Simon Fraser University
8888 University Drive
Burnaby, BC V5A 1S6

November 20th, 2011

Dr. Andrew Rawicz
School of Engineering Science
Simon Fraser University
Burnaby BC
V5A 1S6

Re: Design Specifications for Proximity Detector for the Visually-Impaired

Dear Dr. Rawicz,

Please find enclosed the design specifications for PROXIMIview Technologies' Wearable Proximity Detector for the visually-impaired. The purpose of the device is to aid the user in avoiding obstacles via tactile and auditory feedback by utilizing an array of ultrasonic sensors mounted on sunglasses as well as on the waist-level enclosure.

The comprehensive set of design specifications in the enclosed document will be used extensively by our team at PROXIMIview Technologies for the design of the proof-of-concept device, as well as for integrating and testing purposes. It will also help design, development, and integration engineers gain a working understanding about each component used in the product, and how it contributes to the final operation of the device.

PROXIMIview Technologies is comprised of four talented, dynamic and motivated engineering students namely Renuka Rani, Gary Brykov, Sajith Kulasekare and Marish Lalwani. If you have any questions or concerns, please feel to contact me by phone (778-321-3761) or by e-mail (PROXIMIview@gmail.com).

Thank you very much for your consideration.

Yours sincerely,

Renuka Rani
Chief Executive Officer (CEO)
PROXIMIview Technologies Ltd.

Enclosed: Design Specifications for Proximity Detector for the Visually-Impaired

PROXIMIview Technologies

Design Specifications for a Wearable Proximity Detector to Aid the Visually-Impaired

Contact Person: Gary Brykov
gba2@sfu.ca

Submitted To: Andrew Rawicz – ENSC 440
Mike Sjoerdsma – ENSC 305
School of Engineering Science
Simon Fraser University

Issue Date: November 20th, 2011

Simon Fraser University
8888 University Drive
Burnaby, BC V5A 1S6

PROXIMIview Team

Renuka Rani	CEO
Sajith Kulasekare	CTO
Gary Brykov	CFO
Marish Lalwani	COO

Simon Fraser University
8888 University Drive
Burnaby, BC V5A 1S6

EXECUTIVE SUMMARY

The Wearable Proximity Detector is a product proposed by PROXIMIview Technologies. It is essentially a proximity detector for nearby objects which guides the visually impaired people in a hallway-like area. Wearable Proximity Detector will act as guide for the visually impaired by offering the user audio/physical feedback and warning the user of the proximity of the object relative to him/her. The product will also have the capability to alert the user by a vibrating module whereby the vibrations increase in intensity as the user moves closer to the object. The project will also take into account various sanity tests to ensure the product is up and running without any flaws.

The design specifications for the Wearable Proximity Detector contain a detailed description of the design of our proof-of-concept model. Our goal for this document will be to provide an in-depth analysis of the design of a working prototype for the Wearable Proximity Detector, which is being developed by PROXIMIview Technologies. This document will also present various ideas such as future works for the improvement of the working design prototype. While these ideas will not be implemented in the proof-of-concept model, they might be fulfilled in the future iterations.

The following are the key components developed by PROXIMIview Technologies:

- Sunglasses/Sensor Unit
  This unit consists of sunglasses with three sensors mounted on it. The first sensor is placed in the front of the sunglasses and will acts as our main sensors which drive the other sensors. The other two sensors are mounted to the left and right of the sunglasses and will look at the sides. The signals from the three sensors are sent to the control unit for appropriate action to be taken.
- Control Unit
  This unit consists of a microcontroller with a Voice Shield mounted on its top and a battery holder. Our control unit will be placed inside a custom made enclosure which will be easy to carry around. The function of this unit is to process information sent from the Sunglasses/Sensor Unit and guarantee an auditory and physical feedback to the user. The use of the fourth sensor mounted onto the enclosure would provide us the ability to detect objects at the waist level. The Voice Shield provides various voices that are essentially the audio messages sent to the user.

Hence, this document provides a detailed description of the use of various resources and their respective functional and design specifications. The Quality Assurance of our product is determined by a system test plan that has been provided in the document. This will provide an idea about the thorough testing procedures of all major components of our product. The document also includes a complete plan for testing the product as a whole to ensure that the final product is fully functional and complete by December 15, 2011.

TABLE OF CONTENTS

Simon Fraser University
8888 University Drive
Burnaby, BC V5A 1S6

LIST OF FIGURES

LIST OF TABLES

Simon Fraser University
8888 University Drive
Burnaby, BC V5A 1S6

GLOSSARY

ADC – Analog to Digital Convertor

AVR – Advanced Virtual RISC

AWG – American Wire Gauge

CMOS – Complimentary metal-oxide-semiconductor

CNIB – Canadian National Institute for the Blind

EOM – End of Message

GND – Electrical Ground

ISD – Information Storage Device

KHz – Kilohertz. A unit of measurement of frequency

mAh – Milliampere-hour. A unit of electric charge

MCU - Microcontroller

NiMH – Nickel Metal Hydride

PCB – Printed Circuit Board

PWM – Pulse Width Modulation

RISC – Reduced Instruction Set Computing

SIP - Single-In-Line Package. An IC package or multi-component sub-assembly that has connections or leads in a single row on one side

SPI – Serial Peripheral Interface

Vcc – Voltage collector collector. A positive supply voltage

VDC – Volts Direct Current

VS – VoiceShield™ manufactured by Spikenzie Labs

Simon Fraser University
8888 University Drive
Burnaby, BC V5A 1S6

1.0 INTRODUCTION

The Wearable Proximity Detector by PROXIMIview Technologies is an assistive device for the visually-impaired to aid the user in avoiding objects/obstacles. The device will utilize an array of ultrasonic sensors which will be mounted on sunglasses to inform the user when an object is a certain distance away. A sensor mounted on the control/feedback enclosure is also included to alert the user about waist-level objects. The feedback unit will feature both tactile and auditory elements; a vibrating unit will be used to alert the user of the relative proximity of the object – the closer the user gets to the object, the more frequent the vibrations. Auditory feedback will be composed of messages informing the user about the distance of the object, as well as some directional messages. The design specifications for the Wearable Proximity Detector as proposed by PROXIMIview Technologies are outlined in the following document.

1.1 Scope

This document describes comprehensive design specifications chosen for PROXIMIview's proof-of-concept Wearable Proximity Detector. It deals with satisfying the specifications outlined in the *Functional Specifications* document [1], and also thoroughly discusses the test plan to test the various components of the device. This document applies exclusively to the proof-of-concept model, not the production model – thus, requirements of priority I and only some priority II and III are adhered to in this document.

1.2 Intended Audience

This document contains a detailed description of the design of the proof-of-concept device, and will be used by the development and integrations team at PROXIMIview as a strict guideline to meet all requirements. It will also help design, development, and integration engineers gain a working understanding about each component used in the product, and how it contributes to the final operation of the device. Test engineers will be able to use this information and the detailed test plans to ensure seamless operation of the product.

2.0 System Overview

The PROXIMIview Wearable Proximity Detector utilizes three PING))) ultrasonic sensors mounted on a pair of sunglasses to alert the user of any obstacles at eye-level, as well as another PING))) sensor mounted on the control/feedback enclosure to provide waist-level alerts. The first sensor is mounted looking directly forward from the user's perspective. The second sensor is mounted orthogonally to the first, facing left from the user's perspective. The third sensor is mounted orthogonally to the first, facing right from the user's perspective. Figure 2.1 illustrates the sensor arrangement on the sunglasses.

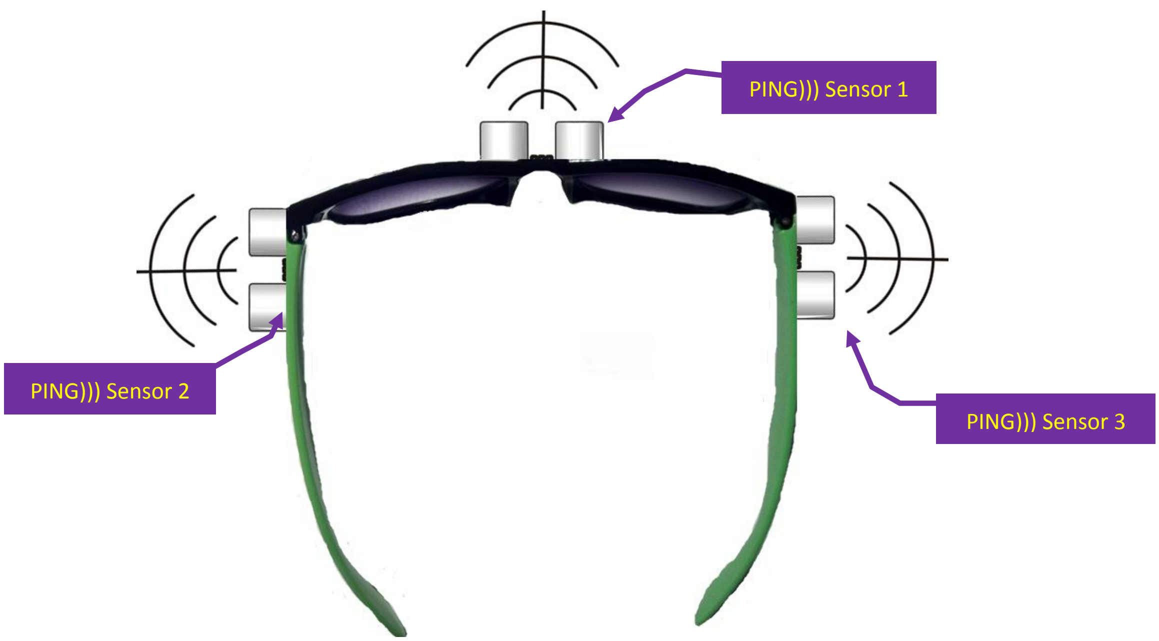


Figure 2.1: Ultrasonic Sensor Arrangement

The sensor wiring will be integrated into the sunglass frame and will run down into the control module. The control module contains the Arduino Uno Microcontroller unit, the Arduino VoiceShield, vibrator, and battery unit. The enclosure is to have a simple ON/OFF switch and a 3.5 mm headphone jack. To receive auditory feedback, a headphone will also be included with the device.

The entire system will be powered by a rechargeable custom 10.8V and 2000mAh battery pack from Polar Batteries©. This battery pack will provide approximately 6 hours of continuous operation. In order to not leave the user with dying/dead batteries, a low battery indicator will also be included, again, providing vibratory and audio feedback of the battery's state.

The PROXIMIview system overview is provided in Figure 2.2.

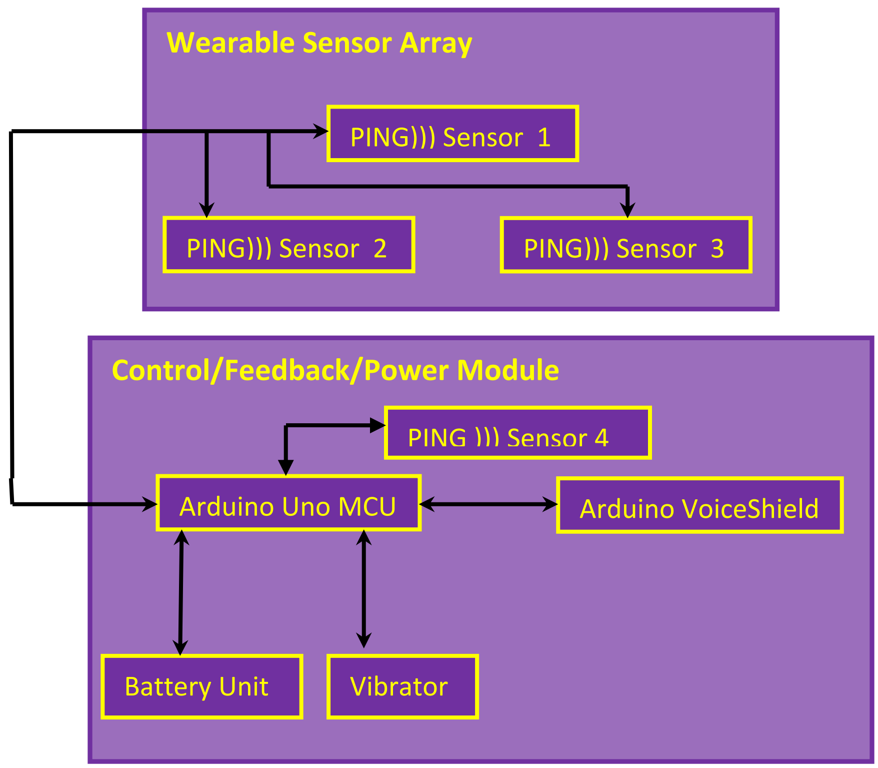


Figure 2.2: System Block Diagram

3.0 PRODUCT DESIGN

3.1 Low-Level System Design

Figure 3.1 provides a clear view of all the components involved in the low level design. The VoiceShield, which is positioned on top of the Arduino Uno board, will be using digital pins 2-5. Four PING))) sensors involved in the overall design will be using Digital Pins 6, 7, 8 and 10. The vibrator module will be using the digital pin 9, which supports pulse width modulation (PWM).

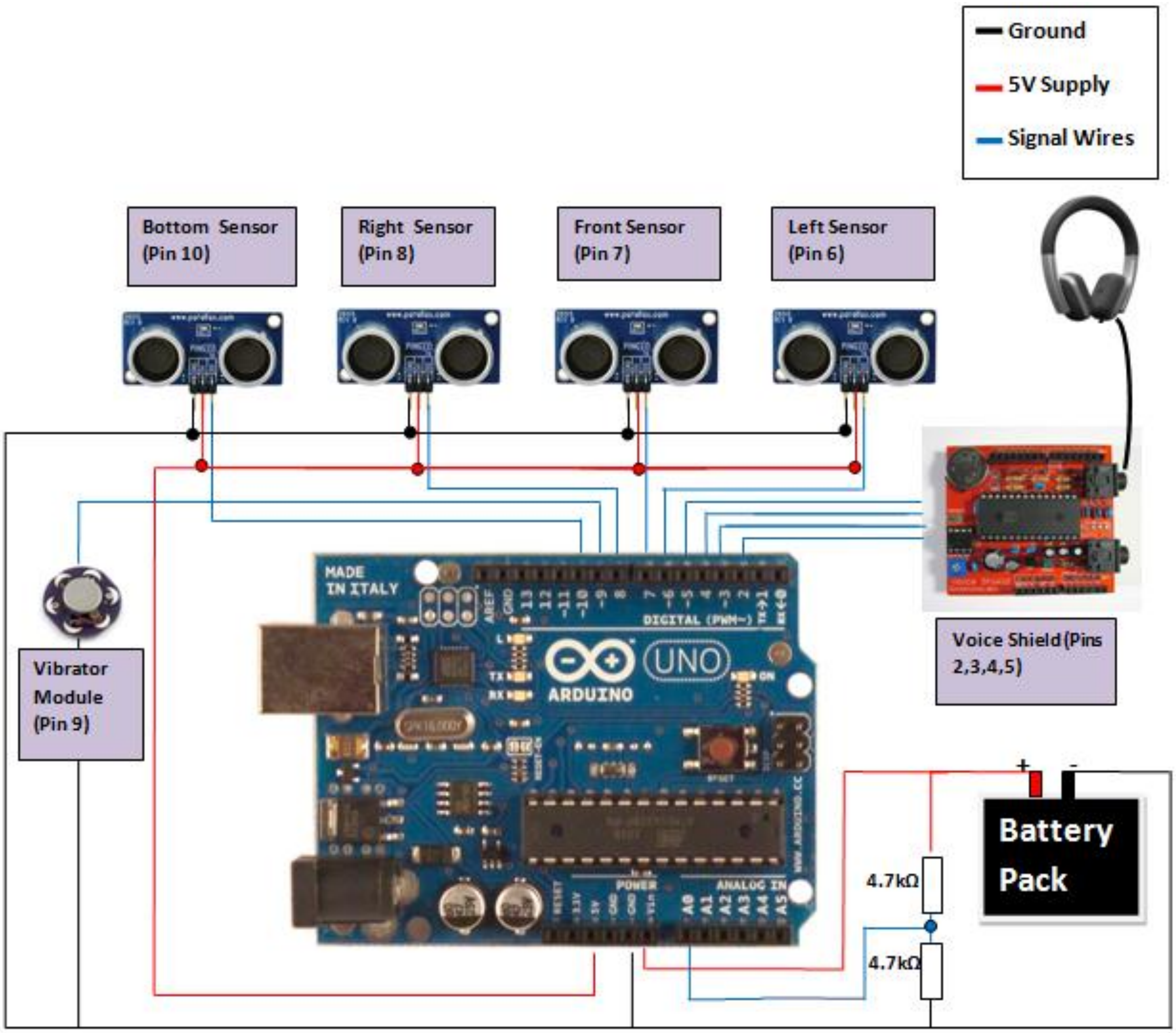


Figure 3.1: System Low-Level Design

Voltage level will be read through Analog Pin 0 of the Arduino Uno to provide auditory feedback when the battery level goes down. All components will be connected to ground using GND pin of the Arduino Uno.

Power will be supplied through the Vin Pin using a 10.8V battery pack. Audio feedback will be obtained using a pair of headphones that will be connected to the audio out of the VoiceShield.

3.2 Wearable Sensor Unit

The wearable sensor unit is the first of two main components of the PROXIMIview device. This section breaks down the mechanical design choices of the device as well as the placement of the ultrasonic sensors on the sunglasses.

3.2.1 Mechanical Design

The design of the sunglass sensor array takes several important factors into consideration. The first of these is the weight of the device. An objective was to minimize the weight of the sunglass unit in order to allow the device to be comfortable and non-restricting to the user's range of motion. Each ultrasonic sensor is fitted into the frame of the sunglasses. This allows the sunglasses to be folded up and stored just as a regular pair of sunglasses would be stored. The wearable sensor sunglasses unit consists of three parts:

1) The sunglasses themselves. These were chosen to be a unisex design, with a discrete black colour and a larger lens size. The large lens sizes allows for easier concealment of the sensors and a more universal fit for the end user.

2) The PING))) ultrasonic sensors. This unit consists of an emitter/receiver pair mounted on a small PCB. The package is a 3-pin SIP, with 0.1" spacing (ground, power, signal). This sensor is relatively small and lightweight, with dimensions of 22 mm H x 46 mm W x 16 mm D and weight of 9 grams. The wearable sunglasses unit utilizes three of these sensors, adding a total weight of 27 grams to regular sunglasses. The sensors run off a supply voltage of +5 VDC and a supply current of 30 mA. The effective range of these sensors is 2 cm to 3 m, which is ideal for the application of a wearable proximity detector. The sensor dimensions are displayed below.

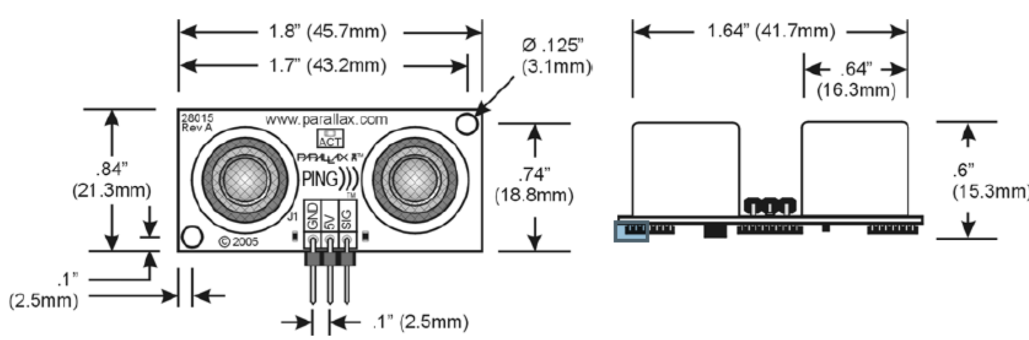


Figure 3.2: PING))) Sensor Dimensions

3) The wiring used to interface the sensors to the Arduino microcontroller unit. The choice of wiring was decided based on the need for flexible, lightweight, and shielded wiring. The wiring chosen was 12 AWG 5/20 copper wire. This wire consists of five 20-gauge wires twisted together into a 12-gauge wire, surrounded by shielding. A 5-conductor wire was chosen because 5 contacts are needed: 3 for each sensor signal pin, one for 5 VDC power, and one for GND.

3.2.1.1 Sensor Placement

The PROXIMIview Wearable Proximity Detector utilizes three PING))) ultrasonic sensors mounted on a pair of unisex sunglasses. The first sensor is mounted looking directly forward from the user's perspective. The second sensor is mounted orthogonally to the first, facing left from the user's perspective. The third sensor is mounted orthogonally to the first, facing right from the user's perspective. The sensors are mounted in a manner that allows unobtrusive wearing of the device without restricting the user's range of head motion. The finished sunglasses unit is accurately illustrated in Figure 3.3.

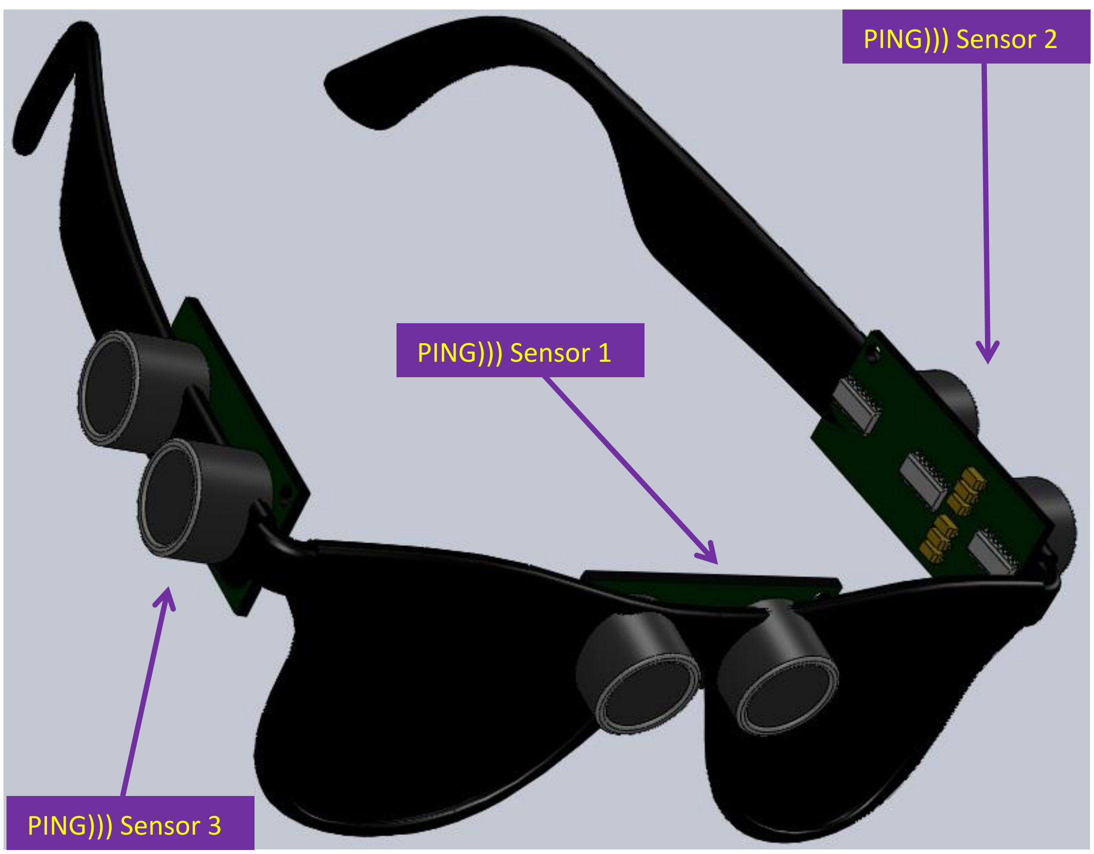


Figure 3.3: Finished Wearable Sensor Unit

For weatherproofing, the exposed PCB on the sensors will be covered in Kapton© tape. Kapton© Tapes are made from Kapton© polyimide film with silicon adhesive that is compatible with a wide temperature range from -269°C to as high as 400°C. This tape will protect the sensor electronics for moisture and scratches.

An overhead view of the sunglasses sensor array as well as the intended position on a user's head is presented in the figure below.

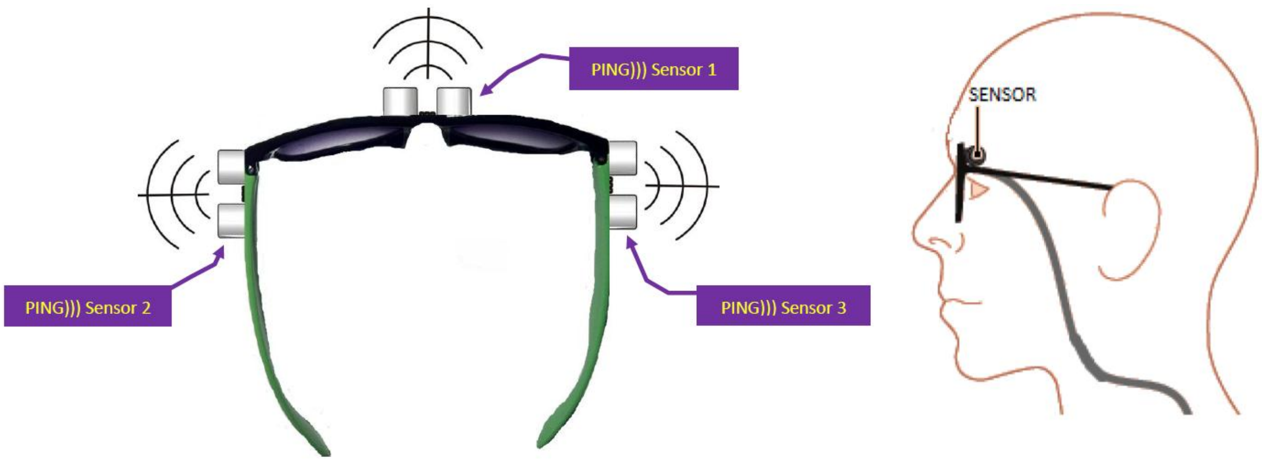


Figure 3.4: Ultrasonic Sensor Arrangement and Intended Wear

The 12 AWG 5/20 wire will be secured to the sunglasses frame and will run down the frame, behind the user's ear, and into the controller module worn around the user's waist.

3.2.2 Electrical Design

Figure 3.5 shows the wiring configuration of the three sensors on the glasses that will be connected to the Arduino Uno (enclosure). 5V supply and the ground conductors will be common to all three sensors, while separate signal conductors will be obtained from all three sensors. These five conductors will be twisted into a 12 AWG 5/20 wire and fed through the right side of the glasses, connecting to the enclosure held at waist level. The fourth bottom sensor will be embedded into the enclosure, which houses the Arduino Uno board and is to be worn at waist level.

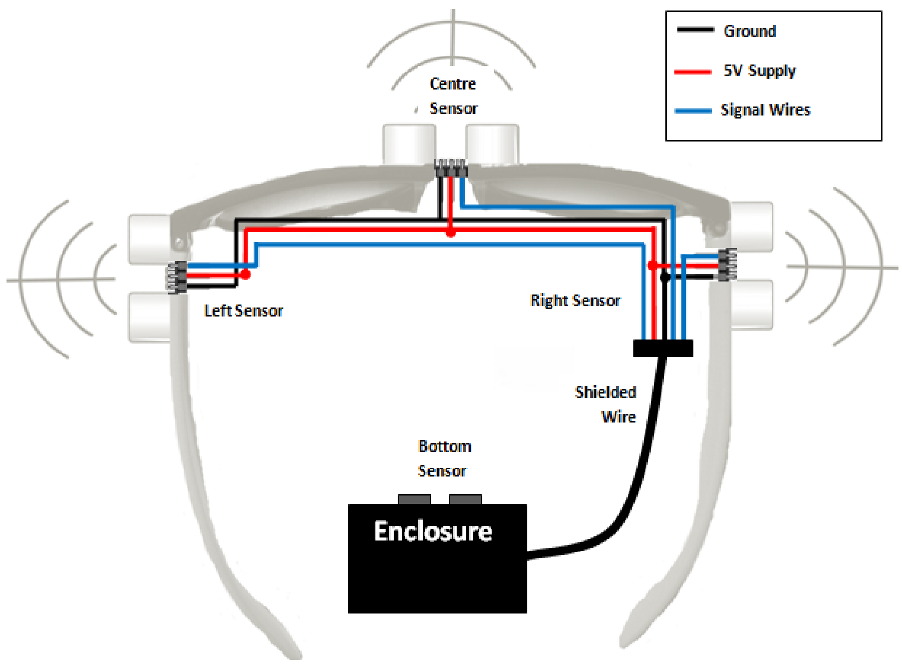


Figure 3.5: Wiring diagram of the wearable unit

3.2.3 Safety Considerations

Safety precautions need to be employed since the wearable unit will be constantly making contact with the visually impaired individual's body. Some considerations are outlined below:

- High Insulation: Thorough insulation will be provided to avoid any damages caused by electricity and overheating.
- Efficient placement of components: Components will be placed to provide higher safety and comfort to the user. For example the battery compartment will be placed on top of the Arduino Uno to avoid damages caused by heating that might affect the user.

- Sanity checks and battery level indicator: Auditory feedback will be provided in case the battery level is low or in case any of the sensors fail.
- Ground: All component involved in the design will be grounded to the GND pin of the Arduino Uno board.
- Power Supply: The rechargeable battery can be charged using a wall charger that will connect directly to handheld enclosure. User will not be required to take the battery out, which will provide security and safety to the device and the user.

3.2.4 Noise Considerations

The following have been taken into consideration to reduce the effects of noise on the overall system:

- Efficient usage of wires: Wires will be routed efficiently to avoid noise and provide balance in the electric circuit.
- Use of shielded wires: Shielded wires will be used in most of the routing to avoid electrical noise from interfering with the signals.

3.3 Control/Feedback/Power Unit

The control/feedback unit is the second of two main components of the PROXIMIview device. This section breaks down the mechanical design choices of the unit as well as the placement of the single ultrasonic sensor on the enclosure. The enclosure measures 6.0 cm H x 10.5 cm W x 5.5 cm D and is to be worn around the user's waist using a lightweight and elastic band.

3.3.1 Mechanical Design

The design of the enclosure takes several important factors into consideration. The first of these is the weight of the device. An objective was to minimize the weight of the enclosure unit in order to allow the device to be comfortable and non-restricting to the user's range of motion. The largest contributor to the mass of the device is the battery pack. The custom battery pack has a mass of 248 grams. Next, the mass of the Arduino MCU and associated components is 70 grams. The mass of the enclosure-mounted PING))) sensor contributes 9 grams. This gives a total component mass of 327 grams (not including the enclosure itself).

In order to minimize the weight, a decision was made to use a lightweight but strong polycarbonate material for construction of the control unit enclosure. This material is light, shatter and crack resistant, and does not transmit heat very well. The enclosure is black in colour and connected on each side to a lightweight, elastic strap which is to be worn around the user's waist.

3.3.1.1 Sensor Placement and Design

In order to provide feedback of obstacles that are below head/neck level to the user, we have installed a fourth PING))) ultrasonic sensor on the control unit enclosure. This sensor is forward-facing and is placed right in the middle of the top of the enclosure. The placement of the sensor as well as the dimensions of the enclosure is shown in the rendering below.

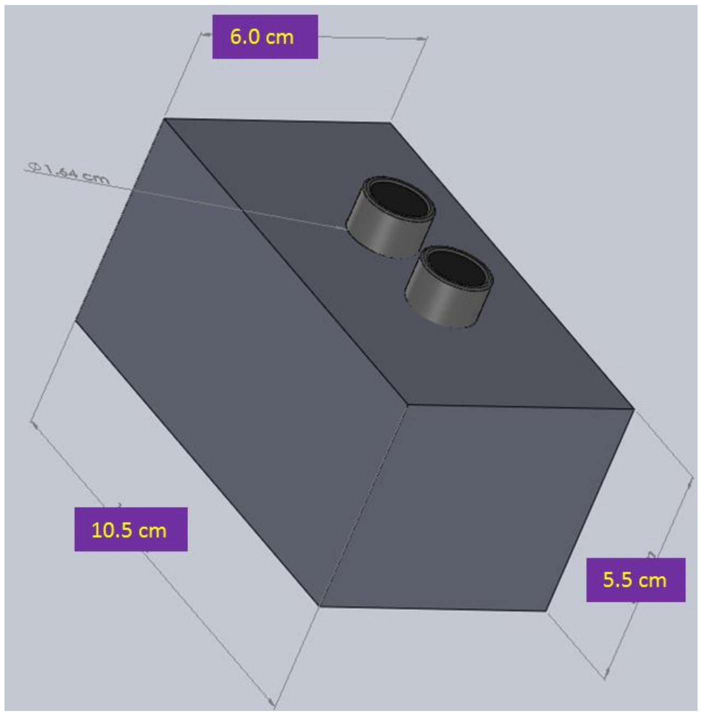


Figure 3.6: Enclosure Sensor Placement and Dimensions

3.3.2 User Interface

Because the PROXIMIview device's primary user is a visually-impaired person, all the features of the device must be easily accessible and controllable without the need for sight.

3.3.2.1 Power Control (ON/OFF Switch)

The "On/Off" switch controls operability of the entire device. It is positioned on the right side of the enclosure and consists of a single push button which cuts power from the battery to the Arduino microcontroller once in the "Off" state.

3.3.2.2 Audio Out Jack

The audio out jack is a single 3.5 mm port that is part of the Arduino VoiceShield control board. The jack will be located on the left side of the enclosure, positioned close to the volume dial in order to allow the user to easily plug in a set of headphones.

3.3.2.3 Battery Compartment

The PROXIMIview device is powered by a custom battery pack. This battery pack was made by Polar Batteries©. It consists of 8 AA-sized NiMH ion cells packaged in thermal plastic and connected to the Arduino MCU power supply input pin. The battery pack is placed within the enclosure, away from the user's body and protected from external elements.

3.4 Hardware

3.4.1 Parallax PING))) Sensors

The PING))) ultrasonic sensor is a low cost and an abundantly used tool for proximity measurement. This sensor uses an 40kHz ultrasonic burst (much larger than maximum human hearing frequency of 20Khz) and hence the distance to the target is calculated by the time it takes for the pulse to return from the object whose distance is to be determined. The transmitter sends out an ultrasonic pulse and receives the echo through the receiver. By measuring the pulse width of the echo, the distance is calculated. The PING))) ultrasonic sensor is a proximity sensor that precisely detects objects between 2cm - 3m. The PING))) sensor is a three terminal device which includes a ground pin(GND), signal pin(Sig) and a power supply pin (5V). It uses a 5V power supply and 30mA supply current.[2]

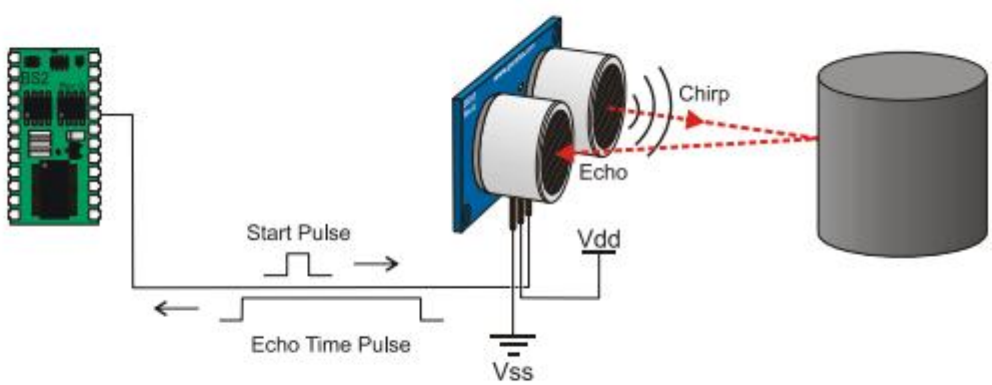


Figure 3.7: PING))) sensor operation

Advantages of the PING))) sensors:
Detection range from 2cm- 3m
Impervious to light conditions
35mA maximum Power Consumption
Easy Interface with Arduino microcontroller [2]

Limitations of the PING))) sensors:
Not efficient in detecting fast moving objects
Narrow angle sensors [2]

3.4.2 Arduino Uno Microcontroller

For our product we chose the Arduino Uno development board with the ATmega328P as the microcontroller chip for out control unit. This is a low power, 8 bit CMOS microcontroller based on the AVR RISC architecture. The AVR family of microcontrollers that this belongs to has a non-volatile on-chip memory which has unique characteristics for program storage . The program may be written in Assembly language code or in C language before uploading it onto the chip. Due to our teams experience and proficiency in the C language, we chose that to write our code. This microcontroller included two 8-bit timers and a 16-bit timer with a separate prescaler and a compare mode, which fulfilled our need for this project. We chose this microcontroller as this requires very low power consumption 0.2mA at 1.8V at room temperature at its active mode. Our project also demanded management of multiple inputs and generation of multiple outputs simultaneously and this microcontroller allowed us to have that parallel processing. Therefore parallelism is an important parameter to consider for portable devices such as ours. This also includes 14 digital I/O pins, six of which are PWM channels. This also includes 6 analog input pins. The DC current per I/O pin is 40mA and the DC current for the 3.3 V pin is 50mA. The Operating voltage is about 5V, where the recommended input voltage is about 7-12 V. The figure below depicts various pins on the ATmega328P chip [3]:

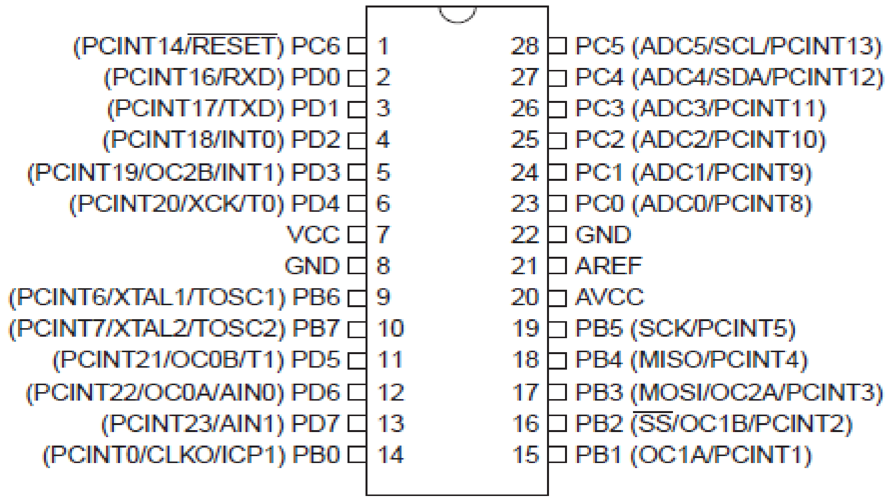


Figure 3.8: Pin diagram of the ATmega328P chip[3]

Table 1 below depicts the various pins to be used by all the components.

Pins	Usage
6	Signal wire from the Top Centre Sensor
7	Signal wire from the Left Centre Sensor
8	Signal wire from the Right Centre Sensor
9	Vibrator Module
10	Signal wire from the Bottom Sensor
2	Voice Shield Pin
3	Voice Shield Pin
4	Voice Shield Pin
5	Voice Shield Pin

Table 1: Pin assignments of the ATmega328P chip for the product[3]

Table 1 provides a clear view of all the components involved in the low level design. This is indicated as follows:

- The VoiceShield that is placed on top of the Arduino Uno board will be using digital pins 2-5.
- Four PING))) sensors involved in the overall design will be using Digital Pins 6,7,8 and 10.
- The vibrator module will be using the digital pin 9, which supports pulse width modulation(**PWM**).
- Voltage level will be read through Analog pin 0 of the Arduino Uno to provide auditory feedback when the battery level goes down.

- All components will be connected to ground using the GND pin of the Arduino Uno. Power will be supplied through the Vin Pin using a 10.8V battery pack.
- Audio feedback will be obtained using a pair of headphones that will be connected to the audio out of the Voice Shield.

3.4.3 Arduino VoiceShield™

Manufactured by Spikenzie Labs, the VoiceShield (VS) is an analog shield for the Arduino that is used to play audio sound bytes by the use of a CMOS-based ISD4003 chip from Windbond with an on-chip oscillator. This shield is utilised to provide auditory feedback to the visually impaired user. The VoiceShield's stackable design also helps in minimizing the size of the device, which is an imperative parameter to consider due to the portable nature of the device.

Since voice messages are stored on to the chip without digitization or compression, it provides high quality voice reproduction. To receive auditory feedback, the user simply has to plug in headphones into the 3.5 mm line out jack. The VS uses digital pins 2, 3, 4, and 5 of the Arduino Uno Board. Since the ISD (Information Storage Device) chip runs at 3V and the Arduino uses 5V, the VS has level shifting circuitry and an on board voltage regulator with filtering capacitors to safely run both devices. Figure 3.9 shows the VoiceShield.

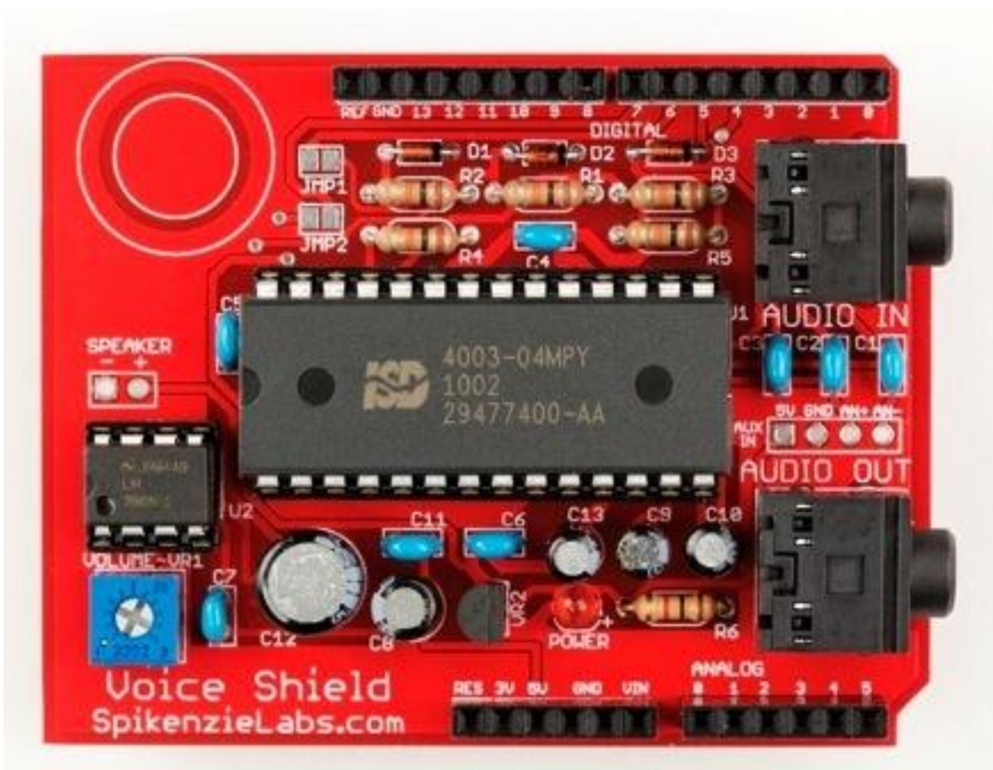


Figure 3.9: Spikenzie Lab's VoiceShield Board [5]

The ISD4003 chip is ideal for our purposes of providing feedback to the user due to its effective record and playback feature. It can record up to 4 minutes of audio at a sampling rate of 8KHz. Moreover, the ISD provides zero-power message storage as a result of its on-chip high density flash memory array. Its typical operating current for playback is a manageable 15mA [6]. To reduce pin count, address and control are achieved through a Serial Peripheral Interface (SPI). Figure 3.10 and 3.11 show the block diagram and pin diagram of the chip respectively.

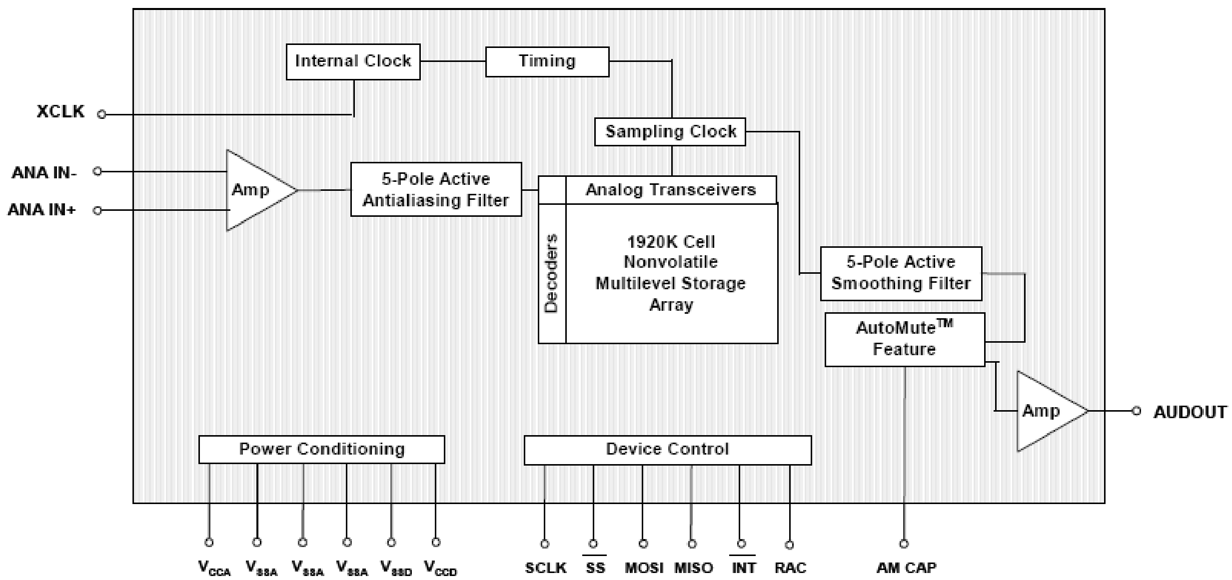


Figure 3.10: Block diagram of the VoiceShield [6]

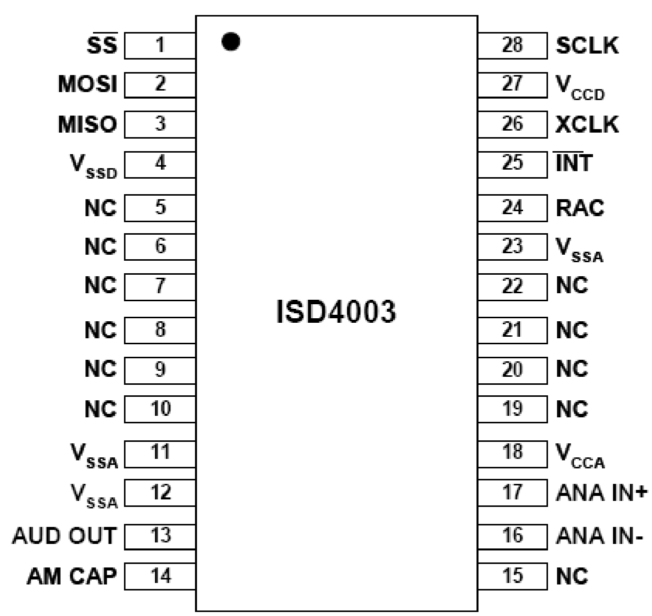
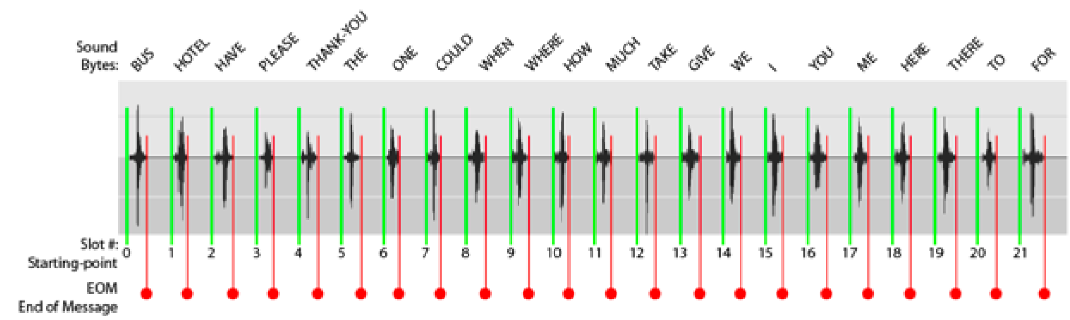


Figure 3.11: Pin diagram of the ISD4003 chip [6]

Messages are recorded onto the ISD using the VS`s 3.5mm line in jack. The 4 minutes of available recording time can be divided into 1-1200 memory slots. Consider the scenario where 80 sound bytes are chosen – this would give 80 messages of 3 seconds each (240 seconds/80 = 3 seconds). Once a message is recorded, the VS inserts an EOM (End of Message) marker immediately after the sound byte. At the end of the message, the EOM pin is pulled LOW to indicate to the VS that the sound is finished. Since the MCU does not have to accomplish this task, both memory and processing time are saved. Figure 3.12 shows an illustration of the audio memory.



The green bars represent the memory divided equally into sound slots, the left most side of each slot is where the sound starts to play when a particular slot is addressed. The red bars (EOM) (end of message) mark the end of a sound recording. The EOM spacing coincides with the point where a recording is stopped, and are automatically inserted into the audio memory. The EOM marker could be at any point within the sound byte and depend on the length of the recorded sound.

Figure 3.12: An illustration of the VS's audio memory [5]

3.4.4 LilyPad Vibe Board

The LilyPad Vibe Board is used in the proof-of-concept device to provide tactile feedback to the user. It uses a 10mm shaftless vibration motor manufactured by Precision Microdrives. This small and lightweight Vibe Board (Figure 3.13) has a 20 mm outer diameter and a thin 0.8 mm PCB. It is driven by a pulse-width modulation (PWM) output from the Arduino Uno, which helps in providing different levels of power to the vibration motor. The more the power applied, the higher the frequency of vibrations and vice versa. Therefore, as the user approaches an object, the vibrations would increase in frequency.

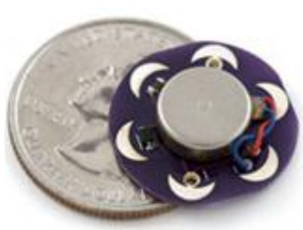


Figure 3.13: The LilyPad Vibe Board

3.4.5 Low Battery Indicator

In order to implement a low battery indicator, we utilize the fact that a read on the MCU's 10-bit ADC will return a value from 0 to 1023. This method is more effective than methods that use comparators and diodes because it eliminates components (and therefore space), and more importantly, consumes less power, which is an important parameter to consider for portable devices such as ours.

Since the Arduino Uno board's 5V output may become unstable if less than 7V is applied to the board [4] (this output is used to drive the ultrasonic PING))) sensors), we set our low battery indicator to sound (via the VoiceShield) if the battery level drops below 7V. Considering that we use a ~10V battery supply, and also keeping in mind that we cannot input more than 5V into the analog pin, we use a voltage divider circuit to bring the input voltage to the analog input to less than 5V. A simple read on this pin results in values from 0 to 1023, where 1023 would represent 5V. This method is used to determine the voltage level of the battery.

3.5 Software

- Sanity checks are included
- Written in a logical fashion to minimize processing times
- The Arduino code to determine the voltage level uses integer math since floating point consumes lots of memory
- Tested extensively to ensure minimal errors

3.6 Power Supply

3.6.1 Power Requirement

The components of the device have different power requirements, as outlined in this section.

3.6.1.1 VoiceShield

ISD 4003 chip on the VoiceShield can be powered using 3V power supply. VoiceShield uses the 5V supply from the Arduino Uno Board, interfacing them using an onboard 3V regulator with filtering capacitors and level shifting circuitry to safely connect both devices. VoiceShield uses digital pins 2,3,4 and 5.[5]

The output current per digital pin:	*40mA*
Number of pins used by the VoiceShield:	*4*
Max Current Supply :	*40 ×4 = 160 mA*

3.6.1.2 PING))) Sensors

A single PING)))sensor can be powered by 5V and 35mA(max) current.[2]

Number of PING))) sensors: *4*
Maximum current per PING))) sensor: *35mA*
Maximum current supply for all PING))) sensors: *35×4= 140mA*

3.6.1.3 Vibrator Module

The module uses a 5V supply and will be connected the PWM digital pin 9 of the Arduino Uno Board.[7]

Max Current Supply : *40mA*
Maximum current requirement of the integrated device : 160+ 140 + 40 = *<u>340 mA</u>*

3.6.2 Custom Battery Pack

A custom battery pack will be used to provide a 10.8V, 2000mAH power supply.

<u>Battery life : 2000mAH / 340mA = 5.88Hrs</u>

4.0 SYSTEM TEST PLAN

Testing will be carried out under 3 main sections: Component Testing, Integrated System Testing and Structural and Durability Testing.

4.1 Component Testing

4.1.1 Arduino Uno Board

A supply voltage of 7-12V guarantees stable operation of the Arduino Uno Board. A supply voltage in this range will be provided using a DC power supply to the Uno board. The voltage and current in each pin will be tested separately using a LED, when a digital write(Pin name, 'High')is written on the pins. This will guarantee the stable operation of all pins in the Arduino Uno.

4.1.2 PING))) Sensors

The device uses 4 PING))) sensors which will be connected to the digital pins of the Arduino Uno Board. Each sensor will be tested separated using the following configuration shown in figure 4.1. Using the sample code provided in the Arduino website, operation of each sensor will be tested using the serial monitor.[8]

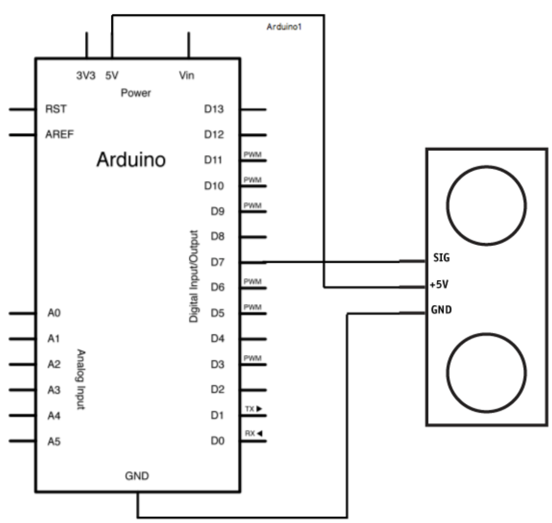


Figure 4.1: PING))) Sensor Testing

4.1.3 Vibrator Module

The vibrator module will be connected to one of the PWM pins of the Arduino Uno Board which will emit different vibration levels at different average voltage supplies. Operation of the vibrator module will be tested by providing different voltage supplies and sensing the intensity of the module.

4.1.4 VoiceShield

The voice shield uses the same power supply from the Arduino Uno board and will be stacked on top of the Arduino Uno board. VoiceShield will be using digital pins 2,3,4 and 5 from the Arduino Uno Board. The voice shield comes with a number of pre-recorded messages. In testing the voice shield, we will try to hear these pre-recorded messages by writing ' vs.ISDPLAY_to_EOM(n) ' on to the Arduino Uno Board(n being the number of the saved voice message) and obtaining the audio output through a pair of headphone that will be connected to the audio out of the VoiceShield. [5]

4.1.5 Power Supply

Using a multimeter, the supply voltage of the battery pack will be measured to guarantee constant supply of 10.8V.

4.2 Integrated System Testing

The Integrated System testing will deal with the testing procedure for the product system as a single unit. The use of specific ranges for detection has been taken into account. This is further explained by all the scenarios taken into account and also an excerpt of the logic is provided with the help of pseudocode as follows:

- **Scenario 1**
  This scenario deals with the proximity of the object about 3m from the user. The following pseudocode depicts the logic used in determining the proximity of the object 3m from the user. It will also alert the user by sending a voice message that says “Object 3m ahead”.
  If object 2.8m<x<3.2m then
  “Object 3m away”
  End if

- **Scenario 2**
  This scenario deals with the proximity of the object about 1.5m from the user. The following pseudocode depicts the logic used in determining the proximity of the object 1.5m from the user. It will also alert the user by sending a voice message that says “Object 1.5m ahead”.
  If object 1.3m<x<1.7m

"Object 1.5m away"
End if

- **Scenario 3**
  This scenario deals with the proximity of the object about 1m from the user. The following pseudocode depicts the logic used in determining the proximity of the object 1m from the user. This is further subdivided into separate subscenarios as the user comes closer to the object.

- *Subscenario 1:*
  The following pseudocode depicts the logic when the center sensor detects an object 1m away from the user and the also the two side sensors, that is, the right and the left sensors detect an object which is less than or equal to 1.5m. In this case a message "Dead-end" would be heard.
  If CenterSensor<=1m
  And if LeftSensor and RightSensor <=1.5m
  "Dead-end"

- *Subscenario 2:*
  The following pseudocode depicts the logic when the center sensor detects an object 1m away from the user and the also the two side sensors, that is, the right sensor is greater than 1.5m and the left sensor detect an object which is less than or equal 1.5m. In this case a message "Turn Right" would be heard. Since in this case, turning right would suffice.
  Else If CenterSensor<=1m
  And if LeftSensor <= 1.5and RightSensor >1.5m
  "Turn Right"

- *Subscenario 3:*
  The following pseudocode depicts the logic when the center sensor detects an object 1m away from the user and the also the two side sensors, that is, the left sensor is greater than 1.5m and the right sensor detect an object which is less than or equal 1.5m. In this case a message "TurnLeft" would be heard. Since in this case, turning left would suffice.
  Else If CenterSensor<=1m
  And if LeftSensor > 1.5and RightSensor <=1.5m
  "Turn Left"

- *Subscenario 4:*

The following pseudocode depicts the logic when the center sensor detects an object 1m away from the user and the also the two side sensors, that is, the right and the left sensors detect an object which is greater than or equal to 1.5m. In this case a message "Turn right or Left" would be heard.
Else If CenterSensor<=1m
And if LeftSensor and RightSensor >=1.5m
"Turn Right or Left"

- **Scenario 4**
  This scenario deals with the proximity of the object about 2m from the user detected by the bottom sensor located on the enclosure. The following pseudocode depicts the logic used in determining the proximity of the object 2m from the user at the wait level. It will also alert the user by sending a voice message that says "Object at Waist Level".
  If BottomSensor < TopCentreSensor and TopCentreSensor <=2m
  "Object at Waist Level"
  End if

- Another set of tests would indicate if the system battery is out of charge by checking to see if the voltage is less than 7V. If this were the case, a message that says "Low Battery" could be heard. Please refer to the Appendix for complete pseudocode.

4.3 Structural and Durability Testing

A number of durability tests will be carried out to analyze the structural integrity of the device.

4.3.1 Temperature Testing

The operation of the device will be observed under different temperature conditions from 0-50 ^{o}C.
Note: The operating Temperature of the PING))) sensors is 0-70 ^{o}C

4.3.2 Drop Testing

The device will be dropped 3 times from 2m from the ground level to observe the structural strength of the device.

4.3.3 Light conditions

The device uses ultrasonic sensors that emit ultrasonic bursts and listen to the echo to measure the distance from the obstacle. To guarantee that the device works efficiently under different light conditions, testing will be carried out in broad day light as well as night time.

5.0 CONCLUSION

The design specifications demonstrate the detailed information about our product design that will satisfy the functional requirements of the Wearable Proximity Detector (as seen in Functional Specifications document). We will ensure that the required functionality of our product is present through our through test plans included in the design specifications. This document would provide clear goals for the proximity detector prototype. The prototype development is well underway and we are expected to have working and a fully deployable prototype by December 15th 2011.

Simon Fraser University
8888 University Drive
Burnaby, BC V5A 1S6

6.0 References

[1] PROXIMIview Technologies, "Functional Specifications for a Wearable Proximity Detector to Aid the Visually Impaired", [Oct. 2011].

[2] PARALLAX INC.(n.d.). PING))) Ultrasonic Sensor [ONLINE]. Available: http://www.parallax.com/tabid/768/ProductID/92/Default.aspx, [Nov. 16, 2011].

[3] ATMEL(n.d.). ATmega328P[ONLINE]. Available: http://www.atmel.com/dyn/products/product_docs.asp?category_id=163&family_id=607&subfamily_id=760&part_id=4198, [Nov. 16, 2011].

[4] Arduino, Arduino Uno [Online]. Available: http://www.arduino.cc/en/Main/arduinoBoardUno, [Nov. 16, 2011].

[5] Spikenzie Labs, VoiceShield [ONLINE]. Available: http://spikenzielabs.com/SpikenzieLabs/VoiceShield.html, [Nov. 16, 2011].

[6] Nuvaton. ISD 4003 Series[ONLINE]. Available: http://www.nuvoton.com/hq/enu/ProductAndSales/ProductLines/ConsumerElectronicsIC/ISDVoiceIC/ISDChipCorder/Documents/ISD4003.pdf, [Nov. 16, 2011].

[7] Sparkfun Electronics, LilyPad Vibe Board[ONLINE]. Available: http://www.sparkfun.com/products/8468, [Nov. 17, 2011].

[8] Arduino, PING))) Ultrasonic Rage Finder [ONLINE]. Available: http://arduino.cc/en/Tutorial/Ping, [Nov. 17, 2011].

6.0 APPENDIX – SOURCE CODE

Current Arduino Code

```
// Include Libraries
#include <VoiceShield.h>
VoiceShield vs(80);
#define LEFT_PING 6
#define CENTER_PING 7
#define RIGHT_PING 8
#define BOTTOM_PING 10
#define AnalogPin  0
// Initialize Variables
int VPin = 9;
long cmLeft= 0;
long cmCentre = 0;
long cmRight = 0;
long cmBottom = 0;
int pinval = 0;
int integer = 0;
int decimal = 0;
// Setup function will run once at the start of the program
void setup(){
  Serial.begin(9600);
  pinMode(VPin,OUTPUT);
//Voltage Level Indicator
  pinval = analogRead(AnalogPin);
  integer = pinval / 102, DEC;
  decimal = pinval % 102,DEC;
  Serial.print("voltage is: ");
  Serial.print(integer);         // print the integer value of volts
```

```
 Serial.print(".");            // decimal point
 Serial.println(decimal);      // hundredths of a volt
 if (integer < 7){
  Serial.println("Low Battery");
  vs.ISDPLAY_to_EOM(46);       //Low Battery  }}
// Loop function
void loop(){
  int VLevel = map(cmCentre, 50,150,255,100);
 cmLeft = ping(LEFT_PING);
 delay(17);
 cmCentre = ping(CENTER_PING);
 delay(17);
 cmRight = ping(RIGHT_PING);
 delay(17);
 cmBottom = ping(BOTTOM_PING);
 delay(17);
//Setting up serial monitor
 Serial.print(cmLeft);
 Serial.print("LEFTcm, ");
 Serial.print(cmCentre);
 Serial.print("CENTERcm, ");
 Serial.print(cmRight);
 Serial.print("RIGHTcm");
 Serial.print(cmBottom);
 Serial.print("BOTTOMcm");
 Serial.println();
// Sanity checks to check functionality of the sensors
 if (cmCentre== 0){
 vs.ISDPLAY_to_EOM(9); }
 if (cmBottom == 0){
```

```
vs.ISDPLAY_to_EOM(10);}
if (cmRight == 0){
vs.ISDPLAY_to_EOM(11); }
if (cmLeft == 0){
vs.ISDPLAY_to_EOM(12);}
// Setting up the vibrator module
if (cmCentre >=50 && cmCentre <= 150 ) {
  analogWrite(VPin,VLevel); }
else {
   analogWrite(VPin,0); }
// All Scenarios
if (cmCentre <= 355){
// Object 3m away
  if (cmCentre <= 320 && cmCentre >= 280){
    Serial.println("Object 3m away");
    vs.ISDPLAY_to_EOM(0); }
// Object 1.5m away
  else if(cmCentre <=170 && cmCentre >= 130){
    Serial.println("Object 1.5m away");
    vs.ISDPLAY_to_EOM(77);   }
// Deadend
  else if(cmCentre <= 100 && cmCentre !=0){
    if (cmLeft <= 150 && cmRight <= 150){
      vs.ISDPLAY_to_EOM(58);
      Serial.print("Deadend ");
     delay (1000);  }
// Turn Right
  else if (cmLeft <=150 && cmRight > 150){
    vs.ISDPLAY_to_EOM(48);
    Serial.print("Turn Right"); }
```

```
// Turn Left
    else if (cmRight <=150 && cmLeft > 150){
      vs.ISDPLAY_to_EOM(49);
      Serial.println("Turn Left");}
// Turn Left or Right
    else if (cmLeft >= 150 && cmRight >=150) {
      vs.ISDPLAY_to_EOM(68);
      Serial.println("Turn Left or Right");  }}
//Functions
long ping(int pin){
  pinMode(pin, OUTPUT);
  digitalWrite(pin, LOW);
  delayMicroseconds(2);
  digitalWrite(pin, HIGH);
  delayMicroseconds(5);
  digitalWrite(pin, LOW);
  pinMode(pin, INPUT);



long duration = pulseIn(pin, HIGH);
  return (microsecondsToCentimeters(duration));}
long microsecondsToCentimeters(long microseconds){
  return microseconds / 29 / 2;}
void threemeters(){
  vs.ISDPLAY_to_EOM(41); //object
  vs.ISDPLAY_to_EOM(3);  //three
  vs.ISDPLAY_to_EOM(30); //meters
  vs.ISDPLAY_to_EOM(60); //infront}
void onepointfivemeters(){
  vs.ISDPLAY_to_EOM(41); //object
```

```
  vs.ISDPLAY_to_EOM(1);  //one
  vs.ISDPLAY_to_EOM(64);  //point
  vs.ISDPLAY_to_EOM(5);  //five
  vs.ISDPLAY_to_EOM(30); //meters
  vs.ISDPLAY_to_EOM(60); //infront}
void turnright(){
  vs.ISDPLAY_to_EOM(72); //turn
  vs.ISDPLAY_to_EOM(63);  //right     }
```