitter with RDS/RDBS Support," Adafruit, [Online]. Available: <https://learn.adafruit.com/adafruit-si4713-fm-radio-transmitter-with-rds-rdbbs-support/overview>. [Accessed 6 March 2016].
- [13] "TEA5767 FM Radio Breakout Board for Arduino," Instructables, [Online]. Available: <http://www.instructables.com/id/TEA5767-FM-Radio-Breakout-Board-for-Arduino/>. [Accessed 6 March 2016].
- [14] "Real time clock module," Hobbyist, [Online]. Available:

http://www.hobbyist.co.nz/?q=real_time_clock. [Accessed 7 March 2016].

- [15] "Tiny RTC schematic," hobbyist, [Online]. Available: http://www.hobbyist.co.nz/sites/default/files/docs/RTC/Tiny_RTC_schematic.pdf. [Accessed 6 March 2016].
- [16] "RF Wireless Transmitter & Receiver Module 433Mhz for Arduino," Summerfuel Robotics, [Online]. Available: <https://sites.google.com/site/summerfuelrobots/arduino-sensor-tutorials/rf-wireless-transmitter-receiver-module-433mhz-for-arduino>. [Accessed 7 March 2016].
- [17] "Arduino 433Mhz Wireless Communication Rc Switch," Instructables, [Online]. Available: <http://www.instructables.com/id/Arduino-433Mhz-Wireless-Communication-Rc-Switch/?ALLSTEPS>. [Accessed 7 March 2016].
- [18] "Generic UNO R3 2.8" TFT Touch Screen with SD Card Socket," Xbox Antics, [Online]. Available: <http://shop.xboxantics.com/product/generic-uno-r3-2-8-tft-touch-screen-with-sd-card-socket>. [Accessed 7 March 2016].
- [19] "2.8" TFT Touch Shield V2.0," Seed Wiki, [Online]. Available: [http://www.seeedstudio.com/wiki/2.8" _TFT_Touch_Shield_V2.0](http://www.seeedstudio.com/wiki/2.8%20TFT_Touch_Shield_V2.0). [Accessed 7 March 2016].
- [20] "2.8" TFT LCD," GE Tech Wiki, [Online]. Available: http://www.geeetech.com/wiki/index.php/2.8%20TFT_LCD. [Accessed 7 March 2016].
- [21] "GPS Function," Adafruit, [Online]. Available: <https://learn.adafruit.com/adafruit-ultimate-gps/parsed-data-output>. [Accessed 6 March 2016].
- [22] "Temperature Sensor Convention," Arduino, [Online]. Available: <http://playground.arduino.cc/Main/LM35HigherResolution>. [Accessed 6 March 2016].
- [23] "Test & Usage," <https://learn.adafruit.com/adafruit-si4713-fm-radio-transmitter-with-rds-rdbs-support/test-and-usage>, [Online]. Available: <https://learn.adafruit.com/adafruit-si4713-fm-radio-transmitter-with-rds-rdbs-support/test-and-usage>. [Accessed 7 March 2016].
- [24] "adafruit/Adafruit-GFX-Library," GitHub, [Online]. Available: <https://github.com/adafruit/Adafruit-GFX-Library>. [Accessed 6 March 2016].

Appendix A: Detailed Test Plans

SafeVision Test Case Sheet 1			
Test Case Number:1	Test Case: Temperature Measurement	Test Date:	Reviewed By:
Scenario:	User would like to know the current temperature of the environment. Temperature should be between -40 °C and 40 °C		
Steps:	1) User would select the appropriate menu to display temperature 2) User would simply look at the LCD Display for the temperature		
Input:	N/A		
Expected Output:	The temperature displayed should match the temperature of the surrounding environment		
Actual Output:			
Comments:			
Results:	Approved <input type="checkbox"/>	Conditionally Approved <input type="checkbox"/>	Not Approved <input type="checkbox"/>

Test Case Number:2	Test Case: Current Speed	Test Date:	Reviewed By:
Scenario:	User would be moving at a specific speed User would like to know whether they are over speeding		
Steps:	1) User would select the appropriate menu to display speed 2) User would simply look at the LCD Display for the speed		
Input:	N/A		
Expected Output:	The speed showed in the LCD should match the speed shown by the speedometer		
Actual Output:			
Comments:			
Results:	Approved <input type="checkbox"/>	Conditionally Approved <input type="checkbox"/>	Not Approved <input type="checkbox"/>

SafeVision Test Case Sheet 2

Test Case Number: 3	Test Case: Location	Test Date:	Reviewed By:
Scenario:	User would be standing in a location that GPS module could detect satellites (e.g. Outdoor soccer field)		
Steps:	1) User would select the appropriate menu to display location 2) User would simply look at the LCD Display for the longitude and latitude		
Input:	N/A		
Expected Output:	GPS coordinates should match with the current location of user (with an error of a few meters)		
Actual Output:			
Comments:			
Results:	Approved <input type="checkbox"/>	Conditionally Approved <input type="checkbox"/>	Not Approved <input type="checkbox"/>

Test Case Number:4	Test Case: Real Time	Test Date:	Reviewed By:
Scenario:	User would be needing to look at the time and date		
Steps:	1) User would select the appropriate menu to display date and time 2) User would simply look at the LCD Display for the time and date		
Input:	N/A		
Expected Output:	The time and date shown on the LCD should match the current time and date		
Actual Output:			
Comments:			
Results:	Approved <input type="checkbox"/>	Conditionally Approved <input type="checkbox"/>	Not Approved <input type="checkbox"/>

SafeVision Test Case Sheet 3

Test Case Number:5	Test Case: RF Communication	Test Date:	Reviewed By:
Scenario:	User would like to communicate with his/her friend at a distance using the RF Communication device. The channel used to communicate must be the same		
Steps:	1) User would press the button located at the side of the helmet to talk		
Input:	N/A		
Expected Output:	The voice of the user should be broadcasted to the another <i>PortableHUD</i> device		
Actual Output:			
Comments:			
Results:	Approved <input type="checkbox"/>	Conditionally Approved <input type="checkbox"/>	Not Approved <input type="checkbox"/>

Test Case Number:6	Test Case: Mechanical Design	Test Date:	Reviewed By:
Scenario:	User would like to mount the device on the helmet and position it to his/hers preference.		
Steps:	1) User mounts the device on a helmet 2) User positions screen for his/hers preference		
Input:	N/A		
Expected Output:	The device holds its extendable position that the user set		
Actual Output:			
Comments:			
Results:	Approved <input type="checkbox"/>	Conditionally Approved <input type="checkbox"/>	Not Approved <input type="checkbox"/>