Ultrasonic Range-Finding Assistive Device for Persons with Low Vision

A Senior Project
presented to
the Faculty of the Computer Engineering Department
California Polytechnic State University, San Luis Obispo

In Partial Fulfillment
of the Requirements for the Degree
Computer Engineering

by

Aaron Morelli, Eric Osgood, Francis San Luis,
Joseph San Diego, Michael Boyd, and Nathan Helenihi

June 10, 2011

Table of Contents

List of Figures

List of Tables

Abstract

The mission of team Engineering 4 Eyes (E4E) is to conceptualize, design, and implement a device which will aid persons with low vision in the detection of obstacles and/or hazards. Persons with low vision are unable to detect objects that are at or above waist level. Team E4E will develop a device based on research conducted with specialists and subject experts. The device will be discrete and will not draw attention to the client. The device will not impair the use of existing equipment used by the client. The device will be encased in an enclosure that would be able to attach to any cane. The device will not threaten the safety of the user or anyone around them.

Chapter 1

Introduction

The Quality of Life Plus (QL+) [1], is a nonprot organization created to foster and generate innovations to aid and improve the quality of life for those injured in the line of duty. By harnessing the exceptional creative and engineering skills of students and faculty at Cal Poly, QL+ succeeds in developing innovative solutions that help our nations heroes to live, to work and to play. At their on-campus laboratory, teams of students and faculty work together to identify solutions that will improve the lives of America's wounded patriots.

The Engineering 4 Eyes team, in collaboration with the QL+ organization, will develop a device that assists people with low vision. The E4E team will conduct research that investigates current devices available to people with lo-vision. Interviews will be conducted with subject experts in order to learn what day to day hindrances current technology is unable to provide assistance with. Based on the aforementioned research, a list of marketing and engineering requirements will be made. Next, E4E will generate concepts and brainstorm ideas for a device to meet all identified marketing and engineering requirements. After weighing benefits and drawbacks to all generated concepts, E4E will choose a concept to design and implement. The completed prototype will be tested by first E4E team members, next by subject experts. Following testing, user feedback and marketing/engineering requirements will be used to measure the success of our product.

Chapter 2

Proposal

2.1 The Problem and Motivation

Existing tools for the visually impaired community are insufficient in alerting the user to all hazards and obstacles which may threaten their safety, health or independence. Through interviews with subject experts, E4E has identified the following common sources of difficulty for people with lo-vision:

- Traffic Signals
- Construction Zones
- Bicyclists
- Waist-high barriers
- Low hanging obstacles

The QL+ lab puts a particular emphasis on developing technology that can improve the quality of life for veterans who have been wounded in service to our country. While E4Es device will certainly perform for any person with lo-vision, we are enthused to work on a project through a lab that takes into special consideration those who have served so bravely.

2.2 The Objectives and Solution

The objective of this project is to develop an assistive device through the QL+ lab which will aid the visually impaired in safely navigating the world in their daily lives.

- Design an effective solution. The top solution may not be the cheapest, but the best system at the best value.
- Design a standalone solution. The software/hardware system must not require regular updates or servicing (upkeep).
- Design a solution that is highly preemptive in nature. The system must be efficient in warning its users of prospective future obstacles.
- Design a solution that looks presentable. The products will be revealed to important clients and therefore must be appealing at first encounter.
- Design a reliable solution. System reliability is very important when dealing with users who are visually impaired.
- Design two exact solutions simultaneously. The client needs to have a completed system on both the east and west coast for presentation means.

Chapter 3

Background

3.1 Related Works

After conducting research on existing products, we found several devices that set out to meet some of the same project objectives weve defined. One patent we found was a locator device for the visually impaired [2]. This allowed the user to locate certain objects by determining the distance and direction to the object. Another product we found was a Polysensory mobility aid [3]. The Polysensory mobility aid gave a combination of audible and tactile feedback to the user giving info about the location, distance, and brightness of whatever was in front of the user.

Perhaps the most radical of the existing products we researched was a tongue placed tactile output device [4]. The device would receive signals from a camera and send electronic signals to the tongue which would represent the input image. However, these products had several limitations that we plan on improving on. This included detecting objects in the range from approximately right below the users waist to directly in front of the users head. In addition, differentiating between static and moving objects and providing feedback through multiple mediums including vibration and audio. One of the products that we felt could be most improved on was the K-Sonar [5]. This product was very similar to our initial design. Our design however, sought to enhance the cane's handle, whereas the K-Sonar actually replaced it and changed the way the user must hold the cane. All these devices were very expensive ranging from approximately $800-$1500. Our price will be substantially less.

3.2 Interviews

Perhaps the most important part of the research we did, was interviewing subject experts (people with lo-vision) and specialists. We interviewed Scott Monett [6], a QL+ administrator who laid down the overall goal of the project, and helped cast the QL+ vision. We also took a team trip to the Central Coast Assistive Technology Center and met with John Lee [7], who is a rehabilitation specialist and was able to show us many devices that people use regularly to aid with lo-vision. We spoke with Dr. Kevin Taylor [8], a kinesiology professor at Cal Poly who's involved with several multidisciplinary kinesiology/engineering projects targeted towards helping people with disabilities.

We also spoke with Jennifer Allen-Barker [9] who works at the Disability Resource Center at Cal poly, and is a subject expert on lo-vision. Through Jennifer, we were able to meet another subject expert Laura Weiss [10]. Both women were able to explain daily hindrances, annoyances, and challenges that current assistive technology is insufficient to help with. The information we gathered from our interviews was paramount in the development of our project requirements and objectives.

All of the notes from our interviews can be found in Appendix B.

Chapter 4

Requirements

In addition to interview research and design objectives, our group developed a list of engineering and marketing requirements that provided a set of standards on which to adhere and guide the conceptual development. These guidelines and specifications are displayed in tables 4.1, 4.2 & 4.3:

Table 4.1: Marketing Requirements

Marketing Number	**Marketing Requirement**
1.	The device is cost-effective.
2.	The device has a standalone solution that does not require regular updates or servicing (upkeep).
3.	The device is highly pre-emptive and can give a early warning for future obstacles.
4.	The end product is very presentable which is good for investors and important clients.
5.	The system is reliable.
6.	The device is discrete and bystanders will not be affected by or aware of the system.
7.	System must be compatible with existing assistive devices such as canes, guide dogs, etc.

Table 4.2: Engineering Requirements[1]

Marketing Number	**Engineering Requirement**	**Justification**
2	1.System will be loaded with static range values (1-10ft) available for user selection using arrow buttons 2.Sys- tem will not be dependent on existing data, only real-time processing 3. The software must be less than 50mbs of memory.	The user should not have to update the sys- tem in any way. Lack of data dependency en- sures that system will not require updates of any kind. The software must be robust so that it doesnt need to be patched.
1	1.System must cost less than $500 while fulfilling all requirements.	The product must be effective and affordable so that it is a realistic commodity for the average population.
3,5	1.System will detect 95% obstacles in a 10 ft radius 2.System must have an error rate of $\leq$ 1% 3. System should warn user within 1 second of obstacle detection	The system must be able to explain to users where near obstacles stand to prevent possible injuries. The users safety is high priority and therefore the system must not error often.
6	1.System will provide output only to user through Bluetooth ear device 2.System will only take input from user through voice activation enabling button/trigger 3.System must weigh less than 10 ounces.	The system must only communicate with the specific user. Outsider communication must not be viable because it will cause system problems. The system must be light and unobtrusive so that surrounding population is not aware.
4	1.Unused ports and components will be covered with thin rubber plugs/tabs 2.Primary functional system will be 2 x 5 x 5.	Open/Visible components are not appealing to the eye. A smaller, compact system is more aesthetically pleasing than a bulky system.

[1] "This specification exists primarily to identify the type of constraints which should be imposed on the design. As such, the specific values supplied are estimations and do not represent immutable constraints."

Table 4.3: Engineering Requirements Cont.[1]

6	Provide audio feedback through headphones every 5 seconds. Deactivate with status button on package.	Maintains reliability and efficiency.
5	It will have a casing that can withstand a 3ft drop.	Will need a protection case to protect the device from shock.
5	Battery must be strong enough to provide 15 hrs of runtime.	Long battery life is crucial for reliability.
2,5	System must make an audible warning of 65dB when obstacle is detected.	Must be able to warn user if user is not wearing a headset.

[1] "This specification exists primarily to identify the type of constraints which should be imposed on the design. As such, the specific values supplied are estimations and do not represent immutable constraints."

Chapter 5

Design

5.1 Technical Overview

The handle system as a whole can be easily divided into a software element and hardware element. Arduinos development environment allows easy and quick integration between the two, as described below.

5.1.1 Hardware Specifications

Arduino Nano 3.0

The Nano was chosen due to its miniature size and generous characteristics.

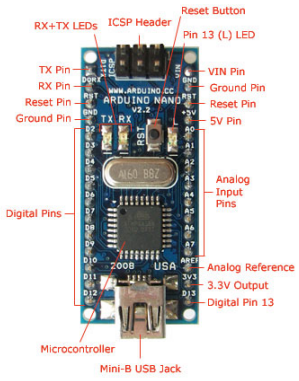


Figure 5.1: Arduino Nano 3.0

Table 5.1: Arduino Nano Specifications

Microcontroller	ATmega328
Operating Voltage (logic level)	5 V
Input Voltage (recommended)	7-12 V
Input Voltage (limits)	6-20 V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	8
DC Current per I/O Pin	40 mA
Flash Memory	32 KB (ATmega328) of which 2 KB used by bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz
Dimensions	0.73" x 1.70"

Programming

It utilizes the ATmega328 processor which is easily programmed using a simple micro usb cable in addition to Arduinos open-source software package, further described in Software Specifications. The ATmega328 comes preburned with a bootloader, which communicates using the original STK500 protocol, allowing a programmer to upload new code without the use of an external hardware programmer.

Power

The Arduino Nano can be powered via 5V regulated external power supply (pin 27), 6-20V unregulated external power supply (pin 30), or through the Mini-B USB connection. The highest input voltage source is automatically selected as the power source. Additionally, the FTDI FT232RL chip on the board is only powered when the board is powered over USB, meaning the 3.3V output which is supplied by the FTDI chip is not available using the remaining two power options listed above.

Memory

The ATmega328 has 32 KB of flash memory for storing code, including the 2 KB used for the bootloader. Additionally, it is packaged with 2 KB of SRAM and 1 KB of EEPROM.

5.1.2 Input and Output

Arduino Nano 3.0

Each of the fourteen digital pins on the Nano may be used as an input or output. These pins may bet set accordingly by using the pinMode(), digitalWrite(), and digitalRead() functions provided by the Arduino software language. Each pin operates at 5 volts and can provide or receive a maximum of 40 mA. Also, each pin has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. Some of the pins have specialized functionality:

- Serial: 0 (RX) and 1 (TX). These are used to receive (RX) and transmit (TX) TTL serial data to the FTDI USB-to-TTL Serial Chip.
- External Interrupts: 2 and 3. These pins can be configured to trigger an interrupt on a low value, rising or falling edge, or a change in value.
- Pulse Width Modulation: 3,5,6,9,10,11. Provide 8-bit PWM output using the analogWrite() function.
- Serial Peripheral Interface (SPI): 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins support SPI communication, however, the Adruino language does not provide custom functions.
- LED: 13. Uses the build-in LED connected to pin 13. When the pin is HIGH the LED is on, and when the pin is LOW the LED is off.

Each of the 8 analog inputs provide 10 bits of resolution (i.e. 1024 different values). By default the inputs measure from ground to 5V, however, the upper range can be changed using the analogReference() function. Pins 4 (SDA) and 5 (SCL) support I2C (TWI) communication using the Wire library.

- AREF: Reference voltage for the analog inputs.
- Reset: Bring LOW to reset the microcontroller.

LV-MaxSonar -EZ3 High Performance Sonar Range Finder

Requiring only 2.5V - 5.5V power, the LV-MaxSonar®-EZ3™weighs 4.3 grams and provides incredibly accrurate short to long range detection and ranging. The EZ3 detects objects from 0 to 254 inches (6.45 meters), providing range data from 6 inches to 454 inches with 1 inch resolution. The sensors output formats include pulse width output, analog voltage output, and serial digital output.

Figure 5.2: LV-MaxSonar®-EZ3™

Features

- Continuous variable gain for beam control and side lob suppression
- Object detection includes zero range objects
- 2.5V to 5.5V supply with 2mA typical current draw
- Readings can occur up to every 50mS (20-Hz rate)
- Free run operation can continually measure and output range information
- Triggered operation provides the range reading as desired
- All interfaces are active simultaneously
- Serial, 0 to Vcc, 9600Baud, 81N
- Analog, (Vcc/512)/inch
- Pulse width, (147uS/inch)

- Learns ringdown pattern when commanded to start reading
- Designed for protected indoor environments
- Operates at 42KHz
- High output square wave sensor drive (double Vcc)

Benefits

- Low cost sonar ranger
- Sensor dead zone virtually gone
- Quality beam characteristics
- Triggered externally or internally
- Fast measurement cycle
- Reliable and stable range data
- Lowest power ranger
- Mounting holes provided
- Reports range reading directly
- Choice of 3 output formats

Beam Characteristics

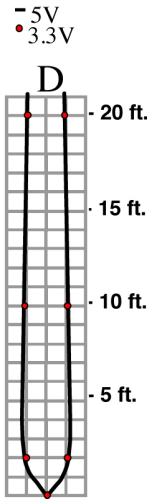


Figure 5.3: LV-MaxSonar®-EZ3™Beam

The beam characteristic to the left was made with an 11-inch wide board moved left to right with the board parallel to the front sensor face and the

sensor stationary. This shows the sensors range capability. The beam intially widens within the first 3 feet and begins to narrow slightly as it progresses towards 20 feet.

Note: The displayed beam width of (D) is a function of the specular nature of sonar and the shape of the board (i.e. flat mirror like) and should never be confused with actual sensor beam width.

LV-MaxSonar®-EZ3™Circuit

The LV-MaxSonar®-EZ3™sensor functions using active components consisting of an LM324, a diode array, a PIC16F676, together with a variety of passive components.

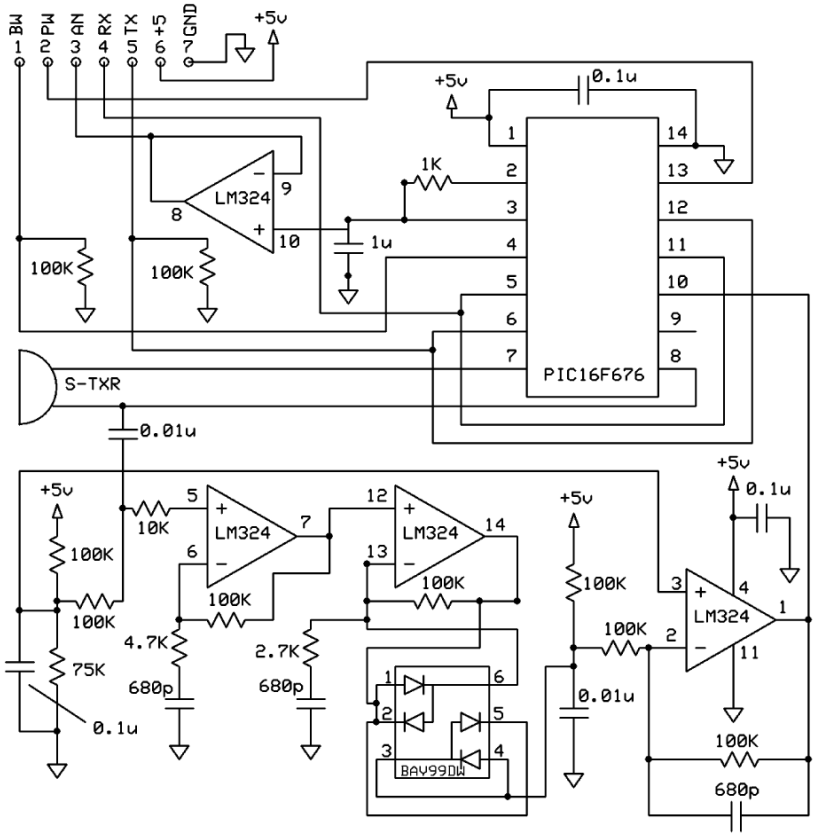


Figure 5.4: LV-MaxSonar®-EZ3™Circuit

- GND - DC power supply return. For best operation, GND and Vcc should be ripple and noise free.
- +5V - Vcc operates on 2.5V - 5.5V. 3mA current is recommended for 5V and 2mA for 3V.
- TX - If BW is held low or open, Tx output delivers asynchronous serial with an RS232 format. The output is ASCII character R followed by 3 digits representing the range in inches up to a maximum of 255, lastly followed by a cariage return.
- RX - While HIGH or OPEN, the EZ3 measures range and output. If held low, the sensor stops ranging.
- AN - Outputs analog voltage with a scaling factor of (Vcc/512) per inch. A supply if 5V yields 9.8mV/inch and 3.3V yields 6.4mV/in. The output is buffered and corresponds to the most recent range data.
- PW - Outputs a pulse width representation of range. The distance can be calculated using the scale factor of 147uS per inch.
- BW - Leave open or held low for serial output on TX output. When held high, the TX sends a pulse (instead of serial data), suitable for low noise chaining.

10mm Shaftless Vibration Motor

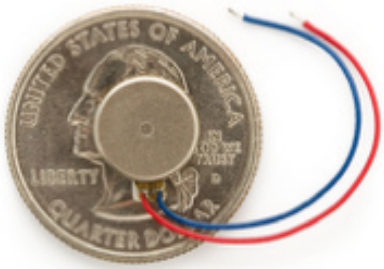


Figure 5.5: 10mm Shaftless Vibration Motor

Table 5.2: Vibration Motor Specifications

Specification	Value
Voltage [V]	3
Frame Diameter [mm]	10
Body Length [mm]	3.4
Weight [g]	1.2
Voltage Range [V]	2.5-3.8
Rated Speed [rpm]	12000
Rated Current [mA]	75
Start Voltage [V]	2.3
Start Current [mA]	85
Terminal Resistance [Ohm]	75
Vibration Amplitude [G]	0.8

Nickel Metal Hydride Battery Pack

After much debate, the team decided to work with the local BatteriesPlus+ store in San Luis Obispo to design a custom power solution. It was important that the power source be mobile, small, and rechargeable. After agreeing on utilizing NiMH batteries, the team discovered the placement of the battery pack played a large role in its shape. It was concluded that a hexagonal shape design would work best, given the battery pack would fit nicely in the back of the handle.

(a) Side

(b) Bottom

Figure 5.6: Nickel Metal Hydride Battery Pack

5.1.3 Software Specifications

The open-source Arduino 0022 environment was a primary reason for purchasing the Arduino Nano V3. The free environment makes it easy to write, compile, and upload code to any Arduino development board. The software is written in Java and runs on multiple operating systems, including Windows, Mac OS X, and Linux. The software environments simplicity and flexibility seemed perfect for project development among a team of six.

The Arduino development environment provides a number of tools necessary for programming, including a text editor for writing code, message area to display feedback and errors, text console to display environment messages, tool bar containing buttons for common functions, and a series of menus. It is prepared to easily connect and communicate with existing Arduino development boards.

5.2 System Overview

The system design revolved around a few fundamental features of the project requirements. The system was required to be mobile, presentable, and detacheable. It was important that the end product be both easy to use and simple to install. From this, after generating different possible design possibilities, the team decided on designing a handle system. After numerous generations of designs, the handle system reflected both a minimalized and simple solution.

The handle was designed to easily slip over a canes handle. The back of

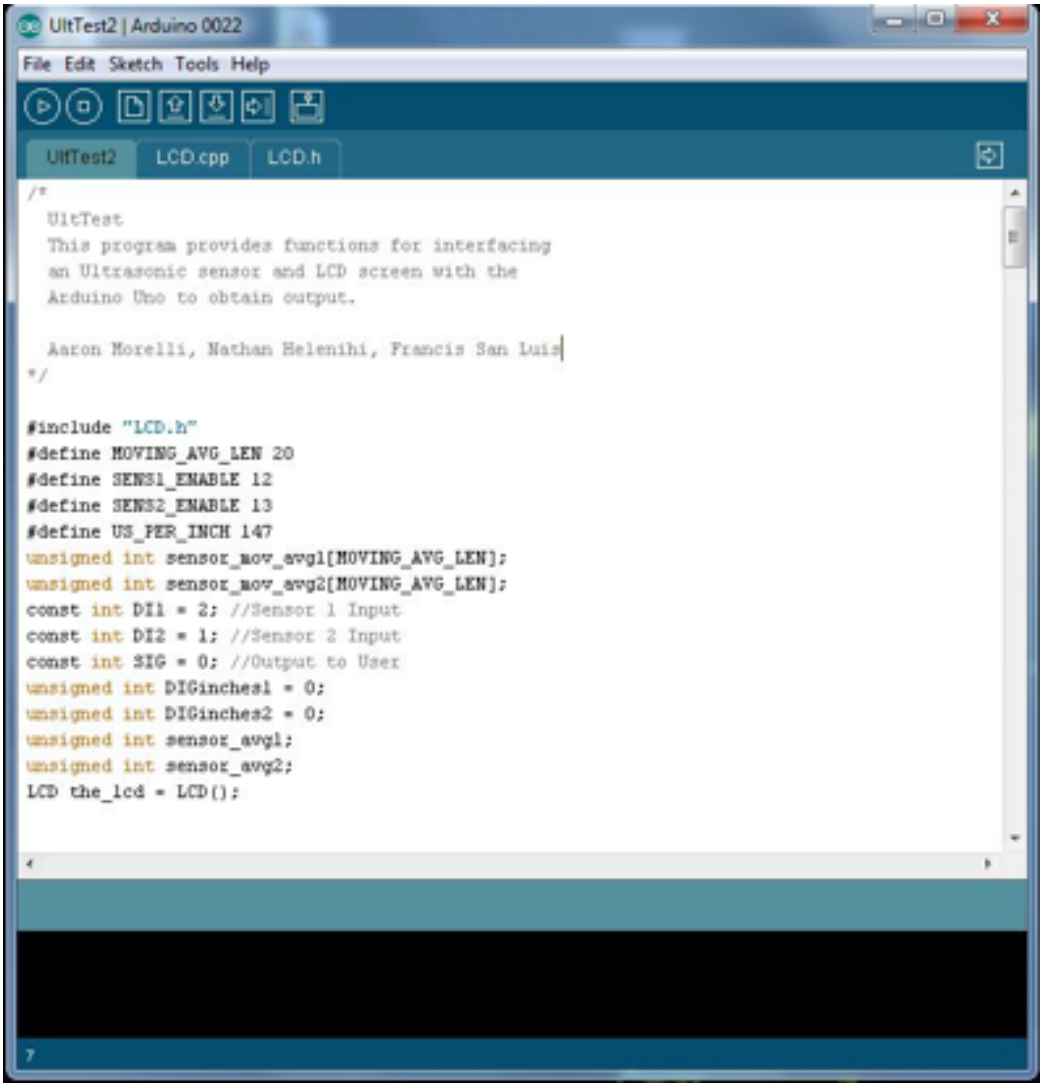
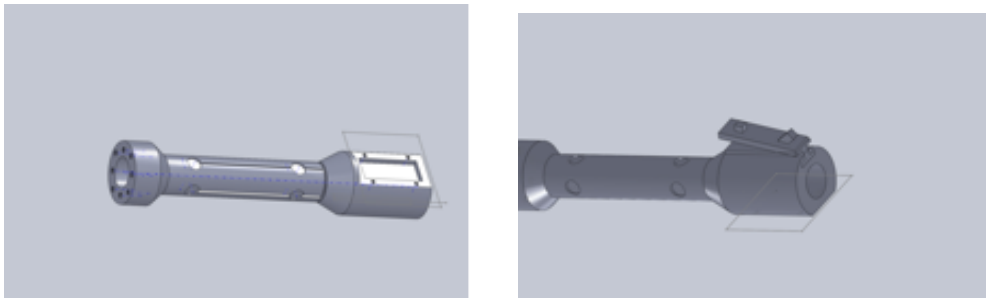


Figure 5.7: Arduino Environment

the handle (left) has a detacheable cavity for the NiMH battery pack. The center of the handle contains eight vibrational motors to provide feedback to the user. The front four motors will vibrate when the bottom sonar sensor detects an object, and the back four for the top sensor. Wire trenches between the battery, motors, and microcontroller are provided to easily route the wiring.



(a) Top (b) Bottom

Figure 5.8: Beta Prototype

The bottom front of the handle has a cavity for the Arduino Nano microcontroller, accompanied by a removeable cover. On the top of the back end of the cane is a detacheable mount housing the sonar sensors. The mount has the ability to pivot, allowing the top sensor to range between 90 degrees and 20 degrees to the ground. This allows users of different heights to normalize the sensors detection scopes. The sensors will be connected to the microcontroller via the same hole being utilized by the battery pack and vibrational motors. The top sensor is positioned 90 degrees with respect to the mounts surface and the bottom sensor 60 degrees. The angle between the two sensors (30 degrees) is fixed, and so was designed to be optimal for a person 59 tall, using a 53 cane, having the lower sensor adjusted to be parallel with the ground, grasping the cane handle 33 from the ground, and receiving warnings about head-height obstacles 5 out from their head.

The numbers used are considered to be the average case. Deviation from them would result in very slight changes to the systems performance. The largest (still slight) discrepancy would be seen in the user whos proportions (ratio of head-height to cane-grasping height) were noticeably different from those we used.

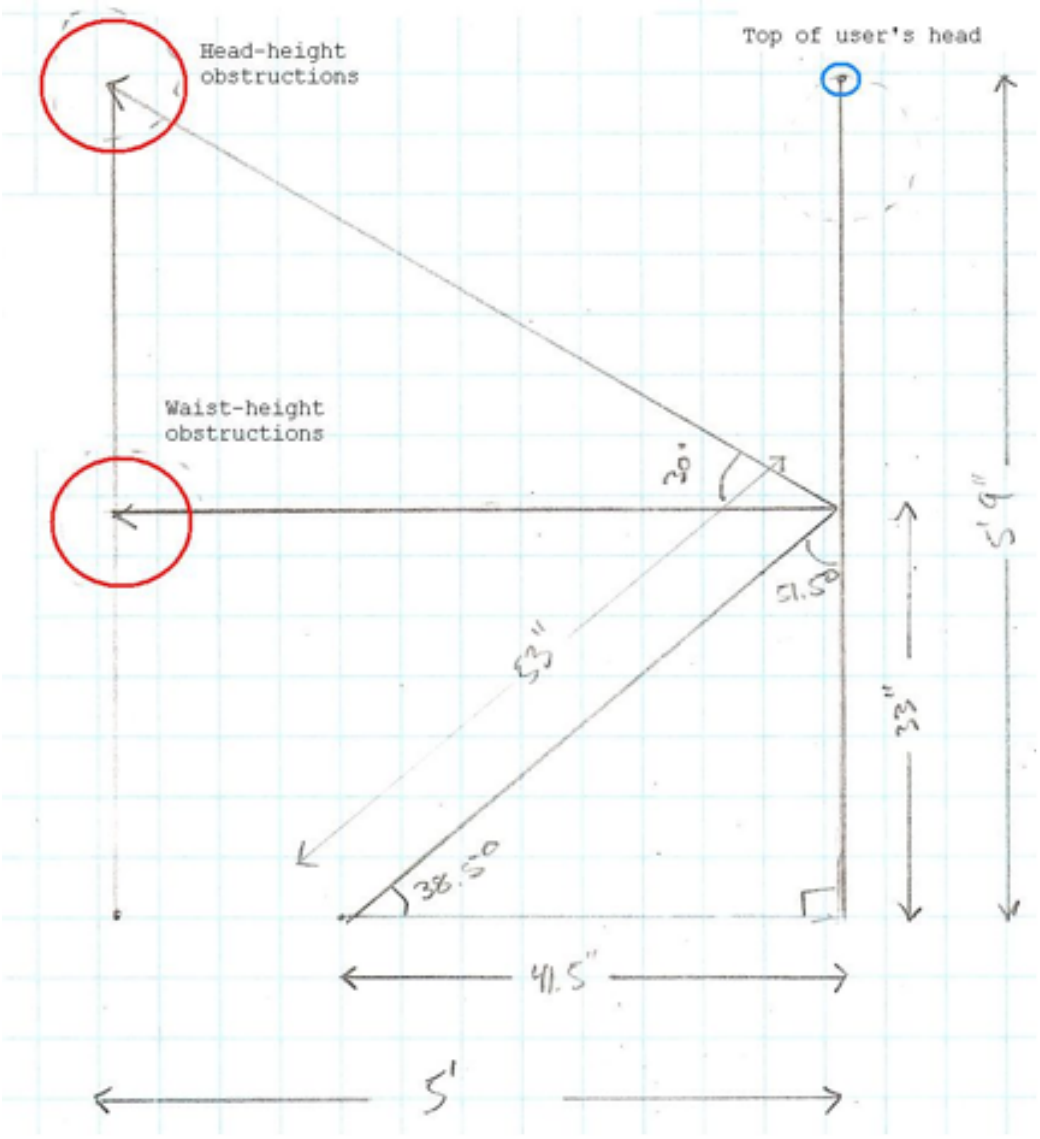


Figure 5.9: Sensor Angle Diagram

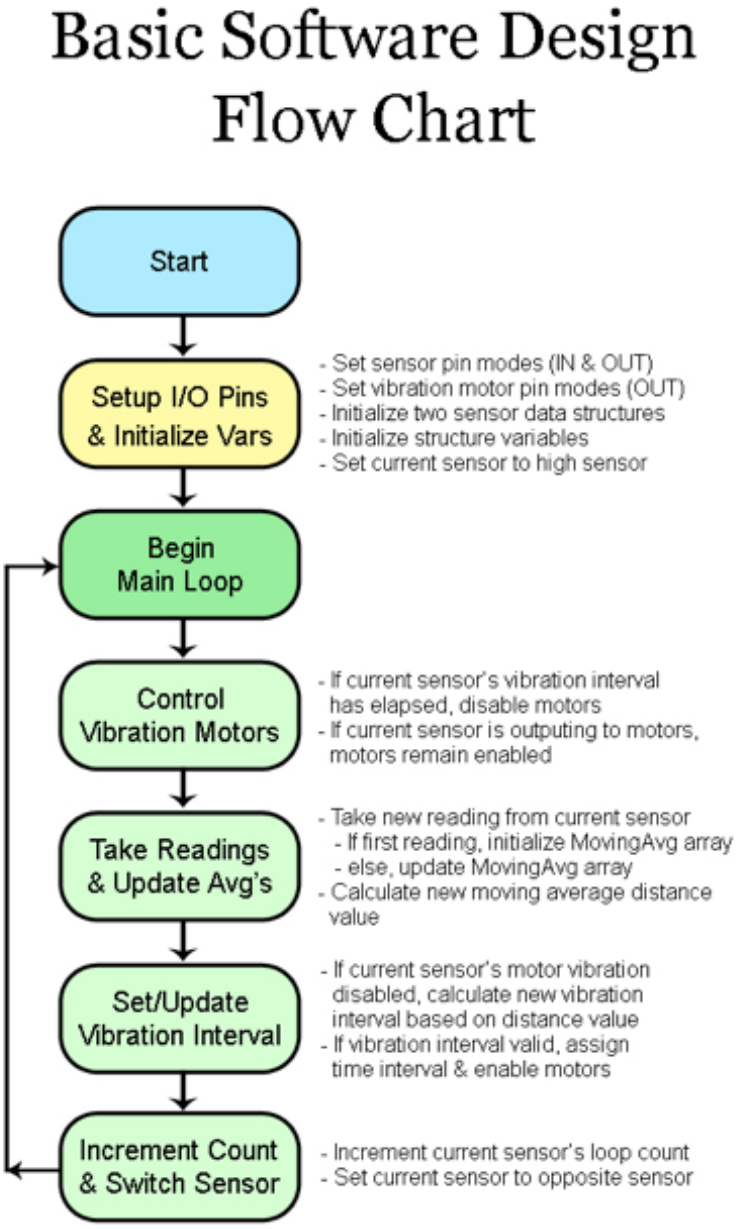


Figure 5.10: Software Design Flow Chart

Chapter 6

Implementation

6.1 Hardware

The hardware platform for E4E's Radar for the Blind has been through two major revisions

6.1.1 Alpha Prototype (AKA "Ghost Buster")

The first build of our device, deemed the alpha prototype, features two weather-proof ultrasonic sensors [11] [12] for detecting high and low obstacles attached to a shorter than usual blind cane. Also attached to the cane is a metal box [13] providing a platform for the following system components:

- 6 AA cell batteries to power the device
- an ATMega 328 micro-controller on the Arduino Uno platform [14]
- an LCD display for debugging purposes
- 3 vibrating motors to provide user feedback [15]
- NPN BJT transistor to switch motors on/off
- a 5V regulator [16] to power the LCD, sensors, and motors

6.1.2 Beta Prototype (AKA "Sight Saber")

The second build of our device, deemed the beta prototype, features the same two ultrasonic transducers for detection and ranging. These sensors however have a much smaller profile due to a non-weather proof packaging. It was decided that trying to weather proof the entire prototype was an

unimportant goal in this phase of the product's development, so a smaller, and less expensive, and non-weatherproofed packaging of the same transducer was picked for this prototype. A new platform for the ATMega 328 chip was also selected. For this protoype we used Arduino's Nano board. This board features only half the program space of the Uno, but is still more than our code base requires.

The Nano has a much smaller footprint. Another major difference is the placement and number of vibrating motors. This prototype features 8 motors total. Four of these are located on a higher position on the handle, and four a located lower. The motors are separated in order to report to the user, independent of each other, when either a high, or low obstruction is detected. The whole system is now integrated into a handle which fits over the handle on the user's cane. This build has no LCD, true to the theoretical final version. Integrated onto the handle are the following components:

- 9V 1400mAh rechargeable battery pack
- ATMega 328 micro-controller on the Arduino Nano platform
- 8 vibrating motors
- 2 NPN BJT transistors to switch each set of 4 motors independently
- 5V regulator to power sensors and motors

6.2 Software

The software for E4E's Radar for the Blind has been through four major versions.

6.2.1 Software v1 (developed on alpha prototype)

The first version of our software essentially ran the sensors continuously and simultaneously, taking readings as fast as they could, and reading analog voltages from them. Using the Arduino's on-board ADC, a digital value (0-1023) was obtained from the analog voltage provided by the sensors at any point in time. This system was found to work reasonably well, with a couple of problems. One particularly interesting bug, was a regularly occurring short reading on both sensors. We discovered by covering one sensor with a hand, that the problem went away. We determined from this that the sensors must be interfering (one sensor picking up the other's ping).

Another inefficiency realized, was reading an Analog voltage. The sensors provide three types of output: Analog, Digital (PWM) and Serial (RS-232 like). We decided that instead of reading an analog voltage, and converting it to digital, we should probably just read the digital output of the sensor. The improvement would theoretically get rid of any roundoff or inaccuracy introduced by the Arduino's on-board ADC. The code was also implemented as one big loop, which infringes on readability and maintainability.

6.2.2 Software v2 (developed on alpha prototype)

The second version of our software attempted to use a task-based approach. Professor John Ridgely has implemented a task-based software development interface for the Mechanical Engineering department's mechatronics course offerings. Adapting his code to work with the ATMega 328, which is slightly different from the ATMega 128 on the mechatronics boards, we would create a system which operates as a state machine. Tasks are objects which are instantiated and given an interval at which to run. Tasks such as take reading and toggle vibrate were developed. This way, the frequency at which certain tasks were performed could be altered to easily find a balance between taking readings and updating the state of user feedback.

This implementation was ultimately found to be more cumbersome. After a couple of weeks fiddling with this implementation, we decided that the added code-base wasn't worth the hassle on a system as simple as ours. The tasks which are system needs to perform are few, and of equal importance.

6.2.3 Software v3 (developed on alpha prototype)

The third implementation of our software essentially broke the tasks of taking a reading, calculating a vibration interval, toggling the vibrating motors on/off into separate functions which would be called in their proper sequential order in the software's main loop. Also, enable lines for the sensors were used, rather than left floating, allowing us to enable the sensors one at a time (one time through the loop, enable sensor one, next time sensor two, repeat). This fixed the early problem of a regularly occurring short reading due to sensor interference. We also began taking readings from the sensor's digital PWM output rather than analog voltages to reduce any loss of accuracy which may have been introduced by the Arduino's on-board ADC.

6.2.4 Software v4 (developed on beta prototype)

The second hardware prototype features separated sets of vibrational motors to alert the user of high or low objects independently. To utilize this hardware improvement, we changed the code base. The code now stores data for two alert systems (high and low) independently. The same functions are used for taking readings, setting vibrational intervals, and toggling vibrators, but now the data for each system is stored in a struct representing that system. Each system's data and state is updated every other time through the program's main loop.

Chapter 7

Testing

7.1 Testing Philosophy

For a device such as ours, testing must be a cautious matter, because the users safety is at stake. For this reason, our testing had to be done in a supervised, controlled manner to ensure the safety of the tester. The quantifiable output of our device to test is a vibration if an obstacle is detected, or silence if no obstacle is detected. If too many false positives are given, the device becomes useless. However, if too many false-negatives (undetected obstructions) are realized, then the device may actually be dangerous if the user is relying on its feedback heavily. For this reason, false positives were considered as strictly less important than false negatives, as a poorly performing device is a better result than a dangerous one.

7.2 Testing Plan

Our testing was done in two major phases: system and component testing. Component tests were executed to verify and explore the functionality of the sensors. Examples of component testing included:

- Range finding capabilities of ultrasonic sensors
- Width of beams emitted by ultrasonic sensors
- Vibrational feedback to the user when an object was detected

Ultrasonic sensors with narrow beams were chosen specifically, as described in the Design section of this document. E4E conducted testing to

Table 7.1: Beam Width of Ultrasonic Sensors: 1st Test Results

Distance Actual (inches)	Distance Detected (inches)	1st Test Left Boundary (inches)	1st Test Right Boundary (inches)	1st Test Cone Width (inches)
12	12	0	3.5	3.5
24	24	.6	4.75	5.35
36	36	1.4	6.3	7.7
48	48	2.6	6.75	9.35
60	60	3.25	10.25	13.5
72	72	3.8	13.4	17.2
84	84	4.75	12.25	17
96	96	3.75	14.3	18.1

Table 7.2: Beam Width of Ultrasonic Sensors: 2nd Test Results

Distance Actual (inches)	Distance Detected (inches)	2nd Test Left Boundary (inches)	2nd Test Right Boundary (inches)	2nd Test Cone Width (inches)
12	12	2.25	1	3.25
24	24	3.25	1.5	4.75
36	36	5	3.5	8.5
48	48	7.25	4.5	11.75
60	60	9.25	6	15.25
72	72	12	9.25	21.25
84	84	14	9	23
96	96	9.25	13	22.25

verify the beam width of the ultrasonic sensors. The testing procedure was set up by immobilizing the test sensor, facing down a line which was marked .5 foot increments. A large rolling whiteboard was rolled toward the center line from the left until the sensor reported its presence. The distance from whiteboard to center line was recorded. The same procedure was done from the right. The sum of the two distances from center line to whiteboard was considered to be the beam width at the marked distance from the sensor

on the center line. Measurements were not taken from top and bottom, as the beam is assumed to be conical, and thus have a symmetric cross-section. Tables 7.1 & 7.2 show the results were congruent with the sensors datasheet.

During this component testing, we discovered that the sensors would periodically give an extremely short reading. We changed the amount of readings that the software averaged, but quickly realized that wasnt the problem. By covering one sensor, the problem stopped. From this we concluded that the sensors were interfering with each other. For the next software version, we eliminated the problem by using the enable lines on the sensors rather than running them both continuously.

The entire system was initially tested by E4E members, around the QL+ lab. We approached various objects and observed the response time of our device. The lab was a bit more crowded than the typical environment this device is likely to be used in however, so we also brought it outside to test in more open areas. We repeatedly tested and tuned the device software until we felt confident in handing the cane over to subject experts to receive their critique and acclaim.

7.3 User Feedback

After completing assembly, programming, and some in-house testing of the beta prototype, E4E was confident enough to hand the device over to Jennifer Barker, who graciously (and excitedly) gave her time to provide some expert feedback with respect to the devices performance.

Jennifer made some great suggestions on how the device could be improved. She mentioned that an adjustable vibration intensity would be good, as a users cane is naturally vibrated during use, depending on the type of terrain the user is walking on. When asked about the weightiness of the device, Jennifer observed that the weight would likely cause wrist fatigue over a long period of time. She also pointed out that the handle would probably be too large for a child, but that children wouldnt be introduced to this device until they were already proficient with a cane. The handle is also different from the shape that the user would be familiar with (a golf grip with one flat side). She pointed out that if we kept a cylindrical shape to the handle, a tactile cue should be added to indicate which way the cane should be held, in order to keep the sensors at the appropriate facing. Also, the detection of head-height obstacles was found to be intermittent, and often too late. We

reasoned that head-height obstacles would be within the transducers detection lobe only briefly by comparison to obstacles approaching at waist height, and so the averaging algorithm should be augmented for the top sensor, to be more responsive.

Jennifer also pointed out that the range at which obstacles were detected was calibrated well, and provided a good forewarning that was neither too early or too late. She mentioned that the overall comfort of the handle wrapped with a bike grip was even better than the standard grip on a white cane. Jennifer was overall very impressed with the product we had turned out in 3 quarters, and sincerely hopes that the project will be continued and improved upon.

The system was also tested by Charles, a veteran who uses a cane frequently. Charles was pleased with the product, and had four major suggestions for improvement. He said that the cane should be made lighter, collapsible, and should have a convenient way of quickly turning off (for talking with people, or walking up to a destination). He also mentioned that keeping the sensors at the appropriate facing was challenging.

Chapter 8

Conclusion

8.1 Current Features

According to the marketing requirements, the latest development is a very reliable and pre-emptive obstacle detection device for persons with low vision. The system is a standalone solution requiring no frequent software updates or services. The intended use of the device is for sliding on the handle of the cane or simply using it by freehand. This allows it to be compatible with existing assistive devices including canes and guide dogs. The size and singularity of the system maintains discretion for the user and will not affect bystanders substantially. In regards to the engineering requirements, the device weighs approximately two pounds which did not fulfill the targeted value of being less than ten ounces. Future iterations of the product should strive to reduce the overall weight.

The device was cost-effective having the components to construct the device cost approximately $150. Other requirements that will be further prioritized in the next iterations of the system will include withstanding a 3 foot drop test and providing audio feedback for users. In addition, the 1400 mA hour battery designed for our system was not explicitly tested. The average current draw for the system wasn't tested and so no assumption can be made about run time. However, the battery never required recharging throughout our testing.

8.2 Suggested Improvements

The system is meant to provide distinct alerts for a low or high obstructions. The difference in vibration (top of handle vs. bottom) was found to be subtle at best, and indistinguishable at worst. Future iterations on the system should set out to improve the distinctiveness of the two alerts. Also, the device was also found to be too weighty. The next iteration of this product should aim to reduce the weight of the device in order to avoid fatiguing the users wrist during extended use. Improved performance could possibly be realized by the use of an infrared range-finding sensor for the head-height obstructions. Head-height obstacles are detected at a particular moment, and an IR sensor may prove more responsive than the ultrasonic range-finder used currently.

Appendices

Appendix A

Source Code

```
1  /*
2    RFBProto2
3    This is the source file for the second prototype of
4    Engineering For Eyes' "Radar for the Blind"
5
6    Changes:
7        The code now has two detection zones (low and high
             ),
8        which are signaled to the user independently via
             two
9        different areas of vib motors on the handle.
10
11       The test/debugging lcd code isn't in this version.
12
13   Aaron Morelli, Nathan Helenihi, Francis San Luis
14 */
15
16 /* Define constants */
17 #define MOVING_AVG_LEN_L 12
18 #define MOVING_AVG_LEN_H 3
19
20 #define US_PER_INCH 147
21 #define H 0
22 #define L 1
23 /* Name some pins we'll be using */
```

```
24 #define DIH 8                    //High Sensor Input
25 #define DIL 4                    //Low Sensor Input
26 #define SIGH 9                   //High Output to User
27 #define SIGL 5                   //Low Output to User
28 #define SENS_ENABLE_HIGH 7  //enable line for High
       Sensor
29 #define SENS_ENABLE_LOW 3   //enable line for Low
       Sensor
30
31 /* This struct will contain all the data that the high
       and low
32  * alert systems will need to function */
33 typedef struct{
34   unsigned long vib_time;                  //time at
       which to activate vibrator
35   unsigned long vib_interval;              //interval at
        which to pulse vibrator
36   unsigned int *readings; //array of most recent
       readings
37   unsigned int distance;                   //the current
        distance measurement of the system (averaged)
38   unsigned int temp_reading;               //the most
       recent one-off measurement
39   boolean vib_on;                          //true if the
        systems vibrator is currently on
40   int input_pin;                           //system's
       input pin
41   int output_pin;                          //system's
       output pin
42   int enable_pin;                          //system's
       enable pin
43   boolean first_reading;                   //true if
       this system has taken no readings
44   unsigned long count;                     //how many
       times this system has been updated
45 } ALERT_SYSTEM;
46
47 /* Declare variables for the program */
```

```
48 ALERT_SYSTEM systems[2];          //array containing
       both the high and low systems
49 int current_sensor;               //tells us which
       sensor to use;
50 unsigned int high_readings[MOVING_AVG_LEN_H];
51 unsigned int low_readings[MOVING_AVG_LEN_L];
52
53
54 /* Set pin modes for system */
55 void setup() {
56   pinMode(DIH, INPUT);
57   pinMode(DIL, INPUT);
58   pinMode(SIGH, OUTPUT);
59   pinMode(SIGL, OUTPUT);
60   pinMode(SENS_ENABLE_HIGH, OUTPUT);
61   pinMode(SENS_ENABLE_LOW, OUTPUT);
62
63   digitalWrite(SIGH, LOW);          //inititalize
       vibrators to be off.
64   digitalWrite(SIGL, LOW);
65   current_sensor = H;               //start with high
       sensor
66   systems[H].vib_time = 0;
67   systems[H].vib_interval = 0;
68   systems[H].vib_on = false;
69   systems[H].input_pin = DIH;
70   systems[H].output_pin = SIGH;
71   systems[H].enable_pin = SENS_ENABLE_HIGH;
72   systems[H].first_reading = true;
73   systems[L].readings = low_readings;
74   systems[H].readings = high_readings;
75   systems[L].vib_time = 0;
76   systems[L].vib_interval = 0;
77   systems[L].vib_on = false;
78   systems[L].input_pin = DIL;
79   systems[L].output_pin = SIGL;
80   systems[L].enable_pin = SENS_ENABLE_LOW;
81   systems[L].first_reading = true;
```

```
82    Serial.begin(9600);
83  }
84
85  /*
86   * Main loop repeatedly performs following tasks,
       alternating between sensors:
87   *  1 - Control Vibration (turn on or off)
88   *  2 - Take Readings and Update systems' averages
89   *  3 - Set a new vibration interval for each system
90   */
91  void loop() {
92
93    /* CONTROL VIBRATOR */
94    if( systems[current_sensor].vib_time <= millis() ){
95      systems[current_sensor].vib_on = false;
96    }
97    if( systems[current_sensor].vib_on ){
98      digitalWrite(systems[current_sensor].output_pin,
          HIGH);
99      printSensorInfo(current_sensor);
100   }
101   else{
102     digitalWrite(systems[current_sensor].output_pin,
          LOW);
103   }
104
105   /* TAKE READINGS - UPDATE AVGS */
106   systems[current_sensor].temp_reading =
        takeSensorReading(current_sensor);
107   if(systems[current_sensor].first_reading){
108     initMovingAvg(current_sensor);
109     systems[current_sensor].first_reading = false;
110   }
111   else{
112     updateAverage(current_sensor);
113   }
114
115   /* SET VIB INTERVAL */
```

```
116     if(!systems[current_sensor].vib_on){
117
118       systems[current_sensor].vib_interval = calcInterval
              (systems[current_sensor].distance);
119
120       //if(systems[current_sensor].vib_time > 0){
121       if(systems[current_sensor].vib_interval > 0){
122         systems[current_sensor].vib_time = millis() +
                systems[current_sensor].vib_interval;
123         systems[current_sensor].vib_on = true;
124       }
125
126     }
127
128     /* INCREMENT COUNT AND SWITCH TO OTHER SENSOR */
129     systems[current_sensor].count++;
130     if(current_sensor){
131       current_sensor = 0;
132     }
133     else{
134       current_sensor = 1;
135     }
136 }
137
138 void printSensorInfo(int sensor){
139     if(sensor == H)
140     {
141         Serial.print("High: ");
142     }else{
143         Serial.print("Low:  ");
144     }
145
146     Serial.print(systems[sensor].distance);
147     Serial.println();
148 }
149
150 /*
```

```
151   *Turns the sensor on, takes a reading, turns the
          sensor
152   *off, converts PWM reading to inches, and returns that
          value.
153   */
154 unsigned int takeSensorReading(int sensor){
155     unsigned long duty;
156     unsigned int inches;
157     digitalWrite(systems[sensor].enable_pin, HIGH);
158     duty = pulseIn(systems[sensor].input_pin,HIGH);
159     digitalWrite(systems[sensor].enable_pin, LOW);
160     inches = duty/US_PER_INCH;
161     return inches;
162 }
163
164 /*
165  * Determine the frequency of the vibrations
166  */
167 unsigned long calcInterval(unsigned int distance)
168 {
169     unsigned long interval = 0;
170
171     if(distance > 0 && distance <= 96){
172       //interval = 150-distance;
173       //interval = 400 - distance*3.5;
174       if(distance > 75){
175         interval = 160 - 35*log(distance);
176       }else if(distance > 50){
177         interval = 225 - 35*log(distance);
178       }else{
179         interval = 250 - 35*log(distance);
180       }
181     }
182
183     return interval;
184
185 }
186
```

```
187 /* on first sensor read, fills moving average with
       first value */
188 void initMovingAvg(int sensor)
189 {
190     int i;
191     int numReadings = MOVING_AVG_LEN_L;
192     if(sensor == H){
193       numReadings = MOVING_AVG_LEN_H;
194     }
195     for(i=0;i<numReadings;i++){
196       systems[sensor].readings[i] = systems[sensor].
             temp_reading;
197     }
198 }
199
200 /* updates the moving average of a particular sensor.
       */
201 void updateAverage(int sensor)
202 {
203   int i = 0;
204   int numReadings = MOVING_AVG_LEN_L;
205   if(sensor == H){
206     numReadings = MOVING_AVG_LEN_H;
207   }
208   systems[sensor].readings[systems[sensor].count\%
        numReadings] = systems[sensor].temp_reading;
209   systems[sensor].distance = 0;
210   for(i = 0; i < numReadings; i++){
211    systems[sensor].distance += systems[sensor].readings
         [i];
212   }
213   systems[sensor].distance /= numReadings;
214 }
```

Appendix B

Interviews

Laura Weiss - low-vision subject expert contacted through Cal Poly Disability Resource Center

- Phone application that could consolidate device functionality
- Detect objects appearing in walkways. Dangling branches problematic
- Prefers not to use the cane so people do not see her as 'blind'
- Vibration feedback for obstacle detection
- Headset for audio feedback or bluetooth headset
- Alert user of when to replace rechargeable battery
- Wrist strap (the device similar to a digital camera)
- Minimal damage to the device if walking and tripping (drop test)
- Rubber coating or similar material enclosure for durability
- Buttons for turning On/Off
- Size, shape, texture, and color important characteristics
- Use the device with one hand
- Buttons are easier interface than talking to device, less disruptive to environment

John Lee - rehabilitation specialist at Central Cost Center for Assistive Technologies

- Minimize the amount of devices carried. (Phone apps)
- Important to have backup if a device stops working
- Vision enhancing devices, magnifying images
- Infrared/reflective dot system that moves mouse of a computer by moving ones head
- Large remotes/keyboards with large buttons
- Important to provide buttons that represent their functionality through color, shape, and texture

Dr. Kevin Taylor - Professor of Kinesiology at Cal Poly

- Large difference between persons with low vision and blind
- People lost independence with vision loss
- Low vision people lost freedom to do things alone

Jennifer Allen-Barker - Cal Poly DRC specialist and lo-vision subject expert

- Walks at a comfortable speed. Important not to move very fast.
- Curbs, cracks, low hanging branches, and debris on walkways need to be detected
- Skateboarders and bicyclists traveling at fast speeds. (Awareness)
- Does not use cane
- Detect barriers and poles holding up construction fences
- Construction zones and caution tape perimeters problematic
- Find a safe path around hazardous and unpredictable obstacles
- Traffic signal beeping for crosswalks helpful audio recognition

Scott Monett - QL+ Administrator

- Obstacle detection device used in addition to the users cane
- Reliability important
- No patching or firmware/software updates. Standalone device
- Build two devices concurrently

Bibliography

[1] Quality of Life Plus, "Ql+," http://qlplus.org/, 2010.

[2] S. N. . Silverman, Hildy S. (4 Lowe Rd., "Identifier/locator device for visually impaired," Patent 5 508 699, April, 1996. [Online]. Available: http://www.freepatentsonline.com/5508699.html

[3] I. S. R. H. F. W. I. Moricca, Larry S. (Churubusco, "Polysensory mobility aid," Patent 3 993 407, November, 1976. [Online]. Available: http://www.freepatentsonline.com/3993407.html

[4] W. K. K. A. M. W. Bach-y rita, Paul (Madison, "Tongue placed tactile output device," Patent 6 430 450, August, 2002. [Online]. Available: http://www.freepatentsonline.com/6430450.html

[5] Bay Advanced Technologies, "The bat k-sonar," http://www.batforblind.co.nz/, 2006.

[6] S. Monett, Oct. 2010.

[7] J. Lee, Oct. 2010.

[8] K. Taylor, Oct. 2010.

[9] J. Allen-Barker, Oct. 2010.

[10] L. Weiss, Nov. 2010.

[11] Sparkfun, "Ultrasonic range finder - maxbotix lv-wr1." [Online]. Available: http://www.sparkfun.com/products/9009

[12] ——, "Ultrasonic range finder - maxbotix lv-wr1 mounting hardware." [Online]. Available: http://www.sparkfun.com/products/9010

[13] ——, “Chameleon enclosure - black.” [Online]. Available: http://www.sparkfun.com/products/9682

[14] ——, “Arduino uno board.” [Online]. Available: http://www.sparkfun.com/products/9950

[15] ——, “Vibration motor.” [Online]. Available: http://www.sparkfun.com/products/8449

[16] ——, “Voltage regulator - 5v.” [Online]. Available: http://www.sparkfun.com/products/107