Batpack

A companion for the visually impaired

Mike Ferraro, Christian Jackson, Lucas Pasqualin, and Connor Roggero

Dept. of Electrical Engineering and Computer Science,
University of Central Florida, Orlando, Florida, 32816-2450

Abstract – The goal of this project was to better the spatial awareness of visually impaired individuals, while at the same time increasing the mobility of the user. With nearly 100 years since the walking stick no improvements have been made to improve the quality of life of a visually impaired person. The Batpack uses an array of sensors that allows the user to pick up on obstacles around them; this includes ultrasonic, laser and computer vision. The information received from these sensors will alert the user using vibrations to keep their sense of hearing keen. All of this will be packed in an everyday backpack to be non-intrusive.

***Index Terms* – Machine vision, Image edge detection, Microcontrollers, Optical sensor, Haptic interfaces, Pulse width modulation, Wearable sensors, Ultrasonic variables measurement**

I. INTRODUCTION

Moore’s law has ensured that we are surrounded by evidence of always smaller, and always faster devices. However, one area that has remained relatively stagnant has been gadgets to aid the visually impaired. With no recent innovations in this field, visually impaired individuals have been forced to utilize the so-called ‘walking stick’. Not only does the walking stick hamper an individual’s use of a hand (and usually their dominant hand), but this clunky, obtrusive method of facilitating mobility for visually impaired people has been unchanged for hundreds of years. Using modern components, our aim is to build a device which offers not only the same benefits of the walking stick with none of the disadvantages, but also provides new innovations to aid the visually impaired in their day to day lives.

The Batpack is this device. A backpack in appearance, the Batpack offers an array of sensors to help the visually impaired move around the world without the need of any of their hands to be compromised. The Batpack will come in two separate modules - a haptic feedback backpack with a belt worn around the midriff, and a similar laser/optical feedback system built into a pair of goggles.

The haptic feedback belt will be outfitted with both ultrasonic sensors and vibrating units. These two components, when paired together, will be used to indicate the direction and distance of any surrounding objects. This should provide the user with a lightweight method of naturally feeling their surroundings.

The goggles module, using a Lidar laser and a vibrating unit, will provide a method of gauging the distance of objects that are further away with higher resolution. Now instead of pointing a stick to feel objects within close range, users should be able to simply point their heads in a direction and know with higher resolution how far away an object is.

As an experimental feature, a small camera will be placed on the strap to aid the detection of street corners and sidewalks. With rapid increases in the field of computer vision, we hope the camera will both offer an immediate use as a navigational aide, and help future iterations of the Batpack take full advantage of new Computer Vision Algorithms as they come out.

II. ULTRASONIC SENSORS

For the ultrasonic distance sensor, the sensor sends out an ultrasonic burst, known as a chirp. This transmission will hit an object and reflect the sound back to the receiver, called an echo. The pulse width corresponds with the distance from the object. The speed of sound, c, is roughly 341 m/s. Using the equation below, we can calculate the distance.

$$Distance = \frac{T * c}{2}$$

Where the time, T, is measured by how long it takes between the chirp and echo. The beams are sent with a certain beam pattern, which the echo is looking for. The beam angle will change over distance until the full range is met. Using the beam pattern, we can calculate the beam angle for an object at a certain length. Smaller objects will be detected over a narrower beam angle and shorter distance. Larger objects are detected over a wider beam angle and longer range. [1]

One known issue when using multiple ultrasonic sensors, is there can be crosstalk (interference) from the other sensors when using multiple sensors in a design. We handle this by storing all the sensors into an array and begin looping. The first sensor will process, then trigger the next sensor after it must had received the signal and so on until the end of the array. The process automatically restarts after reaching the end of the array thanks to the looping function in Arduino studio.

A few things were considered when choosing the ultra-sonic sensors for this design. Including: Power, Current Consumption, Current Consumption, Refresh rate, Pulse width, Environment, Operating Frequency.

The two ultrasonic sensors used in this design will be the MaxSonar-EZ2 and the HC-SR04.

A. Ultrasonic Range Finder - LV - MaxSonar - EZ2:

This sensor provides very low power, which is useful as we will need multiple sensors. One issue with this sensor is it requires a minimum of 6 inches before it begins recording. When powering on this sensor, it's good practice to have nothing within seven inches of it, as the first cycle is calibrating the device. Ignoring this known issue could cause some objects to be ignored at that range. The following timing diagram shows RX and pulse width (pin 2) set to high. At this time the burst is sent. When the object is detected, the pulse width pin is set to low. The serial data is sent in RS232.

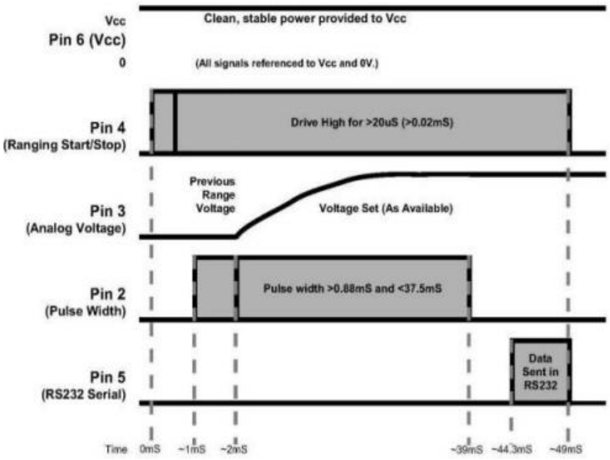

Figure 1. Max Sonar timing diagram

B. Ultrasonic Sensor - HC-SR04:

This sensor will send out a burst of eight cycle ultrasound and raise its echo to high after receiving in input and output a time in microseconds the sound wave traveled. The following timing diagram from the datasheet represents this in a more visual way.

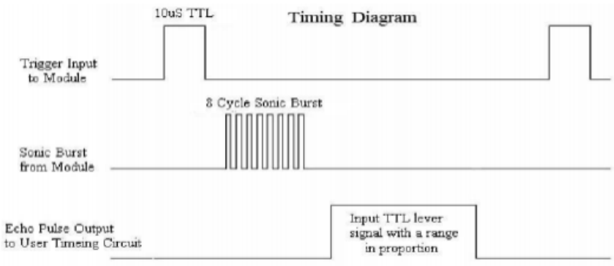

Figure 2. HC-SR04 timing diagram

A quick comparison between the two ultrasonic sensors is listed in table 1.

TABLE 1
Ultrasonic Comparison

Specifications	**MaxSonar-EZ2**	**HC-SR04**
Power	2.5V – 5.5V	5V
Current Consumption	2mA	15mA
Refresh Rate	20 Hz	n/a
Measuring Angle	10 us/Inch	15 degrees
Environment	Indoor	Indoor
Operation Frequency	42kHz	40kHz
Distance	6.45m	4m

In our implementation, we will be using both sensors. The operating frequencies are different allowing us to design the system in a way we minimize the possibility of cross-talk. The SR04 will need to be started manually each time a recording wants to be read, so a loop will be needed. As the data can be noisy during readings, the average of every fifteen recordings will be taken to filter the results, granting a more accurate result. One notable thing between the two devices is the current consumption. The SR04 uses seven times more consumption, increases the overall power used by the system.

Pulse-width modulation is considered for this design as it gives us a control of how much power is being supplied to the device, making it easier to handle any power consumptions we need to fix. This is done by switching on the voltage supply on and off a swift rate. The longer the duration between states, will lead to higher power supplied. Additionally, this will allow us to control how much vibration is going to the user, so when objects appear closer, a stronger vibration will be needed.

III. OPTICAL SENSORS

To compensate for the short range and inability to find small objects from the ultrasonic sensors, a laser distance sensor was added to the Batpack. The LIDAR-Lite V3 is the top of line choice on the market for distance and precision. The component calculates the distance by sending out pulses of near-infrared laser signal. Once the laser beam hits the object, the beam will reflect back to the sensor. The time of flight will give us an idea on how to measure the distance. The distance between two points can be calculated by the following:

$$Distance = \frac{T * c}{2}$$

Where T, is Time and c, is the speed of light. The signal is sent with a coded signature, which it looks for on the reflection. More specifically the device begins by performing a receiver bias correction routine, correcting for changing ambient light levels and allowing maximum sensitivity [2]. Next the device begins the measurement process by sending multiple acquisitions from the transmitter to receive signals back and learn the distance of an object. If there is a match from the signal, the result is stored in memory. The following acquisition is summed with the previous. When multiple signals reflect back to the device after hitting an object, the repeated acquisitions force a peak to show outside the noise. After the peak reaches the maximum value, this value is recorded. This gives the distance of the object. Now that we have the peak value, we take the magnitude of it and will store it. There is another factor the still gets calculated in order to determine the signal strength to be a valid reading. This is known as the valid signal threshold, which is calculated from the noise floor. As long as the peak value is larger than the threshold, the signal is recorded, otherwise returns 1 cm (an invalid reading). It is important to note that after a distance of 1m, accuracy of the reading changes to +/- 2.5cm. This is a relatively small value, but still important to keep in mind as measuring for accuracy is quite important for the visually impaired. As we know a negative side to laser range signals is it has troubles with objects that are too small and not very reflective. The returned signal strength could be improved by attaching infrared reflectors to the target to increase performance. This would be useful to put on useful objects, such as a fire alarm, fire extinguishers or other important objects that could help the visually impaired.

TABLE 2
LIDAR Lite V3 Specs

Specifications	**Lidar Lite v3**
Size	0.8 x 1.9 x 1.6 in
Max Range	40m
Update Rate	270 Hz
Power	5V
Current consumption	105mA – 135mA
Wavelength	905nm
Price	$150

Another issue that arises is the reflectively of an object. There are three different types of reflectively the Batpack worries about: Diffuse reflective: Rough surfaces have a diffuse surface that causes reflected energy to disperse uniformly [2]. This is good as it makes the likelihood of the energy to find its way back to the device increase. These are found in everyday objects due to small imperfections in the surfaces of materials. A good example of this is paper. Even though it may look like a smooth surface, paper actually is rough due to the tiny fibers at the microscopic level.

Specular: Smooth surfaces have specular surfaces which makes it almost impossible for the laser sensor to pick up a reflection that can be used to calculate the distance. This is due to the reflecting have little dispersion, making the beam reflect small. The only way to fix this issue is by taking a measurement with the object directly in front of the sensor. An easy example to picture for these types of objects is a mirror.

Retro-reflective: Retro surfaces will reflect light with minimal scattering, making these items extremely useful for this product. The reflected energy will run along a vector parallel to the transmitted signal. An easy example for this could be a reflector on the back of a bicycle.

After testing on multiple different types of objects, the LIDAR-Lite V3 still proved to be accurate and was able to pick up a reading 90% of the time, proving its worth at $150.

IV. COMPUTER VISION

The visually impaired depend on their walking stick to tell where they are on the sidewalk by the texture of both the grass and the sidewalk. To accommodate this interaction the Batpack use computer vision to give the user awareness of where they are on a sidewalk. The sections below cover how the camera's frames are processed one by one through a pipeline.

A. *Color Thresholding*

Before the frame of the camera is converted into a grayscale image, and the color data is lost, the sidewalk vision algorithm uses color thresholding on certain channels to distinguish from the grass from the walk-way.

The RGB colors and HLS colors were extracted separately to threshold them distinctly, the two channels were merged together after this. In the HLS color space Hue is the position of the pure color it resembles, lightness, is how dark a pixel is and saturation is how white a color is [3].

Looking at each RGB channel separately the R channel separated the sidewalk better than the other two. The image will just take this color channel. Next the HLS channels allow us to take away the color and look at both the lightness and the saturation. Again, looking at Figure 3 below, the S channel defines the sidewalk than the other two channels. The algorithm will then combine the two channels in order to get the best distinction between the sidewalk and the grass.

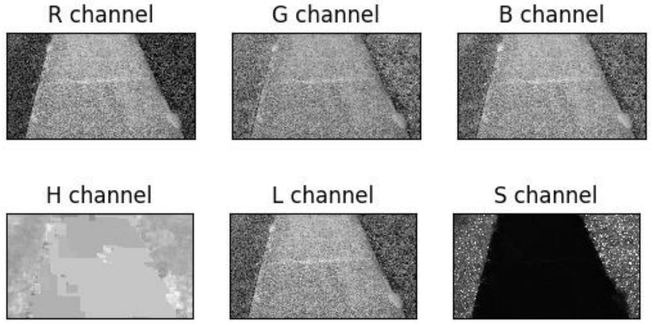

Figure 3. RGB and HLS color channels

The two channels are then extracted separately in order threshold each one individually. The thresholding technique may need to be adjusted in the future to account for different weather conditions. Once thresholded and binarized the two channels are merged together to give us the combined image seen in Figure 4.

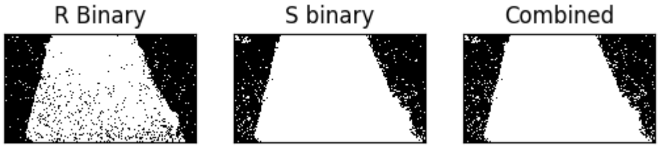

Figure 4. Thresholded channels and combination

This gives a good representation of where the sidewalk is and will be used throughout the algorithm.

B. Contours and Sliding windows

To get the side lines of the sidewalk the boundary detection of white pixels is utilized. This gives us two distinct lines to separate from one another. See Figure 5 below.

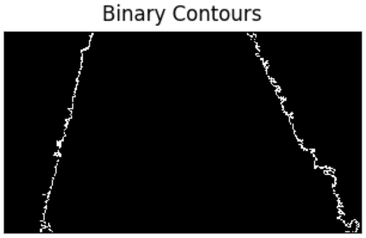

Figure 5. Binary Contours of Sidewalk

This image will then be put into a perspective view to get a perpendicular angle to where the sidewalk is. This can be seen below in Figure 6.

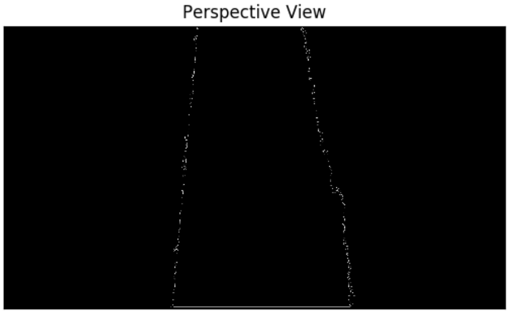

Figure 6. Top Down View

These two lines are put into a histogram by looking at the bottom edge of the image. This tells the sliding windows where to start. After each iteration the slider windows will be moved up until the top of the image is reached, see Figure 7. After each window is placed they will be treated as a separate image but maintaining the same points in the original image.

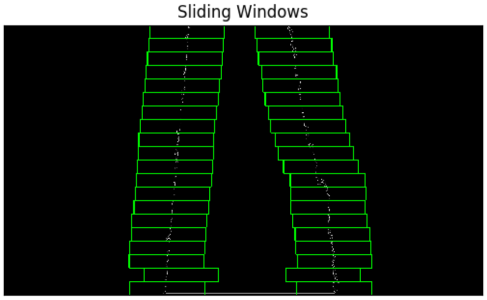

Figure 7. Sliding Windows

C. Transformed Results

Once these two lines are fully defined the image is brought back into normal perspective. Based off the assumption of the average human height being 5.7 the algorithm can tell approximately how close the user is to the right or left edge of the sidewalk. The finished path will continue to update frame by frame to give live feedback. The final result can be seen in the Figure 8 below.

Figure 8. Final Result of Sidewalk Detection

V. PCB DESIGN

The printed circuit board for this project mainly focuses on the laser rangefinder and ultrasonic sensor portions of the device, as the computer vision aspect is primarily handled off-board with the Raspberry Pi unit. At the heart of the board is an ATmega328 microcontroller. This component was selected as it is familiar with most of the group due to being used on the Arduino Uno R3 that we have used previously and are therefore familiar with. This makes coding easy, and we also knew it’d have enough pins for what was needed without going overboard. Smaller parts of the board include a crystal oscillator circuit, to provide the ATmega328 with a more accurate, higher rate clock, and a USB port to connect to the battery with.

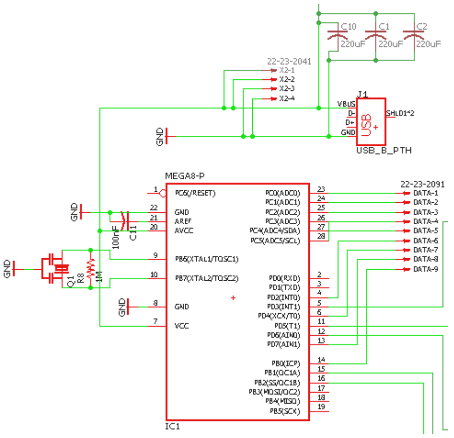

Figure 9. ATmega 328 connections

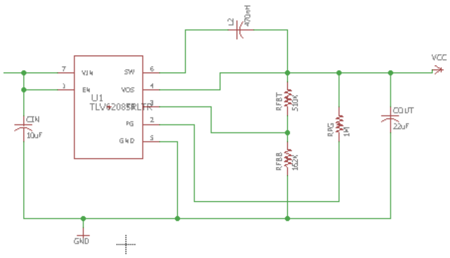

Figure 10. Voltage Regulator

Also housed on the PCB is a voltage regulator circuit. An integrated chip was used to simplify design and reduce overall footprint usage. It produces 3 Volts and can handle a good deal of current, as it is being used to deliver power to the vibrators. This is the only voltage regulation necessary as the sensors and battery were selected to have matching voltage outputs/requirements.

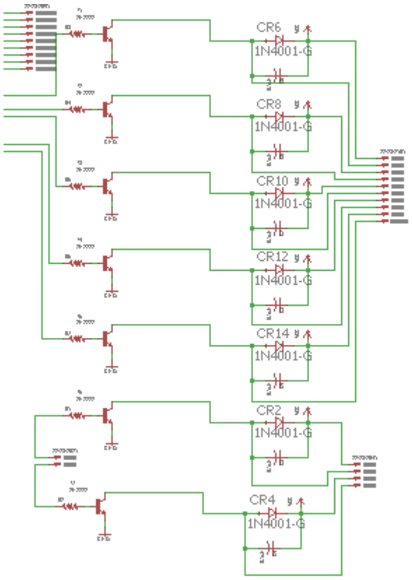

Figure 10. Vibrator Drivers

A large part of the board are the seven vibrator driver circuits, necessary to individually control vibration units. This was necessary as the pulse-width modulations for each unit had to be directly influenced by a single sensor. It was designed to be a simple analog gate that was only opened depending on the modulation signal received by the 2N2222 from the ATmega 328 Microcontroller. When a signal is sent, the gate opens to ground allowing current to flow.

A. Board Layout

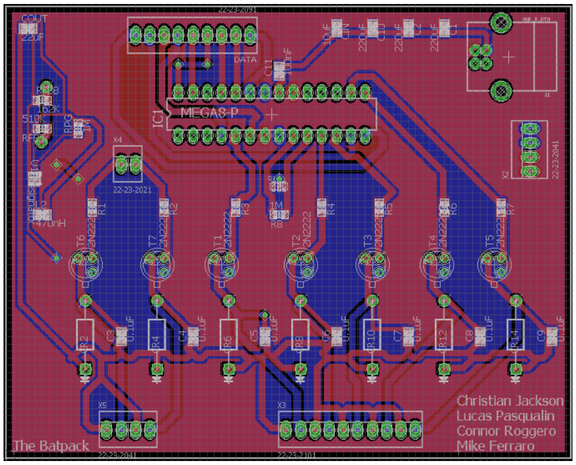

Figure 11. Board Layout

Molex Connector pins are being utilized on the PCB where we needed to connect sensors/vibrators to the board directly. This was done because a direct drill hole would put more physical stress on the solder connection between the PCB and the wire connecting each component. As solder is not a very strong physical bonding, this would make the connection tear very easily. These connectors add a degree of ruggedness to a device that would be vital to the daily function of a user.

VI. BATTERY

To power the entire device, a battery meant for charging phones is being used. It has two ports, a 1 Amp and a 2 Amp port. The 2 Amp port will go to the PCB, and the 1 Amp will go to the Raspberry Pi. The battery itself has a 5 Volt output, which is perfect for the Raspberry Pi, all sensors, and the ATmega 328 microcontroller.

The storage capacity falls at 22,400 mAh, which gives the device a battery life of over a day even when running at maximum power consumption, which it will almost never do. The battery also has a solar recharging feature that will extend its already long battery life.

VII. COMMUNICATIONS

A. Pulse-Width Modulation

Different vibrational intensities and patterns will be used to communicate different messages to the user. Because the vibrational force is a function of the voltage (and power) supplied, it is necessary that we can modulate the voltage to produce different vibrational strengths. Using a microprocessor such as the one on the Arduino, the industry standard way of accomplishing this is with Pulse-Width Modulation.

Pulse Width Modulation, or PWM for short, is a method of modulating a digital signal such that it behaves much more like an analog signal. The goal of this modulation, as previously mentioned, is ultimately to control the power provided to a component. PMW drivers turn a signal on and off very quickly at different frequencies. When the signal is high, that is considered a pulse. The duration for which the signal is high is considered its width. By varying the frequency and duration of the pulse, the PMW driver is effectively modulating its width; hence why it is named Pulse-Width Modulation. Note that when speaking of PWM, duty cycle is typically referred to as the percentage of the time the signal is high for a constant time. In this context, duty cycle is analogous to pulse width.

While duty-cycle and pulse width are very similar measures, it is much easier to talk about the average delivered voltage in terms of duty cycle. In fact, this relation yields a linear relation in the form of:

$$V_{OUT} = DutyCycle \times V_{IN}$$

From the Arduino Software, manipulating the duty-cycle is as simple as writing a percentage to one of our microcontroller's pins. The percentage is represented as a number from 0 - 255, where 255 would represent a 100% duty cycle, and 128 would represent 50%, etc.

B. I2C

Inter-integrated Circuit Protocol is a communication protocol to allow multiple "slave" circuits to communicate to a "master" circuit. The optical range sensor the Batpack will use, LIDAR V3, to detect more precise object detection uses the I2C communication protocol.

I2C only requires two wires to the master to communicate up to 1008 slave devices. It also allows for multiple master chips to be involved on the same bus, see Figure above. During the communication 8 bits of data are sent between 100kHz or 400kHz, along with an extra bit for the ACKs and NACKs. At the hardware level I2C bus contains only two signals, a data signal, SDA, and a clock signal, SCL. The SCL bus is only controlled by the master, making it easy to synchronize multiple devices.

Messages passed between the I2C buses have two types of frames. One is the address frame; this is the frame which tell the slave if it is their message the master is transmitting. The second type of frame is the data frame; more than one can be transmitted but it always is preceded by an address frame. These are transmitted over the SDA bus when SCL is pulled low. Once the message has stopped the master will send a stop condition on the SDA bus and the SCL bus will remain high. Arduino also provides Software for interfacing with the Lidar using I2C.

VIII. SOFTWARE FLOW

The software for the Batpack can be divided into two main components – the Computer Vision modules and the Embedded Modules. As the Computer Vision module is covered elsewhere in this paper, the following will be a breakdown of the Embedded Modules. This includes the Software Drivers for communications with all sensors, and the ensuing modulation of the vibrators.

A. Microcontroller

The main PCB houses the ATmega microcontroller that is in charge of receiving input from the ultrasonic sensors and the optical sensor.

Since the ultrasonic sensors can have crosstalk between them a technique has to be in place to eliminate this interference. Our team has decided that without any filtering of the received waves a timer can be set up to stop the interference. In Figure 12 below is how the ATmega will handle the ultrasonic sensors inputs and transfer them into a PWM to the vibration units for the user.

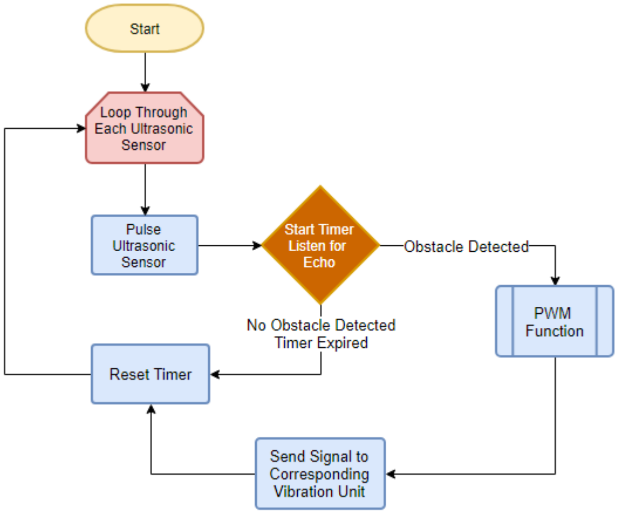

Figure 12. Ultrasonic Sensor Flow chart.

The optical sensor, LIDAR Lite V3, will be controlled by the ATmega as well. This device as mentioned above uses the I2C protocol to communicate its readings. Once the user flips the switch to be on the laser will start reading the objects in front of it. Much like the ultrasonic sensors above this reading will be sent into a PWM function to send the haptic feedback to the corresponding vibrations unit. Below is a flow chart of how this process is implemented.

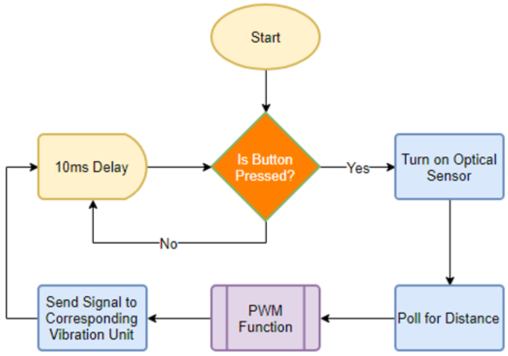

Figure 13. Optical Sensor Flowchart

B. Single Board Computer

The Raspberry Pi v3 will be used in conjunction with the Raspberry Pi Camera v2 to run the machine vision algorithm discussed above. Once the lines are detected and the distances are found the Raspberry Pi's GPIO pins will send signals to the main Batpack PCB to trigger the corresponding vibration motors. All the computing power will be designated to the machine vision algorithms.

IX. USER INTERACTIONS

Every input must be associated with either touch or speech, and cannot have any visual components in order for the user to operate it. The user input will have an on/off switch for all of its components. This includes an on/off switch for the following: the ultrasonic sensors and midriff vibration motor combo, the optical sensor and chest vibration motor combo, and the computer vision.

Our primary method of alerting the user will be done with pulse width modulation (PWM). Using vibration motors with PWM, the user of the Batpack will be alerted of objects within proximity to them. These motors will vibrate at various strengths around the user's midriff, the weaker the vibration the farther away the obstacle is and the stronger the vibration the closer the object is. The vibration units will be placed underneath each ultrasonic sensor, on the side of the person's body, so the user will know which direction the object is.

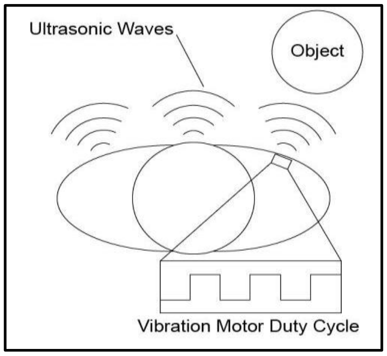

Figure 14. Object Detection via Vibration Motors

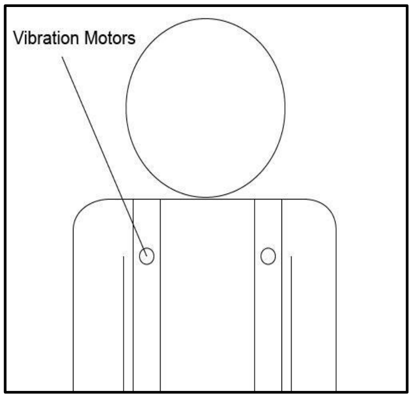

Figure 15. Vibration Straps for Optical Sensor

For the optical sensor, the user will be notified by vibration units located in the straps of the Batpack. Our team decided that it would not be a good user experience if the vibration units were located by the optical sensor, which is located on the head. As this would interfere with everyday life and leave the customer unsatisfied with the product. Our approach to this solution is to move the vibration motors to the straps of the Batpack to ensure the user still gets the haptic feedback needed to feel the world around them. (See Figure 15)

Another sensor that must be taken into consideration is the camera, which provides the computer vision to the system. This subsystem will allow the user to detect the sidewalk they are walking on and provide feedback in order for the user to stay on the path. This subsystem needs to allow a manual switch in which the user will get feedback on if they are looking at the sidewalk or a grass surface. In order to accomplish this task two more vibration motor will be used on the shoulders of the Batpack straps, as seen in the Figure 16.

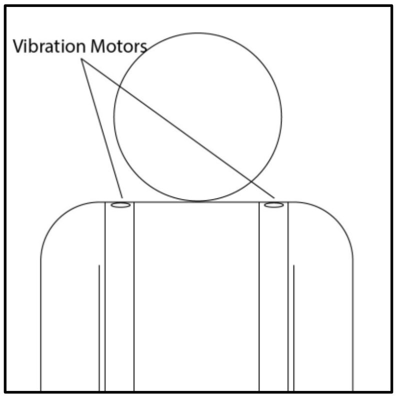

Figure 16. Vibration Straps for Computer Vision

X. CONCLUSIONS

The Batpack created has checked of all our initial specifications listed. However, the full potential of the project has not yet to be meet. Over the research and design of the project we have thought of new ways to enhance the experience the user receives. This includes, but not limited to, facial recognition, cross-walk detection, stair detection, and some audio interactions.

This senior design project has been a great and valuable experience for all the members of our group. Team work, documentation, circuit design, and machine vision are just some of the skills we have enhanced throughout the experience.

XI. REFERENCES

1. Maxbotix. (2017). LV-MaxSonar-EZ Series. [online] Available at: http://www.maxbotix.com/documents/LV-MaxSonar-EZ_Datasheet.pdf [Accessed 6 Jul. 2017].

2. Starlite-intl. (2017). Lidar Lite v3 Operation Manual. [online] Available at: http://static.garmin.com/pumac/LIDAR_Lite_v3_Operation_Manual_and_T echnical_Specifications.pdf [Accessed 6 Jul. 2017].

3. Anon, (2017). Introduction To Color Theory. [online] Available at: http://infohost.nmt.edu/tcc/help/pubs/colortheory/colortheory.pdf [Accessed 1 Aug. 2017].

XII. BIOGRAPHY

Michael Ferraro is a senior at the University of Central Florida and will be graduating with a Bachelor of Science in Computer Engineering in December of 2017. He currently works as an iOS application developer and will begin working at Harris Corporation as a software engineer in the beginning of January 2018.

Christian Jackson is a senior electrical engineering student at the University of Central Florida. He is hoping to pursue a career in the defense industry and to eventually run his own engineering firm. He will also eventually return to school to receive a MBA.

Lucas Pasqualin is a Computer Engineering student at the University of Central Florida. Current interests include Machine Learning, Robotics, and Computer Vision. Lucas is currently employed as a Data Scientist at Abe Ai- an Artificial Intelligence start-up in the Fintech industry.

Connor Roggero is a senior Computer Engineering student at the University of Central Florida. He currently is working with a company to design machine vision algorithms to aid in 3D printing circuits. He hopes to continue his career in creating unique ways to apply machine vision and machine learning.