Senior Design 1
Depth Perception Haptic System (DPHS)

University of Central Florida
Department of Electrical & Computer Engineering
Dr. Samuel Richie
- 120 Page Report -

Group 40
Christa Lawrence - Computer Engineering
Cristopher Matos - Computer Engineering
Kathryn Pagano - Electrical Engineering
Chad Pauley - Electrical Engineering

Table of Contents

Figures

Tables

1.0 Executive Summary

This report covers the conception, design, and development of the Depth Perception Haptic feedback System (DPHS)—a device which intends to aid the blind in navigating the world around them without the need for a cane or the help of others. The end goal of the document is to thoroughly define the hardware and software components of the system such that they may be implemented in a fully operational prototype.

The design will fit into a customly designed article of clothing. Embedded within the clothing will be an array of haptic feedback motors and a variety of peripherals under the operation of an embedded control system. The array of motors will be arranged such that they provide a sense of understanding of where an obstacle is detected over a three-by-three grid on the body. Given the mission of the device, it must be comfortable to use over long periods of time, have a long lasting charge, and implement one or more modes of navigation. In addition to design-specific requirements, strict standards for health and safety involving wearable electronic devices, and ethical design practices must be followed.

Multiple techniques for depth sensing were analyzed to determine the most accurate yet computationally and financially cost-effective solution. As a result, the device implements a Convolution Neural Network (CNN) designed for monocular depth estimation. This network is used in conjunction with single-point LiDAR to provide more accurate depth information as a safety precaution. Given the computational load of the sensing technique, the Raspberry Pi 4 Model B microprocessor was chosen for the control system due to its performance on inference benchmarks. The device will be used for real-time navigation of indoor and outdoor environments, so the software must be able to balance the needs of the LiDAR subsystems with the CNN, as well as sending output to the motors and speaker.

The constraints imposed on the project include: affordability, performance in environment, amount of testing needed, prevention of injury to health, lack of time for construction, and shortage of available resources to test products. Requirements act as a type of specification constraint that gives a range of goals for the end design. These include the physical attributes, usage and maintenance, user interface, and navigation modes concerning the overall expectations of the system.

Similar devices with a shared goal have achieved their objectives in several different ways than the DPHS. Examples of this include the EyeCane and UltraCane which both utilize the standard shape of the cane but with peripherals that allow the user to have additional features that a traditional cane would not have (i.e. warning sounds for streets or steep steps). Other technologies such as BlindSquare and AIRA utilize phone applications to receive assistance through the aid of others. In comparison to the aforementioned examples, the DPHS seeks to give autonomy to the user, give minimal audio feedback, and allow for a hands-free experience. The combination of these three overall goals for the project makes the device unique from other devices on the market.

2.0 Project Narrative

The goal of this project is to develop a wearable guide system to assist the blind or visually impaired in navigating the world around them. The idea is to potentially remove the use of a cane or other similar tool while still maintaining awareness of the surroundings. While other visual impairment products exist on the market; the project differs by concept due to its distinct input features including LiDAR in conjunction with computer vision. Output features such as an external speaker aid and haptic feedback make it a more comprehensive solution to its competitors. Competition in the same market includes experimental vests and cane modifications that the average user would likely find difficult; or even dangerous to use. This design is meant to be more accessible, simple, and effective than what is on the current market.

2.1 Project Goal

The purpose for developing such a device stems from seeking the possibility of a blind individual sustaining an independent daily life in an uncontrolled setting. Current tools such as canes and innovations like the 'Live Braille' *(Digit, 2016)* are helpful, however they are not always convenient nor effective at allowing a person to be in control of their environment; thereby increasing the chances of endangerment and alienation in respect to the individual. One of the goals of this project is to allow a person to be aware of stepping off of and onto curbs or other similar elevated platforms. More advanced ideas pertain to potentially being able to navigate uneven areas such as flights of stairs. Further discussion on competitor tools in section 4.1.

2.2 Proposed Method

There are three possible methods of assembling the project with a similar fundamental design. The input system consists of simple communication between a camera and distance sensor that are programmed to identify obstacles and alert the user through haptic feedback (output system) in the form of a grid or arrangement of vibration motors and an external speaker. The three possible arrangements depicted in *Section 2.4 Concept Sketches* include the system fitted into a jacket design, lightweight vest, and/or sash.

2.3 Computer Vision Methods

Computer vision will be an integral portion of the main function of the system. The idea will be following standard image processing techniques.

2.3.1 Image Differencing Change Detection

Image differencing allows for the detection of change between two image frames, creating a remote sensing index. The frames are compared against one another, to identify the area of change, through the difference of identified pixels *(ESRI, 2021)*. One proposed method that could be applied to the Depth Perception Haptic System (DPHS) includes having the camera monitor rapidly take images and process them through the image differencing raster function. Then the processing device would identify approaching obstacles through comparative image recognition with the obstacle database already predetermined. A potential constraint with this

method involves it normally being used in applications suited for straight-forward recording or stationary camera applications. In the event should the camera physically revolve/turn between the consecutive image frames, which is highly likely with a camera fitted to a person's body, all the pixels would differentiate, which negates identifying a particular obstacle.

2.3.2 Image Gradient

Image gradients, as one of the most basic forms of computer vision and image processing, primarily handles edge detection through the changes in pixel intensity within an image, allowing an edge map to be extracted (Rosebrock, 2021). This method is most useful for still images and might not be an extensive enough method on its own to be used with the depth perception haptic system.

2.3.3 Optical Flow

Optical flow (i.e. image differencing) examines the differences between two consecutive image frames, however instead it focuses on comparing against the change in the pixels. The method primarily follows the motion of objects and estimating their movements from the resulting flow vectors. This could be very helpful to the depth perception haptic system's detection of moving objects. Unlike image gradients, this method may not be comprehensive enough to cover the complex real time dynamic image processing required on its own. The use of this method has also been noted to have additional constraints of being computationally expensive and slow *(Nanonets, 2019)*.

2.3.4 Convolutional Neural Network

The Convolutional Neural Network (CNN) is potentially the most comprehensive computer vision algorithm as it applies deep learning artificial intelligence in several layers with communication between nodes to accurately aid in image classification. With this algorithm applying machine learning and being known as a fore-runner in computer vision, there are already extensively used applications ranging from healthcare to marketing *(IBM, 2020)*. CNN appears to have the intended applications required to dynamically identify obstacles in images captured by the camera fitted for the DPHS.

2.4 Constraints

There were several potential constraints introduced that would impede the completion of the functional device or inability to work as intended. The constraints include a possible limit of processing power in the control system, weather conditions affecting imaging of camera, arm movements caught in the computer vision resulting in redundant feedback, haptic feedback not having the intended effect on user, and a number of other possible issues later listed within the document.

2.4.1 Economic Constraints

Constraints that may arise economically pertain to the affordability and social acceptance the device would possess:

- The event that the device is more expensive than originally thought due to certain peripheral and component prices in comparison to the intended budget.
- Ensuring it conforms to acceptable inconspicuous wear and does not leave misleading connotations that would reflect poorly on the device, company, or user.

2.4.2 Environmental Constraints

With the unpredictability of weather conditions ranging from rain to extremely cold climates, the device is expected to observe many potential risks that must later be accounted for by using contingencies. A few of the expected potential conditions and risks are listed below:

- The event camera lens or LiDAR peripherals stop working as intended due to conditions related to rainfall or snow
- Decrease in durability of device in the event user error occurs causing any of the components to be damaged by impact of a fall or being crushed by obstacle
- Freezing temperatures causing a lack of motor responsiveness

2.4.3 Ethical Constraints

Ethical responsibilities are self imposed constraints. The following are conditions imposed by the current understanding and beliefs of the development team.

- The product can be trusted in certain conditions but users must always be reminded that a cane should be used in conjunction with the device when in any dangerous environments.
- Understanding that no amount of testing is enough and that the device can always be improved in some way in regards to both the hardware and software components.

2.4.4 Health and Safety Constraints

The DPHS holds the same health and safety concerns as any other electrical device when involving a battery and some wire. The following are imposed constraints that will be taken to give the user comfort and prevent any possible injury.

- Ensuring that all exposed circuitry and wires are able to be safely housed within the fabric and all connections are properly covered to prevent moisture from entering the device.
- Test the device in varying levels of the moisture environments including fog and rain to ensure no static may occur.
- Prevention of injury from haptic feedback so that the vibration does not cause pain or leave any marks from long periods of use.

2.4.5 Time Constraints

The time constraints involve the amount of time that can be put into developing the product with limited supplies as well as testing the device within the term of two semesters of college. The following include concerns over these constraints:

- Lack of time to build a library for machine learning including the image analysis of shapes and surfaces.
- Decrease in chip availability due to the chip shortage started in 2020 results in concerns over part availability resulting in having issues in obtaining expected PCB delivery

2.4.6 Testing Constraints

Testing constraints include the limited resources by college students in creating controlled conditions that would test the limits of the device. A large amount of information is expected to be found through specification documentation of certain components.

- Concern of limitation on battery due to requiring the purchase of a premade battery instead of a custom use battery prefered for the device.
- Lack of available participants who have the desired disability (i.e. blindness or visually impaired).
- Lack of equipment for proper measurement of distance and time reaction throughout testing.

The group will reevaluate these constraints throughout the duration of Senior Design.

3.0 Project Description

3.1 Goals and Objectives

The primary objective of DPHS is to create a tactile-visual sensory substitution system. The device would allow individuals that are blind, low vision, or vision obstructed to navigate through obstacles they may encounter in their daily lives. The list below contains the goals divided into categories involved in the design plan: Core, advanced, and stretch. The core goals are the foundation of the project which will be completed by the end of the second term. Advanced goals are additional features beyond the basic system requirements, but should be considered during the course of the design process. Stretch goals include the features with the least priority only to be implemented if time permits.

The device goals include:

- Core
    - Transmit information about the distances of large objects in the environment to the user via haptic feedback (e.g., walls, lampposts, people, vehicles)
    - Have enough battery to last the length of a typical workday

    - Hands-free and not restrict the range of motion concerning the wearer
    - Have a blind-accessible user interface such as braille
    - Be responsive, having a quick response time to changes in the environment
    - Have intuitively meaningful haptic feedback impulse patterns
    - Be a useful, non-trivial product for the target audience
- Advanced
    - Transmit information about the presence and orientation of sharp changes of elevation in front of the user (e.g., curbs, steps)
    - Be capable of transmitting information about the system via a speaker (e.g., the battery level, mode of operation [refer to 'Stretch' goals])
    - Have a low power mode to extend battery life
    - Have an easily extensible software framework for development
- Stretch
    - Be capable of detecting crosswalks and relaying this information to the user via the speaker
    - Implement multiple modes of functioning (i.e., a City mode with pedestrian symbol detection, and a General mode)

As the aforementioned goals are applied some advanced goals may become stretch goals and vice-versa due to the constraints imposed on the project (refer to section 2.4).

3.2 Requirement Specifications

The DPHS is an assistive device and as such should be easily maintainable, accessible, and provide for its intended use within its demographic. The requirements presented in. Although only two modes are named, the system should be designed with the intention to support additional modes easily, such as a low power mode. Therefore the primary goals are ensuring that the system is fit for prolonged use, with a long lasting battery, and multiple modes suited for different environments. The device should also be designed for intuitive usage, to allow for as-close-to-seamless transition between using other solutions to this device.

No.	Statement	Description	Value	Unit
1.X	Physical Attributes			
1.1	Lightweight	Total weight, including the clothing which houses the electronic devices.	< 10	lbs
1.2	Compact	Thickness of the device at any given section.	< 2	in
1.3	Durable	Remains intact and continues to function after being dropped onto concrete flooring from specified height.	≥ 3	ft
2.X	Usage and Maintenance			

No.	Statement	Description	Value	Unit
2.1	Quick to arm and activate	Sighted user time to set up and turn on the device.	< 60	s
2.2	Full Range of Motion (ROM)	ROM for shoulder joints is not significantly hindered.	< 10	% diff.
2.3	Long-lasting battery	Time from full charge to depletion at 80% of maximum haptic feedback output with all input devices active.	> 4	hrs
2.4	Rechargeable	Via a standard US 120 V electrical socket.	1	ct
3.X	User Interface			
3.1	Large buttons	Surface area of each button is large enough to be easy to locate, and to contain braille.	> 0.7	in^2
3.2	Simple operation	A small array of buttons is all that is needed to operate the device.	< 10	ct
3.3	Mode indications	Unique output pattern (audial and/or tactile) on every mode change.	3	ct
3.4	Quick to learn	Time to learn the operation of General Mode and avoid collisions.	< 10	min
4.X	General Navigation Mode			
4.1	Haptic feedback that encodes distance	Provides a relative indication of distance (via feedback intensity) of arbitrarily shaped objects visible to the camera over a discrete range of intensities (including zero).	≥ 3	ct
4.2	Haptic feedback that encodes location	Provides a relative indication of location (via location of feedback) of arbitrarily shaped objects visible to the camera at a number of feedback units on the device.	≥ 6	ct
4.3	Restricted range	No response to arbitrarily large objects that are far from the camera.	< 20	ft
4.4	Quick response time	Time the device takes to react to an arbitrarily large object at fixed distance away from the user within device range.	< 900	ms

No.	Statement	Description	Value	Unit
4.5	Alert to proximal objects	Objects near the device trigger an unambiguous alert.	< 3	ft
5.X	Outdoor Navigation Mode			
5.1	Curb detection	Curbs (rapid inclines/declines in terrain) that appear in front of the user triggers an alert.	5 ± 1	ft
5.2	Curb description	Curb alerts indicate the (1) steepness and (2) direction (incline/decline) of the curb.	True	

Table 3.2 Requirement Specifications

3.3 Hardware and Software Block Diagram

The general designs of the software and hardware have been divided into a series of blocks. A user interface will set the mode of the device which will tell the control system what mode will be active. For example a low power mode, or a general navigation mode. The core system will function as a translator which translates input responses into haptic or speaker output.

3.3.1 Hardware Topology

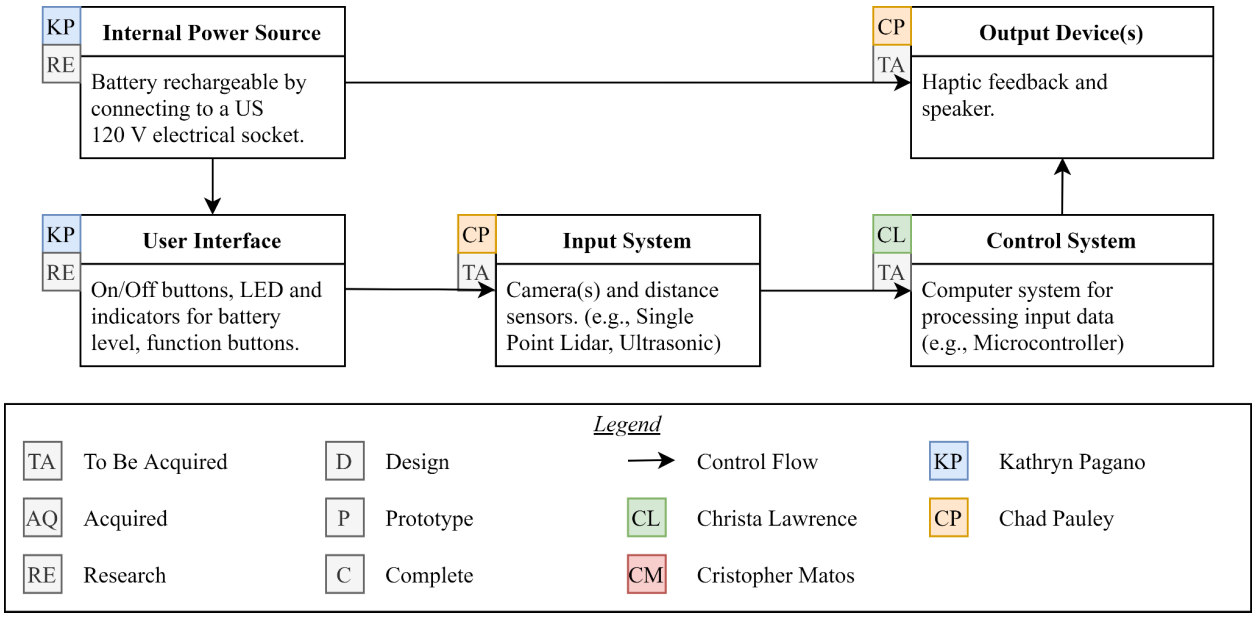


Figure 3.3.1 Hardware Block Diagram

Hardware implementation begins with finding the expected wattage of the system at idle and maximum power consumption, then finding a corresponding power source. The user interface will be the unit before the input system and emergency system as the power button that will be implemented in the design will also act as a kill switch should a bug or error occur that a reboot may fix. The input system will be responsible for various sensors that will be split between the control system and emergency subsystem. In terms of the emergency subsystem, the input system will use the LiDAR in the design to protect the user in case of an emergency in case the camera does not detect an object accurately. Should a fault occur with the camera that the LiDAR detects, an emergency beacon will be sent to a simple output speaker. The input system also provides the control data that will be sent to the control system through pins utilizing SPI, PWM, and I2C.

The control system will be based around a pre-existing model to enable product testing, act as a testing backup, and be a last resort should the recent chip shortage make finding specific parts from manufacturers difficult to acquire (refer to *section 3.5.1* for more details). Connecting to the output devices will be both the control system and internal power system. To explain the reasoning for this connection, the design was planned to utilize MOSFET to allow the haptic feedback motors to display full power without draining the power that could be used by the microprocessor. Following the output of traditional embedded systems; there was an expectation of only having an output voltage of 5 volts per designated motor. As the design may have multiple haptic feedback motors running with varying levels of power, a MOSFET will be used as a probable solution and will be created for each motor.

3.3.2 Software Topology

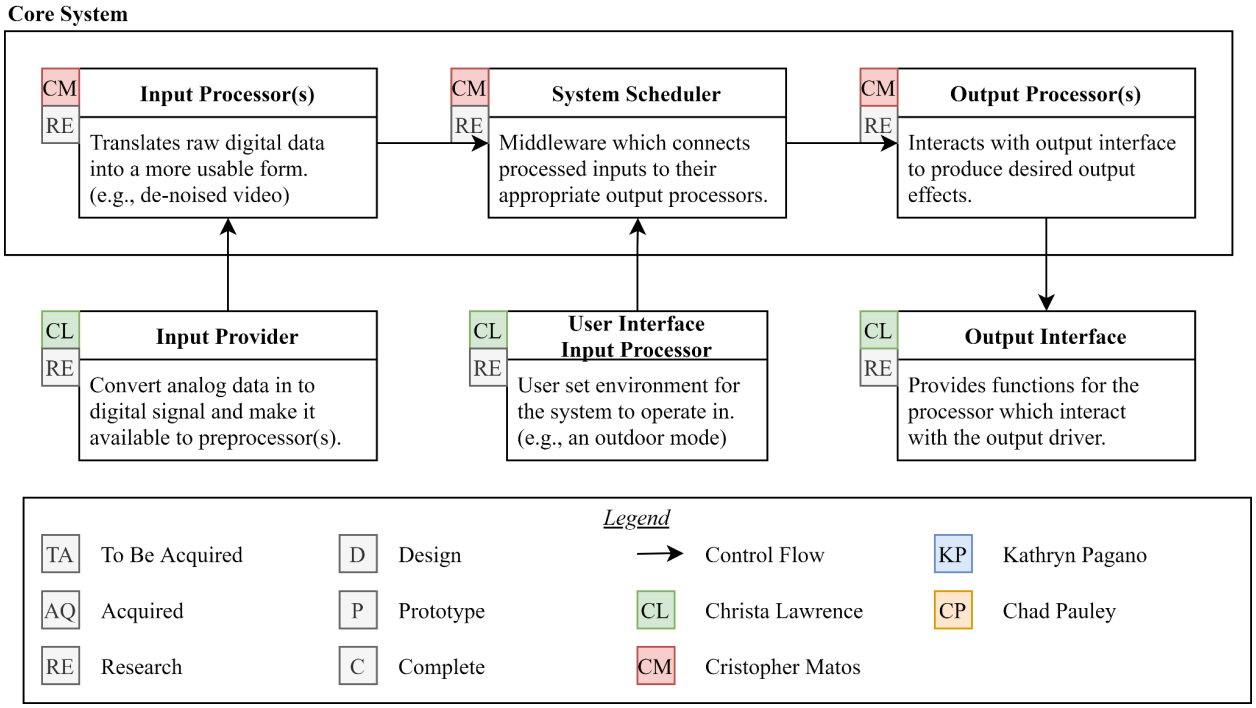


Figure 3.3.2 Software Block Diagram

The overarching design of the system software will resemble a translator. Output will be generated in response to (1) the state of the environment, and (2) the state of the device. This translation process is represented by the Core System (see *Figure 3.3.2*). Information about the environment will be provided to the system by a variety of sensors (e.g., camera, LiDAR, accelerometer). The input space must then be reduced by the input processor(s). These units are responsible for extracting high-level information from the input. For example, using a CNN for depth estimation and combining it with LiDAR to generate a point cloud, or curb detection as a result of instantaneous changes in the height of the floor.

The system scheduler component is responsible for scheduling the input translations. For example, the primary logic of a general navigation mode (see *Table 3.2, Requirement 4.X*) may be scheduled to be serviced after curb detection. It is also the point at which the input space is no longer reduced, but is made available for the output processor(s) to produce feedback patterns. These processors will use the input information and relay it to the output devices by consuming some output Application Programming Interface (API).

The output API will serve as an abstraction over the output devices. Output devices (e.g., array of haptic feedback motors, speaker) can be accessed by the Core System via this API. Operating beneath this layer are the high-level and low-level device drivers which implement the communication interfaces necessary to set the output. These drivers must be created for the array of haptic feedback motors, and the speaker.

Core software will likely be developed within a Windows 10 environment with C/C++. Software may ultimately run on a microcontroller which is supported by TensorFlow lite. When programming the device drivers; an OS such as Linux will be used to allow for ease of installation and testing. LiDAR, camera, accelerometer, and an estimate of one USB 2.0 input will be used to give functionality between the programmer and the device itself. For methods of communication between devices refer to *section 3.3.1*.

3.4 Concept Sketches

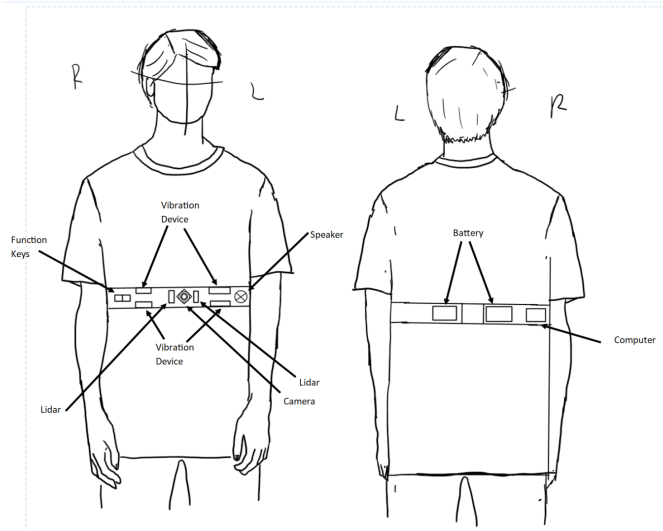


Figure 3.4.1 Belt Design: A sash worn across the center of the torso, similar to a heart rate monitor.

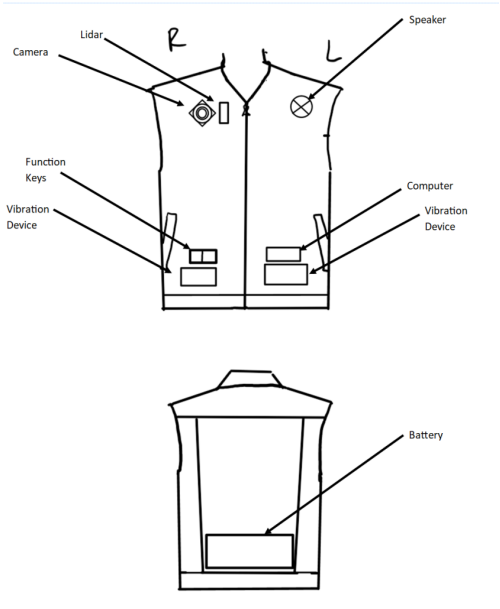


Figure 3.4.2 Vest Design: A vest, which is a common design choice for sensory-substitution devices.

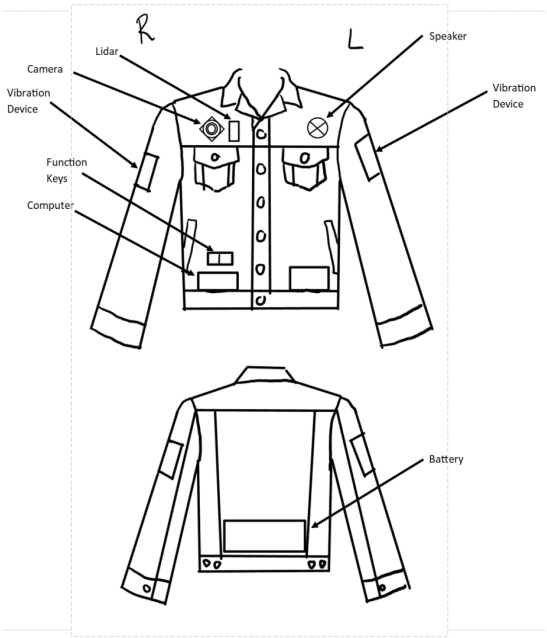


Figure 3.4.3 Jacket Design: A jacket, which could be used to help conceal sensors placed around the arms.

A wearable design will be implemented in the project through one of three designs focusing on comfort and reliability. Three designs were sketched as shown in *Figure 3.4.1-3.4.3* to allow for various options to be implemented in the placement of haptic feedback to the user, as well as the placement of input sensors such as the LiDAR and camera. The form factor of the device will revolve around the safety of the device and the comfort of the user.

3.5 Determined Budget

Item Type	**Quantity**	**Estimated Price (U.S. Dollars)**	**Description**
Fabric	≤ 3 Yds	≤ $15/Yd	Clothing material
LiDAR	≤ 2	$60	Object distance measurement
Power Source	Set acquirement (2)	$40	Device Power (Battery)
Accelerometer	1	≤ $10	Measurement tool for vibration, position, and acceleration
Camera	1	≤ $15	Input sensor for data manipulation and motion sensing
Speaker	1	≤ $10	Output device
Control System	1	≤ $50	An embedded system with I/O
Haptic feedback	Set acquirement (15)	≤ $20	Motors that provide vibration
Wiring	N/A	donate	Minimal gauge power connection
Custom PCB	≤ 2	≤ $30	Prototype and final product motherboard
3D Printed Objects	N/A	donate	3D printed buttons and applicable to clothing
Miscellaneous	N/A	≤ $100	Spare parts, prototyping
Total (Estimated Range)		≈ $390	

Table 3.5 Bill of Materials

Estimated project budget and financing.

The overall standing budget provided by the group is a self-funded $400 for the intended project. Prices are listed above as estimates from well known E-commerce companies including Amazon Inc and Digikey. Estimates displayed in the table do not include the cost of replacement(s) should some major component like the control system fail. Note that the table above includes 'set acquirements' whereby bundles of parts will be acquired at wholesale.

3.5.1 Verification of Control System

The final version of the product will involve the industrial soldering of embedded components for the microprocessor, peripherals, and output components. To ensure product success and proof of theoretical ability; an early PCB (Printed Circuit Board) will be developed solely for the verification in the working operation of the control system. This control system (namely an embedded system) will then be compared to a prebuilt version to ensure that the core components of the project works as intended. These tests may result in a larger overall cost but will serve as backups should the shipment or some error of manufacturing result in a hiatus to the PCB order.

Due to the recent chip shortage the project is expected to face difficulty in procuring all the expected parts, thus a decision may be required to expand the budget to buy pre-existing components that will be soldered onto the incomplete PCB after its delivery from the industrial soldering companies.

3.5.2 Verification of Sensors

Referring to *Table 3.5*; three sensors can be found: LiDAR, accelerometer, and a visual camera. These units will all be tested on the prebuilt version of the embedded system to compare performance to the early model PCB that will be implemented as noted in *section 3.5.1*. Quality tests will be used to verify the quality of the camera through bit rate, accurate distance measurements with a tape measure for the LiDAR, and the observation of the accelerometer in real-time with the use of a physical leveler.

3.6 Project Milestones

Number	Task	Start	End	Status	Responsible
Senior Design 1					
1	**Ideas**	8/23	8/27	Compete	Group
2	**Project Selection & Role Assignments**	8/27	9/17	Complete	Group
	Project Report				
3	Divide & Conquer 1.0	9/10	9/17	Complete	Group
4	Divide & Conquer 2.0	9/17	10/01	Complete	Group
5	First Draft (60 pg)	10/01	11/7	Complete	Group
6	Second Draft (100 pg)		11/19	Not Started	Group
7	Final Document		12/07	Not Started	Group
	Documentation, Research & Design				
8	Microcontroller	9/18	10/4	Research	Cris & Christa
9	Computer Vision	9/18	10/18	Research	Cris & Christa
10	Schematics	9/18	10/4	Research	Chad & Kathryn
11	Haptic Feedback System	9/18	10/18	Research	Chad & Kathryn
12	Accelerometer	9/25	10/25	Research	Chad & Kathryn
13	LIDAR	9/25	10/25	Research	Chad & Kathryn
14	PCB design	9/25	10/25	Design	Chad & Kathryn
15	Power Supply	9/25	10/25	Research	Chad & Kathryn
16	Order & Test Parts	11/01	11/07	Research	Group
Senior Design II					
17	Build Prototype	1/3	1/17	Not Started	Group
18	Testing & Redesign	TBA	TBA	Not Started	Group
19	Finalize Protype	TBA	TBA	Not Started	Group
20	Peer Presentation Final Presentation	TBA	TBA	Not Started	Group
21	Final Report	TBA	TBA	Not Started	Group
22	Final Presentation	TBA	TBA	Not Started	Group

Table 3.6 Project Milestones

Initial project milestone for both semesters.

As previously discussed in *section 3.5.1*; the software implementation of the project will be held back due to the PCB carrier module needing to be developed and tested. Sample code and testing of input instruments will be done after the shipment of the PCB (sometime during December). Building of prototype will occur once schematics and parts are gathered, however this date may be delayed in the case of difficulty in finding specific components during the current chip shortage.

3.7 House of Quality

Marketing Requirements	Engineering Requirements	Justification
1, 2, 3, 5, 7	Reliability	Functions working within a range of weather conditions (i.e. mist or sunshine). Ability to work indoors or outside.
2, 3, 4, 5, 7	Ease of Use	The overall size and shape of the device will affect the user in daily use.
1, 2, 4, 5, 6, 7	Accuracy	The ability to identify objects and distance with precise measurement to enable the user to feel distance through haptic feedback.
1, 2, 3, 4, 5, 7	Cost	As a major portion of the project relies on the embedded system, the cost of components will affect the selected base model and its performance.
1, 3, 6	Ease of Installation	The manufacturing perspective of duplicating the device.
Marketing Requirements 1. Power System 2. User Experience 3. Haptic Feedback Quality 4. Reliability 5. Sensor Resolution 6. Cost 7. Weight		

Table 3.7(a) House of Quality Relationships

House of Quality

	Relationships					
	Strong	●				
	Medium	○				
	Weak	▽				
		Engineering Requirements				
		Reliability	Ease of Use	Accuracy	Cost	Ease of Installation
Customer Requirements	Power System	●	●	▽	●	○
	User Experience and Interface	○	●	▽	○	
	Haptic Feedback system Quality	●	●		○	▽
	Reliability		○	●	●	
	Sensor Resolution	●	○	●	●	
	Cost	○	○	●		▽
	Weight	▽	○	▽	●	

Table 3.7(b) House of Quality

In concern to the designing of the device, the house of quality focuses heavily on reliability, accuracy, and cost. The customer requirements on the left hand side cover the important qualities of the project that need to be considered when designing. The Engineering Requirements cover the qualities that will help create a desirable product. This gives a good idea of the possible trade off that might have to be made based on the Engineering Requirements. Deciding factors for these selected values include the model of microprocessor the device is designed to use for programming overhead, and the quality of components such as the motors for haptic feedback and how much power the device will require. Analyzing each section of the house of quality figure, the cost benefit can be seen.

This first element for the customer elements is the Power System. The Power System will have a strong relationship with the device's reliability, cost and ease of use. In order for the device to act as it should, it needs to be incredibly reliable and with high durability for the user to achieve the full expected use. This type of device would be used for long periods of time and would need to

be supplied with enough power to last over that period of time consistently. In addition to this, the power supply will need to be mobile and easily accessible to the user. The user will need access to the battery in order to charge it, and this will need to be made as simple as possible. To have a power supply that holds a long charge and is able to recharge relatively quickly, as well as being small and mobile, will come at a higher cost.

The user experience and interface will need to take the ease of use as the highest priority. The user's ability to operate the device is one of the most important parts of this project. The user should also be able to operate the device easily as the demographic is visually impaired. This will be done using simple, distanced buttons, a speaker to interact with the user, and the motor to warn the user of possible obstacles.

Looking at the haptic feedback system quality; reliability is going to be the most important factor to consider. This system will be the main feedback to the user to warn of obstacles. It is also important that the system can easily communicate the area of these objects well enough that the user is able to avoid them. This will greatly affect the ease of use. The sensor resolution will also have an effect on all of these in addition to the importance of accuracy when reading input data. The sensors will need to accurately read the environment around them well enough to detect the upcoming obstacles.

The last two customer requirements are cost and weight. Cost will greatly affect the quality of sensors that will be obtained. Since this is a self funded project, some of the quality may have to be sacrificed for the cost, particularly when it comes to the more expensive parts.

Taking the aforementioned quality factors into consideration will be incredibly important going forward. Balancing time, quality and cost constraints are always difficult when not planned out well. It is a goal to balance budget and quality to not only make the device affordable to develop, but also be at a price point where the intended users can buy the reproduction.

4.0 Project Definition Research

4.1 Existing Similar Projects and Products

The goal for developing the DPHS device stems from seeking the possibility of a blind individual sustaining an independent daily life in an uncontrolled setting. Upon investigation, it was found that there exist a plethora of tools such as integrated canes and innovations that possess the same motivation. Innovations like the ‘Live Braille’ *(Digit, 2016)* device, cane attachments, and other various devices are helpful and prove to be competitors to the DPHS, however they are not as convenient nor effective at allowing a person to be in control of their environment as desired; thereby increasing the chances of the endangerment and alienation of the individual, socially, and physically.

The practice of using a cane is the most widely utilized method for the blind or visually impaired today. It has been proven to be a consistent and reliable as well as a cost effective tool that allows an individual freedom to move around somewhat freely. It has been noted however, that objects of the thinner variety are often missed due to the wide sweeping motion used with the cane (*‘See It My Way’, 2020*). Shortcomings such as this can be observed in many of the other competitors

as well. DPHS seeks to identify those discrepancies and provide an effective solution. The removal of a cane will also reduce injury caused by the jabbing effect into the gut from the cane itself; notably when quickly walking into an obstacle with force. A hands free design is a major proposition of the project. The following section includes details into other similar designs with similar means or with the same intended motivation.

A standard cane is a long stick or pole to be held in front of a person and swept across the preceding area until an object is hit, indicating an obstacle to be avoided. For the reasons aforementioned in the document including injury to self and others, the standard cane has seen several variations or alternatives to itself introduced into the market over the years.Standard canes include but are not limited to:

- Common white cane: for those who are totally blind
- Long cane: for road obstacles, street navigation
- The guide cane: short cane used for catching obstacles

Some of these variations add modifications to the cane's form, others opt for a less detectable/tangible cane body. These more advanced versions of a cane include The ICane, LiveBraille, and the UltraCane.

4.1.1 EyeCane ICane

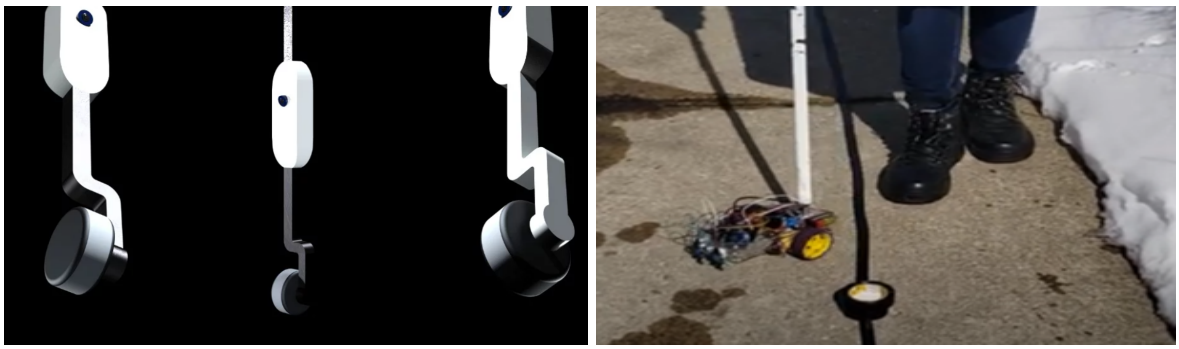


Figure 4.1.1 ICane and ICane prototype using thermoplastic paint lines (EyeCane, 2019)

Created in 2018 by CEO and Founder of EyeCane, Neil Mitra and his team, the ICane is a handheld robotic cane that emits infrared rays and automatically directs users. It features obstacle detection, medical sensors, and obstacle detection to help give blind people a little more independent mobility. The ICane relies on thermo-plastic paint tracks or GPS tracking, infrared sensors handle object detection and the cane navigates the user around obstacles via autonomous wheels fitted to the cane. The intent behind the idea stems from not having to implement tactile pavement, which can be very costly for cities to implement. The device is simple and easy to use, so therefore has a low learning curve, and is advertised to be easily accessible. The device is not costly for individuals to purchase compared to the prices of its competitors of similar technology; approximately $87 compared to upwards of $300 for other devices. Compared to the DPHS it uses somewhat of a similar technology when it comes to object detection. (EyeCane, 2021) As the device is still a cane, it requires hand use with the added implementation of wheels that are required to remain on solid ground. Thus in respect to an average cane, the device is heavier than

the average cane, requires the unit to stay grounded, and has a detection range closer to ground level.

4.1.2 LiveBraille

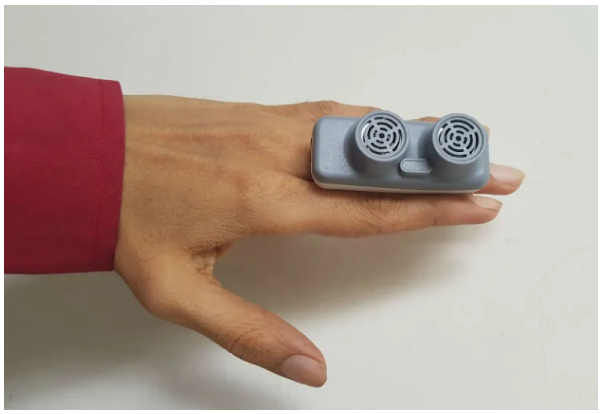


Figure 4.1.2 Live Brille Mini (Digit, 2016)

Created by mechanical engineer and entrepreneur, Abhinav Verma, Live Braille is a device fitted over an individual's finger that can be pointed and used in a similar fashion to a cane. Using ultrasonic sonar, when a person sweeps their finger in front of them a detected object and its distance is processed and a haptic pulse is returned to the user to inform them. The Live Braille device is compact and very lightweight and is stated to have a battery life of up to 4 hours. Again, compared to the DPHS the idea of sensors and haptic feedback is present, With this device the use of hands is still required. and due to the library of different haptic pulses; the device has a higher learning curve since an individual would have to memorize all of the 117 vibration patterns. The device is also very costly running up a price upwards of $380, coming up at $1000 at some retailers (*Digit, 2016*).

4.1.3 UltraCane

Since its start up in 2010, Ultracane is known for helping the visually impaired and blind with their cane device: the UltraCane. It is a cane that implements ultrasonic waves from two sensors from the middle of the cane shaft, one angled down to catch ground level obstacles within a range of 2 to 4 meters and another angled at a slight tilt upwards to catch obstacles at the chest and head level within a range of 1.5 meters. Tactile feedback is provided via vibrations through two buttons on the device's handle at varying frequencies depending on distance. This allows the user to create a spatial mental map. Compared to the DPHS, the Ultracane implements similar schemes in terms of logic, such as dual sensors angled one towards the ground and one angled forward or up to determine objects and distance. The device uses haptics to help users create a spatial mental map, however like the other devices mentioned in prior sections the device is not hands free and costs runs upwards of £500 or $800. (*UltraCane, 2011*)

4.1.4 BeMyEyes

Additional competitors include those who target more, solely technological/ digital approaches to assisting the blind or visually impaired in the form of specialized apps. These include, but are not limited to, BeMyEyes, BlindSquare, and AIRA.

Created in 2015 by Hans Jørgen, a Danish furniture craftsman who is also visually impaired he is the founder of BeMyEyes. It is a free app available for IOS and Android download that allows visually impaired and blind individuals to be connected to one of the current 5 million sighted volunteers via live video calls to help with everyday tasks including: reading a street sign or label on a box of cookies. This app shares some of the same ideals as the DPHS device as in directing and assisting blind individuals, and it implements an added benefit of human connection and community. In terms of independent mobility, this implementation still requires a dependence on others, whereas DPHS is sought to provide individuals with a higher level of autonomy (*Specialized Help, 2015*).

4.1.5 BlindSquare

Released in 2012, BlindSquare is a GPS enabled app available for download on IOS that draws information about a user's surroundings from third-party navigation apps including Foursquare and OpenStreetMap. It features many effective and helpful functions such as providing a user with their current address when the phone is shaken. There is also the feature of painting a picture of the intersection or venues around you or periodically alerting you of your progress to one's destination.The app is powered via Acapela voices (audio feedback) in the 25 languages they provide. The app shares similar ideals with DPHS, being to help visually impaired individuals such as the use of some audio feedback. The DPHS offers more autonomy for a blind individual than BeMyEyes, however the app does cost $39.99. Unfortunately, the app does not help the fundamental issue of avoiding obstacles in an individual's immediate path (Perkins Learning, 2016).

4.1.6 AIRA

AIRA is a subscription service by the Low Vision Specialists of Maryland and Virginia that pairs with smart glasses (e.g. Google Glass), or a smartphone camera. A certified Aira agent can be contacted and they are then able to see a blind or visually imairded individual's surroundings through the camera lense and direct and assist them according to the user's needs. The subscriptions range upwards from $89/month, so while it is a very effective system, it relies on human assistance more than autonomy. In addition, it's intent is not to replace canes or other assistive devices, so it isn't a comprehensively all encompassing solution (*Low Vision Specialists of Maryland, 2020*).

4.1.8 HeadsUp:Optical Mobility Aid

A 2018 Senior Design project at the University of Central Florida by students: Duc-Quy Nguyen, Alex Radulescu,Austin Singh, and Hunter Tanchin is a head-mounted distance response device, worn in a similar fashion to a hat or a helmet. It is intended to be used preferably in conjunction with a cane to alert users of potential obstacles and obstructions at a higher level in height than

would be detected by a cane. The device implements laser triangulation to allow the device to detect obstacles a user may be approaching. It uses haptic and auditory feedback to alert the wearer. The device possesses a detection range of 0.5 meters- 2 meters and is stated to have a battery life of up to 24 hours. Similar to the DPHS, this device implements distance sensing, while it's laser triangulation as opposed to the LiDAR and video processing that was decided to implement prior. In addition, as a head mounted device, there is a level of conspicuousness that could be uncomfortable to an individual, and is limited to mostly detecting objects in the distance or at the head level, meaning if an individual isn't also currently using a cane they could trip or miss those objects closer to ground level. *(Nguyen et al, 2019)*

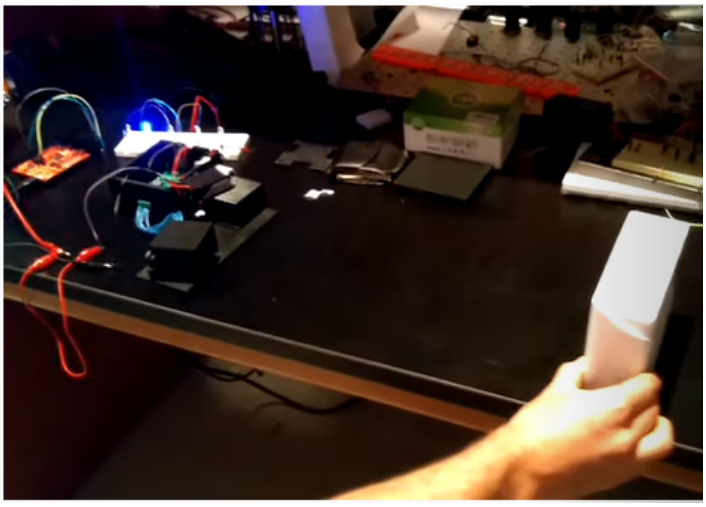


Figure 4.1.8 HeadsUp final demo (Duc-Quy Nguyen, 2019)

4.1.9 BackUp Buddy

BackUp Buddy is a 2018 Senior Design project at the University of Central Florida by students: Coleman Rogers, Luca Silvester, Zachary Slakoff , and Dylan Ortiz. While not necessarily used for blind aid, it is a camera device that is mounted on the back of a vehicle and operates as a rearview camera to indicate obstacles while reversing the vehicle via live video to a user's smartphone. If the distance between the vehicle and an obstacle in the video becomes too small there is an audible warning. This similarity to the DPHS is the concept of using video processing to detect obstacles, however the video playback function is redundant for the purpose of DPHS. (*Luca, 2018*)

As described, it can be observed that most devices for the visually impaired or blind replicate a variation of the cane, it is sought to provide a device that can operate as a cane without some of the limitations that comes with using them.

4.2 Relevant Technologies and Specifications

The system must be able to take in information about the depth of objects in the field of view of the user and translate that information into an output that is perceivable. The focus of the project involves what depth can be sensed and how to make this information be made available to the user. In this section, technologies relevant to the focus will be presented in the following section.

4.2.1 Depth Sensing Options

The depth sensing technologies that will be investigated for use in this system are ultrasonic sound based, light based, or computer vision based. These technologies are common, well-documented, and have cost effective options available.

The HC-SR04 ultrasonic distance sensor is popular for its low cost,ease of use, and wide availability. The dimensions pertain 45 x 20 x 15 mm with a minimum range of 2 cm and a maximum range of 4 m (*Marian, 2020*). It performs best on objects with a minimum surface area of 0.5 m^2 before results are negatively affected in the ultrasonic sensor. Distance measurements may also be affected by materials such as carpet which absorb sound.

The TFMini-S Micro LiDAR module is on the higher end of expense than the HC-SR04. The dimensions are 42 x 15 x 16 mm and has a minimum range of 10 cm and a maximum range of 12 m. Within 6 m the measuring error is within ± 6 cm, and beyond 6 m is a 1% margin of error according to schematics. Distance measurements may be affected by transparent objects or objects which absorb a significant amount of light (*DroneBot, 2018*).

While traditional LiDAR and ultrasonic sensors provide distance information on a single object in view of the module, more advanced solutions exist which generate point clouds. These point clouds provide distance information on groups of pixels in an image (i.e. Intel's LiDAR Depth Camera L515). Although significantly more expensive than the TFMini, the LiDAR produces much more output data. It is 61 mm in diameter and 26 mm high. The output produces 1024 x 768 depth resolution at a range of 25 cm minimum, and 9 m maximum. The device also provides RGB video with 1080p resolution at 30 frames per second. It is noted that this particular device is intended for indoor use (*LiDAR Depth, 2020*).

4.2.2 Depth Estimation with Computer Vision

Although solutions such as the Depth Camera have useful characteristics, options available on the market are: expensive, draw large amounts of current, and function at a resolution beyond which is necessary for the system. A low-resolution equivalent to this device is desired. Several techniques exist to estimate depth using computer vision. Binocular depth estimation relies on the presence of two cameras and geometric calculations to determine the distance of objects in the screen. More advanced methods blend this approach with deep learning. Monocular depth estimation proceeds a step further and attempts to use a single camera to attain similar information (*Tan, 2020*).

Some of the most important cues for visual estimations of depth are motion parallax and binocular disparity. It has been shown that monocular estimations of depth (algorithms that rely on motion parallax and the camera motion of a single camera) produces more accurate estimations of depth beyond a range of 6.5 m. Below a 5.5 m distance; stereo vision performs marginally better (*Mansour, et al., 2019*). Focusing on monocular depth estimation, consideration will be given to whether or not individual images are enough for depth estimation, rather than a collection of several images taken at different points in time. Using video input provides smoother and sharper depth tracking over time than images, however images are still able to produce depth maps at a lower consistency and level of accuracy (*Kopf, et al., 2021*).

4.2.3 Microprocessor and Controller Options

Considerations into the control system will influence the choice of the microcontroller or processor to be placed within the system. Considering the potential for using computer vision; only units which are capable of executing neural networks will be considered. Three such units are NVIDIA's Jetson Nano, the Raspberry Pi Pico, and the Raspberry Pi 4.

The Jetson Nano is a device developed specifically for use in embedded AI development. It is capable of running multiple neural networks in parallel. The development board contains 4 GB of RAM, a quad-core CPU, and a 128-core GPU. The Raspberry Pi 4 is a general purpose microprocessor that has also been utilized for AI. Many variations exist but the core features include a quad-core CPU, high quality video decoding, and a wide selection of RAM. Finally the Raspberry Pi Pico is a board which contains the RP2040 microcontroller unit. It is a low cost chip which contains a dual-core processor, 264 KB RAM, multiple communication units, and is low power (Raspberry Pi Foundation, 2019)..

In terms of performance, the Pico has successfully been used for machine learning applications, but it is generally quite slow at 2200 ms according to detection benchmark (Upton, 2021). It is exponentially slower than its competitors for fully-connected networks (Halfacree, 2021). When benchmarked using MobileNet—a lightweight architecture for CNNs the Raspberry Pi 4 model B and the Jetson Nano both performed at comparable speeds (Howard, et al., 2017). Using TensorFlow a software framework for machine learning (TensorFlow) the Jetson Nano was able to infer the name of an object in an image using MobileNet v1 in 276.0 ms. Similarly the Pi using the same software took 263.9 ms. Using different, more optimized variants of TensorFlow, the Nano and the Pi were able to infer in as little as 61.6 and 82.7 ms, respectively. It is important to note that these tests used fans on the Pi to prevent CPU throttling due to potential heat generation caused by running image inferencing over an extended period of time (Allan, 2019).

4.2.4 Architectures for Depth Estimation

In the process of investigating technologies, it was discovered that the Computer Vision and Pattern Recognition Conference (*CVPR, 2021*) hosted a Monocular Depth Estimation Challenge targeting the Raspberry Pi 4 using TensorFlow Lite (*Plowman, 2021*). The purpose of the competition was to design a CNN model for monocular depth estimation that is lightweight enough to run on resource-limited devices such as the Pi 4. Of the teams that entered, Tencent-GY Lab won the competition by producing a small 3.4 MB model with a 97 ms runtime and high-fidelity results with a root-mean-square error (RMSE) of 3.56. The highest fidelity results were achieved by team HIT-AIIA with a near 56.0 MB model size, a 6146 ms runtime,

and an RMSE of 2.72 (*Ignatov, 2021*). The winning team used the PyTorch library on top of TensorFlow lite to implement their design (*Zhang, 2021*).

4.2.5 Tactile Output Technologies and Techniques

The previous sections have focused on the core components of the system's functionality—considering the question of how depth will be sensed. Now the question of how this information will be transmitted to the user will be considered. User interface components such as buttons and speakers will be discussed in the following section as these components do not affect the system's core functionality.
The system will function as a substitute to sight for navigation using tactile feedback. It must be able to transmit information about the depth of objects in the scene to the user using their sense of touch to prevent them from running into any obstacles. Two major approaches exist for this feedback: Mechanical and haptic. Both approaches map information from different segments of a depth map to an array of points located somewhere on the user's body. Below both approaches will be briefly discussed. See Section 4.4 for a detailed description of the design considerations for the final output device.

During early technology investigations a device was found which uses mechanical pins to press on different points of the body with different amounts of pressure to indicate depth (*Stuff Made Here, 2020*). The approach is novel in that it intends to reproduce a depth map as an image on the skin. In this approach there are discrete levels of pressure where output signals are generated at the moment a threshold is crossed. Both of these features combined allow the user to not only directly interact with the map, but also minimizes the number of output signals available to the user. This allows for an interactive experience with minimal noise. Haptics on the other hand intend to create an immersive experience for users. Whereas with a mechanical approach the user must actively interpret the output image, haptics are intended to be a passive means of recreating embodiment. Haptics are significantly more cost effective, require less design considerations, and are far simpler to program. As such, vibrating motors will be used for the system tactile output approach.

4.3 Part Selection

4.3.1 Introduction

This section will compare the technologies presented in section 4.2 against criteria based on the requirement specifications set forth in Section 3.2. This will begin with identifying the choice of sensors, followed by the computer, the camera, miscellaneous sensors, electronics and UI device selections. One of the most important criteria considered for this section is the cost. This project will be self-funded by the group, making the development under a very limited budget. The following section lists the most cost-effective parts that balance well with the quality needed to match the requirement specifications.

4.3.2 Computer selection criteria

A computer that is both capable of applying a CNN, and able to produce a result significantly faster than 900 ms to allow for processing the input data and handling all concurrent processes is

required (Table 3.2.1, Requirement 4.4). Given the nature of the selection of depth sensing technology, this computer must be capable of running a dedicated operating system (OS) where a machine learning software library such as PyTorch is available. The computer must be designed for use in a deeply embedded environment. Sufficiently detailed datasheets must exist in order for us to be able to design a base-board on which the computer must be able to interface with. Communication hardware and GPIO lines must be available to interconnect all input and output devices. There should be sufficient storage to be able to hold any audio, OS, or models that the system may require. Additionally, there should be enough RAM such that the system can have quick access to all model and sensor data for the General Navigation mode, but also that the system may be easily extended to include additional neural networks for an outdoor mode. See Table 4.3.1 below for a quantified version of the selection criteria.

Key	**Value**	**Unit**	**Justification for Criterion**
MobileNet v1 Inference Time	< 500	ms	For a response time below 900 ms (Table 3.2.1, Requirement 4.4), must ensure that inference is completed as quickly as possible to allow for additional processing.
GPIO	> 8	Pins	A significant number of GPIO should be available for connecting output units if other communication units are unavailable.
HDMI	≥ 1	Port	To display the GUI of an operating system for programming.
USB	≥ 2	Port	To connect equipment such as a flash drive, keyboard, or mouse.
Memory	> 1	GB	CNN models reviewed in Section 4.2 generally fit in 10s of MBs of data. Extra memory for drivers, operating system services, and device software.
Storage	≥ 16	GB	On-board or MicroSD storage capability for storing inference model, application code, operating system, and configuration.
Height	< 1	in	To be easily embedded within some housing which is compact for end user comfort.
Length/Width	< 4	in	
Price	≤ 50	USD	As defined by the Budget in Section 2.5.

Table 4.3.2(a) Computer Selection Criteria

MCU	Raspberry Pi 3	Raspberry Pi 4	Jetson Nano
RAM	1 GB	1 GB	4 GB
Maximum Clock Frequency	1.2 GHz	1.5 GHz	1.43 GHz
Memory	External SD card	External SD card	16 GB
Communication Peripherals	2- I2C, 2-UART, 3-SPI	6-I2C s, 2-SPI, 2-UART	GPIO, I2C, I2S, SPI, UART
Timers	2x8 Bit or 4x16 Bit	2x16 Bit	
Temperature Range (degree)	-40-85	-40-85	
Operating Voltage Range	4.75V-5.25V	4V-5.5V	
Total Pin	40	40	40
Price (USD)	$35	$55	$109

Table 4.3.2(b) Computer Selection Criteria Computer Comparison Table

The computer that most closely matches the selection criteria is the Raspberry Pi 4 model B—in particular the Compute Module 4 (CM4). The compact form factor of the CM4, competitive price, and its proven ability to quickly run CNN inferences make it ideal for the proposed system. Additionally, datasheets and model files for the Raspberry Pi's CM4 IO Board are provided which will guide us in designing a custom base-board. The next step with respect to the microprocessor is to create a board that will expose the interfaces of the CM4 that will be used in the development and deployment of the product.

4.3.3 Sensor selection criteria

There are two types of sensors that were considered for depth sensing, ultrasonic sensor or Light Detection and Ranging sensor (LiDAR). The ultrasonic sensors which use a high frequency sound to determine distance are a cost-effective device. They tend to be good in foggy conditions but have issues with objects that are soft and that can absorb the sound wave. They also tend to have slow reaction times which would inhibit the detection of quick moving objects, and this is a major requirement for the project. The second sensor in consideration for depth sensing is the LiDAR sensor. Unlike the ultrasonic sensor the LiDAR sensor is able to detect fast moving objects and does not have issues detecting soft objects. The only set back to the LiDAR sensor is it tends to be more expensive and has a narrow field of detection. With those pros and cons of each the main parameter for the depth detection sensor is to detect objects coming towards the user suddenly. And it is for this reason the LiDAR sensor was chosen for this application

This project requires a LIDAR and an accelerometer as separate sensors outside of the camera which will be taking on a lot with object detection. The LiDAR for the project is meant to be a quick response system to detect immediate obstacles. Because of this a 2D LiDAR would be efficient for what is needed. The next step in the decision process would be determining range requirements. In this case, a long-range LiDAR would not be necessary for close-up obstacles that are to be detected. The final consideration was the cost. To keep the price of the system low the price of this part needed to be below $60. The table below shows two LiDAR that were considered for this project. Ideal specifications are shown below.

Operating Voltage	3V-5.5V
Operating Current	Small idle current (~2mA) and max of 90mA when in use
Range Detection	1cm-12m
Price	>$60

Table 4.3.3(a) Ideal LiDAR

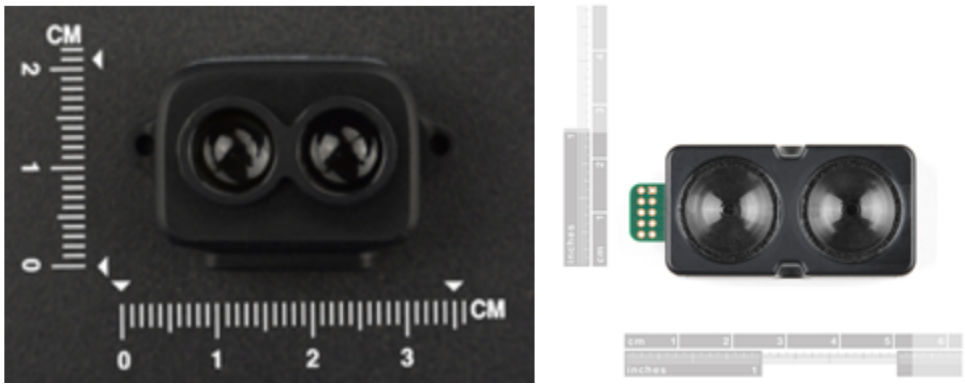


Figure 4.3.3(a) TF-Luna (left), Garmin LIDAR-Lite v4 (right)

The first device considered was the TF-Luna. This LiDAR is small as shown in figure 4.3.3 inexpensive and meets all but one of the requirements. Unfortunately, the range on this LiDAR is a bit shorter than what is preferred. The price is ideal at around $25 dollars and is compatible with the Raspberry Pi devices. Unfortunately, the reliability of being able to get this part might become an issue as many many locations are out of stock.

The next device taken into consideration is the Garmin LiDAR-Lite v4. As shown in the image above, it is a small device, around the same size as the TF-Luna. The range goes up to 10m, which is close to the ideal 12m range. The price is at the top end of the budget unfortunately, but

this LiDAR is more readily available. Many more websites have this LiDAR in stock. That may end up being the deciding factor for this device.

	Garmin LIDAR-Lite v4	**TF-Luna**
Operating Voltage	4.75V - 5.25V	3.7 - 5.2 V
Operating Current	2mA idle, 85mA during acquisition	70 mA
Resolution	1cm	1cm
Update Rate	200Hz	100Hz
Detection Range	5cm - 10m	0.2 m ~8 m
Board Size	52.2 x 21.2 x 24.0 mm	22.25mm x 35mm
Price	$60	$25

Table 4.3.3(b) LiDAR Comparison Table

When comparing the two, the price is the major difference between the two LiDAR's. Since both are within the budget for this device they are considered, but the TF-Luna would be preferred in this category. For all the other specifications needed, the Garmin LiDAR is preferred. It has a better detection range and a good operating voltage. This LiDAR device also has a good operating current that will go into an idle mode to use less current. This is ideal since it would have less power consumption when idle. The last factor taken into consideration is the ability to get the part. The Garmin LiDAR is in stock at more parts websites than the TF-Luna, making it more readily available.

4.3.4 Speaker Comparison

The speakers needed for this project are very basic. The only thing that it needs to do in the basic design is to tell the user if the device is powered on when initially. In the stretch design the speakers would give warning of the types of upcoming obstacles. The basic requirements for the speakers is a low resistance, wattage and size.

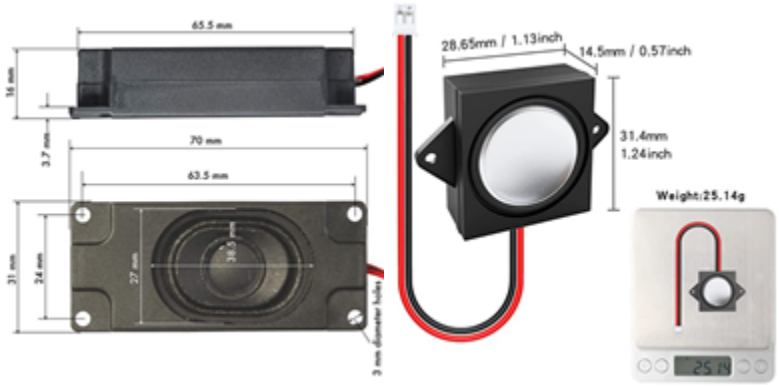


Figure 4.3.4 Degraw DIY Speaker Kit - PAM8403 (left), MakerHawk 2pcs Arduino Speaker (right)

Looking at the first option for a speaker is the Degraw DIY Speaker Kit - PAM8403. It is a simple speaker from amazon, so easily available. The size of this speaker is a bit large for the purposes of this project, but still within the realm of being usable.

	Degraw DIY Speaker Kit	MakerHawk Speaker
Operation Voltage	5 V	3.3 V or 5 V
Power	3 W	3 W
Resistance	4 Ohm	8 Ohm
Efficiency	90%	93%
Price	$11.99	$9.99

Table 4.3.4 Speaker Specifications

The second option for a speaker is the MakerHawk 2pcs Ar Duino Speaker. It is small and compact, which is ideal for this project. Also, a product easily found on amazon with no stock warnings. Even though this speaker has a higher resistance, the lower operating voltage and price make it a great option for this project.

Comparing the two devices, the best option for this project's purposes would be the MakerHawk 2pcs Ar duino Speaker. The main deciding factor is the size. The speaker needs to be as inconspicuous as possible, especially since, if unable to reach the stretch goals, the speaker will just let the user know the state of the device. The other specifications like voltage, resistance and price will also be adequate for the specified needs.

4.3.5 Motor Selection

The haptic feedback motors are incredibly important for this project as they are the main communication between the device and the user. A small, but powerful motor is needed to effectively communicate the area that an obstacle is coming up. The options for this device are from two main styles. A coin type and a cylindrical type. One of the goals of this project is to make the electric components as unnoticeable as possible, so that costs were taken heavily into consideration in the selection process.

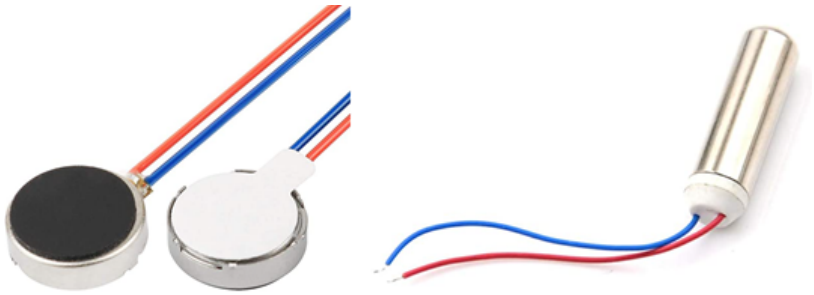


Figure 4.3.5 Vibration Motors

As shown in the figure above, the coin type motor from Tatoko is small and flat, making it very easy to hide in clothing. A small square of fabric can secure this in place much easier than the cylindrical option and it comes with a sticky back to help secure the motor in place. On the technical side, the specifications can be seen in the table below. The rpm should be high enough to be noticed by the user and they do not take too much power to operate.

	Tatoko Mini Vibration Motors Flat Coin Button	BestTong Miniature Vibrating Motor
Operating Voltage	3V	1.5 V- 3 V
RPM	12000	8000-24000
Size	10mm x 3mm	7mm x 25mm
Price/Motor	.75 cents	$2

Table 4.3.5 Motor Specifications

The other option is shown in the figure above, which is the cylindrical motor. This motor is also small and compact. I think when inserted into clothing, it will have a tendency to move around more and impede the user from feeling the vibration when an obstacle is near. The specifications can be seen below. As shown, these are significantly more expensive than the coin type.

Out of both motors, the coin motor was chosen for this project. The size and cost are both ideal and the ease of installation made it the best choice.

4.3.6 Camera Selection

For the purposes of this project, a high-quality camera will not be necessary. The main objective for the camera is being able to detect approximate depth information of the environment. Since a powerful camera is not necessary, the main considerations for this project are going to be cost, compatibility and availability. The ideal type of camera would have specifications similar to the ones listed in the table below.

Resolution	Min 640 × 480 pixels
Price	< $30
Size	< 50 mm

Table 4.3.6(a) Ideal Camera Specifications

Raspberry Pi Camera Module 2

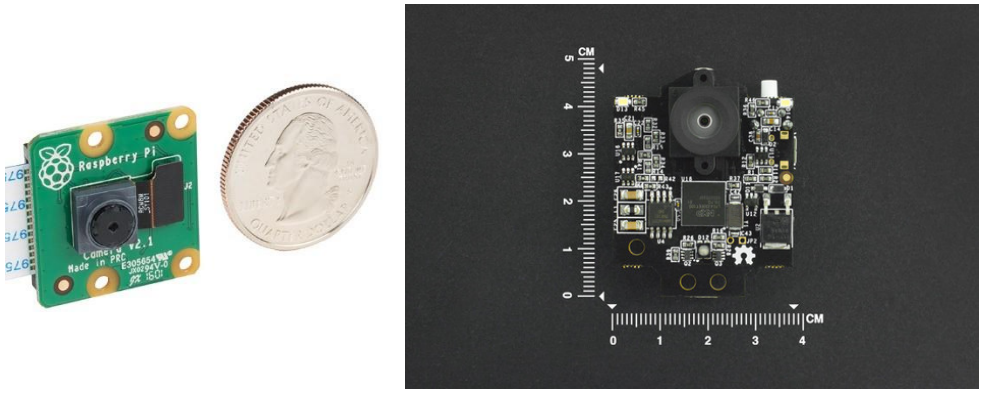


Figure 4.3.6 Raspberry Pi Camera Module 2 (left), Pixy 2 CMUcam5 Image Sensor (right)

The first option for a camera is the Raspberry Pi Camera Module 2. As shown above, the camera module is incredibly small and can easily be concealed. The resolution will be more than enough for the purposes of this project and is within the budget. This camera is compatible with the Raspberry Pi board and would just require a 15 to 22 pin adapter ribbon to work with the board, which is typically inexpensive and worth the cost. It uses the Camera Serial Interface Type 2 (CSI-2).

Resolution	8-megapixel 3280 x 2464 still frame
Video mode	1080p30, 720p60, 480p90
Price	$25.00
Board size	23.86 x 25 x 9mm
Onboard processor	No

Table 4.3.6(b) Raspberry Pi Camera Module 2

The next camera taken into consideration was the Pixy 2 CMUcam5 Image Sensor. This camera was considered because of the onboard processor. The Pixy 2 comes with some basic computer vision software installed. With multiple modes, it can track certain objects by using different methods of Computer vision. The first mode allows for objects to be tracked by color, using a color-based filtering algorithm, which can remember up to 7 different colors. Another mode is line tracking, where the camera can detect lines, intersections and things like bar codes. The last computer vision mode is the Pan-Tilt, which always has a mounted camera to rotate and pan to track objects. Lastly it can just run in a camera-only mode.
This camera was taken into consideration because of its capability to do basic computer vision methods. Unfortunately, the features are a bit light for the implementation of this project. The functions it natively supports are just line detection and simple object detection. The project is going to require different types of computer vision, and although these built-in features could be useful, they are not quite what is needed. The size and cost are also almost double of the other considered cameras.

Resolution	1296 x 976
Video mode	60 frames/sec
Price	$60.00
Board size	53x 50 x 35.6mm
Onboard processor	Yes

Table 4.3.6(c) Pixy 2 CMUcam5 Image Sensor

So comparing the two cameras, it was decided that the Raspberry Pi Camera Module. With the small size and cheaper price it makes it a great choice for a camera. It also doesn't have any unnecessary features that would add to the price. The board compatibility is also ideal.

	Raspberry Pi Camera Module	**Pixy 2 CMUcam5 Image Sensor**
Resolution	8-megapixel 3280 x 2464 still frame	1296 x 976
Video mode	1080p30, 720p60, 480p90	60 frames/sec
Price	$25.00	$60.00
Board size	23.86 x 25 x 9mm	53x 50 x 35.6mm
Onboard processor	No	Yes

Table 4.3.6(d) Camera Comparison

4.3.7 Accelerometer

This is another feature of the emergency response system. A simple accelerometer is needed to detect when an object is rapidly approaching the user. Ideally, it should use little power, having a low power mode that won't drain the batteries too quickly.

When choosing an accelerometer we are faced with the choice of technology shape size and sensing range. The first step in the selection process was to determine how many axis's we need for measurement in our project. It was decided that the tri-axle version would be the best for our implementation. The accelerometer weight and size would be the most important decision to make next. We needed something that was again small and light weight since it will be sewn into a jacket and the user will have it on them. This was a large consideration when looking at accelerometers. Finally we needed something that can interface with either I2C or SPI. At the time of part selection it was unclear which one would be needed so having both available would be ideal

Supply Voltage	1.5 V- 4 V
Interfaces	SPI, I2C
Current	< 8 uA in standby mode < ~150 uA When measuring
Resolution	12 bit - 16 bit
Cost	< $20

Table 4.3.7(a) Ideal Accelerometer Specifications

As seen in the table, the ideal type of accelerometer would have a voltage of 1.5V to 4V depending on if the device is in low power/ standby mode. It would be able to do I2C or SPI interfaces so that both options would be available for use. The resolution would be the next thing to consider, a 12 bit to 16 bit range would be the best since a high resolution is not completely necessary. Last consideration would be the cost of the product. This is an area that might be considered for cutting costs. Ideally, the accelerometer should be significantly under $20. Availability and Interface will be the deciding factor in this area. Below are the two accelerometers that were considered for the project. There is the contrast of cost and quality displayed.

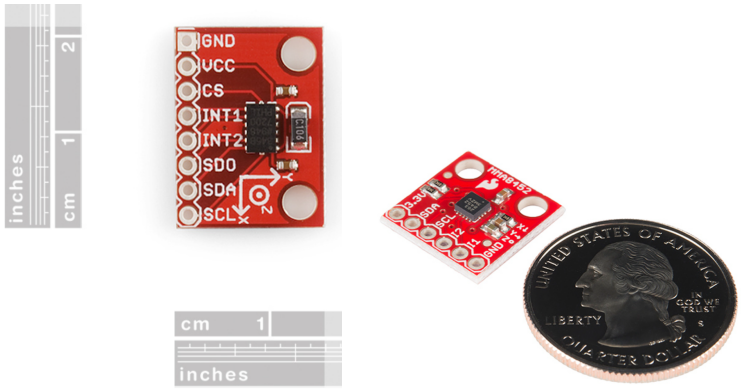


Figure 4.3.7 ADXL345 (left), MMA8452Q (right) Accelerometers

The ADXL345 is a high resolution (13bit) accelerometer that is well suited to measure dynamic acceleration due to motion or shock. It has two convenient modes, an active and inactive mode to reduce the amount of power used. This accelerometer is accessible through I2C or SPI digital interface.

The MMA8452Q accelerometer also has a low power mode and uses I2C interface for access. It can be configured to have an internal wake-up time and has embedded functions that are easily programmable. This accelerometer can also be configured using only two interrupt pins.

Device	**ADXL345**	**MMA8452Q**
Supply Voltage	2V - 3.6V	1.95V - 3.6V
Interfaces	SPI, I2C	I2C
Current	40uA in measurement mode, 0.1uA in standby@ 2.5V	6 µA to 165 µA
Resolution	13-16 bit	12 bit
Cost	$20	$10

Table 4.3.7(b) Accelerometer Comparison

Comparing the two accelerometers above, the ADXL345 should work adequately for the needed specifications of the project. This device has the low power consumption that matches the specifications needed for this project and is half the cost of the other product considered. It also has the two interfaces that would work with the project, giving more options when the interface needed is decided on.

4.4 System Design Considerations

The previous three sections investigated competitors, techniques for sensing and transmitting the sensed information to the user, and selected the parts that will be used for the product. In this section possible configurations for the hardware and special considerations for the software design will be provided.

Given that the device is intended to be an assistive device worn by the user over the span of several hours, user experience (UX) has been a key design consideration. In addition to UX, the complexity of the system design must be balanced with both the product constraints (Section 2.4) and goals (Section 3.1). Below a focus is placed on concepts such as embodiment, device size, output resolution, input argument reduction, output response complexity, response time, and response patterns.

4.4.1 Psychophysiological Considerations

Here attention will be given to the sensory-substitution experience of the user. One of the early design considerations has been embodiment. Embodiment refers to the concept that cognition is constrained, regulated, and shaped by the sensorimotor context in which it functions (F*oglia and Wilson, 2013*). In a fully-embodied approach, output units—collections of haptic motors—would be distributed over a wide area of the user's body. These units would be placed such that sections of the depth map would be mapped symmetrically over that area. This symmetry would provide an intuitive interaction which would likely reduce time-to-learn (TTL) by taking advantage of the body's natural tactile affordances (e.g., retracting the shoulder when bumping into a pole). A common alternative approach is image projection, which projects the depth map onto a smaller area of the user which is typically placed to one side of the body such as the hand (*Stuff Made Here, 2020*). This approach requires the user to consciously interpret the image, but is more compact since output units are placed over a smaller area (See Figure 4.4.1). This asymmetric approach also means that output units can be placed further away from the midline of the user, toward the hands. The sensitivity of the hands means that more information can be transmitted to the user over a smaller area. This is opposed to the torso, which is a poor interface for differentiating between stimuli that are applied near each other (*Stuff Made Here, 2020*).

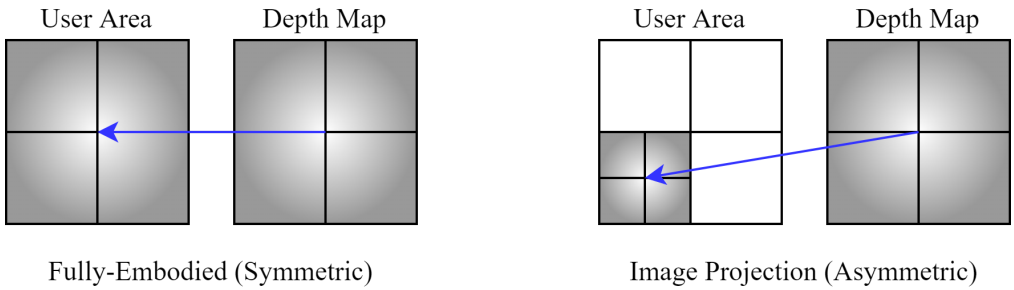


Figure 4.4.1 Symmetry in Output Unit Placement

4.4.2 Output Unit Hardware Architectures

It has briefly been mentioned that output units are bundles of haptic feedback motors. Here the spatial complexity of these devices will be considered. Table 4.4.2 demonstrates three options for creating each unit. The options assume that the devices will not implement a communication interface, and that their functioning is completely dependent on the CPU (i.e., they do not need to maintain their own state). These assumptions were chosen as an initial point from which to begin considering the design of these units, and may be adjusted as design discussions progress. The simplest option is the On/Off case. A single pin would be used to control the intensity of all N motors within the unit. This means that each unit could only transmit information along one dimension: Output intensity. The other options divide the N motors among portions of the unit, which are activated with the specified encoding. An extra +1 pin would be needed to control the vibration intensity of the enabled subsection.

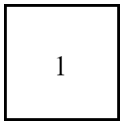
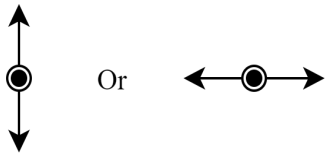
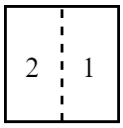
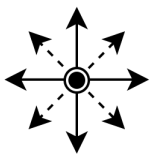
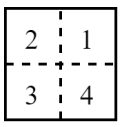
Unit Option	**Directions**	**Configuration**	**Encoding**	**Minimum Pins**
On/Off			[0, 1] - Intensity	1
Single Axis			00 - All Off 01 - (1) On 10 - (2) On 11 - All On	2 (+1)
Dual Axis			000 - All Off 001 - (1) On ... 100 - (4) On 101 - (1) and (3) On 110 - (2) and (4) On 111 - All On	3 (+1)

Table 4.4.2 Output Unit Configurations

4.4.3 Output Unit Signal Design

At a minimum, output units regardless of their configuration will need to respond by vibrating at a certain intensity according to the magnitude of a signal. This wired communication is unidirectional from the CPU to the unit. Techniques for handling collisions which occur when multiple software services compete over output units, and possible configurations of these output signals with respect to UX and information transfer are considered below.

Simple Signal Compositions

Multiple processes within the CPU may complete to send an intensity signal to an output unit. Techniques for handling such collisions will be referred to as composition, and division. Signal composition here refers to taking two or more distinct signals, and transmitting them over a single wire simultaneously. For the user that could mean something like being given depth information at the same time as being informed about the presence and orientation of a curb. Taking the superposition, extrema, arithmetic mean, and weighted mean of the competing signals will be considered as depicted in Figure 4.4.3(a). Consideration will then be given to virtually dividing the range of output intensities into distinct bands which will be allocated to particular system services.

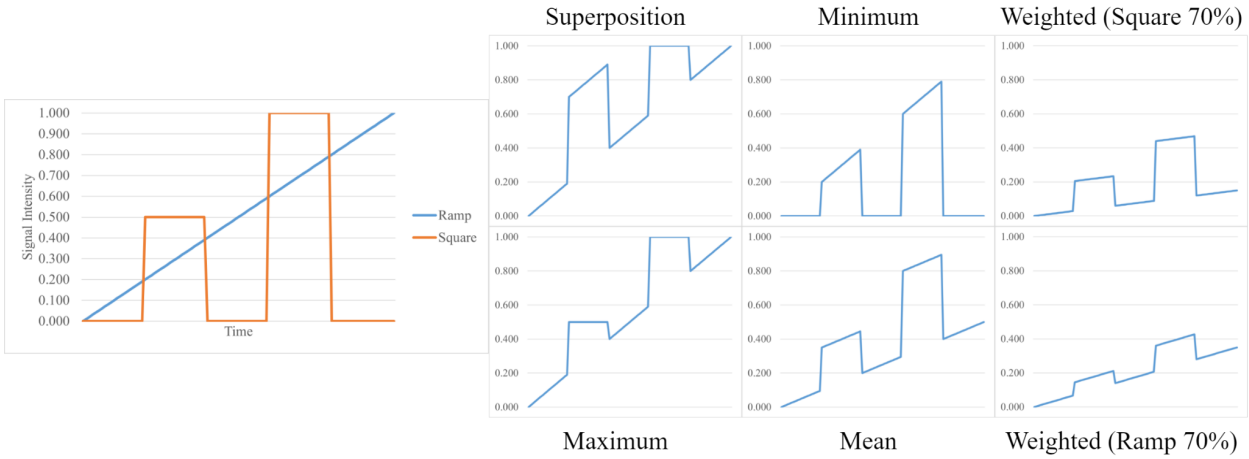


Figure 4.4.3(a) Simple Signal Compositions

Consider two signals produced by an individual approaching a building from the middle of some road. One signal is linearly increasing in response to the user approaching a wall. Another signal is a square wave whose second pulse is more intense than the first to represent an approaching step-up curb. Applying the composition techniques described yields the results shown in Figure 4.4.3(a). The minimum value, weighted means, and mean value all severely diminished if not practically eliminated the ramp signal. Both the maximum value and superposition of the signals significantly maintained the characteristics of the original signals. However consider a similar scenario with a step-down curb. The maximum value approach would likely overshadow the second pulse of the square signal. Superposition would maintain a second pulse, but the difference in the maximum amplitudes of the two pulses would be small. In both cases, the characteristic which intuitively determines the orientation of the curb is removed in the composite signal.

Prioritized Signal Compositions

An alternative approach which requires additional processing is to assign output priorities to services that are requesting to send a signal. The non-zero signal with the highest priority will transmit until it becomes zero. This technique will be referred to as superimposition. In terms of the scenario, giving priority to the curb detection service would mean that the square signals would always be present in the composite (See Figure 4.4.3(b)). To further create a feeling of two distinct signals coming over a single wire, the programmer should make sure that the frequency of the superimposed signal (square here) is much larger than that of the base signal (ramp), and that the time between repetitions of the signal is large. A final, similar approach is interruption. Allow signals from higher priority services to pass even if they are zero. It would be the programmer's duty to ensure that lower priority signals are not interrupted for significant amounts of time as this could be a safety hazard. For both prioritized techniques, a buffer around the superimposed or interruption signal may be added to further differentiate the signal from the lower-priority ones.

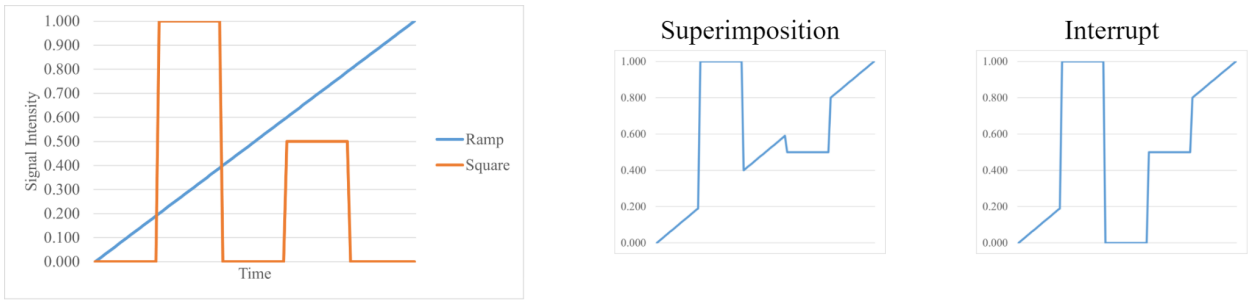


Figure 4.4.3(b) Prioritized Signal Compositions

Bandwidth Division

Allowing services to request output signals over the whole range of intensity levels may be problematic. Consider the superimposition approach example of the previous paragraph. A low-priority service producing an output signal at a constant 50 or 100% intensity would eliminate the perception of the lower or upper level of the square signal, respectively. Assigning ranges of intensities with or without overlap for each service may alleviate this issue. For example, assigning some 'depth response service' an allowable range on [0, 80%] and some 'curb detection service' on [0, 100%]. Since the depth response service can no longer produce output at 100%, the curb service will be guaranteed to have its upper level detected by the user. Proceeding one step further, overlap may be disallowed, creating a special band of signals dedicated to a single service. For example, assigning the depth service [0, 60%], a guard band of (60%, 80%), and the curb service [80%, 100%]. This guarantees that the curb service can always produce an output signal that is unique to it. The obvious consequence of these approaches is wasted bandwidth, which is especially troublesome in the case that the hardware implementation constrains the feedback range to a handful of discrete values (e.g., three intensities {0, 50%, 100%}).

Frequency Modulated Signalling

Thus far only amplitude modulated signals have been explicitly considered, especially with respect to depth estimation. Frequency modulated signals could also be used to transmit

information for a fixed amplitude of a sinusoidal output signal. For example, restricting the depth service to an output amplitude of 50%, but modulating its frequency between some minimum and maximum values to indicate depth. The considerations above still apply to this class of signals. As such, a combination of these schemes—that is, individual services operating with whichever one they choose—are possible for a single system. Extensive consideration however must be given to the time-to-learn requirement when implementing more complex output.

4.4.4 Depth Map Size Reduction

A CNN will be applied to input images provided by the camera. This network will produce a depth map with an estimation of the distance each pixel of an object is away from the user. This depth map will be made available to some process that will translate it into haptic output. To decrease computational complexity the number of parameters that enter this translation process must be reduced.

As mentioned in Section 4.2, the input map will be divided into evenly distributed sections. For example, the image may be divided into nine regions, each having the same number of pixels. Each region will need to be assigned a value based on the depth data available for that region (See Figure 4.4.4). The most representative depth value for the region is not desired. The process is instead interested in assigning values based on the information that is most relevant to the user.

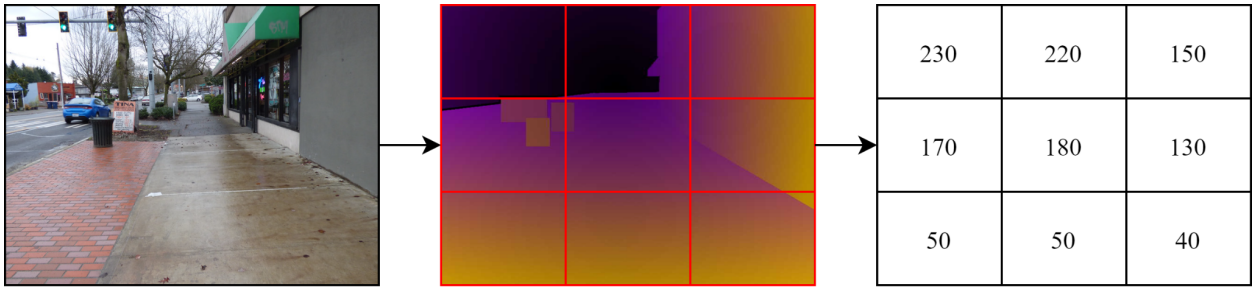


Figure 4.4.4 Raw Image to Depth Map: Raw Image from (SDOT, 2015)

Simple Size Reduction Techniques

Consideration will first be given to functions which are simple to implement such as taking an extremal pixel value, sum of values, euclidean distance, geometric mean, arithmetic mean, and mode. An additional complexity that must be addressed for this aspect of the device can be seen by looking at the image after depth inference in the figure. Although the bottom three regions do not contain any obstacles, a short depth is registered. Any technique implemented must take this gradient into account.

Naive consideration will be given to the approaches brought up in the previous paragraph without reference to the complexity of the gradient. In other words, assume that the pixels contained in the region only contain values associated with the distance of obstacles that the user may need to avoid. Also assume that the depth maps are noiseless. Taking the extremal value of the region is a simple operation and provides us with information on an object that should likely generate a response. However, consider cases where the object of interest is only partially in the

domain of one region, but mostly in another. The extremal value may end up repeated in both regions, which may give the impression that the obstacle in question is larger than it is. The sum of the values and the euclidean distance are also simple calculations which map pixels in a region to a single value. However, consider the case of a large, distant object in one region, and a close but small object in another. The chosen algorithm should alert the user to the small object, but both summing approaches may produce the same depth value for both objects. Mean based approaches will simply collapse the depth value to be somewhere between the extremal depth values, whereas the algorithm should produce a value closest to the minimum. The mode also suffers from similar issues. Consider the case with a large moderately distanced object versus a closer but smaller one. As long as more pixels in the region are dedicated to the large object, its depth value will be the one that is incorrectly given. With these concerns in mind, more complex approaches may be needed to produce the intended result.

4.4.5 Depth Map Translation to Output Units

Having fully described the options available for output units, and the values made available by the CNN, discussion now turns to the process of transmitting the information from the depth map to the user. Possible vibration intensity valued functions of depth, and the mapping of depth map regions to output units will be discussed.

Response Functions

After selecting the depth value for each region of the input image, consideration must now be given to how output units will respond to them. A function which maps a subset of depth values to output intensities must be defined. In general, consideration will fall toward linear, quadratic, and sinusoidal functions on a range of [0, 1] vibration intensity. Note that this discussion also applies to frequency modulated signals if a frequency range of $\{0, [f_{min}, f_{max}]\}$ is considered as analogs to [0, 1]. Using Figure 4.4.5 as a guide, consider the qualitative experience of the user when approaching an obstacle. The upward slope quadratic always under-represents the distance the user is from the obstacle, but as they approach, the intensity will increase more quickly. In this case the user may delay their response to the obstacle until they are very close to it. The downward sloped quadratic always over-represents the distance. Here the user may respond to the obstacle sooner than a sighted person normally would. The sinusoidal case shown in the figure under-represents the distance of far objects, and over-represents the distance of closer objects. Non-linear functions may be desirable to account for sensory attenuation, especially when dealing with inconsistent changes in depth. The velocity of the change in the intensity would serve as an additional cue of distance, which is not present in the linear case.

A function which depends on the velocity or acceleration of the depth may also be created. This would be especially useful in the case where the depth value is constant, in order to save power by reducing output intensity. Other use cases include pre-emptively indicating rapidly approaching bodies, and cutting responses for rapidly leaving bodies. This could be useful for emergency warnings. For example, an approaching car would not register as an obstacle until it is close to the user. Using the depth velocity the system may provide an earlier warning of its approach.

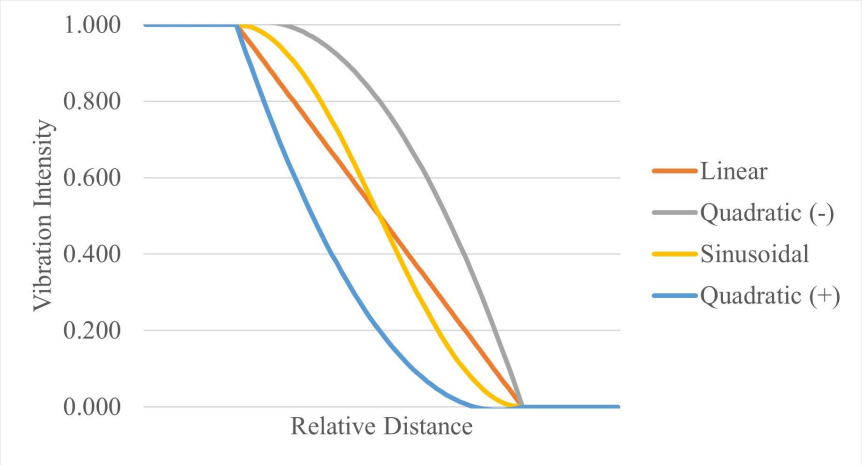


Figure 4.4.5 Intensity Functions of Depth

Depth Map Region to Output Unit Mapping

Consideration will now be given to how output units will interface with each other and the depth map provided by the software. Between regions of the depth map and output units, one of two mappings—one-to-one and many-to-one—may be used. In a many-to-one mapping, each output unit's state depends on a weighted composition of multiple output regions. This increases the size of each unit's receptive field compared to a one-to-one mapping at the expense of acuity. This would be useful in environments which contain several large, contiguous obstacles (covering one or more regions) as opposed to thin ones (covering less than one region).

4.4.6 Power Consideration

There will be a total of three sources of voltage in the schematic design:

1. 9-11 Volt output by the battery
2. 5 Volt output by voltage converter (step-down voltage)
3. 3.3 Volt output by the Compute Module

Following the flow of current, the first source powers the second and third sources. Below are the listed devices and their respective voltages.

In concern to the chance that an excess in power from the 5V source like during the use of multiple motors, may result in devices powered directly by the compute module such as the camera and LiDAR to not function correctly due to power limitations.

Reviewing this problem, the wattage of the battery for the designed device is one of the most important components in preventing fatal errors within the system and possible damage.

Estimated Wattage:

$$Watts = V_{Source} * I$$
$$= 5 * (300mA + 25mA + 1400mA + 85mA) + 3.3(30mA + 250mA + 1.7mA)$$
$$= 9.98W\ (estimate)$$

From the result it can be expected that the battery consumption will be an estimated 10W.

Name	Voltage (VDC)	Current (mA)
HDMI	5	300
Compute Module (idle and under load)	1.4 5	1400 Depends on Peripherals
Motor (per each)	5	9-25
Activity LED	3.3	10-30
Camera	3.3	200-250
LiDAR	5	2 (idle) 85 (load)
Speaker	3.3	1.7

Table 4.4.6 General Circuit Load

4.4.7 Output Motor Consideration

In terms of design, the choice in configuration for the feedback system has numerous advantages and drawbacks. The ones that were considered and will be mentioned include using a digital-to-analog converter (DAC), utilizing the PWM pins with MOSFET, and using DPI with a chip that would allow multiple motora to be daisy chained.

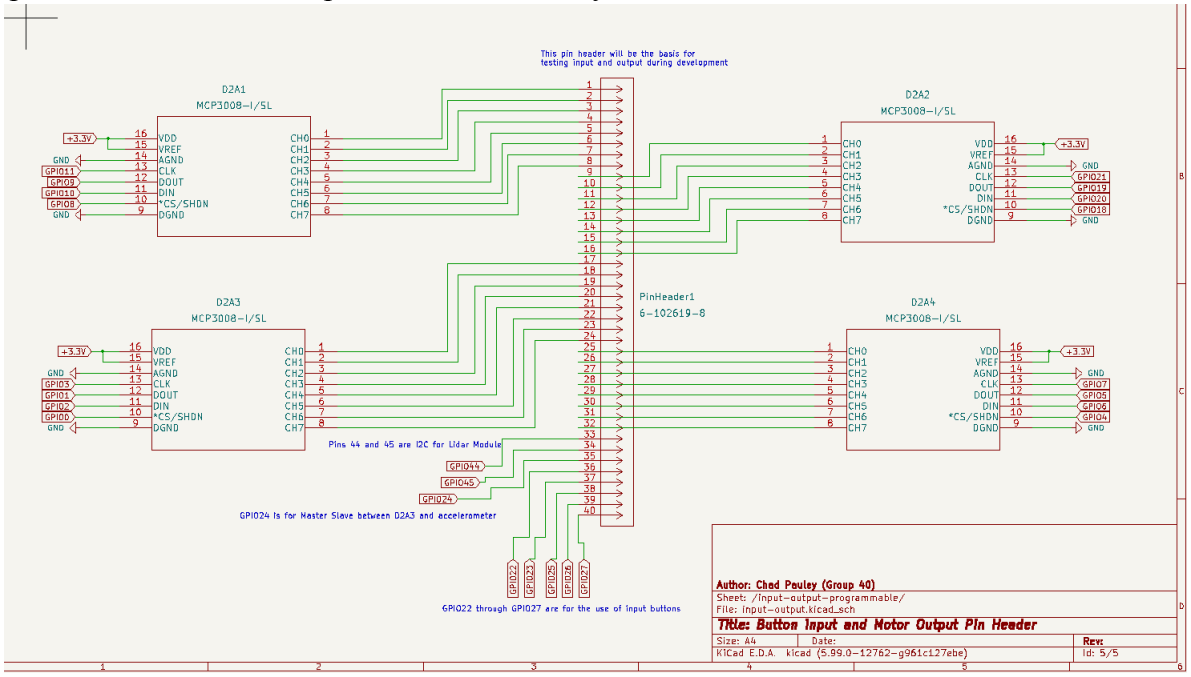


Figure 4.4.6(a) Motor Pinout

The advantage of using a DAC was for the use of implementing I2C as the CM4 has 28 out of the available 45 pins with compatibility. The major disadvantage was a lack of support for digital to analog conversion. There are many analog to digital converters (ADC), however the closest chip that would enable a use case would be utilization of the MCP3008. In hindsight, the intended purpose of the chip is ADC which was why the overall idea was disregarded.

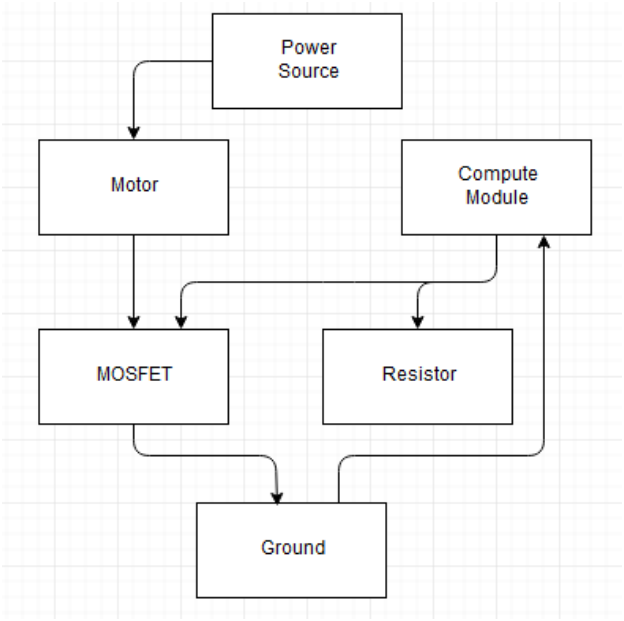
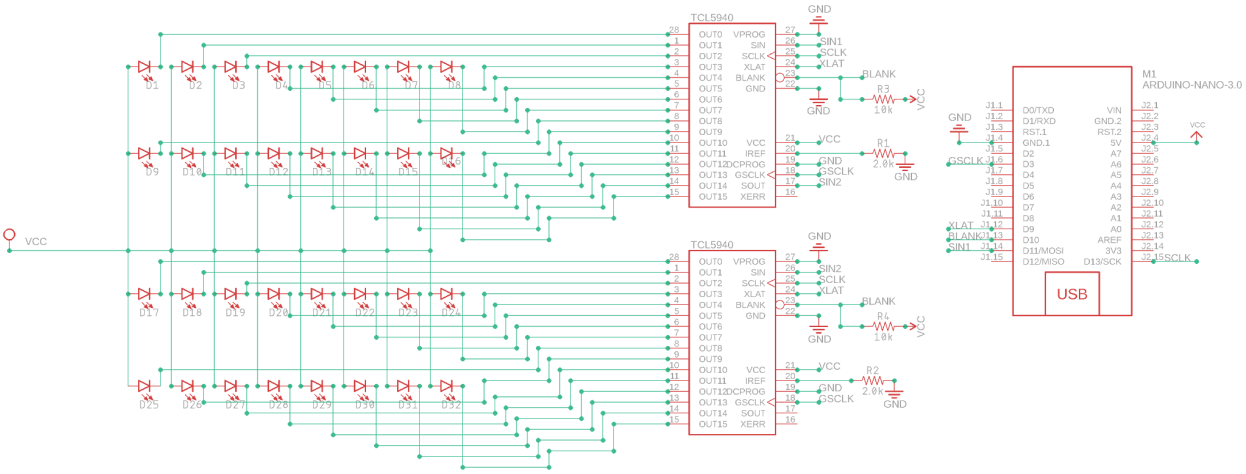


Figure 4.4.6(b) PWM with MOSFET

PWM pins work with the intended idea of using analog outputs to control the power of the haptic motors. Overall the design is also the most straightforward requiring the least amount of components and wiring. The problem with this design is the general lack of outputs supported directly by the CM4. There are a total of only 6 PWM pins that can be utilized, in which the project is regarded to need at least 9 separate motors. A solution considered was to convert the SPI, UART, I2C, or DPI to PWM.



4.4.6(c) Daisy Chaining Example

The final consideration was the daisy chain the motors together so the lack of GPIO was not a concern, however this causes major confusion during testing and debugging. There is also a big concern in regards to the mess of wiring that would have to be implemented into the clothing of the design which may cause discomfort and potential danger when wrapped up by wires (HealthenHacks, 2019).

4.4.8 Power Button Setup

The typical method of turning on and off the Raspberry Pi as well as other embedded systems including Arduino and MSP40 involves the physical removal of the wire that powers the source. This is due to the initial products lacking any form of switch or button that would allow for this feature. In consideration with the expectations set to allow for safe operation and durability of the machine. A button for the feature of turning on and off the device is implemented alongside 4 additional buttons which will be routed through the primary male-header unit found within the schematics section to allow for ease and repositionitioning of the buttons during trial use.
For the purpose of this device a normally open (NO) button is used, meaning that the circuit remains open until the button is pressed. Connected to the button will be a GPIO pin with the SCL configuration and a grounded wire on the opposing terminal. The use of this method is embedded into all Raspberry Pi operating systems and will require no overhead or additional programming to have this functionality.

4.5 Parts Selection Summary

In this section many distinct parts were compared to find the best options for inclusion in this project. When choosing parts, the main considerations were availability, cost, and effectiveness for our use. The parts chosen for this project are the best balance between quality and cost efficiency for the project.

Looking first at the computer selection, the Raspberry Pi 4 will be used in initial testing and has arrived in time for software testing. The Raspberry Pi 4 compute module that will connect to the PCB in the final project, is still on backorder, but should arrive in February at the latest, but hopefully over the holidays. The convenience of the Raspberry Pi 4 for testing is ideal in this case.

Many of the issues that will be encountered with most of these parts is the availability. The LiDAR and Accelerometer have both been put on hold and will hopefully come in around christmas. If any part is faulty or is damaged in the testing phase, this may become an even bigger issue for reordering parts. Taking that into consideration, the testing of each individual part initially will be incredibly important in the case of needing to reorder replacement parts. When the devices are ready for assembly, the current and voltage being supplied to the devices will need to be double checked in order to avoid ruining the part.

Thankfully the camera has also been delivered and is available to begin the computer vision programming and testing as soon as the initial coding is completed.This is ideal as majority of this project will depend on the successful operation of the computer vision programing. The sooner that can be tested and worked with will be better.

The speaker and motor were both ordered and delivered from amazon, so the availability will most likely be stable and both are available for one day shipping.
Our final parts used for this project can be found in the table below. Any other small parts like buttons, resistors, and transistors will be donated from the team's personal collections of electronics parts.

	Part	**Cost**
Computer	Raspberry Pi 4	$55
Camera	Raspberry Pi Camera Module 2	$25
LiDAR	Garmin LIDAR-Lite v4	$60
Accelerometer	MMA8452Q	$10
Motor	tatoko Mini Vibration Motors Flat Coin Button (20)	$15
Speaker	MakerHawk 2pcs Ar Duino Speaker	$10

Table 4.5.1 Final Parts Selection

5.0 Related Standards and Realistic Design Constraints

5.1 Standards

The voluntary standards include the rules, practices, and guidelines for each component in the designed device as well as the compliance for the device as a whole. These components include; but are not limited to the Raspberry Pi Compute Module (CM4) as well as peripheral standards.

The peripherals for the device will comply with the standards for the country the product is used within. Examples of peripherals include the external memory storage (SD), Raspberry Pi camera module (*Raspberry Pi, 2020*), LiDAR unit, and haptic feedback motors.

5.1.1 Raspberry Pi Compute Module

The Raspberry Pi Compute Module 4 will be referred to as CM4 in further existing documentation. Existing warnings and requirements of the CM4 include the use of a 5 volt external power supply unit that is safe for use in charging and discharging applications. The device will be embedded into clothing but will be housed in a well-ventilated case environment in order for the module to work effectively within operating temperatures. Furthermore, the product will be located on a non-conductive surface. Operating temperature for the device is -20 degrees to 85 degrees Celcuis. Form factor for this unit is 64 mm by 74 mm. The unit is expected to not be exposed to water, moisture or conductive surfaces. Heat sources are also to be avoided (*Raspberry Pi, 2020*).

5.1.2 External Memory Storage

External memory storage will be used in the base model of the CM4 through the use of a secure digital (SD) card. The use of this card will be in industry application which requires higher reliability and endurance than the consumer model. The model of SD is expected to operate within a range of -25 degrees to 85 degrees Celsius as according to the SD Association's operating temperature (*ATP Electronics, 2020*). Data integrity is expected to be held through the auto refresh and dynamic data refresh conducted by the file allocation done of the SD component to reduce read disturbance and hold data integrity.

5.1.3 Raspberry Pi Camera Module

Raspberry Pi camera module is applicable with the CM4 specifications with a 22-pin header unit. The MIPI CSI-2 is compatible through the use of a protocol layer of I2C. Due to low power consumption there are no concerns due that would involve the harm of the user. As the camera is on a PCB, a voluntary standard for the unit will be to avoid moisture as expressed in section 5.1.1 for the CM4.

5.1.4 LiDAR Unit

With the use of a LiDAR, regulations are divided into five separate classes depending on radiation emission and transmitted power. The project will use a low power LiDAR device classified in class 1M, noted as a mostly eye-safe; invisible laser category (*EETech Media, 2021*). This category is deemed safe by the American National Standard Institute (ANSI) and International Electrotechnical Commision (IEC) except under the condition of the optical device being amplified or magnified as in unintended use. The device is noted as having a precise measuring range in a controlled weather environment but diminishing performance in environments that may affect the reflectivity of light such as rain or fog.

5.1.5 Haptic Feedback Motors

Motor products have minimal standards in motor design for devices measured in units of millimeters.The reliability of a DC device heavily relies on its intended use. A rating of less than 5 volts will be used to prevent potential dangerous static surrounding the device as well as a layer of insulation to avoid accumulated voltage from the clothes, becoming a concern depending on environmental conditions like humidity (*Precision Microdrives, 2021*). Should the unit be found to create static that would harm the device itself or its user, then a ground wire on the outer shell of the motor will be used to ground any possible discharge.

5.1.6 Raspberry Pi Carrier Board

The carrier board is the connection nest for the Raspberry Pi Compute Module (CM4) to generic output ports (HDMI, USB-B, etc.) and GPIO. Trace length matching in high speed design must be held as in standard to avoid missing timing of critical pulses in the system clock frequency of the CM4 (Cadence Design Systems Inc, 2020). Without this standard the entire design of the control system may be compromised (*NWES, 2020*). The synchronous protocols affected by mismatch timing include SPI, I2X, CAN, ISA, ATA, and PCI.

Due to cost constraints and limited time in prototyping, a 2-layered board with basic peripherals will be implemented into the carrier board. When using a 2-layered board it is far cheaper when developing prototypes. The use of industry standards such as a PCB thickness of 1.6 mm and a green soldering mask will be used in the development as well as the final design of the device.

5.1.7 Packaging and Labeling

As the user for the device is expected to be blind, it is compulsory for the manual to be read to the user prior to first use to avoid misuse of the device. Choking hazard and voltage warnings are needed for wall charger unit. A battery warning for weather operations and expected amount of lead in product are to be noted as per CPSIA requirements. Sharp endpoints should be rounded to avoid the harm of the user by use of contact with the device.

5.1.8 Energy Conservation

According to the United States Energy Policy and Conservation Act (EPCA) electronic products must be measured with estimated annual operating costs for periods of use and charging (*NIST, 2017*). The Federal Trade Commision is responsible for these labeling requirements(*U.S. Government Publishing Office, 2013*). According to the Transportation Security Administration (TSA) any lithium battery exceeding 100 Watt hours will only be allowed in carry-on bags with airline approval including one spare battery but not exceeding 300 watt hours (*TSA, 2021*).

The Electronic Code of Federal Regulations has compliant standards to the design and use of a battery charger for any device developed on or after June 13, 2018. The device must be less than or equal to the maximum unit of energy consumption (UEC) for the appropriate product class and battery rating. The table below includes the specifications and product class descriptions.

Product Class	**Product Class Description**	**Rated Battery Energy (Ebatt)**	**Special characteristics or battery voltage**	**Maximum UEC (kWh/yr)**
1	Low-Energy	≤ 5 Wh	Inductive Connection	3.04
2	Low-Energy, Low-Voltage	<100 Wh	<4V	.1440*E_{bat}+2.95
3	Low-Energy, Medium-Voltage		4-10 V	For E_{batt} <10 Wh, 1.42 kWh/y E_{batt} ≥10 Wh, 0.0255 * E_{batt} + 1.16
4	Low-Energy, High-Voltage		>10V	0.11 * E_{batt} + 3.18

Product Class	**Product Class Description**	**Rated Battery Energy (Ebatt)**	**Special characteristics or battery voltage**	**Maximum UEC (kWh/yr)**
5	Medium-Energy, Low-Voltage	100-3000 Wh	<20V	0.0257 * E_{batt} + .815
6	Medium-Energy, High-Voltage		≥20V	0.0778 * E_{batt} + 2.4
7	High-Energy	>3000 Wh		.0502 * E_{bat} + 4.53

Table 5.1.8 Energy Usage Classification

Table referenced from Code of Federal Regulations for electrical devices (National Archives, 2021). The device design is expected to be in the Low-Energy, Medium-Voltage category as the batteries will be rated at about 9V and stepped down to 5V output thereby powering CM4 and peripheral devices (range within 4 to 10V). Power Supplies using the NEMA 1-15P or 5-15P input plug (typical North America two prong charging method) should have average load adjusted meeting expectations of the ECFR.

5.1.9 Wire Standards

The wire selection for this project was relatively simple. Wires that were thick enough to provide enough power and thin enough that they are not taking up too much real estate and are able to be easily concealed in the jacket and flex with the fabric. Looking at wire standards, it was important to make sure that the maximum voltage was not reached and that the maximum current that will be running through was about 80% under the maximum rated current. The camera will be excluded from this, Since it uses a ribbon cable to connect to the board.

Device	Maximum Voltage Rating	Maximum Current Rating
LiDAR	5.25V	85mA
Compute Module	5V	3A
Motors	3V	150mA
Speakers	5V	625mA
Accelerometer	4V	150uA

Table 5.1.9(a) Device information

Most of our devices use a very small current and voltage to operate under normal conditions. As shown in the table below, the device with the highest current rating is the compute module.

AWG	Conductor Diameter (in)	Maximum Current Ratings	Resistivity (Ohms/1000') At 77deg F	Resistivity (Ohms/1000') At 140 deg F
22	0.0253	3 A	16.5	19
20	0.032	5 A	10.4	11.9
18	0.0403	7 A	6.51	7.51
16	0.0508	10 A	4.09	4.73
14	0.0641	20 A	2.58	2.97

Table 5.1.9(b) AWG Wire Calculations

Seen in the table above is the typical conductor diameter, and maximum current ratings of different gages of wire that could be used for this project (PowerStream 2021). These are the standard for American wire gauges. Best practice for wire selection is to only put 80% of the maximum current on the wire, to ensure that the wire will not burn up. If our max current would be 3 A plus the maximum current for the next highest device .625 A, the minimum current needed for the wire to handle all devices is 2.625 A. Comparing this to the table above, the best voice for these constraints would be the 18 gauge wire. Looking at the maximum current being 7A, the 80% threshold for that would be about 5.6A, well over the required amount.

Next thing to be considered would be the resistance on the different wires that will connect the devices together. Using the formula R = pL/A where A is the area of a circle, L is length, p is resistivity which can be found in the table above (.00651 ohm per ft). The diagram below shows an estimated placement for each device and the estimated length of wire to attach them to the main computer.

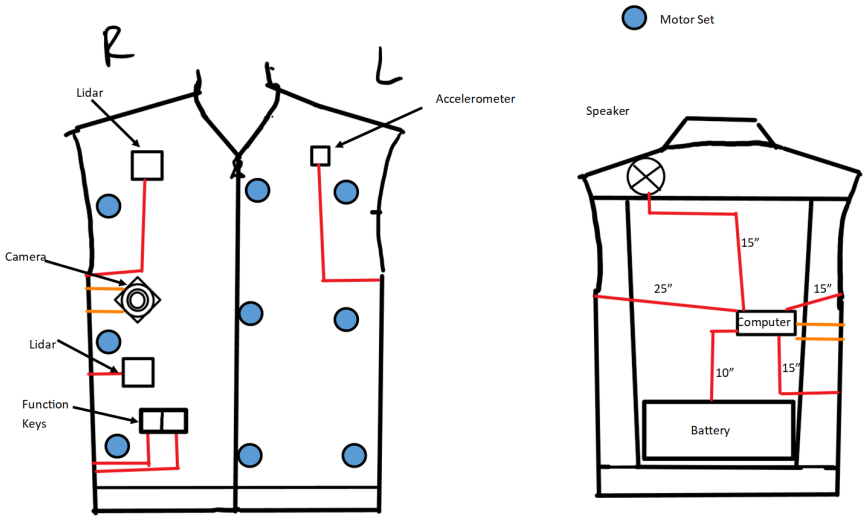


Figure 5.1.9 Wire Resistance Visualization

Using the formula above the estimated resistance per connection. The lengths of these connections are overestimated, allowing for a range of error.

Device connecting to Compute Module	Total Current on Wire	Length	Resistance
LiDAR	3.085A	1.25	.00264
Motors	3.15A	Longest route 2.5’	.00517
Speakers	3.625A	1.25’	.00224
Accelerometer	3A	1.25’	.00271

Table 5.1.9(c) Wire Resistance

One thing that needs to be considered in wire selection is the flexibility and thickness of the wires. They should be able to be hidden very easily inside the lining of the jacket, invisible to an outside user. All wires will be secured by fabric to ensure that no movement can shit or damage the wire during use.

The main concerns regarding wire is the possibility of wires melting or any short that may occur. This is a particular concern considering this is to be worn by a person and so should be as safe as possible. Choosing a wire that is well over our maximum current is important. When soldering on the different components, it will be incredibly important that all of our wiring is placed neatly and that bare wire will not fall in contact with each other.

5.2 Design Impact of Relevant Standards

5.2.1 Raspberry Pi Compute Module

The use of a raspberry pi compute module allows for a custom build of a carrier module. This gives the developer the ability to remove irrelevant components such as ethernet, USB-C, excess hdmi and USB-B thereby creating a more compact design and saving on cost.
Due to constraints concerning moisture; a housing unit for the device will be required that enables for some amount of airflow without exposing the circuit board.

5.2.3 Testing and Training

ANSI does not enforce the use of specific testing facilities but do require the device to be verified as not being hazardness toward one's health, follow safety regulations, and not harm the environment through the use of some third party testing should there be some possibility of the aforementioned three.

Insulation resistance test includes the use of voltage to induce current to flow through a PCB which is then calculated to allow the finding of the resistance value for the entire device's insulation. HiPot testing is another type of test.The dielectric withstanding voltage (DWV) tests the dielectric breakdown, which is often done before the insulation resistance test. The insulation resistance test ensures that the insulation can protect the circuits on the board. Under high voltage the PCB is measured by a HiPot tester which checks for leaks in the current.

5.2.4 Standards on Printed Board Design

The most important standard acknowledged by the general community of PCB design is IPC-2221. This standard covers a variety of topics in all types of designs. The first requirement involves the clearance of component leads, uncoated conducting areas, mounting hardware and heatsinks. The following table shares a comparison of each component to its relative required clearance.

Component	Clearance
leads	.13mm
Uncoated Conducting Areas	.75mm
Test probe sites	No more than 80% of component's maximum height
Mounting hardware	No more than 6.4mm beloy outer layer
Heatsink	2.5mm larger than hole

Table 5.2.4 Clearance in PCB Design

The next requirement involves the thickness of traces made upon a PCB to connect components. Thickness and width varies on the purpose of the trace. Thicker traces are made when connecting power components so many sources can be properly given the current necessary to perform their intended functions. High voltage circuits are to be checked multiple times using the insulation resistance test and HiPot test. It is necessary to refer to the CTI when choosing material where voltage and current will pass otherwise the material will break down (Sagar, 2021).

5.3 Realistic Design Constraints

As discussed in 1.4, several constraints were identified that could potentially impede the functional completion of the project internally and externally. To define a constraint for the purpose of this project, a constraint is any limitation or block on or in the design that must be considered and tackled for a successful solution. There are many different types of constraints both internal within the project or external to the project, from outside factors, as stated above. Future discussion will look into economic, environmental, ethical, health and safety, time, and testing constraints to analyze the external limitations that could arise. To analyze the internal limitations that could arise, software and hardware constraints will be analyzed and discussed.

5.3.1 Economic Constraints

Within economics, being the organization of industry, money, and trade, the external constraints surrounding the affordability and socioeconomic acceptability of the depth perception haptic system is observed. It could arise that although the design completed a reasonably priced projected budget of the duration of the project, the actual total costs of acquiring parts, shipping out and/or circuit boards, and all around assembly using a number of high class tools and services could exceed well over what was intended.
This could lead to a halt in production or problems in the completion of the DPHS. Additionally with the population coming out of a pandemic, certain factory parts have become limited supply and may be more difficult to acquire, whether in shipping time, resupply duration, or increased price. Furthermore, to ensure that by the end of the project, the device must adhere to all industry standards and is marketable to be efficiently used by a blind or visually impaired individual. Therefore, questions must be asked such as the following::

- ‘Is this device comfortable to wear?’
- ‘Could this be cost effectively replicated for distribution at an affordable price?’

5.3.2 Environmental Constraints

Examining the potential environmental performance of the depth perception haptic system, it was identified that any event in which the competence or effective functionality of the device is restricted or compromised classifies a constraint. With the unpredictability of weather conditions, from rain, to extreme high or low temperatures there arises the potential for the technology of the device to be affected, such as if it unintentionally got wet and shorted. Many dangers for the device exist and so to avoid such constraints it is desired to ensure that the product is constructed as a durable device (see section 5.3.7 for more details on hardware constraints).
Most certainly a visually impaired person at any level could experience difficulty navigating their surroundings, and even with assistance can experience some encounters with obstacles,

such as walls, or the floor from a fall. So, even though it is intended for this device to decrease the occurrence of such accidents, it must be ensured that the device is sturdy enough to survive potentially high impacts, thus providing a constraint that can be approached in the design and developmental stages.
Environmental hazards to consider:

- Rainfall
- Excessive sweat
- Extreme cold, i.e below 32°
- Extreme heat, i.e above 100°
- Falling into water
- Falling onto concrete, grass, asphalt, etc.
- Bumping into walls at walking to running speed

5.3.3 Ethical Constraints

To ensure that the device is created as an ethically sound product, it is necessary to ensure that the device follows societal norms and does not offend any major standards. How much testing proves to be enough testing? If only 80% of large obstacles in direct view are detected within 3 feets of the individual, clearly the device needs to go through more testing and be revised, but realistically, especially with this being the first model of the device, 100% of all obstacles large and small, thick and thin, will not be all detected, so at what point the device is appropriately functioning within the predicted means to be efficiently and safely used by an actually blind or visually impaired individual. This proves to be potentially a very plausible constraint, as testing for this device will be extensive. This will be to ensure that trust in the product is widely gained.

5.3.4 Health and Safety Constraints

Observing how the Depth perception haptic system holds up to ethical constraints raises some additional concerns pertaining to how it holds up to health and safety constraints, those being the safeguards implemented. Given that this is an electrical device, the event that an exposed wire could harm an individual or set fire to a flammable object is very possible, so the design must account for any such contingency, largely by ensuring that all exposed circuitry and wires are safely housed within the fabric. The product is also desired to make sure the normal function of the device is within comfort for an individual, i.e making sure the haptic feedback levels never reach a frequency and force that too powerful and actually jarring to the wearer, or even that the speaker levels aren't too high or loud that it an harm the wearer's or surrounding individual's ears.There also arises that a blind or visually impaired individual may be encouraged to navigate streets without any additional aid or assistance, and while the depth perception haptic system is intended to allow an individual to potentially navigate the world around them without the use of additional external tools, it isn't advised that a person exclusively use the dphs in such a fashion to ensure the maximum safety of the individual.
Safeguards to consider:

- Insulation
- Inflammable fabric
- Encased power source
- Universal fit/stretchy fabric
- Use in conjunction with a cane, though use without a cane is possible

5.3.5 Time Constraints

Time constraints, those being limitations imposed onto the project by external factors that affect the completion time of the project, are a huge point of interest to the project. Several factors, such as some mentioned in section 5.3.1, like delays in shipping due to shortages caused by the covid pandemic and then certain functions requiring the hardware parts to be programmed for the project to progress, is a prime example. One of the goals is to tackle time constraints using detailed schedules, primed to put us ahead of typical chronological progress, to allow us ample time to avoid devastating halts in production.
There are also time restraints, stemming from us running out of time on the project to implement intended functions, like not allocating enough time to implement the necessary machine learning for the image processing and analysis of shapes and surfaces for the obstacle detection of the depth perception haptic system. Again, this can be avoided with proper planning and staying ahead of the detailed schedule.
Issues that could prove to be time constraints:

- Machine learning for image analysis of shapes and surfaces
- Chip shortage resulting in issues with finished PCB delivery

5.3.6 Testing Constraints

Examining what could arise to be testing constraints that inhibit the effective and successful testing of the software to ensure peak functional performance, several potential hazards were identified, see also sections 5.3.7 and 5.3.8 for more details. It could arise that testing resources could be miniscule, such as lack of enough willing participants and limited power supply. Additionally, improper testing environments due to miscommunication between coders or improper hardware and software compatibility, along with numerous other constraints could arise. One of the goals for the project is to maintain communication within the group for all team members to ensure seamless assembly and compilation of all of the contributions to ensure the ability to properly test the device as the implementation of the project ensues, and with ensuring all the necessary resources needed for testing are available when the time comes for testing, it is intended that actively preparing for testing in the event errors or bugs occur would later become a testing constraint can be dealt with ahead of time.

- Limitation on battery
- Lack of available participants who have the desired disability (blindness)
- Small population size for testing
- The max detection distance not being far enough to allow proper reaction

Although a lot of consideration and planning went into the design of the Depth Perception Haptic System, there are no doubt also issues that can arise at any step throughout the progress of the project, both in the software and in the hardware.

5.3.7 Hardware Constraints

A hardware constraint is a limitation that is observed to be a potential threat to the project within the hardware architecture of a device. For the software to successfully and effectively be present on a device, these hardware constraints must become a system requirement. To identify any hardware constraints within the physical architecture of the depth perception haptic system the architecture itself is examined, processing power needed, memory, and any peripherals (*V Skills, 2019*).

Hardware Constraints Include:

- As the raspberry pi module and camera are both compatible firmware drivers aren't necessary
- USB-B needs time lag to match serial, see section 6.1 for more information
- Damage can occur with the pinout General purpose Input/Output units. These are essential, as they are a physical interface for the raspberry pi and the external environment. Thus Pin protection for each motor is required and the implementation is done through the use of diodes
- As stated in section 4.4.6, there could arise a case where excess power from the 5V source or Unverified current throughout the system, may cause system to freeze or crash rendering multiple motors or components to function incorrectly, such as the compute module, any motors, or the HDMI, this is because these components and others run on the 5V source
- To prevent burnout in the USB-B port, capacitors are required to block high frequency signals, thus making a low pass filter for AC circuits, or in this case evens out the voltage and prevents against spikes, as this is DC. Serial protection, TPD2EUSB30, is required for the USB-B. Additionally, components must be as close to USB as possible to prevent certain lag.
- Another constraint in the physical hardware is that of the number of layers of the PCB. Between a 4-layer and a 2-layer layout 2-layer was decided, which while being cheaper is more difficult to configure. This is because there is a lack of routing space, thus a lack of ability to control trace impedance. 2-layer also tends to exclude showing propagation delay and, in addition, is much thinner and so impact could cause breakage. See section 7.2.2 for more
- An additional constraint with the physical layout is that the Current version of the board has 40 pins, where all 40 pins are used. These pins obviously are made of metal, and these pin headers are susceptible to break or bend, which will compromise the functionality of the device
- The Raspberry Pi Compute Module, CM4, has particular functioning conditions and thus must be ensured to have these conditions have the best chance of existing without the user needing to provide such an environment. Operating temperature for the CM4 include those within -20 and 85 degrees Celcius and the unit is not intended to be exposed to water, moisture, conductive surfaces, as well as direct heat sources. To assist that the device will be housed in a well-ventilated case environment and be embedded into clothing

- Another constraint is that of the limitations of LiDAR. Just as other sensors, such as ultrasonic sensors, which have trouble detecting through soft objects like clothing, LiDAR may fail when looking at bodies of water or any form of glass/reflection, which will need to be taken into consideration in testing
- It must be taken into consideration that the Barrel Plug, which will be used for an easy conform into the vest for easy charging, may get clogged up with grime inside due to outdoor use
- Within the device, A voltage regulator is used to directly convert the batteries to a usable power source of max 5V, 3A. What can happen, however, is that if a single resistor, inductor, or anything of the like stops working then the board as a whole will stop functioning as well.
- Another constraint observed is that using larger lines allows for easy current transport but also enlarges the board and seperate parts, making for a less compact model and increases the weight
- Another constraint lies in deciding which HDMI to go with. Some versions include 15 pins and 19 pins, decision depends on specifications of I/O depending on the rest of the firmware
- In this project four SPI to analog, digital to analog converters(DAC), are used. It was observed that there exists potentially enough channels but when it comes down to designing the board it becomes possible that signals crossing may cause something to go off unintentionally regardless of the number of board layers so extra consideration must be applied
- Pulse-Width Modulation(PWM), a technique of modulation that varies the width of pulse while keeping the frequency at a constant, will be used for speakers, however the performance of the pins is unknown, where they could potentially only make beep tone or not have ample power and thus may not be loud enough, will see in testing
- Due to lack of availability, another constraint lies with the issue of finding many footprints. It could arise that the alternatives chosen for the HDMI and the USB-B symbol and footprint used to replace the original units could be incompatible, even though all the pins are the same. This also goes for any of the components. Any part that is not part of a model group has the potential of not being able to run with any to even just one other part, and that causes non-functionality, so extra attention to detail and consideration must be taken before ordering parts.
- When it comes to the output haptic feedback, not using a motor controller limits us to only two levels of haptic feedback with only a total of 6 usable motors, which makes the original plan of having a wider range of haptic feedback to create a more comprehensive picture of the wearer's location in proximity to obstacles. Additionally, the motors themselves are fragile and may break if too much voltage or current is applied. Will be monitored in testing .

5.3.8 Software Constraints

Like discussed above, in the hardware constraint section, a constraint is a limitation that is observed to be a potential threat to the project, and within software specifically it is a limitation that affects or slows a system down in any way, thus hindering the system from accomplishing the intended goal and progressing normally. To identify any software constraints within the

architecture of the depth perception haptic system different types of software constraints that could arise are emained, those being policy constraints, paradigm constraints, market constraints, people constraints, and equipment constraints (*Parker, 2020*). For the purpose of this project, the only software constraints that don't directly apply are those of the policy constraint variety, as those are defined normally by corporate or company policies. However, the project could characterize the Senior Design guidelines established by the University of Central Florida College of Computer Science and Engineering to loosely count as a corporate covering.

People Software Constraints:
A people software constraint is one characterized by a limitation caused by the number of people on a project, whether it's having too few or too many people on a project. Additionally, a hindrance to progress is caused by the skill level of each participant, whether it's a lack of knowledge or too many knowledgeable persons in one area and not enough coverage of the entire scope of the project.

- It was observed that the project requires us to use specific programming languages that all group members have had somewhat little knowledge of prior to this semester. It is planned to code in a combination of, or using either or, the programming languages Python and C++. For the CpE majors, who will be handling the bulk of the coding, while the overall knowledge of coding principles is possessed, there could no doubt arise pockets where research of python or c++ dependencies and guidelines is required to complete the desired task.
- The scope of software architecture and design required for this project could prove to be too vast for the number of members in this group, that being four persons. One of the goals is to use diligent pacing and stewarding of required tasks and documentation to overcome any potential lack of coverage.

Paradigm Software Constraints:
A paradigm software constraint is one characterized by a limitation caused by beliefs. These could be disagreements in software structure, method, etc. One such example being in productivity being directly correlated to code line count.

- The choice to code in either Python, which can be easier to implement as the components have a built in compatibility with the language but can have an expensive run time or in C++, which can definitely be faster but harder to implement could be a disagreement that arises between programmers.
- As mentioned above, the amount of physical code that is used can become vast and hard to keep track of so adhering to the coding standards listed in section 6.4.1 could help with ensuring organized productivity is maintained.
- Any personal choice made independent of the group could cause tension in the progress of the project, thus extra attention to maintain thorough communication between all parties within the group and even with supervisors, those being professors and senior design committee members, is essential.

Policy Software Constraints:
A policy software constraint is one characterized by the procedures and policies attributed to the company covering the projects. Essentially it is restrictions in creative design or implementation freedom due to the specific way a company operates.

- Specific to the Depth Perception Haptic System, corporate or company covering doesn't directly apply, as stated above, however, this being a product of the senior design course, the group is limited to adhere to ABET, Eli2, and ucf policies, so and setbacks that result from any such restrictions could classify as a software constraint

Market Software Constraints:
A market software constraint is one characterized by limitations caused by the extent of software engineering being delivered to the consumer in the final project. It could arise that there is over complication of what would otherwise be a simple solution, such as providing features in the software that are not necessary to the system or user, taking up unnecessary space.

- It was included in the project's objectives and goal that the necessary features and functions this device should be able to support at its most basic and final model, these be required functions. Going from there, the stretch and advanced goals and functions that can be implemented given there is enough time. What could cause restrictions in the project process is focussing on these stretch and advanced goals, or even any other scope creep functions, before or over first the required functions.
- Improperly organizing the marketing of the software could create the image of an ineffective solution to the blind and visually impaired aid navigation that this device was initially intended to provide

Equipment Software Constraints:
An equipment software constraint is one characterized by limitations caused by slow running or out of date equipment, faulty or broken equipment, or even lack of necessary space and tools to properly complete a task. These can be anything from computers used, operating systems, IDEs, programming environments, component details, and so on. To identify any equipment software constraints the state of all devices, systems, and applications used throughout the duration of this project are examined.

- Factors like:
  - The age of device
  - Version of operating systems each individual's personal computers
  - Broken keyboards or mouses, usb ports for testing, screens
  - Even for the pi 4, there is discussion for how much ram is required, 2GB or 4GB to ensure enough space is available for things like the camera feed and executing models
  - Compatibility of the environments coded in parallel and upload to a github repository
  - A system crash erasing necessary code that wasn't yet uploaded or shared

  Could all affect the progress of the project at any stage of the project.

The group will reevaluate these constraints throughout the duration of Senior Design

6.0 Project Hardware and Software Design Details

6.1 Initial Design Architectures and Related Diagrams

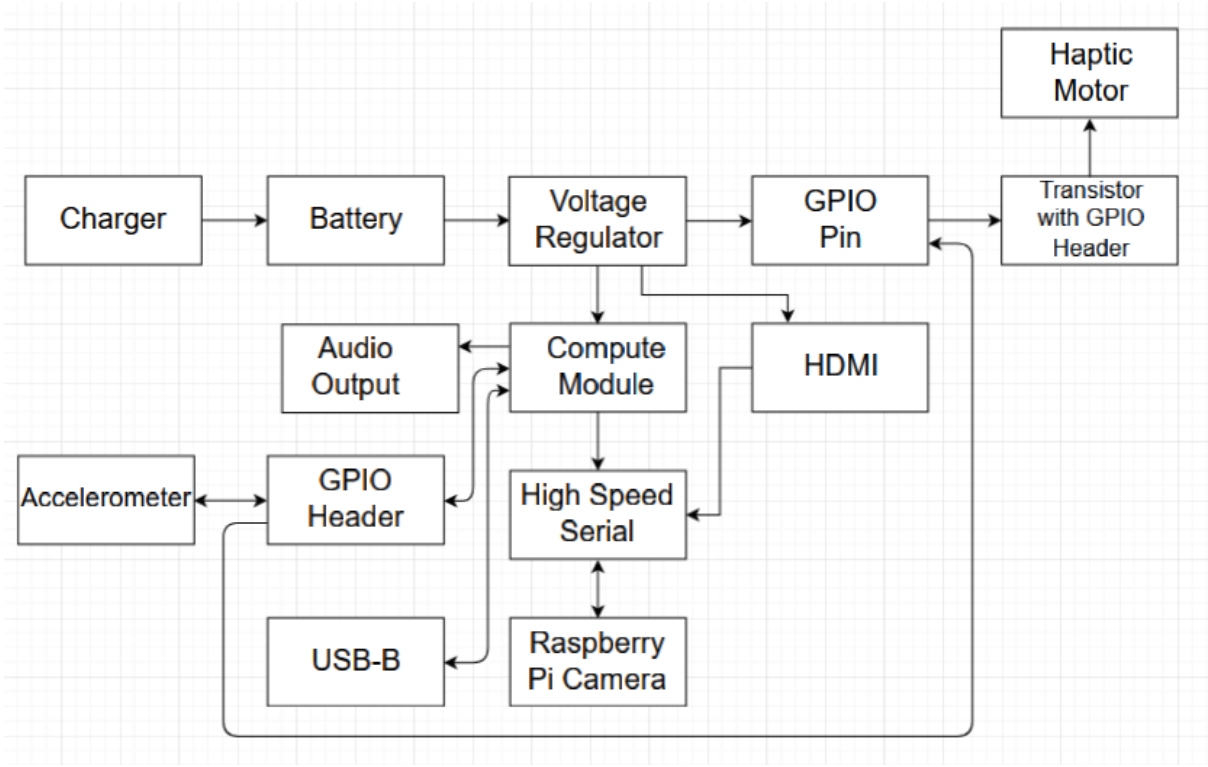


Figure 6.1 Hardware Diagram

In regards to the figure above, each component will be reviewed in this section with priority left to right, top to bottom. The overall design relies heavily on the compute module and the pins it offers to the input and output peripherals.

6.1.1 Charger and Battery Design

Charger specifications include the need for an internal filter. The use of a bandpass filter within the design will protect the batteries within the device and the components used in conversion from alternating current to direct current. Further purpose includes preventing the device from risk of exposure to static when charging.

In terms of battery design, a 9 to 11 volt battery with expected high wattage will be used to power the entire device. A tamiya wire configuration is expected between the battery and voltage regulator to allow the user the ability to swap batteries in cases of long use. Further connection standard between the charger and battery will use a barrel plug to allow for easy charging.

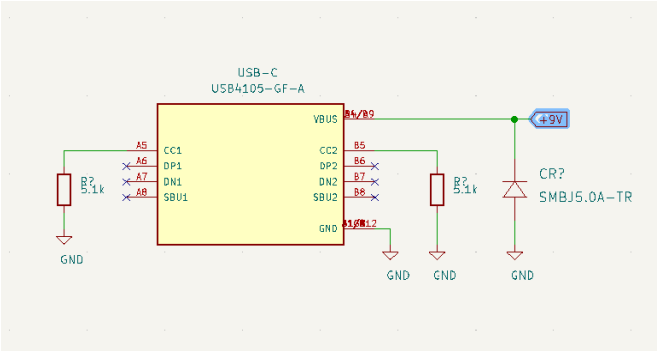


Figure 6.1.1 USB-C Schematic

Issues pertaining to the use of USB-C due to internal regulators limit its ability to be implemented into this project but were initially considered and replaced by a voltage regulator that can be controlled.

6.1.2 Voltage Regulator

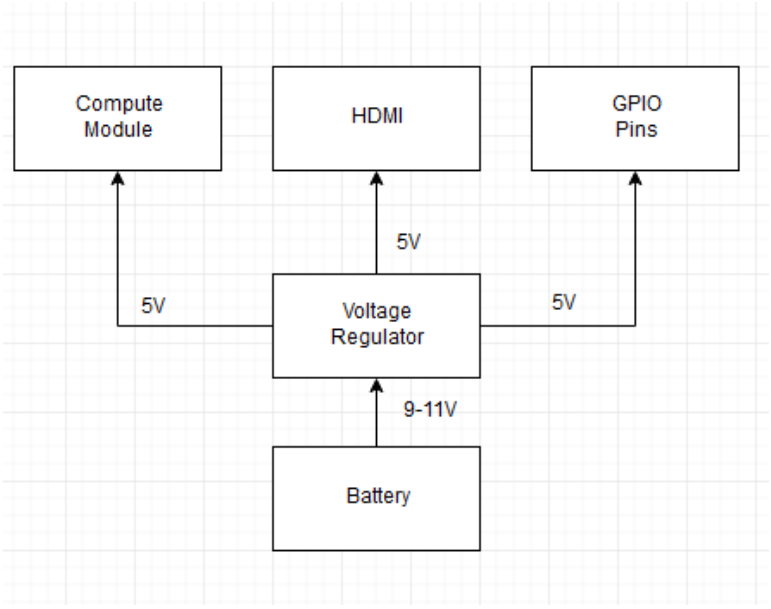


Figure 6.1.2 Voltage Regulator Diagram

The voltage regulator will be embedded directly into the PCB design of the control system, hence it will be housed in the same unit and location as the Raspberry Pi Module. The intended module that will be implemented into the design can accept a voltage between 4.5 to 28 volts with a stable 5 volt output constant. The expected maximum current by this unit is 3 Ampere which is the exact specification needed for the Raspberry Pi Compute Module at maximum usage as mentioned in previous sections.

6.1.3 GPIO Pins and Header

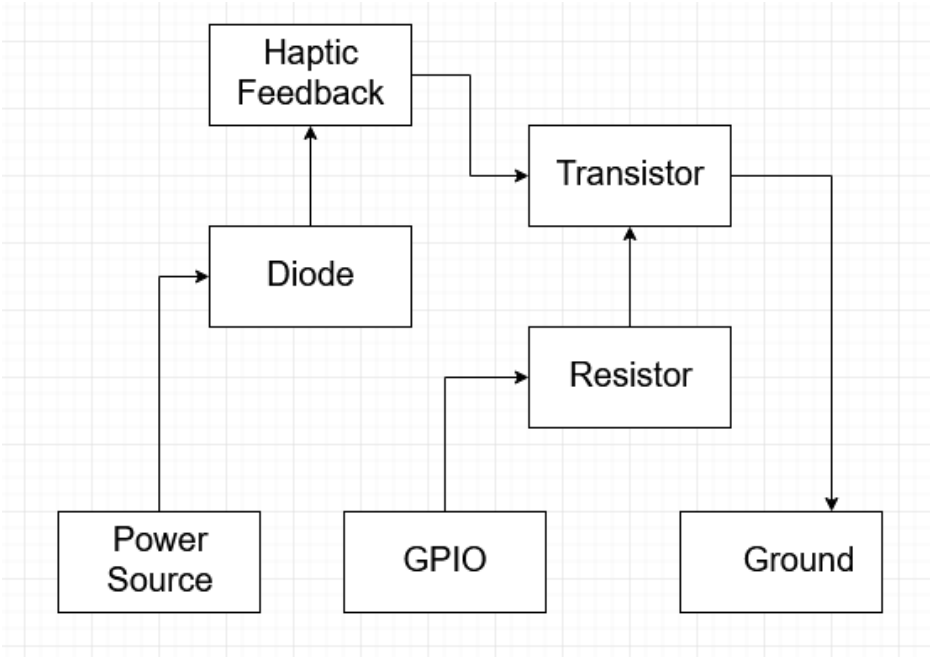


Figure 6.1.3 Haptic Feedback GPIO

Multiple GPIO pins will be connected directly from the compute module to a pin header. Each pin header will act as a control unit to a specific haptic feedback motor arranged with a transistor to allow for a variety of voltage sources into the design including the source voltage directly from the battery; to the 5 volt output source by the Raspberry Pi Compute Module itself. To further elaborate, the transistor used in junction with the GPIO will use the base current to turn on and off the voltage source. As previously elaborated, this prevents burnout from occurring in the control system. Diodes applied to prevent voltage from motor movement affecting pcb. In addition, the GPIO pins will also connect the accelerometer to the Raspberry Pi Compute Module both receiving and sending coordinate (x, y, z) information directly through the use of a GPIO UART connection.

6.1.4 HDMI Design

Use of a HDMI within the design will be used for prototyping and may be used in the final design depending on the development process. This device must be powered directly by the 5 volt source of the voltage regulator. There are a total of 19 pins in the HDMI pin layout with a total of 12 pins connected directly to the High Speed Serial for transmission of data.

6.1.5 Compute Module and High Speed Serial

The primary computation unit is the Raspberry Pi Compute Module. This is a small board hosting the CPU and GPU that will be connected to the carrier board and utilized in junction with peripherals. As the host unit, this portion of the PCB sends and receives data directly to the majority of the device with the exception of the devices connected to high speed serial utilizing

GPU usage. For further information refer to Figure 7.1.1 as to the connection scheme of the CM4.

6.1.6 USB-B communication

Universal Serial Bus Type-B (USB-B) will be the most common type of communication between the developer and user during the design phase of the device. When in the final stages of development a USB-B will still be implemented due to the following reasons: firmware updates, uploading of software upon construction. Due to the limitations of the compute module, the device will be limited to one USB-B due to express lanes being limited to one.

6.1.7 Raspberry Pi Camera

Camera compatibility with MIPI CSI-2 camera. The Raspberry Pi Camera uses a compact 22 pin layout that connects to the high speed serial with an input voltage of 3.3 volts. There is a 15 pin layout version, but 22 pin layout is more compact in the design factor according to dimensions.

Pin Name	**Pin Number**	**Description**
GND	1	Ground
CAM_D0_N	2	MIPI Data Lane 0 Negative
CAM_D0_P	3	MIPI Data Lane 0 Positive
GND	4	Ground
CAM_D1_N	5	MIPI Data Lane 1 Negative
CAM_D1_P	6	MIPI Data Lane 1 Positive
GND	7	Ground
CAM_CK_N	8	MIPI Clock Lane Negative
CAM_CK_P	9	MIPI Clock Lane Positive
GND	10	Ground
CAM_D2_N	11	MIPI Data Lane 2 Negative
CAM_D2_P	12	MIPI Data Lane 2 Positive
GND	13	Ground
CAM_D3_N	14	MIPI Data Lane 3 Negative
CAM_D3_P	15	MIPI Data Lane 3 Positive

Pin Name	Pin Number	Description
GND	16	Ground
CAM_IO0	17	Power Enable
CAM_IO1	18	LED Indicator
GND	19	Ground
CAM_SCL	20	I2C SCL
CAM_SDA	21	I2C SDA
CAM_3V3	22	3.3V Power Output

Table 6.1.7 MIPI CSI-2 Pinout

These pins connect to the high speed serial through a special set of header pins (*Arducam, 2021*). There is no need for driver installation as it properly communicates to raspberry pi on a rudimentary level. Camera uses CSI-3 which is a bidirectional protocol typically used in applications of pictures and video; specification by Mobile Industry Processor Interface (MIPI) Alliance.

6.1.8 Accelerometer

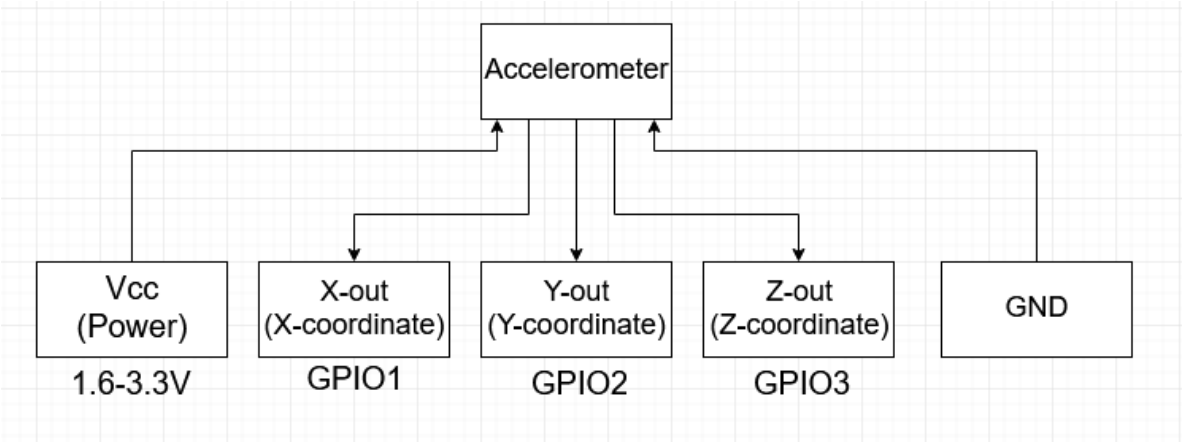


Figure 6.1.8 Accelerometer Diagram

The accelerometer utilizes a total of five pins: Vin, Ground, Xout, Yout, and Zout. In summary, the Vin and ground act as power to the accelerometer whereas the remaining pins output a value in accordance to the x, y, and z-direction respectively. Each of these directional pins uses an analog in from the control system to send information about the axis of the accelerometer as well as movements.

6.2 First Subsystem, Breadboard Test, and Schematics

6.2.1 Haptic Feedback Output

There are two methods of connecting the control system to the output haptic motors. The first method includes connecting GPIO output directly to each haptic motor. The benefits of this approach included a more straightforward design and decrease in redundancy due to less components. The potential issues of this approach include a possibility of burnout of the control system by the constant voltage output needed by the haptic feedback and limitation of a 3.3 volt source. The second method was to design a transistor circuit by using a NPN transistor to turn on and off a 5 volt source from the control system, with each individual GPIO pin being the base current. This allows for control of the high voltages without compromising on the integrity of the control system. The use of a diode will further prevent any current or voltage from interacting with the control system as a voltage may randomly be created from the haptic feedback motor's motion. For the purposes of this design, the second method was used.

Components for testing have been arriving in the mail and they are now available for current breadboard testing. One of the components to come in is the takoto coin motors. The takoto coin motors are a very integral part of the project. These motors are the primary way the system will be interacting with the user. It is incredibly important that each motor is tested in order to see if they operate properly creating a vibration that the user can detect. If any are defective, they will be reordered. Looking at the testing below, the design will be using two of the coin motors connected in parallel to create a more noticeable vibration for the user to notice. A diode is then connected in series for protection of the components attached. In the testing portion, a potentiometer was used to estimate the value of the resistor required for proper operation of the coin motors.

Theis actual system will be connected to a voltage regulator that will step down 5V to 3V, so for the testing, a 3V power supply was attached to this circuit and the resistance was varied to find an ideal value. The max resistor value that would allow operation was about 15 ohms. Varying the potentiometer, little to no resistance would be best for operation. Testing at 0, 4.7 10, and 14 ohms the best results came with the 0 to 4.7 ohm range.

When putting together this design, we will be placing two motors in each location. This is to attempt to make the vibration stronger and more obvious to the user. When a second motor is added, we will be able to cover more surface area of the jacket, which will also assist the user being able to sense what region is going off when. In testing is when the effectiveness will be tested the most, in which case more motors may be required, and the design slightly altered.

In the actual design, This setup will probably be connected to a motor driver. This should effectively regulate the voltage and make sure each motor will individually activate when required. Again, this design may become slightly altered when implemented.

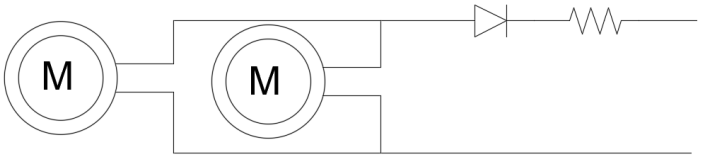


Figure 6.2.1(a) Motor testing Schematic

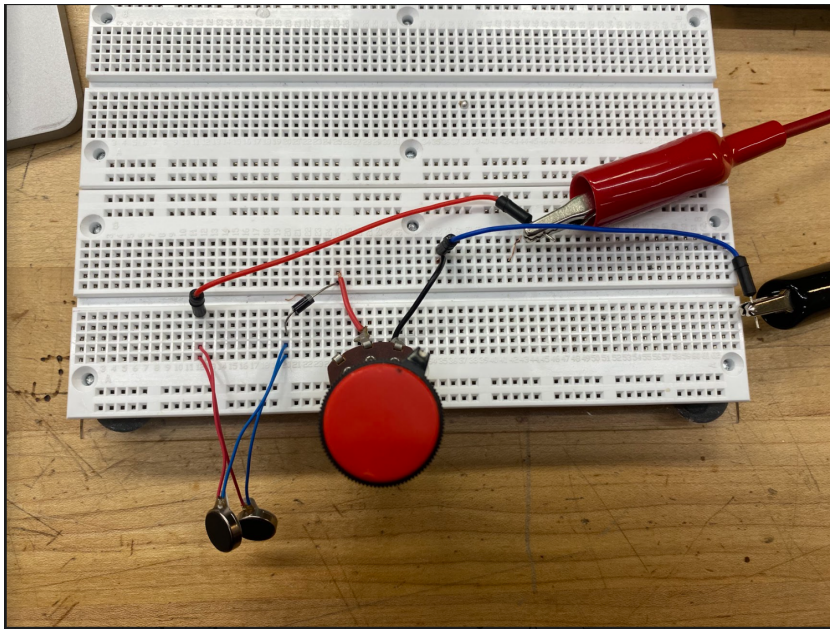


Figure 6.2.1(b) Motor testing

Further testing will determine if a 5 volt source will be enough for the haptic feedback in cases of split current when multiple motors are in use. A high valued resistor is used between the GPIO pin and the transistor base current to protect the integrity of the control system to prevent burnout from any reverse voltage that may occur within the system.

Name	**Emitter Voltage - VEBO (Vdc)**	**Base Voltage - VCBO (Vdc)**	**Collector Voltage - VCEO (Vdc)**	**Current (mAdc)**	**Temperature (C)**	**Dissipation (W)**
2N3903, 2N3904	6	60	40	200	-55 to +150	1.5

Table 6.2.1(a) Maximum Transistor Specifications

A general purpose transistor will be used to increase reliability in the system. The following table is the maximum rating specification for 2N3903.

There are multiple options in selecting diodes including 1N4148, 1N914, and 1N4001. The breadboarding will use the readily available 1N4148 diode as it is regularly used in devices that require fast switching. The following table shows the diode specifications.

Name	**Voltage (Vdc)**	**Current (Adc)**	**Temperature (C)**
1N4148	100	2	-65 to 150

Table 6.2.1(b) Maximum Diode Specifications

The purpose of the breadboard design is to test the amount of vibration within a single motor per volt. In this situation, a resistor may be added between the transistor and motor to decrease the amount of haptic feedback provided. In cases of needing more vibration, the 5 volt source may be changed to be connected directly to the battery source before the voltage regulator allowing for 9 to 11 volts.

6.2.2 Voltage Regulator

The testing of this component will be compared to a pre-existing voltage regulator that is typically used for the arduino.

6.2.3 User interface

The user interface will consist of the power buttons and the motors. The buttons will follow a simple design that will be installed on the outside of the jacket. The exploration of possibly making these operational buttons appear like jacket buttons will be explored. When a button is pressed, the user will hear the speaker say the operation that was pressed. For example, when the user presses the power button, the system will say “power on”or “power off” to let the user know the status of the device, as well as the battery percentage at the start of the use. Once the battery begins to run low, the device will alert the user when it has reached 20% battery life. Adding a button for a low power mode has been discussed as an option as well. If this were to be added in, the speaker would also let the user know what mode the device is in. Switching between full power and low power mode.

There are many options for button configurations on the jacket. With the estimated placement of the current components, the most reasonable placement of the buttons is the bottom right corner. There is plenty of room to place as many buttons as necessary, and there are 5 pins available for buttons. At minimum an on-off switch is necessary. If the low power mode is added then we will need two. The buttons that will be used are simple mini push buttons that will be sewn into the jacket. They will have 3D printed button caps to make them easy to find and press for the user.

The buttons will have some way to differentiate between all five buttons. The images for that can be seen below.

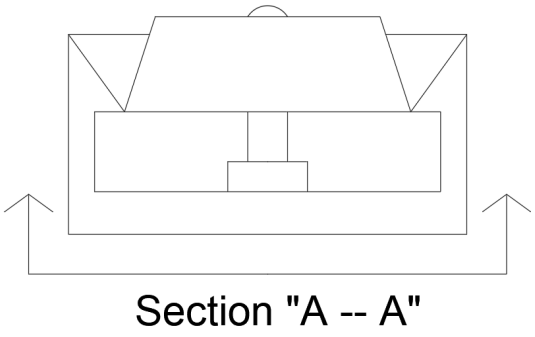


Figure 6.2.3(a) Button Cross-section

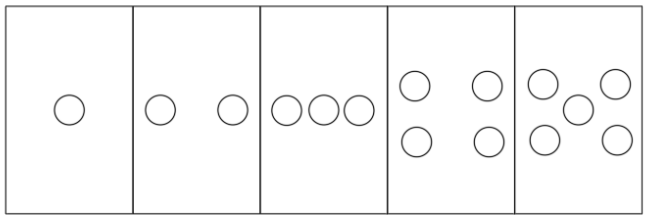


Figure 6.2.3(b) Button Domino Pattern

6.3 Second Subsystem

The second subsystems involve the sensors that will sense the world around the user. For this project the sensors required will need to detect objects, determine the distance from the object and how fast objects are coming at the user. The sensors that will be used are a camera for detecting objects, LiDAR for measuring distances of objects, and the accelerometer to detect any objects that are rapidly approaching.

6.3.1 Sources

The second subsystem will be connected directly to the Raspberry pi compute module through the two pinouts including (1) GPIO, and (2) high speed serial. The power source of each component is different which ranges from 1.8 volts to 3.3 volts. For the purpose of the overview the 1.8 volt source will be ignored.

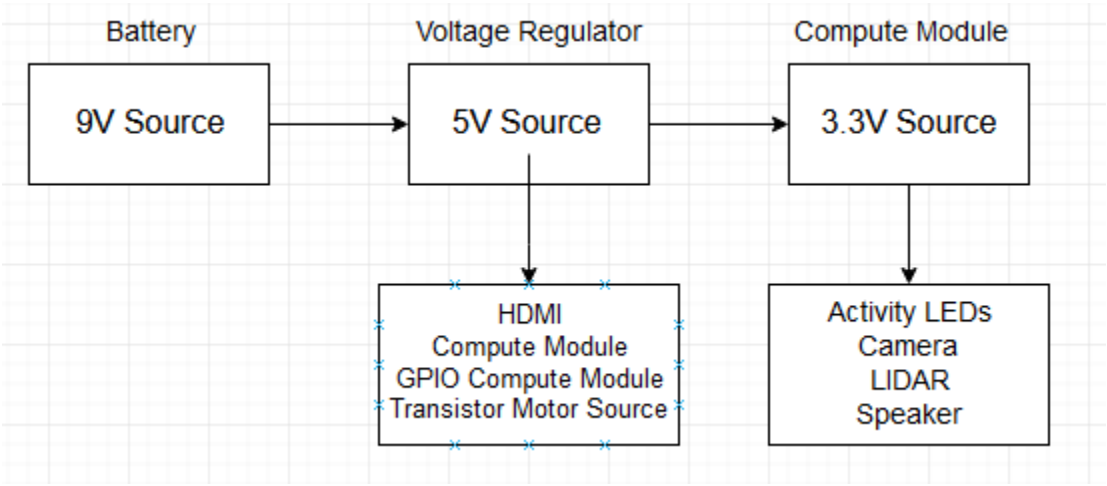


Figure 6.3.1 Voltage Source to Component

6.3.2 GPIO Connections

The GPIO pinout from the compute module controls the components including a total of 45 GPIO pins all programmable for specific needs. First the accelerometer (ADXL345) has the compatibility with languages Inter-Integrated Circuit Protocol (I2C) and Serial Peripheral Interface (SPI). Due to the limiting constraints involving the number of available I2C ports, SPI was chosen instead. As the device is connected directly to the Compute Module (CM4) through the use of a total of 4 pins; it uses the largest amount of GPIO. One of the important design decisions involving this accelerometer is its power saving method in reducing the data rate from 400 Hz to 12.5Hz. This is done by setting the device to low power mode through the use of the LOW_POWER bit (Channel 4).

LiDAR is another peripheral implemented and computed into the DPHS system through the use of two GPIO pins. These pins include I2C which allows it to keep a low pin/signal count.

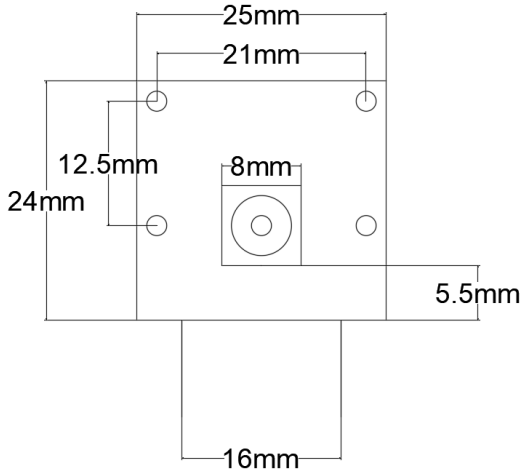


Figure 6.3.2(a) Camera Board

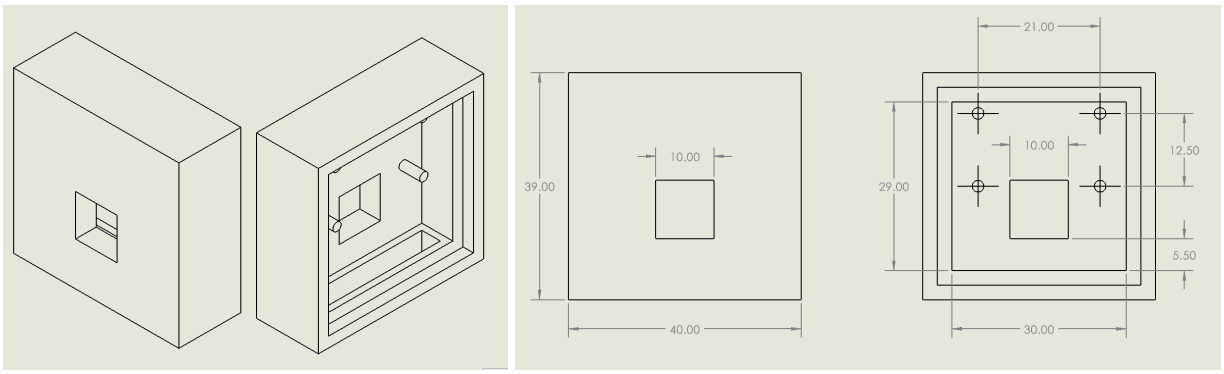


Figure 6.3.2(b) Camera Case

The final component is the Raspberry Pi Camera connected both to the GPIO pin header and the high speed serial. The GPIO pins control . The camera has an IR filter that ranges from 350 nm to 550 nm at the highest transmittance. The pins for this device connect directly to a seperate 15 pin header but are still wired directly to the aforementioned headers outputted by the CM4. Due to the component having a wide hole in the back all of the cables will be routed from this direction and directly into the suit. The camera will be interfaced to the board via a built-in camera serial interface 2(CSI-2) on the compute module. It will be attached by a ribbon cable where it will send and receive signals from the compute module.

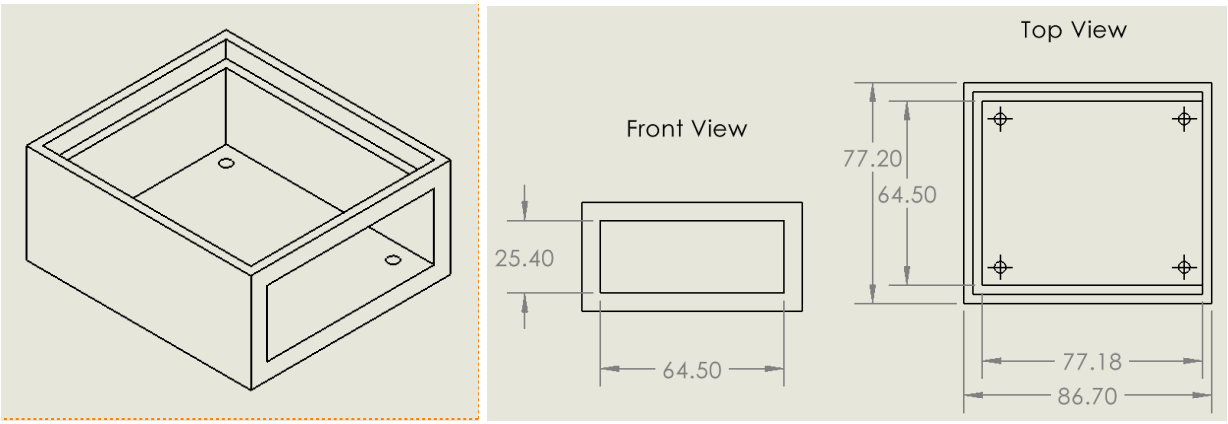


Figure 6.3.2(c) PCB Case Box

Each wire will be routed from the compute module carrier board through the boxed chamber on the side window. An illustration of the boxed chamber can be found in *Figure 6.3.2(c)*. An insulated shroud of plastic will be used to prevent moisture from entering and destroying the PCB itself.

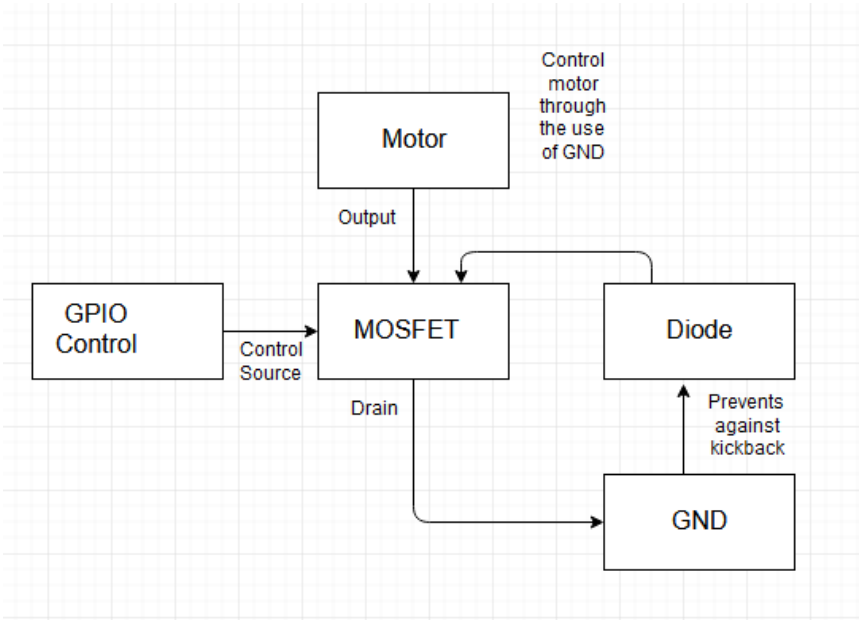


Figure 6.3.2(d) MOSFET Connection

The outputs of the PCB will be from the 40 count pin header and output a total of 9 PWM connections to control motors of which 5 are original signals from the compute module, 2 are analog converted to PWM, and the remaining are from conversion from digital to analog. The motors will have a diode and capacitor connected directly to help in controlling a sudden change in voltage that may burn the motor but also give a similar effect to a motor slowing down before stopping instead of coming directly to a halt when power is cut from the GPIO.

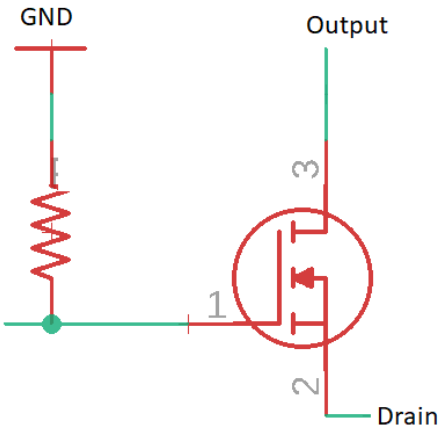


Figure 6.3.2(e) MOSFET Schematic

A MOSFET is used in the design for each motor to allow for the use of a separate voltage of choice including a 9-11 volt battery source, 5 volt source from regulator, and 3.3 volt source from the compute module itself. The testing and effect of the motors will be done during the testing phase of the design to determine the right voltage for the haptic feedback.

6.4 Software Design

This section will describe the software development process, architecture, design, and implementation.

Software Development Requirements Outline

- Product purpose:
  - Device should effectively be able to guide an unsighted person.
  - Project is overseen by Senior Design Professors.
  - Project is developed by Group 40: Software developers and Electrical Engineers
- Scope of Project:
  - Software solution involves convolutional neural network and image processing, real-time systems, embedded systems
- Product Overview
  - Vest/Jacket fitted with approximately 9 vibration units and a speaker that accesses camera footage, LiDAR sensing, and acceleration from an accelerometer to process and identify approaching obstacles.
- System Requirements:
  - Must implement safety features to accommodate for the inaccuracies in CNN approximation
  - Must implement an assortment of modes which allow the user to navigate in a general and outdoor environment
  - Managing power consumption of internal processes and peripherals
- System Features
  - Features will include a total of three peripherals as input devices and 2 methods of output involving speaker and haptic feedback. Varying modes will be utilized to navigate certain areas and environments.

6.4.1 Development Environment

Software development will proceed using the agile lifecycle. Significant time will be spent on specifying the requirements of the system, the architecture intended to meet these requirements, and considering how the architecture may be implemented. After this phase, developers will be assigned to work on each component. In this section, a component is considered to be a piece of software whose internal functioning does not directly affect its interaction with other components. These components will be tested via software, and periodically via hardware during development.

Development Standard
To improve the extensibility, maintainability, and reusability of the code, a consistent coding standard will be required. Coding standards already exist for programming languages such as Python (*Rossum, et. al., 2013*) and C++ (*Stroustrup and Sutter, 2021*). One or more of these standards may be implemented depending on the software requirements and the implementation of the architecture for the Pi. These standards will be used as a reference, rather than a binding set of rules. If an aspect of the code is not covered in the list below, then developers will make a

best effort attempt at following the standard. No guarantees are made about the use of any standard except for what is listed below.

- Document Structure
    - Tabs for line indentation
    - All source code files will be terminated with at least one new-line character
    - Lines should be no longer than 150 characters
- Comment Structure
    - Only single line comments may be used
    - Comments that occupy their own line must be at the same level of indentation as the code around it
    - Comments placed to the right of lines of code must be indented with tabs, preferably to the same level as surrounding comments
- Function Structure
    - Opening and closing curly brackets of function definitions must be on their own line with the same level of indentation as its declaration
    - Function contents must be at an additional level of indentation from the declaration for all elements in the definition
    - Functions will be delineated with two new-line characters (i.e., one empty line)
    - All functions will have unambiguous names which describe their functioning
    - Function parameters will have unambiguous names except in the case that the function is a utility (e.g., minimum of the parameters a and b)
- Variable Structure
    - All variables must have names which describe the content that it contains except in the case of being contained within a utility function

Development Workflow
Software development will be done on each developer's personal computer. An integrated development environment (IDE), such as but not limited to Visual Studio Code, will be used to write the programs, test cases, and run the test cases to verify the functionality of each component.

Github will be used in order to contribute to the software in parallel. Releases will be generated using semantic versioning as a guide (Preston-Werner). See the software development milestones below for a schedule of releases. In general, each major release at a minimum will contain a set of changes to the structure of the system which impacts multiple components. For example, a function definition replaced in an API that is consumed by other components would qualify to be put in a major release. There should be multiple of such changes for major releases. Minor releases should only make changes within the component, which do not affect the structure of its relationship with other components.

The Gitflow workflow (“Gitflow”) will be used in addition to semantic versioning. Code will be reviewed in detail by another developer before being merged into the development branch. This review will be both for correctness, and for adherence to the coding and commenting standards described. Commit history will be squashed such that only one commit will be issued per merge into the development branch. Merging to the master branch and creating releases will be the

responsibility of the owner of the repository. All developers must review and test the code to the best of their abilities before major and minor releases.

Documentation for the software components will be maintained to allow a developer to quickly reference the work done by others as needed. Each directory within the project will contain a markdown file with information that is relevant to other software components. Each file must contain a heading which matches the source code file which it describes, and subheadings which logically separate classes of functions or variables (e.g., functions pertaining to the camera, versus those pertaining to the accelerometer) in alphabetical order. Each entry will be a publicly available function definition followed by a description if necessary. No requirement will be set for the correctness or completion of documentation, but a best effort will be made to produce and maintain it. These documents will be available on the public git repository.

Development Metrics

To ensure an evenly distributed workload among developers, and to measure software development progress and efficiency, a set of simple metrics will be adopted. Meeting the requirements on each metric will be done at best effort. Progress in each of the criteria in Table 6.4.1 will be measured with the specified metric, aiming for the specified value.

Criterion	**Metric**	**Value**	**Unit**	**Error (±)**
Component	Count of discrete software components	24	count	4
Commenting	Percentage of lines in a source code file which contain comments	≥ 40	%	10
Dependency	Number of classes or libraries included in a file	≤ 7	count	2
Development	Line count of all source code including comments and line breaks	5,000	count	1,000
Modularity	Line count of a single source code file	≤ 250	count	50

Table 6.4.1 Software Development Progress Metrics

Value for component criterion estimated using Figure 6.4.1(a), estimating 3 input devices (e.g., two LiDAR units, camera), 3 output devices (e.g., DACs connecting to motors) with 1 additional unit representing a speaker, estimating 3 components per remaining indexed blocks (e.g., service), then adding an additional 1 for every wrapper and remaining block. Value for development criterion estimated similarly with 250 lines of code for 20 components, assuming that 4 components will have a line count significantly less than 100.

Testing Development Software

Testing will be a necessary part of the development workflow, but the hardware may not always be ready or available for it. In addition to thorough review of the code, a set of simulations (test

cases) must be written for every non-trivial (i.e., no getter or setter) function in a component. These test cases should be written and evaluated by one developer, but also evaluated and extended if needed by another. Test cases are meant to eliminate or reduce the number of errors that occur when testing on the hardware. This is essential to prevent damage of the hardware or any of its components, as well as to reduce the amount of time spent debugging during hardware testing days. See the software development milestones for a schedule of hardware tests. Test cases will not be published. In general, hardware testing will be conducted on the Raspberry Pi 4 Model B before it is tested on the CM4.

6.4.2 Defining the Software

It is essential to thoroughly define the boundary of the software, its intended functioning within the environment, the user perspective of the device, and the software requirements. Below each topic will be explored as it pertains to the architecture and design of the software.

Software Boundary
The software in its most general sense translates information about the environment and the state of the system, to its sensory outputs. This relationship is moderated by user input (e.g., selecting a mode). This boundary is depicted in Figure 6.4.1(a). The flow of information is indicated by the direction of the arrows. Solid arrows represent a continuous flow of information (e.g., camera feed), whereas dashed arrows represent discrete events (e.g., button press).

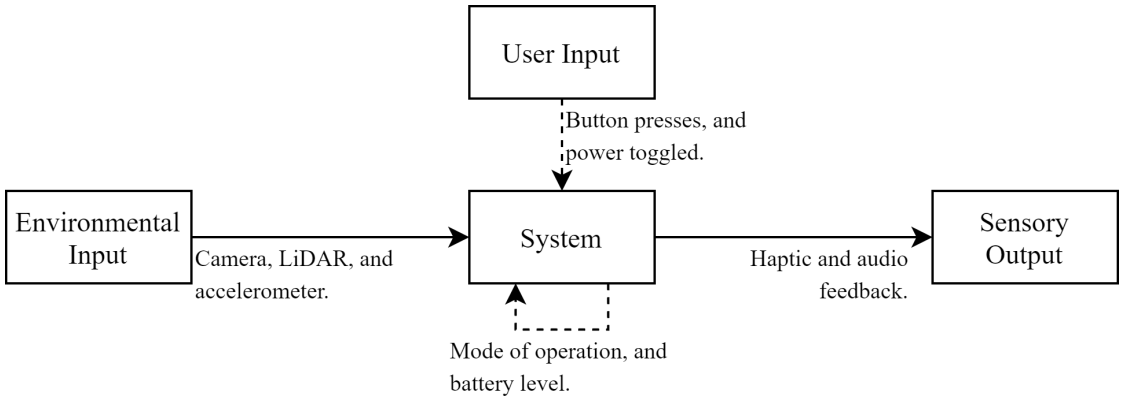


Figure 6.4.2(a) System Boundary

Events and System Responses
Figure 6.4.1(a) provides a convenient way to further analyze the requirements of the system software. In particular, the system state and output response may be described with respect to the arrows (events) that enter the system. The table below presents a non-exhaustive list of events which may occur from the user, environment, or the system itself. Output response and system state will depend on the final hardware and software implementation, but are provided here as examples of what should occur.

No.	Description	Output Response	System State
1.X	User Input		
1.1	Powered on (Button 1 Pressed)	LED enabled Speaker announcement Haptic feedback pattern	CPU power on Program initialized
1.2	Powered off (Button 1 Pressed)	LED disabled Speaker announcement Haptic feedback pattern	Program deactivated CPU power off
1.3	Button 3 Pressed	Haptic feedback pattern (e.g., long followed by short vibration)	General Navigation Mode
1.4	Button 4 Pressed	Haptic feedback pattern (e.g., long followed by two short vibrations)	Outdoor Navigation Mode
1.5	Button 5 Pressed	Haptic feedback pattern (e.g., long followed by three short vibrations)	Low Power Mode
1.6	Button 2 Pressed	Speaker announcement	Recalibrate sensors
2.X	System State		
2.1	Battery < 20%	Speaker announcement Haptic feedback pattern	User idle time before entering low power mode decreased
2.2	Battery < 5%	Speaker announcement Haptic feedback pattern	User idle time before entering low power mode decreased
2.3	General Mode	Speaker announcement	Depth perception service is active
2.4	Outdoor Mode	Speaker announcement	General mode is active in addition to curb detection
2.5	Low Power Mode	Speaker announcement Haptic feedback pattern	Only emergency services active

No.	Description	Output Response	System State
3.X	Environmental Input		
3.1	Object in view of camera	Haptic feedback with intensity and location corresponding to object	
3.2	User stationary for extended time	Speaker announcement Haptic feedback pattern	Enter low power mode
3.4	Object directly in front and near	Haptic feedback pattern	All low priority services suspended while active
3.5	Step-down Curb near	Haptic feedback pattern	
3.6	Step-up Curb near	Haptic feedback pattern	

Table 6.4.2(a) Event Descriptions and Responses

Flow of Information and Subsystems

The flows of information from Figure 6.4.2(a) can also be analyzed. Consider the set of three-tuple vertices where each block in the figure is a vertex in a digraph. The first vertex indicates where the information is retrieved, the second where the information is processed, and finally where the processed information is utilized. There are six flows which follow these criteria, which are provided in Table 6.4.2(b). One may also consider each tuple to represent a software subsystem. The notation is that the subsystem Sys_X is a function which maps an input space (e.g., environment input [EI]) to an output space (e.g., haptic or audio feedback [Out]) and is denoted as Sys_X: EI → Out.

Some of the flows listed in the table are coupled. Consider the tuples starting with UI. Pressing a button not only triggers a response from the speaker, but it also eventually leads to the announcing of the mode that has been activated. This coupling can also be seen in Table 6.4.2(a), for example in events 1.3 and 2.3. Additionally, with this subset of tuples the system state and output state imply each other. User input events must always generate a change in system state and a subsequent output response, and likewise output responses are always followed by a change in state. This means that the systems may be combined to form Sys_{CD}: UI → (Sys, Out). This coupling also exists for the tuples starting with Sys. This subsystem may be represented as Sys_{EF}: Sys → (Sys, Out). This coupling is not obviously observed in the tuples starting with EI. For example, the CNN always produces an output, but does not need to affect the state of the system. Similarly one could conceive of a use case where some input changes the system state without producing an output. Using the two coupled subsystems as an example, Sys_A may be rewritten as Sys_A: EI → (Sys, 0), and Sys_B may be rewritten as Sys_B: (0, Out). Combining these into one Sys_{AB} allows for support of both individually, as well as use cases which may affect both system state and the output.

Information Flow	**Subsystem**	**Example**
EI → Sys → Sys	Sys_A: EI → Sys	Activating low power mode when velocity is less than a threshold value for a set amount of time.
EI → Sys → Out	Sys_B: EI → Out	Setting haptic feedback level in response to depth estimated by neural network.
UI → Sys → Sys	Sys_C: UI → Sys	Button push to activate the General mode.
UI → Sys → Out	Sys_D: UI → Out	Playing a sound when a button is pushed and mode activated.
Sys → Sys → Sys	Sys_E: Sys → Sys	Setting the system to low power mode in response to critically low battery.
Sys → Sys → Out	Sys_F: Sys → Out	Announcing low battery level and system mode.
Where EI is Environmental Input, UI is User Input, Sys is System, and Out is Output		

Table 6.4.2(b) Information Flows

Further decomposing the flows in Table 6.4.2(b), it is observed that ‘Sys’ is depicted as both an input, and an output to the subsystems. Clearly defining system-as-an-input versus system-as-an-output will allow for further clarification of the subsystems. The system-as-an-output, as shown in the examples in the table, always pertains to a mode. The system-as-an-input always pertains to discrete events in the system such as battery level dropping below a certain value. Assuming that these discrete system events are exclusively available via the hardware input interfaces, these may be logically combined with the sensor/environmental input for a general input. This means that Sys_{AB} and Sys_{EF} may be combined to form Sys_{ABEF}: I → (Sys, Out) where ‘I’ is now input pertaining to the system and the sensors. This is summarized in Table 6.4.2 below.

Name	Subsystem	Description
Input	Sys_{ABEF}: I → (Sys, Out)	Pertains to sensors and hardware systems that interface with the processor unit's input interfaces. Data retrieved from these interfaces may result in a change in the system mode and/or produce an output response. Examples include enabling a low power mode when battery is low, enabling modes when velocity is within some threshold, depth perception, and more.
User Input	Sys_{CD}: Sys → (Sys, Out)	Pertains to user interface, or in general digital input events which require an immediate response from the system. No data is passed into the system, rather discrete events which occur at the GPIO trigger interrupts which produce a change in the system mode and/or an output response.
Where UI is User Input, I is all other Input, Sys is System State, and Out is Output		

Table 6.4.2(c) Subsystem Descriptions

Use Cases

From the user perspective, only five buttons are available: Power toggle, sensor recalibration, and three mode buttons. Each button has associated user input events as listed in Table 6.4.2(a). Figure 6.4.2(b) presents this information as a set of use cases that the system must accommodate. The user experience will not only be determined by the functioning of each mode, but the response of the system to each user input event.

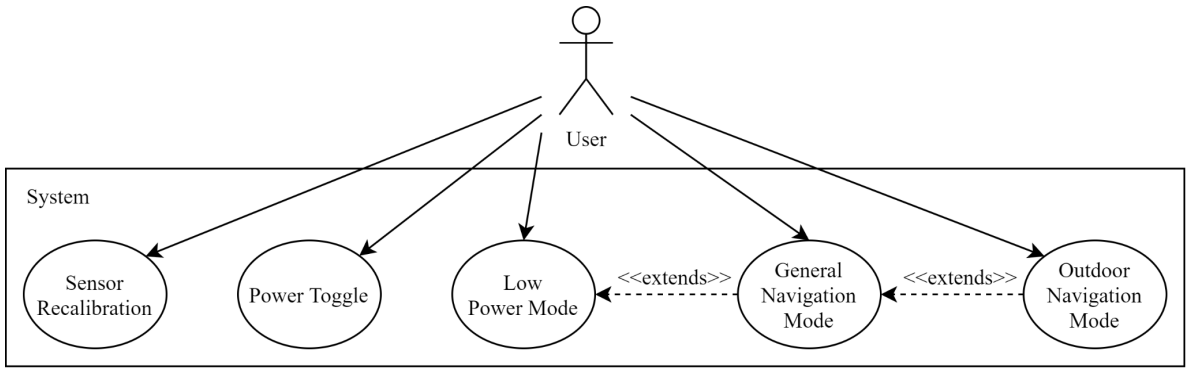


Figure 6.4.2(b) Use Case Diagram

Risk Analysis and Mitigation

The product is an assistive device that relies exclusively on the capabilities of the software. There is tremendous risk involved as severe injury or death may result from the failure of the

system to perform its duties quickly, accurately, and consistently. This section considers the dangers which may be relevant to an individual using the device, and the assumptions made by each mode in mitigating these risks.

An individual navigating through some generic environment may encounter three types of obstacles: Terrain and debris, path obstructions, and overhead obstructions. Terrain and debris occur around or below the legs. This terrain may be uneven, loose, or flat. It may be rough, slippery, liquid, or even a sheer dropoff or incline. It may be covered in garbage, marbles, or wire. Path obstructions are obstacles which occur at the torso level. This may include people, lamp posts, trees, shrubbery, road signs, and more. Overhead obstructions are items such as roofs, hanging signs, and light fixtures. In all cases, the user may be hurt by being unable to identify these obstacles. Additionally, actors within the environment may act as any of these obstacles. A person may cut in front of the individual, a car may speed into them, or perhaps terrain itself may shift such as on an escalator.

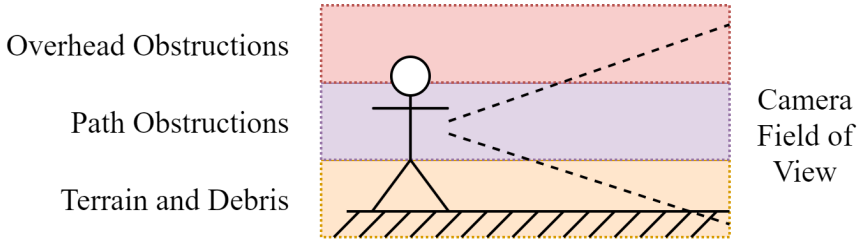


Figure 6.4.2(c) Types of Obstacles and Camera Field of View

An additional complexity is found when considering the camera field of view. Figure 6.4.2(c) visually depicts the regions of the environment described in the previous paragraph. The limits of the field of view of the camera are depicted with dashed lines. Both the terrain and overhead obstructed regions near the user are not within this field. At best, the system may be able to estimate the location of objects in those regions based on previous images and the user speed. Even in that case, unexpected events (e.g., a dog running toward the legs) will continue to pose a danger.

The General Navigation mode is the most basic mode. It only handles transforming images into a depth map, and representing that map with haptic feedback. The purpose of the mode is to allow the user to walk through a generic indoor or outdoor area. This mode will assume that the terrain is smooth, rough, and does not contain debris of any kind. The terrain is continuous, neither having rapid inclines nor rapid declines. It will also assume that there are no overhead obstructions. The only obstacles are objects that may obstruct the path of the user which are around torso level. These obstacles are additionally assumed to be opaque. Responses to these obstacles will be covered below.

The Outdoor Navigation mode operates using the General mode as its base. Ideally, this mode will implement computer vision models for urban environments which detect road signs, pedestrian crossings, and approaching vehicles. The present mode, however, will only remove the assumption of no rapid inclines or declines. This will be done with a LiDAR module angled toward the floor to register changes in elevation. Measurements of elevation are assumed to

change only along the direction of user motion. Points perpendicular to the direction of the motion at the point measured by the LiDAR are assumed to be equal. This will allow the system to detect and alert users to traversable inclines and declines (e.g., curbs) versus non-traversable (e.g., cliffs, manholes). This will not detect drop-offs into bodies of water such as those located at the edge of a dock.

The Low Power mode is unique in that the only assumptions made are that the user is stationary, unless the mode is manually activated. In either case, only the emergency systems (i.e., forward facing LiDAR) are active within the CPU. Clock frequency and power to peripherals may be cut while in this mode. If the user continues to navigate after having manually activated this mode, they assume the risks involved. However when this mode is activated automatically after the user has remained idle for a specified amount of time, the risk that objects may approach the user from any direction other than directly in front of them and being unable to warn them is assumed.

Summary of System Needs
There are two subsystems that must be implemented. A hardware interrupt system must exist to register user input, and execute a routine of system state modifications, and output responses. Another subsystem must retrieve information from sensors and execute yet another routine. The overarching theme for these subsystems is the need to affect the output and/or system states, the potential for each subsystem to consume input, and the potential for a subsystem process to be performed in response to a user input or system event.

6.4.3 Software Architecture

The software architecture is depicted in Figure 6.4.3(a). Input devices will be accessed via an input component which serves as the interface between the sensor hardware and the software. This interface will connect to device drivers and make data available to system services in a compact, human-readable, and efficient way. Services can be thought of as pipes which request information from the input, and filter it into some output response. To create an output response, each service will make a call to a routine. Routines are similar to services, but do not necessarily require input to function. The routine makes calls to an output component, which establishes functions for setting the state of devices such as the speaker and haptic feedback units. Routines may also interact with the system to produce a change in its state. Given the complexity and strict time requirements of the system, a dedicated scheduler will be needed to manage services. This scheduler may allocate services to individual cores in the CPU, or implement some algorithm to run processes concurrently over one or more cores. Hardware interrupts are events which occur at the GPIO which require immediate servicing. In this component, GPIO events are connected to their servicing routines. Although this component is only intended to serve the user interface, it is left general just in case interrupt support is requested for other hardware components. Finally, the system is the component which instantiates the environment as a whole, and contains information about the system state (i.e., mode).

Arrows between the components in the figure may be interpreted as an interface that the former component must supply for the latter. For example, the SystemRoutine component must implement some interface which will allow the hardware interrupt component to interact with it (e.g., a ‘GetRoutine’ method). Below the architecture of each component will be covered in

detail, alongside whatever interfaces the component should supply. It is important to note that some of the arrows in the diagram may be redundant. For example, although the routine components receive requests from both individual services and the hardware interrupt object, the routine may service both with a single 'ExecuteRoutine' method.

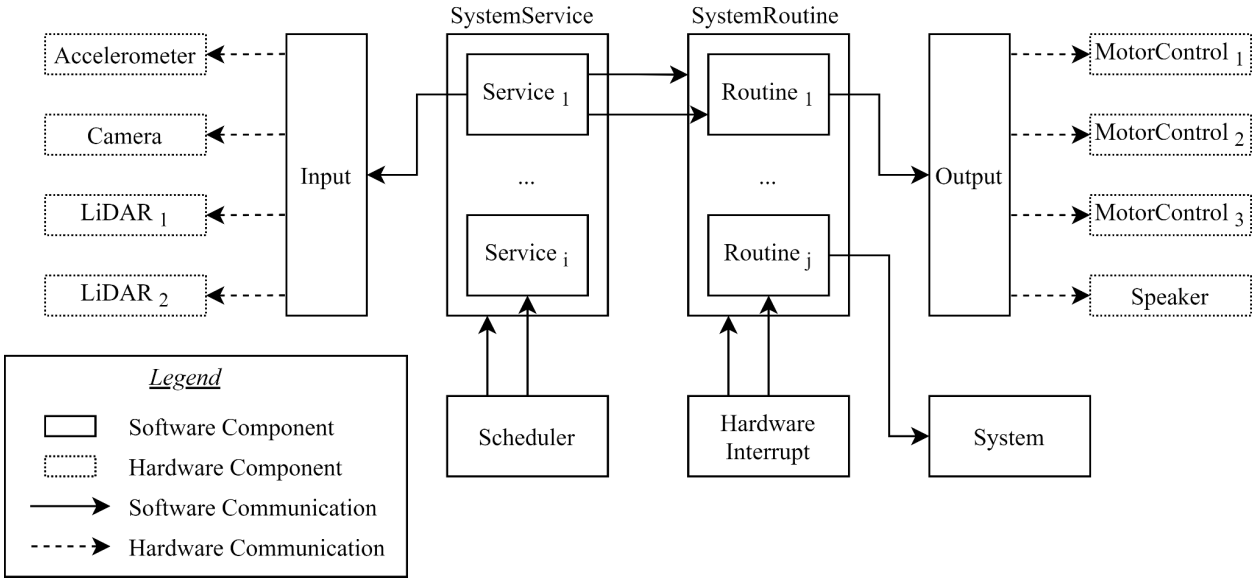


Figure 6.4.3(a) Software Architecture Overview

The sections below will describe the architecture of each software component in Figure 6.4.3(a) in detail, discuss any interfaces which may need to be implemented, and its relationship with other components.

System

Although depicted modestly in Figure 6.4.3(a) as a small component which receives requests from a routine, system actually houses the entire software environment. It instantiates all other major components (i.e., input, output, system service, system routine, scheduler, hardware interrupt), initializes them, and is responsible for system power on and recovery from crashes caused by software failures or hardware being underpowered. The system should maintain references to all major components such that other components may use the system object as a conduit to reference them.

Aside from the responsibilities above, the system also contains information on the state or mode of the system. It must provide enumerations such as 'GENERAL_MODE' and 'LOW_POWER_MODE', and maintain a variable which contains the current mode. Additionally, it must supply an interface which the routine may use to both poll and modify the state. Although not depicted in the figure, an interface should exist for the scheduler as well, which may have to modify its scheduling algorithm to account for the selected mode.

The system component may be thought of as the parent of the major components mentioned above, and the grandparent of each implemented service and routine. Only one instance of the

system will exist during the lifetime of the program, and in general only runs its code at startup and when routines request to modify the mode. Otherwise, most of the dynamics of the program are contained within the services and routines, and the scheduler. As such, the system component will use the master-slave architectural pattern: The system will serve as the master, and the scheduler will be called as its slave to carry out the primary logic.

Services and Routines

Conceptually, a service is a discrete unit of processing which when called by some object (i.e., the scheduler) may request some input from one or more sensors, process that input, and then produce an output response and/or change the system state. This is best modeled by the pipe and filter architecture, as sensed information is taken from the input, filtered over multiple stages, and transmitted to the output. For example, a DepthPerceptionService may transform an image into a depth map, which is then translated to haptic or audio feedback. The approach and example are depicted in Figure 6.4.3(b).

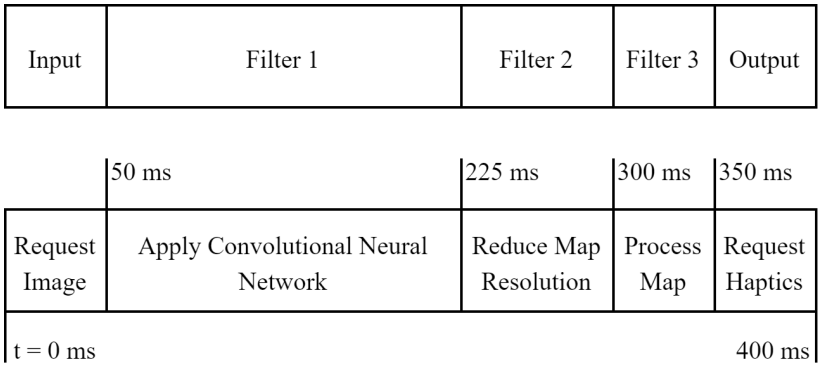


Figure 6.4.3(b) DepthPerceptionService with Pipe and Filter Architecture

In actuality, the services described above are broken into two portions: Services and routines. Services send requests for sensor data to the input component, and perform some initial processing of the information. The processed information may then be transferred to a routine which will use it to produce an output pattern and/or change the system state. Referring back to Figure 6.4.3(b), the actual service portion of the pipe may only go from 0 ms to 300 ms, while the filtering which remains is done by the routine. In general, every service should call a routine, but not every routine needs to be called by a service—this relationship is covered further in the hardware interrupt section below. This division separates code that is intended for input processing from code which produces output responses. This allows for the reuse of routines such as, for example, one that initializes a low power mode.

The service must supply an interface with which the scheduler may interact with. In particular, the filtering process defined within the service must be called by the scheduler. As such, services should contain a run method. When called by the scheduler, the run method should perform all of the tasks for the service. Services assume non-preemption. This means that assuming a service is occupying the CPU, and a GPIO event causes the service to stop early, the service can not resume from where it left off. It must start over. Depending on the scheduling algorithm

implemented, the interface may also require the service to contain an integer priority. This will be discussed further in the scheduler section below.

The routine must supply an interface to the hardware interrupt component. The considerations are similar to that of the service interface. Routines should contain a run method which may be called in response to a hardware interrupt. Priorities in the case of hardware interrupts will not be considered.

Figure 6.4.3(a) depicts a wrapper surrounding the set of all services. This wrapper object will be used to handle metadata associated with each service. For example, services may contain an integer priority level which determines which process the scheduler allocates more CPU time for. The wrapper may then implement a method for iterating through services at different priority levels. The wrapper may also manage the registration and retrieval of service objects by implementing a GetService method, rather than forcing the scheduler to include, instantiate, and track each service itself. Similarly, the routine wrapper should also contain a GetRoutine method and manage each lifecycle. These get methods will return a reference to the instance of the particular service or routine being requested. In general, get requests will only be made once per requesting component, as they should save the returned reference. For example, a particular service during construction should make a call to the GetRoutine method once and save this reference so that it no longer needs to refer to the wrapper.

Each wrapper is responsible for managing the lifecycle of the service or routine, which includes instantiation. Wrappers will contain a list which connects string or enumerated names to their class name. This list will contain one instance of the class per name. Each service is a child of the abstract service superclass, and likewise for routines.

Scheduler
The scheduler serves as the application manager. It is instantiated by the system component and controls which services are run when, and where. It is responsible for setting services up to be processed, establishing any multithreading or process generation, and managing schedule changes in response to changes in the system state.

No component consumes the scheduler, so it does not need to supply an interface. However, special consideration does need to be given to the type of scheduling algorithm implemented, and how services may be divided amongst cores. Additional functions may be required of some service interface to support the algorithm. Assuming that three of four cores are devoted to the most computationally expensive service (i.e., the DepthPerceptionService), round robin should be sufficient for all other services to run on the remaining core. Should this approach prove to not be sufficient, more advanced algorithms involving priority levels and queuing will be considered.

The scheduler may be thought of as a broker, which allocates services to the CPU in response to the system mode which serves as the client. The mode of the system should imply the schedule used by the scheduler for the available services.

Input Devices and API
As stated above, the software architecture will be configured from a system of input devices which will be accessed via an input API, which will connect to device drivers and make data available to the services, in this portion the configuration will be broken down.

The input system will consist of a camera angled directly forward, the raspberry pi camera 22, two LiDAR sensors, one angled directly forward and the second slightly angled downwards, and an accelerometer. To process the input from these devices, A service will communicate instructions for how the system will take in information from each of the input devices. The scheduler would run a service, in which all the tasks would be executed.

Input Devices			
No.	**Name**	**Description**	**Uses**
1.0	Camera	Captures Live feed	High Speed Serial HDMI
1.1	Forward LiDAR	Detects objects in front of user	I2C
1.2	Angled LiDAR	Determines Distance to floor to detect curbs and low objects	I2C
1.3	Accelerometer	Captures speed of individual to determine if they are stationary	SPI

Table 6.4.3(a) Device Description and Interface

Input Architectures
- Camera: Used to extract the general image of the environment before a user.
    - This camera input image is used to form a CNN where data pulled from that output layer is compared and used along with the output of the LiDAR sensors. Processing the images taken in from the camera will require something like a GetCameraImage() function in which a CNN is developed and an output 4D array is returned(More below). This method of working with CNNs follows a layered architecture *(Verma, 2019)*
    - One thing to note is the output layer can be changed to 2D using Image flattening.
- LiDAR: Actively emits light in the form of a rapidly firing laser, where the beams reflect off of objects and the reflected light energy is recorded in the LiDAR sensor. This remote sensing serves to detect objects like those detected by the camera imaging. For the main body of the functionality of the device with a DepthPerceptionService and an

EmergencyDectectionService, functions like GetForwardLiDAR() and GetAngledLiDAR(), are going to be needed *(Wasser, 2020)*

- Accelerometer: Used to determine the speed of the user to determine whether the wearer is idle or actively moving; this helps determine what state the system should operate in.

Services			
No.	**Potential Identifier**	**Description**	**Request to Interface**
1.0	**Input Interface**		
1.1	DepthPerceptionService	CNN and LiDAR sensor image and depth processing to identity obstacles in wearer's path	Accesses image from camera Accesses Depth from forward facing LiDAR
1.2	CurbDetectionService	Platform height difference detected ahead	Accesses depth from downward angled LiDAR
1.3	EmergencyDepthService	Sudden arrival of large or present obstacles detected directly in front of the individual. Any currently running service exited.	Accesses depth from forward facing LiDAR
1.4	Low Power Service	If individual is not moving for a certain threshold time, non-critical subsystems disabled	Accesses Accelerometer

Table 6.4.3(b) Services

Input Designs

- Camera: Camera Input will be gathered and processed through CNN. Camera image taken through Camera 22 as .jpg.

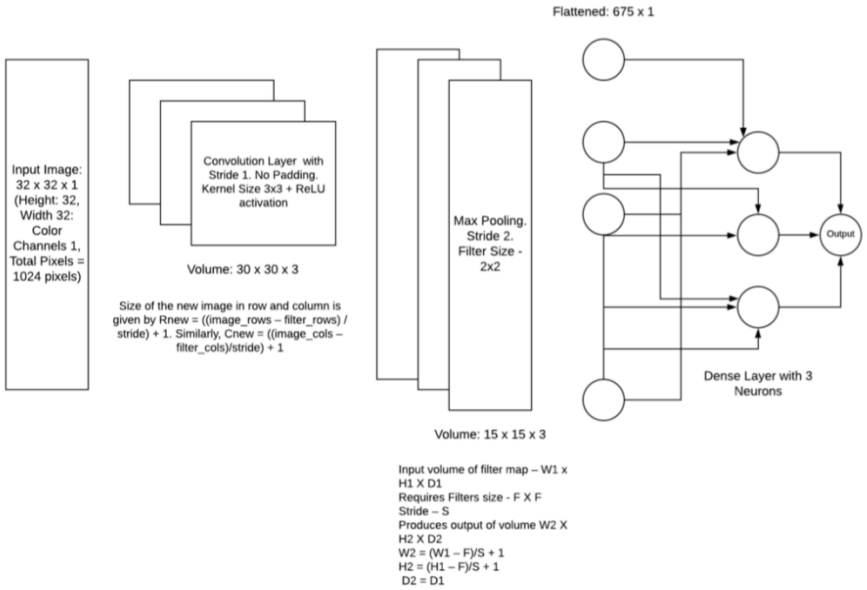


Figure 6.4.3(c) Camera Input (Das, 2020)

- LiDAR:The specification sheet for the Garmin LiDAR model, the LiDAR system purchased for this project, does not list the use of discrete return or full waveform feedback. It is believed the device is a discrete return because other 2D model systems use a similar function with a time step as the primary coordinate (x-axis) in projecting distance and a calculated distance (y-axis) because the x-axis is typically used for input values (*Das, 200*). Full waveform uses intensity as an output which is not typical in cheaper devices such as what is used in the project. Further testing will be made during component testing and adjustments will be made accordingly (*Mathworks, 2021*).

- Accelerometer: Supports SPI and I2C and won't require a driver as it can the raspberry pi can directly interpret the data. (Emmet, 2019)

The Input API is the component that will define the functions that the services will call in order to retrieve information from the input devices. As mentioned above, functions such as, GetCameraImage(), GetForwardLiDAR(), GetAngledLiDAR(),and GetAcceleration(), will handle grabbing data from the camera, 2 LiDARs, and accelerometer. Below see sequence of general input processing.

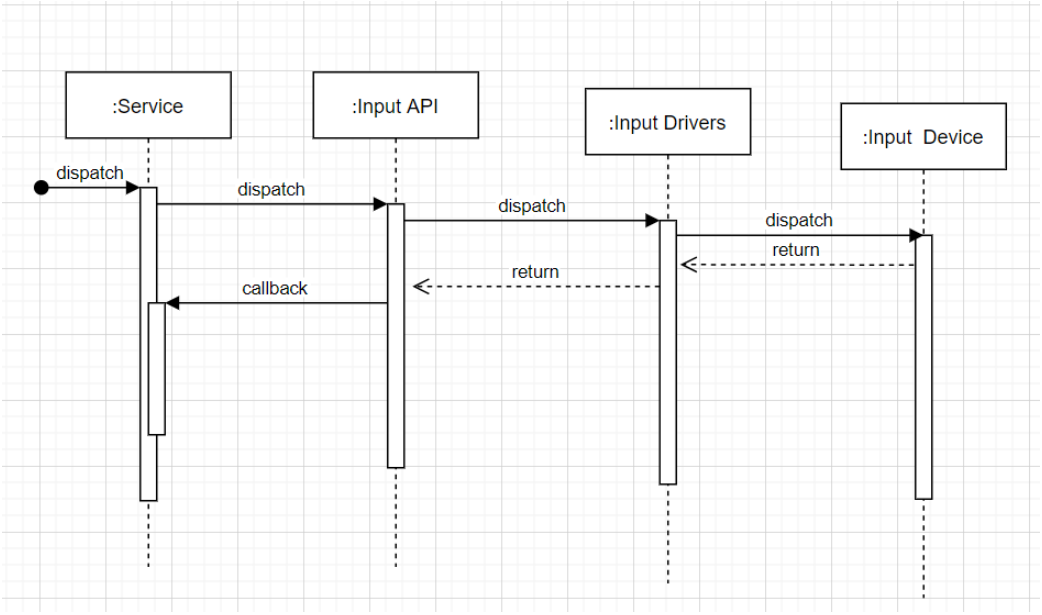


Figure 6.4.3(d) Input API Sequence Diagram

Output Devices and API

Again, as stated above, in addition to the input drivers and input API, the software architecture will include an output API to coordinate output responses through receiving calls from the services and establishing which routines to run in order to set the state of the output devices. The output system will consist of haptic feedback units, those being vibration motors and a speaker.

Output Devices and Drivers			
No.	**Name**	**Description**	**Uses**
1.0	Vibration Motors	-Haptic Feedback -Set into 9 units of 2 throughout the front vest, vibrates in regions to communicate general direction of upcoming obstacle	SPI
1.1	Speaker	-Audio Feedback -Audible warning of upcoming obstacle	PWM
1.2	Digital to Analog Converters	3 setup to CM4 in a MISO configuration to deliver voltage to the motors	SPI

Table 6.4.3(c) Output Devices and Drivers

Output Architecture

Vibration Motors: Used to provide haptic feedback in response to a detected obstacle or need for necessary avoidance, awareness, or identification. These vibration motor units will be arranged in a grid like manner across the front of the vest and will be grouped to represent the direction of the upcoming object or obstacle. Varying levels of voltage delivered to the motor units will determine the intensity of the vibration. Routines will require use of functions
will require DACs

Speaker: Will be used to audibly provide feedback.

Services			
No.	**Potential Identifier**	**Description**	**Request to Interface**
2.0	**Output Interface**		
2.1	DepthPerceptionService	Obstacle Detected, Alert Wearer	Signals Output Units Overall-Haptic motors and Speaker
2.2	CurbDetectionService	Platform height difference detected ahead. Alert Wearer	Signals output units- Signal motors and speaker for stepUp/stepDown responses
2.3	EmergencyDepthService	Sudden arrival of large or present obstacles detected directly in front of the individual. Signal for immediate stop	Signals output units- Emergency vibration pattern Audio Alert
2.4	Low Power Service	Device set to low power mode: Non-essential functions are shut down	No output units signaled

Table 6.4.3(d) Services

Output Design

Vibration Motors: To communicate with and deliver voltage to the motors a set of 3 DACs are connected to the CM4 in a MISO(Master-In-Slave-Out) configuration.

Speakers: Uses PWM and can be passed audio files.

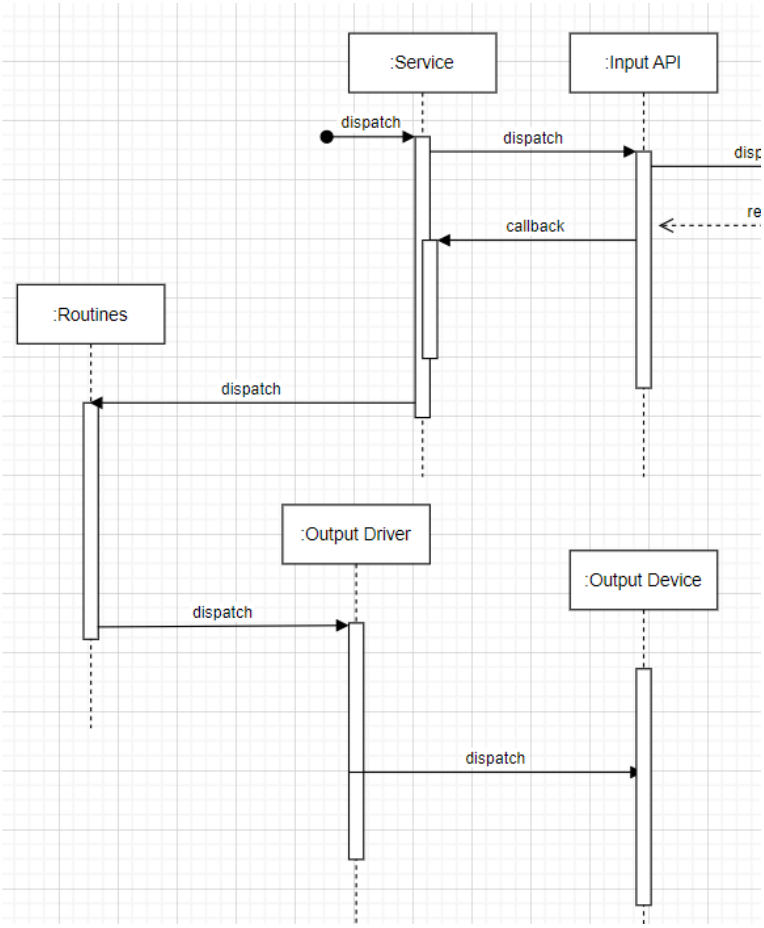


Figure 6.4.3(e) Output API Sequence Diagram

Interrupts

In addition to the input drivers, Input API, and Output API, the system architecture will account for user interrupt via button pushes to change the mode of the device or to toggle power the device on and off and for overrides of currently running processes in the event of an emergency via hardware/software interrupt services.

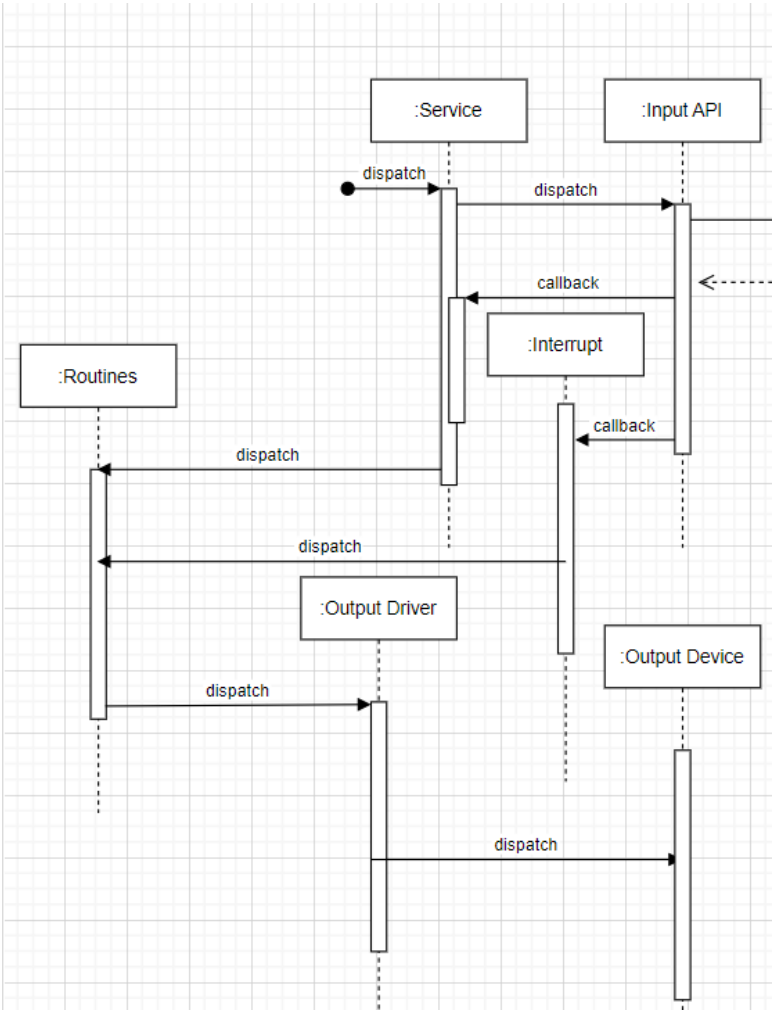


Figure 6.4.3(f) Interrupt Service Sequence Diagram

Hardware Interrupts

The hardware interrupt component connects GPIO events to their servicing routines, particularly for the user input read via buttons. This component is self-contained and does not necessarily need to provide functionality to other components, aside from a constructor for the system to instantiate it. However, some internal structure for convenience will be introduced. For example, a list connecting strings (e.g., "Button1Down") to GPIO numbers, a method for connecting callback functions (i.e., routines) to these events, and potentially ways to disconnect routines from events. This component provided motivation for separating services (relating to input processing) and routines (relating to output processing and system state)—no sensor data is required to service our intended hardware interrupts. This component will use the event-bus architecture, and is visually depicted in Figure 6.4.8. One GPIO signal implies exactly one event, and every event may have one or more callback routines. Part (b) of the figure shows this structure applied to the user interface, which is the intended use case.

Programmers will connect the execution method of a routine to the callback parameter of one named event on the bus. The signal which triggers the event will be configured around the time

of setting the callback. This linking is done once at runtime during the component initialization, assuming no system crashes. After this linking, the component no longer needs to be tracked by the system unless some extra custom functionality is desired to attain stretch goals established in Section 3.1.

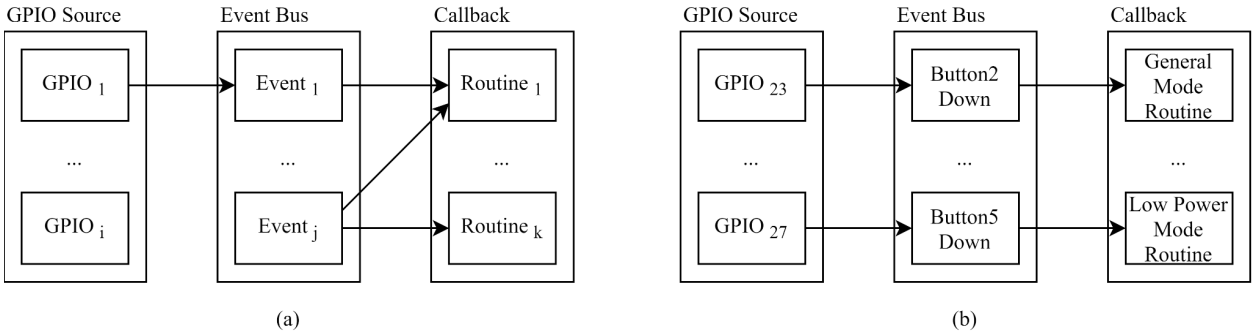


Figure 6.4.3(g) User Interface with Event-Bus Architecture

6.4.4 Software Design and Implementation

Class Diagram

The software class diagram is shown in Figure 6.4.9(a). It takes the concepts presented in Figure 6.4.3(a) and converts them into explicit class definitions. Given the similarity of the SystemService and SystemRoutine structures, a superclass of SystemContainer is defined. These subclasses will be identical to the superclass with the exception of the constructor and an additional convenience method for improved code readability (e.g., GetService instead of GetExecutable). Given the similarity of Service and Routine both needing to have processes run, a common superclass of Executable was created. These subclasses contain instance variables which hold references to instances of Input and Routine, and Output and DPHS, respectively. The DPHS, as mentioned previously, contains the system itself as well as its state. The Scheduler interacts with the SystemService object with its GetService method, and then runs these services on some loop within its Run method. HardwareInterrupt, in general, is only used once at startup to initialize user interface buttons. References to Input and Output instances are given to Services and Routines respectively, and should not be used by any other components. Only one instance of each concrete class is instantiated, and all references are stored within the DPHS instance.

The particular subclasses of Service and Routine are included in a separate diagram for organization and readability. Figure 6.4.9(b) contains the class diagram for these classes. The structure of each service and routine is described below the figure.

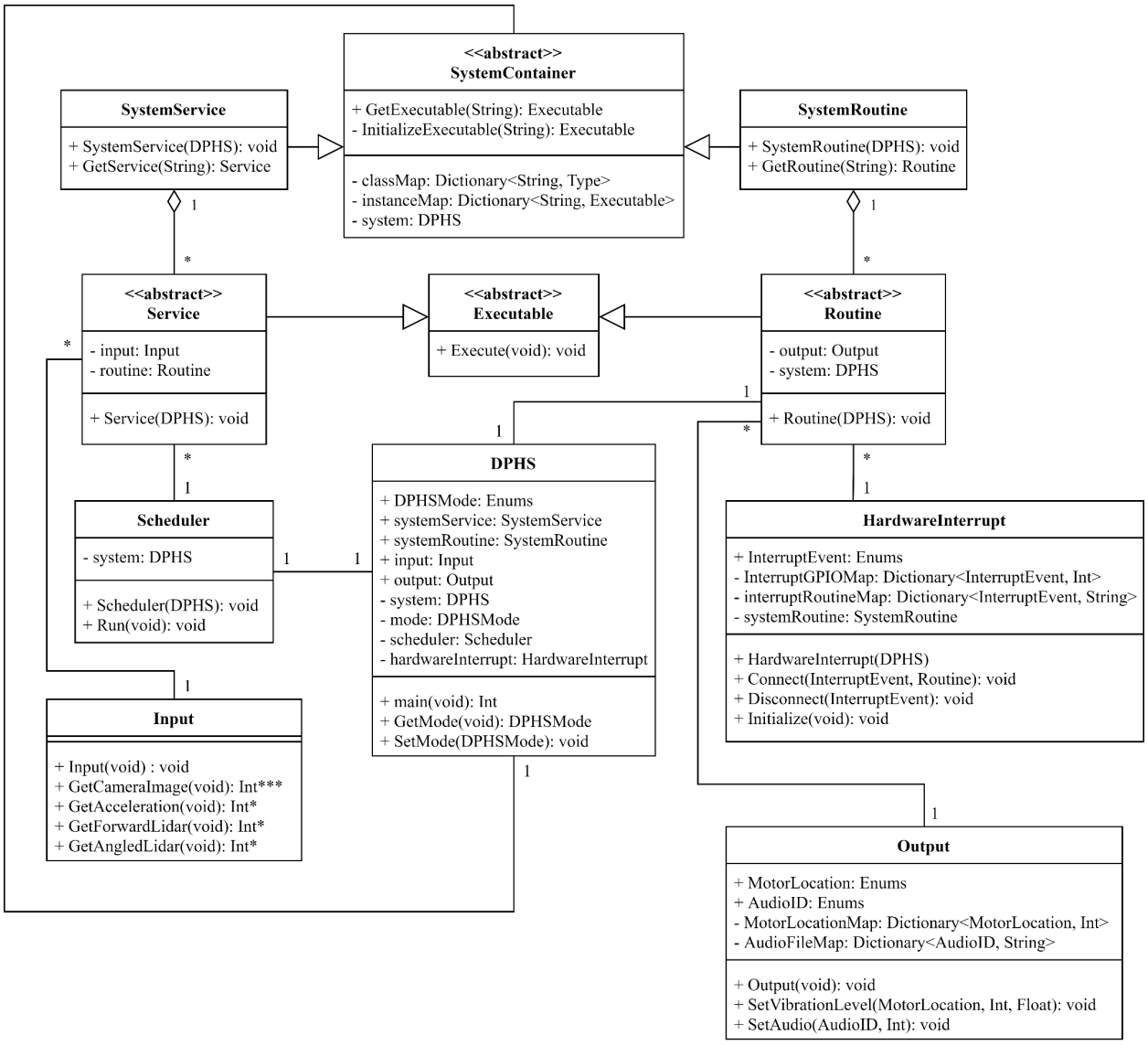


Figure 6.4.4(a) Software Class Diagram

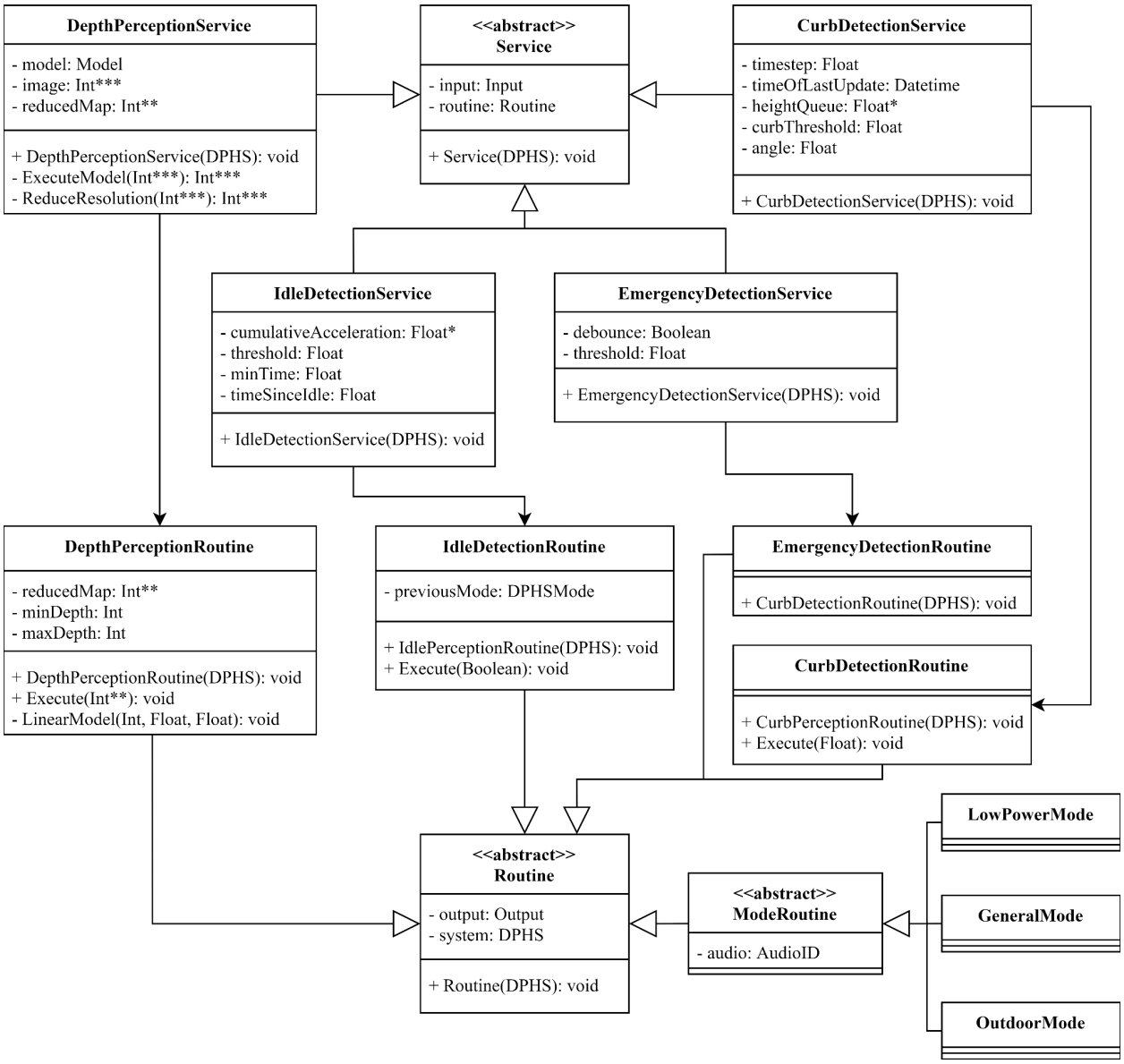


Figure 6.4.4(b) Implemented Service and Routine Class Diagram

DepthPerceptionService

The core of the application is contained within the DepthPerceptionService. It is responsible for taking images provided by the input component and applying the neural network model. The network architecture is introduced in Section 4.2.4. Although the organization running this workshop did not make their dataset public, several monocular depth datasets currently exist ("Monocular"). One or more of these datasets will be used to train and test the model. The chosen dataset(s) must contain a mixture of indoor and outdoor images to ensure robustness of the final model. This model will be stored on the processor long-term storage unit and be loaded on the initialization of this service. On every call to the service, this model will be applied to the input image. This image will then be reduced into its final array of nine depth estimation values

(as depicted in Figure 4.4.4). The result of this parameter reduction will be passed to the routine which will apply its logic for sending haptic feedback.

The parameter reduction will simply take the minimum depth value of a particular region of the output map. More techniques will be explored as testing is carried out in order to fine-tune the device. Each region should be mapped one-to-one with an output unit. The routine will take the nine minimal values (which themselves exist on some range with a minimum and maximum), and request a haptic feedback level which is linearly proportional to it. Once again, different relationships (such as those shown in Figure 4.4.5) may be tested to fine-tune the experience of the system. For a more in-depth review of these techniques, see section 4.4.

EmergencyDetectionService
The EmergencyDetectionService, IdleDetectionService, and CurbDetectionService are all similarly named because they all trigger a routine only after an event is detected. The EmergencyDetectionService relates to the forward-facing LiDAR module. It triggers an emergency alert when the distance between the user and an object directly in front of the unit is below a certain value. This service only needs to maintain a floating point value indicating the minimum distance which triggers the alert, and an additional boolean value to prevent bouncing of the signal. When the service is called, the LiDAR distance will be read, compared against this value, and then generate the alert. The boolean value will be set to indicate that an alert has already been triggered. Once the service is called again, the boolean will be checked with the LiDAR. If the boolean is set and the LiDAR registers above the threshold, it will be reset. The state machine for this boolean value which determines the result of the service call is depicted in Figure 6.4.10.

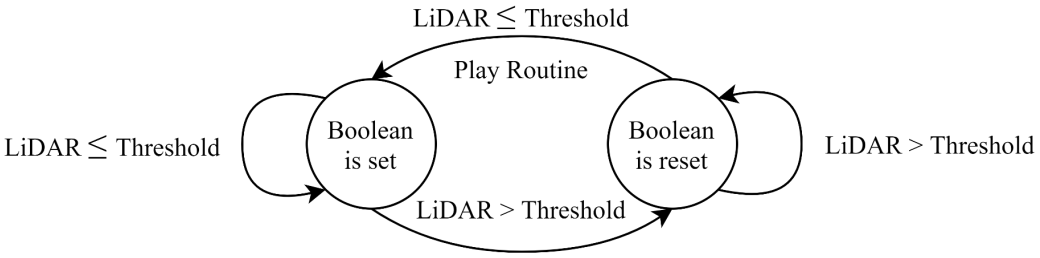


Figure 6.4.4(c) EmergencyDetectionService state machine

The routine associated with this service will rapidly switch between maximum intensity haptic feedback and zero feedback over a certain period of time across all units, irrespective of the data read from the LiDAR.

IdleDetectionService
A simple service which estimates the velocity of the user simply by maintaining three float variables with the forward, upward, and side-to-side acceleration of the user. On execution of the service, the acceleration is read and added to the floats. Assuming that the user begins with a velocity of zero, the sum of the accelerations taken at every timestep should tend toward zero whenever the user is stationary, and in motion otherwise. A threshold will be used to assess the magnitude of the array. When the magnitude is below the threshold for a defined period of time,

the low power mode will be activated via a routine. Once motion is detected above the threshold, the system returns to the previous mode.

CurbDetectionService
This service relates to the angled LiDAR. If an instantaneous change in the distance measurement is read, it will trigger an alert which contains information about the orientation and magnitude of the curb. Similar to the emergency service, this service must also maintain a threshold of change which qualifies as a curb. Additionally, some history of values should be stored to detect changes over differing degrees of steepness.

Float threshold and time-step values will be defined as constants which will determine when samples of the distance are taken. An additional float will contain the angle that the LiDAR beam makes with the plumb line of the device. This will be used to approximate the height of the curb. A finite queue of estimated height values will be maintained, each value of which is only added to the queue of the time since the last update exceeds the time-step. The length of the queue implies the rate of change in the elevation that qualifies as a curb. On every execution of the code, the difference of the maximal and minimal value of the height is compared to the threshold value. If this value is exceeded, a routine is fed the difference, and generates a step-up or step-down signal in the output proportional to the difference. Extra threshold values provide a maximum elevation difference that qualifies as a curb. If this difference is exceeded, some form of emergency response must be generated. This emergency response should specify whether the change in elevation is an incline (e.g. a barricade) or decline (e.g. an open manhole).

User Interface Routines
Each mode will contain a simple routine which enables a mode based on the button pressed. The routine must contain (1) a unique pattern of vibrations to make the mode easily identifiable over the tactile interface, and (2) play audio which matches the mode. Each of the three mode buttons will follow the same logic, depicted in Figure 6.4.11.

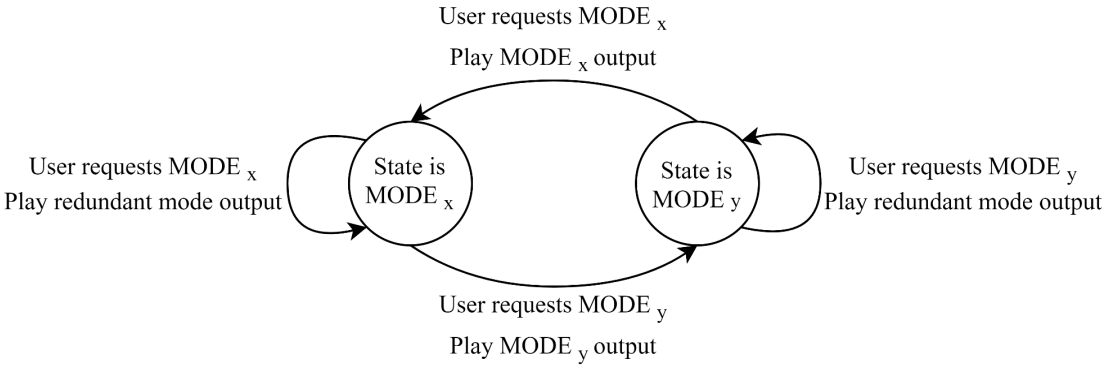


Figure 6.4.4(d) User interface and system mode state machine

This software component will not take button-mashing into account, and will naively adjust the mode whenever called. No interrupts are fired to alert the scheduler either. It is assumed that the scheduler will change its algorithm once it spots the change.

Implementation on Pi 4

The Raspberry Pi 4 is the latest version of the Raspberry Pi computer and consists of an electronic circuit board. It is an all around faster model than its counterparts, possessing fast storage via 3.0 USB, fast network connections via Gigabit ethernet and is able to process up to 4k video feed. The Raspberry Pi Compute Module 4, CM4, was opted for this project

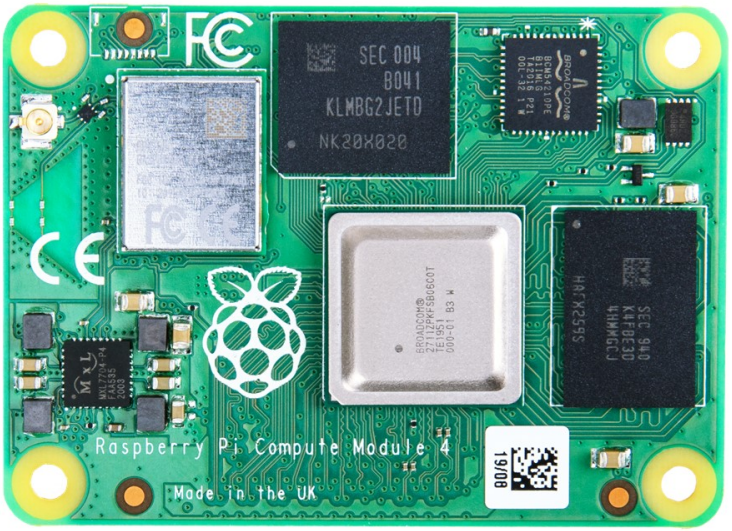


Figure 6.4.4(e) Raspberry Pi Compute Module 4(CanaKit, 2021)

The raspberry pi 4 can run a number of operating systems, including the officially recommended Raspbian OS, Snappy Ubuntu Core, Ubuntu Mate, the non-Linux based Risc OS, and the Kodi-based media centers OSMC and LibreElec.

As stated above, the two programming languages projected to be used are python and C++. Software requirements and the implementation of the architecture for the Pi relies on the standards of either or both of these languages, see section 6.4.1.
Python is already hosted by the raspberry pi, and thus implementing code on the pi in python will have the easiest translation. C++ adds object oriented functionality to C programming. The pi can support C++ and can have a faster runtime than the python.

Schedule of Deliverables and Dependencies

Number	Task	Start	End	Status	Responsible
Software Milestones					
1	**Planning**	8/23	12/07	Not Compete	Cris, Christa
2	Detailed Design Map List	11/07	11/18	Not Complete	Cris, Christa
	Software Documentation	11/07	12/07	Not Complete	Cris, Christa
3	**Implementation**	12/07	1/12		
4	Environment setup.	01/03	01/20	Not Started	Christa
5	Programming drivers	01/20	02/20	Not Started	
6	Process input	02/20	03/20	Not Started	
7	Input API	02/20	03/20	Not Started	Christa
	Services	02/20	03/20	Not Started	Cris, Christa
	Output API	02/20	03/20	Not Started	Christa
8	Establish bare bones proto.	03/15	04/01	Not Started	Cris, Christa
9	Testing	04/01	05/01	Not Started	Cris, Christa

Table 6.4.4 Software Milestones

7.0 Project Prototype Construction and Coding

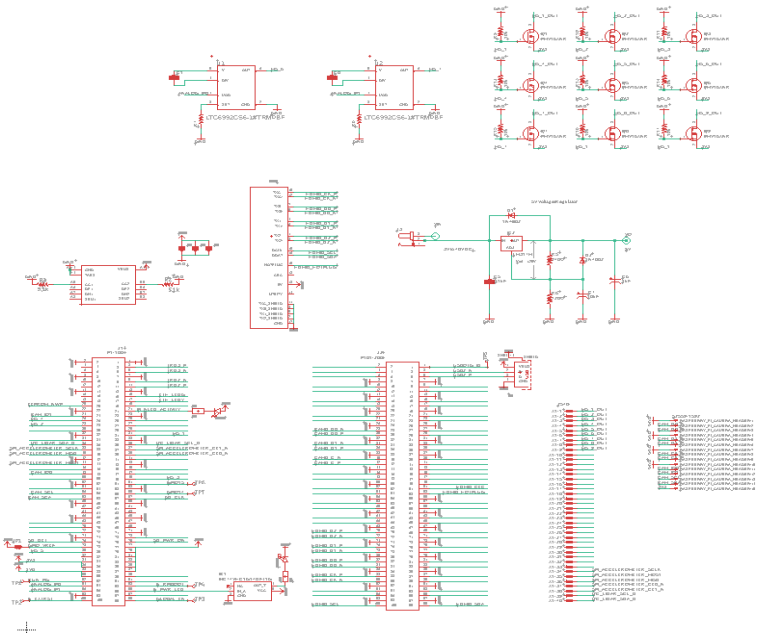


Figure 7.0 Overall Schematic

7.1 Integrated Schematics

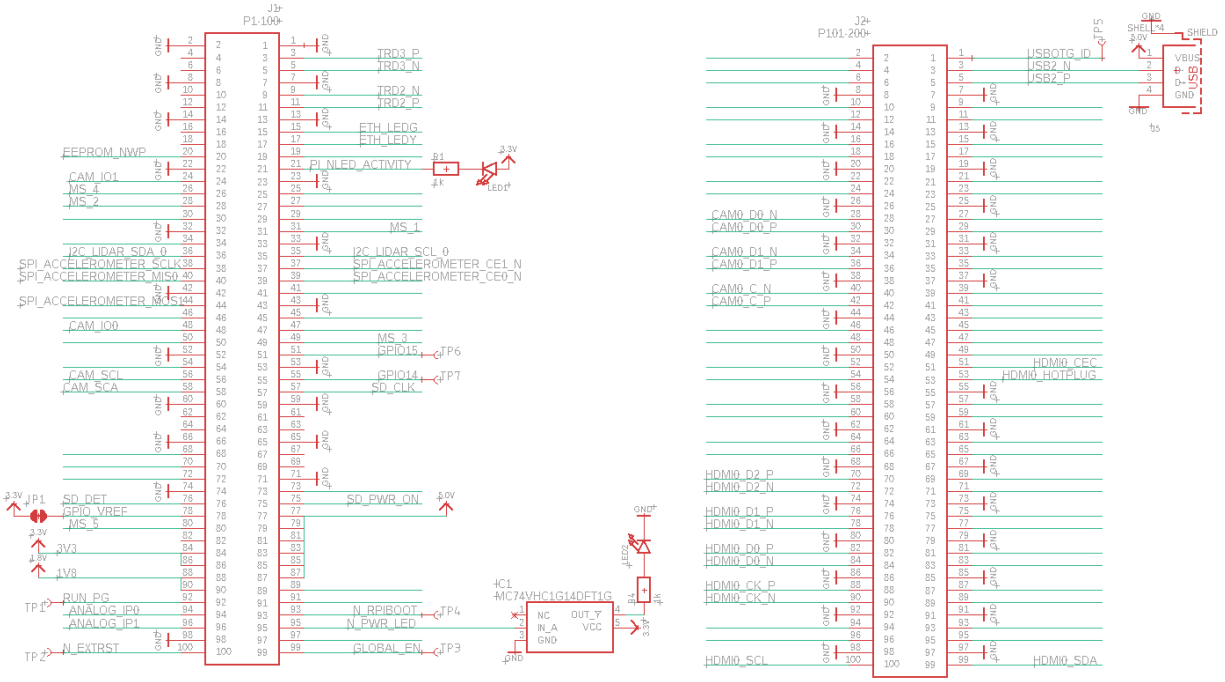


Figure 7.1(a) Compute Module Schematic

The above figure includes component ‘Module1A’ with GPIO pins and connections that report directly to the Raspberry Pi Compute Module 4 chipset. Two LEDs are implemented to show when the device is on (PWD LED) and show relative activity of the device (Activity LED). The number of haptic feedback motors will require implementation and testing in the prototype but is expected to have 9 individual motors. The available pins for motor control use PWM in junction with MOSFET. One pin per motor will be used for control through PWM (separated into individual channels) thus totaling a number of 9 pins for the output of 18 output pins 9 x 2 sets. As each pin uses analog the motors will be cascaded with transistors using a total of 2 pins each, thereby giving 2 voltage sources depending on the MOSFET.

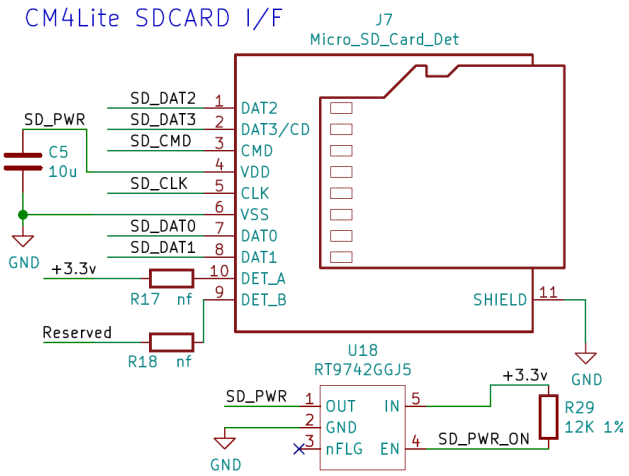


Figure 7.1(b) SD Card Mounting Device Schematic

Memory storage was a concern for this project. Compute modules have a specification of requiring eMMC memory on the carrier board when the CM4 has low RAM, however on models with high RAM the eMMC is embedded onto the CM4 and takes away the ability to connect SD storage onto the carrier board. Also in the design process, it was decided to replace the SD data paths with additional GPIO for more motor control support.

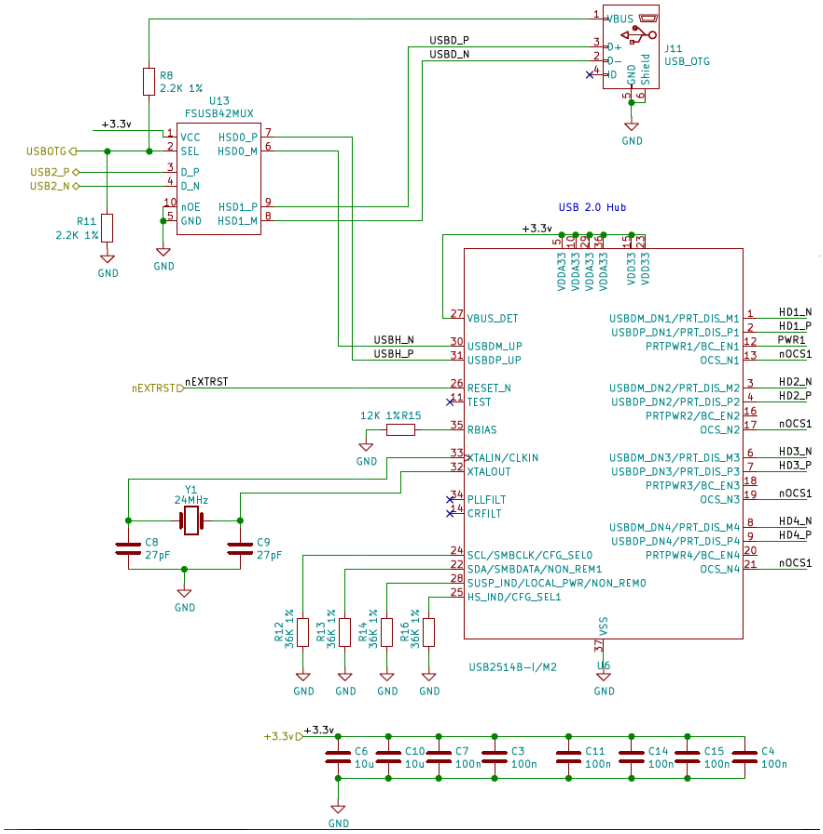


Figure 7.1(c) USB-B Hub

The schematic presented above shows the option to add multiple USB buses, however they share the same lane on the compute module. In the figure below, the high speed serial manages the USB-B pins which only shows one lane for the communication. For the purpose of reducing cost and size of PCB carrier design, a separate USB hub module will be used in the coding design in stages of prototyping as the function will remain relatively the same but with slower speeds due to a difference in voltage.

To verify that the device is working during the prototype stages of development as well as for the purpose of debugging the device, a red and green LED will be configured onto the board to allow the developer to know whether the device is currently on or off.

The above schematic includes the high speed serial BUS which connects many of the complex input devices. The CM4 camera header uses 15 pins of which 8 connect locally to the high speed serial. A global label was needed for the CAM_GPIO and source of 3.3 volts. A single USB-B is connected locally in the schematic. The HDMI component connects locally to the high speed serial for all of its communication pins while power and ground are global. The USB component uses a safety circuit (TD2EUSB30) to prevent burnout of the connected device as well as the port itself.

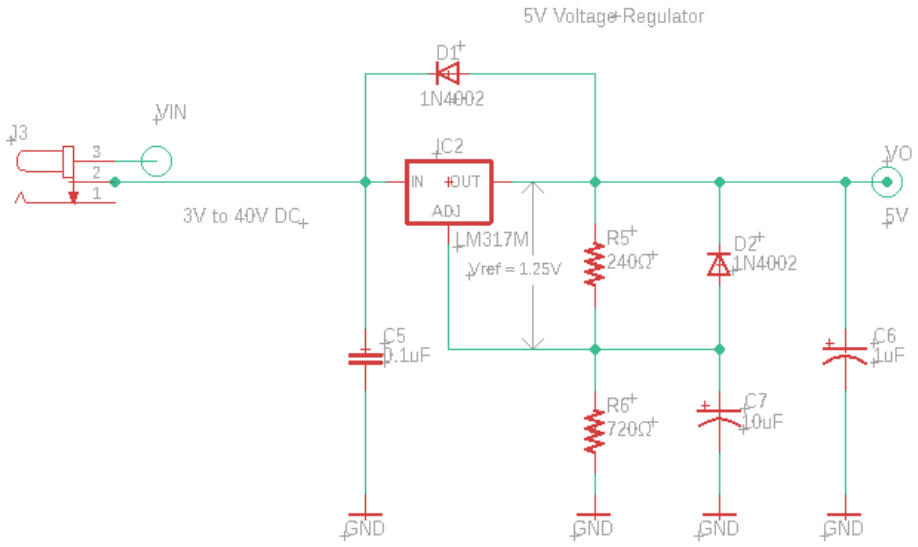


Figure 7.1(d) Voltage Regulator Schematic

The power source of the device will be a battery filtered through a step down voltage regulator. A barrel plug is used for easy plugging of the device, the range of options for chargers, easy setup, and large availability. As the step-down voltage regulator will be used to power the entirety of the device, there may be additional implementation of another unit in cases that the 3 amplitude limits the devices functions. Also referring to section 7.2.4 it can be noted that power lines must have larger width to carry the necessary current to all components. In regards to this issue of needing larger width, bus lanes are created throughout the circuit, however this still does not fix the limiting issue of a 3 amp limit. Capacitor ‘C3’ helps in ripple rejection upto 15dB. The diode ‘D1’ protects the device against any short circuits that may occur.Diode ‘D2’ protects

against output short circuits for capacitance discharging. Capacitor ‘C2’ on the output helps to improve transient response thereby giving a cleaner voltage source.
For development purposes a barrel plug will be used to allow for an easy conform factor into the vest. For list of options and barrel sizes refer to section 7.2.2 for specification.

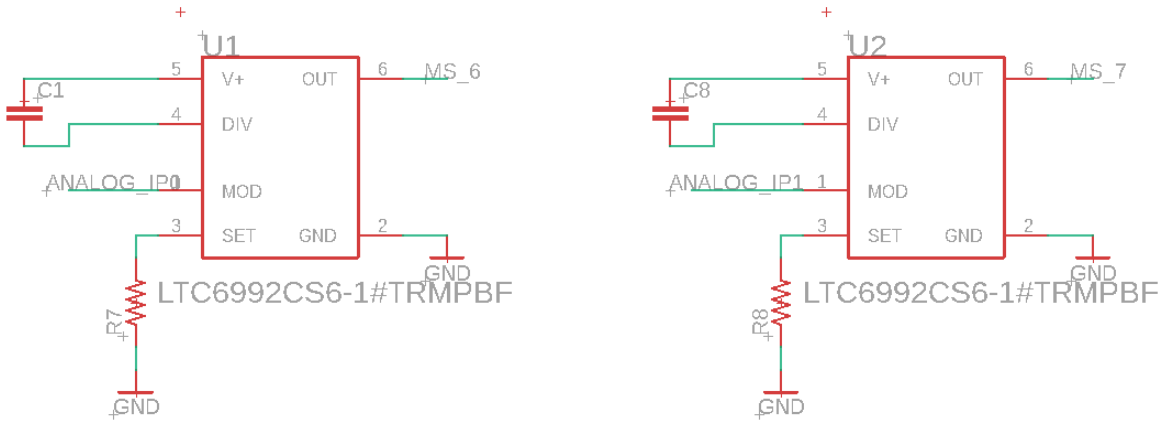


Figure 7.1(e) Analog to PWM Conversion

The two modules above act as conversion modules allowing a standard analog signal to become PWM. Due to the lack of available PWM pins on the CM4 this procedure was done in an effort to get two additional pins for the set of 9 motors.

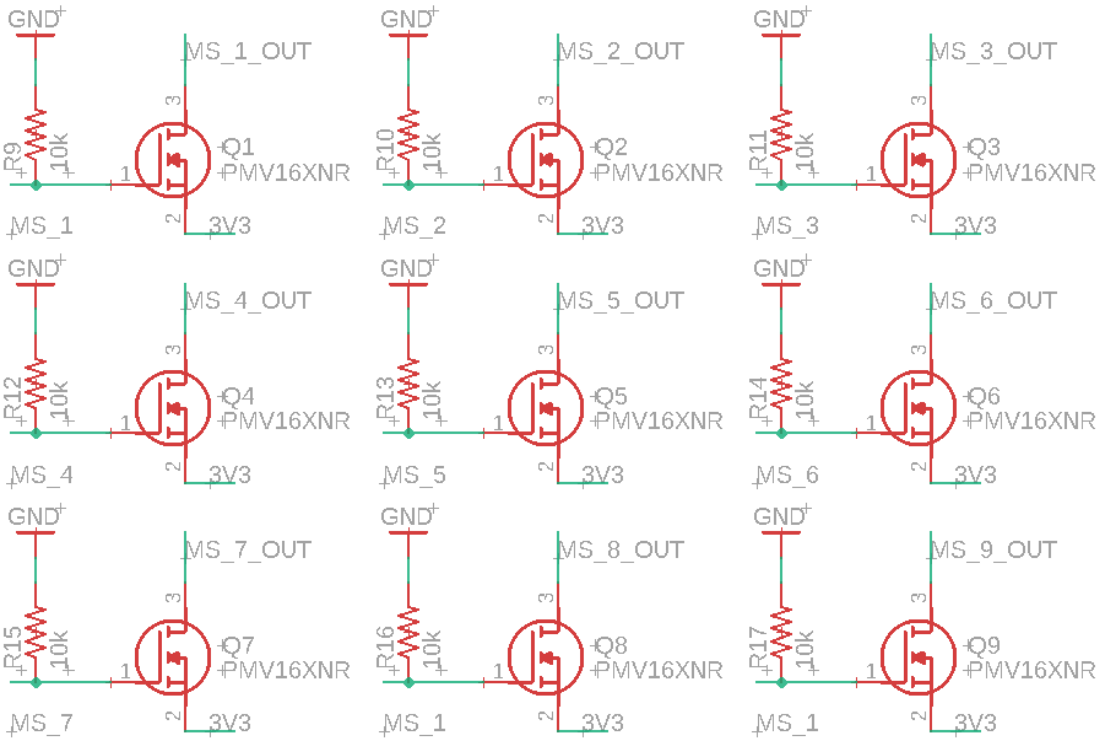
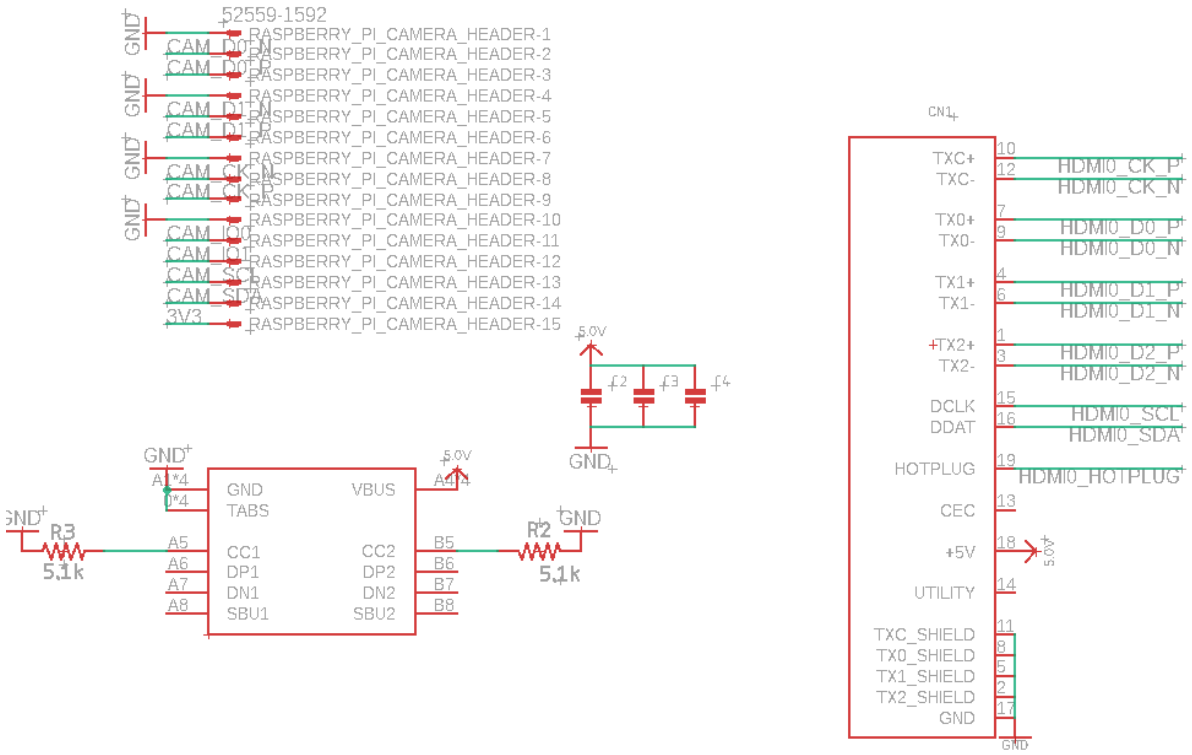


Figure 7.1(f) PWM MOSFET Motor connection

There will be a total of 9 MOSFET devices controlling a set of 9 different motors, controlled by 9 different PWM pins. All motors will be connected to the same power source and share the same ground. A diode and capacitor will be soldered directly onto the motors to make the circuit simpler.



7.1(g) High Speed Serial: Raspberry Pi Serial, USB, and HDMI (top to bottom, left to right)

For the developmental purposes of the initial PCB design, 32 pins for motor outputs, 5 pins for input buttons, and 2 I2C pins for LiDAR, and a Master-Slave pin for the accelerometer will all be on the same pin header totaling 40 pins in use out of a 40 pin header. Each ‘D2A’ chip is used for Digital (SPI) to Analog conversion. Using 4 pins including for each SPI connection with an output of 8 controller pins; the table below shows each pin and its description.

Digital Input (SPI)		Analog Output
Channel	Description	Channel
VDD (pin 16)	3.3V Power	CH0 (pin 1)
VREF (pin 15)	3.3V Power	CH1 (pin 2)
AGND (pin 14)	Analog Ground	CH2 (pin 3)

Digital Input (SPI)		Analog Output
CLK (pin 13)	Clock Pin	CH3 (pin 4)
DOUT (pin 12)	Data Out	CH4 (pin 5)
DIN (pin 11)	Data In	CH5 (pin 6)
CS/SHDN (pin 10)	Chip Select	CH6 (pin 7)
DGND (pin 9)	Digital Ground	CH7 (pin 8)

Table 7.1 Digital Input to Analog Output

7.2 PCB Vendor and Assembly

7.2.1 PCB Vendors

The preferred manufacturer of components is Digi-Key as most of the components have existing symbols and footprints on UltraLibrarian and other websites that allow for the download of EAGLE and KiCAD files.

Outside of manufacturing, the vendor choices for the pcb development are set to three options including PCBWay, OSHPARK, and JLCPCB. All of the listed vendors check the PCB before shipment to ensure that the number of layers and price of components is correct. The PCB vendors listed require gerber files after selecting the size and material of the PCB.

Differentiated from each company, PCBWay is a middle range between cost and reliability. Oshpark is well known for their speed in making components but also their higher overall expense. JLCPCD is noted as the cheapest of the three companies when it comes to enthusiasts making PCBs. The minimum number of PCB boards that may be boarded by each company per design is 5, 3, and 5 units for PCBWay, OSHPARK, and JLCPCB respectively.

7.2.2 PCB Dimensions and Layers

The overall dimensions of the carrier board will vary depending on later specifications after the development period is complete due to a lack of need in certain peripherals in the final version. As the PCB will be designed with 2 layers in mind; the board is expected to be within the dimensions of 75 x 64 x 21 mm, referenced from the Seeed Ethernet Carrier Board Model. The overall shape of the design will be rectangular due to the need for various peripheral interfaces such as the 22 pinout camera header, USB-B, HDMI, and power source.

The decrease in the number of layers will lead to multiple constraints. There is a lack of additional routing space and ability to control trace impedance. This is important for signal integrity when designing a high-speed digital device. It will also be very important to be careful of grounding certain components due to issues that might occur like current loops. In comparison to a 4-layer, a 2-layer will have sufficiently more space (almost double) to place components

throughout the board with minimal cost. Functionality of a 2-layer does not typically show propagation delay whereas a 4-layer can at times show impedance when dealing with multiple layers.

The type of barrel plug for the charging of the device will be selected from a range of coaxial plug power cords. The dimensions on the cord and housing depend on the angle and wire size (AWG) intended on the final design. The table below lists some cords of consideration.

Part No.	EIAJ Class	I.D.	O.D.	Molded Plug	Wire Size (AWG)
TC204	-	1.3mm	3.5mm	Right Angle	24
48-218	-	2.1mm	5.5mm	Right Angle	18
48-410	1	.7mm	2.35mm	Straight	24
TC218	-	2.1mm	5.5mm	Straight	18

Table 7.2.2 Barrel Plug Specifications

7.2.3 Bill of Materials and Availability

Due to the current market situation, a chip shortage is currently taking place limiting the availability of chips, pin headers, and many other essential components. Some parts may be acquired through the modification of previously used parts in premade devices. One example of such an issue is the lack of availability for the circuit protection in the bill of materials. This part may need to be replaced with a component with similar specifications. Another issue noticed is a general lack of female pinout headers available for PCB development. Due to this lack of availability, male headers were selected for the current design. A 15 pin header was used due to the 2-layer implementation discussed in section 7.2.2.

To reduce cost in the PCB there are four main factors that contribute to price and duration of production: the number of layers, color of PCB, type of soldering mask/material, and the shipment cost. To reduce on cost from the number of layers on the PCB, the CM4 is a pre-existing four layers board that will be using a 2 layer custom I/O interface.Two layers allows for enough flexibility to have a decent number of components while keeping costs in the range of below fifty dollars, whereas four layer is double the price per board. The color of PCB can change the cost of the PCB by an amount depending on the number of layers, size of the PCB, and the design of color scheme. For general purposes the board is intended to be at a defaulted green color. Solder mask and the usage of certain materials for the PCB such as steel or copper greatly increases the conductivity when needed for certain high speed applications.

Name	Part Number	Qty	Manufacturer	Unit Cost ($)	Description
LED	160-1478-1-ND, 160-1479-1-ND	2	Digikey	0.24	Light Emitting Diode
Connector Pin Header	1-826658-0	1	TE Connectivity	2.32	Headers & Wire Housings 2X10 POS AMPMODU II PIN HEADER
USB-B	USB4105-GF-A	1	Digikey	0.85	Universal Serial Bus Type-B
Qwiix/Stemma	SM04B-SRSS-TB (LF)(SN)	1	Digikey	0.55	Connector Header Surface Mount
Compute Module	CM4	1	Raspberry Pi	35	Control System
Voltage Regulator	MP1584EN-LF-Z	1	Monolithic Power systems	2.95	Chip for voltage regulator
CM4 Camera Header	FH12-22S-0.5SH(55)	1	Digikey	2	Pin header for 22 pin unit
HDMI_A_1.4 (J5)	2000-1-2-41-00-BK	1	Digikey	1.62	HDMI unit
Circuit Protection	TPD2EUSB30DRTR	1	Texas Instruments	0.65	8V Clamp 1A
Camera	OV564	1	Unknown	9.15	Raspberry Pi Camera 22 pinout

Table 7.2.3 Bill of Materials

7.2.4 Assembly

Before moving around the components on the board it was important to update the PCB rules by the manufacturer. OSHPARK was used as a reference as they have all the specifications posted and are known to be an average facility, meaning that if a PCB within specification can be done for OSHPARK, then it can be applied to most other facilities.

Trance Clearance	Trace Width	Drill Size	Annular Ring
6 mil (.1524mm)	6 mil (.1524mm)	10 mil (2.54mm)	5 mil (.127mm)

Table 7.2.4(a) OSHPARK Minimum PCB Specifications

The footprints and symbols were all updated in accordance to match Raspberry Pi's board measurement in millimeters.When using the DRC check an error will be found when the designed PCB does not meet specification in KiCAD.

USB uses a calculated impedance to prevent data impulses from occurring at different times which would cause an error. It is important to create a differential pair and calculate the impedance and how to route the lines effectively. The best way to do this was using the PCB Calculator embedded in KiCAD

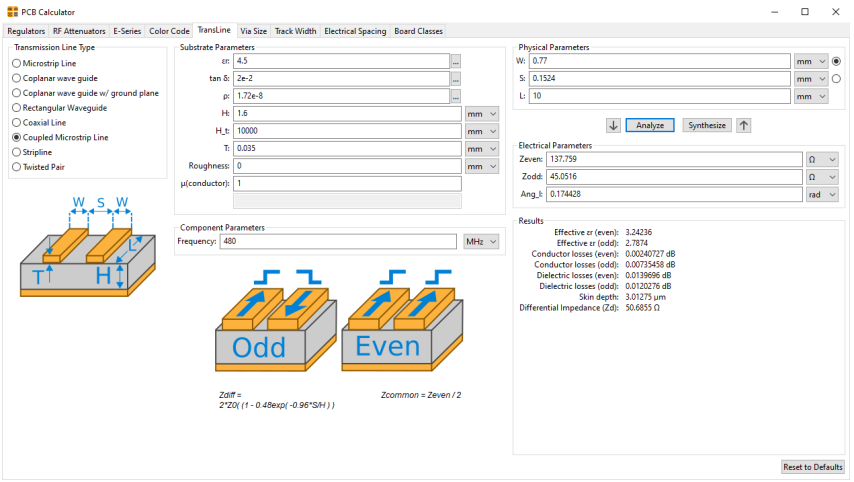


Figure 7.2.4(a) KiCAD PCB Calculator

Can calculate common mode or differential but it was recommended to use the value from differential impedance. Common mode is important, however modern USB-B does not use it as much as differential impedance. Referring to documentation, differential impedance is required to be about 90 ohms.The following values were used within the calculator

Name	**Value**	**Description**
Er	4.5 (FR4)	Dielectric Constant
TanD	2e-2 (FR4)	Dielectric Loss Tangent
Rho	1.72e-8 (Copper)	Sensitivity of Material

Name	**Value**	**Description**
H	1.6mm (OSHPARK Board Thickness)	Height of Differential Traces
H_t	10,000mm	Height of ground plane above traces
T	1.4 mils, .03556mm, 1 oz (Copper, by OSHPARK specifications)	Thickness of the Traces
Rough	0	The roughness of the PCB
Mu Rel C	1	Relative Permeability of the Inductor
Frequency	480 MHz	Expected rate of frequency for wire

Table 7.2.4 (b) PCB Differential Pair Calculation Input

Note the height of the ground plane above traces (H); an infinite value or high value is desired for this PCB design as there is no ground plane above the traces since the design is 2-layered. The roughness of the PCB does not matter for the purposes of this design as no antenna or similar device will be implemented into the closed system.The relative permeability and expected rate of frequency for the wire are negligent for the current design.

Name	Value	Description
W	.77	Width of traces
S	.1524	Separation between traces
L	10	Length

Table 7.2.4(c) PCB Differential Pair Calculation Output

The length (L) does not matter for the finding of differential impedance. The values must be changed multiple times to meet ‘Zodd’ at a value of 45 to give a differential of 90.

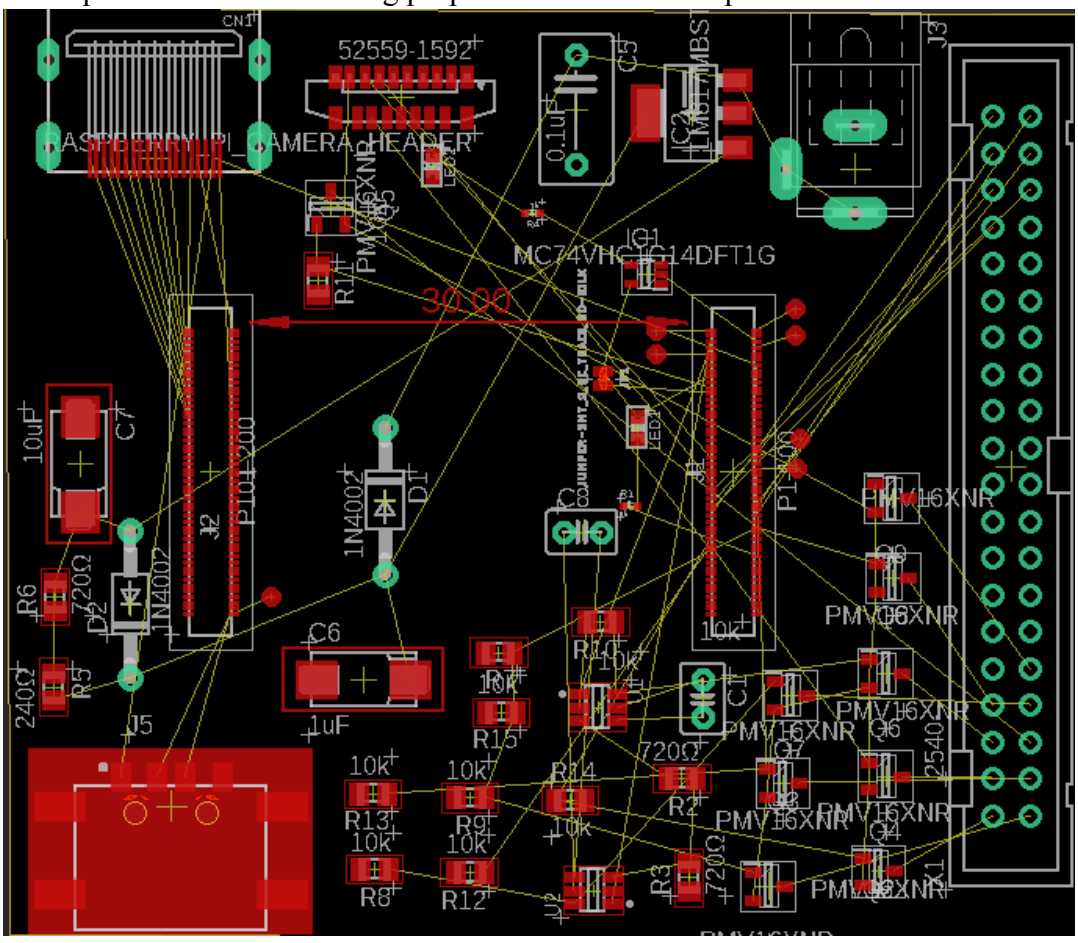
$$Zdiff = 2 * Z0((1 - .48e^{-0.96*S/H})$$

Equation for finding Differential impedance

Name	Value	Description
Zeven	137.777	Z line drive pair with equal and opposite voltages (differential voltage)
Zodd	45.0379	Z line drive pair with same voltage in both lines
Ang_I	.174428	Angle of Mutual Inductance

Table 7.2.4(d) Differential Pair and Mutual Inductance

According to High-Speed Driver documentation up to 480Mb/s the differential impedance is required to be 90Ω ± 10%. This value of Zodd is used for the USB-B connections enabling an approximate path for serial data using proper wire width and separation.



7.2.4(b) Raspberry Pi Carrier Board Ratsnest

Many options were taken into consideration when moving components while the circuit was under ratsnest. Many components had the option to be moved under the compute module using small footprints including resistors, some capacitors, and the voltage regulator. For the purpose of the prototype design the voltage regulator was left on the side of the board as it acts as a voltage source to many components throughout the board which may harm the compute module.

Other components such as HDMI and USB-B were implemented around the compute module as their height may affect the slotting on the module itself should they be embedded directly underneath. While it is an option to find taller connectors, further research showed no product more than 3mm in height. As there is difficulty in procuring the pin headers for the raspberry pi carrier board due to limited availability an exact measurement of the height of the board is undetermined, expected values will be based on the footprints. It is important for certain parts to be tight within the current design due to limitations on cost, form factor being a requirement as the device will be implemented into clothing, and to aid in the I/O protection due to the many high serial communication methods throughout the device.

Many of the nest connections in the leftmost board make it difficult to route the schematic, however as most of the pins on the Compute Module Connector are to be grounded; they were placed as island connections going to the other side of the board.
The protection for the high speed serial such as that of the ‘TPD2EUSB30’ (U2) are connected as closely as possible to the ports to prevent burnout. As in accordance with the regulations learned in Junior Design, the feasible resistors and similar components were grouped together to make manufacturing of the pcb easier in the production line.

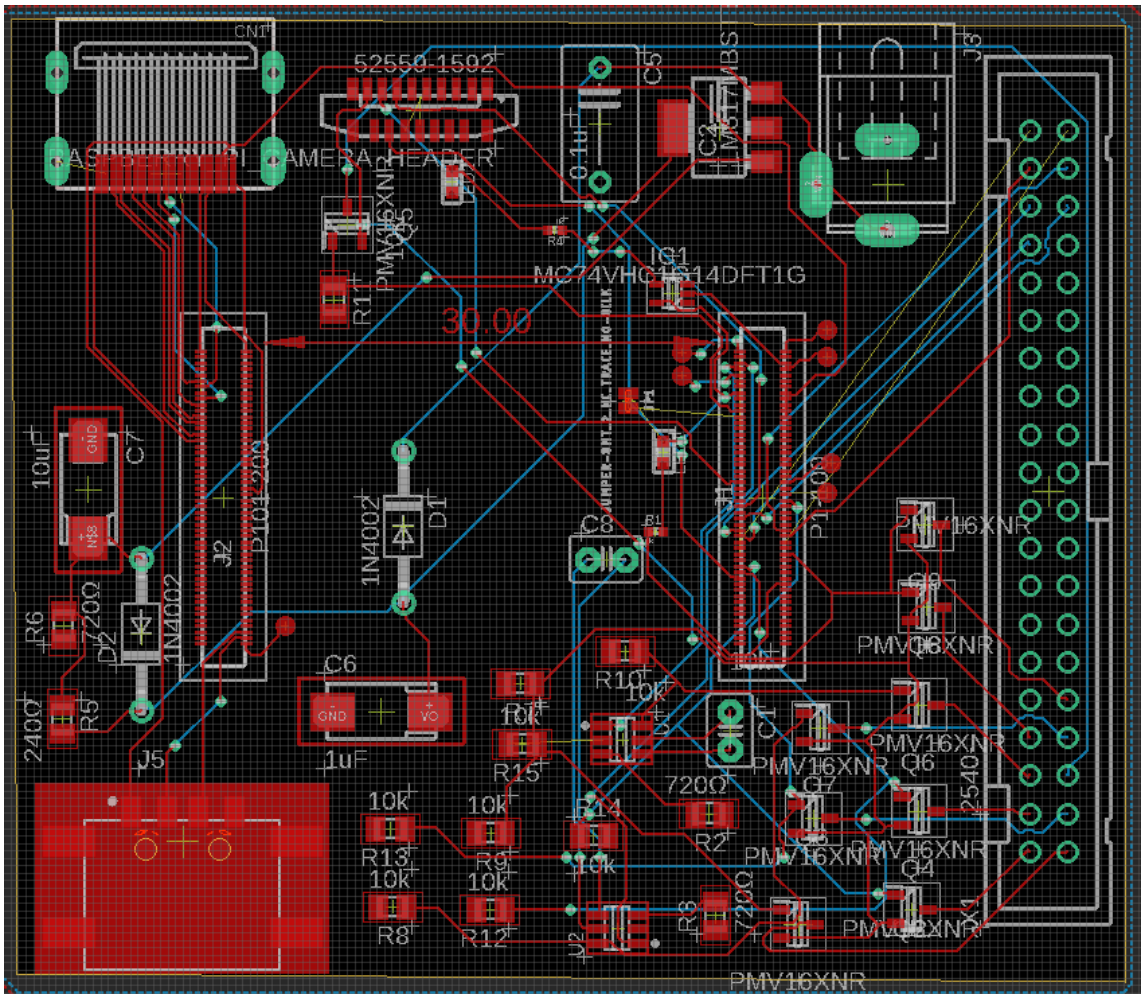


Figure 7.2.4(c) Raspberry Pi Carrier Board Routed

The first path routed in the design was the transmission line with the differential pair because the length matters for each wire and calculations must be performed so the signals meet on time, otherwise errors will result. When routing the differential pair it was required to use the specified value determined from the differential impedance and spreading out the lines at the specified size. The wiring for the differential pairing is enlarged by the amount calculated and carried to the USB-B device without making any right angles, only 45 degree angles. Single wires are then wired into the differential pair. The length of the wires must match within 150 miles (3.81mm) of the distance to prevent bad communication between a differential pair.

$$Length\ Difference = Length\ of\ outer\ wire - length\ of\ inner\ wire$$

Jogs could also have been implemented into the trace to match the total length for the set of traces. Depending on the component it is more important to match the length in high speed communication for applications like parallel buses, whereas other components focus more on the differential impedance or both.. For the purposes of this design acid traps will be ignored as it is noted as not being a problem for manufacturers in the current production line.

After the high speed lines, power lines were done next as they will be fitted to the length of each component to achieve the maximum available power since the Raspberry Pi Compute Module will be pulling up to 3 amps of power. For this purpose the track width formula must be used.

$$I = K * dT^{0.44} * (W * H)^{0.725}$$

Equation for calculating current per trace width on PCB

Name/Symbol	Value	Description
I	3.0	maximum current in Amps
dT	10	temperature rise above ambient in Celsius
W	1.35mm external layer	Width in mils
H	.03556mm external layer	Thickness in mils
K	.048	.024 for internal traces or .048 for external traces
Resistivity	1.72e-6 (copper)	Resistivity of material

Table 7.2.4(e) Width per Current Calculations

As the designed PCB is a 2 layer, there are only external traces and external layers. Using the calculated values above, a bus was made to power components throughout the PCB to ensure that all components have a proper amount of supplied current, otherwise there may be bottleneck issues and components lacking power to work properly.

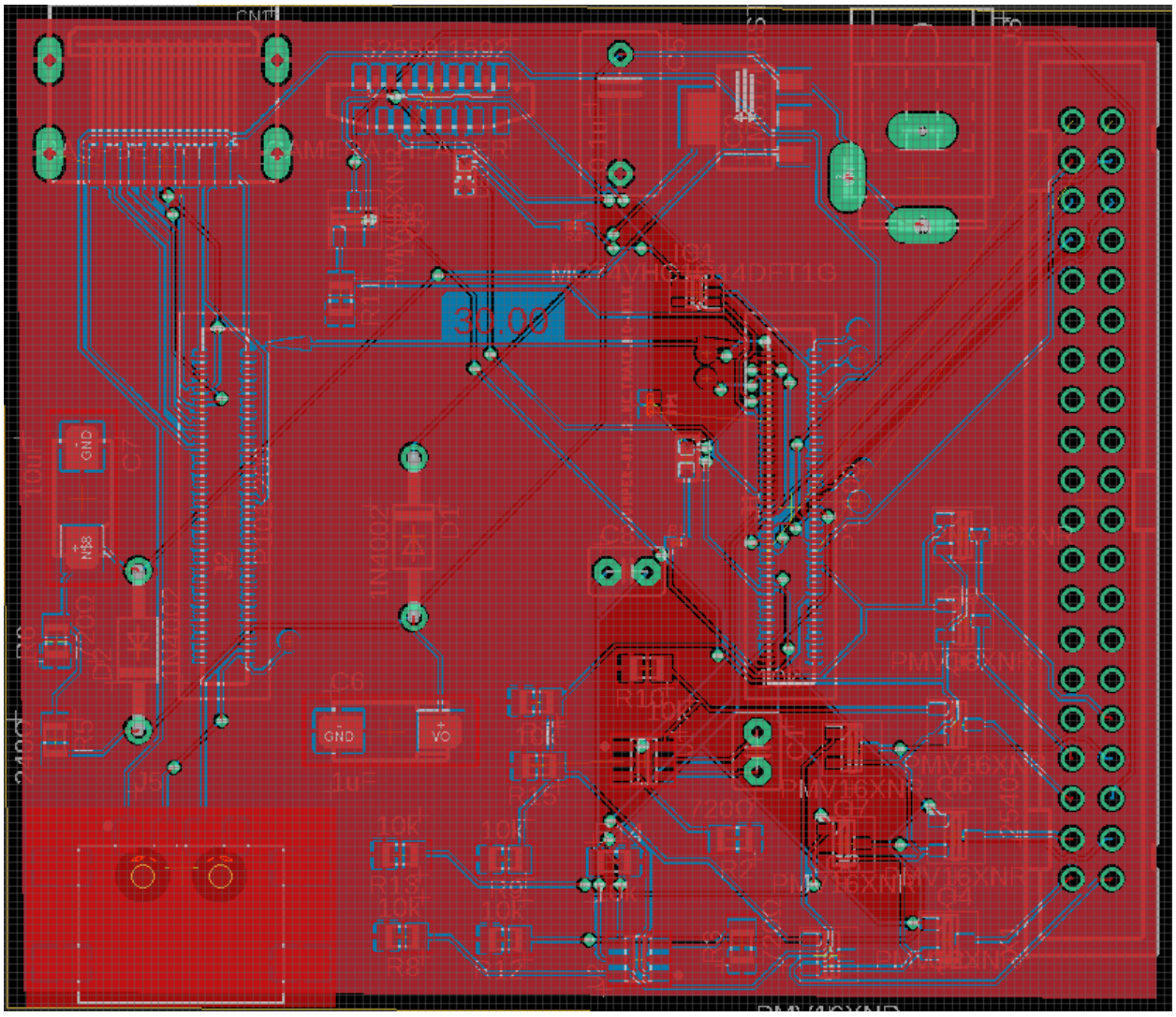


Figure 7.2.4(d) Raspberry Pi Carrier Board Mask

Curved edges were made and will be cut to smooth the edges as shaped by the red mask grounding plane above in Figure 7.2.4(c). It can be noticed on the left-center of the PCB that part of the board does not have the grounded mask. Due to size constraints set by the manufacturer these areas could not be reached, however for the purposes of this device, the entire board does not need to be grounded. Several 1mm Fiducials are added throughout the masked PCB design to help the placing machine in the factory to align the board and create the design.

Referring to the front and back design displayed on the PCB, all of the line tracing and position of components can be seen as placed on the circuit board. The missing of 3D components is due to many of the components being used have a symbol and footprint available but no 3D model. Upon inspection the majority of routes are found between the GPIO pinout from the Compute Module to the four 3008 digital to analog converter chips which each use a total of 4 GPIO and 4 source connections and outputting 8 outputs, thereby totaling 32 output connections.

8.0 Project Prototype Testing Plan

8.1 Hardware Test Environment

The test environment will consist of bread board testing until all parts are ready to be assembled. Once tested, the parts will be assembled and tested as each major assembly is attached.

For each layer of the design, all individual components will need to be tested. This includes the LiDAR, motors, accelerometer, PCB board, camera. After each part is verified to work, the system will be assembled and testing will occur as a part is added on to the final assembly. When all parts are assembled, the final product will go through testing to work out any bugs, or to improve the system.

The environment for testing will most be conducted in the lab initially where there is access to breadboards, power supplies, as well as other smaller components necessary to connect the components together. The lab also provides the necessary tools to measure all our components and verify operation. After all components are verified to work, they will again be tested each time a part is added to the final assembly to work. This way any issues caused to the working assembly will be easier to identify. Once everything is assembled, the final assembly will be tested in a controlled outside environment where common obstacles can be found. Different types of obstacles will be tested, and the correct motors will go off based on where the obstacle is.

The senior design lab will have most of the equipment required for the testing of all electrical hardware. This equipment includes Tektronix Oscilloscopes, Tektronix Dual Arbitrary Function Generators, Tektronix DMM 4050 Digital Multimeters, Keithley 2230-30-1 Triple-Channel Power Supplies, Soldering and Desoldering Stations and other small equipment like breadboards, small electronics and wires. The innovation lab will also be utilized for the 3D printed resources and the general lab space.

Safety precautions will have to be made when testing outside of the lab. When doing a trial where the user is blindfolded, a person will need to be close by to assist if the device does not convey the direction well enough for the user to understand it. The obstacles should also start off very simple and progressively get more difficult as the device improves and the user learns how it works.

User Guide

The user guide, as outlined below, will be adjusted during the testing process. This will act as a basic guide to outline the basics of what the vest is expected to do and how the actual user will be expected to interact with the device. Making sure that the vest is kept to simple operation will be important

Overview

Welcome to Depth Perception Haptic System (DPHS). This device is able to assist those who are visually imparied to navigate the environment around them.

WARNING

• Do not place the charger, plug or batteries in water or liquid or store where it can fall or be pulled into a tub or sink. Do not reach for it when it falls into water. Take off immediately if wet.

• Avoid sharp bending, twisting or squeezing of the jacket or charging cord.

• Does not use without supervision

ADVISORY: This should not be a primary sensing device. A guide is recommended for assistance. Test before use.

Important Information

.•Periodically check the entireproduct/cords/accessoriesfor damage. A damaged or non-functioning unit should no longer be used. If theproduct/cords/accessoriesare damaged, stop use. Do not modify or repair the product. This may causefire, electric shock or injury.

.• Children shall not play with the appliance.

• Use this product only for its intended use as described in this manual.

• Only use the charger and/or special cord provided.

• Keep sensors free of obstruction for optimal use

First Use

Please turn on the device using button 1 and wait for start up to run the motors and the speaker to tell the status of the device. Practice with devices with known obstacles to test out the device to familiarize yourself with the operation of the vest.

Charging and operating

• To charge the battery remove them from the vest and charge externally.
• The charge level is indicated by the LED on the charger. Once it is fully charged the display turns off
• If the battery is running low the speaker will sound off the battery percentage. Once the battery is empty, the device will power down and speak through the speaker.

Using your Vest

Turn on the device via the buttons in the bottom right corner. All motors will vibrate and the speaker will indicate the power is on. Once that has initialized the vest will begin to read the environment around you.

Note: This device takes time to learn how to operate. Make sure to have assistance with training and operation. Make sure batteries are fully charged before going long distances

Features

This device will measure distances and acceleration of objects around the user. It is important to keep those sensors free of obstruction when operating.

This device features a low power mode. In this mode the additional devices that measure distance and acceleration of objects away from the user will turn off. In this mode the user will only be using the camera to gadge obstacles. This is suggested to use only for emergencies. Discretion is advised.

Cleaning recommendations

Spot clean fabric with damp cloth. DO NOT SUBMERGE IN WATER OR PUT IN WASHING MACHINE. Product contains electronic parts and could cause damage to a person when wet. If vest is wet, do not turn on

Environmental Notice

Product contains batteries and/or recyclable electronic waste. For envi-ronment protection do not dis-pose in household waste, but for recycling take to electronic waste collection points provided in your country (Oral-B)

Troubleshooting

This section will be elaborated on in the final document. Any of the possible user interface problems should come forward and hopefully be solved in the testing phase. If any of those persist or a common user error is noticed, these will be recorded in the table below

With any technical problems, always try turning the device off and waiting 30 seconds until attempting to turn the device back on

Problem	Possible Reason	Remedy
Vest will not turn on	Battery is dead Button is stuck	Recharge Battery Press button a few time to let it up

Table 8.1 Troubleshooting

8.2 Hardware Specific Testing

For each of the motors being used in this project, a voltage will be applied to test them individually. This will be to verify that each motor is in working order to remove that variable from the debugging process. The speakers will also be tested in a similar way. Once it is verified that all devices are working, they can then be added into larger sub assemblies and tested there as well. When testing the individual parts, the code developed for the project can also be tested after the initial testing for the device is done. Once it is verified that the code works on the device, It can then be integrated with other hardware components to form the final assembly.

8.2.1 Camera Testing

Camera testing will be relatively simple. The camera will be connected to the Raspberry Pi to verify that the video feed will come on. The board has a camera interface, but it has 22 pins while our Raspberry Pi camera has 15 pins. Thankfully, the adapter for this is very simple to find and implement. The schematic can be seen below.

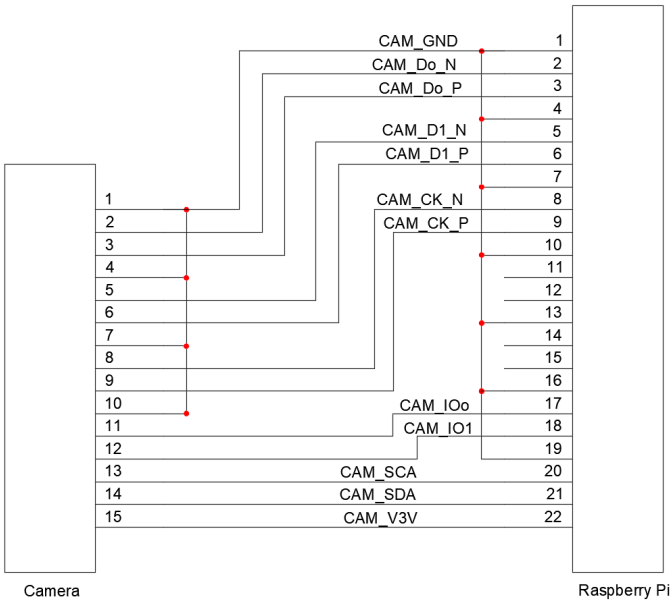


Figure 8.2.1: 15-22 Pin Adapter Schematic for Camera

Once this initial test is done, the computer vision elements will be tested with this hook up before the whole assembly is attached. This will be good for finding any type of errors present in the code and make for a simple testing environment for the computer vision side of the project. Doing a lot of debugging and problem solving at the individual part and assembly levels will allow for easier time problem solving issues with the overall assembly and final prototype.

8.2.2 LiDAR Testing

The initial LiDAR testing will involve simply verifying operation with the built in code and using a ruler to verify correct distances. The basic Raspberry Pi 4 Model B will be used to test this device. The pinout and connections for both can be found in the diagram below. When it is shown to work, it will be added into the main assembly. In the assembly, a similar test will be used to test the LiDARs function.

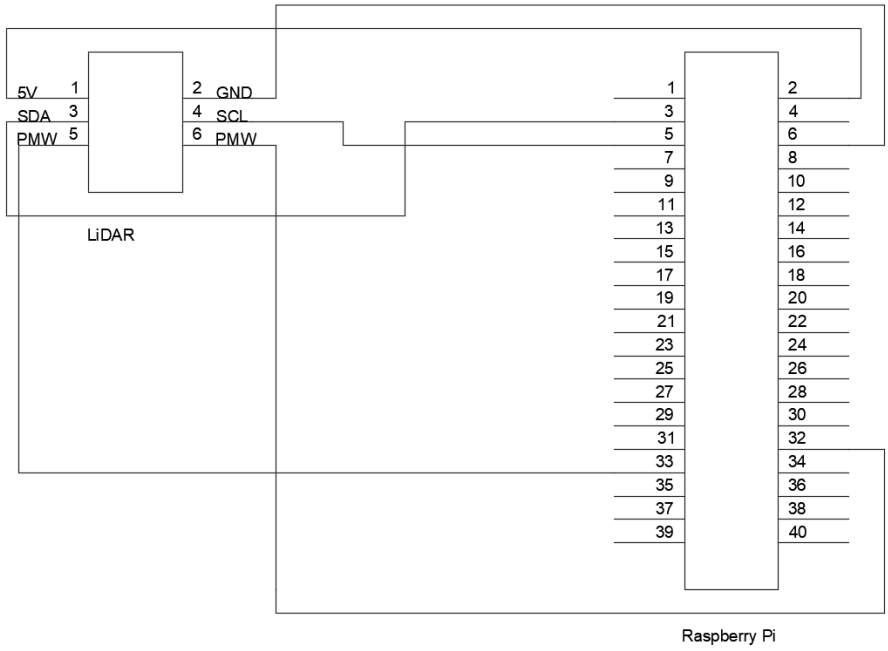


Figure 8.2.2: LiDAR Testing Pinout

8.2.3 Accelerometer

For the Accelerometer, initial testing will be set up on the Raspberry Pi Model B. The pinout and connections can be shown below. The factor code will be used to test if the device is working correctly. It will be tested by accelerating objects towards it and seeing if it is able to pick up the objects coming towards it.

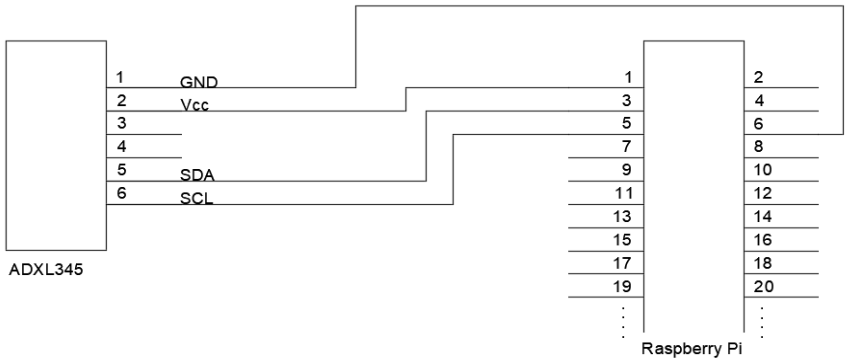


Figure 8.2.3: Accelerometer Testing Pinout

8.3 Software Test Environment

Test environment will consist of personal computational devices, laptops, desktops, etc.; The senior design lab in ENGI, and outdoors or in a well obstacles setting. As stated above, the lab will be one of the most used testing locations, especially in the production stages, where all the components and prototypes will need to be tested with equipment found there, such as a voltmeter to measure how much voltage is getting passed to the motors. Towards the final steps of the project, field tests can be conducted outdoors, walking around on campus or through a building, for example, more specific locations will be considered further into the project.

Also when it comes to volunteers, those would most likely be members of group 40 or any other senior design students in the lab. Due to IRB regulations, getting actual volunteers could prove difficult, thus testing may remain in house.

8.4 Software Specific Testing

For software specific testing, tools will be used to evaluate system functionality. Most tests on a functional level will be conducted manually, though some software at the coding level may be conducted automatically. So with a combination of both, software tests, such as, Integration tests, functional tests, unit tests, performance tests, acceptance tests, end-to-end tests, and smoke tests, will be performed on the device to ensure its effectiveness. (Atlassian, 2021)

Integration tests refer to ensure that the services and modules of the software work well together and prove to be cohesive.

- Referring to the class diagram in 6.4.3, conducting integration tests on all the services to ensure they are successfully able to communicate with the input devices and effectively translate information needed to each routine
- Each of these components successfully communicating the needed information for proper flow of the system will fulfill proper integration

Functional tests refer to verifying the specific output of a system, ensuring the device is functioning properly.

- Referring to the requirements section and the test cases below, functional testing will be a significant part of ensuring the expected results are observed, such as setting a timer before the DepthPerceptionService is run and stopping the timer after a voltage is registered in the motors and expecting the timer to read within a specific timeframe.

Unit tests refer to low level application tests of the individual functions, classes, components of the software. These can be automated with test scripts.

- Individual functions, such as GetCameraImage(), GetAngledLiDAR(), etc. can be tested in this manner

Performance tests refer to evaluating the system's behavior when it's under significant load. These tests evaluate the stability and reliability of the device's functionality, and can be costly but help developer's get a good understanding of how the system will react to certain scenarios.

- Refer below to Table 8.5(b), tests such as the priority management test would fall under this category.
- Thorough performance tests will be conducted on the device throughout the project

Acceptance tests refer to tests replicating user behavior to verify the system satisfies expected requirements. End-to-end tests refer to, similarly to acceptance tests, replicating user behavior, however with the software within an application environment.

- User performed test will fall under these tests

Smoke tests refer to testing the functionality of the entire system, typically after each build and will be a progressive method of testing.

- These will be used periodically with each stage of building the system to ensure proper functional flow

All these methods of software testing will be used throughout the duration of the project.

8.5 Test Cases

Requirement Specific Test Cases				
No.	**Statement**	**Test Scenario**	**Value**	**Unit**
1.X	Physical Attributes			
	Lightweight	Will use a scale to weigh the entire device throughout production to ensure it remains under the desire weight value	< 10	lbs
	Compact	Use Measuring tape or ruler to measure the dimensions of the vest and ensure it remains under the thickness threshold	< 2	in
	Durable	Will elevate it above different floorings(concrete, wood, grass, sand) and drop it from various heights under 6 feet and record how the device is affected	≥ 3	ft
2.X	Usage and Maintenance			
	Quick to arm and activate	Set up a timer and perform a few test runs of different people arming and activating the device, with the goal of the device powering on within 60 secs on average. Record times	< 60	s

No.	Statement	Test Scenario	Value	Unit
	Full Range of Motion (ROM)	-Full normal Range of motion of the shoulders, determined by the American Academy of Orthopedic Surgeons (AAOS), is a flexion and abduction of 180 degrees and external rotation of 90 degrees. -Have user wearing the device, attempt flexion, abduction, and external rotation and record the angles they are able to maintain.	< 10	% diff.
	Long-lasting battery	Have the device be worn and in active use for at least 4 hours and record the battery the device has been able to maintain	> 4	hrs
3.X				
	Mode indications	Ensure that each mode of the device has its own output pattern by testing signaling the system to switch into each mode	3	ct
	Quick to learn	Have individuals outside the group attempt to navigate avoiding obstacles and record the time it takes them to get used to the system and complete the limited obstacle course	< 10	min
4.X	General Navigation Mode			
	Haptic feedback that encodes distance	-Provides a relative indication of distance (via feedback intensity) of arbitrarily shaped objects visible to the camera over a discrete range of intensities (including zero). -Record each voltage delivered to the motors and ensure it increases with an obstacle getting within certain distances	≥ 3	ct
	Haptic feedback that encodes location	-Provides a relative indication of location (via location of feedback) of arbitrarily shaped objects visible to the camera at a number of feedback units on the device. -Test to ensure relative location of obstacle causes the corresponding region of the motor units to activate	≥ 6	ct

No.	Statement	Test Scenario	Value	Unit
	Restricted range	Vary the distance of a large object, such as a car, and ensure that the device doesn't register a response for its detection until it's with the desired detection threshold	< 20	ft
	Quick response time	Set up a timer to record how long it takes between an arbitrarily large object being placed within detection range and output alert being activated and felt by the user. Desired average response time to be within desired range.	< 900	ms
	Alert to proximal objects	Place object within certain close distance and observe that it behaves in expected manner	< 3	ft
5.X	Outdoor Navigation Mode			
	Curb detection	-Curbs (rapid inclines/declines in terrain) that appear in front of the user triggers an alert. -Have user approach a elevated platform, such as a sidewalk curb at UCF, an have the user be able to successfully navigate stepping up or down accordingly	5 ± 1	ft
	Curb description	-Curb alerts indicate the (1) steepness and (2) direction (incline/decline) of the curb. -Test to ensure that the device outputs a distinct alert combination for stepping up and stepping down	True	

Table 8.5(a) Requirement Specific Tests

Additional Test Cases				
No.	**Statement**	**Test Scenario**	**Value**	**Unit**
1.X	Priority Management			
	Button press Vs. Emergency detection	Should a user push a button, such as to change the state of device, and at the same time a large object is detected at close range, ensure that the routine given higher priority is performed		
	Detection in low power mode	Put device in low power mode and ensure that only emergency services are active by trying performing tasks that would normally warrant a response from the device in general mode, record if it behaves as expected		
2.X	Weather Conditions			
	Heat within range	Ensure the device functions within the range of heat the system can bear by placing it in front of a air vent set to just below max temperature	< 80	Celsius
	Cold chill within range	Ensure the device functions within the range of cold the system can bear by placing it in front of a air vent set to just above max temperature	>20	Celcius
	Moisture exposure within range	-If it rains, take the device out right after the rain finishes. -Take the device out in the morning when fog is out. -Ensure the device still functions properly		

Table 8.5(b)Additional Tests

9.0 Administrative Content

This section will review the budget for this project and will review the timeline for the next semester where the project will be developed and tested. This will also review the milestones that have been reached this semester in Senior Design I.

9.1 Milestone Discussion

Looking back at the milestones and goals planned for this semester, our team has done very well to keep on top of those. In our initial planning, extra time to each milestone was a critical factor so that overall implementation could be reviewed early, or in the event of lacking progress, there would be enough time before the actual deadline to get things done. Planning these soft deadlines ended up being incredibly helpful for completing the paper on time.

The first major milestone that needed to be completed was the actual project idea. This often seems like the hardest part of the process. There were many different options to consider since the designers for the project included two electrical engineering and two computer engineering majors. The design choice of the projects was initially discussed to be relatively complex by both hardware and software standards. Also a major consideration was the fact that a sponsor was collectively unwanted while designing the device, so the design can have more flexibility and not be controlled by an outer source. With this decided, it meant that the project would have to be self funded. Taking all of this into consideration, as well as our two semester timeline, many ideas were considered. There were many difficulties of having a project that was too complicated in the hardware design, or with the software design. The current project idea ‘Depth Perception Haptic System (DPHS)’ was the overall choice and found agreed by all members within the span of the first week of brainstorming.

Next milestone was the discussion of basic design choices. We started with the look and design of the actual product. We were able to narrow it down to 3 choices of either a sash, a vest or a jacket. Though the sash was favored in the beginning for it’s small size, it was decided that it might not have enough area for the amount of sensors or equipment we intended to use. If we had more time or a larger budget, it might have been more feasible. We eventually settled on a jacket design that would allow for more placement areas for sensors, motors and other equipment. We did discuss many options for sensors including ultrasonic and LIDAR, but agreed early on that the main input sensor would be the camera.

The next two milestones were the Divide and Conquer 1.0 and 2.0 papers. We used this paper to get a clear direction as to how we would begin the design of our project. We wanted a general overview for us to be able to launch off of for the rest of the paper. It was also a good layout for rudimentary research and making sure we were all on the same page for the design.

During the next milestone, the 60 page draft, our project underwent the most research and development. This draft began to dive deep into the hardware and software design of the project. This was where we began planning the PCB design and going through the different types of techniques for computer vision. While designing the PCB we also began the parts selection process. The current availability of parts was a huge consideration when choosing, as many parts were our first choice, and were often out of stock. We started with the idea of using a Raspberry Pi or Jetson Nano to be the brains of the project, and tried to lean towards components that

would be easily compatible with those. The PCB design was also a challenge, due to limited previous exposure. Learning about which files were necessary for the board house to have all the information to manufacture. On top of that the design itself was quite challenging. We had to make sure there was clear communication between the Electrical and Computer side of design.

The last milestone of this semester will be the completion of this paper, which will be the guide for how our assembly and testing will begin next semester. This paper will act as the blueprint to make next semester start out smoothly. Our Parts have been ordered by this point and will be coming in over the break.

Over the winter break, we will continue with the development of the code to start the project, as well as creating the physical jacket. So that in the spring, we can begin testing our design as soon as possible and have plenty of time to work out any bugs that will come up. Breaking down next semester, It is expected to look like the table below.

Our first two milestones for next semester will be broken between our hardware and software team. The first week will be used for testing parts and assembling the hardware. The software team will be working on the software design over the first month so that in february we can begin to integrate the hardware and software. During that time we will be debugging and issues of communication between the hardware and software. After that we will be testing to make sure the design will work as it’s supposed to, running trails to see if we can navigate an area. Making sure to document all of this processes as required

Once the dates for the presentations and reports are out, the dates for our milestones will be adjusted. Integrating the Hardware and software will most likely prove to be the hardest challenge next semester. Just getting the initial design to start working in any capacity will be the biggest milestone to overcome. After that, debugging and improving on what we are able to get working will be the next greatest challenge.

Next semester might prove to be a challenge for us due to external factors as well. Many of our group intend to look for jobs during this semester for when we graduate. As well as continuing to manage our time wisely to be able to complete the planned milestones at the intended date. This will have to be monitored carefully if we want to be able to complete our project with extra time to spare for emergencies or any other problems that may arise.

Dates are a tentative estimation for next semester. We intended to leave plenty of extra time to debug any problems with integration of hardware and software and for testing our design. Exact dates will be provided next semester, and our deadlines will be adjusted accordingly.

Senior Design II					
1	Test Components	1/3	1/10	Started	Chad & Kathryn
2	Design Initial Software Infrastructure	1/3	2/7	Started	Cris & Christa
3	Assemble Initial Hardware Design	1/10	2/7	Not Started	Chad & Kathryn
4	Integrate Hardware and Software	2/7	3/7	Not Started	Group
5	Testing & Redesign	3/7	3/31	Not Started	Group
6	Finalize Protype	3/31	4/12	Not Started	Group
7	Peer Presentation Final Presentation	TBA	TBA	Not Started	Group
8	Final Report	TBA	TBA	Not Started	Group
9	Final Presentation	TBA	TBA	Not Started	Group

Table 9.1 Milestone Update

9.2 Budget and Finance Discussion

The original budget for this project was a very broad estimate. As the parts are ordered, the budget has been slightly adjusted. Our current goal is to stay below $500 for this project, the current cost can be seen below. Currently, we should be below our budget for this project even when adding in the parts that havent been ordered and a small amount of extra as a buffer. Assuming nothing catastrophic happens we should come in about $100 under our budget.

Looking at individual part selection, there were many parts that were out of stock. For example the LiDAR selection, the availability of the part forced us to go with a more expensive LiDAR, going to the top of our original budget for this part. The more advanced LiDAR might open the door to more available functions in the future.

The Raspberry Pi Compute 4 Module also is proving a challenge to find. Ideally, we want the Compute module to have 4GB or RAM, however we might be limited by what's in stock at the moment. Currently it has been put on back order and will hopefully be able to get here by January. Many of the Raspberry Pi, Adruino, and Jetsen products are scarce and hard to find at the moment. This will most likely affect the final budget if the module gets delayed any more.

As stated in the parts selection, a high resolution camera was not required for this project. This allowed us to look at cheaper options for cameras. The Raspberry Pi camera was a great balance between cost and quality and will work great for what we need. This camera was also difficult to find, being out at most vendors. Thankfully it eventually became available about a week after the initial parts order.

Another big cost that wasn't taken into account into the original budget is that many of the parts required shipping cost on parts. Since availability was limited, we had to order from multiple sites, driving the shipping cost up. That will be added in for the rest of the parts that need to be ordered.

Looking at the table below, the current parts that have been ordered. The compute module ended up being the bulk of the cost as expected. The LiDAR also ended up being one of the larger costs to the project which even though it was budgeted in budget, we were hoping for something a bit less expensive. The accelerometer was the next big cost. Originally, we wanted an accelerometer that would be capable of SPI or I2C so that we would have options depending on what pins were available. It was also decided that a second LiDAR would most likely be necessary for the project, so a second was ordered. Unfortunately this is one of the more expensive parts that is needed.

Item Type	**Quantity**	**Actual Cost**	**Description**
LiDAR	2	$59.99	Garmin LiDAR
Accelerometer	1	$18.95	ADXL345
Camera	1	$25	Raspberry Pi Camera
Speaker	1	$10.69	MakerHawk Speaker
Haptic feedback	Set acquirement (20)	$16.04	Tatoko flat coin motors
Compute Module	1	$65	Raspberry Pi Compute 4 Module 4GB
Shipping		$37.93	Shipping on parts
Current Total (As of 11/19/2021)		=293.60	

Table 9.2.1(a) Ordered Parts

The next table shows the breakdown of what is still required for this project. We have less than $200 left to pay for, assuming that the costs are about what is estimated. These have a chance of being much more expensive than what is listed. Thankfully we do have that large cushion in the budget that should account for that. Our current estimated budget total is going to be around $380.

Item Type	Quantity	Estimated Cost	Description
Fabric	1	$30	Fabric for Jacket/Vest
Power Source	Set acquirement (2)	$40	Device Power (Battery)
Custom PCB	≤ 2	≤ $30	Prototype and final product motherboard
Additional Shipping	-	$15	-
Total Est. (As of 11/19/2021)	-	≈ $145	-
Total Est.+Current (As of 11/19/2021)	-	=$378.60	-

Table 9.2.1(b) Remaining Items to be Ordered

Sources

"4/17/19 HeadsUP FINAL." Performance by Duc-Quy Nguyen, Youtube, 21 April 2019, https://www.youtube.com/watch?v=NeZ7vfBvmes (Duc-Quy Nguyen, 2019)

Abigail K. Leichman. "The gadgets that enable blind people to see" ISRAEL21c, 12 September 2016, https://www.israel21c.org/the-gadgets-that-enable-blind-people-to-see/ (Leichman, 2016)

Adrian Rosebrock, "Image Gradients with OpenCV." PyImageSearch, May 12, 2021, https://www.pyimagesearch.com/2021/05/12/image-gradients-with-opencv-sobel-and-scharr/ (Rosebrock, 2021)

Allan, Alasdair. "Benchmarking TensorFlow Lite on the New Raspberry Pi 4, Model B." Hackster.io, 2019, https://www.hackster.io/news/benchmarking-tensorflow-lite-on-the-new-raspberry-pi-4-model-b-3fd859d05b98.

Arducam "Raspberry Pi Camera Connector Pinout and Type (MIPI CSI-2)." Arducam, 2021, https://www.arducam.com/raspberry-pi-camera/connector-type-pinout/ (Arducam, 2021)

Armenta, Antonio. "Safety Considerations for LiDAR Sensors." EETech Media, 22 March 2021, https://control.com/technical-articles/safety-considerations-for-LiDAR-sensors/ (EETech Media, 2021)

ATP Electronics Inc. "SD Card Requirements for Mission-Critical Applications" ATP Electronics, 8 April 2020, https://www.atpinc.com/blog/industrial-sd-cards-factors-requirements-to-consider (ATP Electronics, 2020)

Be my eyes, "Our Story." Specialized Help, 5 October 2015, https://www.bemyeyes.com/about (Specialized Help, 2015)

Bedi, Aryan. "EyeCane: The Future of Mobility." EyeCane, 2021, https://www.eyecane.ca/copy-of-old-homepage (EyeCane, 2021)

Benson, Lisa "A Guide to United States Electrical and Electronic Equipment Compliance Requirements." National Institute of Standards and Technology (NIST), February 2017, https://nvlpubs.nist.gov/nistpubs/ir/2017/NIST.IR.8118r1.pdf (NIST, 2017)

Cadence PCB Solutions, "Guide on PCB Trace Length Matching vs Frequency." Cadence Design Systems Inc, 2020, https://resources.pcb.cadence.com/blog/2019-guide-on-pcb-trace-length-matching-vs-frequency (Cadence Design Systems Inc, 2020)

CanaKit Corporation, "Raspberry Pi Compute Module 4 (CM4)." *CanaKit*, 2021,

https://www.canakit.com/raspberry-pi-compute-module-cm4.html. (CanaKit, 2021)

Chuan-en Lin. “Introduction to Motion Estimation with Optical Flow.” Nanonets, 2019, https://nanonets.com/blog/optical-flow (Nanonets, 2019)

“Compute Module 4.” Raspberry Pi, https://www.raspberrypi.com/products/compute-module-4/?variant=raspberry-pi-cm4001000.

Das, Angel. “Convolution Neural Network for Image Processing — Using Keras.” Medium, 20 Aug 2020 https://towardsdatascience.com/convolution-neural-network-for-image-processing-using-keras-dc3429056306 (Das, 2020)

DroneBot Workshop, “Getting Started with LIDAR.” YouTube, 13 Jul. 2018, https://youtu.be/VhbFbxyOI1k?t=540.

ECFR, “Code of Federal Regulations, Title 10, Chapter 2.” National Archives, 27 May 2021, https://www.ecfr.gov/current/title-10/chapter-II/subchapter-D/part-430/subpart-C/section-430.32#p-430.32(z)(1) (National Archives, 2021)

Emmet. “Raspberry Pi Accelerometer Using the ADXL345.” *Pi My Life Up*, 16 Nov. 2019, https://pimylifeup.com/raspberry-pi-accelerometer-adxl345/. (Emmet, 2019)

ESRI, “Image Differencing Change Detection.” ArcGIS Solutions, Environmental Systems Research Institute Inc, 2021 https://solutions.arcgis.com/defense/help/image-change-detection/workflows/image-differencing/ (ESRI, 2021)

Foglia, Lucia, and Wilson, Robert. “Embodied Cognition.” WIREs Cogn Sci, vol. 4, no. 3, 8 Feb. 2013, https://wires.onlinelibrary.wiley.com/doi/10.1002/wcs.1226

“Gitflow Workflow.” Atlassian. https://www.atlassian.com/git/tutorials/comparing-workflows/gitflow-workflow.

Hafez A. Afify, “Evaluation of change detection techniques for monitoring land-cover changes.” Public Works Engineering Department, Faculty of Engineering, Tanta University, 30 July 2011, https://www.sciencedirect.com/science/article/pii/S1110016811000421#:~:text=Image%20differencing%20change%20detection%20technique,same%20band%20of%20another%20date.&text=The%20overall%20accuracy%20of%20the,and%20the%20KHAT%20is%200.24. (Afify, 2011)

Halfacree, Gareth. “Eloquent Arduino’s Benchmarking Puts the Raspberry Pi Pico Bottom of the Pile for TinyML Performance.” Hackster.io, 2021, https://www.hackster.io/news/eloquent-arduino-s-benchmarking-puts-the-raspberry-pi-pico-bottom-of-the-pile-for-tinyml-performance-650b33941774.

"Haptic Feedback." *Precision Microdrives*, 28 Oct. 2021, https://www.precisionmicrodrives.com/motors/haptic-feedback. (Precision Microdrives, 2021)

Heath, Nick. "Your Raspberry Pi 4 May Have Just Got an Unexpected Speed Boost." ZDNet, ZDNet, 2 July 2019, www.zdnet.com/article/your-raspberry-pi-4-may-have-just-got-an-unexpected-speed-boost/. (ZDNet, 2019)

HeathenHacks. " Pins, 32 LEDs, 2 LED Drivers [TLC5940]." LGPL, 28 April 2019, https://create.arduino.cc/projecthub/Heathen_Hacks-v2/5-pins-32-leds-2-led-drivers-tlc5940-77f915

"How To Use A White Cane - The Basics." Performance by Unsightly Opinions, Youtube, 27 May 2021, www.youtube.com/watch?v=UQETnkwTg5A. (Unslightly Opinions, 2021)

Howard, Andrew, et al. "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications." ArXiv, 2017, https://arxiv.org/abs/1704.04861.

IBM Cloud Education. "Convolutional Neural Networks." IBM Corporation, 20 October 2020, https://www.ibm.com/cloud/learn/convolutional-neural-networks (IBM, 2020)

Ignatov, Andrey, et al. "Fast and Accurate Single-Image Depth Estimation on Mobile Devices, Mobile AI 2021 Challenge: Report." ArXiv, 17 May 2021, https://arxiv.org/pdf/2105.08630.pdf.

"Jetson Nano Developer Kit." Nvidia Developer, https://developer.nvidia.com/embedded/jetson-nano-developer-kit.

Kopf, Johannes et al. "Robust Consistent Video Depth Estimation." ArXiv, 2021, https://arxiv.org/pdf/2012.05901.pdf.

"LaunchX 2019 EyeCane Final Pitch." Performance by EyeCane, Youtube, 5 April 2019, https://www.youtube.com/watch?v=ew5YDnrcgjw (EyeCane, 2019)

"LiDAR Depth Camera L515." Digi-Key, 30 Jun. 2020, https://www.digikey.com/en/product-highlight/i/intel/LiDAR-depth-camera-l515.

"Live Braille: the world's smallest, lightest travel aid for the blind" by Digit, Youtube, May 18. 2016, https://www.youtube.com/watch?v=UWGNUOzA7gc. (Digit, 2016)

Low Vision Md, "Aira: Explore your world like never before." Low Vision Specialists of Maryland, 2020, https://lowvisionmd.org/aira/ (Low Vision Specialists of Maryland, 2020)

Mansour, Mostafa, et al. "Relative Importance of Binocular Disparity and Motion Parallax for Depth Estimation: A Computer VIsion Approach." Remote Sensing, vol. 11, no. 17, 2019, https://doi.org/10.3390/rs11171990.

Marian, P. "HC-SR04 Datasheet." ElectroSchematics.com, 27 Mar. 2020,

https://www.electroschematics.com/hc-sr04-datasheet/.

Mathworks “Build a Map with LiDAR Data” , 2021, https://www.mathworks.com/help/driving/ug/build-a-map-from-LiDAR-data.html

“Monocular Depth Estimation.” Paperswithcode. https://paperswithcode.com/task/monocular-depth-estimation.

Nguyen, Duc-Quy, et al. “Headsup: Optical Mobility Aid.” *HeadsUP - Group 14 Photonics 2*, University of Central Florida, 21 Apr. 2019, https://www.ece.ucf.edu/seniordesign/fa2018sp2019/g14/. (Nguyen, 2019)

Oral-B. “Environmental Notice - Oral-B Io Manual.” Manuals Library, www.manualslib.com/manual/1909494/Oral-B-Io.html?page=23#manual.

Parker Software, “The Theory of Constraints in Software Development.” Parker Software, 12 Aug. 2020, www.parkersoftware.com/blog/the-theory-of-constraints-in-software-development/. (Parker, 2020)

Pilz, Sandra, “Intro to BlindSquare: iOS Navigation App.” Perkins Learning, 19 Febuary 2016, https://www.perkinselearning.org/technology/posts/intro-blindsquare-ios-navigation-app (Perkins Learning, 2016)

Pittet, Sten. “The Different Types of Testing in Software.” *Atlassian*, 2021,

https://www.atlassian.com/continuous-delivery/software-testing/types-of-software-testing . (Atlassian, 2021)

Plowman, David. “Machine learning and depth estimation using Raspberry Pi.” Raspberry Pi, 11 Feb. 2021, https://www.raspberrypi.com/news/machine-learning-and-depth-estimation-using-raspberry-pi/.

PowerStream. “Wire Gauge and Current Limits Including Skin Depth and Strength.” PowerStream Technology, 23 April 2021, https://www.powerstream.com/Wire_Size.htm

Preston-Werner, Tom. “Semantic Versioning 2.0.0.” https://semver.org/.

Raspberry Pi Trading Ltd. “Product Brief RPi CM4” Raspberry Pi, October 2020, https://static.raspberrypi.org/files/product-briefs/2010016+Product+Brief+RPi+CM4.pdf (Raspberry Pi, 2020)

Rossum, Guido van, et al. “PEP 8 -- Style Guide for Python Code.” Python, 1 Aug. 2013, https://www.python.org/dev/peps/pep-0008/.

“RP2040 Tech Specs.” Raspberry Pi, https://www.raspberrypi.com/products/rp2040/specifications/.

SDOT Photos. “Columbia City Sidewalk Repair Project.” flickr, 16 Dec. 2015, https://www.flickr.com/photos/sdot_photos/23168490614. Image. Used in its resized form, and colored over with gradients to depict depth. Licensed under CC BY-NC 2.0.

Sagar. “Applying IPC-2221 Standards in Circuit Board Design.” Sierra, 20 July 2021, https://www.protoexpress.com/blog/ipc-2221-circuit-board-design/

Sharma, Adamya, “Live Braille: A startup, a wearable and a new future for visually impaired.” Digit, 9 May 2016, https://www.digit.in/features/startups/live-braille-a-startup-a-wearable-and-a-new-future-for-the-visually-impaired-30132.html (Digit, 2016)

Silvester, Luca, et al. “ BackUp Buddy Wireless BackUp Solution.” *BackUp Buddy*, University of Central Florida, Nov. 2018, https://www.ece.ucf.edu/seniordesign/su2018fa2018/g10/. (Luca, 2018)

Stuff Made Here. “See in complete darkness with touch.” YouTube, 20 Jun 2020, https://youtu.be/8Au47gnXs0w?t=349.

Stroustrup, Bjarne and Sutter, Herb. “C++ Core Guidelines.” ISOCPP, 19 Aug. 2021, http://isocpp.github.io/CppCoreGuidelines/CppCoreGuidelines.

“Taking a Walk with My White Cane.” Performance by See It My Way, YouTube, 20 Aug. 2020, www.youtube.com/watch?v=Z_6s4-8tf0M. (See It My Way, 2020)

Tan, Daryl. “Depth Estimation: Basics and Intuition.” Towards Data Science, 13 Feb. 2020, https://towardsdatascience.com/depth-estimation-1-basics-and-intuition-86f2c9538cd1.

TensorFlow. “Why TensorFlow.” https://www.tensorflow.org/about.

“TFMini-S - Micro LiDAR Module.” Sparkfun, https://www.sparkfun.com/products/16977.

TSA Government, “Lithium batteries with more than 100 watt hours.” Transportation Security Administration (TSA), 2021, https://www.tsa.gov/travel/security-screening/whatcanibring/items/lithium-batteries-more-100-watt-hours (TSA, 2021)

ULTRACANE, “The UltraCane - an award winning primary mobility aid.” UltraCane, January 2011, https://www.ultracane.com/about_the_ultracane (UltraCane, 2011)

“Ultrasonic Ranging Module HC - SR04.” ElectroSchematics.com, ElecFreaks, https://www.electroschematics.com/wp-content/uploads/2013/07/HCSR04-datasheet-version-1.pdf.

Upton, Eben. “tinyML Talks Eben Upton: Inference with Raspberry Pi Pico and RP2040.” YouTube, 4 Mar. 2021, https://youtu.be/xTI4UeEOzkY?t=1955.

USCODE GPO “Title 42 Chapter 77 The Public Health and Welfare.” U.S. Government Publishing Office, 2013, www.gpo.gov/fdsys/pkg/USCODE-2013-title42/pdf/USCODE-2013-title42-chap77.pdf (U.S. Government Publishing Office, 2013)

V Skills Verified, “Software and Hardware Requirements.” Software and Hardware Requirements, V Skills Verified, 2 Apr. 2019, www.vskills.in/certification/tutorial/software-and-hardware-requirements/. (V Skills, 2019)

Verma, Shiva. “Understanding Input Output shapes in Convolution Neural Network | Keras.” medium, 31 Aug 2019, https://towardsdatascience.com/understanding-input-and-output-shapes-in-convolution-network-keras-f143923d56ca

Wasser, Leah. “The Basics of LiDAR - Light Detection and Ranging - Remote Sensing.” neon, 7 Oct 2020, https://www.neonscience.org/resources/learning-hub/tutorials/LiDAR-basics

Zhang, Ziyu, et al. “A Simple Baseline for Fast and Accurate Depth Estimation on Mobile Devices.” 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2021, https://openaccess.thecvf.com/content/CVPR2021W/MAI/papers/Zhang_A_Simple_Baseline_for_Fast_and_Accurate_Depth_Estimation_on_CVPRW_2021_paper.pdf.

Zm Peterson, “Guide to PCB Trace Length Matching in High Speed Design." Northwest Engineering Solutions LLC (NWES), 8 April 2020, https://www.nwengineeringllc.com/article/guide-to-pcb-trace-length-matching-in-high-speed-design.php (NWES, 2020)

Appendix

10.1 Appendix A: Copyright Permissions

Unless otherwise noted, all figures and tables above were created by the members of this group. Citations are provided to denote otherwise.

10.2 Appendix B: References

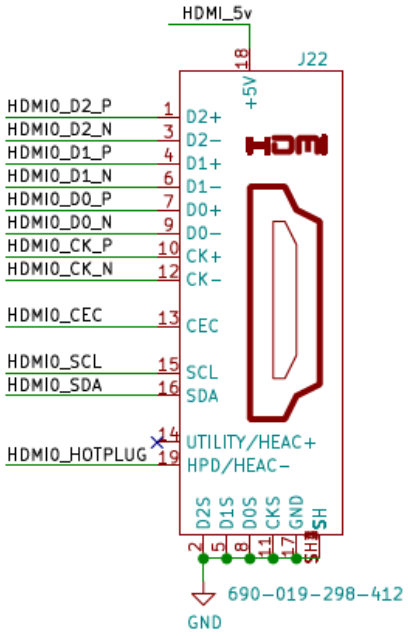
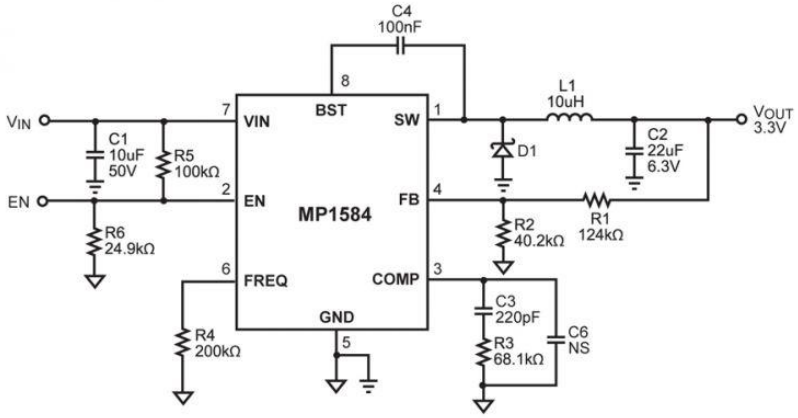
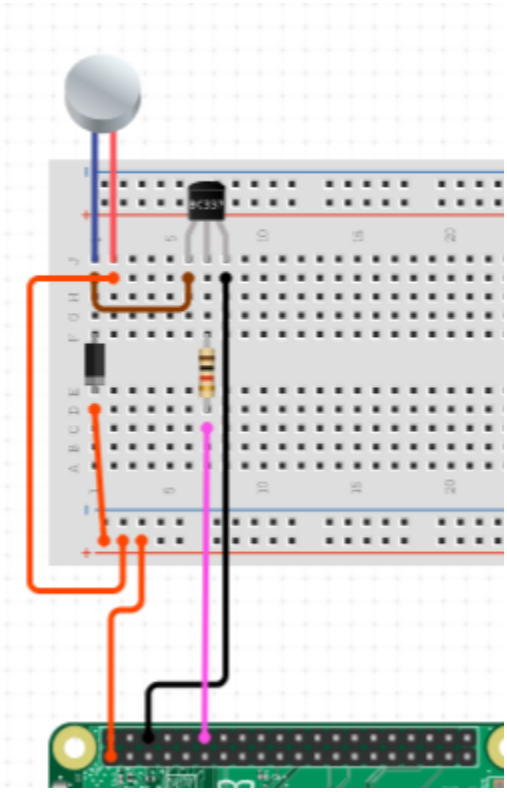


Figure 10.2.1 HDMI Pinout

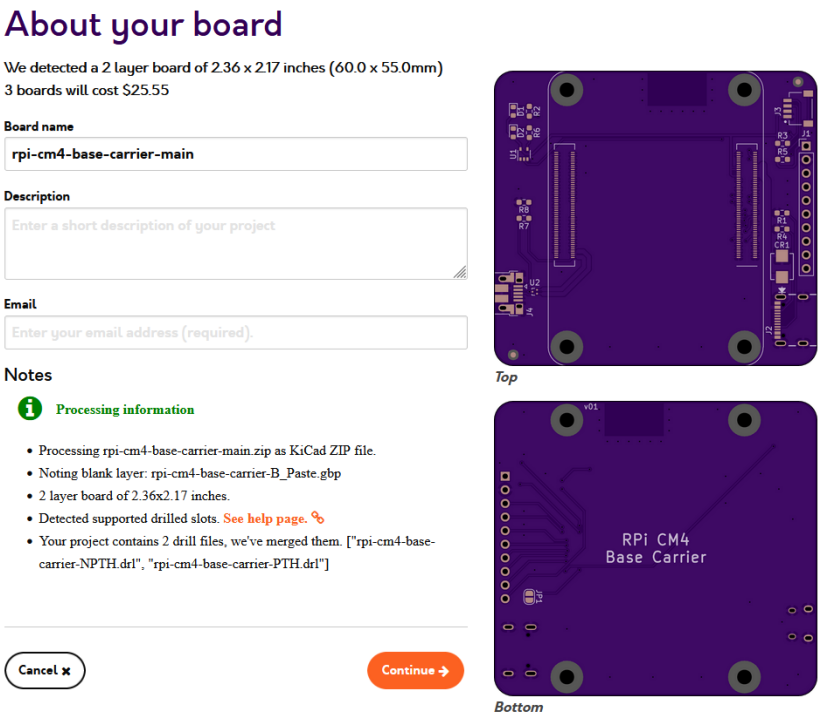


10.2.2 Voltage Regulator Reference



10.2.3 Haptic Feedback GPIO Reference

10.3 Appendix C: Shipment Info



10.3.1 Base Model of Carrier Board Cost

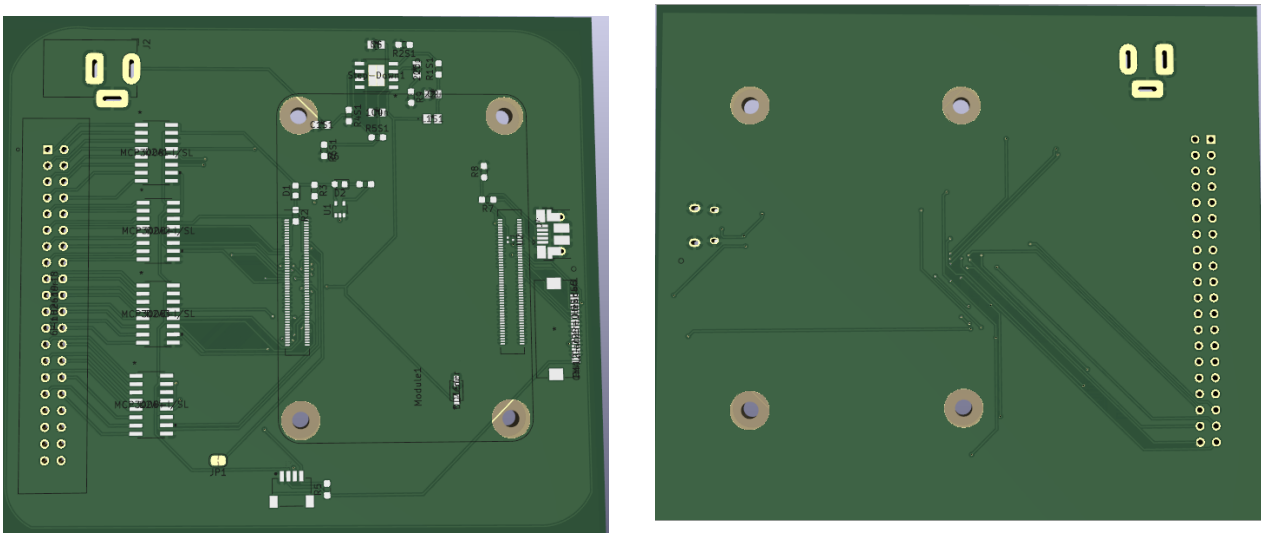


Figure 10.3.2 Raspberry Pi Carrier Module 3D View; Front (Left) and Back (Right)

PCB BOM								
Value	Package	Order code	Manufacturer	Manuf. Code	Availability	Price (from)	Description	
	C2.5-3	unknown						
0.1uF	C075-052X106	72W1753	TDK	C1005X7R1H104K050BB	1050000	0.021	TDK - C1005X7R1H104K050BB - SMD Multilayer Ceramic Capacitor, 0.1 µF, 50 V, 0402 [1005 Metric], ± 10%, X7R, -55 °C	
1uF	085CS_1AR	72W1753	TDK	C1005X7R1H104K050BB	1050000	0.021	TDK - C1005X7R1H104K050BB - SMD Multilayer Ceramic Capacitor, 0.1 µF, 50 V, 0402 [1005 Metric], ± 10%, X7R, -55 °C	

10uF	085CS_1AR	47AC4867	PANASONIC	EEUEB1J100SH	633	0.138	PANASONIC - EEUEB1J100SH - Electrolytic Capacitor, 10 µF, 63 V, ± 20%, Radial Leaded, 5000 hours @ 105°C, Polar	
	HDMI_MOLEX_47151-0001	92M4040	MOLEX	47151-0001	unkown	1.65	MOLEX - 47151-0001 - 1.0 HDMI R/A HEADER ASSY / Molex	
1N4002	DO41-10@1	05R5979	VISHAY	1N4002-E3/54	unkown	0.064	VISHAY - 1N4002-E3/54 - Standard Recovery Diode, 100 V, 1 A, Single, 1.1 V, 30 A	
MC74VHC1G14DFT1G	SOT65P210X110-5N	81Y6710	ONSEMI	MC74VHC1G14DFT1G	unkown	0.087	ONSEMI - MC74VHC1G14DFT1G - INVERTER SCHMITT TRIGGER, SOT-353-5	
LM317MBSTT3	SOT223	29X6007	ONSEMI	LM317MBSTT3G	unknown	0.256	ONSEMI - LM317MBSTT3G - Linear Voltage Regulator, Adjustable Positive, 1.2V to 37V/500 mA Out, SOT-223-3	
P1-100	DF40C100DS04V51	31W9017	VISHAY	MCW0406MD1003BP100	unknown	0.295	VISHAY - MCW0406MD1003BP100 - SMD Chip Resistor, 100 kohm, ± 0.1%, 250 mW, 0406 [1016 Metric], Thin Film, Precision	
P101-200	DF40C100DS04V51	unknown		Digikey	unknown	unkown	Chip	
	SPC4077	unknown		Digikey	unkown	unknown	Chip	

	6.29E+11	15AC9040	WURTH ELEKTRONIK	Unknown	228	1.21	WURTH ELEKTRONIK - 629104190121 - USB Connector, USB Type A, USB 2.0, Receptacle, 4 Positions, Surface Mount, Horizontal
JUMPER-SMT_2_NC_TRACE_NO-SILK	SMT-JUMPER_2_NC_TRACE_NO-SILK	unknown		unknown	unknown	unknown	connector
	CHIPLED_0603	32AC1319	OSRAM	LS Q976-NR-1	52000	0.027	OSRAM - LS Q976-NR-1 - CHIPLED 0603 SUPER RED ROHS COMPLIANT: YES
PMV16XNR	SOT95P230X110-3N	68Y7841	NEXPERIA	PMV16XNR	0	0.146	NEXPERIA - PMV16XNR - MOSFET Transistor, N Channel, 6.8 A, 20 V, 0.016 ohm, 4.5 V, 650 mV RoHS Compliant: Yes
1k	_0402MP	64R5430	PANASONIC	ERJ-8ENF1001V	34260	0.033	PANASONIC - ERJ-8ENF1001V - SMD Chip Resistor, 1 kohm, ± 1%, 250 mW, 1206 [3216 Metric], Thick Film, Precision
720Ω	M0805	52K8703	VISHAY	CRCW0603720RFKEA	0	unknown	VISHAY - CRCW0603720RFKEA - SMD Chip Resistor, 720 ohm, ± 1%, 100 mW, 0603 [1608 Metric], Thick Film, General Purpose

240Ω	M0805	65T8389	PANASONIC	ERJ-3EKF2400V	39312	0.008	PANASONIC - ERJ-3EKF2400V - SMD Chip Resistor, 240 ohm, ± 1%, 100 mW, 0603 [1608 Metric], Thick Film, Precision
10k	M0805	71M1030	VISHAY	CCF5510K0FKE36	6359	0.03	VISHAY - CCF5510K0FKE36 - Through Hole Resistor, 10 kohm, CCF Series, 500 mW, ± 1%, Axial Leaded, 250 V
52559-1592	52559-1592	unknown		Dikikey	unkown	unkown	Connector
TEST-POINT3X5	PAD.03X.05	unknown		Digikey	unkown	unkown	Chip
LINEAR_TECHNOLOGIES_LTC6992CS6-1#TRMPBFLINEAR_TECHNOLOGIES_LTC6992CS6-1#TRMPBF_0_0	LINEAR_TECHNOLOGIES_LTC6992CS6-1#TRMPBF_0	505-LTC6992CS6-1#TRMPBFCT-ND	Octopart	Digi-key	1888	4.27	Transistor
	PAK100/2500-40	unknown		Digi-key	unknown	unkown	Connector

Table 10.3.3 Requirement Specifications

10.4Appendix D: Original Requirements

No.	Description	Test(s)	Value	Unit
1.X	Power System			
1.1	Long-lasting battery	Lasts at an estimated 80% haptic feedback output with all input devices active.	> 4	hrs
1.2	Easy to charge	Rechargeable via a standard US 120 V electrical socket.	True	
*1.3	Audio to indicate power system state	Tones are played from a speaker to indicate: (1) powering on, (2) powering off, (3) battery level below 20%, and (4) battery level below 10%.	True	
*1.4	Low-power mode	Low power mode is activated after a period of no detected motion (or after user interface input).	≈ 20	s
2.X	User Experience and Interface			
2.1	Buttons to operate the system	Buttons must be available to (1) turn the system on and off, and (2) switch operating modes.	True	
2.2	Large buttons	Each button's surface area can contain braille.	> 0.7	in^2
2.3	Lightweight	Total weight, including the clothing which houses the electronic devices.	< 10	lbs
2.4	Unrestricted range of motion (ROM)	Arm ROM for any joint with the device present, is no different from deviceless ROM.	< 10	% diff.
2.5	Quick to arm and activate	Sighted user time to set up and turn on the device.	< 60	s
2.6	Quick to learn	Rate of collision against arbitrary objects (see requirement 3.1) after 10 mins of training.	< 20	% hit.
2.7	Durable	Remains intact and continues to function after being dropped onto tile flooring from specified height.	≳ 1	m
* Indicates Advanced goals, ** Indicates Stretch goals				
Note: All tests above will be conducted during daylight hours, in the absence of precipitation and fog.				

Table E(a) Requirement Specifications

No.	Description	Test(s)	Value	Unit
3.X	General Navigation Mode			
3.1	Haptic feedback that encodes distance	Provides a relative indication of distance (e.g., within 5 ft, within 10 ft, beyond 15 ft) of arbitrarily shaped objects (> 3 $ft.^3$ volume) visible to the camera.	> 80	% hit.
3.2	Haptic feedback that encodes velocity	Provides a relative indication of velocity (e.g., quickly approaching, slowly leaving, etc.) of arbitrarily shaped objects (> 3 $ft.^3$ volume) visible to the camera.	> 80	% hit.
3.3	Quick response time	Time the device takes to react to an object (from requirement 3.1) 10 ft away from the camera.	< 900	ms
3.4	Restricted range	No response to objects that are far from the camera.	> 20	ft
3.5	Alerts to proximal objects	Objects near the device trigger an alert.	< 3	ft
*3.6	Curb detection	Curbs (≲ 10 inch rapid inclines/declines) that appear in front of the device triggers an alert.	5	ft
*3.7	Curb description	Curb alerts indicate the (1) steepness and (2) direction (incline/decline) of the curb.	True	
**4.X	Outdoor Navigation Mode	A primarily computational system that will detect high-level entities such as crosswalks, walking paths, and roads.		
*5.X	Immediate Emergency Alerts	A primarily electrical system that will immediately alert (i.e., bypass computer processing) the user to dangers such as rapidly approaching objects.		
* Indicates Advanced goals, ** Indicates Stretch goals				
Note: All tests above will be conducted during daylight hours, in the absence of precipitation and fog.				

Table E(b) Requirement Specifications (continued)

10.5 Appendix E: Glossary of Abbreviations

ANSI	American National Standard Institute
API	Application Programming Interface
AWG	American Wiring Guage
CM4	Compute Module 4
CNN	Convolutional Neural Network
DPHS	Depth Perception Haptic System
Ebatt	Rated Battery Energy
ES	Embedded system
I2C	Inter-Integrated Circuit Protocol
IEC	International Electrotechnical Commission
MIPI	Mobile Industry Processor Interface
PCB	Printed Circuit Board
SD	External Memory Storage
SPI	Serial Peripheral Interface
UEC	Unit of Energy Consumption