March 14th, 2013

Dr. Andrew Rawicz
School of Engineering Science
Simon Fraser University
Burnaby, British Columbia
V5A 1S6 Canada

Re: ENSC 305/440 Design Specification for an Indoor Direction Finder for the Visually Impaired

Dear Dr. Rawicz:

The attached document, "Design Specification for an Indoor Direction Finder for the Visually Impaired", outlines our design choices in our capstone project in ENSC 305/440. We seek to design and implement an indoor system of signal beacons that enable the blind and visually impaired to navigate safely and effectively to their destinations with a portable device. We named this system the **WhereTo**.

The purpose of the design specification is to convey our design choices and illustrate the technical workings of the **WhereTo** system in its proof-of-concept prototype and planned production states. This document will serve as a record of our design through the development phases.

Envied Solutions consists of four talented engineering students near graduation: Alan Fang, Phillip Peach, Shaham Shafiei, and I, Wilson Chen. For any questions or concerns you may have, please contact me by phone at 778.386.3284 or by email at wilson_chen@sfu.ca.

Sincerely,

Wilson J. L. Chen
President and CEO
Envied Solutions

Enclosure: Design Specification for an Indoor Direction Finder for the Visually Impaired

WhereTo

Design Specification for an

Indoor Direction Finder for the Visually Impaired

Project Team

Wilson Chen

Alan Fang

Phillip Peach

Shaham Shafiei

Contact

Wilson Chen
wilson_chen@sfu.ca

Submitted to

Dr. Andrew Rawicz – ENSC 440

Steve Whitmore – ENSC 305

School of Engineering Science, Simon Fraser University

Issued Date

March 14th, 2013

Executive Summary

Standard building way-finding is primarily visual. It consists typically of things like signs, arrows, paths painted on walls or floors, and other methods which are of minimal or no use to the visually impaired. In an unfamiliar building, this can leave these people at a distinct disadvantage. The **WhereTo** system by Envied Solutions aims to alleviate this disadvantage.

To use the **WhereTo** system, the building owner installs a number of beacons throughout the building. These beacons are programmed during set-up with their location inside the building and are used by the handheld unit to determine the user's position. The user can then use a **WhereTo** handheld unit (either their own or one that the building owner lends out to visitors) to navigate to a destination within the beacon coverage. To best serve the target market, all interaction with the handheld unit is audio based. The user commands the system by voice and is given directions over a pair of headphones. Should the user be unable to speak properly, the device may be driven solely by its hardware buttons. As a bonus, the handheld can alert its user to any obstructions detected – again through an audio warning.

Development is currently on-going and the system is being optimized for low-cost, high accuracy and high reliability. Detailed design and construction of a prototype system consisting of three or more beacons and a single handheld unit is enclosed. The prototype system will be ready to demonstrate in April 2013.

Table of Contents

Executive Summary i
Glossary iv
Introduction 1
Scope 1
Audience 1
System Design Overview 2
System State Tree 2
Mechanical Design 3
Message Protocol 4
Binary Frequency Shifting Keying (BFSK) 4
Handheld Device 6
Positioning Strategy 6
Electrical 7
Computational 8
Signal Beacons 10
Electrical 10
Computational 10
User Interface 11
Handheld Unit Software Interface 12
Audio Output 12
Audio Input 12
Handheld Unit Hardware Interface 12
Power Button 12
Previous Button 12
Next Button 13
Confirm/Repeat Button 13
Cancel Button 13
System Test Plan 13

Unit Testing 13
Beacon Transmitter 13
Position Calculator 13
Path Calculator 14
Collision Calculator 14
Voice Recognition/UI 14
Sound Output 14
Normal Use Testing 15
Case 1 - List Destinations by Voice 15
Case 2 - List Destination by Button 15
Case 3 - Manually Input Destination Accomplished Purely by Voice 15
Case 4 - Manually Input Destination Accomplished with Hardware Button Navigation 16
Case 5 - Navigate to Complex Destination (Requires Turning Corner) 16
Extreme Case Testing 16
Case 1 - Outside of Building but Still Receiving Signals 16
Case 2 - Purposely Broadcast Improper Signal from One Beacon to Cause Them to Be Insufficient. 17
Case 3 - Purposely Broadcast an Extra Unencoded Signal from One Extra Beacon 17
Conclusion 18
References 19

List of Figures

Figure 1: System State Diagram 2
Figure 2: Beacon Figure 3:Handheld Back View 3
Figure 4: Handheld Sensor Placement Figure 5: Handheld Trimetric 3
Figure 6: BFSK Modulator[2] 5
Figure 7: BFSK Demodulator[2] 6
Figure 8: Bandpass Filter Schematics 8
Figure 9: Voltage Controlled Oscillator Schematics 10
Figure 10: User Interface Block diagram 11

Glossary

ADC	Analog to Digital Converter
Arduino	A system based on an 8-bit microcontroller designed for facilitating quick implementation of electronics projects
Beacon	The part of the **WhereTo** system affixed to buildings
BFSK	Binary Frequency Shift Keying
Byte	8 Bits
DAC	Digital to Analog Converter
Handheld Unit	The portable part of the **WhereTo** system carried by the user
PLL	Phase Locked Loop
SCC	Standards Council of Canada
VCO	Voltage Controlled Oscillator
UI	User interface
USB	Universal Serial Bus

Introduction

Envied Solutions' **WhereTo** system is designed to guide the visually impaired to their destination within a suitably outfitted structure. The owner of a building will have beacons installed covering the interior. These beacons will guide the user's handheld unit to their destination. The purpose of this design specification is to document the details which will be used in building the system.

Scope

This design specification contains implementation details relating to the functionality discussed in our functional specification. Specific circuits, algorithms and the supporting technology that will be used are discussed. This document focuses on the needs required for a minimal demonstration system. This includes short-cuts for rapid prototyping such as using a Raspberry Pi and Arduino combination where as a full production system would use a more specialised ARM microcontroller in place of the Raspberry Pi and specifically designed interface and IO circuits instead of the Arduino.

Audience

The intended audience of this document is the engineers at Envied Solutions involved in creating the demonstration system. They will refer to the document for specific designs of subsystems and an overall vision of how these subsystems will interact together. This document will also guide them in executing the test plan at various stages of implementation.

System Design Overview

System State Tree

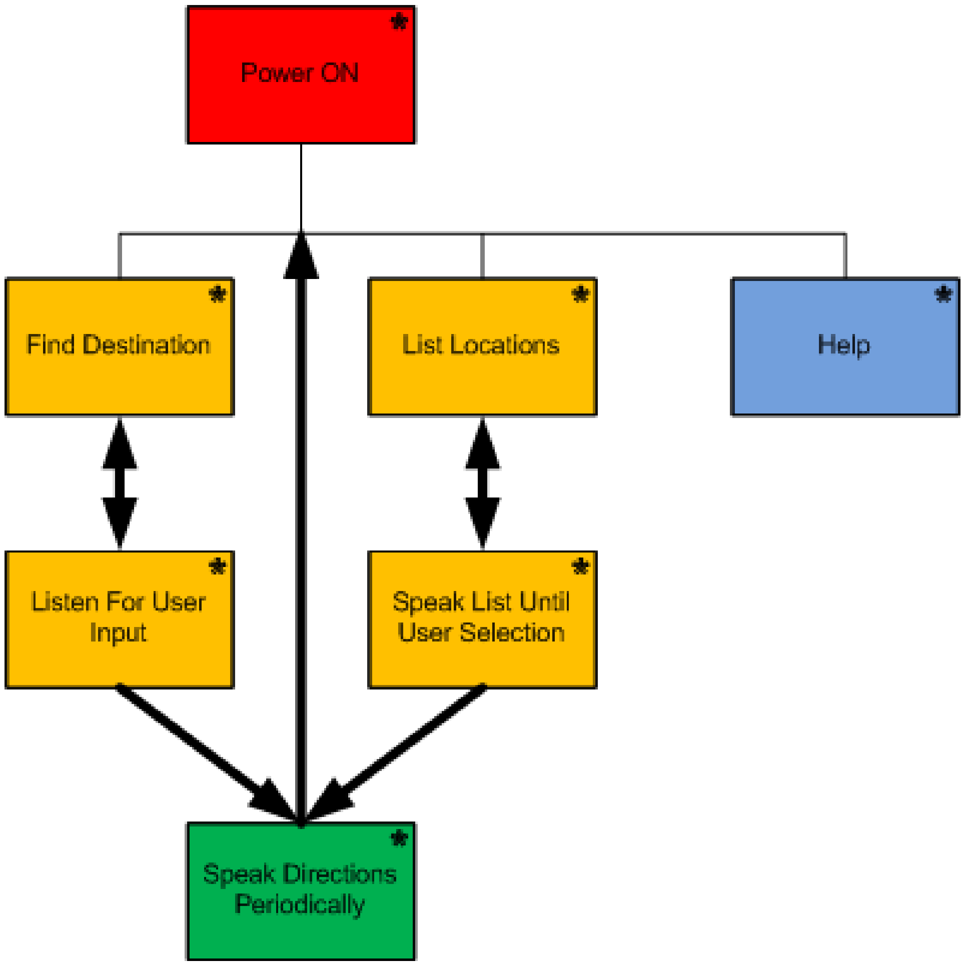

Figure 1: System State Diagram

The figure above is an overview of the major system states and its flow.

Mechanical Design

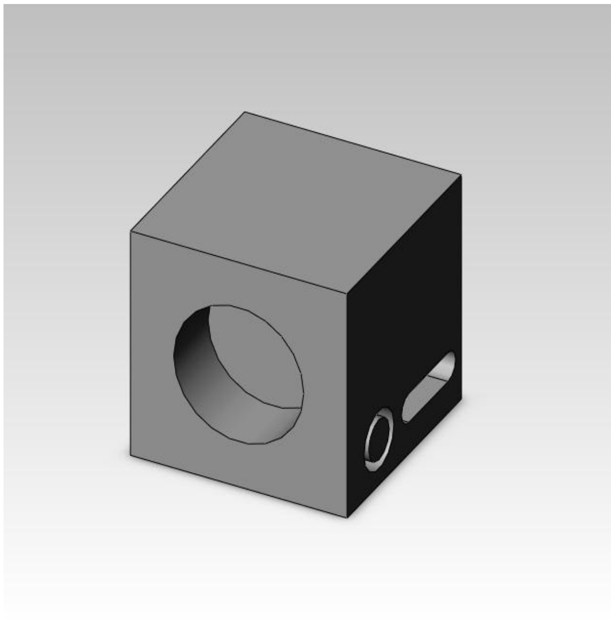

Figure 2: Beacon

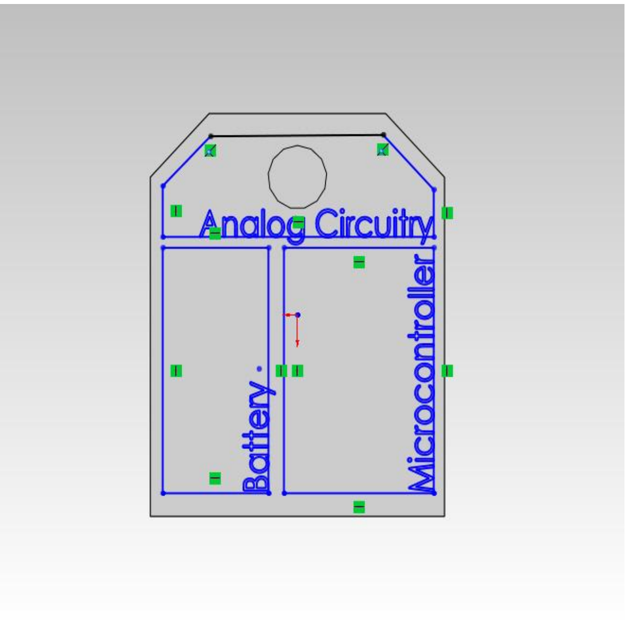

Figure 3: Handheld Back View

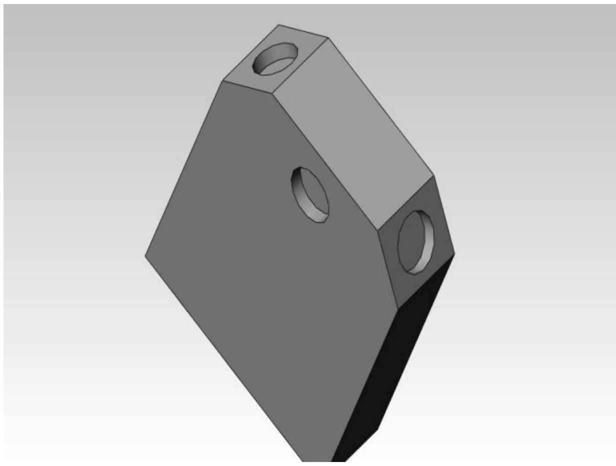

Figure 4: Handheld Sensor Placement

Figure 5: Handheld Trimetric

The above figures shows what will be look like prior to the market. From Figure 2, the beacon itself would have power cord to power up and the micro USB socket would help for configuration device. As figure 5, the handheld would have four buttons along with micro USB socket, 3.5mm headphone jack for audio input/output, and device power button.

Message Protocol

The messages from the beacons to the handheld are 10 bytes in length and formatted as follows. There are 2 bytes for a beacon identification number, 4 bytes for the current Unix Time, 3 bytes to represent the number of microseconds past the current second and lastly 1 byte for a CRC type checksum. Data within a group of related bytes is transmitted most significant bit first. (Henry, 2012)

ID_1	ID_0	S_3	S_2	S_1	S_0	μS_2	μS_1	μS_0	CRC_0
0	1	2	3	4	5	6	7	8	9

Binary Frequency Shifting Keying (BFSK)

Frequency Shifting Keying (FSK) has two sinusoid waves with different frequencies to represent the binary symbols of 0 and 1. 1 can be called marked frequency which usually higher frequency and 0 can be call space frequency which usually lower frequency. The general expression for BFSK is as followed,

$$V_{fsk}(t) = V_c\cos\{2\pi[f_c+V_m(t)*\Delta f]t\} \quad (1)$$

(Tomasi, 2003)

Where

V_c is peak analog carrier amplitude (Volt)

f_c is analog carrier centre frequency (Hertz)

$V_m$is binary input symbol voltage, usually +1 or -1

Δf is defined as the peak change of analog carrier centre frequency

When binary input is 1, V_m can be represent as +1 and formula will as followed,

$$V_{fsk}(t) = V_c\cos\{2\pi[f_c+ \Delta f]t\} \quad (2)$$

When binary input is 0, V_m has voltage of -1 and the expression will like below,

$$V_{fsk}(t) = V_c\cos\{2\pi[f_c-\Delta f]t\} \quad (3)$$

The peak change of analog carrier centre frequency will be around 1 kHz as the frequency deviation.

The calculation of the deviation as followed,

$$\Delta f = \frac{|fm - fs|}{2} \qquad (4)$$

Where

f_m is marked frequency

f_s is space frequency

f_m frequency would have 5kHz differences and f_s frequency would have 1 kHz differences for each beacon.

The minimum of bandwidth can be represent as followed,

$$B = |(f_s - f_b) - (f_m - f_b)| \qquad (5)$$

Where f_b is the input bit-rate.

The modulator as the transmitter part, which used 0 and 1 binary signal, connects with VCO (Voltage Controlled Oscillator) and then produces the analog signal output, as following figure,

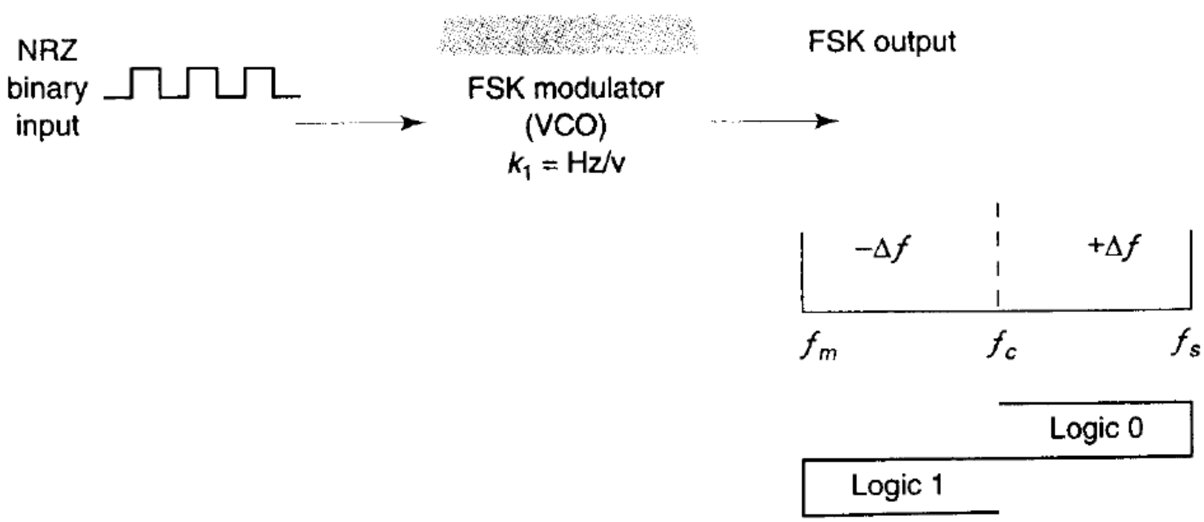

Figure 6: BFSK Modulator (Tomasi, 2003)

The receiver part is the different from the transmitter. The PLL (Phase Locked Loop) shows as following figure, “the dc error voltage at the output of the phase comparator follows the frequency shift.” (Tomasi, 2003)Since BFSK only has two input frequency, therefore, there are only two output voltage which are represented as symbol 1 and symbol 0.

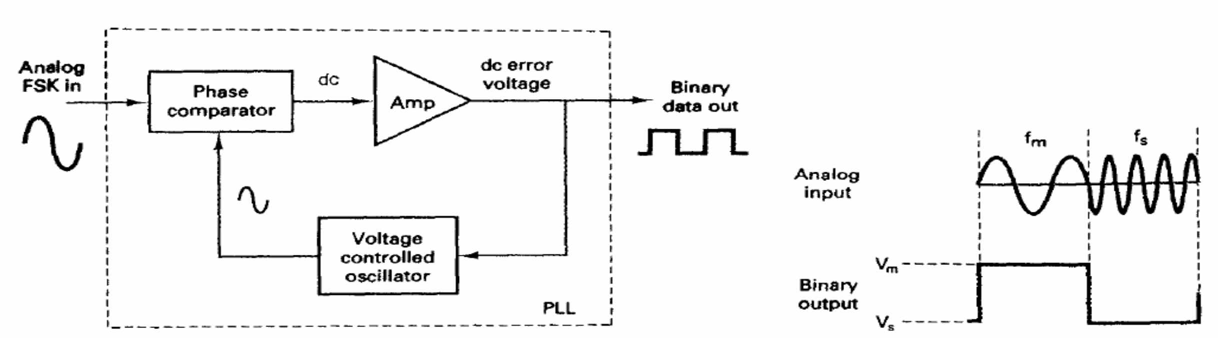

Figure 7: BFSK Demodulator (Tomasi, 2003)

Handheld Device

Positioning Strategy

Initially all we have is messages from the beacons and the time these messages arrived. The messages contain the time they were sent. We can determine the distance r, using the time the message was sent t_0, the time it was received t_a, and the speed of sound v_s.

Then $r = v_s\,(t_a - t_0)$ (6)

If have we have the distances to a minimum four beacons which have known positions we can calculate our position using trilateration. (Hereman & Murphy, 1995) (Talk:Trilateration)Conceptually this method works by determining the intersection of multiple spheres centred on the beacons whose radii are the distance from the beacon to the handheld. (Hereman & Murphy, 1995) Defining the position of the beacons $\vec{B}_i = (x_i, y_i, z_i)$ and the distance from that beacon to the handheld as r_i. Taking the information for three of the beacons $\vec{B}_1, \vec{B}_2, \vec{B}_3, r_1, r_2, r_3$ a position is calculated as follows.

A new coordinate system is established placing all three beacons in a common plane.

$$\hat{e}_x = \frac{\vec{B}_2 - \vec{B}_1}{\|\vec{B}_2 - \vec{B}_1\|},\ \hat{e}_y = \frac{(\vec{B}_3 - \vec{B}_1) - \hat{e}_x(\hat{e}_x \cdot (\vec{B}_3 - \vec{B}_3))}{\|(\vec{B}_3 - \vec{B}_1) - \hat{e}_x(\hat{e}_x \cdot (\vec{B}_3 - \vec{B}_3))\|},\ \hat{e}_z = \hat{e}_x \times \hat{e}_y \quad (7)$$

(Talk:Trilateration)

We also define the following values:

$$h = \|\vec{B}_2 - \vec{B}_1\|,\ i = \hat{e}_x \cdot (\vec{B}_3 - \vec{B}_1),\ j = \hat{e}_y \cdot (\vec{B}_3 - \vec{B}_1) \quad (8)$$

(Talk:Trilateration)

We can then find the coordinates in this frame using the following equations:

$$X = \frac{r_1^2 - r_2^2 + h^2}{2h}$$

$$Y = \frac{r_1^2 - r_3^2 + i^2 + j^2 - 2iX}{2h}$$

$$Z = \pm\sqrt{r_1^2 - X^2 - Y^2} \quad (9)$$

(Talk:Trilateration)

The position in the standard Cartesian coordinate frame can then be found by

$$\vec{B}_1 + \hat{e}_x X + \hat{e}_y Y + \hat{e}_z Z \quad (10)$$

(Talk:Trilateration)

Note the ± in the equation for Z means that there are two solutions, one of which is the correct one. Therefore we must redo the calculation substituting the information from the fourth beacon for that of one of the other three getting a second pair of solutions. The two that are identical from the two different calculations (discounting error) are the correct solution for the handhelds current location.

Note this is a basic algorithm; the paper Determination of a Position in Three Dimensions Using Trilateration and Approximate Distances (Hereman & Murphy, 1995) has more advanced algorithms that are more robust against error for a very similar type of positioning system.

Electrical

Due to the receiving signal is too small, the microcontroller may not recognize the signal. Also, there may be some unwanted noise. As the result, the bandpass filter with high gain would be the solution for the product.

As bandpass filter, the calculation is as following,

$f_{c1} = \frac{1}{2\pi R_1 C_1}$ for higher frequency band around 50 kHz (11)

$f_{c2} = \frac{1}{2\pi R_2 C_2}$ for lower frequency band 10 kHz (12)

$A = -\frac{R_2}{R_1}$ for gain 10000 (13)

After calculation, the value for bandpass is shown below

R_1	C_1	R_2	C_2
33Ω	10nF	330k Ω	47pF

Table 1: Component Value

The schematic is shown in the following figure,

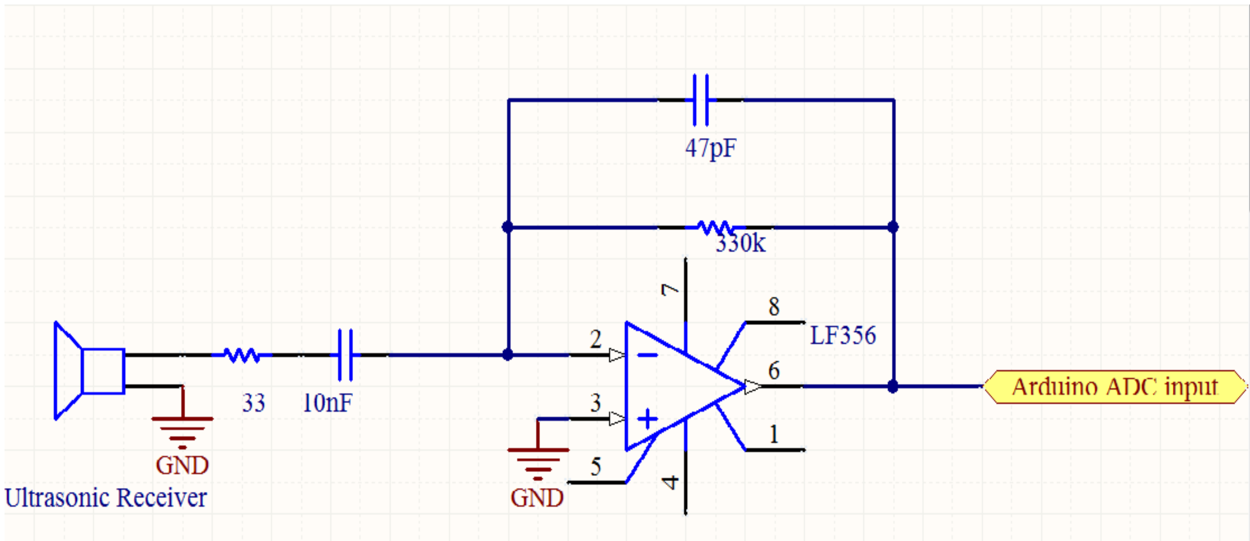

Figure 8: Bandpass Filter Schematics

Computational

The Arduino Uno's built-in ADC has a resolution of 10 bits representing values between 0 and 1023. (Arduino analogReference)The ADC's reference voltage (V_{ref}) can be set to either 5 V or 1.1 V without requiring an external reference. (AnalogRead) The smallest difference of value that can be sensed is $\frac{V_{ref}}{1023}$. Therefore when using a reference of 5V the smallest measurable change is $\frac{5\,V}{1023} = 4.8\ mV$, for a 1.1V reference this value is $\frac{1.1\,V}{1023} = 1.1\ mV$. Atmel's reference sheet for the microcontroller implies a typical error of around 3-4LSB for the ADC. (Atmel) Assuming the error is as high as 5LSB this implies that the receiver circuit should deliver waveforms with a peak to peak value of at least 10 mV for an ADC reference voltage of 1.1 V and 50 mV if the reference voltage is 5 V. The receiver design should generate signals with a peak-to-peak no higher than V_{ref} when the receiver is close to the transmitter and no lower than the minimums noted above at maximum designed range. However as will be shown below a design which only just hits the minimum will not be desirable. In either case the waveform should be centred on half the reference voltage. In other words the receiver's ground reference should be $\frac{V_{ref}}{2}$.

The ADC documentation states: "For optimum performance, the ADC clock should not exceed 200 kHz. However, frequencies up to 1 MHz do not reduce the ADC resolution significantly." (Atmel) and that "When using single-ended mode, the ADC bandwidth is limited by the ADC clock speed. Since one conversion takes 13 ADC clock cycles, a maximum ADC clock of 1 MHz means approximately 77k samples per second." (Atmel) With the overhead of storing the results in memory a practical sample rate with a 1MHZ is 60 kHz. (Faster Analog Read? - Arduino Forum) However there are examples of Arduino projects that run the ADC clock as high as 8 MHz with acceptable performance. (Faster Analog Read? - Arduino Forum) (Higher Precision scope (Mac based) - Arduino Forum) Running the ADC clock at 4 MHz we can sample three sensors with a theoretical sample rate of $\left(4\ MHz\right)\left(\frac{1\ sample}{13\ cycles}\right)\left(\frac{1}{3\ samples}\right) = 103\ KHz$. Even with overhead a sample rate of around 80 kHz should be practicle as long as the input wave form is big enough. By the Nyquist sampling theorem this means a signal up to 40 kHz should be recordable.

Higher level processing such as position calculating, path calculation, voice recognition, sound output, and collision avoidance are performed on a sufficiently powerful microcontroller that can run a simple Linux platform to leverage freely available software libraries. As proof of concept, the Raspberry Pi is used as it is low in cost and simple to interface.

Position calculation is performed by a process of trilateration implemented and path calculation determined with the A* (A star) algorithm. Both are implemented in C/C++ to minimize system resource consumption which will be heavily used by the voice recognition suite.

Voice recognition is accomplished through the use of three software packages: Voxforge, HTK, and Julius.

Voxforge enables the creation an acoustic model that purposefully suits the needs of this project. Acoustic models are basically a statistical representation of an association of words with sounds.

HTK is the Hidden Markov Model Toolkit which uses hidden Markov models to properly determine the phonemes that a recorded voice sample contains. Phonemes are the elementary sounds of which speech in all languages are composed of.

Julius performs the vocabulary lookup against a dictionary of recognized words to efficiently determine entire phrases with correct grammar.

The combination of the aforementioned three software packages allows the specific definition of a set of words to listen for and recognize when they are close enough of a match. This result is then used by higher logic to determine how to react to and prompt the user.

Signal Beacons

Electrical

The beacon will send the serial numbers like 0100 to the handheld so that the handheld can notice which beacon it is. The signal from microcontroller is sending constant pulse. However, the ultrasonic transducer works better with analog signal such as sinusoid wave or triangular wave. The centre frequency for this project would achieve 30 kHz with triangular wave.

The centre frequency can be calculated as the following formula,

$$f_0 = \frac{V_I}{8RC(V_{TH} - V_{TL})} \qquad (13)$$

$$V_{OH} = \frac{V_{cc}}{(1+\frac{R_2}{R_1})} \qquad (14)$$

Since V_{TL} = 0 V and entire device with 5V, as the result, we set V_{TH} = 5 V and the R= 6.7kΩ C=1nF to achieve 30 kHz signal. The schematic is showed below,

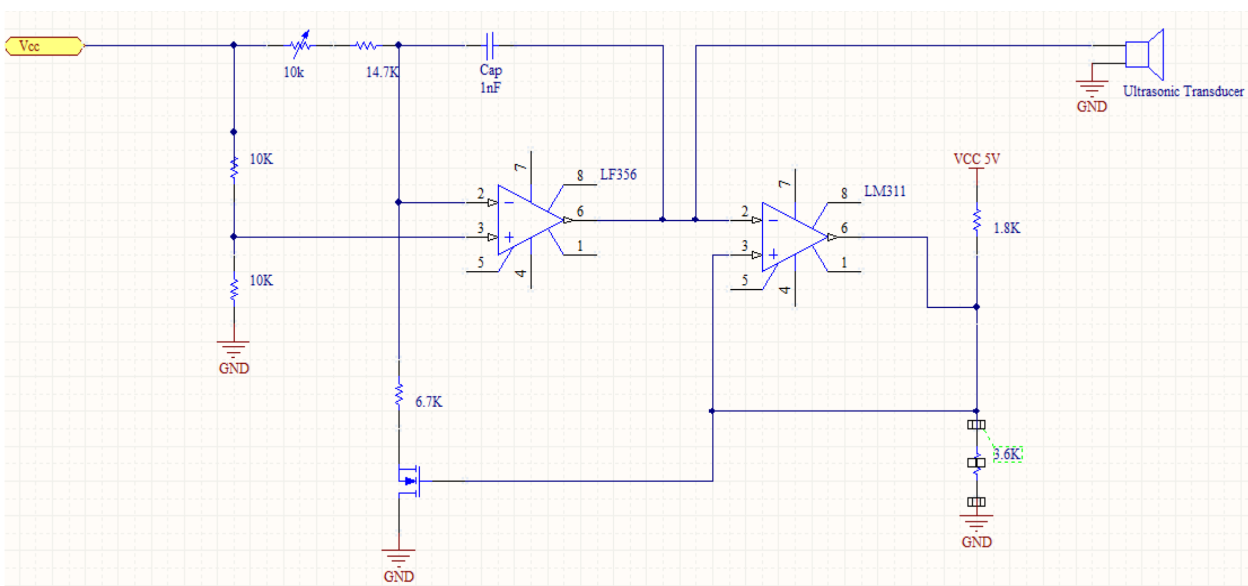

Figure 9: Voltage Controlled Oscillator Schematics

Computational

The Raspberry Pi will send the serial number message to the DAC and then will go through the VCO so that it can be send out as analog signal.

User Interface

Due to the primary market, the user interface will have to be usable with no visual output or feedback. In order to meet this goal, the primary user input will be through spoken word into a microphone and primary device output will be spoken audio cues delivered to a pair of headphones. The device can additionally be operated by its hardware buttons for powering on and off, previous, next, confirm/repeat, and cancel selection.

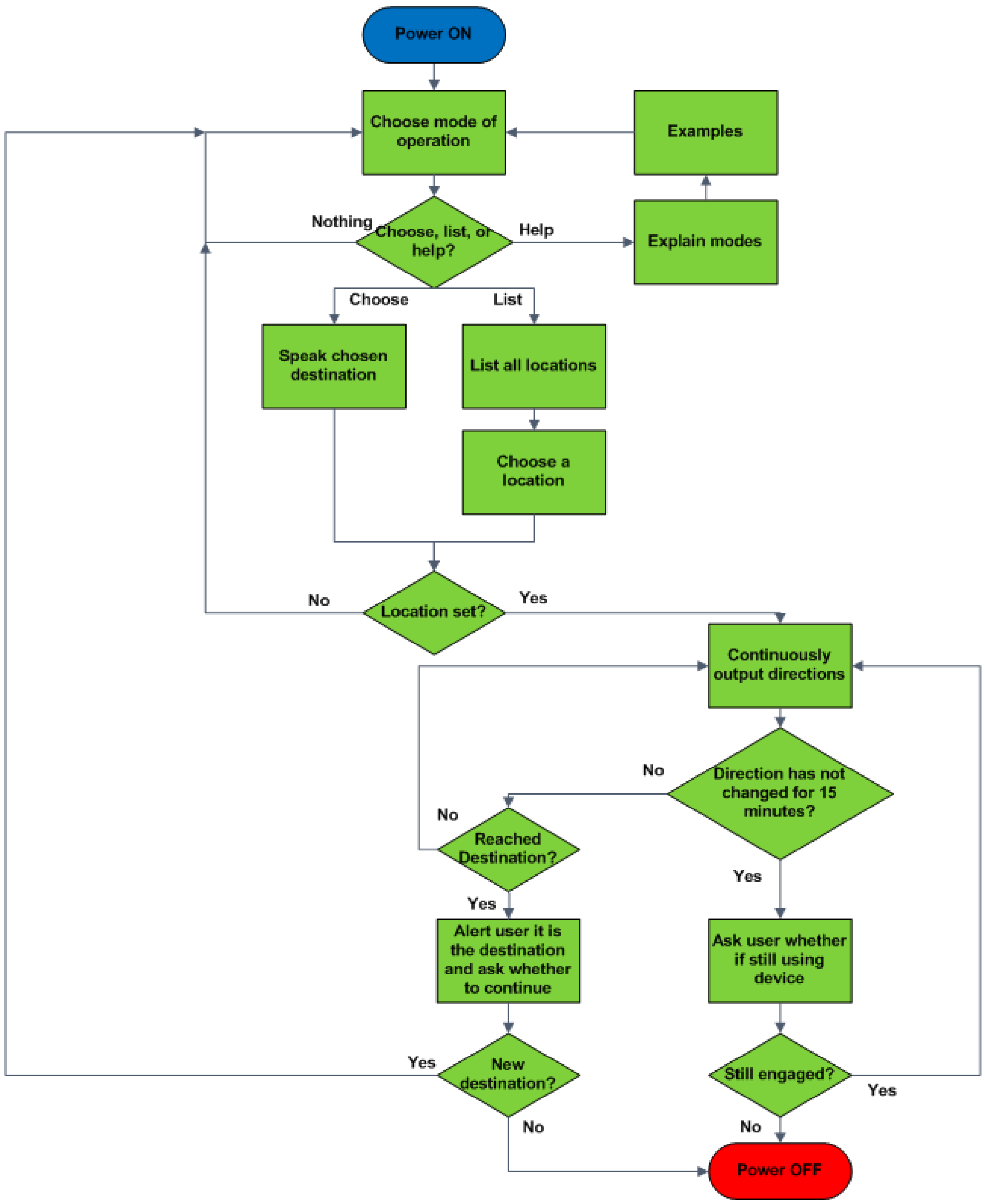

Figure 10: User Interface Block diagram

Handheld Unit Software Interface

Audio Output

The handheld device outputs natural speech to prompt the user for input and/or selection.

Audio Input

Immediately after powering on, the user may select to speak a location name, listen to a list of all locations, and request for help.

To choose to speak a location name, the user can say “Go”, “Destination”, “Find”, “Look”, or “Set” to enter that system state.

To choose to listen to a list of locations, the user can say “All”, “List”, “Location”, or “Locate” to enter that system state.

To choose to listen to device usage help and examples, the user can say “Help” to enter that state.

While in the location selection states, the device listens for room names and numbers and outputs the closest match for the user to confirm before continuing with navigation.

In all states, the user can say “Again” or “Repeat” to repeat the last audio output.

In all states except while actively navigating, if there is no user input for 5 seconds after the last system output, the device will repeat itself up to 2 times – after which the device will revert back to its initial powered on state.

While navigating, the device will repeat a single direction if it has not changed for up to 15 minutes before asking the user if they are still engaged with the device. If there is no input within 10 seconds of the prompt, the device will then suspend/power off.

Handheld Unit Hardware Interface

Power Button

The power button is positioned at the base of the handheld unit and is flush with the surrounding case and the user has to press into it to ensure that there is no accidental turning on and off of the device.

Previous Button

The previous button is the leftmost of a set of the main operational hardware keys. It is used to navigate backwards during the location selection process when the device outputs its full list of destinations one by one.

Next Button

The next button is paired with the previous button to allow bi-directional navigation through selections.

Confirm/Repeat Button

The Confirm/Repeat button locks into user selections as well as triggers a repeat of the last spoken direction while actively navigating.

Cancel Button

The cancel button moves the user one level up the system state tree.

System Test Plan

Unit Testing

Beacon Transmitter

The verification of proper beacon signals will be performed in stages.

The first stage occurs during assembly of components by connecting a series of set input voltages and taking measurements through an oscilloscope to ensure that the circuitry performs as it is designed for.

First, it is verified that it is generating the proper signal carrier wave with a neutral input to the VCO. Next is to ensure that a varying input to VCO generates an appropriately shifted frequency in the output. Throughout this stage it should be verified that the frequency is close to the desired frequency and is stable.

In the second stage, the test is to verify that the microcontroller generates the proper digital signal by replacing the set input voltages with the microcontroller. The microcontroller should be commended to send a signal (at extremely slow bit rate) and the changes in frequency should be observable on the oscilloscope.

In the third stage, the test is to verify that the ultrasonic signal carries the correct encoded analog signal shape within the desired frequency range and is stable. This is done so by adding in the ultrasonic transducer and measuring the signal through a separate ultrasonic sensor at close range.

In the final stage, the test is to verify that the encoded ultrasonic signal maintains its integrity over the desired range. This is performed by displacing the transmitting and sensing components by various distances.

Position Calculator

The verification of proper position detection consists of three parts.

The first part is to ensure that the signal integrity is maintained from beacon to handheld (essentially, the last stage of beacon transmitter testing but performing the detection through the handheld's hardware.

Furthermore, the decoded signal needs to yield the original signal encoding. To accomplish this, the decoded signal is compared against list of accepted codes (registered beacons in the room map). Invalid beacon signals should be rejected and excluded from distance calculations.

The second part to position calculation testing is to perform time-of-flight calculation from synchronized clocks for distance from each recognized beacon. To verify, place the beacon and handheld at a known distance apart and verify the calculated distance to be within tolerance for an overall +/- 5% in location.

The final step to position calculation testing is to verify that the position determined by trilateration is within the required +/- 5%. To verify, compare the computed location with the actual location.

Path Calculator

The verification of proper path determination involves finding a reasonable path to the destination that avoids known obstacles (encoded into the room map). This is partitioned into two stages.

For the first stage, position the handheld device directly in front of an obstacle blocking a straight path to the destination. Test to ensure that the device produces a path that navigates around the obstacle and back on track to the destination.

The second stage is to position the handheld and destination such that it requires turning a blind corner (the handheld does not have a direct path to the destination) and pass by intermediate destinations (the corner). The verification is in ensuring that the directions accurately point towards the intermediate destination before pointing towards the final destination.

Collision Calculator

Collision prediction and detection leverages the pre-programmed room-map and the fact that ultrasonic transducer pairs can also be used as proximity sensors. To verify, gradually move the handheld device towards walls as well as unlabeled obstacles to trigger a notification that the user is too close to an obstacle.

Voice Recognition/UI

The voice recognition and control software is to be tested against a library of recognized words (the commands for manually dictating a destination, listing the available locations, requesting a repeat of the last issued direction, numbers, and letters) for accuracy and real world error tolerance with noise, different accents and intonation.

The second part to voice recognition and user interfacing testing is to ensure that the software flow follows the design and that the user does not become stuck in any of the software states.

Sound Output

Sound output testing is to be tested for loudness and clarity in the spoken prompts to the user.

Normal Use Testing

Case 1 - List Destinations by Voice

Setup procedure:
Be in the centre of a building with sufficient beacon coverage.

Steps to follow:
1) Power ON handheld
2) Speak "list"
3) Choose a destination by voice and confirm
4) Follow spoken directions

Expected results:
1) Handheld powered ON
2) Handheld lists all available locations one by one
3) Handheld prompts for confirmation and proceeds
4) User reaches destination

Case 2 - List Destination by Button

Setup procedure:
Be in the centre of a building with sufficient beacon coverage.

Steps to follow:
1) Power ON
2) Navigate to and enter list locations mode with hardware buttons
3) Choose a destination by hardware buttons and confirm
4) Follow spoken directions

Expected results:
1) Handheld powered ON
2) Handheld lists all available locations one by one
3) Handheld prompts for confirmation and proceeds
4) User reaches destination

Case 3 - Manually Input Destination Accomplished Purely by Voice

Setup procedure:
Be in the centre of a building with sufficient beacon coverage.

Steps to follow:
1) Power ON
2) Speak "find"
3) Speak a valid location and confirm
4) Follow spoken directions

Expected results:

1) Handheld powered ON
2) Handheld prompts for a user-defined destination
3) Handheld prompts for confirmation and proceeds
4) User reaches destination

Case 4 - Manually Input Destination Accomplished with Hardware Button Navigation

Setup procedure:
Be in the centre of a building with sufficient beacon coverage.

Steps to follow:
1) Power ON
2) Navigate to and enter manual-input destination mode with hardware buttons
3) Speak a valid location and confirm
4) Follow spoken directions

Expected results:
1) Handheld powered ON
2) Handheld prompts for a user-defined destination
3) Handheld prompts for confirmation and proceeds
4) User reaches destination

Case 5 - Navigate to Complex Destination (Requires Turning Corner)

Setup procedure:
Be in the centre of a building with sufficient beacon coverage.

Steps to follow:
1) Power ON
2) Navigate to and enter list locations mode with hardware buttons
3) Choose a complex destination (one requiring a corner turn) by hardware buttons and confirm
4) Follow spoken directions

Expected results:
1) Handheld powered ON
2) Handheld lists all available locations one by one
3) Handheld prompts for confirmation and proceeds
4) User reaches destination

Extreme Case Testing

Case 1 - Outside of Building but Still Receiving Signals

Setup procedure:
Be outside of a building and still have sufficient beacon coverage.

Steps to follow:
1) Power ON

2) Navigate to and enter list locations mode with hardware buttons
3) Choose any destination
4) Listen for voice prompts (if any)

Expected results:
1) Handheld powered ON
2) Handheld lists all available locations one by one
3) Handheld prompts for confirmation and proceeds
4) Handheld determines that the user is off the room-map and alerts that the user is out of range

Case 2 - Purposely Broadcast Improper Signal from One Beacon to Cause Them to Be Insufficient

Setup procedure:
Be in the centre of a building with a single bad beacon that otherwise has sufficient coverage.

Steps to follow:
1) Power ON
2) Navigate to and enter list locations mode with hardware buttons
3) Choose any destination
4) Listen for voice prompts (if any)

Expected results:
1) Handheld powered ON
2) Handheld lists all available locations one by one
3) Handheld prompts for confirmation and proceeds
4) Handheld determines that there is insufficient coverage to reliably compute location and directions and alerts user

Case 3 - Purposely Broadcast an Extra Unencoded Signal from One Extra Beacon

Setup procedure:
Be in the centre of a building with sufficient coverage + an extra beacon with unencoded signal.

Steps to follow:
1) Power ON
2) Navigate to and enter list locations mode with hardware buttons
3) Choose any destination
4) Listen for and follow voice prompts (if any)

Expected results:
1) Handheld powered ON
2) Handheld lists all available locations one by one
3) Handheld prompts for confirmation and proceeds
4) Handheld determines that the extra beacon is useless and proceeds to provide correct directions

Conclusion

This design specification will enable the members of Envied Solutions to create a system which conforms to the functional specification which has already been produced. The detailed calculations and plans in this document will guide our implementation of the **WhereTo** system. The test plan in this document will let us know if we are meeting or functional goals and alert us to issues during development as they arise. This will allow us to deliver the demonstration prototype of the **WhereTo** on schedule.

References

(n.d.). Retrieved from Atmel: http://www.atmel.com/Images/doc2559.pdf

AnalogRead. (n.d.). Retrieved from Arduino: http://arduino.cc/en/Reference/AnalogRead

Arduino analogReference. (n.d.). Retrieved from Arduino: http://arduino.cc/en/Reference/AnalogReference

Faster Analog Read? - Arduino Forum. (n.d.). Retrieved from Arduino: http://arduino.cc/forum/index.php?topic=6549

Henry, W. S. (2012). *Hacker's Delight.* Boston: Addison-Wesley Professional.

Hereman, W., & Murphy, W. S. (1995). Determination of a Position in Three Dimensions Using Trilateration and Approximate Distances. *Decision Sciences* .

Higher Precision scope (Mac based) - Arduino Forum. (n.d.). Retrieved from Arduino: http://arduino.cc/forum/index.php/topic,8795.0.html

Talk:Trilateration. (n.d.). Retrieved from Wikipedia: http://en.wikipedia.org/wiki/Talk:Trilateration

Tomasi, W. (2003). *Advanced Electronic Communications Systems.* Upper Saddle River: Prentice Hall.