

Survival Wristband

ECE 445 Design Document

Derek Niess, John Quinn and Bartu Alp

Group 47

TA: Madison Hedlund

2/27/20

1. Introduction

1.1 Problem and Solution Overview

Avalanches, mining collapses and landslides are regular occurrences that often result in loss of life for many of the individuals that are unfortunate enough to become caught up within these natural disasters. One of the major causes of death in these cases is suffocation as the people who get caught in these disasters are usually severely disoriented and do not know which direction to attempt to dig. Time is extremely crucial once buried underneath snow or other debris and any attempt to dig in the incorrect direction can waste this valuable time and cause suffocation. In order to maximize a person's chance of survival in the event that they become buried, they need to quickly and efficiently dig in the correct direction in order to avoid suffocation. Our solution is a wristband device that displays the approximate direction of shortest distance to the surface so that users will be able to know the direction that they need to dig in order to maximize their chances of survival. The survival wristband will be battery powered and correctly display the direction of the shortest path to the surface to some degree of accuracy at all times. The survival wristband will help to maximize people's chance of survival in the event that they are buried under snow or other debris and lose their bearings.

Avalanches are a terrifying natural phenomena that generally causes more than 150 deaths each year [1]. People that are unlucky enough to be buried underneath the snow are taught to thrust their hand or another part of their body above the snow surface as they are beginning to be buried [2]. The issue is often times people are so severely disoriented and see endless white that they are told to guess where the snow surface is and try to get a body part to break the surface in an attempt to be rescued. This is similarly the case with mining collapses and landslides as people that are buried are blocked from the light and almost all of the time have no sense of direction. The survival wristband would be extremely useful in any of these situations as it would display the correct direction of the path of shortest distance to the surface so that the user is able to stick a body part in the correct direction and have a better chance of being rescued.

In the event that an individual is buried too deeply to be able to reach the surface with a body part, the survival wristband can still significantly increase the chances of survival. Time is extremely essential to avoid suffocation and death. If a person buried in an avalanche can be rescued within the first 18 minutes, the chance of survival is 91% and, in the range of 19 minutes to 35 minutes, the chance of survival drops significantly at about 34% [1]. Suffocation and hypothermia are two of the leading causes of death from avalanches and both

causes have the potential of being avoided if time is conserved. The survival wristband can increase the chances of survival for victims buried fairly deeply by pointing them in the correct direction to dig and conserving time. Similarly, suffocation is a cause of death in mining collapses and landslides and the survival wristband can also be used to determine the direction of digging by shortest path to the surface and thus saving crucial time that is likely critical to the victims survival.

1.2 Visual Aid

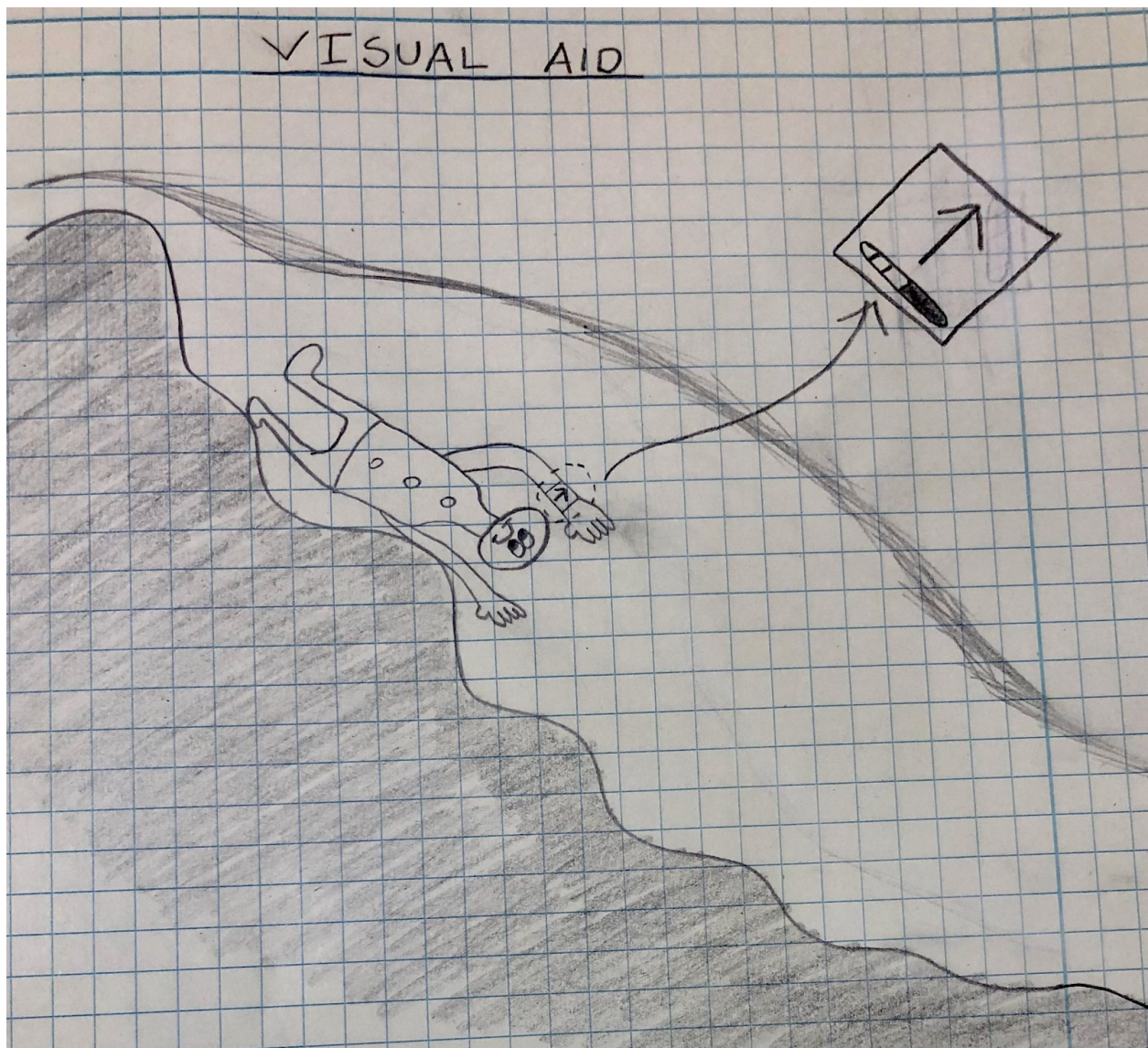


Figure 1. Visual representation of the Survival Wristband

1.3 High-level Requirements List

- The refresh rate of the displayed arrow is such that it gets recalculated at least once every 2 seconds.
- Defining 0 degrees as the top orientation perpendicular with the surface the user is standing on, when the device is tilted up to ± 90 degrees in the XY XZ and YZ plane, the displayed arrow gets updated after the next refresh cycle to point in the direction of shortest path to the surface where the calculation behind this update has a margin of error within ± 22.5 degrees in each of the three listed planes.
- The wristband must ideally be under 1 pound of total weight.

2. Design

2.1 Block Diagram

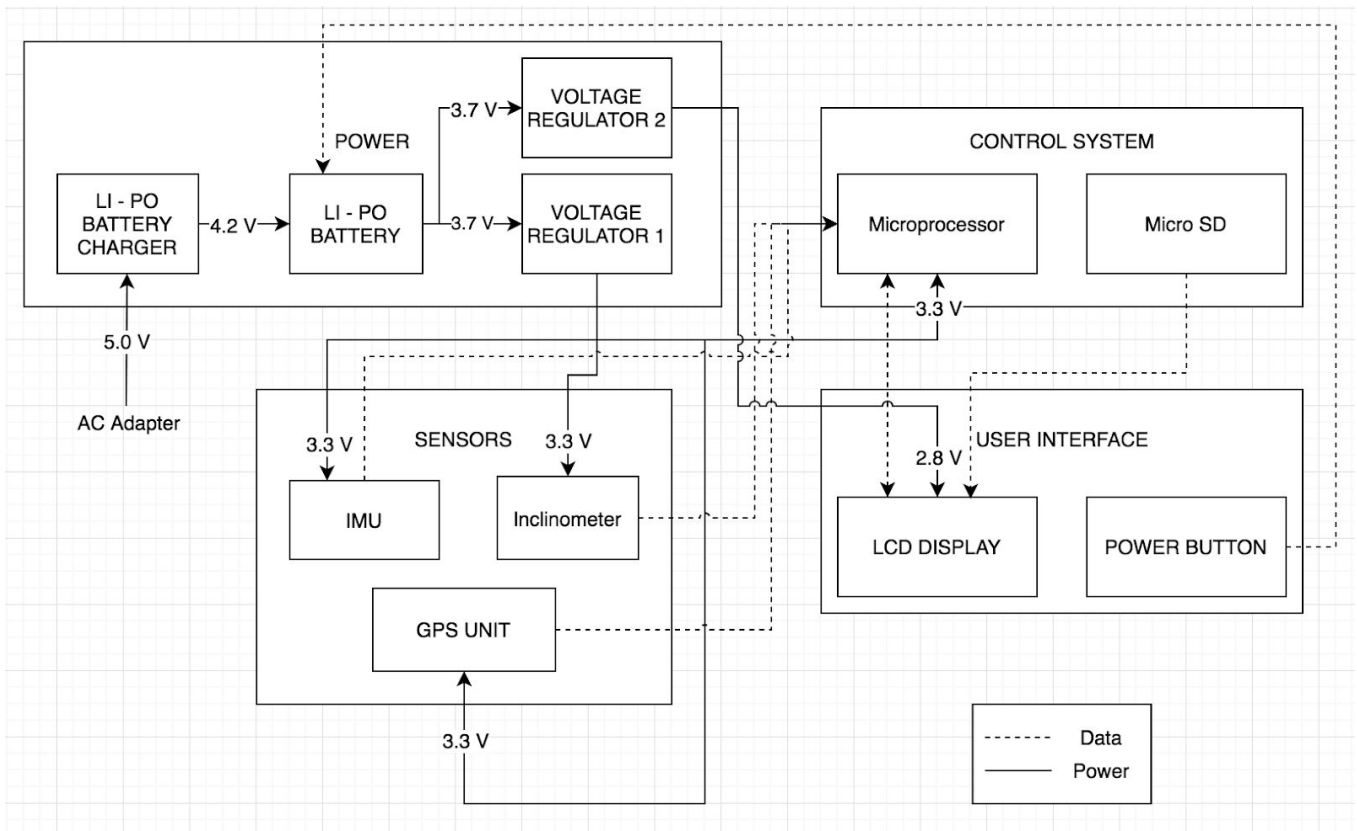


Figure 2. Block Diagram

The design of the survival wristband consists of four subsystems. The power subsystem is responsible for providing adequate power to the components in the other subsystems so that the survival wristband can function. The sensors subsystem is responsible for collecting the necessary data to be sent to the

control system. The control system is responsible for taking all of the sensor data and calculating the correct direction to be displayed. The user interface subsystem is responsible for receiving the data from the control system and displaying the proper directional arrow accordingly. The user interface subsystem is also responsible for powering on/off the wristband.

2.2 Physical Design



Figure 3. Top View Physical Design [3]

Figure 3 shows a rough approximation of the end product of the survival wristband. The LCD display will display an arrow, a status bar and the actual 3 angle measurements for demo verification. In reality, the survival wristband will likely be much bulkier and larger than shown in Figure 3. The PCB will likely be underneath the LCD display and the battery may go on the underside of the survival wristband. The exact location of every component is still unknown at this time.

2.3 Subsystems

2.3.1 Li-Po Battery

The Li-Po battery is crucial to powering all of the components of the survival wristband. The battery will power the sensors, microcontroller and LCD display. This battery will lead into two separate voltage regulators to supply the rest of the components of the survival wristband. After looking at all of the components of the survival wristband and their rated voltages and currents, we are able to adequately characterize the battery necessary for power. The GPS unit that we are using requires an average supply current of 40 mA but in an effort to reduce the overall power, we will be operating it in low-power mode where the average supply current is reduced to 11 mA. The inclinometer that we will be using has a current consumption of 1.2 mA in standard operation mode. The LCD display that we are using requires the largest current at 60 mA while the IMU unit has a total supply current of 12.3 mA at a reasonable operating point. For 8 MHz operation of the ATmega1280 microcontroller [10], the typical current usage is approximately 7 mA as well. We will be operating the microcontroller at 8 MHz instead of the maximum frequency of 16 MHz due to a much lower required current for operation and the frequency still being quick enough to meet our data transfer requirements. With the addition of all of these currents, the total current that needs to be supplied by the battery is 92.4 mA. We require a fairly significantly long battery life for the survival wristband and, after deciding that eight hours of battery life is sufficient, chose a lithium polymer battery (LP-503562) [8] that is rated for 1200 mAh and will supply approximately 12 hours of battery life. This particular battery is rated for 3.7 V which will also be able to supply sufficient voltage to every component as a majority of the components will operate at approximately 3.3 V while the LCD display will operate at a lower voltage. The Li-Po battery is also rechargeable so that the user is able to charge the power source without having to replace it periodically. Size was also a concern in choosing the battery to keep the wristband as lightweight and compact as possible. But upon exploring a multitude of possible batteries, there is not much freedom when it comes to size with the voltage and current specifications that we require. The LP-503562 battery [8] that we will be using is not the ideal size which will make it bulkier than we originally intended. However, it will still be able to fit into the wristband and perform as expected.

Requirements	Verification
<ol style="list-style-type: none"> 1. The battery stores 800 mAh or larger and supplies a voltage of between 3.6 V and 4 V. 2. Ensure that the battery operating temperature remains under 45 °C. 	<ol style="list-style-type: none"> 1. <ol style="list-style-type: none"> A. Ensure that the battery is fully charged and has a voltage between 3.6 V and 4 V. If not supply 4.2 V DC to battery according to datasheet until battery voltage is 3.6 V to 4 V. B. Using a constant current load of 800 mA, discharge the battery for exactly one hour. C. Measure the voltage across the battery and ensure that the discharge voltage is 3 V or greater as in accordance with the datasheet. 2. <ol style="list-style-type: none"> A. While completing steps 1.B-C, use the IR thermometer to ensure that the battery operating temperature remains under 45 °C.

Table 1. RV Table for Battery

2.3.2 Li-Po Battery Charger

Because the battery that we will be using is rechargeable, we will need a battery charger that is user friendly and compatible with the specific Li-Po battery. This battery charger will need to be able to take the 120 V AC from an outlet and convert this to 4.2 V DC in order to be able to charge the Li-Po battery that we require. In order to meet these requirements, we will be using the USB Li-ion/Li-Po charger [9] from Adafruit Industries. This charger is convenient as USB connections are fairly common so the user will easily be able to charge the survival watch with access to any outlets. The standard charging current supplied by this battery charger is 500 mA but a resistor can be soldered on the PCB of the charger so that the charging current can be reduced. The Li-Po battery can be charged in a quick charge mode which requires 1 A or a standard charge mode which requires 200 mA. With the Adafruit battery charger, we will be able to charge the battery fairly quickly.

Requirements	Verification
<ol style="list-style-type: none"> 1. The Li-Po battery must be able to be charged to 3.6 V to 3.7 V when a continuous 4.2 V from the Adafruit USB charger is applied 2. The Li-Po battery must remain below its maximum charging temperature of 45 °C during the charging process. 	<ol style="list-style-type: none"> 1. A. Discharge the Li-Po battery completely until it is approximately 3.0 V. B. Connect the Adafruit USB charger to the Li-Po battery and monitor the battery voltage during the charging cycle to ensure that the battery can be charged to 3.6 V to 3.7 V 2. During steps 1.A-B, use the IR thermometer to ensure that the battery temperature remains under 45 °C.

Table 2. RV Table for Battery Charger

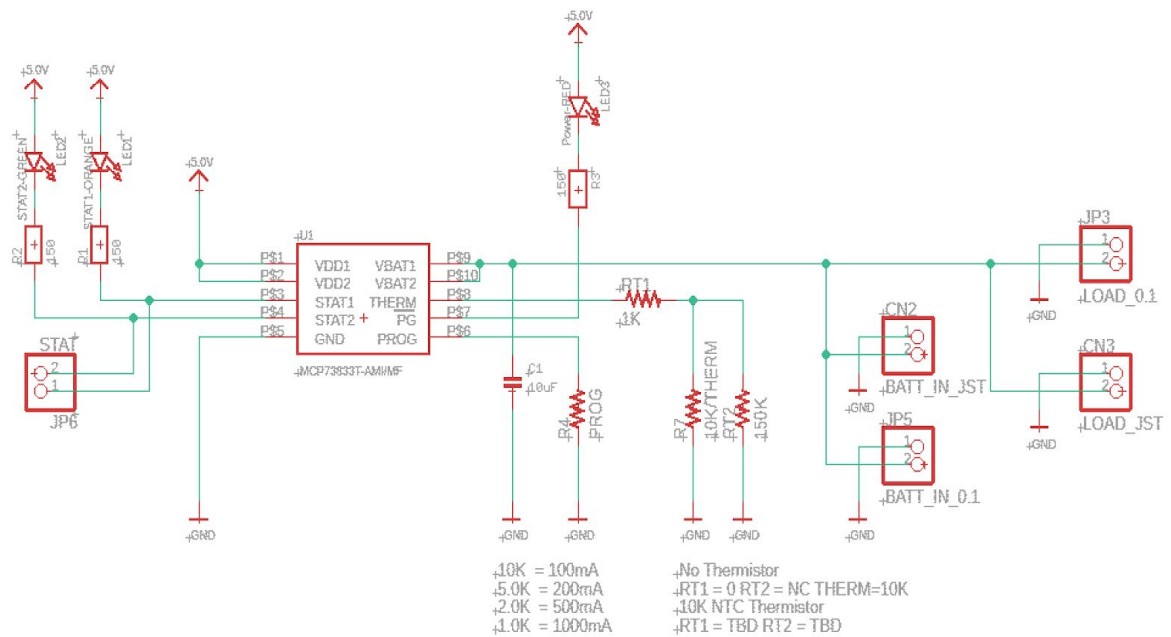


Figure 4. Battery Charger Schematic 1 [18]

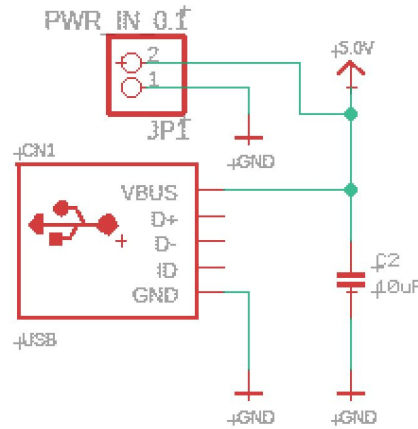


Figure 5. Battery Charger Schematic 2 [18]

2.3.3 Voltage Regulator(s)

To be able to successfully power all of the components of the survival wristband, we will require two voltage regulators. These two voltage regulators will both connect to the Li-Po battery and produce fixed output voltages that will be used by other components of the wristband. We will need one voltage regulator that is able to step down the 3.7 V battery voltage to 3.3 V. This voltage regulator will then provide this 3.3 V to the GPS unit, the inclinometer, the microcontroller and the IMU unit. The overall current that all of these components will require is approximately 25 mA so it is important that the first voltage regulator will be able to supply at least 25 mA. With these specifications in mind, we chose the TPS75833 [7] voltage regulator that produces a fixed output voltage of 3.3 V from a 5.5 V max input voltage or lower. This fixed output voltage is able to be sustained for an output current range of 0 - 3 A. This 3 A maximum is much larger than the 25 mA that we require but this voltage regulator is the most ideal one to meet the voltage requirements.

The second voltage regulator that we require will step down the 3.7 V battery voltage to approximately 2.8 V. This 2.8 V will then be provided to the LCD display to power it. The current that the LCD display requires is approximately 60 mA so it is important that this voltage regulator will be able to supply at least 60 mA. We then chose the MCP1700 [6] voltage regulator that is able to provide a fixed 2.8 V output voltage from a max 6.0 V input voltage or

lower. The output current that the MCP1700 [6] is able to provide is 0 - 235 mA which will be adequate enough to supply the LCD display.

Requirements	Verification
<ol style="list-style-type: none"> 1. For the TPS75833, with a supply voltage of 3.5 V to 3.7 V, output a voltage between 3.25 V and 3.35 V at a current of 0 to 25 mA. 2. For the MCP1700, with a supply voltage of 3.5 V to 3.7 V, output a voltage between 2.75 V and 2.85 V at a current of 0 to 60 mA. 3. Ensure that the operating temperature remains under 45 °C. 	<ol style="list-style-type: none"> 1. A. Use a DC power supply set between 3.5 and 3.7 V with a current limit of 25 mA as a voltage supply to the voltage regulator. B. Meter the output voltage of the regulator and ensure that the voltage is within the range of 3.25 V to 3.35 V. 2. A. Use a DC power supply set between 3.5 and 3.7 V with a current limit of 60 mA as a voltage supply to the voltage regulator. B. Meter the output voltage of the regulator and ensure that the voltage is within the range of 2.75 V to 3.35 V. 3. A. While completing steps 1.A-B and steps 2.A-B, use the IR thermometer to ensure that the temperature of the regulator remains at or under 45 °C.

Table 3. RV Table for Voltage Regulators

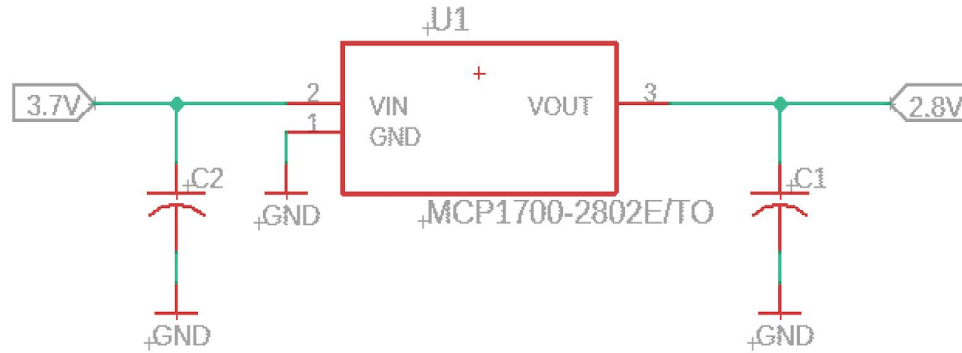


Figure 6. MCP1700 Voltage Regulator

2.3.4 Power Button

The power button will be used to power down the survival wristband when it is not in use. It will interface with the power supply and essentially disconnect the battery power supply from the sensors, microcontroller and display. The power button is useful to the survival wristband as it is able to allow the survival wristband to shut down, conserve power and prolong the time between charges of the battery. The power button will also be used to power up the survival wristband from an off state when its use is needed. We decided to go for a push button (GPTS203211B) [16] that is able to disconnect the Li-Po battery from the rest of the components. This button is rated for a maximum voltage across it of 30 V and a maximum current of 1 A. Both of these ratings will be more than sufficient for the 3.7 V battery and the current being drawn from the battery.

Requirements	Verification
<ol style="list-style-type: none"> 1. There must be approximately no power being supplied to the sensors, microcontroller and display after 30 seconds of pressing the button to turn the wristband off. 2. The sensors, microcontroller and display must be receiving approximately rated power within 30 seconds of pressing the button to turn the wristband on. 	<ol style="list-style-type: none"> 1. A. After having the wristband powered on for at least 2 minutes, press the button to power off. B. Wait 30 seconds and meter voltage and current across each component, confirming that the power being used is approximately 0 W. 2. A. After having the wristband powered off for at least 2 minutes, press the button to

3. The button must be easily accessible and easy to press.	<p>power on.</p> <p>B. Wait 30 seconds and meter voltage and current across each component, confirming that each component is receiving approximately rated power.</p>
--	--

Table 4. RV Table for Power Button

2.3.5 Inclinometer

To be able to correctly display the direction of the shortest path to the surface, the angle of inclination in two-dimensional space is needed. The output of the two-axis inclinometer, SCL3300-D01 [14], will be two angles of inclination with respect to the x and y defined axes. These two angles are needed in determining the rotation and orientation of the users wrist with respect to the calculated normal force direction. These angles are necessary in displaying the correct direction of the shortest path to the surface while taking into account the user rotating and moving their wrist. The inclinometer outputs 16 bit 2's complement digital data to represent the two angles of inclination and has an SPI interface which will transmit data to the microprocessor. The inclinometer will be powered from the power subsystem and requires a supply voltage between the ranges of 3.0 V to 3.6 V and draw a current of 1.2 mA under normal temperature operating conditions of -40 °C to 125 °C. We chose this particular inclinometer because it is relatively inexpensive in comparison to other inclinometers and fairly small in size. The inclinometer must also operate at 45 °C or under to ensure user safety.

Requirements	Verification
<ol style="list-style-type: none"> 1. The inclinometer will correctly read the angle of inclination for 2-axii within an accuracy of 10 degrees (5 degrees either direction). 2. Maintains an operating temperature of under 45 °C. 	<ol style="list-style-type: none"> 1. A. Connect the inclinometer to a 3.3 V DC supply. B. Rest the inclinometer on a flat surface such as the ground. C. Connect the SPI output to any microprocessor using the inclinometer data sheet guide.

	<p>D. Utilize the data sheet of the inclinometer, convert the 16 bit outputs to physical angles and verify that the two angles are within +/- 5 degrees.</p> <p>2. Throughout the cycle of 1.B-D, use an IR thermometer to make sure that the operating temperature remains under 45 °C.</p>
--	--

Table 5. RV Table for Inclinometer

2.3.6 IMU (Inertial Measurement Unit)

We require an IMU sensor for the magnetometer that is included within it. We need a magnetometer for the 2D compass direction that we are able to obtain from its data so that we can have a frame of reference for the angle measurements of the inclinometer. We chose to include an IMU because magnetometers alone are expensive but, when included in an IMU, are less costly. The IMU that we chose is the BNO055 [11] and it will be powered by the power subsystem and more specifically the output of the 3.3 V voltage regulator. We require the euler heading orientation output for the 2D compass direction and this output data format is 2 bytes of digital data. The IMU has a UART digital output interface for sending data to the microcontroller.

Requirements	Verification
1. The IMU will be able to output the compass orientation within +/- 4°.	<p>1. A. Connect the IMU according to its datasheet to an arduino (uno or redboard).</p> <p>B. Using another compass (iphone compass), orient the IMU to be facing due north.</p> <p>C. Ensure that the data collected from the IMU shows the compass orientation to be</p>

within $\pm 4^\circ$ of due north

Table 6. RV Table for IMU

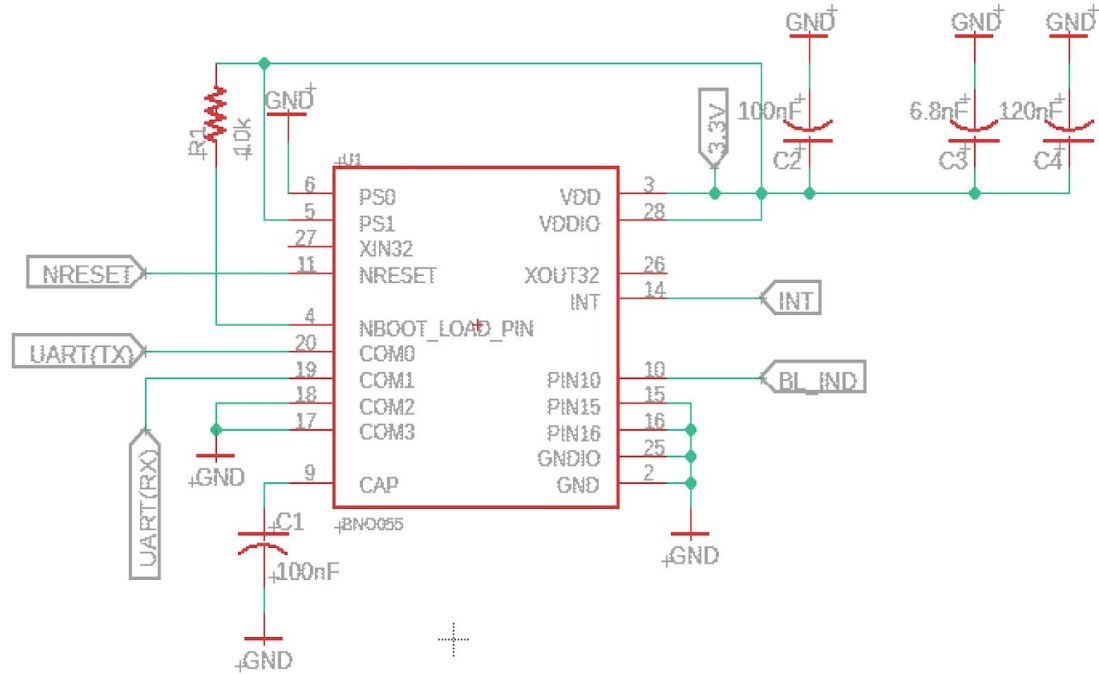


Figure 7. IMU Schematic

2.3.7 GPS Unit

The GPS unit that we require and will be using within our survival wristband is the NEO-6M GPS [13]. We will be using this GPS unit for a variety of reasons. A GPS unit is necessary for the survival wristband in order to calculate the correct direction of the normal force. This GPS unit outputs the current longitudinal and latitudinal coordinates as well as the current altitude with respect to sea level. The longitudinal and latitudinal coordinates are needed to calculate the change in the horizontal direction between data points while the built in altimeter is needed to calculate the change in the vertical direction between data points. The GPS unit will be powered by the power subsystem, and more specifically the 3.3 V voltage regulator. The changes in the horizontal and vertical directions will be computed within the microcontroller. The NEO-6M [13] was chosen because it operates at a reasonably low voltage, includes an altimeter and is able to interface with the ATmega1280 microcontroller due to its SPI interface ports. It is also relatively small in physical size and contains a backup power supply.

Requirements	Verification
<ol style="list-style-type: none"> 1. The NEO-6M GPS unit must be able to output the correct latitude and longitude within +/- 4 meters. 2. The NEO-6M GPS unit must be able to output the correct altitude within +/- 8 meters. 	<ol style="list-style-type: none"> 1. A. Connect the GPS unit to an arduino (uno or redboard) according to the datasheet. B. Determine the latitude and longitude values as well as the altitude of a specific position using another application (Iphone) C. Place the GPS in that specific position and ensure that the data collected from the NEO-6M is within +/- 4m of the measured latitude and longitude and +/-8m of the measured altitude.

Table 7. RV Table for GPS Unit

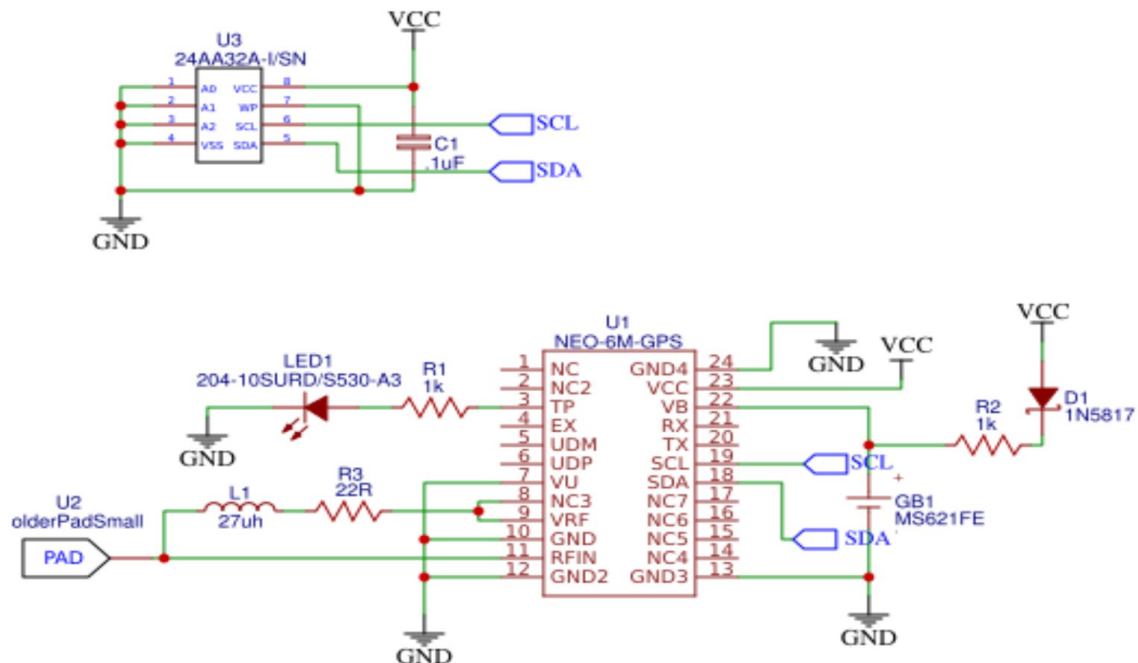


Figure 8. GPS Schematic [17]

2.3.8 LCD Display

The LCD display, ST7789 [12], will communicate with the microcontroller and receive data to determine what needs to be displayed. The LCD display has 8 digital pins used for I/O that are used to communicate with the microcontroller in order to display the correct images. The LCD display will consist of a status bar that is split into 5 sections and will display one red bar if the user is facing 180 degrees away from the correct direction to dig. Every 36 degrees closer to the correct direction, another bar of the display will be filled until the user is within 36 degrees of the correct direction, in which case all five bars will be filled and green. When 2-4 bars are filled, the color will be yellow. The display will also be able to display 26 three dimensional arrows in order to correctly communicate the digging direction to the user. The actual three angles (XY, XZ, YZ) will be displayed underneath this status bar for demo purposes where the margin of error for the displayed arrow can be verified with the actual raw data. This LCD display will also be powered by the power subsystem and will require an analog supply voltage between 1.65 and 3.3 V with a typical value of 2.8 V. The logic supply voltage of the display is also between 1.65 V and 3.3 V with a typical value of 2.8 V. The frame frequency of the LCD display is also 65 Hz.

Requirements	Verification
<ol style="list-style-type: none">1. Images on the screen must be clearly visible when the screen is tilted 60 degrees towards and away from the viewer.2. The LCD display is able to display the graphic images stored on the micro-sd.3. The LCD display must be able to refresh in 2 seconds or under.	<ol style="list-style-type: none">1. With the LCD powered on, and an image being displayed, tilt the wristband approximately 60 degrees towards and away from you and verify that the image is still clearly visible.2. After the startup phase of the device has completed, a graphic image from the micro-sd is being displayed.3. After the startup phase is complete, the user tilts their wrist by an angle less than 90° and verify that the arrow being displayed changes accordingly in 2 seconds or less.

Table 8. RV Table for LCD Display

2.3.9 Microcontroller

The microcontroller that is going to be used for this project, an ATmega1280 [10], will act as the primary source for handling the communication. It reads from the inclinometer and the GPS unit, and writes to the display via SPI while reading from the IMU via UART. In other words, the microcontroller will send the required signals to display the correct arrow on the display, while getting the data signals from the sensors. It is the primary source of the control system and controls the output of the device. We chose this microcontroller because it has both SPI and UART communication protocols as well as having a maximum frequency of 8 MHz while having 3.3V in its operating Vcc range (1.8 - 5.5 V). The frequency is needed to satisfy the refresh rate clause on the high level requirements (1.3, 1st bullet point).

Requirements	Verification
<ol style="list-style-type: none">1. Microcontroller must be able to read and write data via SPI and UART.2. Microcontroller must be able to read and write via SPI at speeds more than 4.5Mbps.3. Microcontroller must be able to read and write via UART at speeds more than 115.2kbps.	<ol style="list-style-type: none">1.<ol style="list-style-type: none">a. To test UART speeds, first connect the microcontroller to PuTTY terminalb. Connecting the microcontroller to FT4222, a USB-UART bridgec. Complete the setup of the terminal to 115.2kbaudd. Send 100 characterse. Echo them backf. Verify that they match2.<ol style="list-style-type: none">a. To test SPI speeds, first connect the microcontroller to PuTTY terminalb. Connecting the microcontroller to FT4222, a USB-SPI bridgec. Complete the setup of the terminal to 4.5Mbaudd. Send 100 characters

	<p>e. Echo them back</p> <p>f. Verify that they match</p>
--	---

Table 9. RV Table for Microcontroller

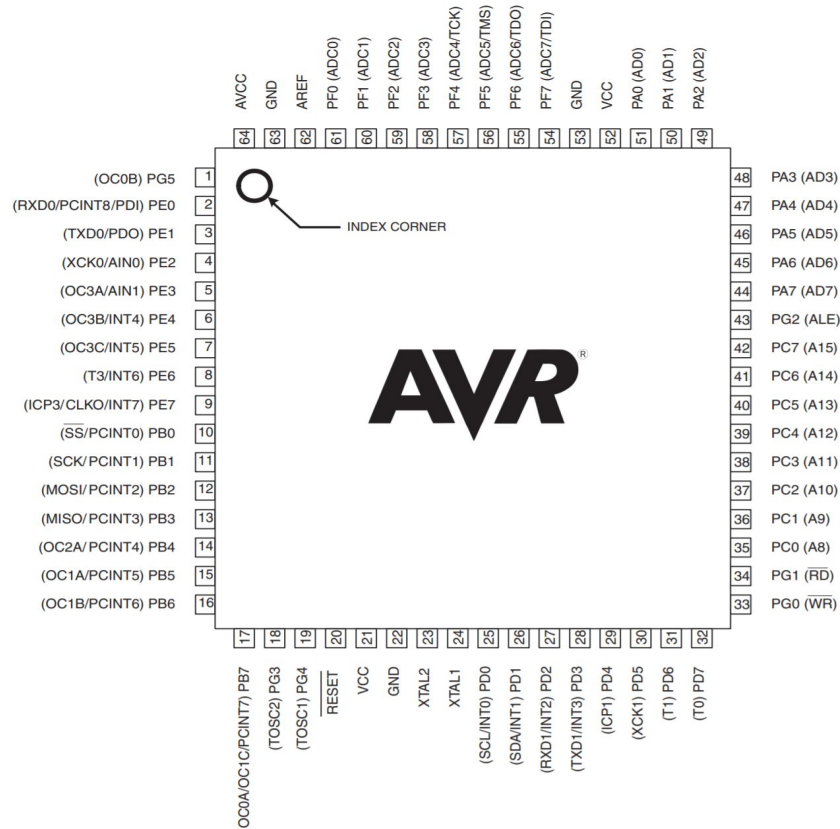


Figure 9. Microcontroller Pinout [10]

2.3.10 Micro-sd Card

The Micro-sd card we will use for the project, a SanDisk Ultra 32GB microSD [15], will be connected to the display and will be used to store the 26 different arrows that will be displayed on the LCD screen. In other words, we will store the 26 different sprite sheets on the Micro-sd card and when the proper signal comes from the microcontroller to the display, the selected sprite sheet will appear on the display.

Requirements	Verification
<ol style="list-style-type: none"> 1. The micro-sd card must have enough storage to store 26 different sprite sheets, with each of them having 100*100 pixels, in its memory. 2. The micro-sd card must properly be connected to the LCD display and must be able to transfer data. 	<ol style="list-style-type: none"> 1. Connect the Micro-sd to a computer. 2. Make sure to observe all 26 sprite sheets on the computer with a 100*100 pixels resolution.

Table 10. RV Table for Micro-SD

2.4 Tolerance Analysis & Software

The most critical feature of this project is its ability to calculate the direction of shortest path to the surface within a reasonable margin of error of +/- 22.5 degrees. Without this, the project has no applicable real-world use-case. In determining the whether our design meets the above interval or not, we must consider the two main calculations:

1. Calculate the 3 planar angles of the surface relative to cardinal North and East.
2. Calculate the 3 planar angles of the device relative to cardinal North and East.

For the first calculation, the outputs of the GPS are solely responsible for determining the slope of the surface. These outputs are altitude (h), latitude (ϕ), and longitude (λ). Latitude and longitude coordinates, when converted to horizontal position coordinates, are accurate within +/- 2.5 meters according to the data sheet [13]. The altitude reading is accurate within < +/- 5.0 meters according to a thorough presentation on analyzing the GPS model's altitude readings [5]. Given that the average slope of the surface the user is on is considered the target measurement, we can measure the margin of error of our result based solely on the deviations of the measured data points (λ , ϕ , h) during our calculation of the slope. The equations for these two are:

Let λ_0 be the central meridian, R be the radius of Earth, and ϕ_m be the latitude of midpoint between the two data points. Running down the below calculations, starting with two (λ , ϕ , h) points, we get:

$$B_x = \cos\phi_2 * \cos(\lambda_2 - \lambda_1) \quad (1)$$

$$B_y = \cos\phi_2 * \sin(\lambda_2 - \lambda_1) \quad (2)$$

$$\phi_m = \text{atan}^2(\sin\phi_1 + \sin\phi_2, \sqrt{(\cos\phi_1 + B_x)^2 + B_y^2}) \quad (3)$$

$$X = x_2 - x_1 = R(\lambda_2 - \lambda_0)\cos\phi_m - R(\lambda_1 - \lambda_0)\cos\phi_m = R(\lambda_2 - \lambda_1)\cos\phi_m \quad (4)$$

$$Y = y_2 - y_1 = R(\phi_2 - \phi_0) - R(\phi_1 - \phi_0) = R(\phi_2 - \phi_1) \quad (5)$$

$$Z = h_2 - h_1 \quad (6)$$

$$\theta_{XZ} = \tan^{-1}(Z/X) \quad (7)$$

$$\theta_{YZ} = \tan^{-1}(Z/Y) \quad (8)$$

$$\theta_{XY} = \tan^{-1}(Y/X) \quad (9)$$

where equations 7, 8 and 9 are the final theoretical angles of incline of the XZ, YZ, and XY planes respectively. The software will be taking in readings from the GPS for the (λ , ϕ , h) data points, and running through these calculations to determine the angle of incline of the surface for displaying the 3 dimensional direction of the device relative to the direction of this incline. The direction of incline of the device will be explained and analyzed later on. The final resulting direction to display is determined by the difference between the angles of incline of the surface and the angles of incline of the device.

Consider the maximal effect of a +/- 5 meter deviation on the altitude of two data points. This would result in an error of 10 meters in determining the altitude difference between the two points. When analyzing the tolerance of the GPS in terms of the altitude, we need only examine the scenario where the actual change in altitude is 0 meters to determine its maximal effect on the calculation of the XZ and YZ angles of the surface; as the true difference in altitude increases, the significance of the 10 meter variation is reduced, so 0 meters causes the variation causes this variation to be maximally pronounced. We can see this clearly by the calculation

$$\theta_{S, XZ} = \tan^{-1}\left(\frac{Z+10}{X}\right) \quad (10)$$

where Z is the the altitude difference, X is the x-positional distance, and theta is the angle of the surface along the XZ plane (this equation can be applied to the YZ plane as well, just replace X with Y, the y-positional distance).

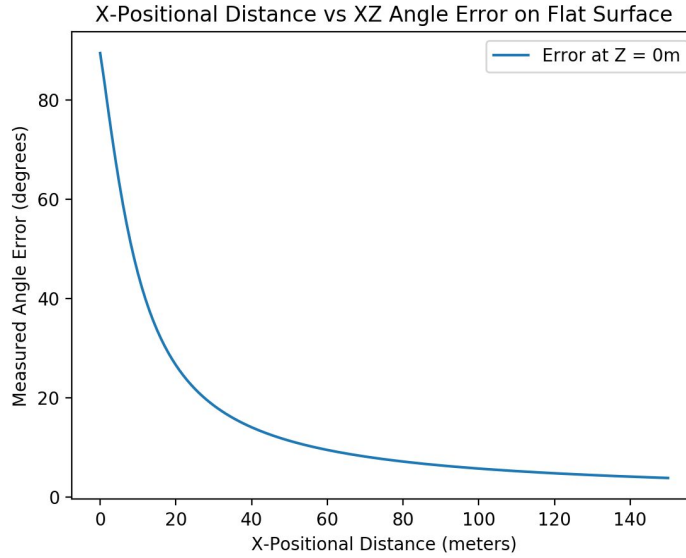


Figure 10: Graph plotting the effect of an applied 10 meter error in measured Z distance on calculating the angle of the surface in the XZ plane. This is with respect to keeping the actual change in altitude as 0 meters. Note that the X positional distance is 24.14 meters when the error in angle measurement is 22.5 degrees.

The angle error, as shown in figure 1, is greater than 22.5 degrees when the distance between the two data points is below 24.14 meters. This same analysis and its results can be applied to the Y positional distance where the measured angle error is with respect to the YZ plane. This is because the same tolerance and equations are used for both calculations. To correct this issue, data points whose X and Y values are within 25 meters of the previous data point's X and Y values respectively will be considered noisy data and will be filtered out accordingly. For instance, in the case where the user is barely walking somewhere and the error in altitude measurement is therefore highly pronounced, it'd be sufficient to take previous data points further than 25 meters from the current data point to use in recalculating the average slope. To determine the angle measurement error in the XY plane, we must use the equations

$$\theta_E = \theta_{S,XY} = \tan^{-1}\left(\frac{Y+5}{X-5}\right) \quad (11)$$

$$\theta_T = \theta_{S,XY} = \tan^{-1}\left(\frac{Y}{X}\right) \quad (12)$$

Equation 10 expresses the maximal effect of the tolerance resulting from the +/- 2.5 meter variation in the output X and Y positions. In determining the margin of error produced by this variation, we must calculate the magnitude of the difference of these two calculations. The below plot expresses this as the measured angle error.

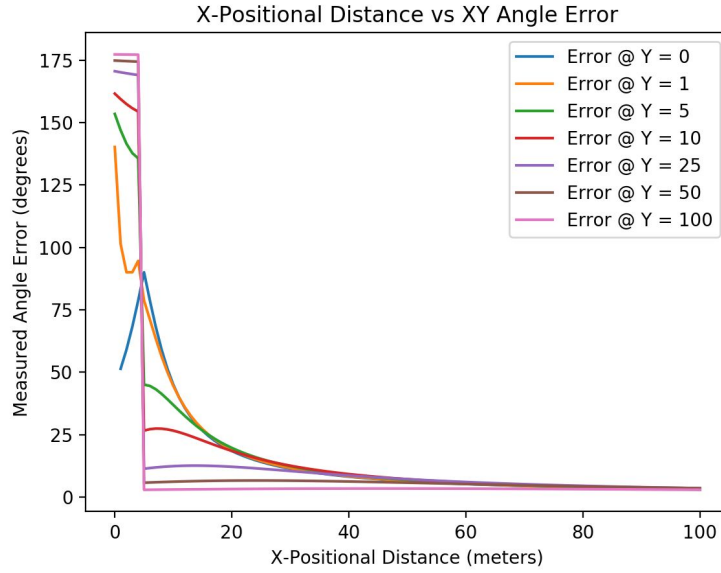


Figure 11: Graph plotting the effect of an applied 5 meter error in both X and Y distances on calculating the angle of the surface in the XY plane. Note that the error is most consistently high, excluding the initial spikes of all plots, when $Y = 0$. Given $Y = 0$, the error in angle measurement is less than 22.5 degrees when the X positional distance is 17.07 meters or less.

As figure 2 demonstrates, the tolerance affects the fulfillment of the project's requirement of ± 22.5 degrees error when the X distance is 17.07 meters or less. However, since this interval is within the interval of 0 to 25 meters, it is considered noisy data and will thus be filtered.

Next one must consider the tolerance resulting from the calculation of the device's three planar angles. Let θ_1 be the compass heading output from the IMU and θ_2 be the output y angle of the inclinometer. θ_1 has an accuracy of within 2.5 degrees [11]. θ_2 has an accuracy of within 1.15 degrees [14]. The below equations calculate the coordinates of the vector for the inclinometer y angle output and the compass heading in 3 dimensions.

$$\begin{aligned} x_2 &= 0 & y_2 &= \cos(\theta_2) & z_2 &= \sin(\theta_2) \\ x_1 &= \sin(\theta_1) & y_1 &= \cos(\theta_1) & z_1 &= 0 \end{aligned} \quad (13)$$

Using the angle between two 3D vectors and the fact that we are only concerned with direction and there are using unit vectors, we can deduce

$$\theta_{D,YZ} = \cos^{-1}(x_1 * x_2 + y_1 * y_2 + z_1 * z_2) = \cos^{-1}(y_1 * y_2) \quad (14)$$

which is the angle of the device along the YZ plane. Note that the same procedure of calculation applies to the XZ plane as well. The following analysis is therefore identical to the XZ angle calculation.

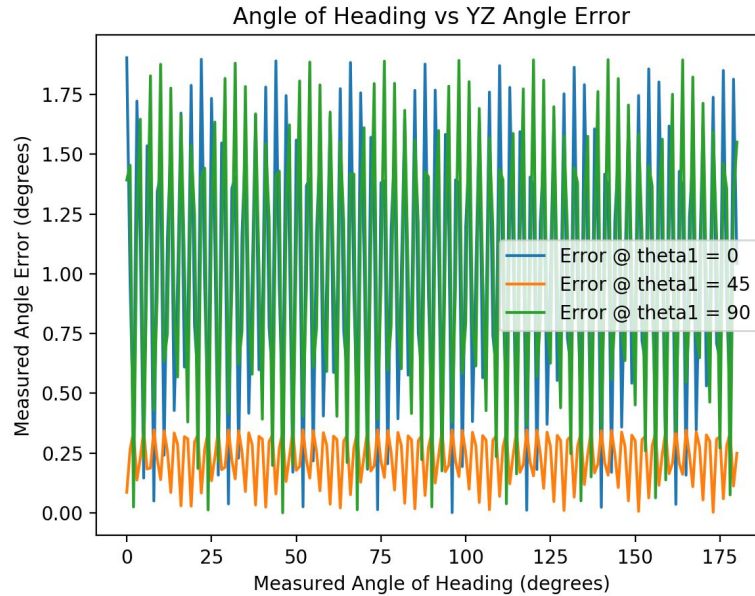


Figure 12: Graph showing the error in measured angle on the YZ plane as a result of maximal variation in the outputs of the inclinometer and IMU compass setting. As one can see, the angle error stays within 2 degrees indefinitely.

The calculation of the device's orientation in the YZ plane, and therefore XZ plane, are well within the margin of error of 22.5 degrees. Also, given that the angle of the XY plane with respect to the device is directly outputted from the IMU, and the IMU has an accuracy of within 2.5 degrees, all three planes with respect to the device are within the required margin of error and therefore do not need further investigation.

3. Cost and Schedule

3.1. Cost Analysis

Some assumptions that need to be made for labor costs are that the salary of an electrical engineer involved with this project is approximately \$40 per hour and the number of hours a week is, on average, 10 hours. With the course being 16 weeks long, the labor costs for 3 people can be calculated as:

$$\text{Labor costs} = \$48,000 = 3 * \frac{\$40}{\text{hr}} * \frac{10 \text{ hrs}}{\text{week}} * 16 \text{ weeks} * 2.5 \quad (15)$$

We need to make a couple of assumptions with regards to the total parts cost. One of these assumptions is that the cost of our PCB board will be approximately \$30. Another assumption we need to make in determining cost is that the physical casing of the wristband will cost about \$20 in materials.

Description	Manufacturer	Part #	Quantity	Cost
Li-Po Battery	Adafruit Industries	LP-503562	1	\$9.95
3.3 V Voltage Regulator	Texas Instruments	TPS75833	1	\$5.88
2.8 V Voltage Regulator	Microchip Technology	MCP1700	1	\$0.37
USB Li-ion/Li-Po Charger	Adafruit Industries	USB Lilon/LiPoly charger – v1.2	1	\$12.50
Microcontroller	Microchip Technology	ATMEGA1280 -16AUR	1	\$10.77
IMU	Bosch Sensortec	BNO05	1	\$11.16
LCD Display	Adafruit Industries	ST7789	1	\$19.95
GPS Unit	U-blox	NEO-6M	1	\$16.67

Inclinometer	Murata Electronics	SCL3300-D01	1	\$40.61
Micro-SD Card	SanDisk	U1 A1 – SDSQUAR-03 2G-GN6MA	1	\$7.98
Power Button Switch	CW Industries	GPTS203211B	1	\$1.53
PCB	N/A	N/A	1	~\$30.00
Wristband Casing	ECE Department	N/A	1	~\$20.00
Total				\$187.37

Table 11. Parts Cost

Section	Cost
Labor	\$48,000
Parts	\$187.37
Total	48,187.37

Table 12. Cost Break Down

3.2. Schedule

<u>Week</u>	<u>Task</u>	<u>Responsibility</u>
2/24	Preparation of the Design Document	All
	Attending the Design Document Check	All
	Circuit Schematics for Modules	John
	Mathematical model and calculation of margin of errors	Derek
	Part selection for every subsystem	Bartu
3/2	Preparing for the Design Review Presentation	All
	Discussion with the Machine Shop	Derek
	Ordering parts	John
	Initial design of the PCB board layout	Bartu
3/9	Initial hardware assembly	Derek
	Soldering Assignment	All
	Building and testing of the battery module	Bartu & John
3/16	Implementation of the mathematical model onto the microcontroller	John
	Creating the different sprite sheets for the display	Derek
	Integration of the power button to the battery module	Bartu

3/23	Finishing the battery module with the regulators and the button	John
	Combining the mathematical model with the software implementation	Derek
	Finishing the PCB design with the circuit schematics for every submodule	Bartu
3/30	Revision and reordering (if needed) of the PCB board	John
	Individual Progress Reports	All
	Finishing the sprite sheets and assembling the memory and the display with the microcontroller	Derek & Bartu
4/6	Finishing the mathematical model (software) and assembling the sensors with the microcontroller	Derek
	Final check of all the submodules and last reorder round	John
	Connecting the battery module to the rest of the device	Bartu
4/13	Verifications and tests	John
	Start the work for the extra credit opportunities	Bartu
	Debugging	Derek
4/20	Preparing for the demonstration	Derek
	Last adjustments and tests	John
	Starting the final report	Bartu

4/27	Demonstration	All
	Review and adjust the extra credit work	John
	Prepare for presentation	Bartu & Derek
5/4	Presentation	All
	Finalize and submit the extra credit work	All
	Finishing the final report	All

Table 13. Table featuring the schedule for the rest of the semester

4. Discussion of Ethics and Safety

Ethical values were our driving force when we were brainstorming during the idea creation process. Our product aims to find the shortest path out of a snow or debris in the event of an avalanche, landslide or mining collapse which can create the small difference between surviving and dying. A product that's main aim is to help people survive from a natural disaster is highly ethical. We always prioritized the well-being of our users, which made us come up with this idea. This approach aligns with the 1st clause of the IEEE code of ethics, "to hold paramount the safety, health, and welfare of the public, ... might endanger the public or the environment" [4].

While the main idea behind our product is to accentuate the safety of the user, we also need to think about how reliable and consistent the product will be. A power outage, overheating or a high margin of error on the microcontroller can have a huge impact. To avoid a situation like this we regulated our battery with a power supply with a voltage regulator connected to it. This will make our battery reliable, eliminating the possible power outage issue. Considering the possibility of an overheating case, we will always keep the temperature in between the battery's operating temperature values.

We are also aware of the fact that we always need to try to find the shortest path out of the debris/snow/water. So, in our calculations we always need to pinpoint the slope of our current location. We will bear this in mind while producing our software so that a "calculation error", or a high percentage of systematic error, won't be a safety hazard. As stated on the 3rd clause of the IEEE code of ethics, "to be honest and realistic in stating claims or estimates based on available data" [4], we prioritize being both honest and realistic in all the data and calculations we use.

Another safety hazard would be the use of a lithium ion battery. There are some cases where a damaged or punctured lithium ion battery can cause a problem. Thus, having a solid, reliable cover for the battery will protect it from getting damaged in most of the instances. Since we are going to use a rechargeable battery, we also need to consider a possible scenario of overcharging. To prevent overcharging and a possible event of fire, we will closely check and make sure the battery specifications are suitable both before purchasing and before using it. In addition, we will never let the battery to be charged above its “final charge voltage” to again prevent any possible cathode breakdown that might cause fire.

The last safety hazard is again part of making our product more reliable. Our battery will have sufficient power so that the device won’t die while it is in use. This is why the battery will be powerful enough to run at the maximum current rating for 8 hours. We are aware that the device dying while it is much needed might be fatal, so it is another ethical priority that we consider while choosing the parts of the device.

We will keep trying to make our product better by making the required adjustments throughout the project to make the device more effective, reliable and consistent. This approach aligns with the 6th and 7th clause of the IEEE code of ethics, “to seek, accept, and offer honest criticism of technical work, to acknowledge and correct errors, and to credit properly the contributions of others” [4].

5. Citations

[1] “11 Facts About Avalanches,” *DoSomething.org*. [Online]. Available: <http://www.dosomething.org/us/facts/11-facts-about-avalanches>. [Accessed: 28-Feb-2020].

[2] “NOVA Online | Avalanche! | Snow Sense,” *PBS*. [Online]. Available: <http://www.pbs.org/wgbh/nova/avalanche/snowsense.html>. [Accessed: 28-Feb-2020].

[3] “Cartoon Hand 3D Model,” *Free3D*. [Online]. Available: <https://free3d.com/3d-model/cartoon-hand-9109.html>. [Accessed: 28-Feb-2020].

[4] “IEEE Code of Ethics,” *IEEE*. [Online]. Available: <https://www.ieee.org/about/corporate/governance/p7-8.html>. [Accessed: 28-Feb-2020].

- [5] "Comparison of Precision Between Neo-M8N and Neo-6M Modules," *Youtube*. [Online]. Available: <https://www.youtube.com/watch?v=FmdG66kTfsl>. [Accessed: 28-Feb-2020].
- [6] "Datasheet for MCP1700," *Microchip*. [Online]. Available: <http://ww1.microchip.com/downloads/en/DeviceDoc/MCP1700-Low-Quiescent-Current-LDO-20001826E.pdf>.
- [7] "Datasheet for TPS75833KCG3," *TI*. [Online]. Available: <http://www.ti.com/lit/ds/symlink/tps758.pdf>.
- [8] "Datasheet for Li-Polymer 503562," *Pkcell*. [Online]. Available: https://cdn-shop.adafruit.com/product-files/258/C101-_Li-Polymer_503562_1200mAh_3.7V_with_PCM_APPROVED_8.18.pdf.
- [9] "Datasheet for MCP73833/4," *Microchip*. [Online]. Available: <https://cdn-shop.adafruit.com/datasheets/MCP73833.pdf>.
- [10] "Datasheet for ATmega640/V-1280/V-1281/V-2560/V-2561/V," *Atmel*. [Online]. Available: http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-2549-8-bit-AVR-Microcontroller-ATmega640-1280-1281-2560-2561_datasheet.pdf.
- [11] "Datasheet for BNO055," *Bosch*. [Online]. Available: https://ae-bst.resource.bosch.com/media/_tech/media/datasheets/BST-BNO055-DS000.pdf.
- [12] "Datasheet for ST7789," *Innolux*. [Online]. Available: https://cdn-shop.adafruit.com/product-files/3787/3787_tft_QT154H2201_____20190228182902.pdf.
- [13] "Datasheet for NEO-6M," *U-blox*. [Online]. Available: [https://www.u-blox.com/sites/default/files/products/documents/NEO-6_DataSheet_\(GPS.G6-HW-09005\).pdf](https://www.u-blox.com/sites/default/files/products/documents/NEO-6_DataSheet_(GPS.G6-HW-09005).pdf).
- [14] "Datasheet for SCL3300-D01," *Murata*. [Online]. Available: https://www.mouser.com/datasheet/2/281/datasheet_scl3300-d01-1532018.pdf.
- [15] "Datasheet for SanDisk Ultra 32GB microSD," *SanDisk*. [Online]. Available: <https://www.alliedelec.com/m/d/04db416b291011446889dbd6129e2644.pdf>.

[16] "Datasheet for GPTS203211B," *CW Industries*. [Online]. Available: <http://switches-connectors-custom.cwind.com/Asset/GPTS203211BR2.pdf>.

[17] "gps NEO-6M," *EasyEDA*. [Online]. Available: <https://easyeda.com/neelamrani/gps-neo-6m>. [Accessed: 28-Feb-2020].

[18] Adafruit, "adafruit/Adafruit-USB-Lilon-LiPoly-Charger-PCB," *GitHub*. [Online]. Available: [https://github.com/adafruit/Adafruit-USB-Lilon-LiPoly-Charger-PCB/blob/master/Adafruit USB Lilon-LiPoly Charger.sch](https://github.com/adafruit/Adafruit-USB-Lilon-LiPoly-Charger-PCB/blob/master/Adafruit%20USB%20Lilon-LiPoly%20Charger.sch). [Accessed: 28-Feb-2020].