

HANDS-FREE SUPPORT FOR THE VISUALLY IMPAIRED

ECE 445, Senior Design

By

Arrian Movahedi
Napat Sutthasinwong
Yash Agrawal

December 2018

Abstract

The walking stick has been the primary navigation aid for the visually impaired. Despite its popularity, it has several limitations. It keeps an arm engaged at all times and provides limited feedback. We aim to provide an alternative method of navigation for the visually impaired by employing a series of ultrasonic sensors to detect obstacles and provide feedback to the user by generating a series of directional audio tones to indicate where and how far an obstacle is from the user. All the components are laid out on a vest and can be easily worn atop most clothing.

Contents

1	Introduction	1
1.1	Objective	1
1.2	Background	1
1.3	High-Level Requirements	2
2	Design	3
2.1	Block Diagram	3
2.2	Physical Design	4
2.3	Block Design	5
2.3.1	Sensor Module	5
2.3.2	Control Module	8
2.3.3	Power Module	13
2.3.4	Feedback Module	15
3	Cost and Schedule	18
3.1	Cost Analysis	18
3.2	Schedule	19
4	Ethics and Safety	21
4.1	Ethics	21
4.2	Safety	21
	References	22
	Appendix A	23

1. Introduction

1.1 Objective

We propose to make daily life easier for the visually impaired by designing a device that provides feedback about the location of nearby objects. In order to make the project appropriate for ECE 445, we aimed for a hands-free, 240 degrees obstacle detection, and aimed for making the prototype as non-invasive as possible.

This device will be secured onto a vest, allowing friendly use. The user will receive audio feedback distinguishing the location of the nearest object. This requires using a pair of stereo headphones and circuitry that will be able to create a tone that gives meaningful information to the user regarding how far he or she is from obstacles.

1.2 Background

According to the World Health Organization, there are roughly 253 million people in the world are visually impaired, of which 36 million are blind [1]. The most commonly used aid is the walking cane. The cane keeps one arm engaged and limits the user's range and speed. It is tiresome to use and only signals the user that an object, usually near the ground, is present. It does not provide information about objects above the ground or people behind the user. Previously people have come up with solutions to this problem in this class.

Our proposed solution differs from said projects via the feedback mechanism. We decided to use auditory feedback as opposed to haptic feedback. The device alerts the user of the presence of objects in front and behind the user. Distance information is encoded on the intensity of the signal such that the closer the object is, the louder the tones become. Additionally, we generated directional tones using the 'webAudio' API along with a custom bit encoder.

1.3 High Level Requirements

We consider this project a success if the user can maneuver around objects in any direction without the needing vision. Our high level requirements for this project are as follows:

1. Receive echo signals within 10 cm - 120 cm and 80 - 120 degrees azimuthally on both the front and back sides.
2. Accurately provide at least 24 audio tones based on the position of the nearest object.
3. Overall error in distance measurements should be less than 5 inches (12.7 cm).

2 Design

2.1 Block Diagram

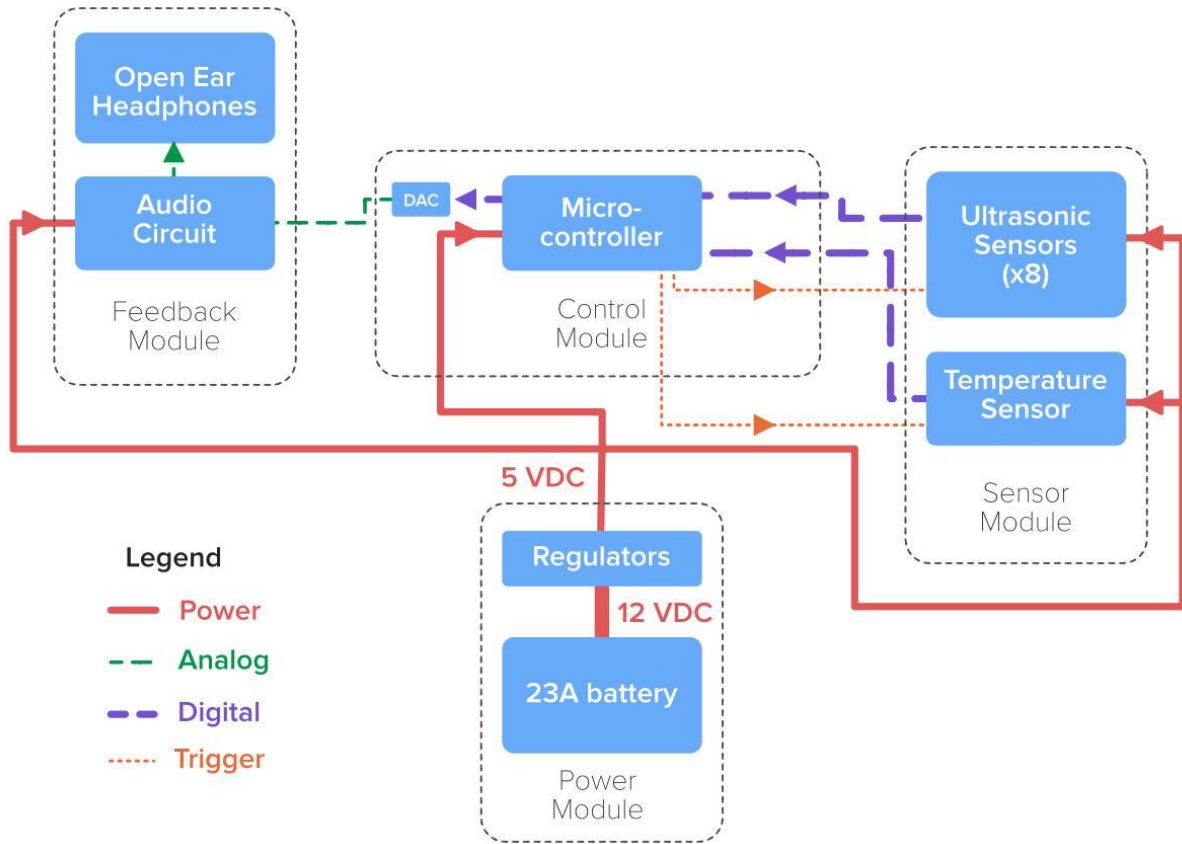


Figure 1. Block Diagram of the Project

To satisfy requirement #3, the sensors must be calibrated upon turning the device on. Using an Atmega328P-PU microcontroller chip, we will program a boot-up sequence where at least a tenth of a second is used for calibration purposes. This step is also essential because our algorithm for generating directional tones for feedback relies on time differences between echo arrivals across the vest.

To satisfy requirement #1, we placed four ultrasonic sensors on the front side of the vest and four of them on the backside. Each sensor should have an azimuthal sensing range of 15-25

degrees. To satisfy requirement #2, we stored directional audio files onto a SD card and interfaced the SD card with the microcontroller.

2.2 Physical Design

We mounted the system on a vest using velcro, for ease of use. We attached the power and audio modules on the back of the vest and the battery pack and switch at the front. Feedback is provided through a pair of earbuds. Using a vest allowed us to optimally position the sensor for the most accurate feedback.

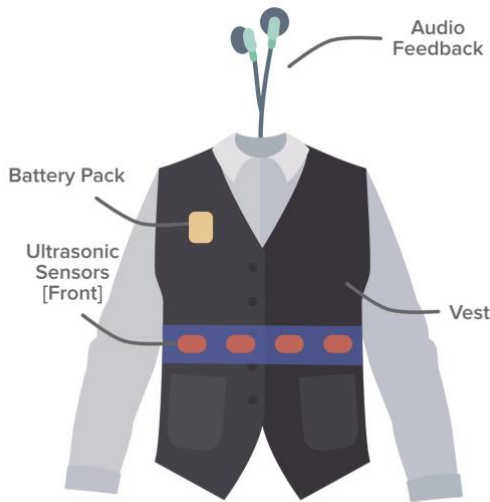


Figure 2. Front view of the vest

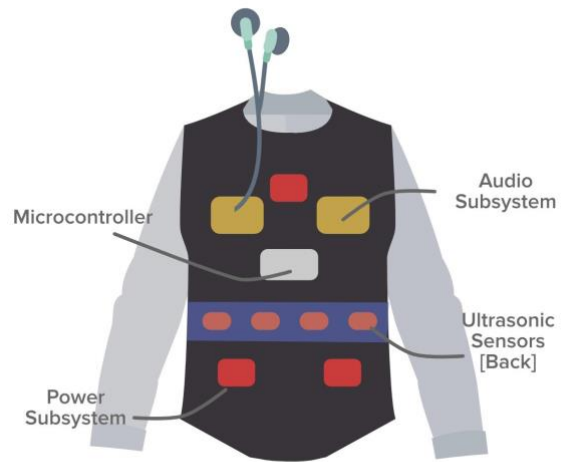


Figure 3. Back view of the vest

As clearly visible in the top view figure shown in figure 4, each sensor mounted on the vest would have an azimuthal sensing angle of 15° - 30° , sweeping a range of 240° overall. The range of object detection would lie within a 80 cm - 120 cm radius of the user. Since we want our device to be non-invasive, we aim to extensively test the system in various distances to strike the balance between providing feedback loud enough for the user to navigate safely and discrete enough to avoid irritating the user with continuous tones.

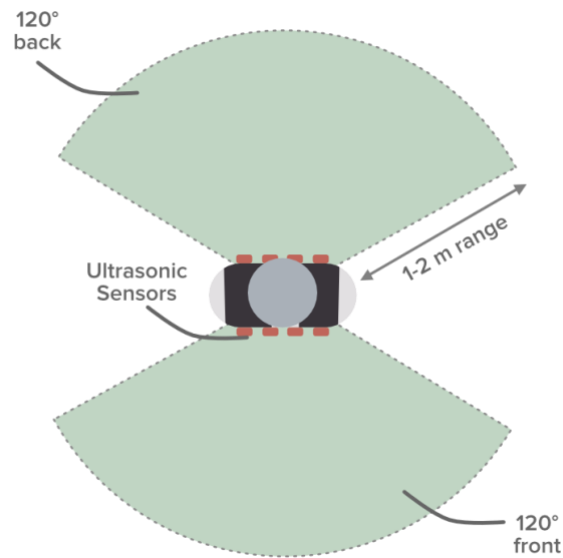


Figure 4. Top view of the device

2.3 Block Design

2.3.1 Sensor Module

Ultrasonic Sensors

We use a total of eight HC-SR04 ultrasonic sensors to achieve our ranging goals. Each sensor can send out eight ultrasonic bursts at 40k Hz. The bursts occur when the microcontroller sends the trigger signal to the sensor's trigger input pin. The echo output pin is used to receive the presence of an echo signal, which is a digital signal produced by the chip onboard each sensor module. The sensors are a necessary component for all of our high-level requirements.

Temperature Sensor

For accuracy in distance data, we needed to account for the fact that the speed of sound depends on the temperature of the medium of propagation, which in our case is air. We used a temperature sensor to correct the speed of sound calculations in the microcontroller. The temperature sensor would be crucial in very hot or cold environments, where the speed of sound changes significantly.

Requirements and Verifications Table

Table 1 RV table for the sensor module

Requirements	Verification
<p>Ultrasonic sensors</p> <ol style="list-style-type: none"> 1. Sensors allow sensing of 80°-120° azimuthally on the front 2. Sensors allow sensing of 80°-120° azimuthally on the back 3. Sensors allow sensing of 15°-30° “cone of confusion” in the zenith angle, on the front and back. $\theta = S/r = 0.5/2 * (180/\pi) = 14^\circ$ where S is roughly the object’s vertical dimension and r is the distance from the user and the object. This requirement is therefore for an object with a vertical dimension of 0.5 meter. 4. Must be able to detect objects in the 10 cm - 120 cm range 	<ol style="list-style-type: none"> 1. Azimuthal sensing angle <ol style="list-style-type: none"> a. Power up each ultrasonic sensor with 5V b. Use the Arduino Uno board to pass an ON signal to the “trigger” pin of the sensor c. Read the “echo” pin of each sensor serially d. Use the serial comm display to view the measured distances in real time e. Move an object azimuthally across a sensor f. Mount all of the sensors onto the vest g. Ensure that objects within a 80°-120° from the front of the vest can be detected 2. Same procedure as in step 1 3. Apply the same procedure for determining the azimuth angle, except rotate the object radially from the sensor axis in the zenith direction instead of azimuth 4. Range <ol style="list-style-type: none"> a. Hold a large flat object in front of the sensors b. Trigger the sensor and process its echo signals (as outlined in step 1) to determine the distance to the object c. Move the object away from the sensors while continuously measuring the distance d. Observe when the serial data on the computer screen becomes invalid. This is the maximum range.

Requirements and Verifications Table (Cont.)

<ol style="list-style-type: none"> 5. Must have an accuracy of $\pm 10\text{cm}$ 6. The minimum spacing between each sequence of pulses should be at least 11.7 ms 7. Temperature sensor can interface with the microcontroller and transmit temperature data 	<ol style="list-style-type: none"> 5. Accuracy <ol style="list-style-type: none"> a. Use a ruler to compare measurements made by the sensor at 10 cm, 20 cm, 30 cm up to 120 cm and plot error vs. distance 6. Transmit-time spacing <ol style="list-style-type: none"> a. Connect the sensors to the controller module, consisting of the MCU and multiplexers b. Power the sensors to $V_{cc} = 5\text{V}$ a. Connect the echo pins to the appropriate microcontroller pins as set up earlier in step 1 b. Measure the time difference between each consecutive echo signal being read by the microcontroller 7. Temperature sensor-MCU interfacing <ol style="list-style-type: none"> a. Connect temperature sensor V_s, V_{out}, and GND pins correctly to power source, an analog voltage pin of the MCU, and ground, respectively c. Use the serial monitor of the Arduino to continuously measure the voltage from the temperature sensor output d. Convert the voltage value to degrees Celsius by multiplying the values by $(5*100)/1024$
---	---

Supporting Material

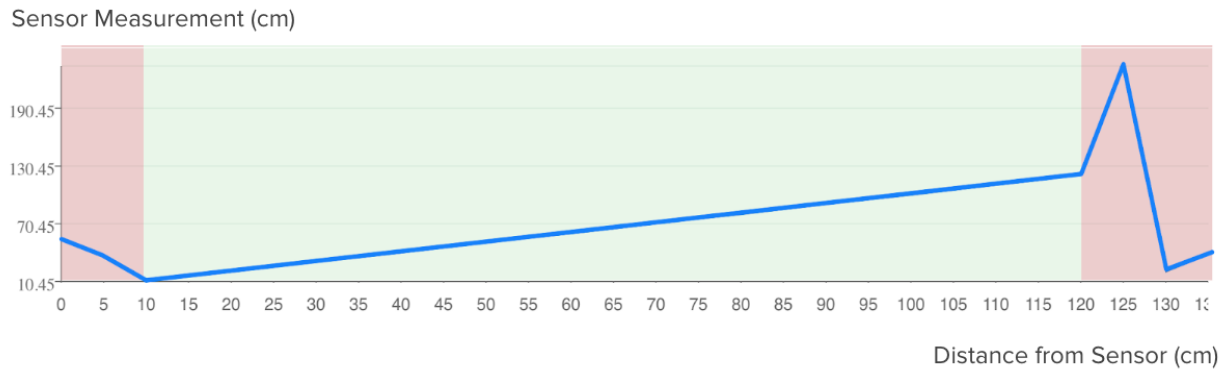


Figure 5. Ultrasonic sensor verification

The ultrasonic sensors are accurate within 5 mm. Figure 5 displays three different regions, marked by green and red color. The green region is the distance range values that were accurate within 5 mm. The red regions represent where the microcontroller (MCU) receives unreliable data and unusual spikes. We used this information to account for false-positive and false-negative readings by disregarding data in the red regions. The slope of the line for the unreliable data ranges is a crude average of data values recorded.

2.3.2 Control Module Functional Overview

Microcontroller (MCU) Unit

The MCU handles the data input from the sensors, data processing, and audio feedback tone playback. The system control logic is autonomous upon bootup. The chip runs using an external crystal oscillator and interfaces with the ultrasonic and temperature sensors. The MCU also interfaces with an external memory, micro SD card, where the audio tones will be stored. The MCU operates in 4.5V - 5.5V range at 20 MHz.

The MCU selects the ultrasonic sensors to trigger in sequence. This sequence simply starts from one sensor, ends at the very last sensor, and repeats continuously. To minimize latency between object detection and audio feedback we used the same select lines for the front and back set of sensors. The MCU receives input data from both the front and back sensors simultaneously. With this setup, we can track the nearest object faster and use less memory to

handle data value comparisons. Using one-hot encoding, we decide on which detected object is closer and which audio file should be played.

External Crystal Oscillator

To provide a faster clock signal source than the internal RC oscillator that runs at 8 MHz, we used a 16 MHz crystal oscillator as an external source. Using an external oscillator increases the calibration accuracy from ten percent to one percent.

External Memory

The audio tones are stored in a formatted file within the micro SD card. The MCU interfaces with the SD card through a SPI interface. We used the SimpleSDAudio library to interface with the SD card. The audio files are 16 bit, full-rate stereo, and at a sampling rate of 44.1 kHz.

Digital to Analog Converter

To convert the fetched audio files from the SD card to analog waveforms, we used a DAC0830LCN DAC. This chip requires the connection of a 0.1 pF capacitor in series with a 10 k Ω resistor for both the left and right audio output channels.

1 to 4 Multiplexer (MUX)

The MUX is used by the MCU to select the appropriate ultrasonic sensor for triggering. The power supply must be at 5 V \pm 0.5 V and the switching speed should be below 10 picoseconds [2]. The analog outputs of the MUX are connected to the trigger line of the sensor appropriately.

4 to 1 Demultiplexer

This dual DEMUX [4] is used by the MCU to select the appropriate ultrasonic sensor for reception an echo signal. The power supply must be at 5 V \pm 0.5 V and the switching speed should be 10 picoseconds. The analog input to the DEMUX will be from the output of each ultrasonic sensor. The analog output of the DEMUX will be translated by the internal analog to digital converter as an input before reaching the microcontroller unit.

Requirements and Verifications Table

Table 2 RV table for the control module

Requirements	Verification
<p>DAC</p> <ol style="list-style-type: none"> 1. Successfully convert the digital signal back to analog at a minimum frequency of 44.1 KHz 	<ol style="list-style-type: none"> 1. Check the conversion rate <ol style="list-style-type: none"> a. Use the oscilloscope to see the digital data signal frequency at 44.1 kHz
<p>Microcontroller</p> <ol style="list-style-type: none"> 1. Must be able to trigger and sequence the sensors in the order from first to the fourth 2. Must be able to repeat the sensor sequencing in an autonomous loop 3. Must be able to capture the echo data from each individual sensor 	<ol style="list-style-type: none"> 1. Sequence of sensor triggering <ol style="list-style-type: none"> a. Power up the microcontroller and sensors to 5V b. Trigger one set of four sensors 2. Repeatability <ol style="list-style-type: none"> a. Follow the same procedures as step 1 b. Ensure that the software program is autonomous such that the sensors are always in the process of transmitting and receiving in the correct sequence c. Check that the sensors are being triggered continuously and the controller is receiving data. 3. Capturing echo data <ol style="list-style-type: none"> a. Follow the same procedures as step 1 b. Place a flat object in front of the sensors c. Send a trigger signal to one or more of the sensors with a duration of at least $10\ \mu s$ d. Read the echo data and ensure validity e. Ensure correct reading order of the sensors (1st to 4th)

Requirements and Verifications Table (Cont.)

<ol style="list-style-type: none"> 4. Must be able to convert the echo data to distance values and store them 5. Must be able to compare the calculated distances and determine which sensor detects the nearest object. If there is no nearby objects, no audio output is played. 6. Must be able to interface with the external SD card and read audio files from memory 7. Must be able to output the audio data to the audio controller in less than 170 ms to account for human reaction time 	<ol style="list-style-type: none"> 4. Data processing <ol style="list-style-type: none"> a. Follow step 3 for echo data b. Output the calculated values onto the serial display c. Compare the distance to the actual distance using a measuring tape/stick. Audio feedback 5. Audio output <ol style="list-style-type: none"> a. Power up the system. b. Place a flat object in front of one of the sensors. c. Following step 4, output the sensor that detects the nearest object. d. Compare which sensor is determined to the actual sensor that senses the nearest object. 6. Interfacing with memory <ol style="list-style-type: none"> a. Connect the microcontroller to the external memory IC. b. After the sensor with the nearest object is detected, fetch the corresponding audio file from the memory system using SPI protocol. 7. Audio feedback output <ol style="list-style-type: none"> a. Connect the audio controller to the microcontroller b. Implement a timer to restrict the frequency of audio feedback from the microcontroller. c. Use timer to send the output signal to the audio controller
--	--

Supporting Material

Microcontroller

The distance value is calculated by using the speed of sound. For testing purposes the speed of sound is defaulted to 331.3 m/s based on room temperature. The speed of sound is corrected based on the temperature sensor's output value. The MCU records the time it takes for an echo signal to arrive after transmission. The signal travel time is divided by two to account for the two way travel time of the ultrasonic signals.

$$Distance = \frac{Signal\ Travel\ Time}{2} \times 331.3 \times \sqrt{1 + \frac{Temperature\ Sensor\ Reading}{273.15}} \quad (1)$$

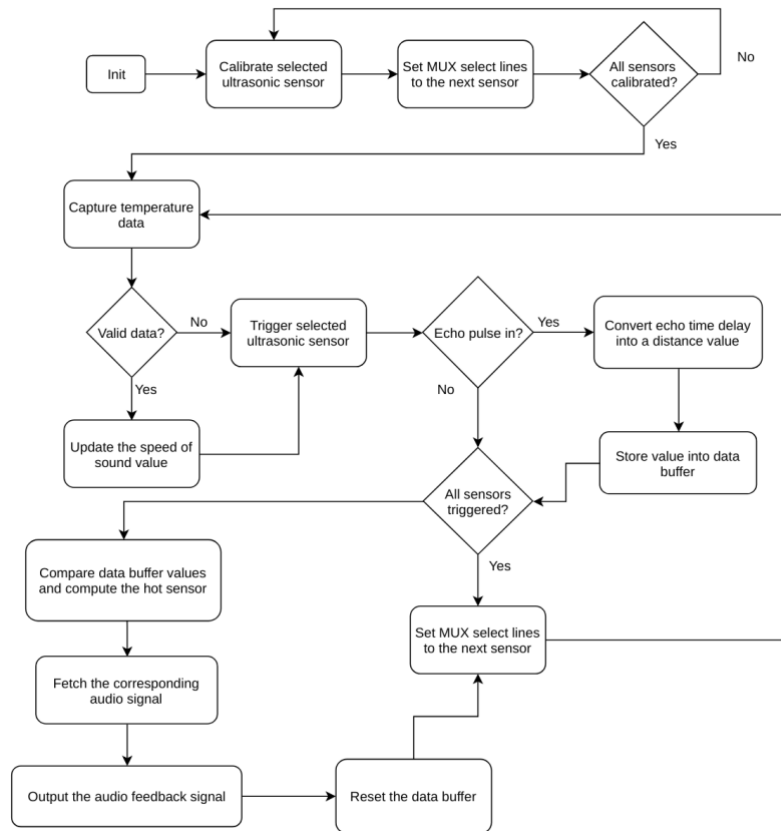


Figure 6. System logic flow chart

This chart represents the high-level logic flow of our system. The calibration sequence will run on bootup. The temperature sensor will continuously send analog data to the MCU, updating the speed of sound value. The sensor triggering and data processing are all also a part of the control logic loop.

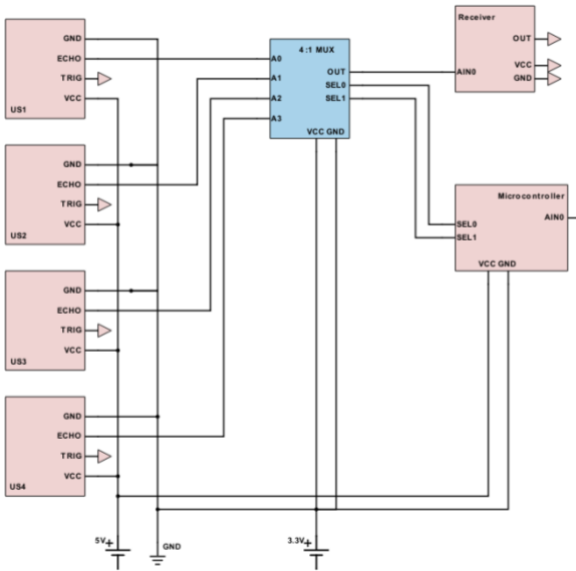


Figure 7. 4:1 Multiplexer schematic (front)

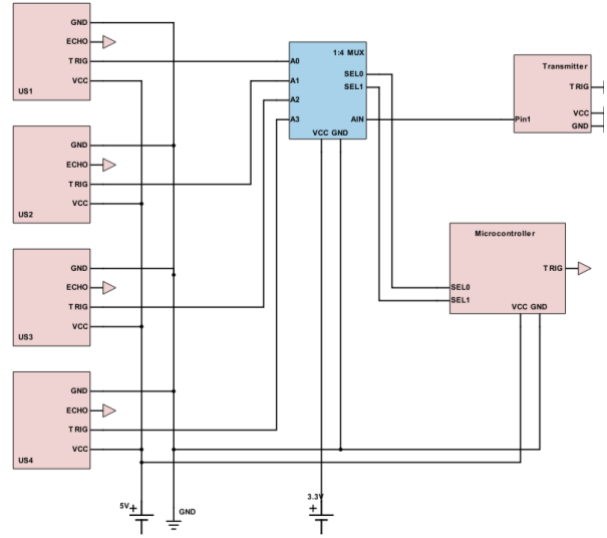


Figure 8. 1:4 Multiplexer schematic (front)

2.3.3 Power Module

Functional Overview

Since our application uses batteries, it would be ideal to minimize unnecessary power loss. We think that using DC-DC [5] switching type regulators would be best for our application. Although they would introduce a lot more noise than linear regulators, they would be more efficient in terms of power conversion. With that being said, using DC-DC switching regulators would require the use of a control signal, itself triggered by a voltage source or a counter driven by a clock. For this reason, we will use standard LDOs for our project.

12V Lithium-Ion Battery

In order to energize all of the modules in this system, we used two 12 V Lithium-ion batteries. We soldered a battery holder onto a separate through hole board for stability. This module is absolutely essential in satisfying all three of our high-level requirements.

Linear voltage regulators

To be able to power the sensor, control, and feedback modules, we had to supply 5 V stepped down from our 12V battery source.

Requirements and Verifications

Table 3 RV table for the power module

Requirements	Verifications
<p>Linear voltage regulators</p> <ol style="list-style-type: none">1. Must be able to generate a steady voltage in the 3.3V-5V range2. Must have a drop-out voltage of less than 3V3. Must deliver an output current of 100 mA - 150 mA	<ol style="list-style-type: none">1. Voltage<ol style="list-style-type: none">a. Connect a reference voltage of 12V from the Li-ion battery to the input of the regulatorb. Measure the output voltage of the regulator using an oscilloscope2. Drop-out voltage<ol style="list-style-type: none">a. Connect two cables to the input of the regulator circuitb. Measure the output of the regulatorc. Gradually increase the input voltage from 1V up to 12V in 1V incrementsd. Note the dropout voltage3. Current<ol style="list-style-type: none">a. Connect the output of the power module to the submodulesb. Observe the current draw from the power supply

Supporting Material

Voltage Regulator

We used a LM7805 voltage regulator IC to step down 12V down to 5V. The LM7805 IC steps down any voltage within the range of 7.5V to 37.5V to 5V. We also included capacitors with values of 0.33 μ F and 0.1 μ F in parallel to the voltage input and voltage output of the IC respectively as specified by the datasheet.

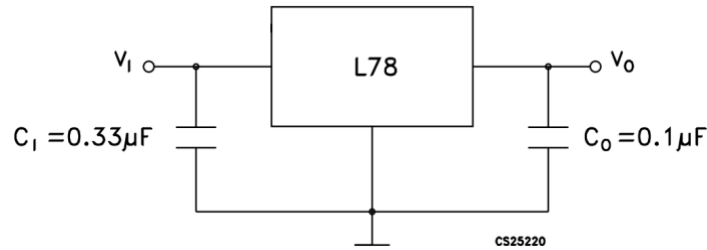


Figure 9. Schematic of the power module using the LM7805 IC

2.3.4 Feedback Module

Functional Overview

Audio Controller

This module is necessary for satisfying an important requirement, which is to detect alert the user about the azimuthal direction of the object through auditory feedback. The microcontroller fetches audio data from a SD card and outputs it to a DAC. The output of the DAC then feeds into the audio controller.

The audio controller consists of two BJT amplifiers in the common emitter configuration, as well as two low pass filters to filter out the output of the DAC. The amplifier for each channel, left and right, is shown in figure 10.

Requirements and Verifications

Table 4 RV table for the audio controller module

Requirements	Verifications
<p>Audio controller</p> <ol style="list-style-type: none">1. The audio controller has a gain of at least 20 dB2. The audio controller has a 3dB bandwidth of at least 3.5 kHz3. Output does not get saturated for inputs in the 50 mVpp - 80 mVpp range4. The lowpass filter has a 3 dB bandwidth at $2.5\text{ kHz} \pm 100\text{ Hz}$	<ol style="list-style-type: none">1. Gain<ol style="list-style-type: none">a. Using a BNC cable, connect the output of a signal generator to the input of the audio controllerb. While the audio controller power supply is turned on, generate a tone at 500 Hzc. Divide the output voltage by the input voltaged. Take the logarithm of said quantity and multiply by 20.2. 3dB Bandwidth<ol style="list-style-type: none">a. Measure the output voltage when the input is fixed at 70 mVb. Sweep the frequency until the output voltage divided by the input voltage becomes 0.707. This will be the 3dB bandwidth3. Output saturation<ol style="list-style-type: none">a. Follow the same procedure as part 1 for connectionsb. Increase the signal generator voltage until the first sign of distortion occurs at the audio controller output (e.g. clipping)4. Low pass filter 3dB bandwidth<ol style="list-style-type: none">a. Follow the same procedure in step 2, except for the filter

Supporting Material

The DAC introduces some attenuation to the audio files, therefore we designed two common emitter amplifiers for our audio controller. One for the left channel and one for the right channel. The setup is shown in figure 10. The verification for the amplifier's 3dB bandwidth is shown in figures 11 and 12, which show that the gain is roughly 20 dB and 30 dB, respectively.

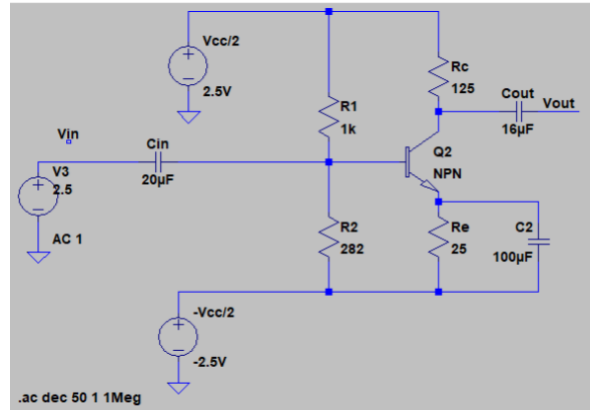


Figure 10. Common emitter amplifier with no DC offset present

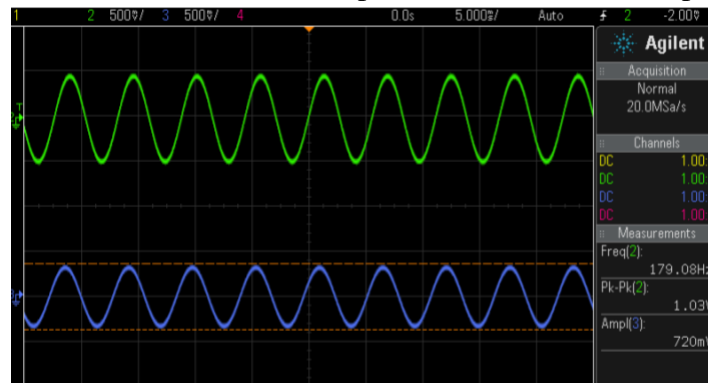


Figure 11. 3 dB bandwidth at 179 Hz

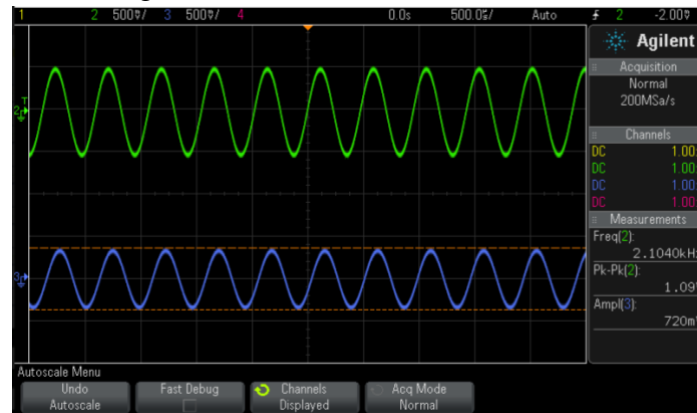


Figure 12. 3 dB bandwidth at 2.1 kHz

3 Cost and Schedule

Table 5 Labor Costs

	Hourly Rate	Hours Invested	Cost
Arrian Movahedi	\$30.00	240	\$7200.00
Napat Sutthasinwong	\$30.00	240	\$7200.00
Yash Agrawal	\$30.00	240	\$7200.00
Unadjusted Total		720	\$21,600.00
Adjusted Total (2.5x)			\$54,000

Table 6 Selected Parts and Costs table

Description	Part Number	Quantity	Cost/Part	Total Cost
Ultrasonic Range Finder	HC-SR04	10	\$3.29	\$32.9
Temperature Sensor	AD22100KTZ-ND	1	\$4.74	\$4.74
Microcontroller	ATmega328P-PU	1	\$2.15	\$2.15
Quadruple Norton Op Amps	LM3900	3	\$0.48	\$1.44
LDO	TLV704	3	\$0.83	\$2.49
DAC	TLC5602CDW	2	\$4.64	\$9.28
4:1 MUX	MAX4734EGC	1	\$1.95	\$1.95
1:4 MUX	CD4555BE	1	\$0.46	\$0.46
			Total:	\$64.45

$$\text{Grand total} = \$54,000 + \$64.45 = \$54,064.45 \quad (2)$$

3.2 Schedule

Table 7 Project Schedule and Work Division

Week	Team	Arrian	Napat	Yash
9/17	Proposals	Finalize Proposal	Mock Design Review Prep	Determine ultrasonic sensors to use
9/24	Mock Design Review Sign-up	Sign up for Mock Design Review Finish Eagle Assignment	Finish Eagle Assignment Find microcontroller	Finish Eagle Assignment
10/01	Mock Design Review	Simulate Tx, Rx and audio circuits in LTspice Research Audio subsystem	Finish Design Document Find ADC, DAC, MUXES	Complete Mock Design Review Finish Design Document
10/08	Design Review	Participate in the Design Review Soldering Assignment	Participate in the Design Review Soldering Assignment	Participate in the Design Review Soldering Assignment
10/15		Determine audio feedback subsystem Verify sensor module	Assemble sensor module Determine control module	Assemble sensor module Determine control module
10/22	Submit first round PCB order	Finish Tx/Rx PCB	Finish PCB layout Assemble control module	Begin PCB layout
10/29		Verify control module Assemble audio feedback module	Complete backlog	Verify control module Complete backlog
11/05		Reorder PCB if	Assemble all	Mount device on

		required	modules	the vest
11/12		Verify all modules Complete backlog	Verify all modules Complete backlog	Verify all modules Complete backlog
11/19	Fall Break			
11/26	Mock Demo	Revise the requirements and verification table if necessary	Work on feedback received	Commence work on the final paper
12/03	Presentation	Prepare for the final presentation Complete Final paper	Prepare for the final presentation Complete Final paper	Prepare for the final presentation Ensure requirements on par for final paper
12/10	Final Paper	Proofread Final Paper Lab checkout	Proofread Final Paper Lab checkout	Proofread Final Paper Lab checkout

4 Ethics and Safety

4.1 Ethics

According to the IEEE Code of Ethics, we need to be honest and realistic in stating claims [6]. Our project aims to provide a means for hands-free navigation to the visually impaired and while it is an ethically sound idea, the safety of the user is paramount. Given the short development period, it is not recommended that our device be used at this stage as a primary means of navigation. Upon further work and extensive testing this device could potentially help navigation without the need of sight.

In addition, there are notable risks involved with wearable electronic devices that need to be acknowledged. In accordance with the IEEE Code of Ethics, we must promptly disclose factors that might endanger the public [7]. Malfunction of the power module would arguably be the biggest risk involved in our project.

4.2 Safety

Given the nature of the project, certain precautionary measures need to be taken in order to be safe. We plan to use a Lithium-Ion battery to power our circuit. In general, any device that stores energy carries the risk of overheating. Lithium ion batteries are smaller and lighter but are substantially more flammable as compared to lead acid batteries [8]. We will undergo the required fire safety training before using this battery and we will make sure to properly test our power system. All the charge/discharge tests with the battery were performed while the battery was inside of a lithium safety bag and the complete procedure addressed on course website was duly followed.

We aim to design our device to be as non-invasive as possible so that a user can go about his or her daily activities without any issues, but since there's an audio module at play, we must make sure that the directional tone does not interfere with the user's daily tasks.

References

- [1] World Health Organisation, “Visual impairment and blindness fact sheet,” Available at: <https://www.who.int/news-room/fact-sheets/detail/blindness-and-visual-impairment>, [Aug. 2014].
- [2] Mouser Electronics, “Low voltage, 4 Channel Analog Multiplexer,” Available at: <https://www.mouser.com/datasheet/2/256/MAX4734-52800.pdf>, [2016].
- [3] Microchip, “Voltage Output Digital to Analog Converter with SPI Interface,” Available at: <http://ww1.microchip.com/downloads/en/DeviceDoc/22248a.pdf>, [Jan 2010].
- [4] Texas Instruments, “Dual Binary 1 to 4 Decoder/Demultiplexer,” Available at: <http://www.ti.com/lit/ds/symlink/cd4555b.pdf>, [Accessed 1 Oct. 2018].
- [5] Texas Instruments, “DC-DC fundamentals - switching regulator overview,” [online] Available at: <https://training.ti.com/dc-dc-fundamentals-switching-regulator-overview> [Accessed 1 Oct. 2018].
- [6] IEEE, “IEEE Code of Ethics,” 2018. [online]. Available at: <https://www.ieee.org/about/corporate/governance/p7-8.html> [Accessed 20 Sept. 2018].
- [7] DSMT, “IP Rating Chart,” 2018. [online]. Available at: <http://www.dsmt.com/resources/ip-rating-chart/> [Accessed 23 Sept. 2018].
- [8] UIUC, “Safe Practice for Lead Acid and Lithium Batteries,” 2018. [online]. Available at: <https://courses.engr.illinois.edu/ece445/documents/GeneralBatterySafety.pdf> [Accessed 15 Oct. 2018].
- [9] Gadgetronicx, “Ultrasonic Transmitter Circuit using IC 555,” 2015. [online]. Available at: <http://www.gadgetronicx.com/ultrasonic-transmitter-circuit-ic555/> [Accessed 25 Sept. 2018].
- [10] Drexel, “Ultrasonic Receiver,” 2018. [online]. Available at: http://www.pages.drexel.edu/~pyc23/ultrasonic_receiver.html [Accessed 27 Sept. 2018].

Appendix I: Design of an Analog Ultrasonic Transceiver

At the initial stages of this project, we decided to design an ultrasonic transceiver for the front end of the system. The goal was to design the sensor module ourselves to have greater control over the transmission and reception process, as well as adding complexity to the project. We have included some test results, as well as an explanation as to why we decided to move forward with off-the-shelf solutions for the front end of the design.

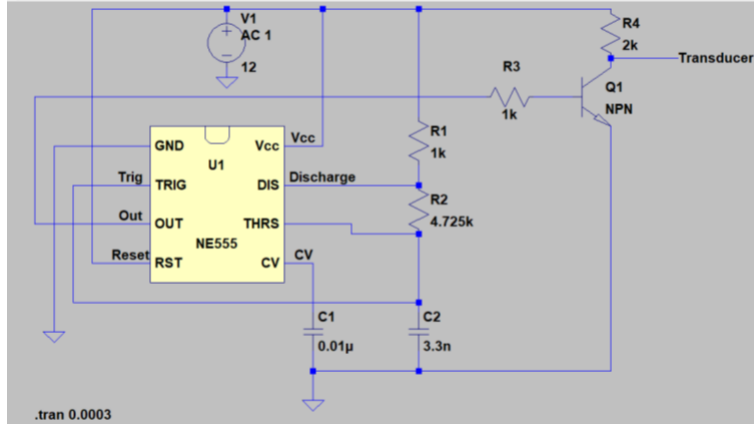


Figure 13: LTspice schematic of the 555 timer used to transmit ultrasonic pulses

Figure 13 shows the transmitter circuit design we initially planned on using for the transmission of ultrasonic pulses [9]. The circuit uses a 555 timer to as an oscillator where the frequency and duty cycle of the output square waves are set by the values of R_1 , R_2 , and C_2 .

$$f = \frac{1.38}{(R_1 + 2R_2)C_2} = \frac{1.38}{(1k\Omega + 9.45k\Omega)(3.3 \text{ nF})} = 40.01 \text{ kHz} \quad (3)$$

$$\text{Duty cycle} = \frac{R_1 + R_2}{R_1 + 2R_2} = \frac{5.725 \text{ k}\Omega}{10.45 \text{ k}\Omega} = 54.78 \% \quad (4)$$

Figures 14 and 15 show the output of the output of the transmitter when no transducer is attached vs. when a transducer is attached, respectively. It can be seen in figure 15 that the output of the transmitter is distorted due to the loading by an ultrasonic transducer. Such distortion before the signal is even transmitted and the echo is received led us to decide on choosing off-the-shelf sensors for the front end. Our receiver circuit [10] schematic is shown in figure 16. The measurement results in figure 17 show that even with a clean square wave generated by a signal generator, the receiver output is distorted.

These results show that designing and integrating such analog modules without purchasing expensive and sensitive components would cause the transmission and reception of ultrasonic signals to fail. For these reasons, we decided to purchase off-the-shelf sensor modules and shift our attention to the audio processing, since that is the one characteristic of our project which makes it unique from past projects done in this course.

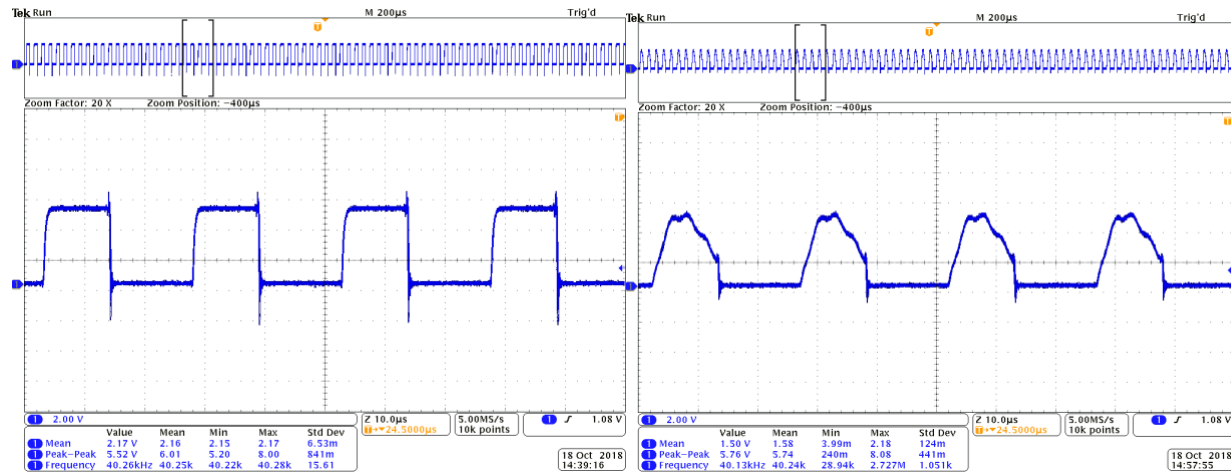


Figure 14. Transducer input (without transducer) Figure 15. Transducer input (with transducer)

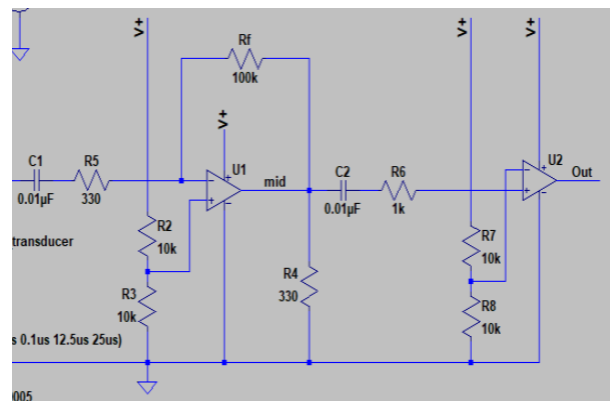


Figure 16. Ultrasonic receiver

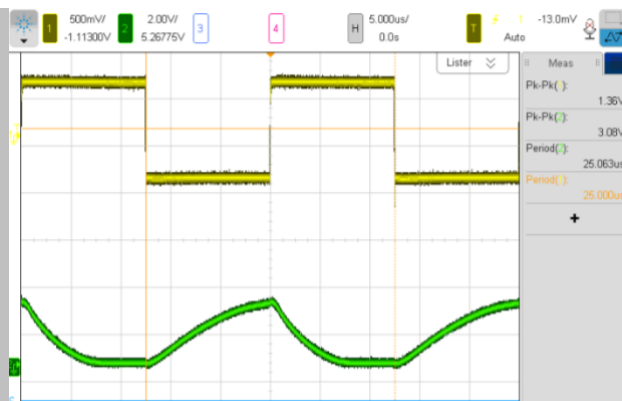


Figure 17. Receiver output (Green)