

4-2-2003

A multi-sensor sonar system for indoor range measurement as a navigational aid for the blind

Maroof Haider Choudhury
Florida International University

DOI: 10.25148/etd.FI14060818

Follow this and additional works at: <https://digitalcommons.fiu.edu/etd>

 Part of the [Electrical and Computer Engineering Commons](#)

Recommended Citation

Choudhury, Maroof Haider, "A multi-sensor sonar system for indoor range measurement as a navigational aid for the blind" (2003).
FIU Electronic Theses and Dissertations. 2348.
<https://digitalcommons.fiu.edu/etd/2348>

This work is brought to you for free and open access by the University Graduate School at FIU Digital Commons. It has been accepted for inclusion in FIU Electronic Theses and Dissertations by an authorized administrator of FIU Digital Commons. For more information, please contact dcc@fiu.edu.

FLORIDA INTERNATIONAL UNIVERSITY

Miami, Florida

A MULTI-SENSOR SONAR SYSTEM FOR INDOOR RANGE MEASUREMENT AS
A NAVIGATIONAL AID FOR THE BLIND

A thesis submitted in partial fulfillment of the

requirements for the degree of

MASTER OF SCIENCE

in

ELECTRICAL ENGINEERING

by

Maroof Haider Choudhury

2003

To: Dean Vish Prasad
College of Engineering

This thesis, written by Maroof Haider Choudhury, and entitled A Multi-sensor Sonar System for Indoor Range Measurement as a Navigational Aid for the Blind, having been approved in respect to style and intellectual content, is referred to you for judgment.

We have read this thesis and recommend that it be approved.

Subbarao V. Wunnava

Malek Adjouadi

Armando Barreto, Major Professor

Date of Defense: April 2, 2003

The thesis of Maroof Haider Choudhury is approved.

Dean Vish Prasad
College of Engineering

Dean Douglas Wartzok
University Graduate School

Florida International University, 2003

DEDICATION

I dedicate this thesis to my wife, Rosy. Without her encouragement, support, patience and love, the completion of this thesis would not have been possible.

ACKNOWLEDGMENTS

I extend my sincere gratitude and appreciation to the people who made this master's thesis possible. Special thanks are due to my supervisor Dr. Armando Barreto for his support, guidance, suggestions and great teamwork. I am highly indebted Dr. Adjouadi Malek for his support and kind consideration. Many thanks are due to Dr. Subbarao Wunnava for his encouragement and feedback. I would also like to acknowledge with much appreciation the valuable feedback, cooperation, teamwork and wonderful company of my fellow researchers at the DSP laboratory.

In addition, I would like to thank the staff members of the electrical and computer engineering department for their invaluable assistance and kindness.

The work behind this thesis was sponsored by NSF grant EIA-9906600.

ABSTRACT OF THE THESIS

A MULTI-SENSOR SONAR SYSTEM FOR INDOOR RANGE MEASUREMENT AS A NAVIGATIONAL AID FOR THE BLIND

by

Maroof Haider Choudhury

Florida International University, 2003

Miami, Florida

Professor Armando Barreto, Major Professor

This project involved the design, implementation and testing of a multi-directional range measurement system for indoor applications, as an independent sensing module for a navigational tool for blind individuals. The system provides real-time distance measurements to the closest surrounding obstacles in six radial directions, and a North direction indication. Custom hardware and software were developed to interface six ultrasonic ranging modules and a digital compass to a PIC16F77 microcontroller. A multi-directional range measuring technique appropriate for this setup was identified experimentally. The final product of this project is a navigational instrument integrating the custom hardware and software, a portable PC, six ultrasonic range sensors and a digital compass. The thesis provides details about the components selected for the system, the design philosophy, the system hardware and software, and the test procedures followed to evaluate the performance of the overall system.

TABLE OF CONTENTS

| CHAPTER | PAGE |
|--|------|
| 1 Introduction..... | 1 |
| 1.1 Previous Approaches in Blind Navigation Aids..... | 2 |
| 1.2 Electronic Travel Aid (ETA)..... | 2 |
| 1.3 Scope of work..... | 8 |
| 1.3.1 Head Related Transfer Function (HRTF)..... | 8 |
| 1.4 Problem Specification..... | 11 |
| 2 System Outline..... | 13 |
| 2.1 Sonar Ranging Principle..... | 14 |
| 2.2 Component Specification Requirements..... | 16 |
| 2.2.1 Ultrasonic Sensor..... | 17 |
| 2.2.2 Controller Platform..... | 17 |
| 2.2.3 Devantech SRF04 Ultrasonic Ranger..... | 18 |
| 2.2.4 Devantech Magnetic Compass CMPS01..... | 20 |
| 2.2.5 Object Oriented Programmable Integrated Circuit (OOPIC)..... | 22 |
| 2.2.6 Counter Registers..... | 26 |
| 2.3 Mechanical Layout of the Sensor Array..... | 26 |
| 2.4 Data Communications..... | 28 |
| 2.5 Host System..... | 28 |
| 3 SCU Hardware..... | 29 |
| 3.1 System Architecture..... | 29 |
| 3.2 Instrumental Technique for Registering Sonar Range Readings..... | 29 |
| 3.3 OOPIC I/O Configuration..... | 31 |
| 3.4 Ultrasonic Sensor Interface..... | 32 |
| 3.4.1 Channel Ping Control..... | 34 |
| 3.4.2 Counter Rollover Prevention..... | 34 |
| 3.5 Compass Module Interface..... | 35 |
| 3.6 Serial Communications Interface..... | 36 |
| 3.7 Reference Clock..... | 36 |
| 3.8 SCU Construction..... | 37 |
| 4 Software..... | 38 |
| 4.1 Pinging Schemes..... | 38 |
| 4.1.1 Sweep Mode..... | 38 |
| 4.1.2 Flood Mode..... | 39 |
| 4.1.3 Interlaced Mode..... | 39 |

| | | |
|-------|------------------------------------|----|
| 4.1.4 | Register Clear..... | 40 |
| 4.2 | SCU Program Flow..... | 41 |
| 4.3 | Host Program Flow..... | 41 |
| 4.4 | Query ID..... | 43 |
| 4.5 | Packet Structures..... | 45 |
| 4.5.1 | Command Packet..... | 46 |
| 4.5.2 | Data Packet..... | 47 |
| 4.6 | Expandability..... | 47 |
| 5 | Test and evaluation..... | 48 |
| 5.1 | Circuit Testing..... | 48 |
| 5.2 | Serial Communications Testing..... | 48 |
| 5.3 | Rollover Prevention Test..... | 49 |
| 5.4 | Compass Test..... | 50 |
| 5.5 | Ranging accuracy..... | 50 |
| 5.6 | Performance Evaluation..... | 53 |
| 6 | Conclusion ANd Future Work..... | 56 |
| 6.1 | Suggestions..... | 56 |
| | List of References..... | 58 |
| | Appendices..... | 60 |

LIST OF TABLES

| TABLE | PAGE |
|---|------|
| Table 2-1. Technical specifications of the Devantech SRF04 sonar | 18 |
| Table 2-2. The Devantech CMPS01 specification..... | 21 |
| Table 2-3. CMPS01 Register assignment | 21 |
| Table 2-4. Pin configuration of the OOPIC 40 pin IDE connector | 25 |
| Table 3-1. I/O Configuration table of the OOPIC for the SCU application | 31 |
| Table 3-2. Summary of the rollover prevention logic..... | 35 |
| Table 4-1. Command packet structure | 46 |
| Table 4-2. 1 Byte Encoded Command..... | 46 |
| Table 4-3. Data Packet Structure | 46 |

LIST OF FIGURES

| FIGURE | PAGE |
|--|------|
| Figure 1-1. Photograph of a person using the Guide-Cane..... | 3 |
| Figure 1-2. Photograph of the KASPA Sonicguide (http://www.sonicvision.co.nz)..... | 4 |
| Figure 1-3. The concept of NavBelt | 5 |
| Figure 1-4. Functional diagram of an HRTF rendering engine | 9 |
| Figure 1-5. Components of the proposed ETA system..... | 10 |
| Figure 2-1. Schematic showing the arrangements of the front-end components..... | 13 |
| Figure 2-2 Sonar operation principle | 14 |
| Figure 2-3. The Devantech SRF04 Ultrasonic Ranger module (left); and the Ultrasonic beam pattern of the transducers (right) [from www.acroname.com] | 18 |
| Figure 2-4. Functional Schematic of the Devantech SRF04 module..... | 19 |
| Figure 2-5 SRF04 handshake protocol | 20 |
| Figure 2-6. The Devantech CMPS01 Digital Compass Module, top view (left), and bottom view (right) [from www.acroname.com]..... | 21 |
| Figure 2-7. Photograph of the OOPIC™ board showing its different features [from www.oopic.com]..... | 22 |
| Figure 2-8. Architecture of the Microchip PIC16F77 micro controller [from www.microchip.com] | 24 |
| Figure 2-9 The 74LS590 8 bit counter register..... | 26 |
| Figure 2-10. Ultrasonic sensor mounting directions and frame of reference | 27 |
| Figure 2-11. Photograph of the six ultrasonic sensors mounted on the headgear; Notice the compass module mounted on top of the center sonar housing..... | 27 |
| Figure 3-1. Hardware Implementation for Sonar Range Measurement Applying Equation 2.6 (top) ; and the timing diagram (bottom)..... | 30 |

| | |
|---|----|
| Figure 3-2. Schematic Diagram of the OOPIC sonar interface circuit | 33 |
| Figure 3-3. timing diagram explaining the OOPIC sonar interface..... | 34 |
| Figure 3-4. Interface between the OOPIC and the Compass module..... | 35 |
| Figure 3-5. Reference clock circuit for the Sonar Echo Return Counters | 36 |
| Figure 3-6. Photograph showing the SCU Circuit construction and Housing..... | 37 |
| Figure 4-1. Program Flowchart of the SCU..... | 42 |
| Figure 4-2. The program flow chart for the host system | 44 |
| Figure 4-3. Graphical presentation method for the acquired ranging and bearing data. The Blue line indicates the Earth's magnetic North, with respect to the user's facing direction. The red asterisks indicate the detected obstacle positions in the respective azimuth. The dashed lines connecting the asterisks are used only to enhance the visibility of the asterisk positions. Range measurements are available at only the six designated azimuths. | 45 |
| Figure 4-4. Packet Exchange Process Between Host and SCU..... | 45 |
| Figure 5-1 Serial port data monitoring..... | 49 |
| Figure 5-2. Measurements taken in the three modes, graphically displayed for cases 1 and 2 | 51 |
| Figure 5-3. Measurements taken in the three modes, graphically displayed for cases 3 through 6..... | 52 |
| Figure 5-4. Beam overlap causing cross talk between active neighboring sensors..... | 53 |
| Figure 5-5. Top and bottom images showing alternating sensor measurements using the interlace mode | 54 |

1 INTRODUCTION

Visually impaired individuals face several major challenges while navigating their way through indoor environments. The first challenge is to identify exact locations of surrounding obstacles. Based on this information, they need to figure out a safe route through these obstacles. In addition to this, they also need to find openings and branching pathways (i.e., door openings, side paths, corridors etc.). While they are navigating, blind individuals also need to keep track of their orientation and heading. Finally, they need to memorize their environment so that they can estimate their position in reference to their destination.

Traditionally the white cane has been the standard for blind navigation. Basically the cane gives the user an idea about obstacles in the directions sought by the user. Guide dogs are helpful in blind navigation as they can be trained to walk along certain landmarks such as pavements; and also help the blind individual in choosing routes at intersection points; thus increasing the confidence and mobility of the blind individual.

Certain types of electronic navigation system commonly known as Electronic Travel Aides (ETA's) assist visually impaired people in navigation and mobility. Several devices have been developed in the past as an electronic replacement of the guide dog. The versatility and ease of use of these ETAs have improved significantly with advances in electronic devices, computing platforms and Digital Signal Processing (DSP) techniques. In essence, the function of an ETA is to acquire information of the user's surrounding environment and present this information to the blind individual through a non-visual interface.

1.1 Previous Approaches in Blind Navigation Aids

The most successful and widely used blind navigation aid is the white cane, which defines the golden standard for blind navigation. This purely mechanical device is used to detect obstacles on the ground, uneven surfaces, steps, holes and other hazards. The fact that it is inexpensive and easy to carry, makes it widely popular. The main problem with the white cane is that users require over one hundred hours of training, which is a substantial hidden cost. The user has to actively scan the area in front of him/her. In addition, the user cannot scan anything out of his/her reach (3-6 feet). Therefore the user perceives limited information about the environment. Another drawback of the white cane is that obstacles can be detected only by contact. This can become inconvenient to the traveler as well as the surrounding objects, for example, when traveling in a crowded street. Travel can even become unsafe through improper use of the white cane. The white cane is also not well suited for detecting potentially dangerous obstacles at head level.

1.2 Electronic Travel Aid (ETA)

During the last thirty years, there has been significant development in range finding and positioning technologies such as radar, sonar, laser reflectometry, Global Positioning System (GPS) etc. The fact that such instruments have become highly compact and requires only small amount of power to operate, has enabled developers to design highly portable range finding systems that can be worn by a blind person. Advances in DSP and computational platforms have empowered real time data processing such that, range information can be processed and delivered through vibration

or acoustic means with negligible delay. As a result, a wide variety of ETAs are available today.

The concept of the ETAs, was first introduced with the C5 laser Cane [1]. This ETA, consists of three sets of optical transmitters, one that detects obstacles at head level, the second detects objects within a 1.5m-3.5m range, and the third detects drop-offs in front of the user.

The Russel Pathsounder was one of the earliest ultrasonic aids consisting of two ultrasonic transducers mounted on a board placed around the user's neck. The user is warned by a series of clicks (3) that increase in intensity depending on the distance to the object [6].

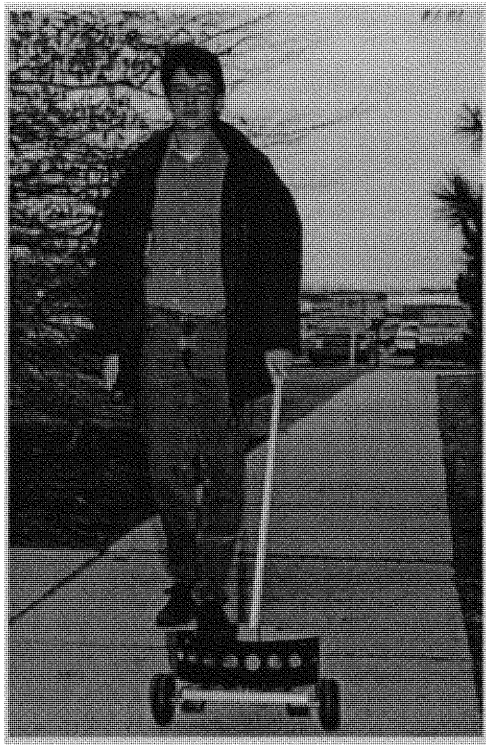


Figure 1-1. Photograph of a person using the Guide-Cane

The Guide-Cane (Figure 1-1) [9], developed by the University of Michigan Advanced Technologies Lab, is an obstacle avoidance system, which is equipped with a pair of computer-controlled individually breaking wheels that guides a blind person around obstacles. It utilizes a sonar array that scans a total angular spacing of 120° for obstacles, an electronic odometer to measure traveled distance and a compass to assess heading. On detecting approaching obstacles, the system computes a route to avoid the obstacle, then applies breaks on one of the wheels which steers the Guide-Cane toward that direction.

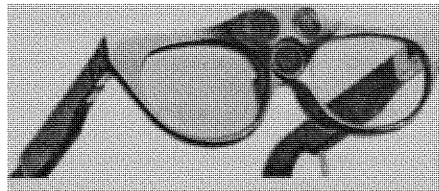


Figure 1-2. Photograph of the KASPA Sonicguide (<http://www.sonicvision.co.nz>)

Another system that relies on ultrasonic guidance is the KASPA Sonicguide [5] (Figure 1-2), which consists of an ultrasonic wide-beam (55° cone) transmitter mounted between the lenses of a special set of spectacles and one receiver located on each side of the transmitter, which present the echoed signals to the user's left and right ear separately. In order to present the distance of objects in an audible form, the signals are encoded and delayed in the frequency of the demodulated low-frequency tone. With the additional help of exploration by head motion, it provides clear information about the objects location [7]. A problem presented by the Sonicguide was the low angular resolution. This was later improved by adding a third high-resolution monaural sensor to

the wide angle binaural sensing system, improving the resolution by a factor of 6. The enhanced system is marketed by Sonic Vision Ltd. [3]

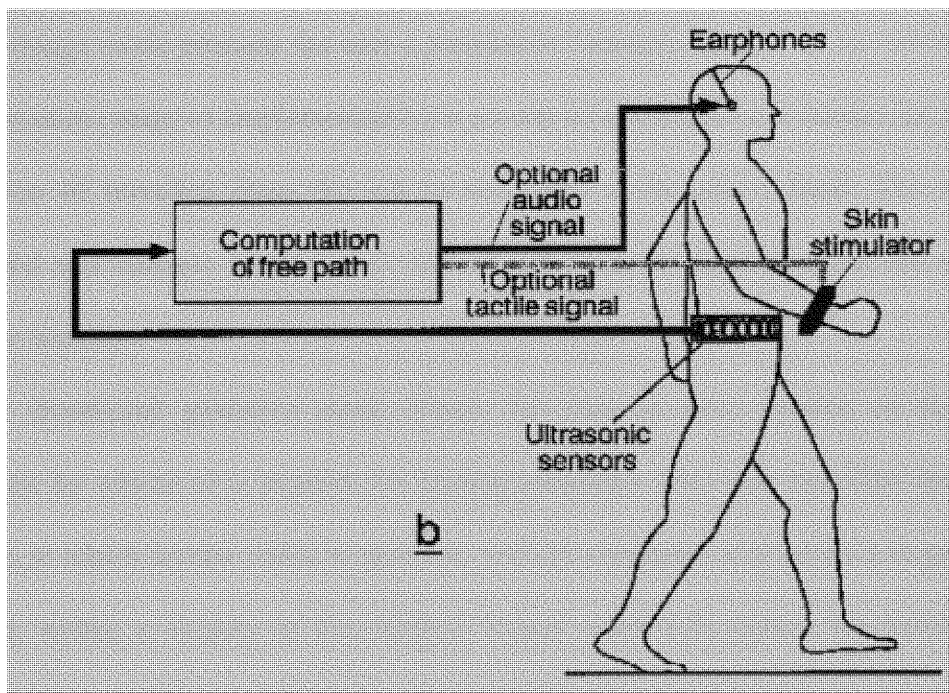


Figure 1-3. The concept of NavBelt

Another ETA is the NavBelt (Figure 1-3), which consists of a belt with ultrasonic sensors that cover a larger area regardless of the user orientation, and a large computer, carried in a backpack. The computer is in charge of processing the information collected by the sensors, and applying algorithms similar to those used for obstacle avoidance by ambulatory robots [8]. The information processed is then delivered to the user via stereophonic headphones. The NavBelt has three operational modes: guidance mode, image mode, and directional guidance mode.

The guidance mode provides the user with information regarding the recommended travel direction, speed and proximity of objects through a single stereophonic tone. The

tone changes in frequency determine the recommended travel speed. In this mode the user enters the target position into the system by indicating its coordinates.

The image mode presents the user a panoramic “acoustic image” of the environment. This acoustics image is implemented by a sweep of stereophonic sounds that give the impression of traveling from the left to the right of the user’s head. The sound is spatialized and changes in pitch and volume, allowing the user to avoid the obstacles closest to him/her.

The directional guidance mode allows the user to have control of the global navigation by means of a joystick for testing purposes. The user specifies the overall direction in which he/she wants to proceed, and the computer will select a target in that direction, within a vicinity of five meters. Future work will replace the joystick with special auditory coding or a speech control device. The NavBelt developers report that having three operational modes allows different levels of assistance and different level of conscious effort from the user [2]. Given the complexity of the NavBelt system, the user requires extensive training in order to become familiarized with it.

Another type of traveling aid specially directed towards providing the global position of the user and guiding him/her to the desired target, is known as the global navigation aid (GNA)[7][2]. The GNA, like the Talking Signals [10] does not integrate obstacle avoidance, but rather orients the user with equipment that receives the signals from infrared directional transmitters that are installed throughout the travel environment. The relative location of those transmitters is communicated to the user through digital speech. A disadvantage of this technology is the cost of installing and maintaining a network with these characteristics [11]. Another approach is being developed by researchers at

University of California. They have proposed a system that consists of a module that determines the traveler's position and orientation in space, a Geographic Information System (GIS). The system incorporates the software and the spatial database required to indicate the traveler's orientation and GPS coordinates in the context of the surrounding environment, and a user interface that uses a spatialized voice to advise the user of the names of buildings, street intersections, etc. coming from the appropriate locations [11]. In their simulation they evaluated four different auditory display modes. Three of them were implemented with conventional non-spatialized speech and the fourth one involved spatialized sounds. They reported that spatialized sound was the mode with the best rating.

Researchers at the University of Florida saw the limitations in contextual awareness that previous studies reported, and designed a prototype called Drishti, with similar features to the one described previously by Loomis, but allowing the user to interact with the database, making the system dynamic. The location information in Loomis' system is updated every 10 seconds, and the interaction is restricted to one way (database-to-user). In the Drishti system the user is capable of querying the information of environmental conditions and landmark information from a spatial database that supports a dynamically changing environment, providing contextual awareness to the user [12]. This system integrates different types of technologies such as Geographic Information System (GIS), Global Positions System (GPS), voice recognition and synthesis, wireless networks, and wearable computers. Intelligence can be added into the system while the user is navigating. For example, if the user encounters a pothole in the walkway, he/she can

make a note of it in the database and the system will then warn the user of the pothole the next time he/she passes by that particular location.

1.3 Scope of work

It is clear that researchers are working towards the development of ETAs that can mimic a sighted person's ability to navigate the real world. It is also clear that they have appreciated the value of sound spatialization as a helpful technique to enhance user learning in "virtual environments" and in order to provide orientation when navigating in the real world. A limitation in some of these systems is the extensive amount of training hours they require before the user becomes proficient in their operation. The promising results by Loomis when guiding their subjects with spatialized sounds and spending only two hours per session, back up the idea that a system that delivers the information of obstacles in the surroundings with the use of this type of 3-D auditory display, could be intuitive and not overwhelming to the user, minimizing even more the training time for one blind individual to become accustomed to it.

1.3.1 Head Related Transfer Function (HRTF)

The Digital Signal Processing Laboratory at Florida International University has been developing enhancements to a 3D surround sound modeling technique called "Head Related Transfer Function" (HRTF) described in [13]. Sound spatialization techniques based on the HRTF concept are capable of producing 3D sounds that the listener can hear with stereo headphones or speakers.

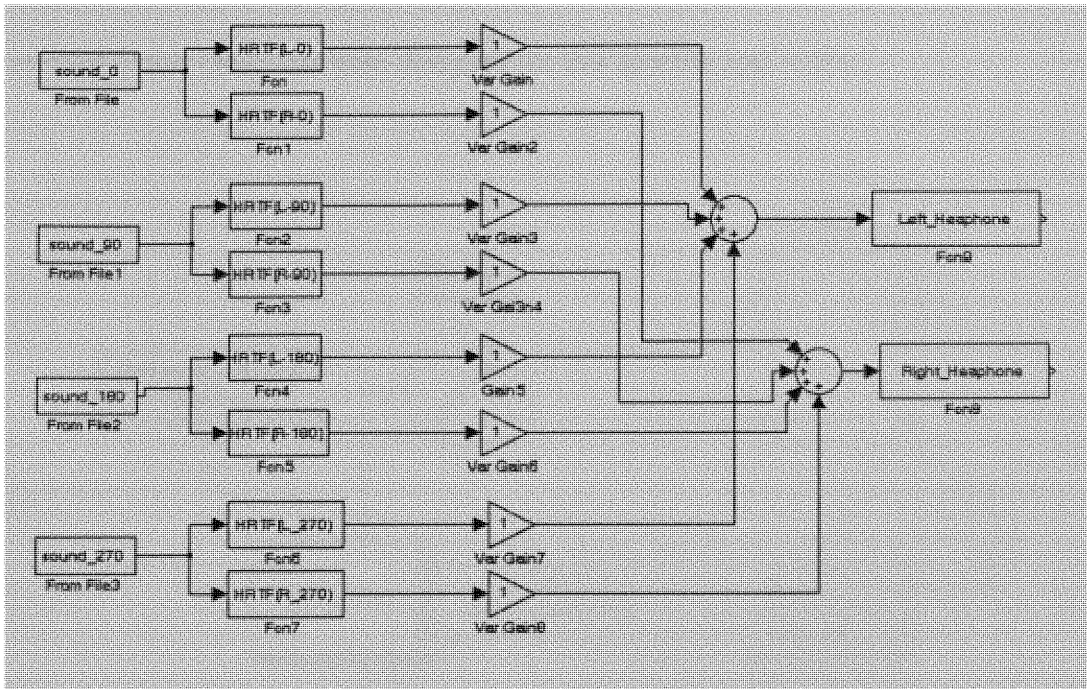


Figure 1-4. Functional diagram of an HRTF rendering engine

Experimental results show that the HRTF spatialization technique enables creation of virtual acoustic sources distinguishable with an angular resolution of approximately 30° in any location within the 360° span of the horizontal plane [26]. The laboratory has also been in the process of developing virtual acoustic environments using this technique [14]. This virtual acoustic environment generation technique could be integrated with a navigation system to develop a form of blind acoustic vision. Real time implementation of HRTF is fast and not computation intensive (Figure 1-4). This makes it an excellent method of presenting the physical world in a virtual acoustic form to a blind person in real time. Such a system would however require a “front-end”, which in practice is an acquisition system that measures distance of objects in the angular span of what would be

the user's frontal and peripheral vision. The sonar based range measurement system presented in this thesis has been designed for this specific application.

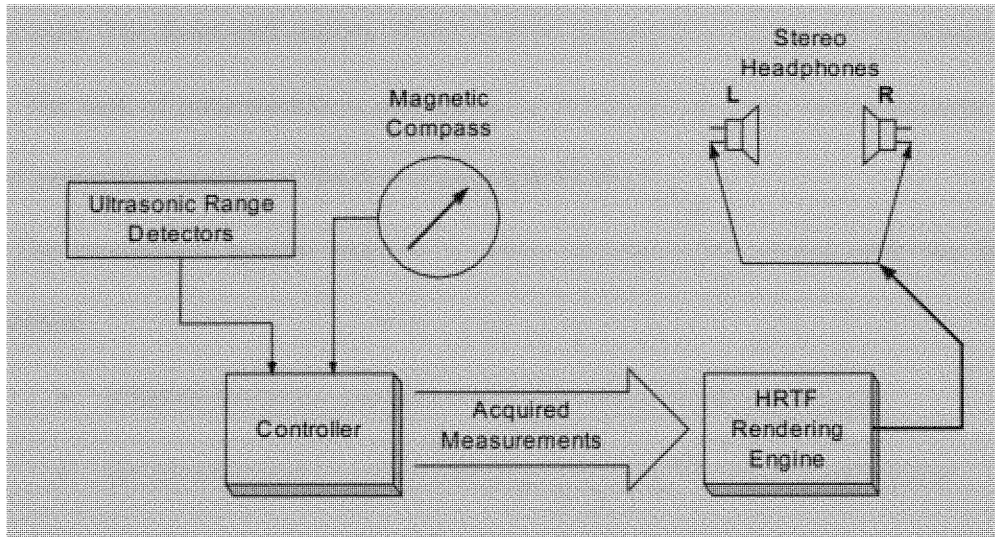


Figure 1-5. Components of the proposed ETA system

As illustrated in Figure 1-5, the proposed ETA employs multiple ultrasonic sensors for obstacle detection. The system also acquires the bearing of the user, from an electronically interfaced magnetic compass. The instrumentation components are interfaced to a controller that acquires the data and translates the raw data to a suitable form, which is then transmitted to an HRTF rendering engine. Based on the information gathered by the sensors, the HRTF rendering engine generates a virtual acoustic environment and presents the spatial sound to the user through stereo headphones. The ultrasonic sensors, magnetic direction sensor and the controller form the “front-end” of the ETA system, while the “back-end” includes the HRTF rendering engine and the stereo headphones.

1.4 Problem Specification

As explained in the previous section, an ETA performs two basic tasks – 1) gather information about the users physical environment, and 2) present this information to the blind user. The front-end section of the system proposed in Figure 1-5 performs the first task. The task of presentation is performed by the back-end. This thesis presents a sonar-based navigation system as the front-end of the ETA.

The navigation system requires a range measurement system with the following properties:

- it has to be easily integrated into the sound rendering process
- the measurement system should be suitable for indoor applications
- the system has to be small, light weight and low power in order to be portable
- the range measurements should have adequate consistency and resolution for faithful reproduction of the surrounding map in the spatialized sound environment
- any part of the instrument worn by the user should be ergonomic, comfortable and safe
- the design has to be cost effective so that it can be afforded by many of the potential users who could benefit from it

This thesis features the design, integration and testing of the front-end of the ETA system. The following sections of the thesis cover the design philosophy, architecture, component specifications, sensor interface, hardware integration, software and data

communication aspects of the front-end data acquisition system. Later sections discuss the test and evaluation procedures of the range measurement system.

2 SYSTEM OUTLINE

This chapter presents the overall outline of the front-end data acquisition system. Component specifications and features, standard specifications and underlying technology behind the sensor operations are presented in detail. Figure 2-1 shows how all the sensor and control components are inter-connected.

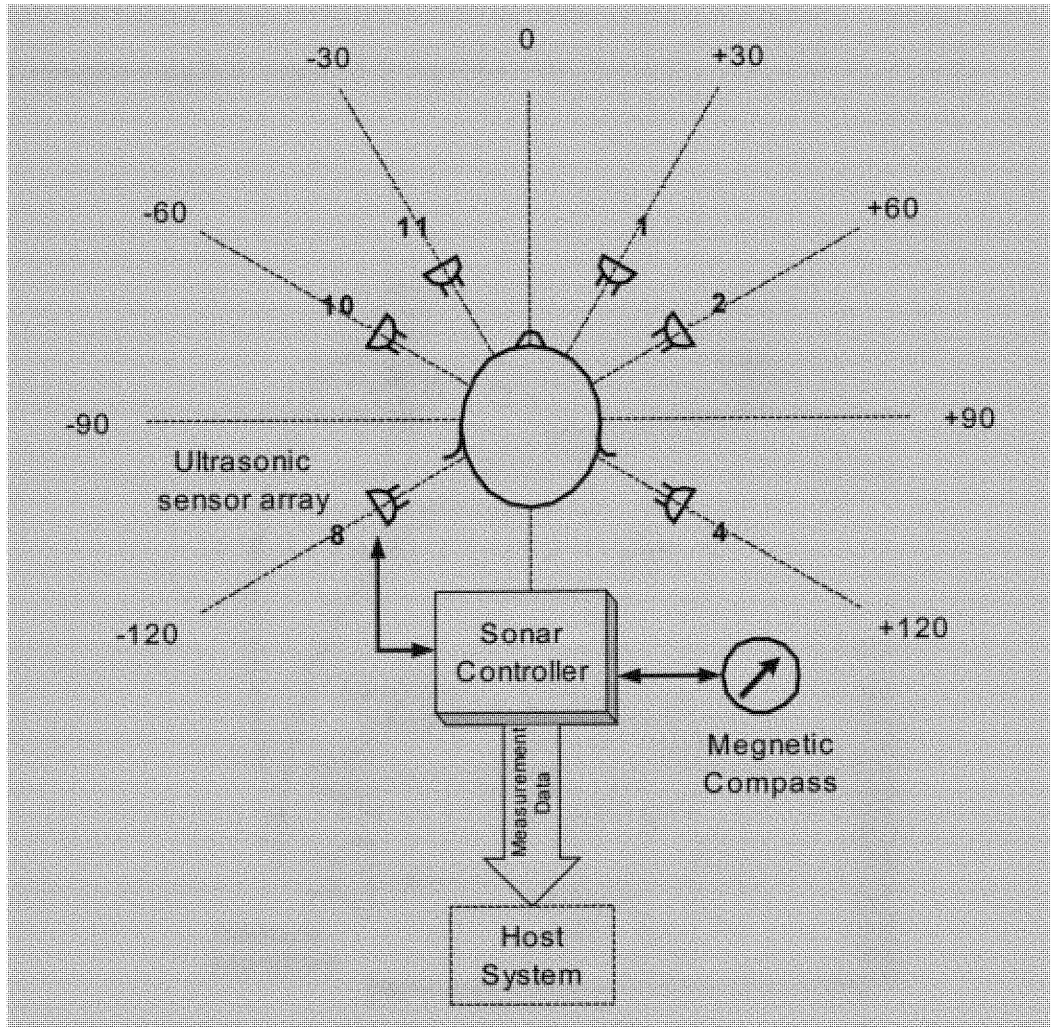


Figure 2-1. Schematic showing the arrangements of the front-end components

The Sonar Control Unit (SCU) lies at the center of the system hardware. This control unit is a microcontroller-based system that interfaces with the ultrasonic sensors, compass module and the host system.

2.1 Sonar Ranging Principle

The system employs ultrasonic sensors to measure the distance of obstacles. Sonar is an indirect method of distance measurement, which actually measures the travel time of ultrasonic signals. Figure 2-2 explains the principle behind sonar operation.

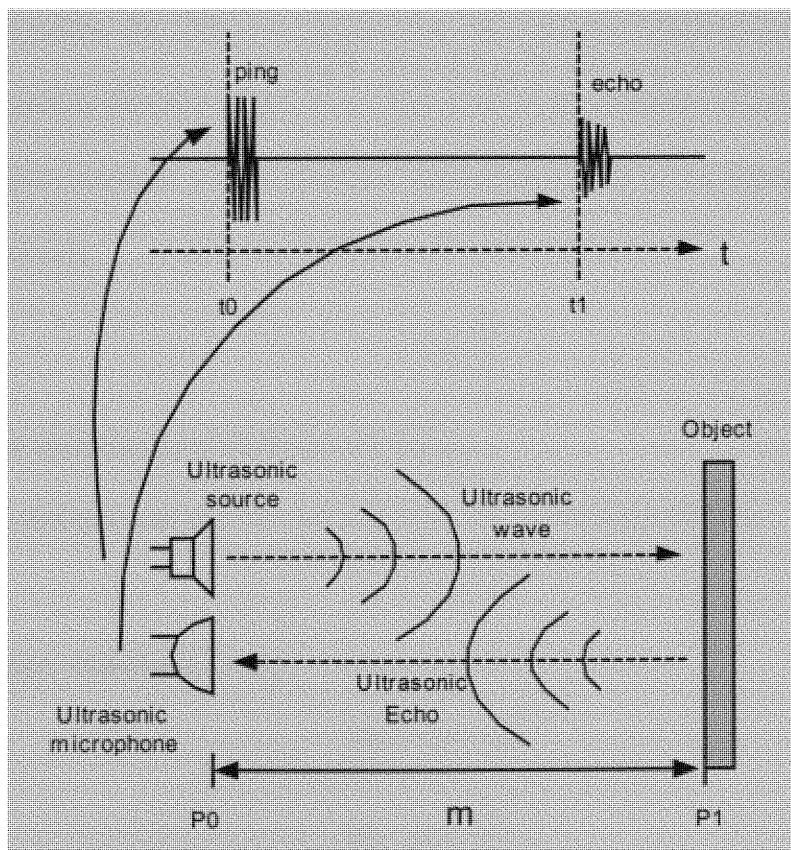


Figure 2-2 Sonar operation principle

Imagine the sonar located at position P_0 and the object is positioned at P_1 . The distance between the sonar and the object is $m = (P_0 - P_1)$. At time t_0 , the sonar emits a burst of ultrasonic waves. This process is known as *ping initiation*. The ultrasonic wave travels through the surrounding media and after traveling a distance m , approaches the object at position P_1 . The object absorbs a finite amount of the ultrasonic energy. The rest of the ultrasonic energy is reflected back towards point P_0 . This reflecting ultrasonic energy, known as the ultrasonic echo, propagates through the media and after traveling the distance m , reaches point P_0 . This echo is recorded on the ultrasonic microphone at time t_1 . The total distance traveled by the ultrasonic signal from ping origination to echo detection is therefore, $2m$. The time required for this complete cycle of events is, $\Delta t = (t_1 - t_0)$. Δt is known as the *echo return time*. The velocity of ultrasonic waves v , in open air, at 1 atmospheric pressure and an ambient temperature of 27°C is approximately 1138 ft/sec. By definition,

$$v = \frac{d}{t} \quad (2.1)$$

Where, d is the distance traveled within the time t .

Therefore, the distance can now be calculated by the following equation,

$$d = vt \quad (2.2)$$

From the scenario explained above,

$$2m = v(t_1 - t_0) \quad (2.3)$$

$$\text{or, } m = \frac{v\Delta t}{2} \quad (2.4)$$

From equation 2.4,

$$\Delta t = \frac{2m}{v} \quad (2.5)$$

The echo return time for an object at a distance of 1 inch can now be calculated by putting $m=1$ inch in equation 2.5. This value is calculated to be, $\Delta t_{inch} = 146\mu\text{S}$, approximately. Therefore, $146\mu\text{S}/\text{inch}$ can be used as a standard unit for echo return time to distance conversion. The distance measured by a sonar can now be calculated by,

$$m = \frac{\Delta t}{\Delta t_{inch}} \quad (\text{inches}) \quad (2.6)$$

This is a very important conclusion from an instrumentation point of view. Equation 2.6 suggests that, the distance in inches can be computed by a system, in reference to a time-base calibrated at Δt_{inch} .

2.2 Component Specification Requirements

The components used in the sonar system fall into three categories: 1) sensors, 2) control and computation, and 3) communications. During the preliminary stage of development, several alternative components were tested in each category, to identify their suitability for this application. The following issues were considered while selecting the components, 1) simplicity of design, 2) ease of integration, 3) configurability, 4) low power, 5) size, 6) detail in documentation, and 7) developer support. The components that were ultimately selected and integrated into the system are specified in the following sections.

2.2.1 Ultrasonic Sensor

Since the sonar system is designed for indoor range measurements, it requires the ultrasonic transducers to have certain characteristics. The illustration in Figure 2-1 shows the physical arrangement of the ultrasonic sensors, which suggests that the sensors ideally have a beam spread of not more than $\pm 15^\circ$. The physical dimensions and weight of the sensor modules are important because they need to be mounted on a headgear worn by the user. This also requires that the operating voltage be within a safe range. The ultrasonic power emitted by the transmitter needs to be comparatively low, with a high sensitivity receiver, to ensure elimination of multiple reflections from walls. It was determined that an accurate range reading of 10 feet would be adequate for a blind person to walk freely in an indoor environment. Therefore, the ultrasonic sensors require generating reliable range readings of obstacles within 10 feet while operating in the proposed array configuration of Figure 2-1.

2.2.2 Controller Platform

Figure 2-1 suggests a controller to interface with all the sensors and the host system. This central component of the multi sensor sonar system constitutes the *Sonar Control Unit* (SCU). The SCU is built on an embedded microcontroller platform. The specification requirements for this component are as follows: 1) computational platform fast enough to compute distance from the echo return times in real time, 2) low power consumption, 3) adequate numbers of configurable I/O lines, 4) serial communications capability, 5) some nature of standardized bus expansion feature, 6) detailed documentation and 7) developer support.

2.2.3 Devantech SRF04 Ultrasonic Ranger

The Devantech SRF04 sonar was chosen as the range finding sensor to be used because of its compact construction and comparatively narrow beam pattern. Table 2-1 shows the electrical and acoustic specification of this ultrasonic ranger module.

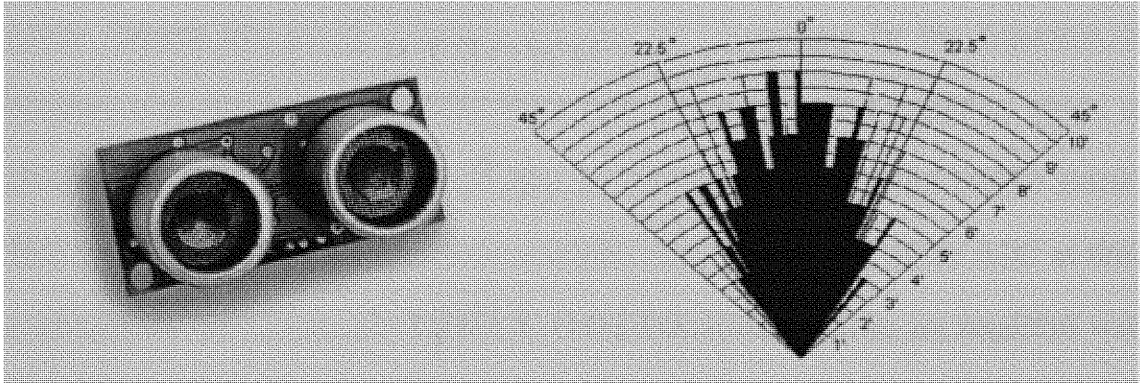


Figure 2-3. The Devantech SRF04 Ultrasonic Ranger module (left); and the Ultrasonic beam pattern of the transducers (right) [from www.acroname.com]

Table 2-1. Technical specifications of the Devantech SRF04 sonar

| | |
|---------------------|--|
| Operating voltage | 5V DC |
| Current consumption | 30 mA (Typ.), 50 mA (Max.) |
| Operating frequency | 40 KHz |
| Minimum range | 3 cm |
| Maximum range | 3 m |
| Sensitivity | Detects a 3cm diameter stick at > 2m |
| Input trigger | 10 μ S minimum TTL level pulse |
| Echo detect output | TTL high signal proportional to echo return time |
| Weight | 0.4 oz |
| Dimensions | 1.75" w x 0.625" h x 0.5" d |

As seen in Figure 2-3 (right), the Devantech module shows a beam spread of less than $\pm 22.5^\circ$ at a distance of 10 feet. The beam spread widens to about $\pm 30^\circ$ at about 5 feet. Even though this does meet the exact requirements stated in section 2.2.1, a technique has

been implemented (section 4.1.3) to compensate for the 7° beam overlap on neighboring sensors.

Figure 2-4 shows the functional components inside the Devantech SRF04 module. An embedded controller coordinates the ping initiation and echo detection process. The system features low power ultrasonic emission. To compensate for the dispersion of returning echo signals, the controller implements a gain controlled amplifier, which, in collaboration with the controller, acts like an *Automatic Gain Control* (AGC) circuit. After initiating the ping, the controller gradually increases the gain of the AGC as time progresses, so that fading echo signals traveling from distant objects are amplified well enough to be above the threshold of echo detection. The controller provides two control pins for external interface - a *PING* (trigger) input and an *ECHO* output.

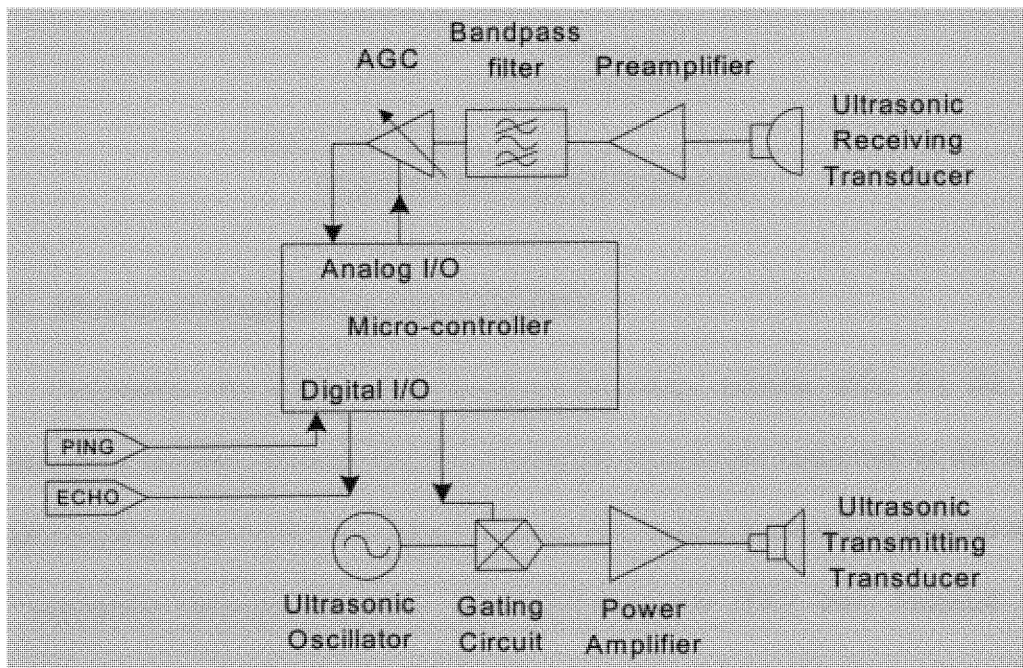


Figure 2-4. Functional Schematic of the Devantech SRF04 module

Figure 2-5 shows the handshake process of the SRF04 module with external circuitry. The PING input must to be triggered by a TTL high-level signal for a minimum period of $10\mu\text{S}$. At the falling edge of the trigger pulse, the sensor emits a burst of ultrasonic signal for $200\mu\text{S}$, and then pulls the ECHO output to TTL high-level. The ECHO pin remains high until the controller records the ultrasonic echo. If no echo is detected within 8mS , then a time-out occurs and the ECHO pin is returned back to logic 0. A waiting period of 10mS is required after the ECHO signal returns to logic 0, for the next ping to be initiated.

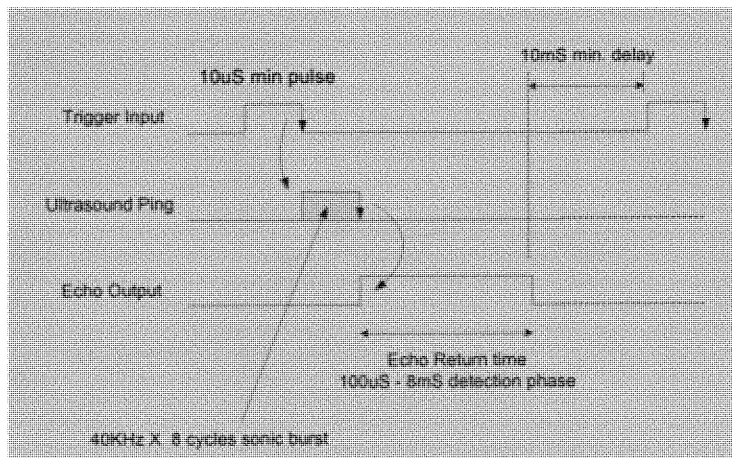


Figure 2-5 SRF04 handshake protocol

2.2.4 Devantech Magnetic Compass CMPS01

Originally, the system was designed for range sensing only. As the design progressed, it became obvious that the orientation is as much important as the sense of the surrounding environment to a blind traveler. The system architecture allowed trouble-free expansion through the Inter IC Communication (I^2C) bus, featured in the microcontroller used to implement the SCU (see section 2.2.5). The Devantech CMPS01 compass module, which is equipped with I^2C interface, was chosen for the application.

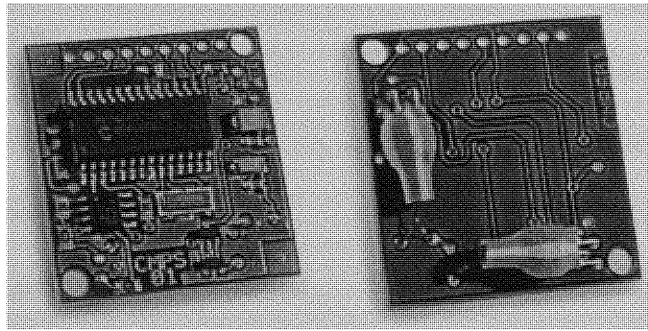


Figure 2-6. The Devantech CMPS01 Digital Compass Module, top view (left), and bottom view (right) [from www.acroname.com]

Table 2-2. The Devantech CMPS01 specification

| | |
|---------------------|------------------------------------|
| Operating voltage | 5V DC |
| Current consumption | 20 mA (Typ.) |
| Resolution | 0.1° |
| Accuracy | 3° |
| Maximum range | 3 m |
| Interface | I ² C or Analog PWM |
| Data format | 8 bit (0-255) or 16 bit (0 – 3599) |
| Bus speed | 1 MHz (max) |
| Weight | 0.3 oz |
| Dimensions | 32mm x 35mm |

Table 2-3. CMPS01 Register assignment

| Register | Function |
|----------|--|
| 0 | Software Revision Number |
| 1 | Compass Bearing as a byte, i.e. 0-255 for a full circle |
| 2,3 | Compass Bearing as a word, i.e. 0-3599 for a full circle, representing 0-359.9 degrees |
| 4,5 | Internal Test - Sensor1 difference signal - 16 bit signed word |
| 6,7 | Internal Test – Sensor2 difference signal - 16 bit signed word |
| 8,9 | Internal Test - Calibration value 1 - 16 bit signed word |
| 10,11 | Internal Test - Calibration value 2 - 16 bit signed word |
| 14 | Calibration Done Flag |
| 15 | Calibrate Command |

The compass uses the Philips KMZ10A [15] magnetic field sensor, which is sensitive enough to detect the Earth's magnetic field. Two of them mounted at right angles (Figure 2-6, right) to each other can be used to compute the direction of the horizontal component of the Earth's magnetic field. The onboard micro-controller measures the ratio of Hall effect [16] voltage generated by the projection of the Earth's magnetic flux on the two magnetic sensors. The controller then computes the direction of the earth's magnetic field. The I²C node address of the compass is 0xC0. Table 2-2 shows the specification of the compass module. The Register assignment is shown in Table 2-3. The compass measurement was read from register 1 for this application.

2.2.5 Object Oriented Programmable Integrated Circuit (OOPIC)

The micro-controller selected for the SCU is the OOPIC-II™ board (Figure 2-7), manufactured by Savage Innovations Inc. [17]. The OOPIC-II is a PIC16F77 [18] based micro-controller board with a resident operating system.

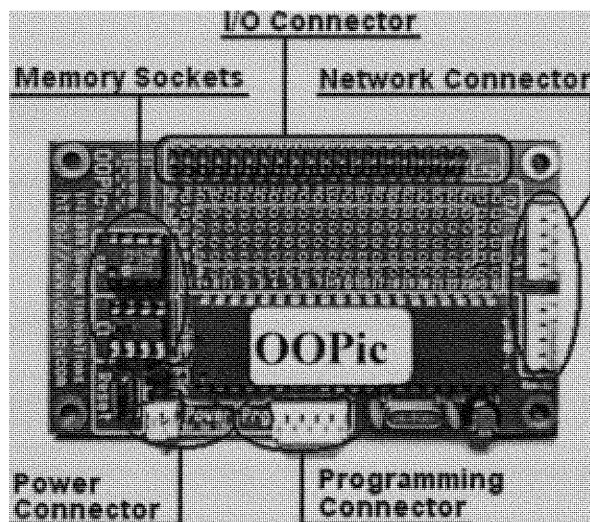


Figure 2-7. Photograph of the OOPIC™ board showing its different features [from www.oopic.com]

Figure 2-8 shows the architecture of the PIC16F77 micro-controller. The PIC16F77 resources are controlled by the resident operating system. These resources are presented to the programmer as objects. The OOPIC system offers a choice of two programming languages – 1) Visual Basic and 2) Java. While programming, each hardware resource is initiated by setting up the properties and interconnecting the objects. During runtime the hardware resources can be read as well as be manipulated by and setting the object properties and methods. The resident operating system provides the service of continuously updating all object properties during run-time. The useful features of the OOPIC board are summarized below.

- Multitasking Operating System
- Object Oriented Programming
- 2000 Instructions / second execution speed
- 4KB EEPROM, upgradeable to 32KB
- 31 individually configurable I/O pins – programmable to function as bit, byte or word
- 32 bit programmable timer
- Synchronous/Asynchronous serial communication module
- I²C bus Interface

A 40 pin IDE connector (Figure 1-1) on the OOPIC board provides electrical connections to external device. All configurable I/O lines, power, ground, serial and I²C interface are accessible through this connector. The complete pin configuration and pin descriptions are displayed in Table 2-4.

Table 2-4. Pin configuration of the OOPIC 40 pin IDE connector

| Pin | Name | Dir | Function | Pin | Name | Dir | Function |
|-----|--------|-----|---|-----|--------|-----|----------------------------------|
| 1 | LSDA | ↔ | Local I2C Serial Data | 2 | GND | — | Ground |
| 3 | LSCL | → | Local I2C Serial Clock | 4 | Power | — | +6v to +18v Power Supply |
| 5 | Reset | ↔ | Reset (Active Low) | 6 | I/O 15 | ↔ | I/O Line 15 I/O Group 1 bit 7 |
| 7 | I/O 1 | ↔ | I/O Line 1 A2D 1 | 8 | I/O 14 | ↔ | I/O Line 14 I/O Group 1 bit 6 |
| 9 | I/O 2 | ↔ | I/O Line 2 A2D 2 | 10 | I/O 13 | ↔ | I/O Line 13 I/O Group 1 bit 5 |
| 11 | I/O 3 | ↔ | I/O Line 3 A2D 3 | 12 | I/O 12 | ↔ | I/O Line 12 I/O Group 1 bit 4 |
| 13 | I/O 4 | ↔ | I/O Line 4 A2D 4 | 14 | I/O 11 | ↔ | I/O Line 11 I/O Group 1 bit 3 |
| 15 | I/O 5 | ↔ | I/O Line 5 | 16 | I/O 10 | ↔ | I/O Line 10 I/O Group 1 bit 2 |
| 17 | I/O 6 | ↔ | I/O Line 6 | 18 | I/O 9 | ↔ | I/O Line 9 I/O Group 1 bit 1 |
| 19 | I/O 7 | ↔ | I/O Line 7 | 20 | I/O 8 | ↔ | I/O Line 8 I/O Group 1 bit 0 |
| 21 | +5v | — | + 5 Volts regulated | 22 | +5v | — | + 5 Volts regulated |
| 23 | GND | — | GND | 24 | GND | — | GND |
| 25 | I/O 16 | ↔ | I/O Line 16 I/O Group 2 bit 0 Timer 1 Xtal | 26 | I/O 31 | ↔ | I/O Line 31 I/O Group 3 bit 7 |
| 27 | I/O 17 | ↔ | I/O Line 17 I/O Group 2 bit 1 Timer 1 Xtal PWM-2 | 28 | I/O 30 | ↔ | I/O Line 30 I/O Group 3 bit 6 |
| 29 | I/O 18 | ↔ | I/O Line 18 I/O Group 2 bit 2 PWM-1 | 30 | I/O 29 | ↔ | I/O Line 29 I/O Group 3 bit 5 |
| 31 | I/O 19 | ↔ | I/O Line 19 I/O Group 2 bit 3 I2C Serial Clock | 32 | I/O 28 | ↔ | I/O Line 28 I/O Group 3 bit 4 |
| 33 | I/O 20 | ↔ | I/O Line 20 I/O Group 2 bit 4 I2C Serial Data | 34 | I/O 27 | ↔ | I/O Line 27 I/O Group 3 bit 3 |
| 35 | I/O 21 | ↔ | I/O Line 21 I/O Group 2 bit 5 | 36 | I/O 26 | ↔ | I/O Line 26 I/O Group 3 bit 2 |
| 37 | I/O 22 | ↔ | I/O Line 22 I/O Group 2 bit 6 Serial Transmit | 38 | I/O 25 | ↔ | I/O Line 25 I/O Group 3 bit 1 |
| 39 | I/O 23 | ↔ | I/O Line 23 I/O Group 2 bit 7 Serial Receive | 40 | I/O 24 | ↔ | I/O Line 24 I/O Group 3 bit 0 |

2.2.6 Counter Registers

The counter registers play an important role in the sonar interface circuit. These components register the echo return event with comparison to the reference clock.

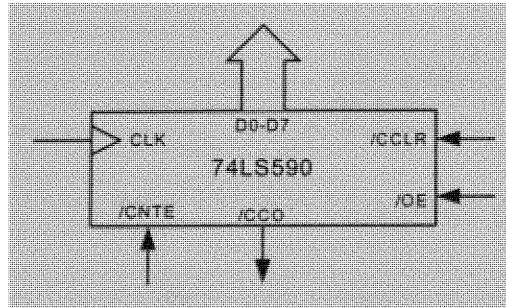


Figure 2-9 The 74LS590 8 bit counter register

The TTL family 74LS590, 8-bit up counter registers have been integrated into the design of the SCU. Six counters are used to interface the OOPIC with the six ultrasonic sensors. The storage register has an 8 bit parallel output. The /OE pin is a negative logic output enable control pin for the data bits. The counter clock is positive edge triggered. The /CCO pin goes to logic zero when the count value reaches 255. The count operation is disabled when the /CNTEN pin is held low. Pulling /CCLR low clears the count value of the counter.

2.3 Mechanical Layout of the Sensor Array

In the navigation system, the six sonar sensors are configured in an array that allows each sensor to take measurements in a certain angle. The range sensors are mounted on a visor and worn by the user like a headgear (Figure 2-11). Each sensor is directed toward a certain angle, as illustrated in Figure 2-10. These angles are labeled according to matching clock positions. These clock position labels will be used to refer to

the six sonar directions in the remainder of this thesis. The angular spacing between the 10 and 11 o'clock sonar pairs and 1 and 2 o'clock sonar pairs is 30° . On the other hand the angular spacing between 8 and 10 o'clock; 11 and 1 o'clock; and 2 and 4 o'clock sonar pairs is 60° .

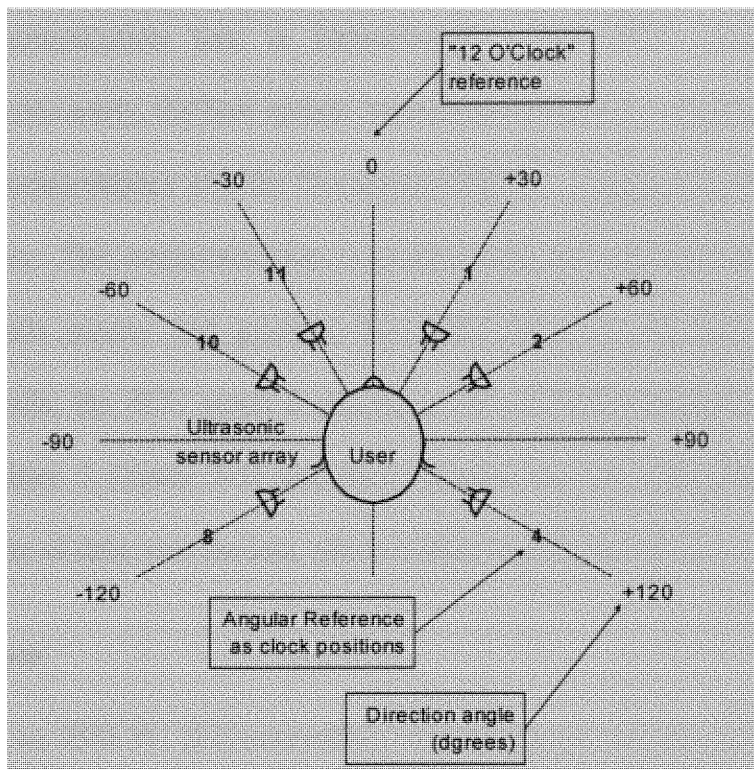


Figure 2-10. Ultrasonic sensor mounting directions and frame of reference

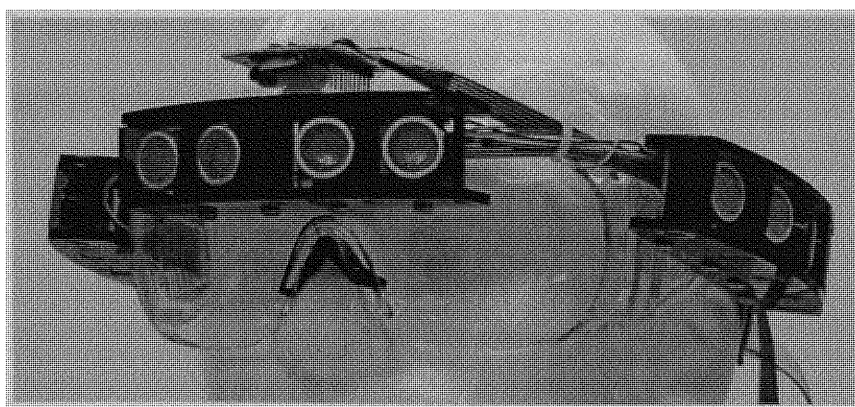


Figure 2-11. Photograph of the six ultrasonic sensors mounted on the headgear; Notice the compass module mounted on top of the center sonar housing

2.4 Data Communications

Data communications plays a significant role in the operation of the navigation system. The SCU is equipped with two different types of data communication interfaces. The SCU communicates with the host system through an RS232 [19] asynchronous serial link. This communication link is bidirectional. The SCU transmits acquired measurements; conversely, the host system commands to the SCU using this RS232 link. The SCU also communicates with the Devantech CMPS01 compass module using an I²C [20] bus interface. This communication is synchronous and bidirectional. The SCU is configured as bus master and governs the traffic flow on the bus. During every measurement cycle, the SCU sends a request to the CMPS01 module to transmit the contents of register 1 (Table 2-3) and then releases the bus. The CMPS01 module transmits one byte of data over the bus, which is then read by the SCU.

2.5 Host System

The host is an IBM PC [21] compatible microcomputer system. A Matlab TM [22] script runs the host system that commands the SCU to take measurements, receives the measurement data from the SCU and then further processes the data for presentation purposes. Further details about the software running on the host system can be found in chapter 4.

3 SCU HARDWARE

This chapter presents detail of the hardware design. Because the SCU is the central component of the system, most of the information concerns the interfacing of different components with the OOPIC board. Details of the data communication hardware and device interconnection are also covered in this chapter.

3.1 System Architecture

The SCU performs the following tasks; 1) receive and decode command set transmitted from the host system; 2) control six sonar transceiver modules, triggering them in a prescribed sequence; 3) record the echo return obtained in each sonar module; 4) calculate object distance from the time delay; 5) request reading from the compass module 6) receive compass reading via I²C bus; 7) store all measurement data in a defined sequence; 8) apply an error detection mechanism on the data, to ensure its validity, and form a data transmission packet; 9) serially transmit the range data to the PC via a serial link (COM port).

3.2 Instrumental Technique for Registering Sonar Range Readings

As described in section 2.1, the fundamental approach of echo detection based range measurement has been modified for instrumental techniques. Equation 2.6 summarizes the technique that can be adopted in hardware implementation. Figure 3-1 shows a simplified diagram of such a hardware implementation. A binary up counter is driven by a clock circuit with a period, $T = \Delta t_{inch}$ (section 2.1). The *Timer Gating* circuit

activates the count enable pin, when a ping is triggered and deactivates it when an echo is received as illustrated in the timing diagram.

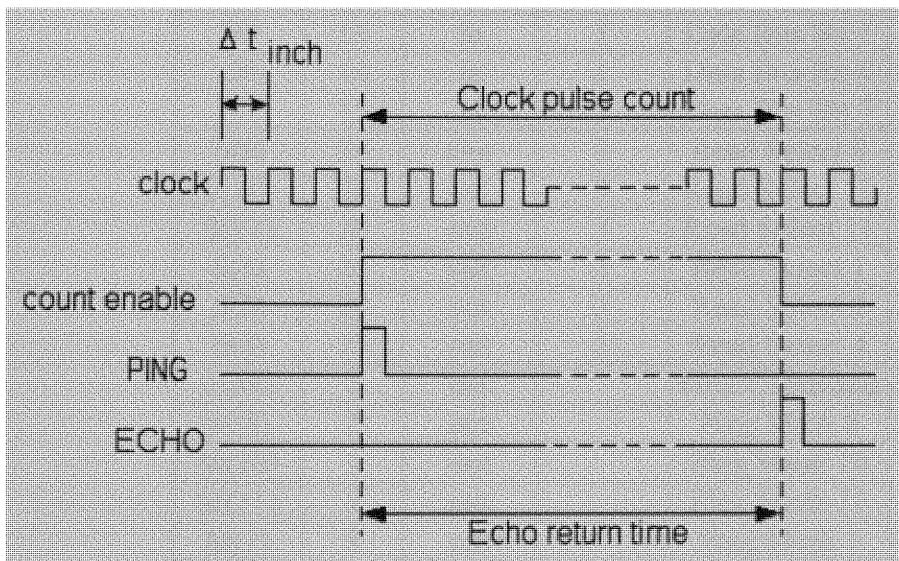
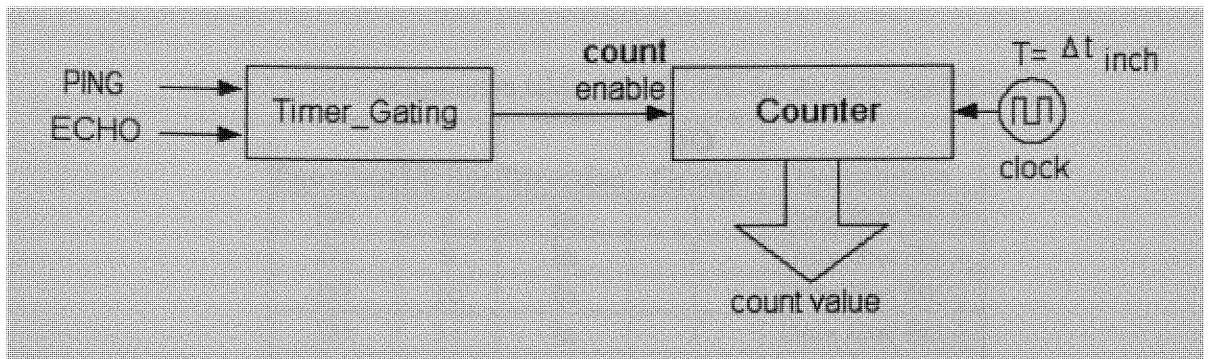


Figure 3-1. Hardware Implementation for Sonar Range Measurement Applying Equation 2.6 (top) ; and the timing diagram (bottom)

The count value registered on the counter is the number of clock periods within the window of the echo return time. Since each clock period represents the echo return time for one inch, the count value actually records the echo return path measured in inches. Therefore a reference clock frequency of $1/ \Delta t_{inch} = 1/146\mu S = 6.85 \text{ KHz}$ can be used with a counter-based recording system for directly converting the echo detection time into distance. The advantage of such an approach is obviously the direct conversion,

which does not require any extra computational effort. The approach described in Figure 3-1 has been implemented in the range measurement system with minor modifications.

Table 3-1. I/O Configuration table of the OOPIC for the SCU application

| Function | I/O line | Direction | IO Group | Description |
|-----------------|-----------------|------------------|-----------------|--------------------|
| D0 | 8 | In | 1 | Data Bus Bit 0 |
| D1 | 9 | In | 1 | Data Bus Bit 1 |
| D2 | 10 | In | 1 | Data Bus Bit 2 |
| D3 | 11 | In | 1 | Data Bus Bit 3 |
| D4 | 12 | In | 1 | Data Bus Bit 4 |
| D5 | 13 | In | 1 | Data Bus Bit 5 |
| D6 | 14 | In | 1 | Data Bus Bit 6 |
| D7 | 15 | In | 1 | Data Bus Bit 7 |
| S(8) | 25 | Out | 3 | Read Select 8 |
| S(10) | 24 | out | 3 | Read Select 10 |
| S(11) | 26 | out | 3 | Read Select 11 |
| S(1) | 28 | out | 3 | Read Select 1 |
| S(2) | 29 | out | 3 | Read Select 2 |
| S(4) | 27 | out | 3 | Read Select 4 |
| PING | 7 | out | | Ping Command |
| M(8) | 2 | out | | Ping Mask 8 |
| M(10) | 1 | out | | Ping Mask 10 |
| M(11) | 3 | out | | Ping Mask 11 |
| M(1) | 5 | out | | Ping Mask 1 |
| M(2) | 6 | out | | Ping Mask 2 |
| M(4) | 4 | out | | Ping Mask 4 |
| LSDA | | In/Out | | I2C Serial Data |
| LSCL | | Out | | I2C Serial Clock |
| Tx | 22 | out | | Serial Tx |
| Rx | 23 | in | | Serial Rx |

3.3 OOPIC I/O Configuration

The OOPIC is equipped with a highly configurable I/O subsystem. The I/O lines can be configured as individual bits or may be grouped as 8-bit or 16-bit groups. The

input or output mode of each pin is modifiable during runtime. This makes the OOPIC a highly versatile component for peripheral interfacing. Table 2-4 shows the complete description of all configurable I/O pins available for external interface. These I/O pins have been assigned for specific interconnections in the SCU application, as shown in Table 3-1.

3.4 Ultrasonic Sensor Interface

Figure 3-2 shows the schematic diagram of the sonar interface circuits. U1 to U6 are 74LS590, 8 bit binary counter registers (Section 2.2.6). Each counter interfaces one channel of the ultrasonic range sensors to the OOPIC. The count value recorded in these counters are actually the range values detected by the ultrasonic sensors. The process of echo return time to range conversion has been detailed in sections 2.1 and 3.2.

The IO lines 8 through 15 have been configured as an 8-bit *DATA* bus for the OOPIC, in the read (input) mode. The three state outputs of U1 to U6 are connected to this *DATA* bus. The *Register Select* pins $S(n)$ connect to the matching channel register's */OE* pins. The select pins are normally logic high. When the register value of a channel is read, the select line for that channel is set to logic low (Figure 3-4, segment 3 to 5). The ping command output from the OOPIC is inverted and fed in parallel to the */CCLR* input of all the counters. As illustrated in Figure 3-3, segment 1-2 and 5-6, while the PING signal is at high level the counters are cleared. The */CCLR* returns to high level as soon as the PING command is set to logic low.

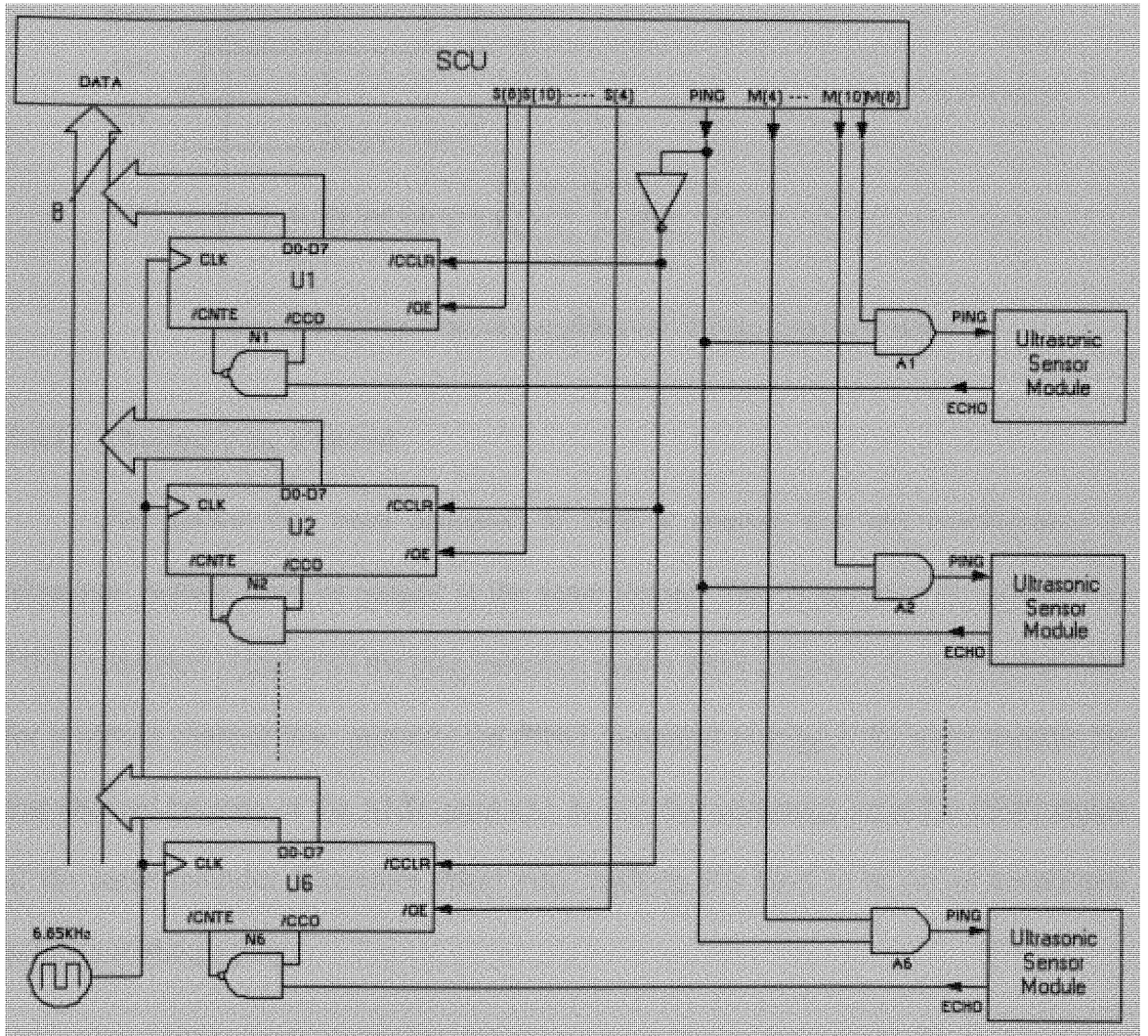


Figure 3-2. Schematic Diagram of the OOPIC sonar interface circuit

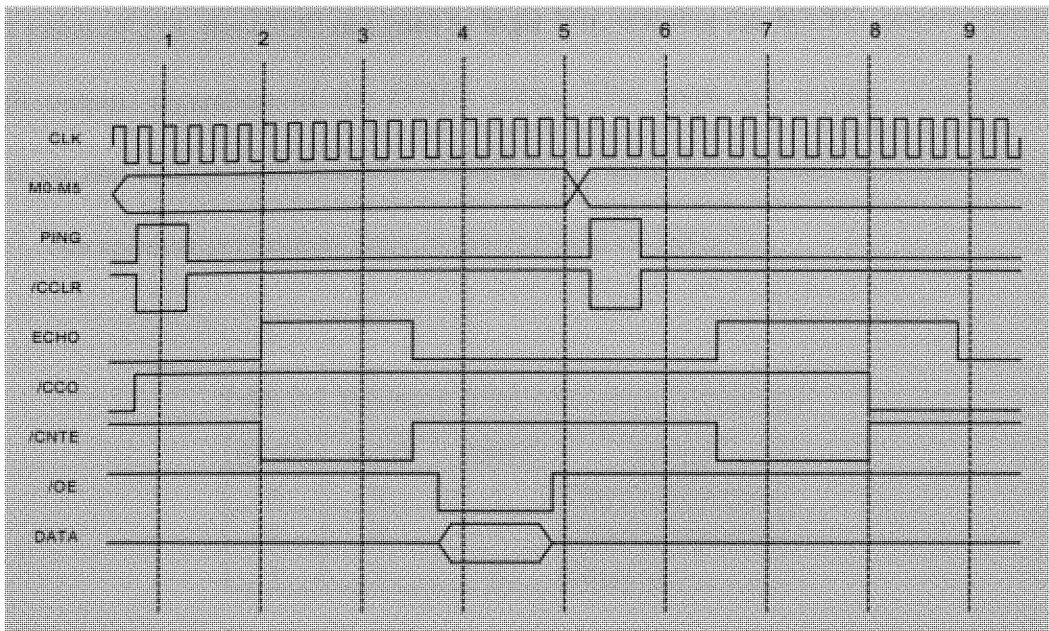


Figure 3-3. timing diagram explaining the OOPIC sonar interface

3.4.1 Channel Ping Control

The system has been designed in such a manner that any combination of ultrasonic sensors can be pinged at the same instant. This is possible by masking the PING command signal with the ping selection matrix $M(n)$ (Figure 3-2). Setting the $M(n)$ for a certain channel low, prevents the PING command to trigger the sonar module for that channel because of the AND gates (A1 to A6) that ultimately drive the PING inputs of the sonar modules. Therefore, $M(n)$ ping masking bits have to be set as intended before initiating a ping.

3.4.2 Counter Rollover Prevention

The ECHO output from each sonar module is gated through corresponding NAND gates (N1 to N6) before activating the /CNTE (count enable) pins of the associated counter. As illustrated in Figure 3-3, when the counter reaches its maximum

count value of 255, the /CCO out goes to logic low, causing the NAND gates to generate a high output, regardless of the ECHO status. This stops the counter from counting further on and rolling back to zeros. Therefore, a count value of 255 indicates an “over-range”. The rollover prevention logic is summarized in Table 3-2.

Table 3-2. Summary of the rollover prevention logic

| ECHO | /CCO | /CNTE | Action |
|------|------|-------|-----------------|
| X | 0 | 1 | Count suspended |
| 0 | X | 1 | Count suspended |
| 1 | 1 | 0 | Count Up |

3.5 Compass Module Interface

The OOPIC communicates with the compass module using the I²C interface on pins 1 and 3 of the 40-pin IDE connector. The OOPIC performs as the local bus master and provides the serial clock on the LSCL line. The serial data line is pulled up using a 3.3K Ω resistor [23].

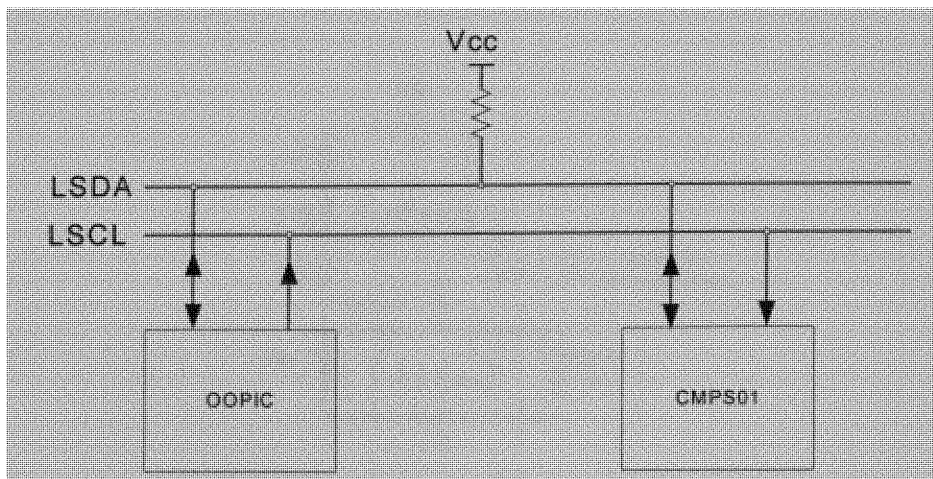


Figure 3-4. Interface between the OOPIC and the Compass module

3.6 Serial Communications Interface

The OOPIC is equipped with a serial communications unit, which is used in this application to communicate with the host PC RS232 interface. However the serial IO lines on the OOPIC (Table 2-4) are TTL compatible. Therefore, a Maxim 3223, TTL-RS232 level interface IC has been used [24]. The serial communications parameters are set to 8 data bits, no parity, 1 stop bit with asynchronous NRZ signaling. The serial link uses software flow control with fixed buffer size. The receive buffer on the SCU port is 2 bytes and on the host side is 9 bytes. This means that, when the serial port receiver is filled up with the assigned number of data bytes, the operating system is notified and no more data is accepted until the receive buffers are read and cleared out.

3.7 Reference Clock

A circuit generates a local clock for the counters as a reference to convert the echo delay time to inches. The clock is calibrated such that the clock period is Δt_{inch} or $146\mu\text{S}$. A 555 timer circuit (illustrated in Figure 3-5) has been designed in the astable mode to generate this clock frequency.

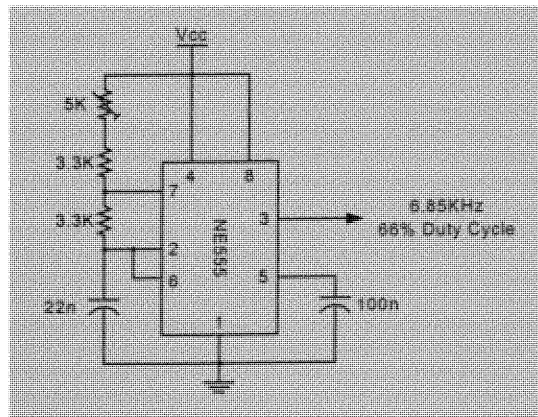


Figure 3-5. Reference clock circuit for the Sonar Echo Return Counters

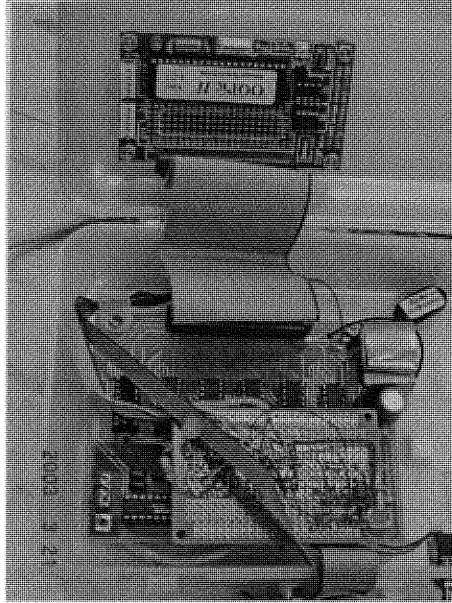


Figure 3-6. Photograph showing the SCU Circuit construction and Housing

3.8 SCU Construction

The circuitry required for the sonar interfacing and RS232 compatibility, was built on a printed circuit board (Figure 3-6). All connections between the OOPIC board and additional logic circuits are made through the 40-pin IDE connector. The SCU is housed in a plastic casing. A rechargeable 9.6V Ni-MH battery powers the SCU.

4 SOFTWARE

The system involves software development on multiple platforms. The software developed for the OOPIC board executes the sonar control protocol, communicates with the Devantech compass module and the host PC. The software on the Host PC commands the SCU, receives measurement values through serial communication and presents the measurement to the user.

4.1 Pinging Schemes

The SCU features complete autonomy between the six sonar channels in the echo delay and range recording process. This enables the system to trigger any combination of the six sonar modules at any instance. Several types of pinging formations for the sonar sensor have been tested, to identify the most effective method of acquiring a panoramic ultrasonic snapshot of the surrounding environment. The I/O activity during a ping operation was summarized in Figure 3-3.

4.1.1 Sweep Mode

The *Sweep Mode* pings each sonar module at a time, starting from the 8 o'clock position moving clockwise to the 4 o'clock position. A 50 millisecond timing gap is maintained between co-channel ping pulses. The pseudo code for sweep operation is as follows.

```
SET PING MASK BITS TO DEACTIVET ALL CHANNELS
FOR SONAR DIRECTIONS (8, 10, 11, 1, 2, 4)
    SET PING MASK BITS TO ACTIVATE CORRESPONDING CHANNEL
    ACTIVATE PING SIGNAL
    DEACTIVATE PING SIGNAL
    SET REGISTER SELECT BITS FOR CORRESPONDING CHANNEL
```

```
SET BUFFER POINTER TO CORRESPONDING LOCATION
READ DATA BUS CONTENT INTO BUFFER LOCATION
CLEAR REGISTER SELECT BITS
NEXT SONAR DIRECTION
```

The host system acquires an ultrasonic panoramic snapshot of the environment every 740 milliseconds.

4.1.2 Flood Mode

The *Flood Mode* triggers all six ultrasonic sensors at the same instance. Once all the six ranges have been registered, the SCU reads each register into the associated buffer position as illustrated by the pseudo code below.

```
SET PING MASK BITS TO ACTIVATE ALL CHANNELS
ACTIVATE PING SIGNAL
DEACTIVATE PING SIGNAL
FOR SONAR CHANNEL (8, 10, 11, 1, 2, 4)
    SET REGISTER SELECT BIT FOR CORRESPONDING CHANNEL
    SET BUFFER POINTER TO CORRESPONDING LOCATION
    READ DATA BUS CONTENT INTO BUFFER LOCATION
    CLEAR REGISTER SELECT BIT
NEXT CHANNEL
```

The host system acquires a complete set of six readings in 390 milliseconds.

4.1.3 Interlaced Mode

The *Interlaced Mode* operation is a hybrid between the sweep and flood mode. Sensors in alternating directions are triggered in two groups. During the first trigger, channels 8, 11 and 2 are triggered and the recorded. The second trigger registers channels 10, 1 and 4. The pseudo code below explains the SCU operation in this mode.

```

SET PING MASK BITS TO ACTIVATE CHANNEL (8, 11, 2)
ACTIVATE PING SIGNAL
DEACTIVATE PING SIGNAL
FOR SONAR CHANNEL (8, 11, 2)
    SET REGISTER SELECT BIT FOR CORRESPONDING CHANNEL
    SET BUFFER POINTER TO CORRESPONDING LOCATION
    READ DATA BUS CONTENT INTO BUFFER LOCATION
    CLEAR REGISTER SELECT BIT
NEXT CHANNEL
SET PING MASK BITS TO ACTIVATE CHANNEL (10, 1, 4)
ACTIVATE PING SIGNAL
DEACTIVATE PING SIGNAL
FOR SONAR CHANNEL (10, 1, 4)
    SET REGISTER SELECT BIT FOR CORRESPONDING CHANNEL
    SET BUFFER POINTER TO CORRESPONDING LOCATION
    SEAD DATA BUS CONTENT INTO BUFFER LOCATION
    CLEAR REGISTER SELECT BIT
NEXT CHANNEL

```

The *Interlaced Mode* enables the host system to acquire a complete panoramic range reading every 445 milliseconds.

4.1.4 Register Clear

The *Register Clear* is a special function implemented in the SCU for diagnostic purposes. The operations are similar to a ping operation, except that all sonar channels are deactivated using the ping mask bits. As a result, during the high logic level of the ping signal (Figure 3-3), all registers are cleared. All register values then are sequentially loaded into the buffer, which clears the buffers.

```

SET PING MASK BITS TO DEACTIVATE ALL CHANNELS
ACTIVATE PING SIGNAL
DEACTIVATE PING SIGNAL
FOR SONAR CHANNEL (8, 10, 11, 1, 2, 4)
    SET REGISTER SELECT BIT FOR CORRESPONDING CHANNEL
    SET BUFFER POINTER TO CORRESPONDING LOCATION
    READ DATA BUS CONTENT INTO BUFFER LOCATION
    CLEAR REGISTER SELECT BIT
NEXT CHANNEL

```

4.2 SCU Program Flow

The SCU and the host systems operate in a sequence of command and response. At power up, the SCU initializes the I/O, memory, data buffers, and communications interfaces and waits for a command packet from the host. Once the serial port receive buffer is full, the SCU decodes the command byte and copies it to a pre-assigned position in the transmit buffer, then jumps to the commanded ping mode subroutine. The appropriate range readings are loaded in the pre assigned locations of the transmit buffer. The magnetic compass reading is queried through the I²C interface and loaded in the appropriate position in the transmit buffer. At this point the 1 byte *Query ID* (section 4.4) that it had received in the command packet, is copied to the first position of the buffer, completing the formation of the data packet, which is loaded into the serial port FIFO and asynchronously transmitted to the host PC. The sequence of event is illustrated in the flowchart for the SCU software shown in Figure 4-1.

4.3 Host Program Flow

A Matlab™ based program runs on the host, which controls the range measurement operation on the SCU and provides a user interface to present the acquired ranging and bearing information. On execution, the Matlab script initializes the serial port of the PC with appropriate parameters. The receive and transmit buffers are initialized with nine and two bytes respectively. The program queries the user for the intended ping mode. A one-byte command (Table 4-2) is placed in a pre-assigned position in the command packet (Table 4-1) The program generates an unsigned, one byte random number as a

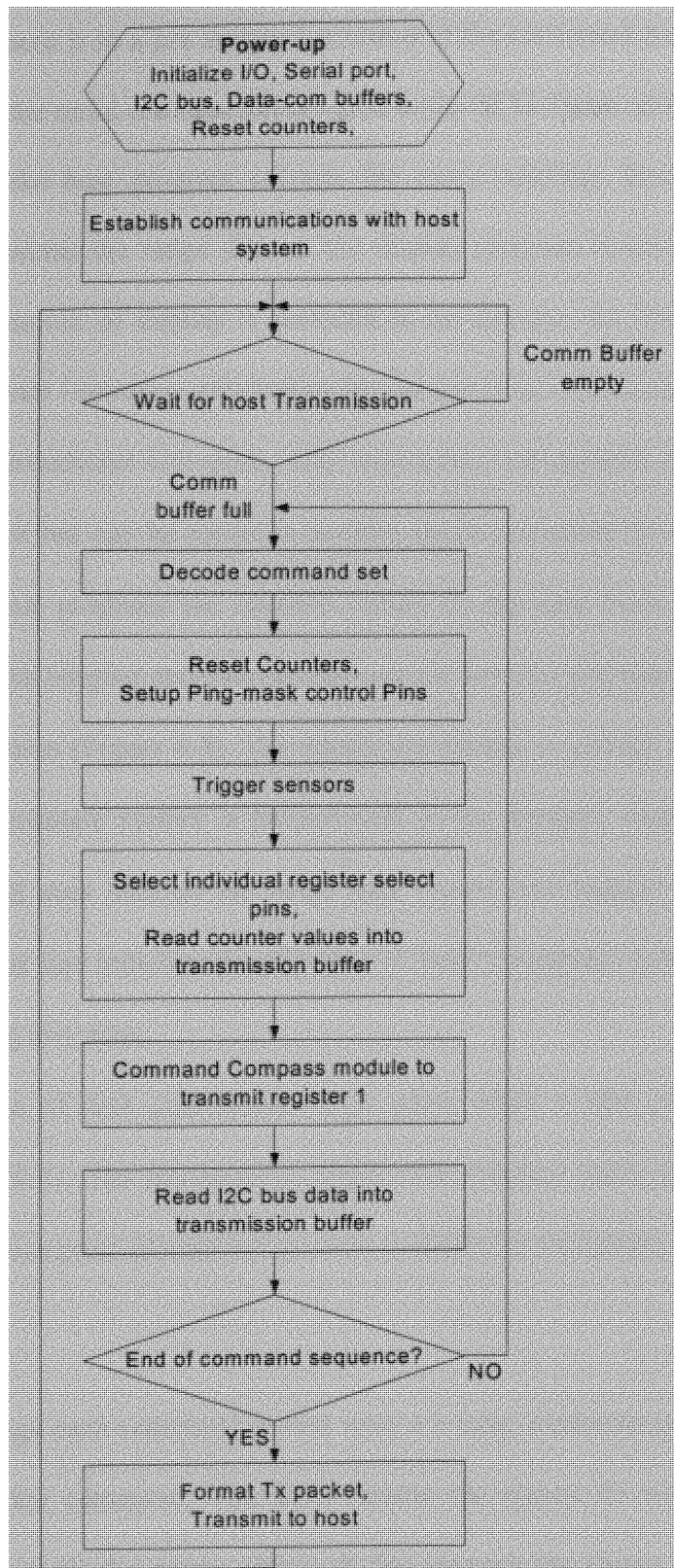


Figure 4-1. Program Flowchart of the SCU

Query ID and places it in a pre assigned position in the command packet. The packet is transmitted over the serial communications port to the SCU. Upon receiving a data packet from the SCU, the host program extracts the range readings from the pre assigned positions and presents the range data to the display program module. As explained in section 1.3, the nature of display is intended to be acoustic for blind navigation. The delivery of the sonar / compass information by the use of 3D virtual sounds, is the project of the Florida International University DSP laboratory. Information about that project is not discussed here but can be found in another thesis generated in our laboratory [14]. For debug and test purposes the numeric values of the data have been evaluated. In addition to this, the ranging and bearing information are also presented in a polar plot, in a ‘radar’ like fashion (Figure 4-3). The coordinate system is similar to Figure 2-10. The radius of the coordinate system is proportional to the distance measured by the sonar direction. The red stars indicate the range data, closed with dashed lines for improved visibility. The blue line indicates the direction of earth’s magnetic north with reference to the user’s heading direction.

The process of data presentation takes place while a new set of data is being acquired by the SCU. The autonomous ‘fetch while display’ tactic increases the rate at which sonar and compass information is made available by the system.

4.4 Query ID

The host tags each command packet with a 1 byte, randomly generated Query ID. Upon receiving the command, The SCU returns the Query ID with the data packet acquired for that particular query. Through this process, the Query ID returned by the

SCU, helps the host system keep track of the acquisition sequence. It also helps identify problems with the serial transmission, such as misaligned data frames. Packets with mismatched Query IDs are assumed corrupt and discarded.

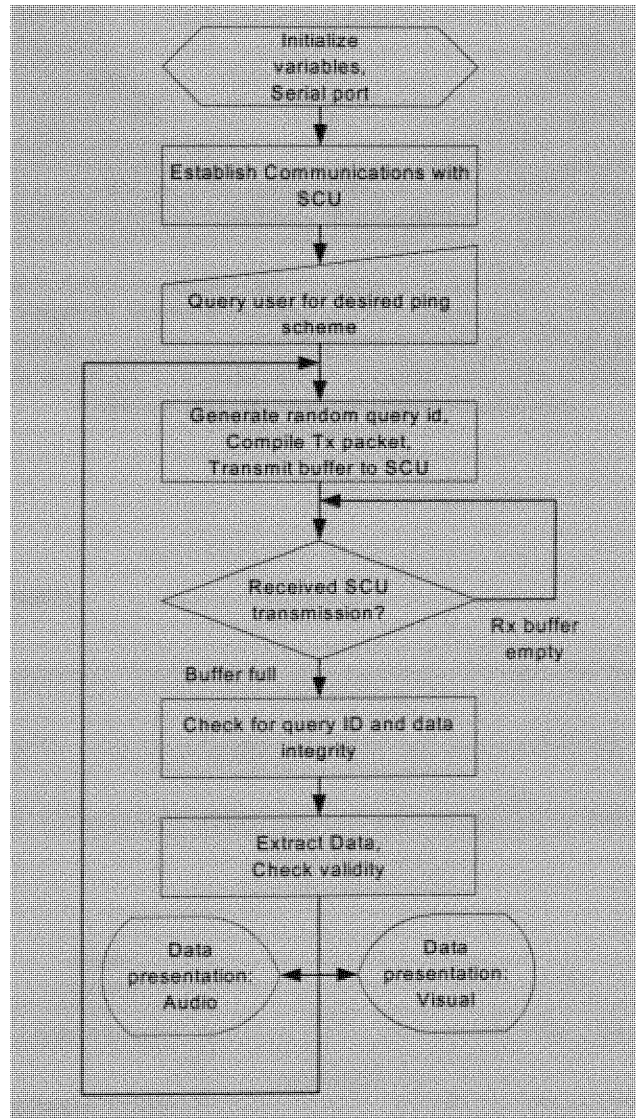


Figure 4-2. The program flow chart for the host system

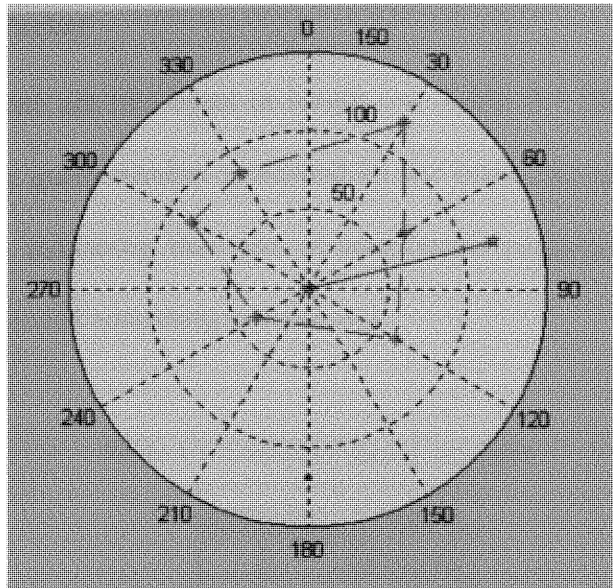


Figure 4-3. Graphical presentation method for the acquired ranging and bearing data. The Blue line indicates the Earth's magnetic North, with respect to the user's facing direction. The red asterisks indicate the detected obstacle positions in the respective azimuth. The dashed lines connecting the asterisks are used only to enhance the visibility of the asterisk positions. Range measurements are available only at the six designated azimuths.

4.5 Packet Structures

All packet structures have been standardized to ensure proper communication between host and SCU. The command packets are assembled on the host system and received by the SCU. Conversely, Data packets are compiled by the SCU and transmitted to the host system.

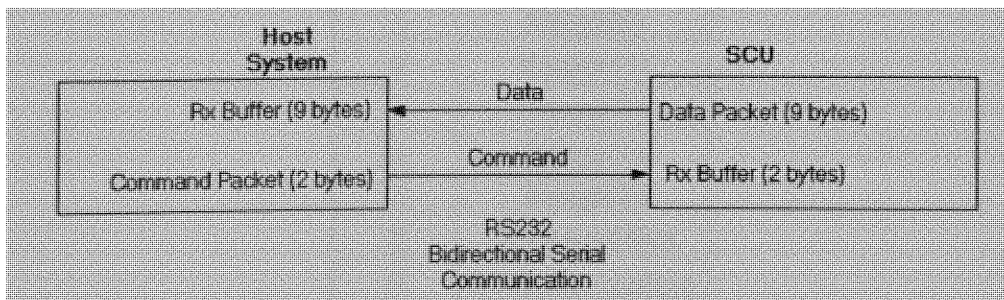


Figure 4-4. Packet Exchange Process Between Host and SCU

4.5.1 Command Packet

The command packets are two bytes in length. As Table 4-1 illustrates, the first byte contains the random query number generated by the host. The second byte contains a one-byte command, indicating the type of ping mode requested to the SCU. The Command byte is encoded according to Table 4-2.

Table 4-1. Command packet structure

| Location | Content |
|-----------------|----------------|
| 1 | Query ID |
| 2 | Command |

Table 4-2. 1 Byte Encoded Command

| Command Byte | Ping Mode |
|---------------------|------------------|
| 0x55 | Sweep |
| 0xAA | Flood |
| 0x64 | Interlace |
| 0x09 | Register Clear |

Table 4-3. Data Packet Structure

| Location | Content |
|-----------------|-----------------|
| 1 | Query ID |
| 2 | Range(10) |
| 3 | Range(8) |
| 4 | Range(11) |
| 5 | Range(4) |
| 6 | Range(1) |
| 7 | Range(2) |
| 8 | Command byte |
| 9 | Compass reading |

4.5.2 Data Packet

The SCU generates a data packet that contains all the range and bearing information as well as the command byte and Query ID that is being served. The data packet, which is nine bytes long, as shown in Table 4-3, is transmitted to the host.

4.6 Expandability

The system offers software expandability to accommodate new ping schemes. The OOPIC also has expansion capability through the I²C bus. More sensors interfaced with other controllers could transmit measurements through the I²C bus. The SCU, in such architecture, would collect data measurements from remote controllers and transmit this information to the host system.

5 TEST AND EVALUATION

5.1 Circuit Testing

All Functional blocks of the SCU analog and digital circuits were tested individually before integrating them into one circuit board. The clock circuit was tested for maximum stability. After building the circuit with initially calculated component values, the timing components were altered through trial and error, to achieve a duty cycle as close as possible to 50% as well as the desired frequency value of 6.85KHz. The logic operations associated with the counters were initially tested in simulation. After obtaining the desired outcomes in the MicroCap [25] simulation, the counter operations were verified on a prototype vector board. After identifying potential flaws in the logic, modifications were made, and the circuits were ultimately fabricated on a printed circuit board. The RS232-TTL level converter circuit was also tested on a prototype board and the performance was examined to find potential problems in integration.

The sonar modules were initially tested with discrete logic circuits. After verification of the proper operations, they were interfaced with the OOPIC.

5.2 Serial Communications Testing

The serial data communication was tested by debugging the raw data flow using a serial port scanner. A PalmOS based organizer used as a serial terminal, was programmed to dump serial communications activity on the screen while the device was connected in parallel with the serial data lines between SCU and the host. The captured data was

verified with the transmitted and received data. The test was performed on both the SCU and the host data transmit lines.

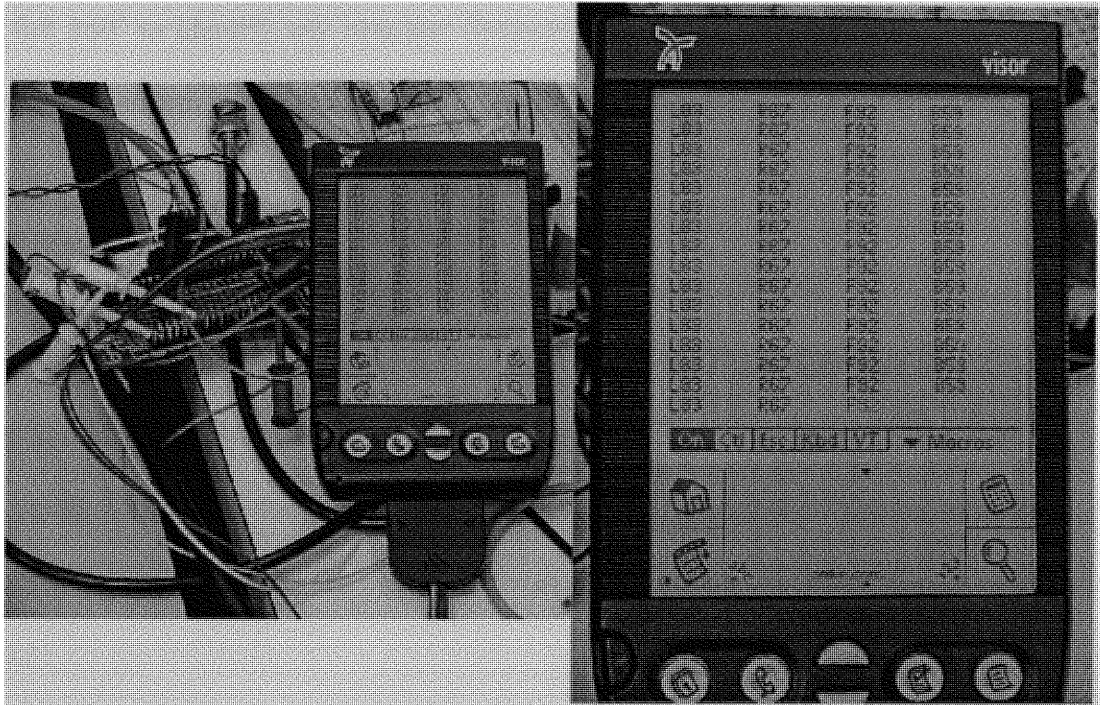


Figure 5-1 Serial port data monitoring

5.3 Rollover Prevention Test

To verify that the rollover protection mechanism prevents the clearing of the counters after they have reached a count of 255, the ECHO lines were disconnected from the sonar modules. The Ping signal was triggered while the ECHO input lines on the SCU were pulled up high. After waiting for about one second, the registers were read into the OOPIC. The fact that all of the register values were set to 255 verifies that the rollover protection mechanism was successful.

5.4 Compass Test

The compass directions acquired from the designed system was compared with a regular magnetic compass. The headgear where the compass was mounted was rotated to a full 360° to identify any irregular fluctuation in the North indicator. The system was found to perform in agreement with the manufacturer specifications.

5.5 Ranging accuracy

The accuracy of the omni-directional range measurements was tested in Sweep, Flood and Interlaced modes to identify the most suitable configuration for the intended application. The Ranging system was placed in several indoor test scenarios. Actual physical distances to the surrounding obstacles were measured. A measure of accuracy in each panoramic snapshot is established through an index of error, shown in equation 5.5.1.

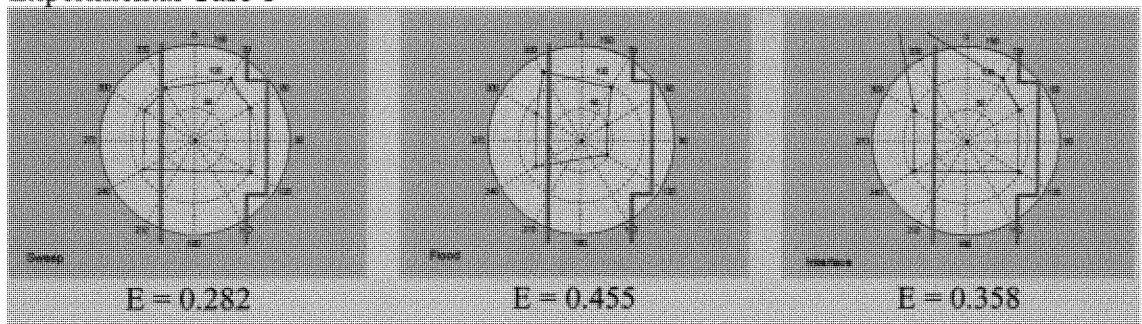
$$E = \frac{\sum_{n=1}^6 |d_n - m_n|}{\sum_{n=1}^6 d_n} \quad (5.5.1)$$

d_n is the actual distance in each of the six measured directions, $n=1 \dots 6$; and m_n is the distance measured by the sonar system for each of those directions.

The omni directional range measurements were acquired and the errors were calculated to compare the accuracy between the three ping modes. Figure 5-2 and Figure 5-3 display a graphical comparison between the three modes of range readings, tested in five different cases, representative of the different test scenarios. The graphical display for range measurement is the same as the method shown in Figure 4-3, with the actual

obstacle layout superimposed in the plot. It is observed that, almost in all cases, the flood mode shows smaller distances. From case 4, it is evident that the range reading accuracy of the sweep and interlace modes are comparable to each other. However, case 5 shows that the interlaced mode achieves better accuracy. From case 2 and case 4, it appears that the interlace mode shows better dynamic range. Similar errors have been recorded in all three modes in case 1, though the Interlace mode seems to record a comparably larger distance than reality on sensor 11.

Experimental Case 1



Experimental Case 2

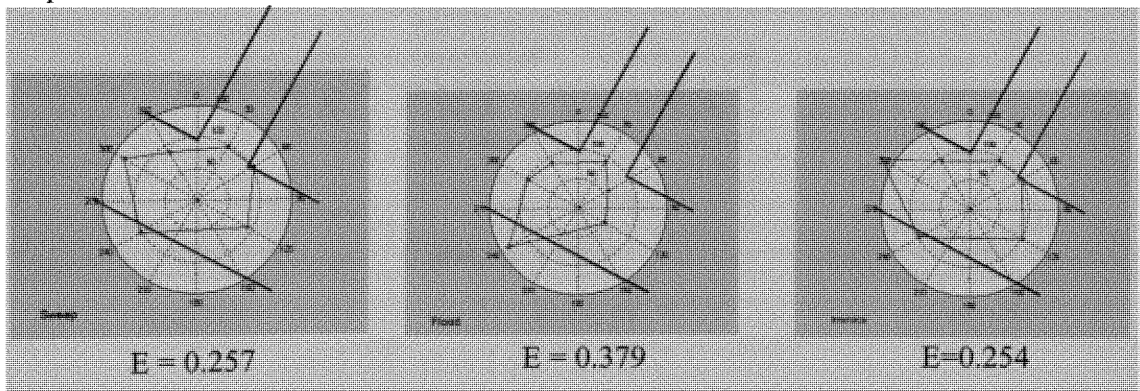
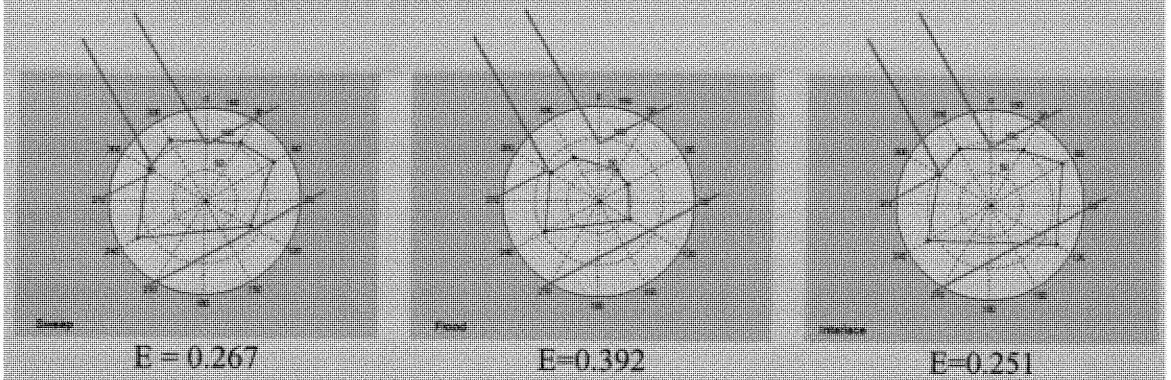
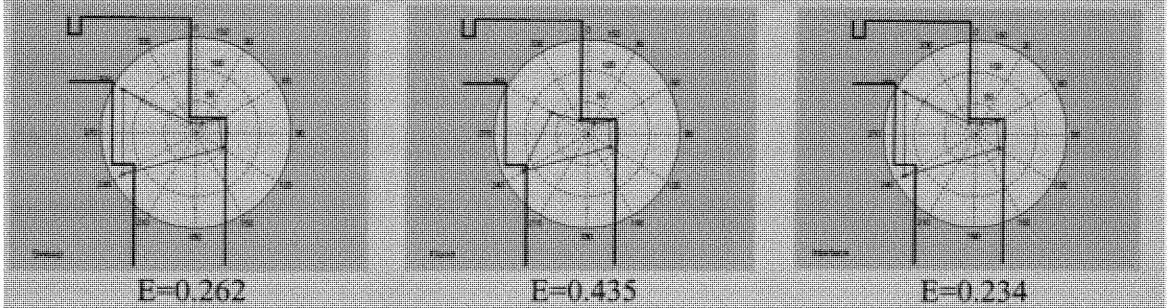


Figure 5-2. Measurements taken in the three modes, graphically displayed for cases 1 and 2

Experimental Case 3



Experimental Case 4



Experimental Case 5

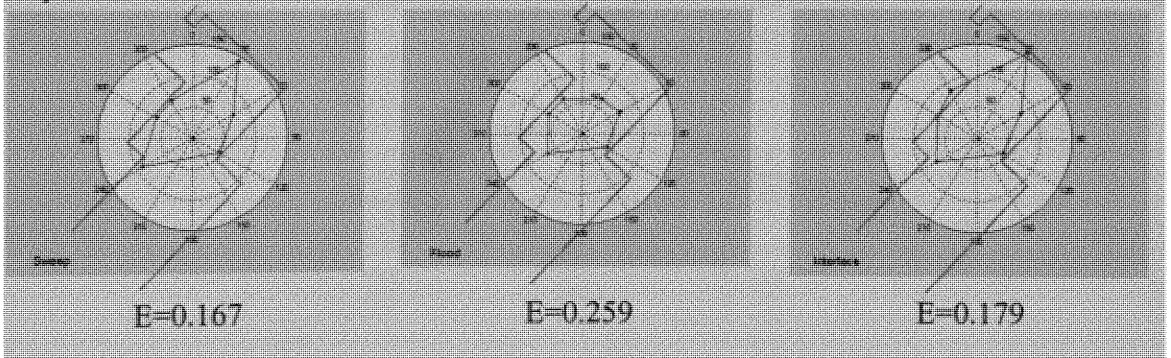


Figure 5-3. Measurements taken in the three modes, graphically displayed for cases 3 through 6

5.6 Performance Evaluation

The ranging accuracy tests show that the flood mode records a shorter distance in nearly all cases. The cause for this has been identified as co-channel cross talk, due to the narrow lobe placement between neighboring sensors. As illustrated in Figure 5-4, when two neighboring sensor are active at the same time, their beams overlap at a certain distance away from the sensors. This distance was theoretically calculated to be approximately 7 inches from the sensors. However, in practice this distance is farther away, since the acoustic power concentration is minimal at the extreme borders of the angular lobes. From the experimental results of Figure 5-2 and Figure 5-3, it can be assumed that this region of convergence lies beyond 50 inches. This means that sensors can discriminate between objects within 50 inches. This mode may produce satisfactory performance for short ranges. However, this mode did not meet the required performance characteristics for a goal of 120 inches.

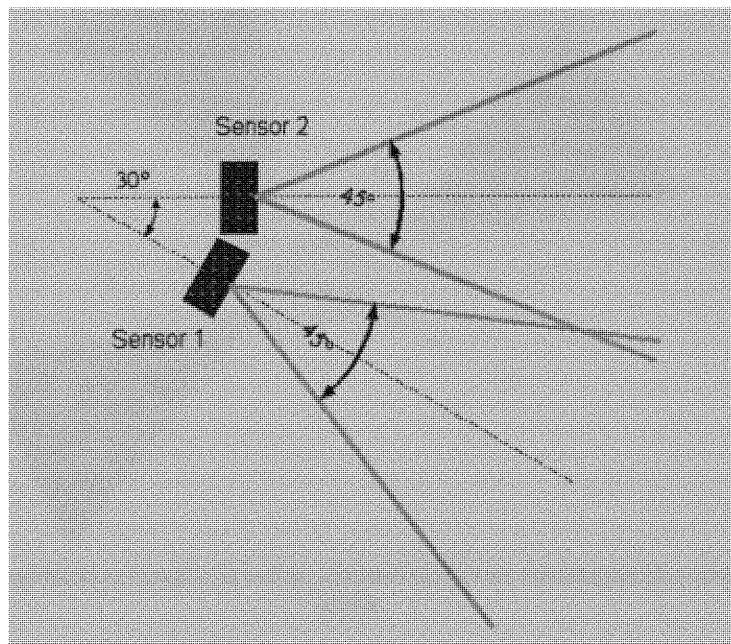


Figure 5-4. Beam overlap causing cross talk between active neighboring sensors

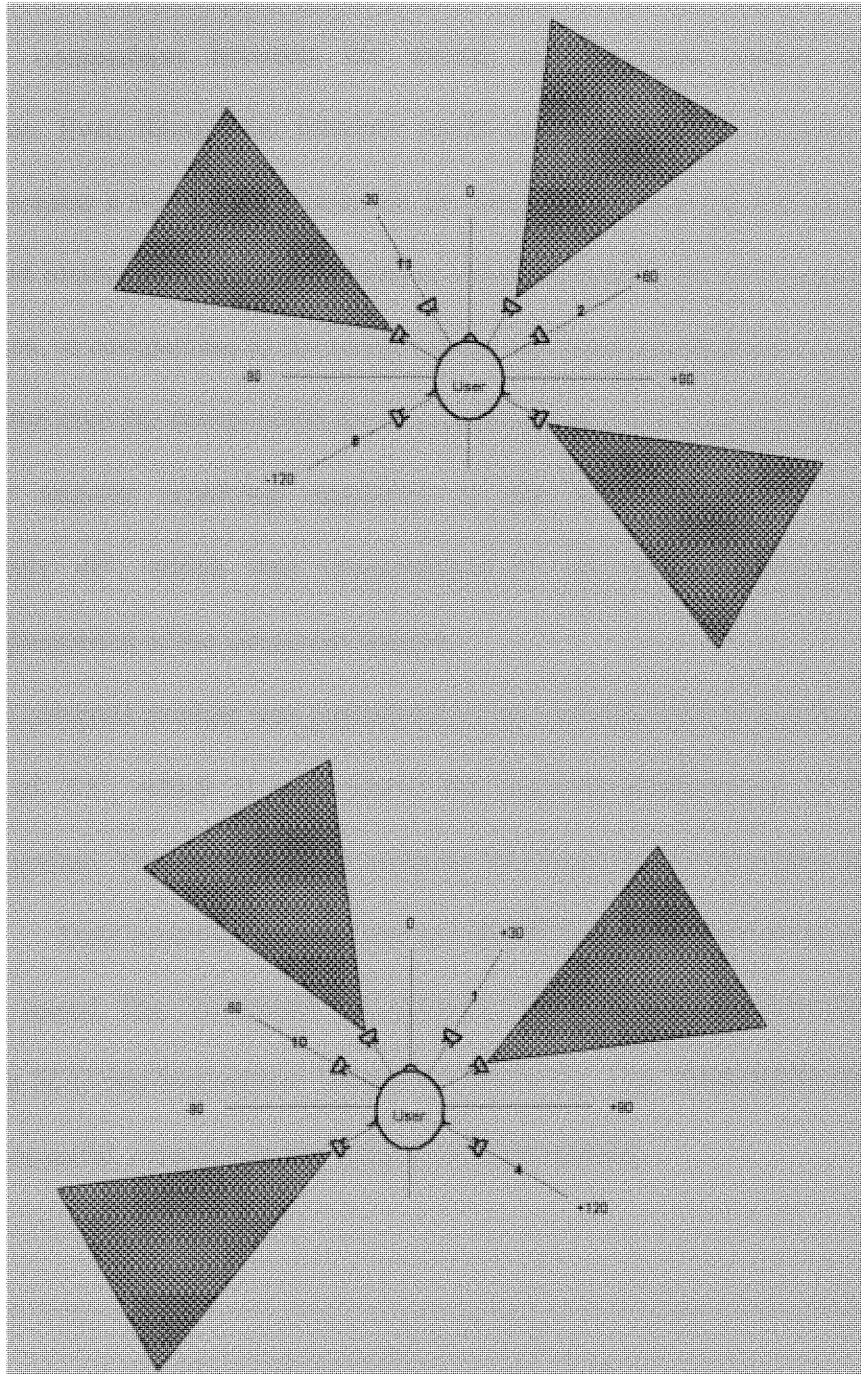


Figure 5-5. Top and bottom images showing alternating sensor measurements using the interlace mode

The average Index of error for the flood mode was found to be 0.375. The index of error for the sweep and interlace modes were found to be 0.257 and 0.253 respectively. Therefore, comparison between the sweep and interlace modes show similar performance. Even though the interlace mode activates multiple sensors at the same instant, it creates a wider lobe placement by activating alternating sensors (Figure 5-5). This method improves the spatial resolution while decreasing the time required capturing all six ranges, in comparison to the sweep mode. The interlace mode was found to require 40% less time than the sweep mode for a complete panoramic range recording.

The angular resolving capability of the sensor array was found to be approximately 40°, which was characterized by the spatio-acoustic emission and sensitivity properties of the ultrasonic sensors.

The compass module was guaranteed by the manufacturer to produce a minimum resolution of 3.5°, which is well above the required performance of the intended navigation system. The stability of the compass was found to be well within satisfaction.

Data communications between the SCU and the host were reliable, with satisfactory speed. The “command and response” protocol designed over the serial link between the SCU and host system functioned flawlessly. The “fetch while display” tactic was also successful, resulting in prompt presentation updates.

6 CONCLUSION AND FUTURE WORK

This research has resulted in the design, implementation and preliminary evaluation of an information gathering system for a navigational instrument that is intended to be used in indoor environments with significance to blind navigation. The test results show that the system is capable of performing the tasks of obstacle and direction detection. The system presents the information to the data display unit with no cost in terms of computational effort by the host system for navigational data processing. More over, the autonomy of the localized computation and harmonized command and response operation guarantees a flawless operation loop. The host system requests an update only when it is finished with the presentation task. The SCU operation is fast enough to ensure that a new set of data will be ready for the host system on the next request cycle. This means that the SCU will wait an updated data, for the host to complete the data presentation process. This asynchrony adds to the robust behavior observed in the system performance. In the present system, the angular resolution capability of the sonar array is approximately 45° . The angular resolution in azimuth in the HRTF rendering process is 30° . Improving the angular resolution capability of the sonar system to match the HRTF capability would maximize the spatialization effect.

6.1 Suggestions

Several suggestions could be made to improve the system. The sonar cross talk deserves the highest attention amongst potential system enhancements. Sonar transceivers with different frequencies could be used to reduce the cross talks between different sensors. Another possibility is using coded ping signals. The system angular resolution

could be improved by adding more sensors, particularly in the 9, 12 and 3 o'clock directions. However, adding more ultrasonic sensors is likely to cause more crosstalk. A horizontally scanning infrared reflectometric laser sensor could improve the angular resolution of the system significantly, without compounding the problem of co-channel interference. However, this would probably require major re-design of the control and processing hardware.

LIST OF REFERENCES

- [1] Benjamine J.M., Ali, N.A. and Schepis A.F., "A Laser Cane for the Blind", in Proc. San Diego Biomedical Simp., 1973, vol. 12, pp. 53-57.
- [2] Shoval S., Borenstein J., and Koren, Y. *Auditory Guidance with the Navbelt—A Computerized Travel Aid for the Blind*. IEEE Transactions on Systems, Man and Cybernetics. Vol. 28, No. 3, 1998, pp. 459 – 467.
- [3] <http://sonicvision.co.nz>
- [4] Kay, L. "Auditory perception of objects by blind persons, using a bioacoustic high resolution air sonar," J. Acoust. Soc. Am. 107, 3266–3275 (2000).
- [5] Kay, L. "A sonar aid to enhance spatial perception of the Blind: engineering design and evaluation". The Radio and Electronic Engineer, vol 44, No. 1, pp 605-627.
- [6] Russel, L. 1965, "Travel Path Sounder", in Proceedings of Rotterdam Mobility Res. Conference. New York: Amer. Foundation for the Blind, 1965.
- [7] Shoval, S., Borenstein, J., and Koren, Y., 1998, "The NavBelt - A Computerized Travel Aid For The Blind Based On Mobile Robotics Technology." IEEE Transactions on Biomedical Engineering, Vol. 45, No 11, Nov. 1998, pp. 1376-1386.
- [8] Tachi S. and Komoriya K., "Guide Dog Robot", presented at 2nd Int. Congress on Robotics Research, Kyoto, Japan, 1984.
- [9] Ulrich I. And Borenstein J., "GideCane — Applying Mobile Robot technologies to Assist the Visually Impaired", IEEE Transactions on Systems, Man, and Cybernetics, —Part A: Systems and Humans, Vol. 31, No. 2, March 2001, pp. 131-136
- [10] Brabin, J. A. "New Developments in Mobility in Mobility and Orientation Aids for the Blind", IEEE Trans. Biomed. Eng., vol. BME-29, pp. 285-290, Apr. 1982.
- [11] Loomis, J. M., Golledge, R. G., & Klatzky, R. L. (1998). "Navigation system for the blind: Auditory display modes and guidance". Presence: Teleoperators and Virtual Environments, 7, 193-203.
- [12] Helal, A., Moore, S. and Ramachandran, B. "Drishti: An Integrated Navigation System for Visually Impaired and Disabled", Proceedings of the 5th International Symposium on Wearable Computer, October 2001, Zurich, Switzerland

- [13] Gupta N., Barreto A. and Ordonez C. "Spectral Modification of Head Related Transfer Functions for Improved Virtual Sound Spatialization", Proceedings of the 2002 IEEE International Conference on Acoustics, Speech and Signal Processing (CASP 2002), May 13-17, 2002, pp. II-1953
- [14] Ordonez C., "Development of a 3-D-Sound Guidance System for Blind Navigation in Virtual Environments", 2002, M.S. thesis, Florida International University.
- [15] KMZ10A Magnetic field sensor; Document order number: 9397 750 03584; © Philips Electronics N.V. 1998
- [16] Streetman G. B., Solid State Electronic Devices, 2nd edition, Prentice-Hall, 1980.
- [17] <http://www.oopic.com/tech.html>
- [18] Microchip PIC16F7x Data Sheet: 28/40-pin, 8-bit CMOS FLASH Microcontrollers; © 2002, Microchip Technology Incorporated
- [19] Hall D. V., "Microprocessors and Interfacing: Programming and Hardware", Glencoe McGraw-Hill
- [20] I²C, A networking solution for integrated circuits, <http://www-us2.semiconductors.philips.com/i2c/>
- [21] Technical Reference Manual, IBM Corporation, Personal Computer, 1981
- [22] <http://www.mathworks.com>
- [23] I²C Bus specification version 2.1; Philips Semiconductors, January 2000.
- [24] The MAX3221/MAX3223/MAX3243 Data Sheet; document 19-0306; Rev 7; 10/02 Maxim Semiconductors Inc.
- [25] <http://www.spectrum-soft.com>
- [26] HeadZap: "AuSim 3D HRTF measurement system manual". AuSim, Inc., 4962 El Camino Real, Suite 101, Los Altos, CA 94022, 2000.

APPENDICES

APPENDIX A. SCU CODE LISTING

```
'SCUr3.bas
'03/17/03
'Maroof H. Choudhury,
'DSP Lab, Florida International University
'This program is designed to configure the OOPIC to function as the Sonar Control Unit (SCU)

'-- IO Channel mapping --
'I/O channel   sonar clock Direction
'  0           10
'  1           8
'  2           11
'  3           4
'  4           1
'  5           2
'---

'-- IO objects --
'S0-S5: 1 bit, out: register select pins: Active low
'P0-P5: 1 bit, out: Ping mask (trigger enable) pins: Active high
'Trig: 1 bit, out: Ping trigger: Active high
'Data: 8 bit, in: Data Bus (connected to 3 state devices)
'Compass: I2C node for Compass

'--- Code begins here ---
dim S0 as new oDio1
dim S1 as new oDio1
dim S2 as new oDio1
dim S3 as new oDio1
dim S4 as new oDio1
dim S5 as new oDio1
dim P0 as new oDio1
dim P1 as new oDio1
dim P2 as new oDio1
dim P3 as new oDio1
dim P4 as new oDio1
dim P5 as new oDio1
dim Trig as new oDio1
dim Data as new oDio8
Dim Compass As New oI2C

'-- Serial Comm objects --
dim HostLink as new oSerialPort
dim TxBuff as new oBuffer(9)  '-- Transmit buffer: 9 bytes --
dim slot as new oByte         '-- FIFO slot --
dim Ptr as new oByte          '-- Buffer pointer --

'-- Host control protocol elements --
dim Command as new oByte      '-- Command byte from host --

sub main()
  call SySinit  'Resource Initialisation on power up
do
```



```

    call ClipRange      '-- clip range measurements to 120" --
    call HostRx        '-- receive command packet --
    call HostTx        '-- transmit data packet --
    TxBuff.location = 7
    TxBuff.value = Command    '-- return command byte in data packet--
    select case Command      '-- Execute commanded ping type --
        case 100
            call interlace
        case 170
            call FloodPing
        case 85
            call SweepCW
        case 58
            call SweepCCW
        case 0,1,2,3,4,5
            call PointPing(Command)
        case else
            call ClearAll
    end select
'-- compass bearing --
    Compass.Location = 1      '-- Address of single byte bearing --
    Compass.Width = cv8bit    '-- Compass Data is 1-byte wide --
    TxBuff.location = 8
    TxBuff.value = Compass.Value '-- Read compass measurement --
    loop
end sub

sub Trigger()
    Trig.set    '-- ping active high --
    Trig.clear
end sub

sub FloodPing()    '-- subroutine for flood mode ping --
    P0=1: P1=1: P2=1: P3=1: P4=1: P5=1
    call Trigger
    P0=0: P1=0: P2=0: P3=0: P4=0: P5=0
    oopic.delay=1
    TxBuff.location=1: S0.clear: TxBuff.value=Data: S0.set
    TxBuff.location=2: S1.clear: TxBuff.value=Data: S1.set
    TxBuff.location=3: S2.clear: TxBuff.value=Data: S2.set
    TxBuff.location=4: S3.clear: TxBuff.value=Data: S3.set
    TxBuff.location=5: S4.clear: TxBuff.value=Data: S4.set
    TxBuff.location=6: S5.clear: TxBuff.value=Data: S5.set
end sub

sub PointPing(pchan as byte)
    TxBuff.location = pchan+1
    select case pchan
        case 0
            P0.set: Trigger: P0.clear: oopic.delay = 1
            S0.clear: TxBuff.value=Data: S0.set
        case 1
            P1.set: Trigger: P1.clear: oopic.delay = 1
            S1.clear: TxBuff.value=Data: S1.set
    end select
end sub

```

```

    case 2
      P2.set: Trigger: P2.clear: oopic.delay = 1
      S2.clear: TxBuff.value=Data: S2.set
    case 3
      P3.set: Trigger: P3.clear: oopic.delay = 1
      S3.clear: TxBuff.value=Data: S3.set
    case 4
      P4.set: Trigger: P4.clear: oopic.delay = 1
      S4.clear: TxBuff.value=Data: S4.set
    case 5
      P5.set: Trigger: P5.clear: oopic.delay = 1
      S5.clear: TxBuff.value=Data: S5.set
  end select
end sub

sub Interlace()
  P0=0: P1=1: P2=1: P3=0: P4=0: P5=1
  call Trigger
  P0=0: P1=0: P2=0: P3=0: P4=0: P5=0
  oopic.delay=1
  TxBuff.location=2: S1.clear: TxBuff.value=Data: S1.set
  TxBuff.location=3: S2.clear: TxBuff.value=Data: S2.set
  TxBuff.location=6: S5.clear: TxBuff.value=Data: S5.set
  P0=1: P1=0: P2=0: P3=1: P4=1: P5=0
  call Trigger
  P0=0: P1=0: P2=0: P3=0: P4=0: P5=0
  oopic.delay=1
  TxBuff.location=1: S0.clear: TxBuff.value=Data: S0.set
  TxBuff.location=4: S3.clear: TxBuff.value=Data: S3.set
  TxBuff.location=5: S4.clear: TxBuff.value=Data: S4.set
end sub

sub SweepCW()
'ping sweep from C8 to C4
  PointPing(1)
  PointPing(0)
  PointPing(2)
  PointPing(4)
  PointPing(5)
  PointPing(3)
end sub

sub SweepCCW()      '-- clockwise ping sweep --
'ping sweep from C4 to C8
  PointPing(3)
  PointPing(5)
  PointPing(4)
  PointPing(2)
  PointPing(0)
  PointPing(1)
end sub

sub ClearAll()      '-- clear Tx Buffer --
  call Trigger

```

```

    TxBuff.location=1: S0.clear: TxBuff.value=Data: S0.set
    TxBuff.location=2: S1.clear: TxBuff.value=Data: S1.set
    TxBuff.location=3: S2.clear: TxBuff.value=Data: S2.set
    TxBuff.location=4: S3.clear: TxBuff.value=Data: S3.set
    TxBuff.location=5: S4.clear: TxBuff.value=Data: S4.set
    TxBuff.location=6: S5.clear: TxBuff.value=Data: S5.set
end sub

sub ClipRange()          '-- function to clip range readings to 120 inches --
  for TxBuff.location = 1 to 6
    if TxBuff.value > 160 then TxBuff.value = 160
  next TxBuff.location
end sub

sub HostRx()             '-- subroutine to receive command packets --
  do: loop until(HostLink.Received = cvtrue) '-- Wait until Host polls for next data --
    slot=HostLink.value          '-- reset received flag & read key --
    Command=HostLink.value      '-- receive command byte --

    TxBuff.Location=0
    TxBuff.value=slot          '-- return query id --
end sub

sub HostTx()            '-- subroutine to transmit Tx buffer over serial comm--
  while TxBuff.location<=TxBuff.width
    slot=TxBuff.value
    HostLink.value=slot
    while HostLink.Transmitting=1
      wend
    TxBuff.Location=TxBuff.Location+1
  wend
end sub

sub SysInit()           '-- IO objects initialization ---
  S0.ioline = 24
  S0.direction = cvoutput
  S0.set
  S1.ioline = 25
  S1.direction = cvoutput
  S1.set
  S2.ioline = 26
  S2.direction = cvoutput
  S2.set
  S3.ioline = 27
  S3.direction = cvoutput
  S3.set
  S4.ioline = 28
  S4.direction = cvoutput
  S4.set
  S5.ioline = 29
  S5.direction = cvoutput

```

```

S5.set

P0.ioline = 1
P0.direction = cvoutput
P0.clear
P1.ioline = 2
P1.direction = cvoutput
P1.clear
P2.ioline = 3
P2.direction = cvoutput
P2.clear
P3.ioline = 4
P3.direction = cvoutput
P3.clear
P4.ioline = 5
P4.direction = cvoutput
P4.clear
P5.ioline = 6
P5.direction = cvoutput
P5.clear

Data.iogroup = 1
Data.direction = cvinput

Trig.ioline = 7
Trig.direction = cvoutput
Trig.clear

'-- I2C bus setup for compass --
Compass.Node = 96      '-- Decimal of Hex address 0xC0 shifted right by 1 --
Compass.Mode = cv10bit '-- I2C mode is 10-Bit Addressing --
Compass.NoInc = 1     '-- Don't increment --

'-- Communications setup (9.6K 8N1) --
HostLink.baud=cv9600
HostLink.mode=0
HostLink.operate=cvtrue
end sub

```

APPENDIX B. HOST SYSTEM MATLAB™ SCRIPT

```
%Sonhost.mat
%03/03/03
%Maroof H. Choudhury
%Floria International University
%This script runs on the host system in conjunction with the SCU
%The program commands the SCU, recieves measurements and presents the
%measurements in real-time

% -- Code begins here --
clc; close all; clear all;

% -- Configure PC Serial port to communicate with the SCU --
pic = serial('COM1');
pic.Baud=9600;
pic.DataBits=8;
pic.Parity='none';
pic.StopBits=1;
pic.FlowControl='none';
pic.ReadAsyncMode='continuous';
pic.Timeout=3;
pic.InputBufferSize=9;

% -- Variables --
RxPacket=uint8(zeros(8,1)); % -- input (data) buffer contents --
Range=zeros(6,1); % -- Range data array --
ch=6; % -- Number of sonar channels channels --
w=3; % -- Running average taps --
Datalog = zeros(1,7); % -- captured measurement history --
phid=[240,300,330,30,60,120,240]; % -- Direction angles in degree --
phir=(phid .* pi)/180; % -- Direction angles in radians --

% -- Clock direction to control channel assignment map --
C8 = 1;
C10 = 0;
C11 = 2;
C1 = 4;
C2 = 5;
C4 = 3;

% -- SCU command set --
sweep = uint8(85);
flood = uint8(170);
wipe = uint8(9);
interlace = uint8(100);

% -- Start PIC datacom --
fopen(pic);

% -- Clear Pic buffer --
Qid = floor(rand*255); % -- random query number --
```

```

fwrite(pic,[key wipe]);          % -- command SCU --
RxPacket = fread(pic,8,'uint8'); % -- read SCU data buffer --

% -- User Interface --
Q = input('Ping mode (flood/sweep/wipe/interlace/channel#) : '); % -- user input for desired ping method
--
T = input('Number of samples to log: ');          % -- number of repeating pings --
tic;                                             % -- record start time for datalog --

while (T > 0)

    % -- Tx Command packet to SCU --
    Qid=floor(rand*255); % -- generate random query id --
    fwrite(pic,[Qid Q]); % -- transmit command packet --

    % -- Rx packet sent from PIC --
    RxPacket = fread(pic,9,'uint8'); % -- retrieve data packet from com port receive buffer --

    % -- Check Query id validity --
    if RxPacket(1)~=Qid
        disp('Sequence mismatch!');
        beep;
    end

    % -- Extract Range data --
    Range(1)=RxPacket(C8+2); % -- Range 8 o'clock --
    Range(2)=RxPacket(C10+2); % -- Range 10 o'clock --
    Range(3)=RxPacket(C11+2); % -- Range 11 o'clock --
    Range(4)=RxPacket(C1+2); % -- Range 1 o'clock --
    Range(5)=RxPacket(C2+2); % -- Range 2 o'clock --
    Range(6)=RxPacket(C4+2); % -- Range 4 o'clock --
    Range' % -- numeric display --
    Datalog = [Datalog; [toc Range]]; % -- log history with timestamp --

    % -- Extract compass data --
    BB=RxPacket(9); % -- bearing byte from dtaa packet --
    Bearing=BB*pi/180; % -- bearing angle in radians --

    % -- Graphical Display --
    radr=[Range',Range(1,1)];
    polargeo([0 Bearing],[0 120],'bp-'); hold on; % -- bearing diplay --
    polargeo(pi, 120, 'k. '); hold on; % -- to seperate bearing display from range display --
    polargeo(phir,radr,'r*--'); hold off; % -- range display --

    pause(.5); % -- small delay to refresh display --

% Note:
% This would be the typical entry point to the HRTF rendering engine.
% The range matrix and compass data would be passed to the HRTF rendering
% engine using a function similar to this ...
% SpialSound = hrtfrender([Range, 1],[phir, Bearing]);

    T=T-1;
end

```

```
% -- Termination --  
beep;  
disp('Closing commlink to OOPIC...')  
  
% -- close SCU communications --  
fclose(pic);  
delete(pic);  
clear pic;  
  
disp('Success!');
```