

# **CERTIFICATION OF APPROVAL**

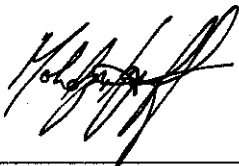
## **ULTRASONIC VISION SYSTEM**

By

**KARSOU AHMED OBAID**

A project dissertation submitted to the  
Electrical & Electronic Engineering Department  
Universiti Teknologi PETRONAS  
in partial fulfilment of the requirement for the  
Bachelor of Engineering (Hons)  
(Electrical & Electronic Engineering)

Approved:



---

Dr. Mohd Zuki Bin Yusoff  
Project supervisor

**UNIVERSITI TEKNOLOGI PETRONAS  
TRONOH, PERAK**

December 2011

## **CERTIFICATION OF ORIGINALITY**

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and acknowledgements, and that the original work contained herein has not been undertaken or performed by unspecified sources or persons.



---

**KARSOU AHMED OBAID**

## **ABSTRACT**

Visually impaired people face major problems in activities which require visual awareness such as: navigation, reading, sport, etc. This project attempts to contribute to the visually impaired community as well as to academia by developing low cost, easy to construct navigation tool which aids visually disabled to avoid objects while navigation. The proposed method utilizes EZ1 ultrasonic modules, PIC18F4550 microcontroller, and 6 Volts buzzer like alarm. For user convenience and comfort, the system is wholly implemented on an ice cap. A coherent development methodology was adopted to implement the proposed prototype by the use of potentiometers and an LCD during the early and mid phases of the project to emulate the ultrasonic modules and visually debug the intermediate processing results. The practicality and uniqueness of the implementation on an ice cap with the ease to wear feature shape the significance of our contribution. Remarkably, the prototype was tested in different scenarios and was able to detect objects up to 6.45 meters and send hearable distinct alarms at 2 meters and 0.5 meter far. By the use of the proposed prototype, visually impaired users can independently navigate while avoiding objects with two levels of alarm indication. In addition, the prototype is developed to be flexible in such way that it can be easily exploited for future extension and improvements for both hardware and software by utilizing the abundant resources of PIC18F4550 microcontroller and CCS C compiler library.

## TABLE OF CONTENTS

CHAPTER 1 INTRODUCTION.....	1
1.1 Background of Study.....	1
1.2 Problem Statement .....	1
1.3 Project Objectives and Scope .....	2
1.4 Limitations.....	2
1.5 Organization of Report.....	2
CHAPTER 2 LITERATURE REVIEW .....	3
2.1 Preliminaries .....	3
2.2 Related Work.....	3
CHAPTER 3 METHODOLOGY .....	6
3.1 Project Development Flow Chart.....	6
3.2 System Identification and Required Tools .....	7
3.2.1 Ultrasonic module: Max Sonar – EZ1 ultrasonic range finder.....	7
3.2.2 PIC18F4550 microcontroller.....	10
3.2.3 CCS C compiler .....	12
3.2.4 SK40C PIC development kit and UIC00B programmer.....	12
3.3 Hardware Development.....	15
3.3.1 Simplified circuit with emulated ultrasonic module.....	15
3.3.2 Single, dual ultrasonic-module based system.....	16
3.3.3 Construction of final prototype.....	16
3.4 Software Development .....	17
3.4.1 PIC programming methodology .....	17
3.4.2 Overview of the algorithm and code.....	22
3.4.2.1 Header file: main.h .....	26
3.4.2.2 PIC_init( ) .....	27
3.4.2.3 read_range( ) .....	27
3.4.2.4 range2cm( ) .....	28
3.4.2.5 show_status( ).....	29
3.4.2.6 System monitor.....	30
3.5 Summary.....	31
CHAPTER 4 RESULTS AND DISCUSSION .....	32

4.1 Simplified Ultrasonic Module Emulation .....	32
4.2 Interfacing Ultrasonic Range Finder .....	33
4.3 Ultrasonic Vision System with Three LEDs Alarm Levels .....	34
4.4 Ultrasonic Vision System Prototype with Buzzer Alarm.....	36
CHAPTER 5 CONCLUSION AND FUTURE DIRECTION .....	39
5.1 Conclusion .....	39
5.2 Future Directions.....	39
REFERENCES .....	40
APPENDIX A PROJECT GANTT CHART .....	41
APPENDIX B PROGRAM LISTING FOR EXPERIMENT 4.1.....	43
APPENDIX C PROGRAM LISTING FOR EXPERIMENT 4.2.....	45
APPENDIX D PROGRAM LISTING FOR EXPERIMENT 4.3.....	47
APPENDIX E PROGRAM LISTING FOR EXPERIMENT 4.4 (MAIN.C) .....	49
APPENDIX E PROGRAM LISTING FOR EXPERIMENT 4.4 (MAIN.H) .....	51

## LIST OF FIGURES

Figure 1: Wearable Navigation System Developed by [3] .....	4
Figure 2: RFID Based Indoor Navigation System Suggested in [4] .....	5
Figure 3: General Flow chart of Project work.....	6
Figure 4: LV-MaxSonar-EZ1 Ultrasonic range finder .....	7
Figure 5: Ultrasonic Range Finder Schematic .....	9
Figure 6: Ultrasonic Range Finder Beam Characteristics.....	10
Figure 7: PIC18F4550 Pin Diagram .....	11
Figure 8: SK40C PIC Start-up Kit.....	13
Figure 9: UIC00B USB ICSP PIC Programmer.....	14
Figure 10: PIC Board and Programmer Connection.....	15
Figure 11: Potentiometer Emulating the Output of Ultrasonic Range Measurement Module.....	16
Figure 12: Project Wizard .....	18
Figure 13: Code generation using PIC wizard for the fuses .....	19
Figure 14: Plain code with automatic generated function calls according to the setup .....	20
Figure 15: Full Microcontroller Programming Flow.....	21
Figure 16: Ultrasonic Vision System Algorithm.....	23
Figure 17: System Program Highlighting the Included header files, Used Functions, Compiler Directives and the main( ) Function .....	25
Figure 18: The compiler directive to set fuses in the <i>main.h</i> header file.....	26
Figure 19: Initialization function.....	27
Figure 20: read_range() function.....	27
Figure 21: range2cm() function.....	28
Figure 22: show_status() function .....	29
Figure 23: System Monitor Code Utilizing Timer 2 Interrupt Service Routine .....	30
Figure 24: Simplified Vision System with Potentiometers Emulating Ultrasonic Modules .....	32
Figure 25: The Raw Value and the Value in cm for Distance Range.....	33
Figure 26: Ultrasonic Module with LEDs Alarm indicators.....	35
Figure 27: Ultrasonic Vision System Prototype .....	36
Figure 28: Ultrasonic Vision System (Range Finder Unit).....	37
Figure 29: Buzzer Fixture for the Ultrasonic Vision System.....	38

**Figure 30: Ultrasonic Vision System (the Mother Board and Circuitry from Inside).38**

## **LIST OF TABLES**

Table 1: LV- MaxSonar-EZ1 Ultrasonic Range Finder Specification .....	8
Table 2: PIC18F4550 Specification.....	11



## **LIST OF ABBREVIATIONS**

<b>PIC:</b>	<b>Peripheral Interface Controller</b>
<b>UART:</b>	<b>Addressable Universal Asynchronous Receiver/Transmitter</b>
<b>A/D:</b>	<b>Analogue to Digital</b>
<b>ADC:</b>	<b>Analogue to Digital Converter/ Conversion</b>
<b>D/A:</b>	<b>Digital to Analogue</b>

# **CHAPTER 1**

## **INTRODUCTION**

In this chapter, the background of the project, the problem statement, and the objectives of the project are discussed.

### **1.1 Background of Study**

Blindness or vision impairment is a prominent problem faced by millions of people all over the world according to the World Health Organization (WHO) [1]. Visually impaired people face major problems in activities which require visual awareness such as: navigation, reading, sport, etc. Physicians, inventors, entrepreneurs, and researchers have worked hard throughout the last century to present solutions which could allow visually impaired people to do many of their activities independently.

This project attempts to contribute to the visually impaired community by developing a low cost, easy to construct navigation tool which aids visually disabled people to avoid objects while navigating.

### **1.2 Problem Statement**

Visually disabled people need some external support to tell them about surrounding objects when navigating. Prior work as in [2]-[4] presents different techniques and solutions which allow visually impaired people to navigate independently. The methods used employ GPS, wireless networks, camera, and ultrasonic sensors. However, most of them are not easy to afford as in [3] or have limited battery life as in [3] or require additional hardware as in [2] or have limited coverage as in [4]. So, the aim is to provide a reliable yet affordable tool which enables visually impaired people to avoid obstacles while navigating.

### **1.3 Project Objectives and Scope**

The aim of this project is to construct an ultrasonic based vision system which provides the following features:

1. Allow visually disabled people to avoid fixed objects (e.g. wall)
2. Multi-level alarming
3. Wide detection coverage
4. Comfort of use

The project is envisaged to deliver a prototype which makes use of multiple ultrasonic sensors, microcontrollers and a buzzer like alarm. It is proposed that the ultrasonic sensors be located at the head level of the end user. The system is to be controlled by a microcontroller.

### **1.4 Limitations**

In order to implement a reliable and low cost solution, some potential problems have to be considered. Sensors generally require extra analog circuitry to filter noise readings as well as to amplify and adapt filtered signals in order to be easier to process by other devices. This scenario typically implies the use of different circuit components to build the system. However, having several components in the system consume more energy in comparison to optimized single integrated modules. In portable solutions, minimizing the power consumption is a key design concern. Therefore, in this project, we prefer the use of a microcontroller unit rather than microprocessor or FPGA based systems.

Moreover, in order to enable the continuous development on the prototype, a flexible and high processing microcontroller unit is suggested.

### **1.5 Organization of Report**

The report starts with this chapter which includes some background information, the scope and the objectives of this work. The remaining four chapters of the report are organized as follows: a literature review for the most relevant work are discussed and presented in Chapter 2. The methodology employed and the stages of the development are illustrated in Chapter 3. The experimental results are listed and discussed in Chapter 4. A conclusion and future directions are included in Chapter 5.

## **CHAPTER 2**

### **LITERATURE REVIEW**

In this section, some preliminaries and previous work citing the most recent research papers and scientific work relevant to this project are presented.

#### **2.1 Preliminaries**

The term “vision system” in most scientific sources likely refers to two different meanings: a biological one in which the term means: “the neurobiological circuitry that enables living beings to see” [5] or an artificial one in which the term means: “computer based systems where software performs tasks similar to “seeing”, usually aimed at industrial quality assurance, part selection, defect detection etc” [6]. In the context of this work, vision systems are those sensing systems which help visually impaired people to gain some sensory information regarding the surrounding objects. According to the defined scope, technological aids which help in object avoidance, navigation, reading textual data and detecting objects all fall under the vision systems class [7].

Vision systems generally use sensors, processing units and sensitization units to detect obstacles and convey to the user the information about it to enable him/her to take a proper action. According to the requirement of the system and the function it does, different sensing, processing and warning components are used such as: ultrasonic sensors, cameras, switches for sensing; microcontroller, microprocessor for processing; buzzer, speaker, vibrator to alarm and warn the user. In the following section, some of the relevant vision systems are considered for discussion and analysis.

#### **2.2 Related Work**

In the last two decades, several attempts to develop aiding vision systems have been

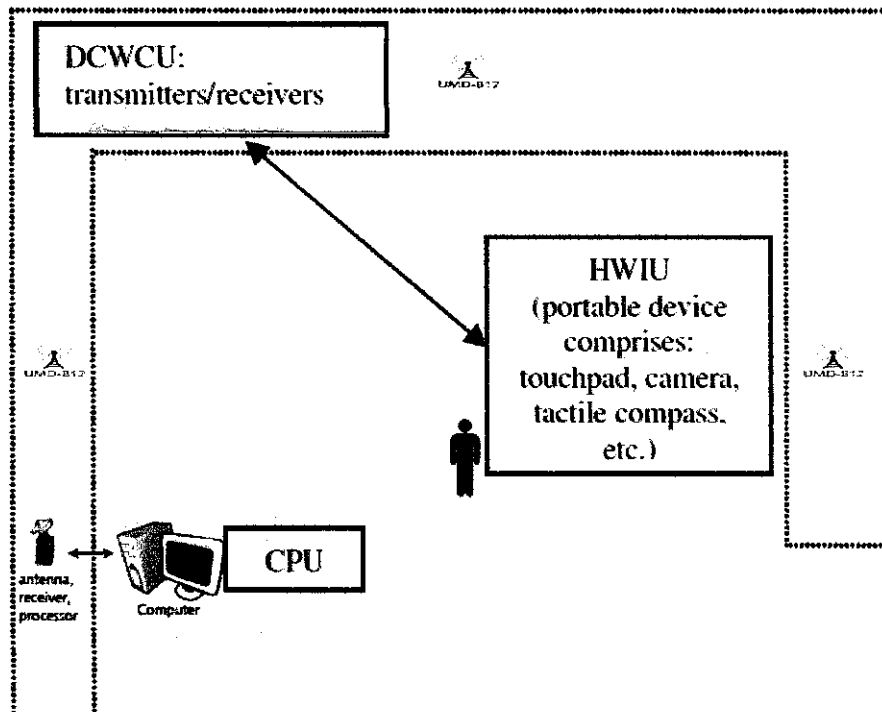
made. A. Helal *et al.* in [3] have developed a wireless based navigation system. Their method employs several wearable computer technologies, wireless network, Geographic Information System (GIS), and Global Positioning System (GPS) as in Figure 1. The proposed system provides the user with highly sophisticated interface and the luxurious amount of data which enables turn by turn navigation. Remarkably, the proposed system is costly as it is more intended for navigation purposes rather than simply avoiding objects. Aside from the complexity and the navigation features provided by this system, it is bulky for simple usage. Moreover, the limited coverage of GPS in buildings makes it a non viable option for indoor navigation.



**Figure 1: Wearable Navigation System Developed by [3]**

Julius Yumin et al. in [8] discussed the implementation of an ultrasonic device in a vision system for visually disabled. The proposed implementation in [8] utilizes the ultrasonic device to measure the distance of surrounding objects and alert the user via a buzzer when approaching an object. The ultrasonic device consists of a receiver and transmitter to measure the distance of near objects. It also utilizes two microcontrollers to manage the manipulation of signals. The implementation in [8] provides simple hardware, but it takes into account only one detection direction (front side); if an object approaches from sides, the device will not detect it. Additionally, this system does not provide the user the range within which the detected object resides. This is important for the convenience and comfort of the user. This is because, in order for the user to take the appropriate decision and act accordingly, the user requires knowing, by some mean, how far the object is. Having this knowledge

helps the user to know the existence of the object which might be a human being. However, if the distance is far (i.e., 3 meters) the user does not have to be alerted, but the information at that range could be of benefit. However, as the range of the object shortens (e.g., 50 cm), the alarm volume should go louder.



**Figure 2: RFID Based Indoor Navigation System Suggested in [4]**

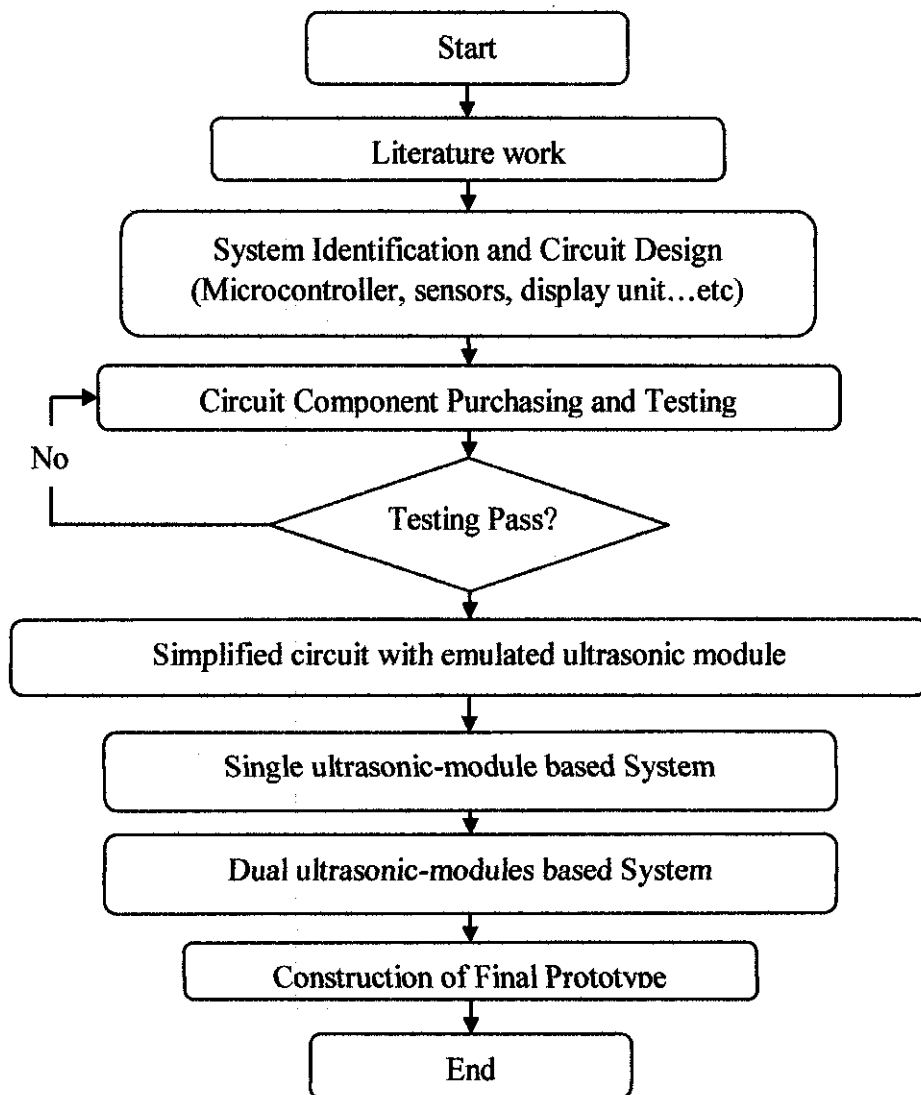
R. Oktem *et al.* in [4] proposed an indoor navigation system based on an RFID concept as illustrated in Figure 2. This method provides a robust technique which enables smooth navigation with the help of a wireless network, an RFID device, a compass, and a PC. Despite its robustness, the extra hardware limits its portability.

Developers in [2], [9], [10], proposed different methods based on ultrasonic devices. Their methods provide reliable object avoidance they only take into account the detection at the front direction. In addition, the proposed systems do not indicate to the user how far the object is. This feature is important to give the user more flexibility and awareness by what surrounds him and how far they are. This could enable the user to have more decisions and actions to take according to the location of the object.

## CHAPTER 3 METHODOLOGY

In this part, the methodology used to realize the project including the tools used is discussed.

### 3.1 Project Development Flow Chart



**Figure 3: General Flow chart of Project work**

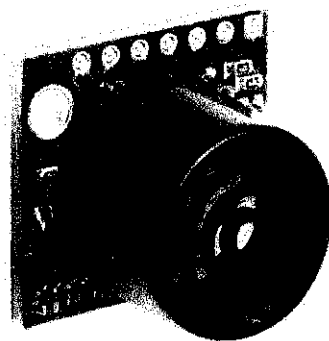
The adopted methodology to realize the project is shown in Figure 3. The key milestones considered for the project are identified and shown in Gantt chart in Appendix A. The methodology is further discussed as follows:

### **3.2 System Identification and Required Tools**

Throughout the development stage of the prototype of the project, several tools are potentially considered.

#### **3.2.1 Ultrasonic module: Max Sonar – EZ1 ultrasonic range finder**

An ultrasonic module which consists of a transmitter, a receiver and a calculation chip is to be used. The Ultrasonic module should provide range reading for the distance of nearest object at its transmission direction. According to the electrical specification and project requirement of the proposed system, the MaxSonar-EZ1 [11] ultrasonic range finder is considered in this project.



**Figure 4: LV-MaxSonar-EZ1 Ultrasonic range finder**

In Figure 4, the MaxSonar-EZ1 ultrasonic range finder is shown. The specification of this module is listed in Table 1. The voltage and current requirements of this module are compatible with embedded applications.

Additionally, the measurement frequency provided by the module is around 20 Hz which enables the availability of a new reading for the distance of every 5 hundredth of a second. Assuming that the critical distance at which the system should sign an alarm is 1 meter, and assuming the time to react and time to stop or change direction take zero second, we can imply that the maximum speed which this module could



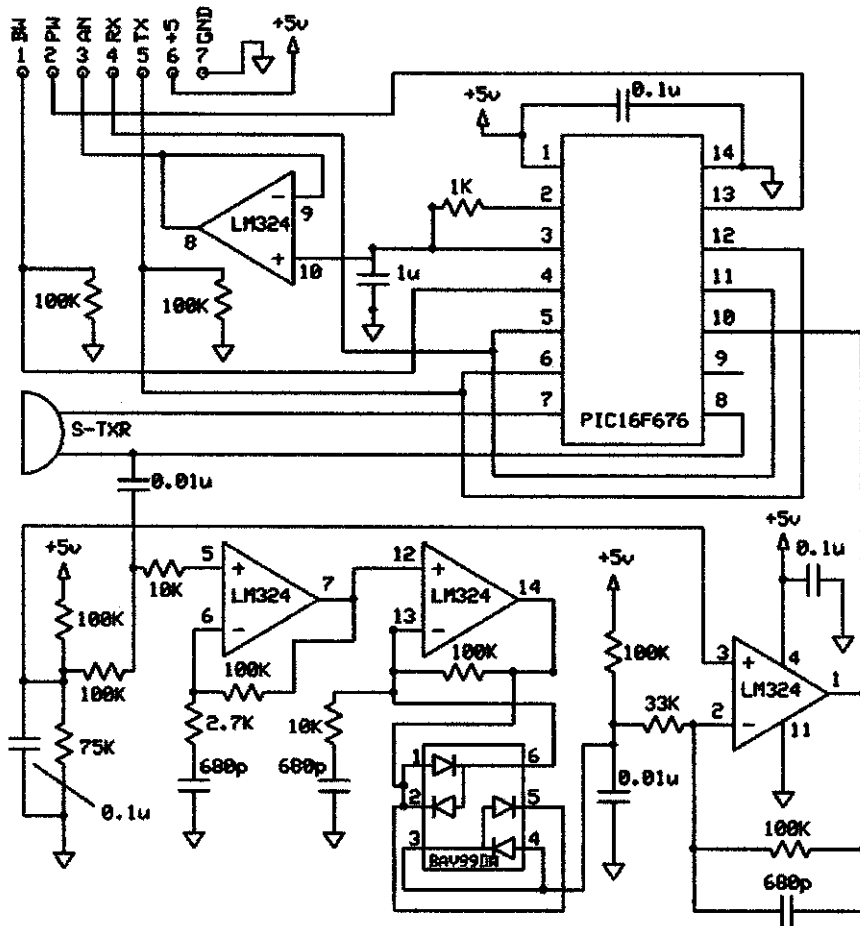
support is around 20 m/s.

**Table 1: LV- MaxSonar-EZ1 Ultrasonic Range Finder Specification**

Item	Value
Ultrasonic signal frequency	42 KHz
Supply Voltage	2.5- 5.5 V
Supply current	2 mA
Reading Rate	20 Hz (0.05 second)
Output Signals	RS232, PWM, Analog
Measurement Range	0 - 6.45 m
RS232 Serial Output	9600 bps, pins: Tx, Rx, V <sub>cc</sub> , Gnd
PWM Output	147uS/inch, pins: V <sub>cc</sub> , Gnd, AN
Analog Output	10mV/inch, pins: V <sub>cc</sub> , Gnd, BW

This measurement speed is more than enough for a normal user. The module can be interfaced via 3 ways (Table 1): RS232, PWM signal, and analog signals. As most microcontrollers have built-in analog to digital converters with multiple channels, the analog signal is used to communicate the measurement to the microcontroller. This allows up to 13 measurement modules to be attached to the system compared to 2 and 1 if PWM and RS232 are to be used, respectively.

The schematic circuitry for the module is shown in Figure 5. It is important to note that the characteristics of the beam signal for the ultrasonic wave show different patterns.



**Figure 5: Ultrasonic Range Finder Schematic**

These patterns are described by the rod diameter (width) of the beam and it is a function of the measured range. Figure 6 shows the four beam shapes A, B, C, and D. Shape A corresponds to short range measurements, while shape D corresponds to long range measurements. From coverage point of view, when the objects get further from the device, it can cover as wide as 11 inches throughout the range of reading. However for short ranges, the beam detects as wide as 0.25 inch only. This can be helpful for the user to cover wide range of area without sacrificing the device efficiency and power. This is because, having a wide beam requires more wave and thus more power.

## Beam Characteristics

Sample results for measured beam patterns are shown below on a 12-inch grid. The detection pattern is shown for:

- (A) 0.25-inch diameter dowel, note the very narrow beam for close small objects,
- (B) 1-inch diameter dowel, note the long narrow detection pattern,
- (C) 3.25-inch diameter rod, note the long controlled detection pattern,
- (D) 11-inch wide board moved left to right with the board parallel to the front sensor face and the sensor stationary. The displayed beam length shows the long-range capability of the sensor.

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

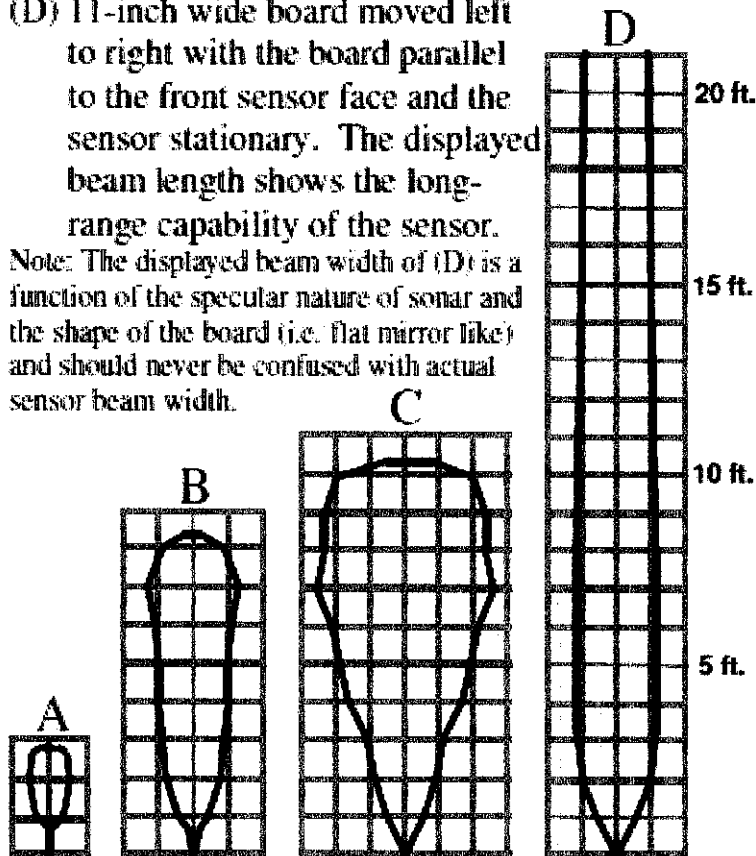


Figure 6: Ultrasonic Range Finder Beam Characteristics

### 3.2.2 PIC18F4550 microcontroller

A mid range microcontroller from Microchip is to be used. This selection enables the developers to deal with the prototype with more flexibility and efficiency.

The PIC18F4550 is among the most commonly used Microchip microcontrollers because of its USB communication support capabilities. The PIC18F4550 is a 40 pin high performance microcontroller which is equipped with several built-in peripherals (Figure 7). Even though the proposed system does not explicitly require the USB support, the selection is made to enable future development and flexibility of

functionality expansion.

Along with the USB support, the microcontroller is featured with different processing modes, configurable internal oscillators, extendable instruction set which makes it a high performance yet power efficient microcontroller. The most important specifications to consider are reported in Table 2. The 32KB flash memory allows long programs (more than 16 thousands assembly code lines) to be executed. The data of variables used along the execution of the program are saved in SRAM memory which is 2KB in size for the PIC18F4550 microcontroller.

Some of the peripherals of the microcontroller are not considered as the proposed system does not require them, however, it is likely that normal I/O operation are to be used instead to allow access to other direct digital transmission based devices such as LCD.

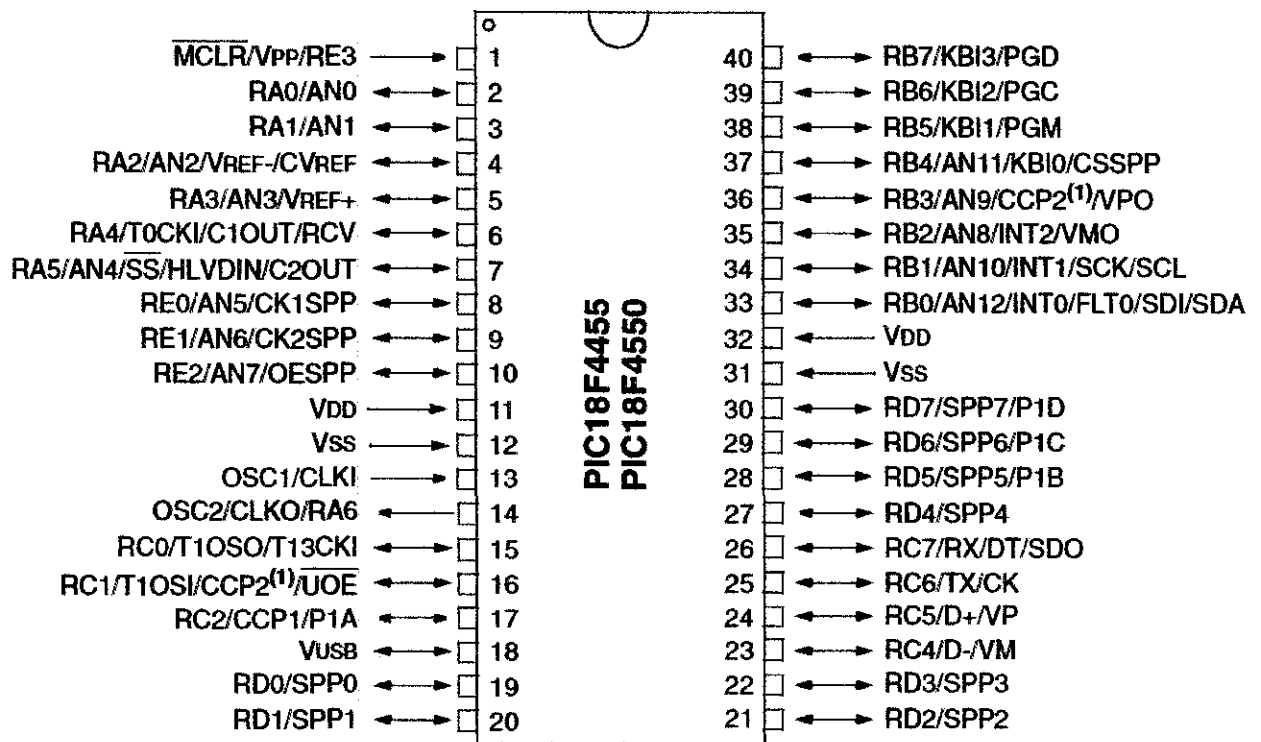


Figure 7: PIC18F4550 Pin Diagram

Table 2: PIC18F4550 Specification

Program Memory	Data Memory	I/O	A/D	CCP/ECCP	SPP	MSSP	EAUSA	RT	Comparators	Timers

Bytes	Instructions	SRAM	EEPROM			PWM		SPI	I2C			
32K	16384	2048	256	35	13	1/1	Yes	Yes	Yes	1	2	1/3

### 3.2.3 CCS C compiler

In order to program the microcontroller, a compiler is to be utilized. In this project, the PIC C compiler from CCS is to be used.

The CCS C compiler is easy to use, and almost immediate to get started due to the project wizard feature and the different startup codes which it offers.

Among the features of CCS C compiler:

- Automatic fuses configuration
- Extensive built-in functions providing direct access to PIC hardware
- Extensive source code driver library
- Arithmetic library
- Integrated development environment

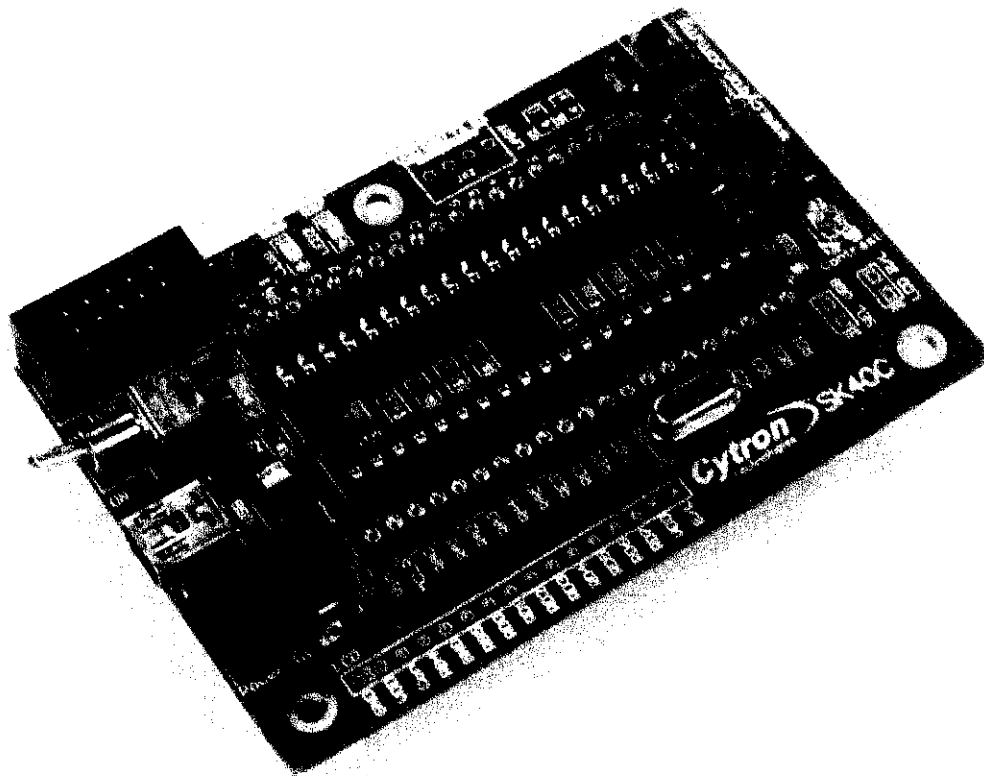
### 3.2.4 SK40C PIC development kit and UIC00B programmer

To speed up the development phase, a startup kit is used as the platform of the microcontroller circuit. The use of this tool provides easier and more robust circuit to be built. The board in Figure 8 provides several functionalities and circuitry support.

The kit is a robust development platform which offers:

- Voltage regulation circuitry (9 V input voltage to 5 V)
- Reset button
- USB port
- Connector to programmer
- 2 push buttons and 2 LEDs connected to Port B

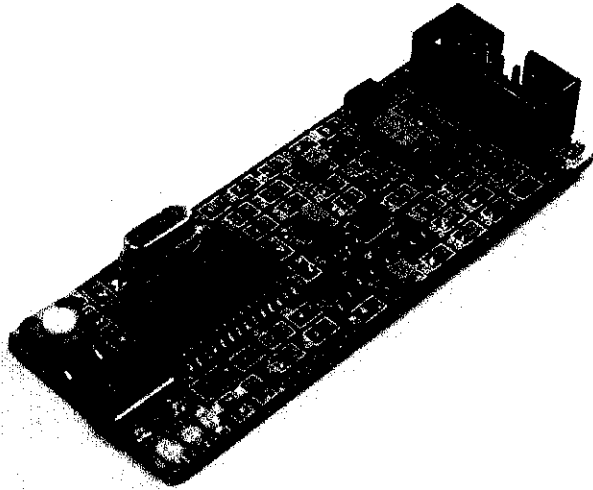
- 20 MHz crystal oscillator
- Optional connection to LCD and UART
- Optional jumpers to enable the back light of the LCD and the USB port
- Potentiometer to adjust the contrast of the LCD



**Figure 8: SK40C PIC Start-up Kit**

The pins of the SK40C board are labeled clearly as shown in Figure 8. This assists in speeding up the prototyping and reduces the errors in connecting to external circuitries.

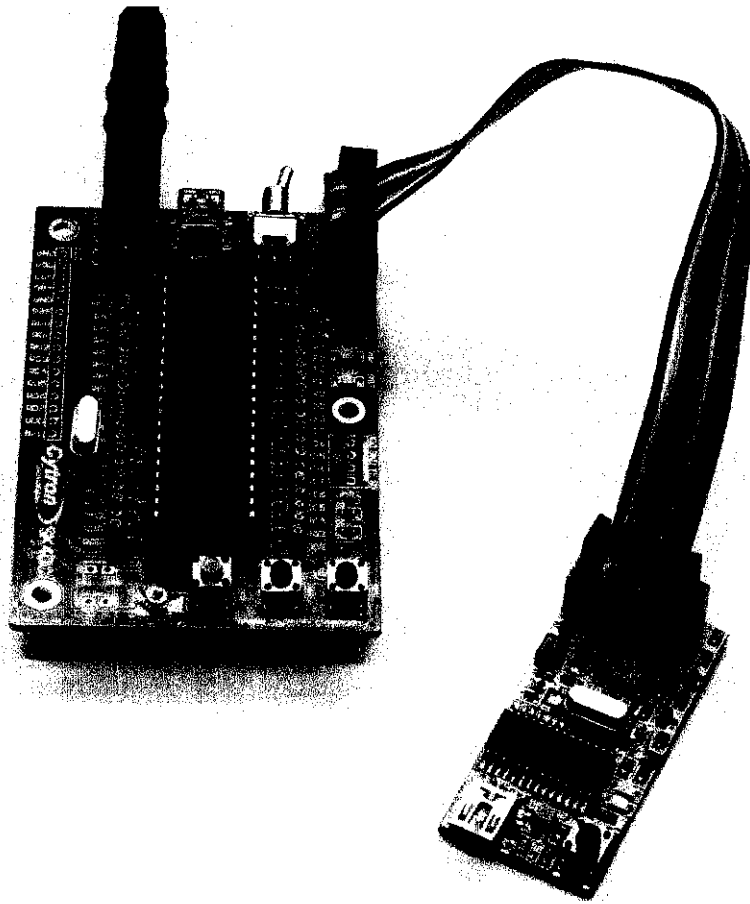
In order to transfer the C codes to the program memory of the microcontroller, USB ICSB programmer (UIC00B) is considered (see Figure 9). This programmer offers a cheap programming solution and is compatible with the SK40C startup kit.



**Figure 9: UIC00B USB ICSP PIC Programmer**

The programming software (PIC kit 2) takes the hex file which is produced by CCS compiler and loads it to the microcontroller memory via UIC00B programmer. The interface between the PIC board and the programmer is shown in Figure 10.

The programmer has two sockets; A USB socket to be connected to a PC and serial socket to be connected to the board.



**Figure 10: PIC Board and Programmer Connection**

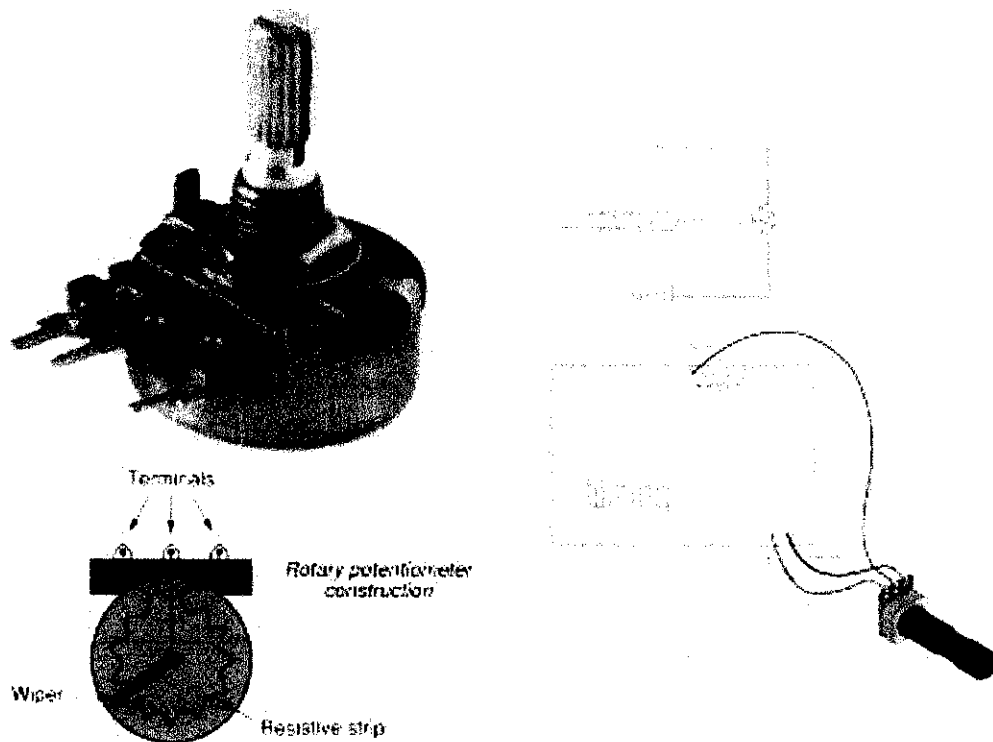
### **3.3 Hardware Development**

In order to realize the required system, the hardware of the proposed system is to be developed in several steps to ease the process.

#### ***3.3.1 Simplified circuit with emulated ultrasonic module***

In order to realize the circuit of the proposed ultrasonic vision system, the ultrasonic module is emulated by simple potentiometers connection. The configuration allows the voltage to vary in a range of 0-5V. This simulates the typical output of ultrasonic device as per specification described in Table 1. The potentiometer and its equivalent circuitry are shown in Figure 11.





**Figure 11: Potentiometer Emulating the Output of Ultrasonic Range Measurement Module**

In an actual circuit connection, 5V is to be connected to Vcc of development board and by varying the position of knob a 0-5V voltage can be obtained.

### ***3.3.2 Single, dual ultrasonic-module based system***

After a simplified circuitry is made, the potentiometer is replaced by an ultrasonic module. Upon the success of a single ultrasonic based vision system, additional module is to be added to the system. As the microcontroller allows up to 13 analog modules to be added, two modules would not cause any circuitry overhead to the system. This procedural way of interfacing is adopted to enable robust final product and bug free circuitry.

### ***3.3.3 Construction of final prototype***

The final prototype is expected to be small enough to be located at the head level of the end user. The placement is proposed to be on a cap-like gear to be worn by the user.

### 3.4 Software Development

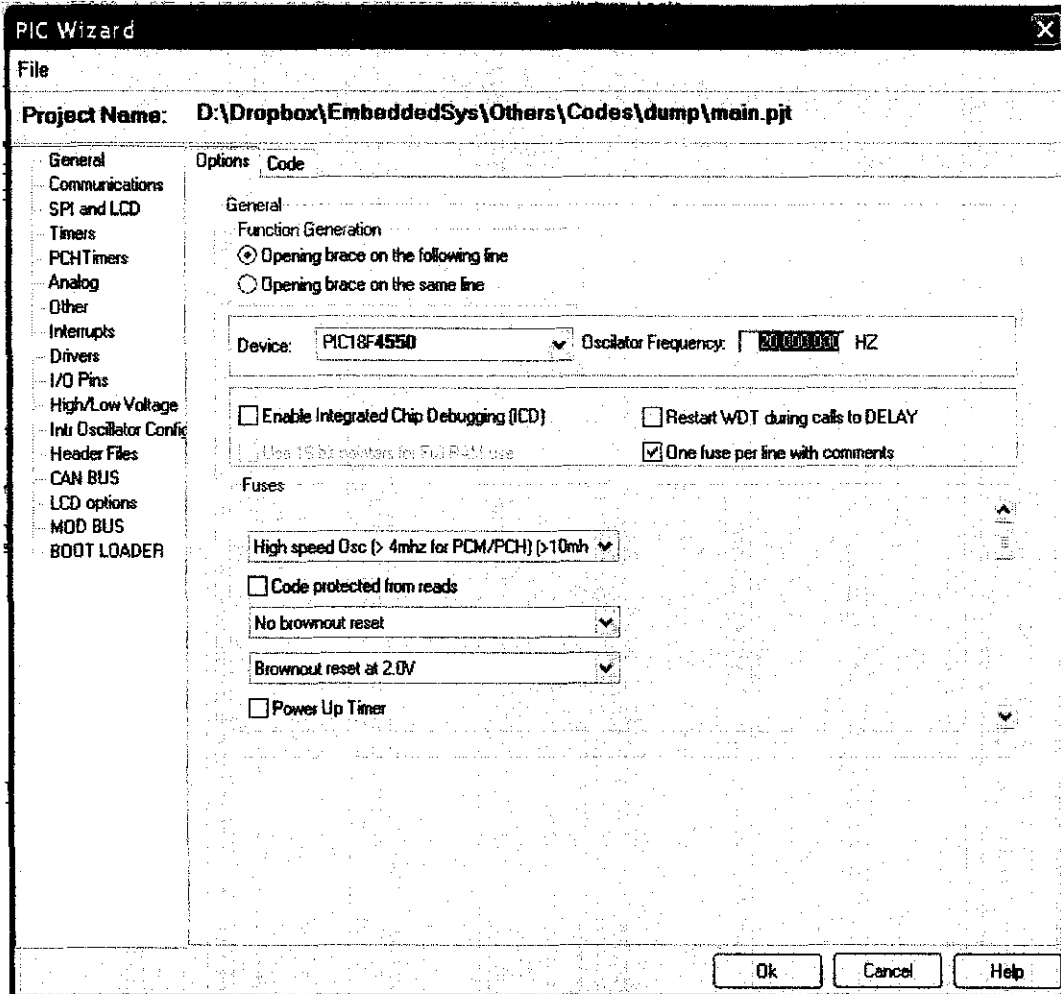
The software development considers all the necessary programming work and the relevant programming tool suite which are needed to realize the project. In this section, firstly the general methodology of PIC programming and the environment required for that is discussed. Secondly, the methodology adopted to implement the various requirement of the system is shown.

#### 3.4.1 PIC programming methodology

Several steps are required to program the developed C codes on the target board. The steps involve the following:

- **Project Creation using project wizard:** In CCS C compiler the best way to develop code is to utilize a project wizard. The project wizard provides a good utility which automatically generates startup codes with the proper fuses (configuration) setting, #include files, peripheral setups and main function. This is very useful for beginners and produces very robust codes. The first step in coding is to set up the PIC correctly. The project wizard is used and the following selections are made:
  1. Selecting PIC18F4550 (as it is the target PIC)
  2. Oscillator frequency: 20,000,000 Hz
  3. For the fuses, the following configurations are chosen:
    - a. High speed Osc (> 4 MHz, for PCM/PCH) (>10 MHz for PCD)
    - b. Unselect PORTB pins are configured as analog input channels
  4. For the analog configuration, the following is used:
    - a. A0,
    - b. Range 0-Vdd
    - c. Units: 8 bits conversion (0-255)

d. Internal 2-6u for the clock



**Figure 12: Project Wizard**

The project wizard is shown in Figure 12. In this wizard, all the settings of the microcontroller peripherals can be set and a corresponding start-up code will be generated. The peripheral settings mainly include: general (for the device type and fuses), communication (RS232, I2C, CAN and SPI), timers, interrupts, oscillators and I/O pins.

In addition to the hardware aspect, the settings include software development aspects such as: drivers (to support external devices e.g., LCD, keypad and EEPROM), header files (e.g., ctype.h, string.h and stdio.h) and boot loader. The software and

hardware setting utilities significantly ease the development process, reduce typo-related errors and require less awareness by the lowest level of the PIC peripheral.

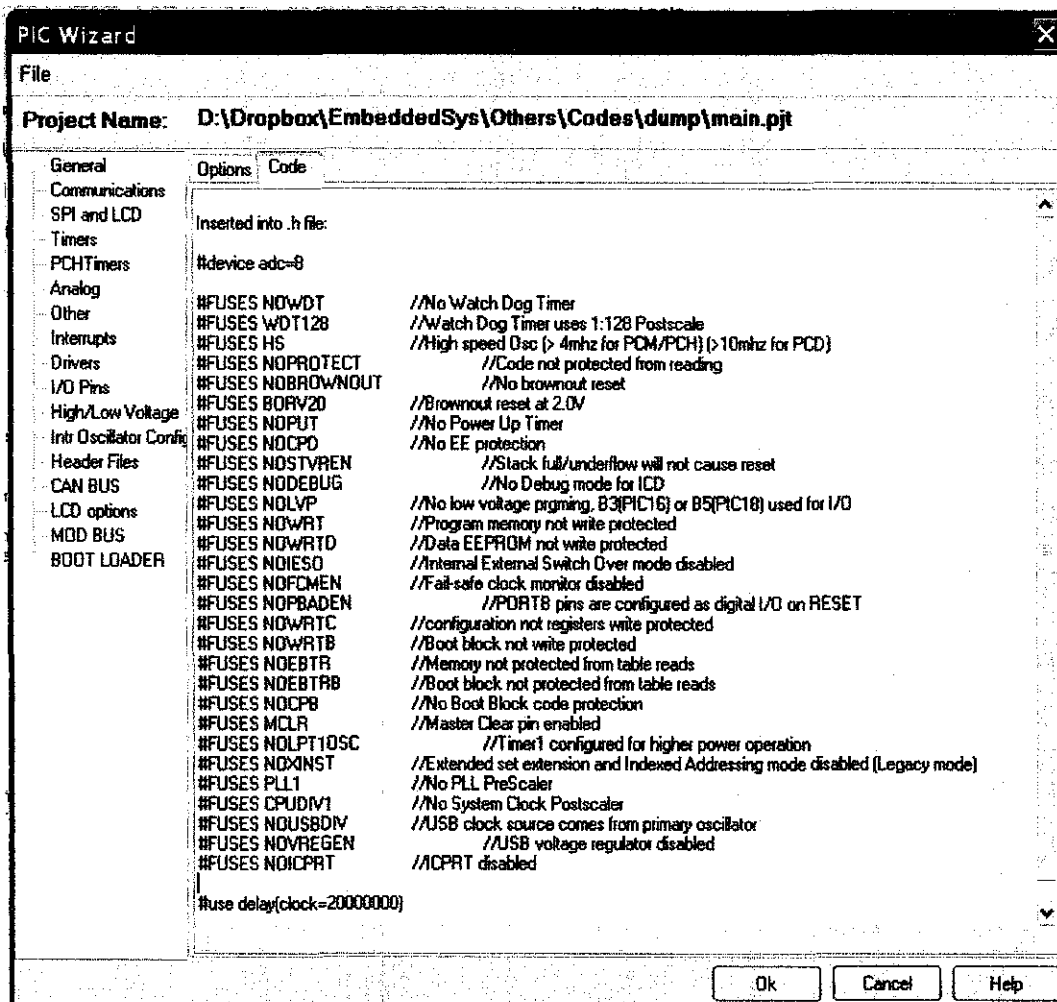
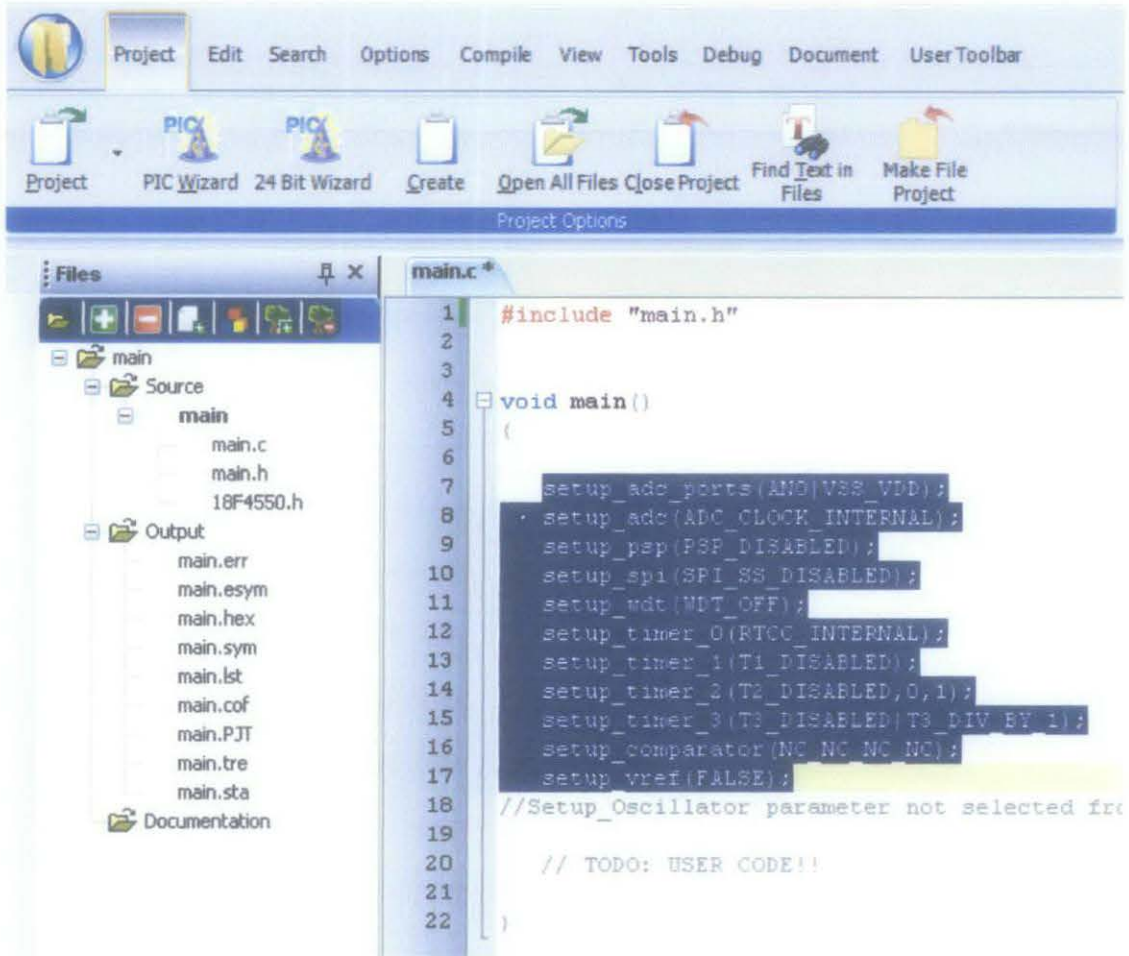


Figure 13: Code generation using PIC wizard for the fuses

Figure 13 shows the generated code for the fuses.

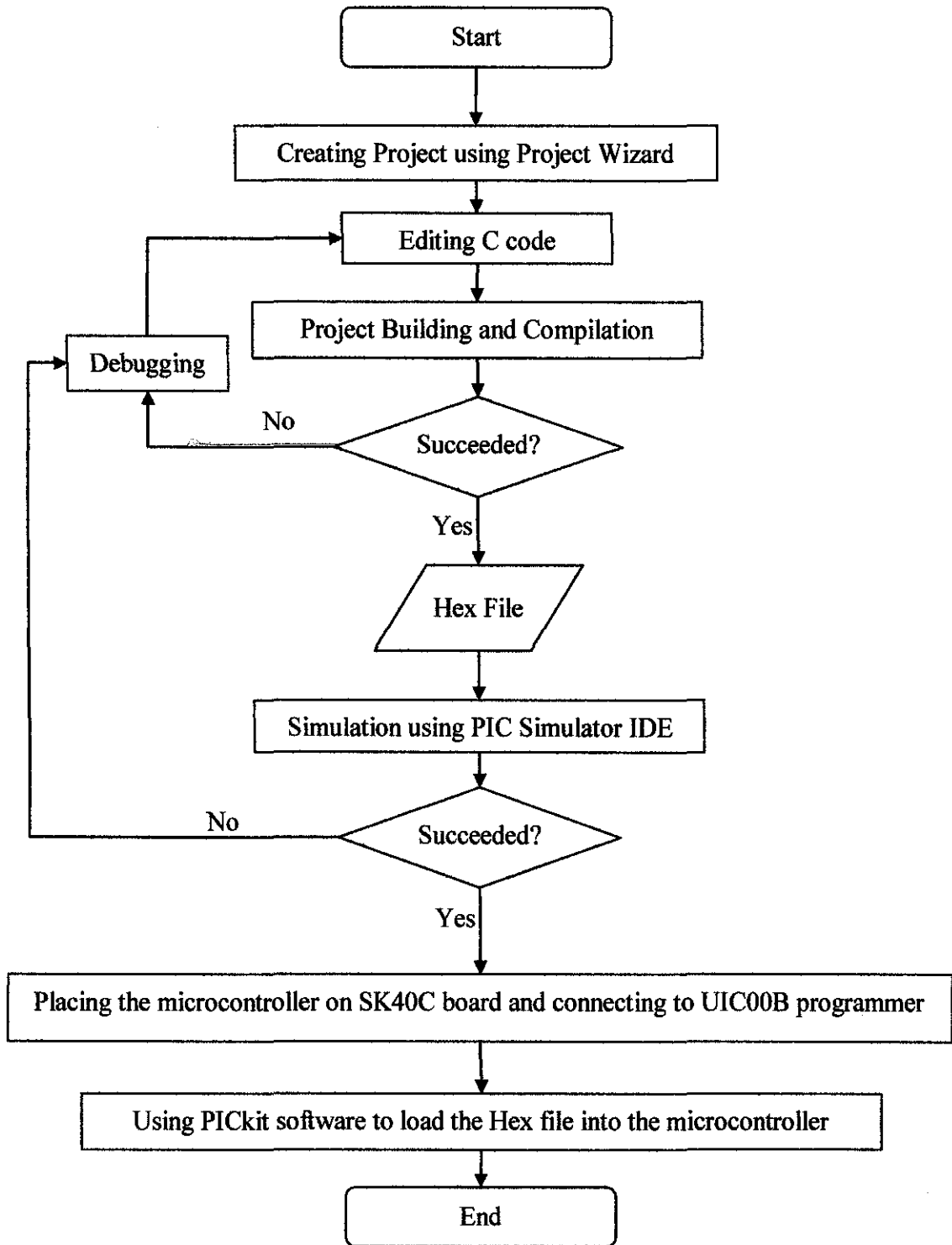
- **C Code edition:** This is where the developer writes the actual code by editing the main function and adding more functions according to the needs of the application. The CCS C syntax follows the syntax of the standard ANSI C to a good extent.

The CCS C compiler automatically generates calls back to some built in functions in main( ) according to the setup as explained previously and highlighted in Figure 14.



**Figure 14: Plain code with automatic generated function calls according to the setup**

- **Building and Compilation:** The C codes remain as C codes until they are built and compiled by the compiler to produce the machine codes and hex file specifically. In this step, the compiler checks the codes for syntax errors and report to the programmer if any exists.
- **Debugging:** This is where the errors are rechecked and corrected by the programmer.
- **Simulation:** Before loading the produced machine code (by the C compiler), it is worthy to see how the PIC would work when the code is to be loaded to the microcontroller. PIC simulator IDE offers realistic simulation interface whereby all the inputs and outputs can be monitored. Input pins can be easily stimulated and several configurable output devices are available.



**Figure 15: Full Microcontroller Programming Flow**

- **Preparing the system for program loading:** The microcontroller has to be connected to the programmer (UIC00B) via certain pins as shown in the respective datasheet. In our case, fortunately the pins are accessible to the programmer via a dedicated connection port and cable as shown in Figure 10.

The programmer has to be known for the PC hosting the PICkit software by proper installation as described in the respective installation manual. The UIC00B programmer is USB device.

**Loading the Hex file to the microcontroller:** The PICkit is used to load the hex file to the microcontroller. After this step the system is ready to run and the programmer can be disconnected from the board as the microcontroller does not require the connection anymore, and in fact it may cause some malfunction to the circuit.

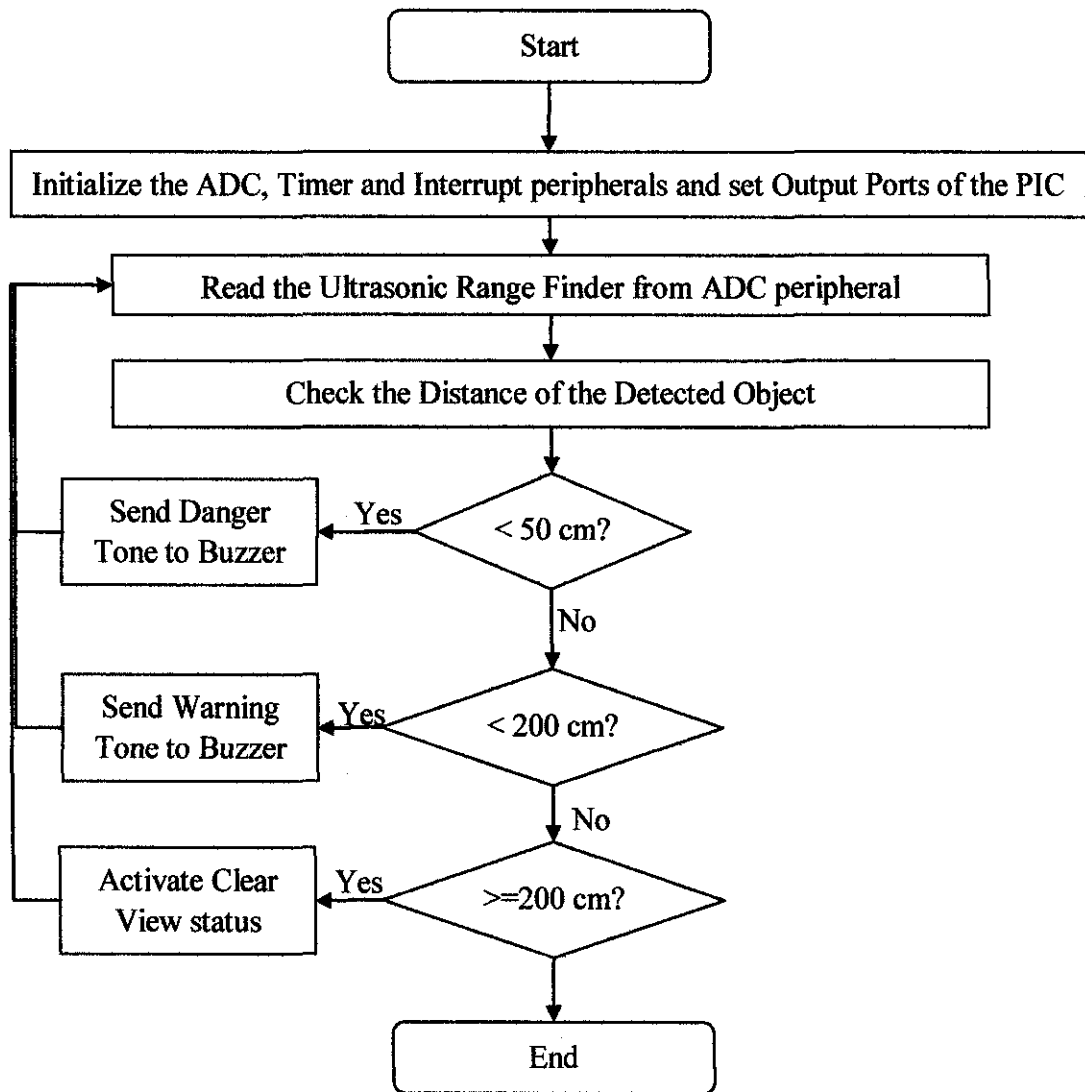
After the circuitry connection is made, the software implementation is initiated using a CCS C compiler. The development of the code requires some knowledge of the hardware specifically the crystal oscillator frequency and the inputs and outputs pins used in the circuit and the respective required function for each pin. The following steps explain the typical code and the considerations taken at each step.

#### ***3.4.2 Overview of the algorithm and code***

To implement the various functions of the vision system, the PIC is required to continuously perform the following:

1. Read the sensor value
2. Convert sensor readings to known units
3. Show the status of the detected objects on the LEDs as follows:
  - a. DANGER: by activating a RED LED at PIN B7 and constantly activating a buzzer.
  - b. WARNING: by activating an ORANGE LED at PIN B6 and alternately activating a buzzer.

CLEAR VIEW: by activating a GREEN LED at PIN B2



**Figure 16: Ultrasonic Vision System Algorithm**

The proposed algorithm and procedural steps are shown in the flow chart in Figure 16. The flow chart shows the necessary steps required to realize the system. The PIC has several peripherals which can work in different ways according to how they are configured. Therefore, to ensure the system will utilize the correct peripherals and at the right settings, the initialization step has to be performed at the early start of the system.



After the proper settings have been configured the system goes to read the ultrasonic module which is interfaced to the microcontroller via its analogue input pins. However, the output of the ADC peripheral is always a digital value in the range of 0 to 255, if configured to work in 8 bit mode, corresponding to range of VSS (0 V) to VDD (5 V). Without referring to how the ultrasonic module work as in the manual, neither the numerical value nor the voltage tell a distance, therefore, the obtained values are adjusted to be used in the other subsequent operations.

The system then checks the distance of the detected object and activates special visual and audible outputs accordingly. The detection range of the sensor is 645 cm. This detection range is arbitrary divided into three zones or sub ranges. The first zone is the critical one to the user, whereby an object is detected at range less than 50 cm. This range is referred as in this work as “dangerous” zone. Considering a navigating scenario, in this zone the user is expected to take an immediate action to prevent collisions. The second zone is the “warning” zone. This zone is defined between 50 cm and 200 cm of distance between the user and the detected object. In this zone, the system sends different audible output (for the user) and visual output (for diagnosis purposes) to that sent in the “dangerous” zone. The third zone is “clear view” zone. The detection distance of this zone is for greater than 200 cm to the maximum detection range (i.e. 645 cm). In this zone the system activates visual output (i.e. LEDs), for diagnosis purposes, and does not activate any audible output.

The system is set to continuously work in a closed loop of reading the distance of the detected object if any and set the status and activating the outputs accordingly. This continuous monitoring and alerting enables object avoidance and thus safe navigating.

Additionally, since the system is proposed for the use of visually impaired users, the system is provided by special feature which is an audible system monitor. The aim of such feature is to acknowledge to the user that the system is healthy and running. This is because; when the system relies on a battery, it can run out and stop the functionality of the system without the user being informed or alerted. The audible monitoring feature periodically sends a special audible tone to tell the user that the system is running. The technical implementation details are discussed in later parts of this section.

```

#include "main.h"

void PIC_init    ();
void read_range  ();
void range2cm    ();
void show_status ();
unsigned char range;
int16 range_cm;

#define DANGER_PIN_B6      //DANGER LED is connected to PIN B6
#define WARNING_PIN_B7    //WARNING LED is connected to PIN B7
#define CLEAR_VIEW_PIN_B3 //CLEAR_VIEW LED is connected to PIN B3
#define BUZZER_PIN_B2     //BUZZER is connected to PIN B2

void main()
{
    PIC_init();
    while (TRUE)
    {
        read_range();
        // range2cm();
        show_status();
    }
}

```

**Figure 17: System Program Highlighting the Indluced header files, Used Functions, Compiler Directives and the main() Function**

To realize the procedures as shown in Figure 16 and explained previously, a modular code is developed as highlighted in Figure 17. This figure shows the core functions and compiler directives considered for the implementation. The developed code is discussed and explained thoroughly as follows:

### 3.4.2.1 Header file: main.h

```
#include <18F4550.h>
#device adc=8

#FUSES NOWDT           //No Watch Dog Timer
#FUSES WDT128         //Watch Dog Timer uses 1:128 Postscale
#FUSES HS             //High speed Osc (> 4mhz for PCM/PCH) (>10mhz for PCD)
#FUSES NOPROTECT     //Code not protected from reading
#FUSES NOBROWNOUT    //No brownout reset
#FUSES BORV20        //Brownout reset at 2.0V
#FUSES NOPUT         //No Power Up Timer
#FUSES NOCPD         //No EE protection
#FUSES NOSTVREN      //Stack full/underflow will not cause reset
#FUSES NODEBUG       //No Debug mode for ICD
#FUSES NOLVP         //No low voltage prgming, B3(PIC16) or B5(PIC18) used for I/O
#FUSES NOWRT         //Program memory not write protected
#FUSES NOWRTD        //Data EEPROM not write protected
#FUSES NOIESO        //Internal External Switch Over mode disabled
#FUSES NOFCHEN       //Fail-safe clock monitor disabled
#FUSES NOPBADEN      //PCORTE pins are configured as digital I/O on RESET
#FUSES NOWRTC        //configuration not registers write protected
#FUSES NOWRTB        //Boot block not write protected
#FUSES NOEBTR        //Memory not protected from table reads
#FUSES NOEBTRB       //Boot block not protected from table reads
#FUSES NOCPB         //No Boot Block code protection
#FUSES MCLR          //Master Clear pin enabled
#FUSES NOLPT1OSC     //Timer1 configured for higher power operation
#FUSES NOXINST       //Extended set extension and Indexed Addressing mode disabled
#FUSES PLL1          //Divide By 1
#FUSES CPUDIV1       //System Clock by 1
#FUSES NOUSBDIV      //USB clock source comes from primary oscillator
#FUSES NOVREGEN       //USB voltage regulator disabled
#FUSES NOICPRT       //ICPRT disabled

#use delay(clock=20000000)
```

Figure 18: The compiler directive to set fuses in the *main.h* header file

The PIC configuration words are the most influential settings to how the PIC would work at the system level. The configuration words primarily control some switches inside the microcontroller according to the oscillator, supply voltage, special peripherals, programming and debugging requirements. Different compilers have different ways of setting the configuration words. In CCS C compiler the configuration words are mainly set by #fuses and #device directives. The generation of the code can automatically be generated exploiting the project wizard facility. Figure 18 shows the generated code by the project wizard for the proposed system. Interestingly, the code is automatically generated with comments. This code actually sets the microcontroller to use 8 bit ADC conversion result, 20 MHz oscillator frequency and the master clear pin while disabling the other features. By these lines, the configuration words are set and the PIC is ready to use the peripherals and execute commands and instructions.

### 3.4.2.2 PIC\_init( )

```
// device_init() function used to initialize the PIC
void PIC_init()
{
    setup_adc_ports(AN0|VSS_VDD); //use A0 as analog input, input range: VSS-VDD
    setup_adc(ADC_CLOCK_INTERNAL); //use internal clock for conversion
    setup_psp(PSP_DISABLED); //Disable PSP
    setup_spi(SPI_SS_DISABLED); //Disable SPI
    setup_wdt(WDT_OFF); //Disable watchdog timer
    setup_timer_0(RTCC_INTERNAL);
    setup_timer_1(T1_DISABLED); //Disable timer0, timer1
    setup_timer_2(T2_DIV_BY_16,195,16); //set timer2 with 10 ms overflow
    setup_comparator(NC_NC_NC_NC); //Disable comparators
    setup_vref(FALSE);
    enable_interrupts(INT_TIMER2); //Set timer2 to flag interrupt when overflows
    enable_interrupts(GLOBAL); //Enable interrupts
    set_tris_b(0x00); //setting port B as output (for buzzer and LEDs)
}
```

**Figure 19: Initialization function**

Initializing the PIC to work as required is as important as the core program. This is because; the PIC has wide range of peripherals multiplexing sharing the same pins. This requires careful programming and therefore highlights the significance of this work. The initialization code is listed in Figure 19.

The code in Figure 19 essentially enables and disables peripherals according to the requirement. Since the proposed system utilizes the ADC for reading the sensor, timer 2 to generate the necessary timing, port B to communicate with buzzer and LEDs. These peripherals are consequently enabled. The other unused peripherals (i.e. watchdog timer, PSI, PSP, comparators, etc.) are disabled.

### 3.4.2.3 read\_range( )

```
void read_range()
{
    set_adc_channel(0);
    delay_us(60);
    range=read_adc();
}
```

**Figure 20: read\_range() function**

This function reads the distance of the detected object by reading from the ADC module. Since the sensor is attached to PIN A0, `set_adc_channel(0)` is used. The `read_adc()` function is called to read the conversion results from ADC peripheral. A delay of 60 us is required for the conversion to settle. Figure 20 shows the implemented function.

#### 3.4.2.4 `range2cm()`

```
//range2cm converts the reading to cm by considering that
//127 reading corresponds to 645cm according to the manual
void range2cm()
{
    range_cm=range*645/127;
}
```

**Figure 21: `range2cm()` function**

The reading of the sensor is converted to cm using this function as shown in Figure 21. From the datasheet of the sensor [11], the appropriate mathematical equation is formed and used in the function.

### 3.4.2.5 *show\_status()*

```
void show_status()
{
    //if ( ( range_cm < 50 ) ) //can be used alternatively
    if ( ( range >= 0 ) && ( range < 10 ) )
    //LED=DANGER (RED), Buzzer always ON
    {
        output_low (WARNING);
        output_high (DANGER);
        output_low (CLEAR_VIEW);
        output_high (BUZZER);
    }
    else
    //if ( ( range_cm > 50 ) && ( range_cm < 150 ) )
    if ( ( range >= 10 ) && ( range < 29 ) )
    //LED=WARNING (ORANGE), Buzzer ON 100ms/OFF 100ms
    {
        output_high (WARNING);
        output_low (DANGER);
        output_low (CLEAR_VIEW);
        output_low (BUZZER);
        delay_ms(100);
        output_high (BUZZER);
        delay_ms(100);
    }
    else
    //if ( ( range_cm > 2 ) )
    if ( ( range >= 29 ) )
    //LED=CLEAR VIEW (GREEN), Buzzer ON 100ms/OFF 100ms
    {
        output_low (WARNING);
        output_low (DANGER);
        output_high (CLEAR_VIEW);
        output_low (BUZZER);
    }
}
```

**Figure 22: show\_status() function**

The status of the reading is implemented using this function as shown in Figure 22. The function is implemented using if else statements and the three levels of output are activated using `output_low()` and `output_high()` functions.

### 3.4.2.6 System monitor

```
#int_TIMER2
void TIMER2_isr(void) //10 ms overflow
{
    if(++time_elapsed==2000) //20 sec
    {
        disable_interrupts(INT_TIMER2);
        output_high (BUZZER);
        delay_ms(20);
        output_low (BUZZER);
        delay_ms(20);
        output_high (BUZZER);
        delay_ms(20);
        output_low (BUZZER);
        delay_ms(20);
        time_elapsed=0;
        enable_interrupts(INT_TIMER2);
    }
    clear_interrupt (INT_TIMER2);
}
```

**Figure 23: System Monitor Code Utilizing Timer 2 Interrupt Service Routine**

The target user of the vision system is visually impaired people. When the system is out of battery, the user will be at the risk. Even though, the system is provided by visual indicator (LED) to show the status of the power supply (battery), the target user can not see any visual indication. Therefore, the monitoring system is proposed in the prototype. The idea is to periodically play a special tone which its purpose is to tell the user that the system is healthy.

To implement such concept without affecting the way the system work, the timer peripheral is utilized. In the PIC microcontroller, there're three commonly used

timers, timer 0, 1 and 2. Timer 0 is normally used as real time clock, timer 1 is sometimes used as an internal clock and low power counter for the PIC. Therefore, timer 2 is chosen to implement such feature.

With the help of the project wizard utility, timer 2 is empirically set such that it set an interrupt flag every 10 ms at 20 MHz oscillator frequency. Once the interrupt occurs, the program counter inside the PIC goes directly to timer 2 interrupt service routine, where the system monitor code is located and as shown in Figure 22. Once the timer overflows the mentioned function is executed. As shown in the figure, the interrupt service routine mainly increments a counter variable called **time\_elapsed** and check if its value is equal to 2000. When **time\_elapsed** is 2000 which implies that 20 second has been elapsed (i.e.  $2000 \times 10$  ms), the PIC activates the buzzer with a double switches ON/OFF to indicate special alert that the system is running.

### **3.5 Summary**

This section identifies the necessary components and tools to be used. Additionally, to realize the project, the hardware and software development aspects are discussed extensively. All development steps are followed as described in this section. The codes are attached in Appendix B and C, and the results obtained are analyzed and discussed in the next chapter.



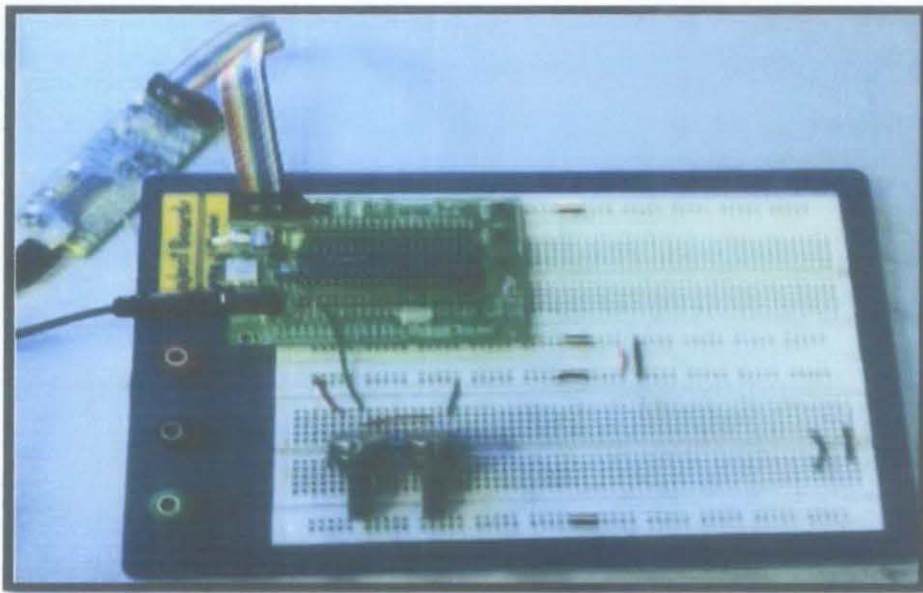
## CHAPTER 4

### RESULTS AND DISCUSSION

This chapter presents the results obtained and some analysis and discussion on the results.

#### 4.1 Simplified Ultrasonic Module Emulation

An initial experiment to interface the PIC to external peripherals is carried out. In this experiment, a simplified circuit consisting of two potentiometers is constructed and connected to the analog ports (RA0 and RA1) of the PIC18F4550 as shown in Figure 24.



**Figure 24: Simplified Vision System with Potentiometers Emulating Ultrasonic Modules**

A C code is written and loaded to the microcontroller. The program reads the analog value at port A. At each loop the program compares the read value with a threshold value. Two LEDs connected to port B indicate the status of the analog values as follows:

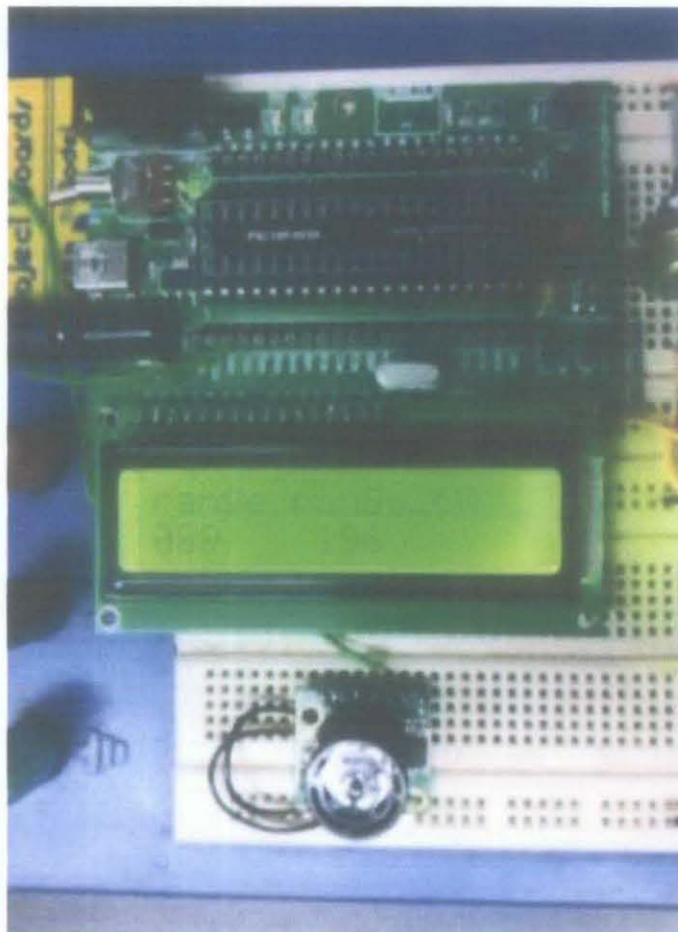
- Orange LED is ON if the analog input is more than the threshold value. This indicates a normal condition where the sensor does not detect near objects.
- Red LED is ON if the analog input is less than the threshold value. This is indicating an alarm condition in which the sensor is detecting a near object.

The threshold value is set in this experiment to 100. This value corresponds to 2V. In the typical visual system, the threshold value would correspond to the maximum allowed distance from the nearest detected object. In later settings, this distance is set to be within 1 meter which is discussed in the coming sections.

Moreover, the LEDs are to be replaced potentially by buzzer to provide sound based alarm instead of visual based alarm.

#### 4.2 Interfacing Ultrasonic Range Finder

An important step is to be able to get instantaneous readings from The MAX range finder.



**Figure 25: The Raw Value and the Value in cm for Distance Range**

In this experiment, the PIC is successfully interfaced with the ultrasonic distance module as shown in Figure 25. The figure shows the board, the LCD and the ultrasonic range finder module. The reading of the sensor is captured continuously via the analog input pin (A0). The PIC reads the sensor value from the ADC peripheral and the value at that stage is stored in an 8 bit integer number (0-255) representing 0-5 V range.

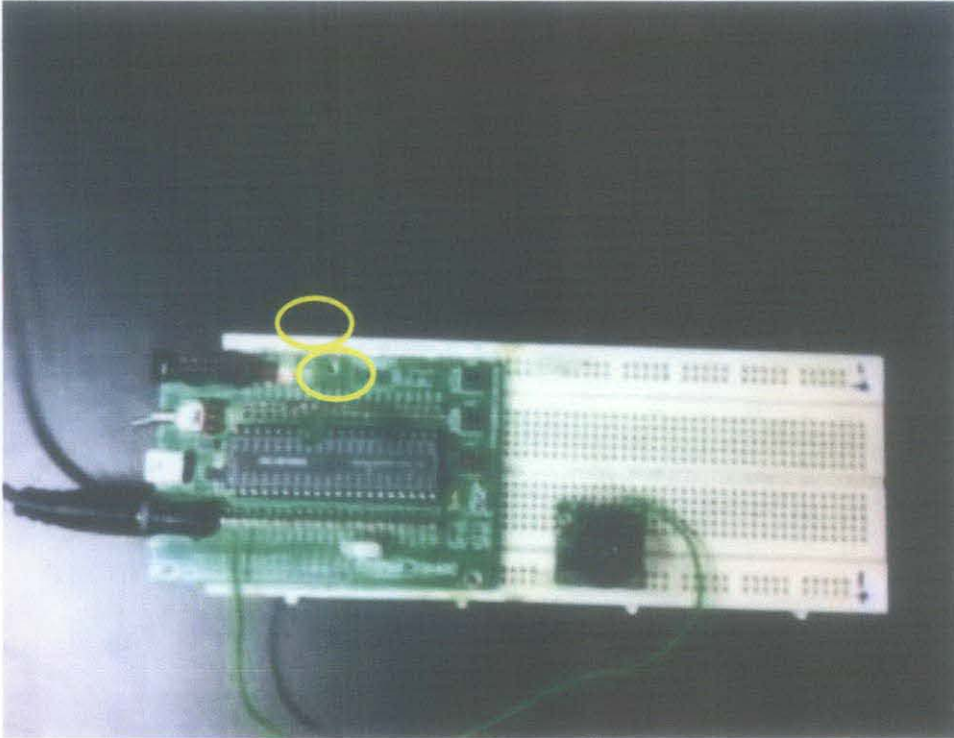
From the specifications of the ultrasonic module, it is illustrated that the maximum range value is 2.5 V which corresponds to 127.

This is important to determine and display the correct distance measurement. From the datasheet, it is given that the maximum output of the sensor is 127 which correspond to 6.45 meters. This knowledge is used to convert the value of the input to cm, inch and meters accordingly as shown on the LCD.

The sensor is tested with different object distances and its functionality is confirmed correct.

### **4.3 Ultrasonic Vision System with Three LEDs Alarm Levels**

In this experiment we need to test the main alarm concept of the project. To do that, the PIC is interfaced with the ultrasonic range finder module. The PIC reads the input and gives alarm via the three LEDs according to the range of the detected object. This is show in Figure 26.



**Figure 26: Ultrasonic Module with LEDs Alarm indicators**

The alarm level is set as following:

1. RED (DANGER or near object)

The distance is less than 1 meter.

2. Orange (WARNING or object detected):

The distance is greater than 1 meter and less than 3 meters.

3. Green (CLEAR VIEW):

The distance is greater than 3 meters.

The RED status is indicated by a red LED connected to PIN B7, while the ORANGE status is indicated by an orange LED connected to PIN B6; whereas the GREEN status is indicated by a green LED connected to PIN B3.

The red and orange LEDs are built-on the board and smaller in size that is why it is not well clear. The green LED is normal LED. The LEDs are highlighted by yellow circles as shown in Figure 26.

#### 4.4 Ultrasonic Vision System Prototype with Buzzer Alarm

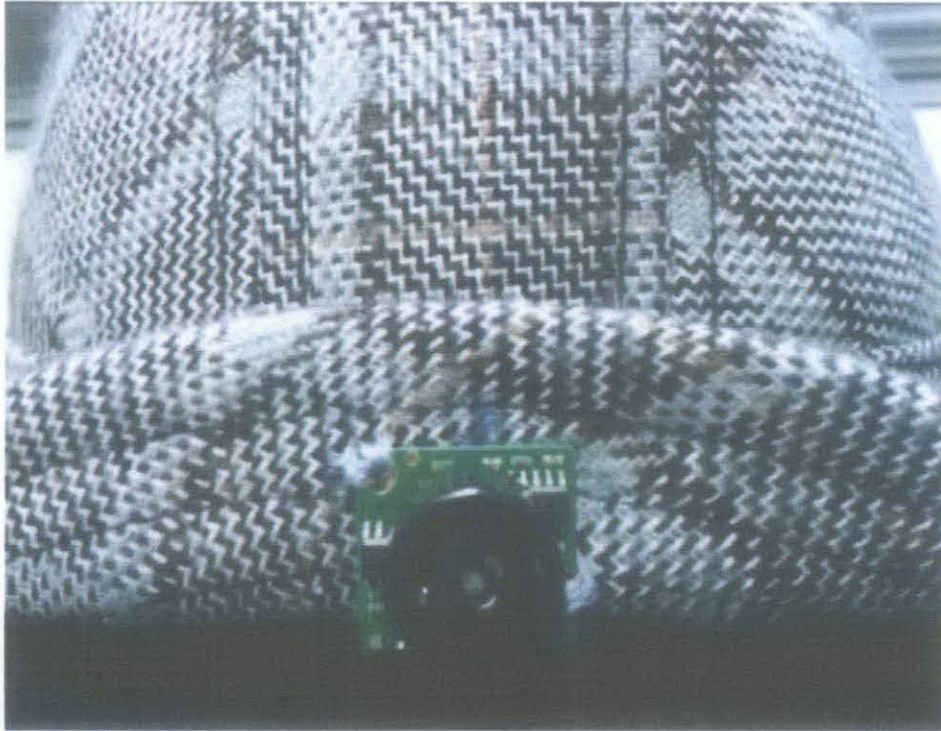


**Figure 27: Ultrasonic Vision System Prototype**

Due to the promising experimental results discussed previously, a prototype for the system is constructed. To ensure the comfort of the user, a hat shaped wearable cap is considered for implementation as shown in Figure 27, Figure 28, Figure 29, and Figure 30. The material of the hat is chosen such that it can accommodate the hardware of the system without the burden of unnecessarily carrying extra accessories.

The items used in this implementation were: ultrasonic range finder, buzzer and a mother board. To aid the portability of the system, a nine volt battery module can be used and located beside the mother board. A top view for the prototype is described in Figure 27.

The location of the sensor is chosen to face the expected objects at the front as well as to enable large angle coverage for object detection. Figure 28 shows the location and fixture of the sensor. The system is highly extendable to use up to 8 range finders on Ports A and E without excessive efforts.



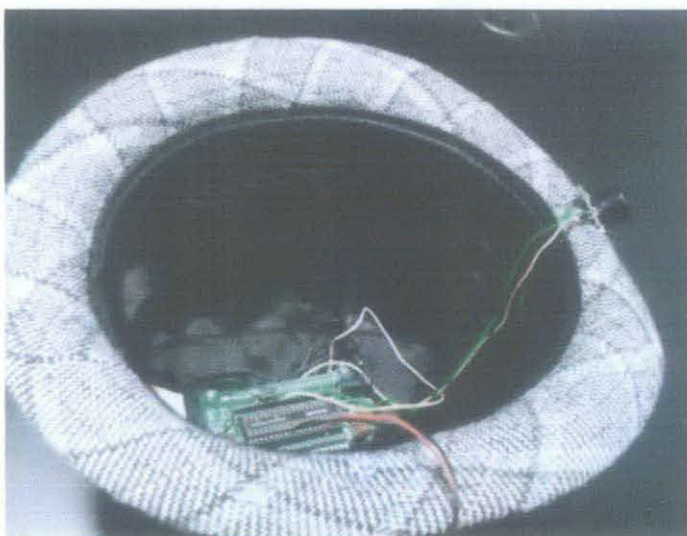
**Figure 28: Ultrasonic Vision System (Range Finder Unit)**

The buzzer is located at the left side of the prototype. This intuitive placement near the ear enables the buzzer to work at a low sound volume. Additionally, a head set can replace this unit for future consideration. The fixture and location of the buzzer is shown in Figure 29.



**Figure 29: Buzzer Fixture for the Ultrasonic Vision System**

The mother board of the system which is based on the SK40C board is located at the top of the hat. The high empty capacity of the hat enables this placement. The advantage of this placement is that all the items of the system are within the hat. This extremely increases the easiness to wear, use and take off. Whenever the user does not require the system, it can be switched OFF using the switch which is attached to the mother board. Figure 30 shows the system internals.



**Figure 30: Ultrasonic Vision System (the Mother Board and Circuitry from Inside)**

## **CHAPTER 5**

### **CONCLUSION AND FUTURE DIRECTION**

In this chapter some conclusive statements on the progress the proposed project and the expected future work are discussed.

#### **5.1 Conclusion**

An ultrasonic vision system prototype is developed, tested and constructed which ensures ease of use. The prototype which comprises of ultrasonic sensors, a mother board and a buzzer is constructed on a hat-like gear to be worn at the head. The multi-level warning prototype provides a reliable and practical solution which aids visually impaired people to avoid obstacles while navigating. The implemented prototype works as required. Moreover, the system is made highly flexible to deal with up to eight sensors and different output devices for future exploitation and improvements.

#### **5.2 Future Directions**

Currently the coverage of the ultrasonic sensor is limited to a narrow range. In order to maximize the coverage so that many objects can be detected, the sensor needs to be located with an angle with respect to the vertical axis. The exact angle and the number of sensors needed are to be determined through experiments which are scheduled to finish in the near future. In addition to the three-leveled alarming buzzer, a headset or vibrators are considered potential alternatives to increase the level of comfort and convenience to the user.



## REFERENCES

- [1] World Health Organization, 'Prevention of Blindness and Visual Impairment', 21 June 2011, <http://www.who.int/blindness/en/>
- [2] A. Iqbal, U. Farooq, H. Mahmood, and M.U. Asad "A Low Cost Artificial Vision System for Visually Impaired People", Second International Conference on Computer and Electrical Engineering, 2009
- [3] Helal, A., Moore, S.E., Ramachandran, B., "Drishti: an integrated navigation system for visually impaired and disabled," Proceedings of Fifth International Symposium on Wearable Computers, pp.149-156, 2001
- [4] Oktem, R., Aydin, E., Cagiltay, N.E., "An indoor navigation aid designed for visually impaired people", 34th IEEE Annual Conference of Industrial Electronics, vol., no., pp.2982-2987, 10-13 Nov. 2008
- [5] Vision Systems, Wikipedia, ONLINE:  
"http://en.wikipedia.org/wiki/Vision\_system", retrieved on 29/11/2011
- [6] Alexander Hornberg; , "Handbook of Machine Learning", ISBN-13: 978-3-527-40584-8, WILEY-VCH Verlag GmbH & Co KGaA, Weinheim, 2006
- [7] Pissaloux, E.; , "A characterization of vision systems for blind people mobility," 2002 IEEE International Conference on Systems, Man and Cybernetics, vol.4, no., pp. 6 pp. vol.4, 6-9 Oct. 2002, doi: 10.1109/ICSMC.2002.1173278, URL: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1173278&isnumber=26298>
- [8] Yumin, Julius; Johnson, Louis; , "An implementation of an ultrasonic device for the visually impaired," Region 5 Conference, 2006 IEEE , vol., no., pp.287-290, 7-9 April 2006
- [9] Sehyung Park; Laehyun Kim; Sungdo Ha; Hyunchul Cho; Soo Yong Lee; , "An electronic aid for a visually impaired person using an ultrasonic sensor," Consumer Electronics, 2009. ICCE '09. Digest of Technical Papers International Conference on , vol., no., pp.1-2, 10-14 Jan. 2009
- [10] Do-Hoon Kim; Heung-Gyoon Ryu, "Obstacle recognition system using ultrasonic sensor and Duplex Radio-Frequency Camera for the visually impaired person," Advanced Communication Technology (ICACT), 2011 13th International Conference on , vol., no., pp.326-329, 13-16 Feb. 2011
- [11] Maxbotix Inc., "The MaxSonar-EZ1 High Performance Sonar Range Finder", 8757 East Chimney Spring Drive, Tucson AZ, 85747 USA, available: [www.maxbotix.com](http://www.maxbotix.com)

No	Phase	Weeks																											
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
1	Selection of Project Topic	■	■	■																									
2	Preliminary Research Work (Literature review, Related software, tools)	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
3	Submission of Extended Proposal Defense						●																						
4	Proposal Defense																												
5	Project work continues (Simulation, Equipment Purchase, Circuit interfacing, Circuit Testing, Programming)				■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
6	Submission of Interim Draft Report												●																
7	Submission of Interim Report												●																
8	Project Work Continues (Prototyping, Circuit interfacing and testing, programming)												■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
9	Submission of Progress Report																					●							
10	Project Work Continues (Finalizing Prototype, Result analysis)																					■	■	■	■	■	■	■	■
11	Pre-EDX																							●					
12	Submission of Draft Report																								●				
13	Submission of Dissertation (soft bound)																									●			
14	Submission of Technical Paper																										●		
15	Oral Presentation																											●	
16	Submission of Project Dissertation (Hard Bound)																											●	

● Reports Submissions

■ Process

## APPENDIX B

### PROGRAM LISTING FOR EXPERIMENT 4.1

```
include "main.h"

oid read_range    ();
oid range2cm     ();
oid show_status  ();
nt8 range;
nt8 range_cm;

define WARNING PIN_B7
define DANGER PIN_B6
define CLEAR_VIEW PIN_B2

oid main()

    setup_adc_ports(AN0_TO_AN5|VSS_VDD);
    setup_adc(ADC_CLOCK_DIV_2);
    setup_psp(PSP_DISABLED);
    setup_spi(SPI_SS_DISABLED);
    setup_spi2(SPI_SS_DISABLED);
    setup_wdt(WDT_OFF);
    setup_timer_0(RTCC_INTERNAL);
    setup_timer_1(T1_DISABLED);
    setup_timer_2(T2_DISABLED,0,1);
    setup_timer_3(T3_DISABLED|T3_DIV_BY_1);
    setup_comparator(NC_NC_NC_NC);
    setup_vref(FALSE);

    while (TRUE)
    {
        read_range();
        range2cm();
        show_status();
    }

oid read_range()

    set_adc_channel(0);
    delay_us(60);
    range=read_adc();

oid range2cm()

    range_cm=range*645/127;

oid show_status()

    if ( ( range_cm < 30 ) )
    {
        output_low (WARNING);
        output_high (DANGER);
        output_low (CLEAR_VIEW);
    }
    else
    if ( ( range_cm > 20 ) && ( range_cm < 80 ) )
    {
        output_high (WARNING);
        output_low (DANGER);
        output_low (CLEAR_VIEW);
    }
    else
    {
```

## APPENDIX C

### PROGRAM LISTING FOR EXPERIMENT 4.2

```
include "main.h"
include "LCD.h"

void read_range ();
void range2cm ();
void show_status ();
int8 range;
int8 range_cm;

#define WARNING_PIN_B7
#define DANGER_PIN_B6
#define CLEAR_VIEW_PIN_B2

void main()
{
    setup_adc_ports(AN0_TO_AN5|VSS_VDD);
    setup_adc(ADC_CLOCK_DIV_2);
    setup_psp(PSP_DISABLED);
    setup_spi(SPI_SS_DISABLED);
    setup_spi2(SPI_SS_DISABLED);
    setup_wdt(WDT_OFF);
    setup_timer_0(RTCC_INTERNAL);
    setup_timer_1(T1_DISABLED);
    setup_timer_2(T2_DISABLED,0,1);
    setup_timer_3(T3_DISABLED|T3_DIV_BY_1);
    setup_comparator(NC_NC_NC_NC);
    setup_vref(FALSE);
    set_tris_b(0x00);

    while (TRUE)
    {
        read_range();
        range2cm();
        char message1[16]="range range_cm", message2[16];
        lcd_display_str(0,message1 );
        sprintf(message2, "%03u %03u",range,range_cm);
        lcd_display_str(1,message2 );
        show_status();
    }
}

void read_range()
{
    set_adc_channel(0);
    delay_us(60);
    range=read_adc();
}

void range2cm()
{
    range_cm=range*645/127;
}

void show_status()
{
    if ( ( range_cm < 30 ) )
    {
        output_low (WARNING);
        output_high (DANGER);
        output_low (CLEAR_VIEW);
    }
    else
    if ( ( range_cm > 20 ) && ( range_cm < 80 ) )
    {
        output_high (WARNING);
        output_low (DANGER);
    }
}
```

## APPENDIX D

### PROGRAM LISTING FOR EXPERIMENT 4.3

```

include "main.h"

oid read_range ();
oid range2cm ();
oid show_status ();
unsigned char range;
int8 range_cm;

define WARNING PIN_B7
define DANGER PIN_B6
define CLEAR_VIEW PIN_B3

oid main()

    setup_adc_ports(AN0|VSS_VDD);
    setup_adc(ADC_CLOCK_INTERNAL);
    setup_psp(PSP_DISABLED);
    setup_spi(SPI_SS_DISABLED);
    setup_wdt(WDT_OFF);
    setup_timer_0(RTCC_INTERNAL);
    setup_timer_1(T1_DISABLED);
    setup_timer_2(T2_DISABLED,0,1);
    setup_timer_3(T3_DISABLED|T3_DIV_BY_1);
    setup_comparator(NC_NC_NC_NC);
    setup_vref(FALSE);
    set_tris_b(0x00);
/Setup_Oscillator parameter not selected from Intr Oscillator Config tab

    while (TRUE)
    {
        read_range();
/        range2cm();
        show_status();
    }

oid read_range()

    set_adc_channel(0);
    delay_us(60);
    range=read_adc();

/void range2cm()
/{}
/    range_cm=range*6/127;
/{}

oid show_status()

    //if ( ( range_cm < 1 ) )
    if ( ( range >= 0 ) && ( range < 10 ) )
    {
        output_low (WARNING);
        output_high (DANGER);
        output_low (CLEAR_VIEW);
    }
    else
    //if ( ( range_cm > 1 ) && ( range_cm < 2 ) )
    if ( ( range >= 15 ) && ( range < 40 ) )
    {
        output_high (WARNING);
        output_low (DANGER);
        output_low (CLEAR_VIEW);
    }

```

**APPENDIX E**  
**PROGRAM LISTING FOR EXPERIMENT 4.4**  
**FINAL C CODE (MAIN.C)**

```
include "main.h"

define DANGER PIN_B6 //DANGER LED is connected to PIN B6
define WARNING PIN_B7 //WARNING LED is connected to PIN B7
define CLEAR_VIEW PIN_B3 //CLEAR_VIEW LED is connected to PIN B3
define BUZZER PIN_B2 //BUZZER is connected to PIN B2

oid PIC_init ();
oid read_range ();
oid range2cm ();
oid show_status ();
nsigned char range;
at16 range_cm;
at16 time_elapsed=0;

int_TIMER2
oid TIMER2_isr(void) //10 ms overflow

f(++time_elapsed==2000) //20 sec

    disable_interrupts(INT_TIMER2);
    output_high (BUZZER);
    delay_ms(20);
    output_low (BUZZER);
    delay_ms(20);
    output_high (BUZZER);
    delay_ms(20);
    output_low (BUZZER);
    delay_ms(20);
    time_elapsed=0;
    enable_interrupts(INT_TIMER2);

lear_interrupt(INT_TIMER2);

oid main()

    PIC_init();
    while (TRUE)
    {
        read_range();
        show_status();
    }

/ device_init() function used to initialize the PIC
oid PIC_init()

    setup_adc_ports(AN0|VSS_VDD); //use A0 as analog input, input range: VSS-VDD
    setup_adc(ADC_CLOCK_INTERNAL); //use internal clock for conversion
    setup_psp(PSP_DISABLED); //Disable PSP
    setup_spi(SPI_SS_DISABLED); //Disable SPI
    setup_wdt(WDT_OFF); //Disable watchdog timer
    setup_timer_0(RTCC_INTERNAL);
    setup_timer_1(T1_DISABLED); //Disable timer0, timer1
    setup_timer_2(T2_DIV_BY_16,195,16); //set timer2 with 10 ms overflow
    setup_comparator(NC_NC_NC_NC); //Diabile comparators
    setup_vref(FALSE);
```

```

enable_interrupts(INT_TIMER2); //Set timer2 to flag interrupt when overflows
enable_interrupts(GLOBAL); //Enable interrupts
set_tris_b(0x00); //setting port B as output (for buzzer and LEDs)

/ read_range() function used to read the range of object
/via the ADC at channel 0 (RA0) to get the distance measurement
void read_range()

    set_adc_channel(0);
    delay_us(60);
    range=read_adc();

/range2cm converts the reading to cm by considering that
/127 reading corresponds to 645cm according to the manual
void range2cm()

/   range_cm=range*645/127;

void show_status()

    //if ( ( range_cm < 50 ) ) //can be used alternatively
    if ( ( range >= 0 ) && ( range < 10 ) )
        //LED=DANGER (RED), Buzzer always ON
        {
            output_low (WARNING);
            output_high (DANGER);
            output_low (CLEAR_VIEW);
            output_high (BUZZER);
        }
    else
        //if ( ( range_cm > 50 ) && ( range_cm < 150 ) )
        if ( ( range >= 10 ) && ( range < 29 ) )
            //LED=WARNING (ORANGE), Buzzer ON 100ms/OFF 100ms
            {
                output_high (WARNING);
                output_low (DANGER);
                output_low (CLEAR_VIEW);
                output_low (BUZZER);
                delay_ms(100);
                output_high (BUZZER);
                delay_ms(100);
            }
        else
            //if ( ( range_cm > 2 ) )
            if ( ( range >= 29 ) )
                //LED=CLEAR VIEW (GREEN), Buzzer ON 100ms/OFF 100ms
                {
                    output_low (WARNING);
                    output_low (DANGER);
                    output_high (CLEAR_VIEW);
                    output_low (BUZZER);
                }

```

## APPENDIX F

### PROGRAM LISTING FOR EXPERIMENT 4.4

#### FINAL HEADER CODE (MAIN.H)

```

include <18F4550.h>
device adc=8

FUSES NOWDT                //No Watch Dog Timer
FUSES WDT128               //Watch Dog Timer uses 1:128 Postscale
FUSES HS                   //High speed Osc (> 4mhz for PCM/PCH) (>10mhz for PCD)
FUSES NOPROTECT           //Code not protected from reading
FUSES NOBROWNOUT          //No brownout reset
FUSES BORV20               //Brownout reset at 2.0V
FUSES NOPUT                //No Power Up Timer
FUSES NOCPD                //No EE protection
FUSES NOSTVREN             //Stack full/underflow will not cause reset
FUSES NODEBUG              //No Debug mode for ICD
FUSES NOLVP                //No low voltage prgming, B3(PIC16) or B5(PIC18) used for I/O
FUSES NOWRT                //Program memory not write protected
FUSES NOWRTD               //Data EEPROM not write protected
FUSES NOIESO               //Internal External Switch Over mode disabled
FUSES NOFCMEN              //Fail-safe clock monitor disabled
FUSES NOPBADEN             //PORTB pins are configured as digital I/O on RESET
FUSES NOWRTC               //configuration not registers write protected
FUSES NOWRTB               //Boot block not write protected
FUSES NOEBTR               //Memory not protected from table reads
FUSES NOEBTRB              //Boot block not protected from table reads
FUSES NOCPB                //No Boot Block code protection
FUSES MCLR                 //Master Clear pin enabled
FUSES NOLEPT1OSC           //Timer1 configured for higher power operation
FUSES NOXINST              //Extended set extension and Indexed Addressing mode disabled
Legacy mode)
FUSES PLL1                 //Divide By 1
FUSES CPUDIV1              //System Clock by 1
FUSES NOUSBDIV             //USB clock source comes from primary oscillator
FUSES NOVREGEN             //USB voltage regulator disabled
FUSES NOICPRT              //ICPRT disabled

use delay(clock=20000000)

```



# Ultrasonic Vision System

KARSOU AHMED OBAID<sup>1</sup>, Mohd Zuki Bin Yusoff<sup>2</sup>

<sup>1</sup>Student Support Services

Universiti Teknologi PETRONAS, Bandar Seri Iskandar, 31750 Tronoh, Perak, Malaysia

Email: karsou\_ao@hotmail.com

<sup>2</sup>Electrical and Electronics Engineering Programme

Universiti Teknologi PETRONAS, Bandar Seri Iskandar, 31750 Tronoh, Perak, Malaysia

Tel: +60-5-3687807, Email: mzuki\_yusoff@petronas.com.my

ct

impaired people face major problems in activities require visual awareness such as: navigation, sport, etc. This project attempts to contribute to the impaired community as well as to academia by a low cost, easy to construct navigation tool which aids visually disabled to avoid objects while navigation. The proposed method utilizes EZ1 ultrasonic modules, PIC16F4550 microcontroller, and 6 Volts buzzer like alarm. For convenience and comfort, the system is wholly implemented on an ice cap. A coherent development methodology was adopted to implement the proposed system by the use of potentiometers and an LCD during the initial and mid phases of the project to emulate the hardware modules and visually debug the intermediate stage results. The practicality and uniqueness of the system is demonstrated on an ice cap with the ease to wear feature as a significant of our contribution. Remarkably, the system was tested in different scenarios and was able to detect objects up to 6.45 meters and send hearable distinct alarm at 2 meters and 0.5 meter far. By the use of the proposed prototype, visually impaired users can independently navigate while avoiding objects with two tone alarm indication. In addition, the prototype is designed to be flexible in such way that it can be easily modified for future extension and improvements for both hardware and software by utilizing the abundant resources of PIC16F4550 microcontroller and CCS C compiler.

rds

System, Microcontroller, Ultrasonic sensor, Hat.

ction

Sight or vision impairment is a prominent problem affecting millions of people all over the world according to the World Health Organization (WHO) [1]. Visually impaired people face major problems in activities which require visual awareness such as: navigation, reading, sport, etc. In the past, inventors, entrepreneurs, and researchers have

worked hard throughout the last century to present solutions which could allow visually impaired people to do many of their activities independently.

This project attempts to contribute to the visually impaired community by developing low cost, easy to construct navigation tool which aids visually disabled to avoid objects while navigating.

Visually disabled people need some external support to tell them about surrounding objects when navigating. Prior work as in [2]-[4] presents different techniques and solutions which allow visually impaired people to navigate independently. The methods used employ GPS, wireless networks, camera, and ultrasonic sensors. However, most of them are not easy to afford as in [3] or limited battery life as in [3] or require additional hardware as in [2] or limited coverage as in [4]. So, the aim is to provide reliable yet affordable tool which enables visually impaired people to avoid obstacles while navigating.

Therefore the aim of this project is to construct an ultrasonic based vision system which provides the following features: allowing visually disabled people to avoid fixed objects (e.g. wall), multi-level alarming and comfort of use.

## Background

The term "vision system" in most scientific sources likely refers to two different meanings: a biological one in which the term means: "the neurobiological circuitry that enables living beings to see" [5] or an artificial one in which the term means: "computer based systems where software performs tasks similar to "seeing", usually aimed at industrial quality assurance, part selection, defect detection etc" [6]. In the context of this work, vision systems are those sensing systems which help visually impaired people to gain some sensory information regarding the surrounding objects.

According to the defined scope, technological aids which help in object avoidance, navigation, reading textual data and detecting objects all fall under the vision systems class [7].

Vision systems generally use sensors, processing units and sensitization units to detect obstacles and convey to the user the information about it to enable him/her to take a proper action. According to the requirement of the system and the

it does, different sensing, processing and warning units are used such as: ultrasonic sensors, cameras, for sensing; microcontroller, microprocessor for buzzer, speaker, vibrator to alarm and warn the the following section, some of the relevant vision are considered for discussion and analysis.

Over the last two decades, several attempts to develop aiding systems have been made. A. Helal et al. in [3] have developed a wireless based navigation system. Their method utilizes several wearable computer technologies, wireless Geographic Information System (GIS), and Global Positioning System (GPS). The proposed system provides a highly sophisticated interface and the luxurious amount of data which enables turn by turn navigation. However, the proposed system is costly as it is more expensive for navigation purposes rather than simply avoiding obstacles. Aside from the complexity and the navigation provided by this system, it is bulky for simple navigation. Moreover, the limited coverage of GPS in buildings makes it a non viable option for indoor navigation.

Umin et al. in [8] discussed the implementation of a vision device in a vision system for visually disabled. The proposed implementation in [8] utilizes the ultrasonic sensor to measure the distance of surrounding objects and warn the user via a buzzer when approaching an object. The device consists of a receiver and transmitter to detect the distance of near objects. It also utilizes two microcontrollers to manage the manipulation of signals. The implementation in [8] provides simple hardware, but it takes into account only one detection direction (front side); if an object approaches from sides, the device will not detect it. Additionally, this system does not provide the user the range which the detected object resides. This is important for convenience and comfort of the user. This is because, for the user to take the appropriate decision and act accordingly, the user requires knowing, by some mean, how far the object is. Having this knowledge helps the user to determine the existence of the object which might be a human. However, if the distance is far (i.e., 3 meters) the user has to be alerted, but the information at that range is of little benefit. However, as the range of the object decreases (e.g., 50 cm), the alarm volume should go louder.

Umin et al. in [4] proposed an indoor navigation system based on an RFID concept. This method provides a robust system which enables smooth navigation with the help of a wireless network, an RFID device, a compass, and a PC. Due to its robustness, the extra hardware limits its use.

Researchers in [2], [9], [10], proposed different methods using ultrasonic devices. Their methods provide reliable navigation as they only take into account the detection at a specific direction. In addition, the proposed systems do not inform the user how far the object is. This feature is added to give the user more flexibility and awareness by informing him and how far they are. This could enable

the user to have more decisions and actions to take according to the location of the object.

## Methodology

Throughout the development stage of the prototype of the project, several tools are potentially considered:

### 1. Ultrasonic module: Max Sonar – EZ1 ultrasonic range finder

An ultrasonic module which consists of a transmitter, a receiver and a calculation chip is to be used. The Ultrasonic module should provide range reading for the distance of the nearest object at its transmission direction. According to the electrical specification and project requirement of the proposed system, the MaxSonar-EZ1 [11] ultrasonic range finder is considered in this project.

The specification of the MaxSonar-EZ1 is listed in

Table 1. The voltage and current requirements of this module are compatible with embedded applications.

Additionally, the measurement frequency provided by the module is around 20 Hz which enables the availability of a new reading for the distance of every 5 hundredth of a second. Assuming that the critical distance at which the system should sign an alarm is 1 meter, and assuming the time to react and time to stop or change direction take zero second, we can imply that the maximum speed which this module could support is around 20 m/s.

**Table 1: LV- MaxSonar-EZ1 Ultrasonic Range Finder Specification**

Item	Value
Ultrasonic frequency	42 KHz
Supply Voltage	2.5- 5.5 V
Supply current	2 mA
Reading Rate	20 Hz (0.05 second)
Output Signals	RS232, PWM, Analog
Measurement Range	0 - 6.45 m
Analog Output	10mV/inch, pins: V <sub>cc</sub> , Gnd, BW

### 2. PIC18F4550 microcontroller

A mid range microcontroller from Microchip is to be used. This selection enables the developers to deal with the prototype with more flexibility and efficiency.

The PIC18F4550 is among the most commonly used Microchip microcontrollers because of its USB communication support capabilities. The PIC18F4550 is a 40 pin high performance microcontroller which is equipped with several built-in peripherals. Even though the proposed

does not explicitly require the USB support, the is made to enable future development and / of functionality expansion.

### CCS C compiler

to program the microcontroller, a compiler is to be In this project, the PIC C compiler from CCS is to

C compiler is easy to use, and almost immediate to d due to the project wizard feature and the different odes which it offers. Among the features of CCS C : Automatic fuses configuration, Extensive built-in providing direct access to PIC hardware, Extensive ode driver library, Arithmetic library, Integrated ent environment

### SK40C PIC development kit and UIC00B programmer

up the development phase, a startup kit is used as orm of the microcontroller circuit. The use of this ides easier and more robust circuit to be built. The vides several functionalities and circuitry support. is a robust development platform which offers: egulation circuitry (9 V input voltage to 5 V), Reset SB port, Connector to programmer, 2 push buttons EDs connected to Port B and 20 MHz crystal

of the SK40C board are labeled clearly. This assists ng up the prototyping and reduces the errors in g to external circuitries.

to transfer the C codes to the program memory of icrocontroller, USB ICSB programmer (UIC00B) is id. This programmer offers a cheap programming nd is compatible with the SK40C startup kit. The aing software (PIC kit 2) takes the hex file which is by CCS compiler and loads it to the troller memory via UIC00B programmer.

### hm and code

ment the various functions of the vision system, the quired to continuously perform the following:

Read the sensor value  
 Convert sensor readings to known units  
 Show the status of the detected objects on the LEDs as follows:

1. DANGER: by activating a RED LED at PIN B7 and constantly activating a buzzer.
2. WARNING: by activating an ORANGE LED at PIN B6 and alternately activating a buzzer.
3. CLEAR VIEW: by activating a GREEN LED at PIN B2

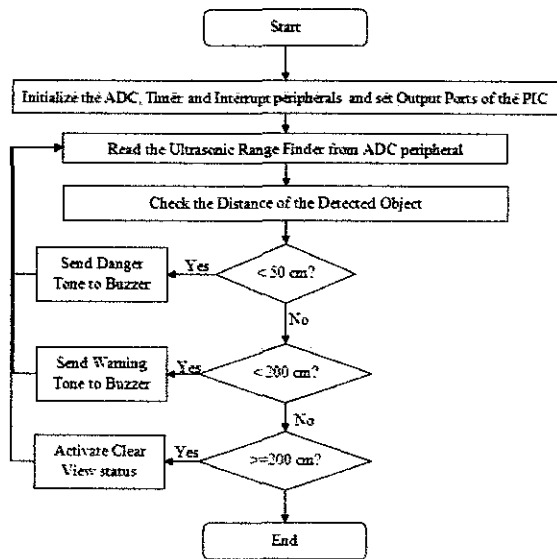


Figure 1: Ultrasonic Vision System Algorithm

The proposed algorithm and procedural steps are shown in the flow chart in Figure 1. The flow chart shows the necessary steps required to realize the system. The PIC has several peripherals which can work in different ways according to how they are configured. Therefore, to ensure the system will utilize the correct peripherals and at the right settings, the initialization step has to be performed at the early start of the system.

After the proper settings have been configured the system goes to read the ultrasonic module which is interfaced to the microcontroller via its analogue input pins. However, the output of the ADC peripheral is always a digital value in the range of 0 to 255, if configured to work in 8 bit mode, corresponding to range of VSS (0 V) to VDD (5 V). Without referring to how the ultrasonic module work as in the manual, neither the numerical value nor the voltage tell a distance, therefore, the obtained values are adjusted to be used in the other subsequent operations.

The system then checks the distance of the detected object and activates special visual and audible outputs accordingly. The detection range of the sensor is 645 cm. This detection range is arbitrary divided into three zones or sub ranges. The first zone is the critical one to the user, whereby an object is detected at range less than 50 cm. This range is referred as in this work as “dangerous” zone. Considering a navigating scenario, in this zone the user is expected to take an immediate action to prevent collisions. The second zone is the “warning” zone. This zone is defined between 50 cm and 200 cm of distance between the user and the detected object. In this zone, the system sends different audible output (for the user) and visual output (for diagnosis purposes) to that sent in the “dangerous” zone. The third zone is “clear view” zone. The detection distance of this zone is for greater than 200 cm to the maximum detection range (i.e. 645 cm). In this

system activates visual output (i.e. LEDs), for purposes, and does not activate any audible output. The system is set to continuously work in a closed loop of the distance of the detected object if any and set the and activating the outputs accordingly. This as monitoring and alerting enables object avoidance safe navigating.

ally, since the system is proposed for the use of impaired users, the system is provided by special which is an audible system monitor. The aim of such s to acknowledge to the user that the system is and running. This is because; when the system relies ery, it can run out and stop the functionality of the without the user being informed or alerted. The monitoring feature periodically sends a special one to tell the user that the system is running. The implementation details are discussed in later parts ction.

```
#include "main.h"

void PIC_init    ();
void read_range ();
void range2cm   ();
void show_status ();
unsigned char range;
int16 range_cm;

#define DANGER_PIN_B6
#define WARNING_PIN_B7
#define CLEAR_VIEW_PIN_B3
#define BUZZER_PIN_B2

void main()
{
    PIC_init();
    while (TRUE)
    {
        read_range();
        range2cm();
        show_status();
    }
}
```

**Figure 2: System Program Highlighting the Induced Files, Used Functions, Compiler Directives and the main() Function**

the procedures as shown in Figure 1 and explained y, a modular code is developed as highlighted in This figure shows the core functions and compiler considered for the implementation.

configuration words are the most influential settings the PIC would work at the system level. The tion words primarily control some switches inside ocontroller according to the oscillator, supply special peripherals, programming and debugging ents. Different compilers have different ways of ie configuration words. In CCS C compiler the tion words are mainly set by #fuses and #device . The generation of the code can automatically be exploiting the project wizard facility. This header ally sets the microcontroller to use 8 bit ADC n result, 20 MHz oscillator frequency and the

master clear pin while disabling the other features. By these lines, the configuration words are set and the PIC is ready to use the peripherals and execute commands and instructions.

Initializing the PIC to work as required is as important as the core program. This is because; the PIC has wide range of peripherals multiplexing sharing the same pins. This requires careful programming and therefore highlights the significance of this work. The initialization code is listed in The initialization code essentially enables and disables peripherals according to the requirement. Since the proposed system utilizes the ADC for reading the sensor, timer 2 to generate the necessary timing, port B to communicate with buzzer and LEDs. These peripherals are consequently enabled. The other unused peripherals (i.e. watchdog timer, PSI, PSP, comparators, etc.) are disabled.

To read the range of the detected object, read\_range() function is used. This function reads the distance of the detected object by reading from the ADC module. Since the sensor is attached to PIN A0, set\_adc\_channel(0) is used. The read\_adc() function is called to read the conversion results from ADC peripheral. A delay of 60 us is required for the conversion to settle.

The status of the reading is implemented using the show\_status() function. The function is implemented using if else statements and the three levels of output are activated using output\_low() and output\_high() functions.

The target user of the vision system is visually impaired people. When the system is out of battery, the user will be at the risk. Even though, the system is provided by visual indicator (LED) to show the status of the power supply (battery), the target user can not see any visual indication. Therefore, the monitoring system is proposed in the prototype. The idea is to periodically play a special tone which its purpose is to tell the user that the system is healthy. To implement such concept without affecting the way the system work, the timer peripheral is utilized. In the PIC microcontroller, there're three commonly used timers, timer 0, 1 and 2. Timer 0 is normally used as real time clock, timer 1 is sometimes used as an internal clock and low power counter for the PIC. Therefore, timer 2 is chosen to implement such feature.

With the help of the project wizard utility, timer 2 is empirically set such that it set an interrupt flag every 10 ms at 20 MHz oscillator frequency. Once the interrupt occurs, the program counter inside the PIC goes directly to timer 2 interrupt service routine, where the system monitor code is located. Once the timer overflows the mentioned function is executed. As explained, the interrupt service routine mainly increments a counter variable called time\_elapsed and check if its value is equal to 2000. When time elapsed is 2000 which implies that 20 second has been elapsed (i.e. 2000 x

The PIC activates the buzzer with a double switches to indicate special alert that the system is running.



**Figure 3: Ultrasonic Vision System Prototype**

The prototype for the system is constructed by following the methodology explained previously. To ensure the comfort of the user, a hat shaped wearable cap is considered for implementation as shown in Figure 3, Figure 4, Figure 5, and Figure 6. The material of the hat is chosen such that it can accommodate the hardware of the system without the burden of necessarily carrying extra accessories. The items used for the implementation were: ultrasonic range finder, buzzer, and mother board. To aid the portability of the system, a battery module can be used and located beside the mother board. A top view for the prototype is described in Figure 3. The location of the sensor is chosen to face the objects at the front as well as to enable large angle for object detection. Figure 4 shows the location of the sensor. The system is highly extendable to 8 range finders on Ports A and E without excessive



**Figure 4: Ultrasonic Vision System (Range Finder Unit)**

The buzzer is located at the left side of the prototype. This intuitive placement near the ear enables the buzzer to work at a low sound volume. Additionally, a head set can replace this unit for future consideration. The fixture and location of the buzzer is shown in Figure 5.



**Figure 5: Buzzer Fixture for the Ultrasonic Vision System**

The mother board of the system which is based on the SK40C board is located at the top of the hat. The high empty capacity of the hat enables this placement. The advantage of this placement is that all the items of the system are within the hat. This extremely increases the easiness to wear, use and take off. Whenever the user does not require the system, it can be switched OFF using the switch which is attached to the mother board. Figure 6 shows the system internals.



**Figure 6: Ultrasonic Vision System (the Mother Board and Circuitry from Inside)**

## LUSION

onic vision system prototype is developed, tested  
ructed which ensures ease of use. The prototype  
mpriprises of ultrasonic sensors, a mother board and a  
constructed on a hat-like gear to be worn at the  
e multi-level warning prototype provides a reliable  
ical solution which aids visually impaired people to  
stacles while navigating. The implemented  
: works as required. Moreover, the system is made  
xible to deal with up to eight sensors and different  
vices for future exploitation and improvements.

## MMENDATIONS

the coverage of the ultrasonic sensor is limited to a  
nge. In order to maximize the coverage so that  
ects can be detected, the sensor needs to be located  
ngle with respect to the vertical axis. The exact  
l the number of sensors needed are to be determined  
xperiments which are scheduled to finish in the  
e. In addition to the three-leveled alarming buzzer,  
or vibrators are considered potential alternatives to  
he level of comfort and convenience to the user.

## ENCES

l Health Organization, "Prevention of Blindness and  
Impairment", 21 June 2011,  
[w.who.int/blindness/en/](http://www.who.int/blindness/en/)  
bal, U. Farooq, H. Mahmood, and M.U. Asad "A  
t Artificial Vision System for Visually Impaired  
Second International Conference on Computer and  
Engineering, 2009  
, A., Moore, S.E., Ramachandran, B., "Drishti: an  
l navigation system for visually impaired and  
' Proceedings of Fifth International Symposium on  
: Computers, pp.149-156, 2001  
m, R., Aydin, E., Cagiltay, N.E., "An indoor  
n aid designed for visually impaired people", 34th  
nual Conference of Industrial Electronics, vol., no.,  
2987, 10-13 Nov. 2008  
/ision Systems, Wikipedia, ONLINE:  
.wikipedia.org/wiki/Vision\_system", retrieved on  
ll  
xander Hornberg; , "Handbook of Machine  
' , ISBN-13: 978-3-527-40584-8, WILEY-VCH  
mbH & Co KGaA, Weinheim, 2006  
oux, E.; , "A characterization of vision systems for  
ple mobility," 2002 IEEE International Conference  
ns, Man and Cybernetics, vol.4, no., pp. 6 pp. vol.4,  
2002, doi: 10.1109/ICSMC.2002.1173278, URL:

<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1173278&isnumber=26298>

[8] Yumin, Julius; Johnson, Louis; , "An implementation of an ultrasonic device for the visually impaired," Region 5 Conference, 2006 IEEE , vol., no., pp.287-290, 7-9 April 2006

[9] Sehyung Park; Laehyun Kim; Sungdo Ha; Hyunchul Cho; Soo Yong Lee; , "An electronic aid for a visually impaired person using an ultrasonic sensor," Consumer Electronics, 2009. ICCE '09. Digest of Technical Papers International Conference on , vol., no., pp.1-2, 10-14 Jan. 2009

[10] Do-Hoon Kim; Heung-Gyoon Ryu, "Obstacle recognition system using ultrasonic sensor and Duplex Radio-Frequency Camera for the visually impaired person," Advanced Communication Technology (ICACT), 2011 13th International Conference on , vol., no., pp.326-329, 13-16 Feb. 2011

[11] Maxbotix Inc., "The MaxSonar-EZ1 High Performance Sonar Range Finder", 8757 East Chimney Spring Drive, Tucson AZ, 85747 USA, available: [www.maxbotix.com](http://www.maxbotix.com)