School of Engineering Science • Burnaby, BC • V5A 1S6
• sonitus-340@sfu.ca

October 20, 2003

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

Re: ENSC 340 Design Specifications for a 3D Ultrasonic Motion Tracking System.

Dear Dr. Rawicz,

Attached is Sonitus Solutions' *Design Specifications for a 3D Ultrasonic Motion Tracking System*. The purpose of this system is to track the movements of a user and feed the data into any IBM-compatible computer in real time. The technology is based on ultrasonic transmitters and receivers that promise a low cost alternative to existing designs.

The design specifications attached describes the tracking system in detail. All aspects of hardware, firmware, and software design will be fully documented and discussed. We will also be providing a test plan to which we will adhere to in order to ensure that we meet our specifications..

Sonitus is composed of six highly motivated and resourceful individuals: Alan Chuang, Kenneth Fong, Henry Lin, Warren Lee, Edward Loo and Richard Sheng. If you have any questions or concerns about our proposal, please feel free to contact me by phone at (604) 771-9719 or by e-mail at sonitus-340@sfu.ca.

Sincerely,

Richard Sheng
President
Sonitus Solutions

Enclosure: Design Specifications of the 3D Movement Tracking System

Design Specifications for an 3D Ultrasonic Motion Tracking System

Project Team:	Alan Chuang Kenneth Fong Warren Lee Henry Lin Edward Loo Richard Sheng
Submitted To:	Andrew Rawicz Steve Whitmore
Contact Person:	Alan Chuang **sonitus-340@sfu.ca**

Issued Date: October 20th,2003

Executive Summary

In today's changing world, technology is advancing at a rapid pace. As users demand more and more advanced programs, application software is moving into 3D space. At Sonitus Solutions, we see that the market is ripe for a 3D input system. Thus, we have decided to create the Sonitrac. Sonitrac is a system which will use ultrasonic transmitter and receivers in order to track the location of various modules placed on a user's body. Software will then interpret that information for its own use.

In stage one of research and design, Sonitus Solutions plans to create a prototype of the Sonitrac. The prototype will be able to:

- Transmit and receive ultrasonic signal
- Send data into the PC through the RS-232 serial port
- Correctly triangulate the position of each transmitter
- Display the positions of the transmitters in a graphical manner in real-time
- Save those positions into a text file

In stage two, Sonitus Solution will be developing a production model of the Sonitrac, which will implement all of the functions we plan to have. Some of these functions include:

- The ability to pause the system using a button on the mobile unit.
- 2D positional graphs on selected channels, in which velocity and acceleration graphs can be derived and displayed.
- Display an orthogonal representation of the user, which will further develop into a 3D representation using Direct3D
- Load and replay a saved session
- Seamless switching between battery power and AC power in the base station
- Recharging mobile unit by attaching it to base station

Stage one of design is to be completed by December, 2003. Stage two of design will be contingent on the result of the prototype, which would give a clear indication of the feasibility and effectiveness of Sonitrac.

Table of Contents

List of Tables

List of Figures

1. INTRODUCTION

This document is the functional specification of Sonitus Solution's Sonitrac. Sonitrac is a 3D tracking system that will use ultrasonic technology in order to determine the location of numerous modules. With the modules placed on the body of the user, Sonitrac will be able to determine the posture and movements of the user in real-time. Software will then be able to use this data for various purposes such as displaying them on screen as a model of a person. This is an example of one of the numerous applications possible with this system, and will be the objective of stage two of research and design.

1.1. Scope

The main purpose of this document is to outline the design goals of this project. These targets include user interface descriptions, as well as hardware and software requirements. A test plan is also outlined to ensure that the final product meets all the specified requirements.

1.2. Glossary and Abbreviations

GUI	Graphical User Interface
Kb	Kilobyte
R&D	Research and Design
RAM	Receiver Array Matrix
STP	standard temperature and pressure
P3DMTS	precision 3d motion tracking system
RF	radio-frequency
LED	Light Emitting Diode
User Space	The area in which the user is expected to be in when using the system
Channel	An active transmitter whose coordinates are to be determined
Failure	The system ceases to detect an object's position, and resume without human intervention.
Pointer	A variable which holds the address(location) of another variable
Receiver	A device that reads in a signal
Transmitter	A device that sends out a signal
Gaussian Elimination	A procedure that can solve a system of linear equation.

2. SYSTEM OVERVIEW

Sonitus Solutions has chosen to use the acoustic technology for Sonitrac, which consists a base station, a mobile unit, a receiver array matrix, and a PC software user interface. The receiver array matrix will be composed of receivers attached on a rigid frame that defines the allowable user movement space. Receivers will detect ultrasonic waves emitted from the transmitters of the mobile unit attached at various locations on the user. The base station will then provide the elapsed time of the detected sound waves to the PC software, which will pin-point the exact coordinates of each transmitter channel. The software user interface will then display correspondent orthogonal 2D images of the user body.

Sonitrac will be a superior product because it allows flexibility of space, provides cost efficiency and accuracy, and is lightweight and easy to set-up. Users only have to connect the base station to their computer through the serial port and run the software. After setting up the 4 receiver poles around them, the user can attach the mobile unit and the sensor bands on to their body. Once the user is ready, they can control the program via the mobile unit. Please see the figure below for a visualization of the product.

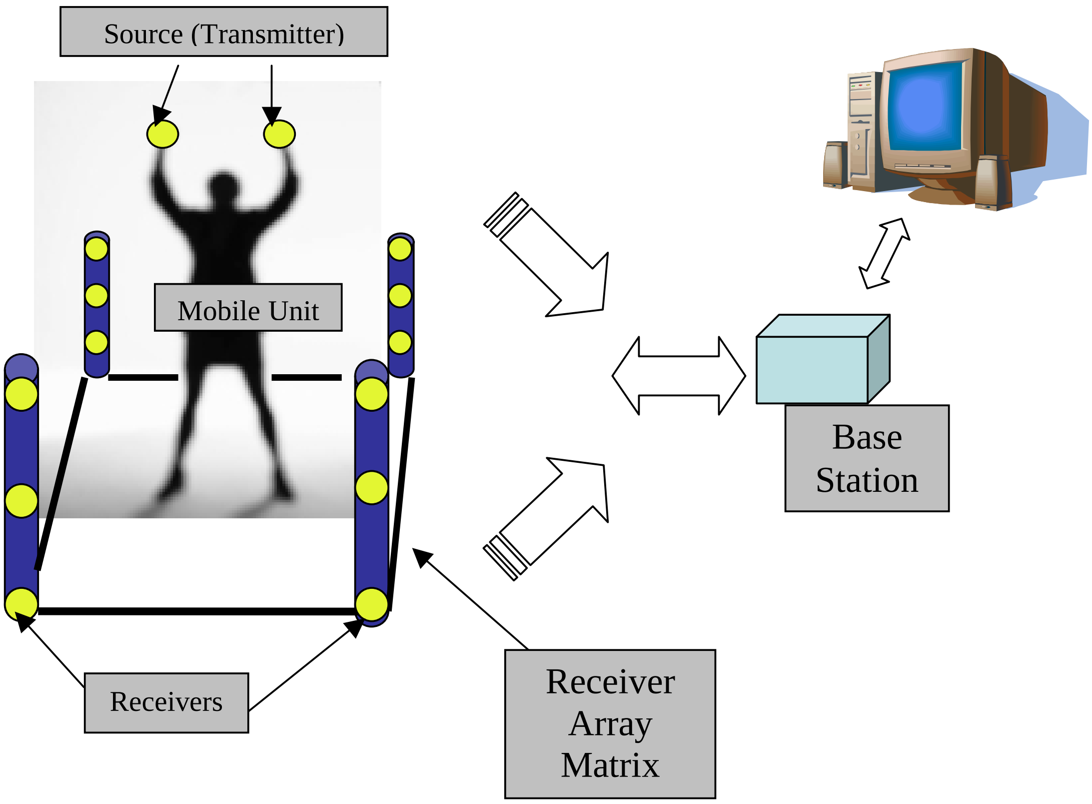


Figure 1: System Overview

3. SYSTEM HARDWARE

3.1. Component Description

The main features of the key components selected for implementing Sonitrac are described in this section, as well as the reasons behind each choice. The components chosen are based on their functionality, efficiency, physical size, costs and accessibility. However, since Sonitrac is a prove-of concept model, and also due to the limited time and accessibility, the components selected for our design may not be the most ideal or could be over-qualified for our use.

3.1.1. Microcontroller – PIC18F242

The controller has the following main features:

- Maximum 40MHz system clock
- Timer0 module: 8-bit/16-bit timer/counter with 8-bit programmable prescaler
- Timer1 module: 16-bit timer/counter
- Timer2 module: 8-bit timer/counter with 8-bit period register
- Timer3 module: 16-bit timer/counter
- Two Capture/Compare/PWM (CCP) modules. CCP pins that can be configured as:
  - Capture input: capture is 16-bit, max. resolution 6.25 ns
  - Compare is 16-bit, max. resolution 100
- Universal Asynchronous Serial Port (MSSP) module with fast sampling rate
- Low power, high speed EPROM technology
- Wide operating voltage range (2.5V to 5.5V)
- Industrial and Extended temperature ranges
- Low power consumption

The fast system clock is essential to achieve high resolution in triangulation. The external interrupts and A/D conversion capability are also necessary for the basic operations for our system. Furthermore, the built-in Universal Asynchronous Serial Port feature simplifies the need for implementing our own communication protocol, and the wide operating voltage allows for more flexibility to accommodate other peripheral hardware. For convenience and accessibility reasons, the two microcontrollers implemented on Base Station and Mobile Unit will be the same. The figure below is the pin diagram of the microcontroller.

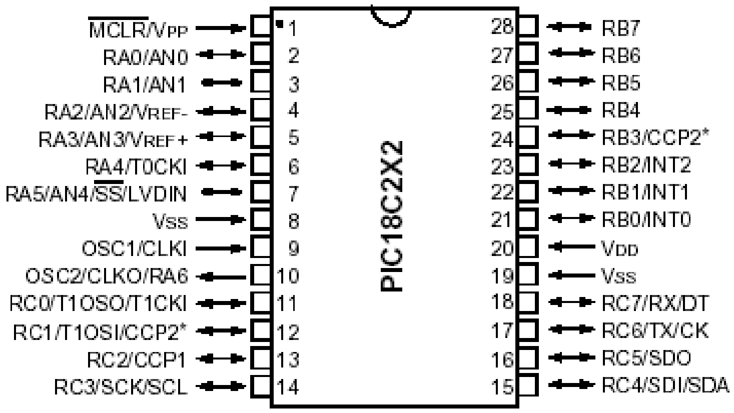


Figure 2: Pin diagram of PIC 18F242

3.1.2. Ultrasonic Transducers – Panasonic Ultrasonic Ceramic Sensors (Type U/H)

The main reasons for choosing this type of transducers are their easy accessibility, small sizes, and relatively low costs. These transducers are chosen solely for demonstration purposes, and will not be the ultimate choice if Sonitrac is to be produced for public use. Figure 3 illustrates the sensor sensitivity over frequency, and figure 4 is the physical dimension of the sensors.

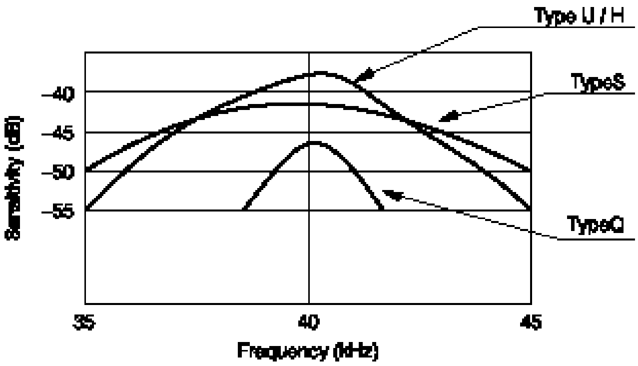


Figure 3: Frequency Characteristics (Sensitivity) of the ultrasonic transducers

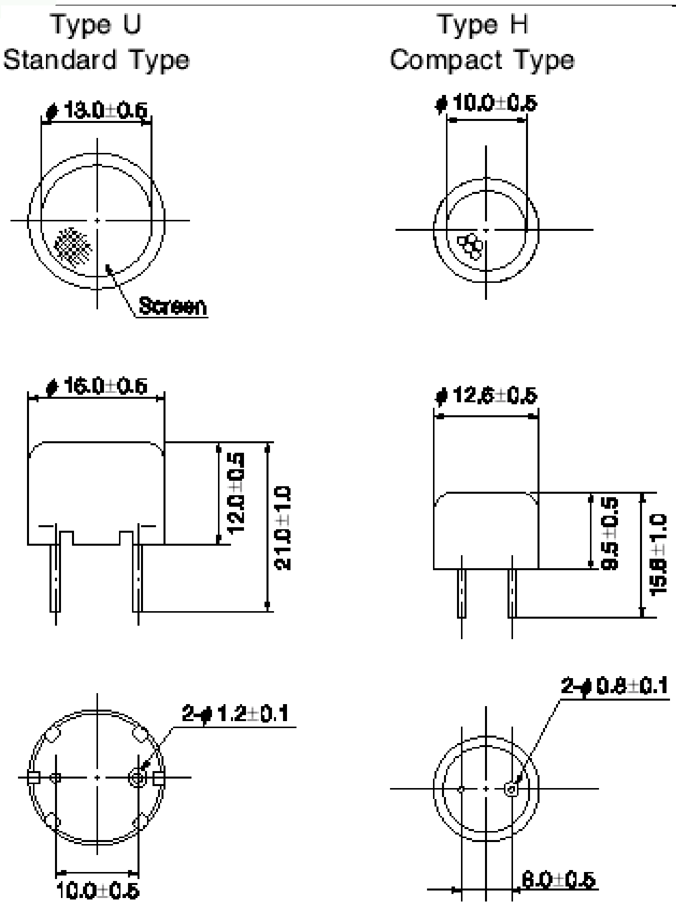


Figure 4: physical dimensions of the ultrasonic transducers (in mm)

3.1.3. FPGA: Altera Flex 10K

The main features for this device are listed as follows:

- Embedded array for implementing megafunctions, such as efficient memory and specialized logic functions
- High density
  - 10,000 to 250,000 typical gates
  - Up to 40,960 RAM bits; 2,048 bits per embedded array block (EAB), all of which can be used without reducing logic capacity
- System-level features
  - 5.0-V tolerant input pins in FLEX® 10KA devices
  - Low power consumption (typical specification less than 0.5 mA in standby mode for most devices)
- High speed
  - Embedded gate arrays provide reduced die area and increased speed compared to standard gate arrays.

– ClockLock$_{TM}$ and ClockBoost$_{TM}$ options for reduced clock delay/skew and clock multiplication

This device is used for its large number of I/O pins and logic gates, which is essential in our system. The Receiver Array Matrix (RAM) requires an extensive number of logics for the data conversion between ultrasonic transducers and Base Station, and most of them will be facilitated within the FPGA to minimize the physical size of the system. The high processing speed is also advantageous on increasing Sonitrac's accuracy.

3.1.4. RF Transmitter & Receiver: Wenshing RWS-F916 & TWS-F916

The RF modules used in our design will be bought off-shelf to prevent further extending the scope of the project. The main features of the RF modules chosen are listed below:

- Date Rate: 3K~100 Kbps
- Supply Voltage: 2.7~ 5.5 V
- High Sensitivity Passive Design
- Simple To Apply with Low External Count

These modules are easy to operate, and since triangulation algorithms are timing critical, their high data rate will help us minimize the total system delay as well as enable us achieving higher accuracy. The following figures are the pin diagrams of the modules.

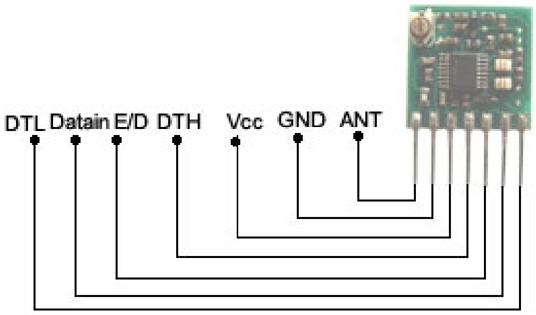


Figure 5: The Pin Diagram of the *TWS-F916* RF Transmitter Module

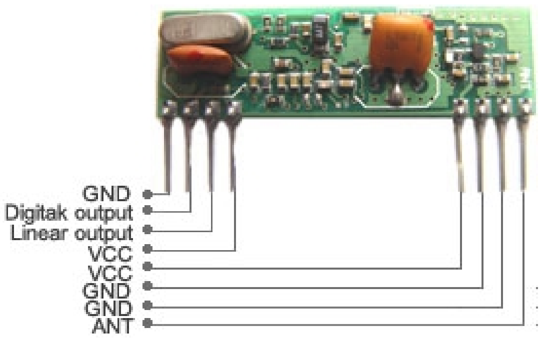


Figure 6: The Pin Diagram of the *RWS-F916* RF Receiver Module

3.2. Receiver Array Matrix

RAM consists of a collection of ultrasonic receivers and a receiver module circuitry. Four poles are set up around the user and each pole has three receivers attached to it. Each receiver is wire-connected to the receiver module. Signal obtained from the receivers are amplified and registered by the receiver module. At last, the receiver module outputs the signal to the base station microcontroller.

3.2.1. Receiver Module Architectural Overview

The diagram below shows one of the twelve receiver circuits. Note that each of the twelve RS latch output connects to the input of the address encoder and each of the twelve peak-hold detector output connects to the input of the multiplexer.

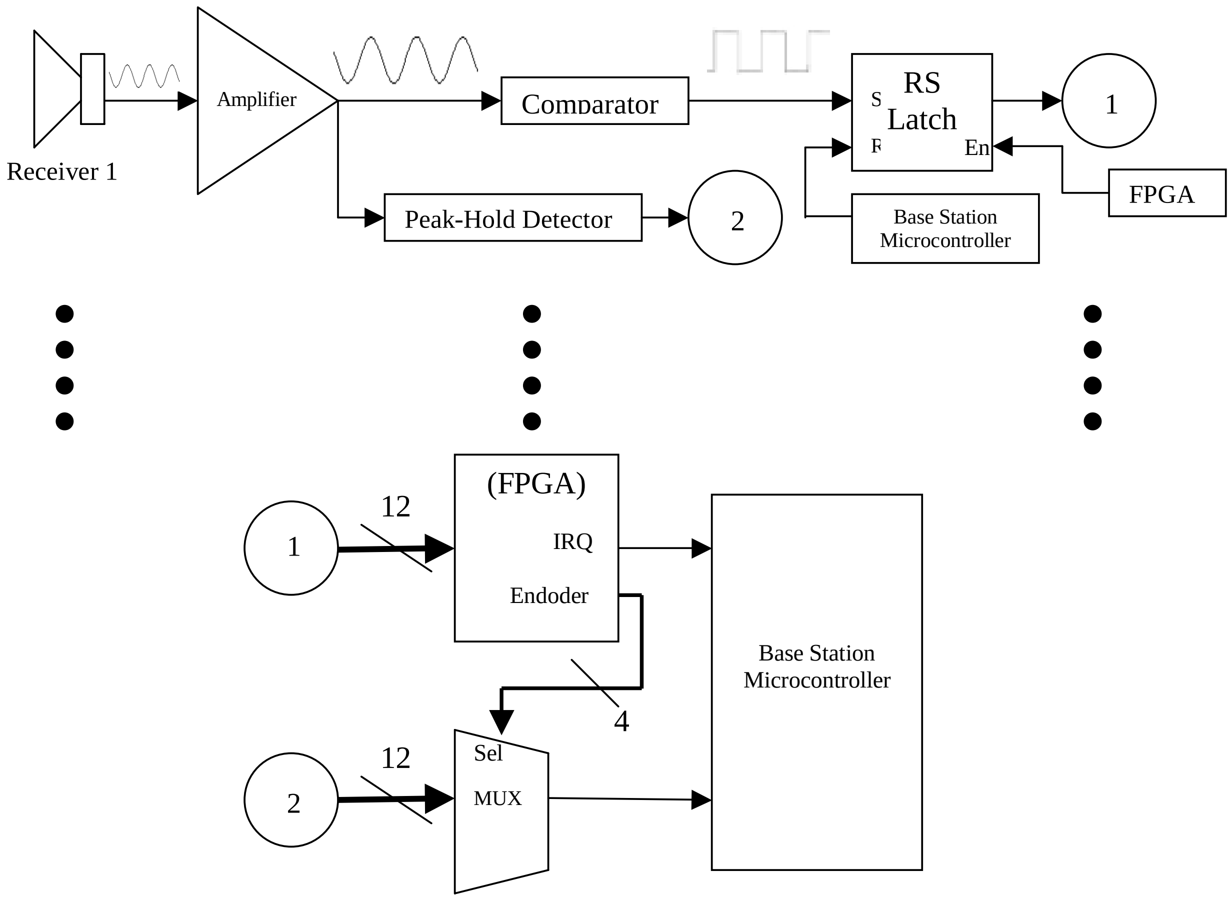


Figure 7: Receiver Module Overview

3.2.2. Peak-Hold Detector

Description

Peak-hold detector is implemented to avoid timing errors caused by the differential input signal strength. The difference between the time when the input signal crosses the threshold value and the time when it reaches the peak is registered and processed by software to calculate the actual input timing.

Refer to Appendix A figure 24 for the full schematic.

3.2.3. Altera EPF10K10LC84-3 FPGA

Description

A 12 to 4 address encoder is implemented with the use of Altera FPGA. The twelve encoder input signals are coming from the twelve RS latches' outputs. The outputs of the encoder control the select input of the multiplexer. The FPGA also controls each of the RS latch enable input. While one of receivers obtains the input signal from the transmitter and after the input data is latched, all eleven other RS latches are disabled to prevent multiple input signal collision. Upon receiving the signal from the ultrasonic receiver, the FPGA generates an IRQ signal to notify base station microcontroller of the incoming signal.

Device Programming

The Altera EPF10K10LC84-3 FPGA is programmed through a configuration device upon power cycling. The configuration device EPC1441 is a one-time programmable EPROM chip.

3.3. Mobile Unit

3.3.1. Hardware

Description:

The mobile unit will consist of a PIC18F242 microcontroller, a RF Receiver Module, Transducer selection circuitry and a body suit. The function of the Mobile Unit is to receive the RF Data from the Base Station, interpret the data, and output the correct signal to the correct Ultrasonic Transducer located on the Body Suit.

Block Diagram:

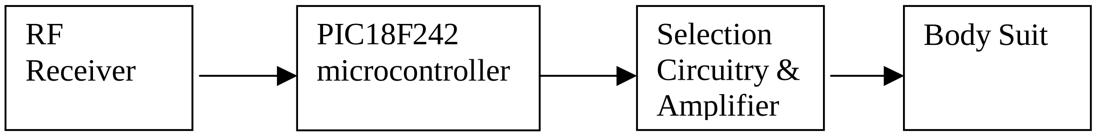


Figure 8: Block Diagram of Mobile Unit

RF Receiver:

A Wenshing RWS-F916 FM RF receiver module is used to receive control data from the Base station. The digital output pin of the receiver will be applied to an input pin of the PIC18F242 microcontroller. The antenna pin will be attached to a small wire protruding from the mobile unit's enclosure.

PIC18F242 Microcontroller:

The microcontroller will be running at 40 MHz in order to minimize the timing discrepancies between the internal clock and real time. In the table below, we show the connections to the PIC18F242 microcontroller.

Table 1: PIC18F242 Connections

Pin/Label	Connected to
RC2/CCP1	Digital output of RWS-F916 (pin 2)
RC1/CCP2	Input pin (pin 1) of Demux CD4067BE
RB0	Channel Select A (pin 10) of Demux CD4067BE
RB1	Channel Select B (pin 11) of Demux CD4067BE
RB2	Channel Select C (pin 14) of Demux CD4067BE
RB3	Channel Select D (pin 15) of Demux CD4067BE

The microcontroller's main function is to read in a stream of 6 bits of data from the CCP1 input pin and determine the correct Ultrasonic transducer to select. This will be controlled by outputting 4 selection bits to a 1-to-16 demultiplexer. The microcontroller will also generate a 40 kHz square wave to the input pin of the demultiplexer.

Selection Circuitry & Output Amplifier:

The selection circuitry & output amplifier consists of a CD4067BE 16-to-1 mux/demux, voltage comparators, and a TPS6140 DC-DC boost converter.

As stated previously, the channel select pins (ABCD) are controlled via the microcontroller. A 16-to-1 Mux/Demux is being used because it would allow for easy implementation of additional Ultrasonic Transducers on the Body Suit.

The correct channel would then have the 40 kHz square wave and would be fed into its corresponding voltage comparator. This voltage comparator will compare the square wave to approx 3V and would rail the output accordingly. The output of the voltage comparator would receive 28V. A DC-DC Boost converter generates this voltage. The output of the voltage comparator would then be applied to the corresponding Ultrasonic Transducer located on the Body Suit.

Schematics for the Selection Circuitry & Output Amplifier can be found in appendix A.

Body Suit:

The Body Suit will consist of a button-up long sleeve shirt with 7 Ultrasonic Transducers attached to it at the locations shown below. The Transducers will be attached to the shirt using a hot glue gun. Also, there will be a Transducer mounted on the user's forehead using a headband.

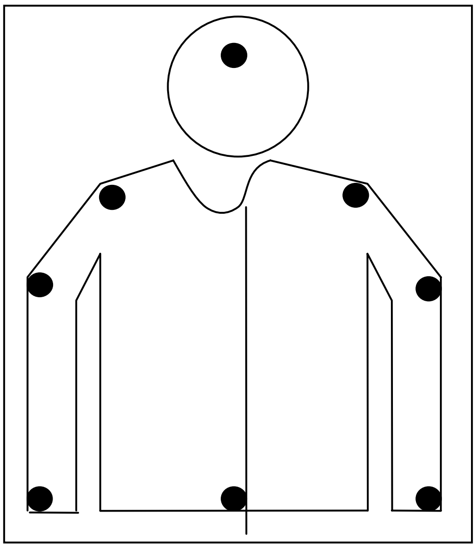


Figure 9: Location of Ultrasonic Transducers on user.

3.3.2. Firmware

Description

The mobile unit will be controlled using a PIC18F242 microcontroller. It will be written in assembly language and will perform the following functions:

1) Interpretation of received control data from the Base Station via RF communication modules.
2) Selection of the corrected ultrasonic transducer as per the control data.
3) Generation of the correct output frequency to be transmitted to the RAM.

Device Programming

The PIC18F242 will be preprogrammed prior to use and requires no further programming upon power cycling.

Device I/O

Refer to Table 2 for pinouts of the PIC18F242.

Table 2: Pinouts of the PIC18F242

Pin/Label	**Function**
CCP1/Input Capture	Input: Received control data from Base Station via RF Receiver Module
RB[0..3]/Standard I/O	Output: Selection bits to channel select of the analog Mux/Demux CD4067BE
CCP2/Output Compare	Output: Output Signal to be transmitted via Ultrasonic Transducer

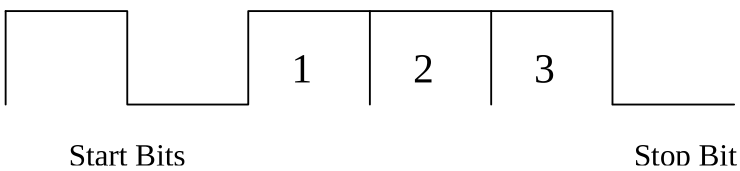


Figure 10: Input Data Format

Where: 1,2,3 represent the Control Bits.

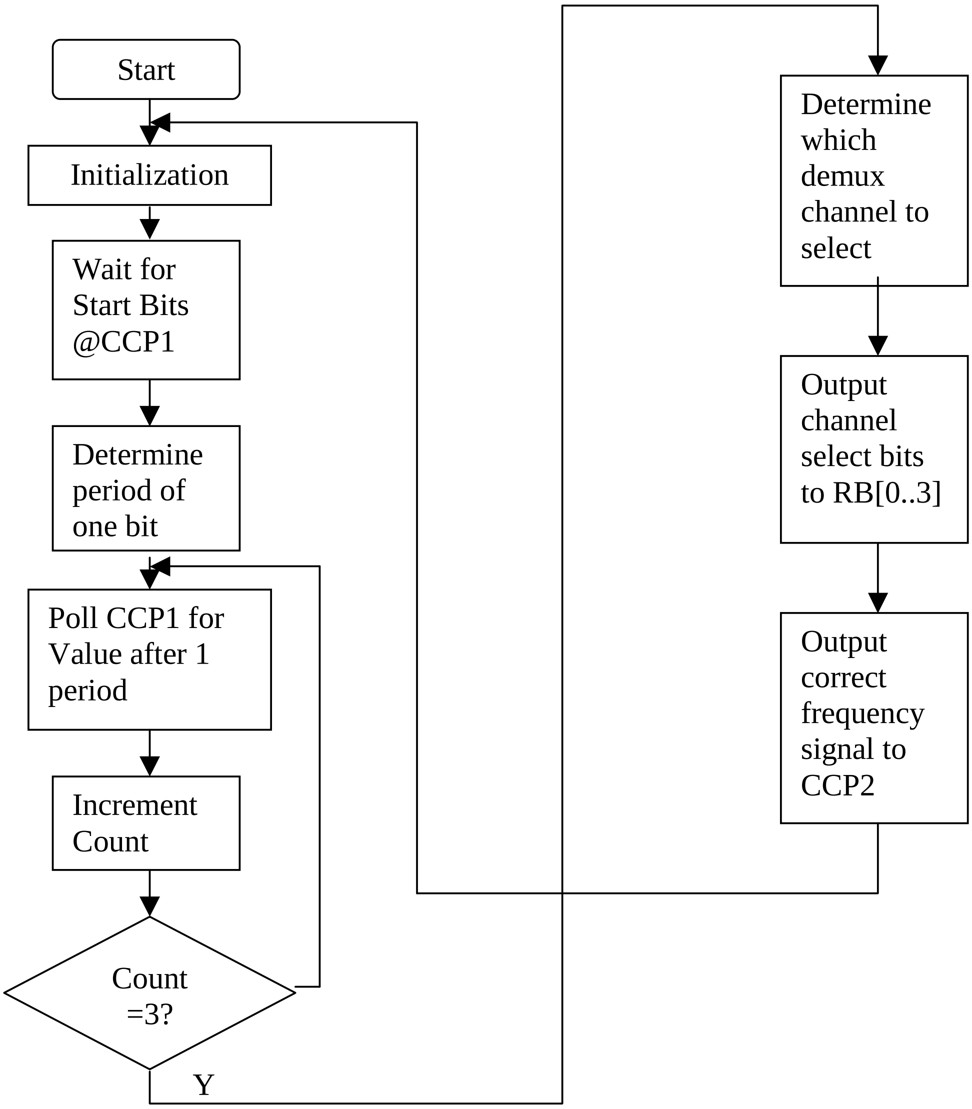


Figure 11: Block Diagram of PIC Firmware

3.4. Base Station

The base station is designed to serve as 1) the power source to RAM and itself, 2) the communication medium between hardware and software, and 3) the central control unit of the hardware module. The hardware and firmware implementation will be discussed separately in the following sections.

3.4.1. Hardware

Power Supply Circuitry

The power supply circuitry will supply stable power to RAM and Base Station. It sources from a regular 120V-60Hz AC outlet and provides regulated 5V DC. The main components of the circuitry include a step-down voltage rectifier and a 7805 voltage regulator. The step-down rectifier takes in 120V AC voltage and converts it to 7.1V DC. The converted voltage is then fed into the 7805 voltage regulator, which regulates and stabilizes the output power. The circuit supplies a minimum average current of 10mA with maximum voltage oscillation smaller than 0.1V. These criteria are essential to accomplish accurate operation of the delay capturing algorithm.

Communication

The TWS-F916 RF Transmitter and MAX 3232 serial communication IC are used for communication with the mobile unit and the PC software respectively.

The input of the RF transmitter is connected directly to the PIC 18F242 microcontroller, the output data format of which follows the description discussed in the previous section.

MAX 3232 IC is implemented to realize the serial communication between the Base Station microcontroller and the PC serial port. The operating circuit is shown in the figure on the right. The IC takes in data at the microcontroller's logic level and converts it to meet the RS232 standard. Output from the IC is then connected to pin 2 of the RS232 DB9 Female connector (refer to Figure 10), which is connected to the PC serial port.

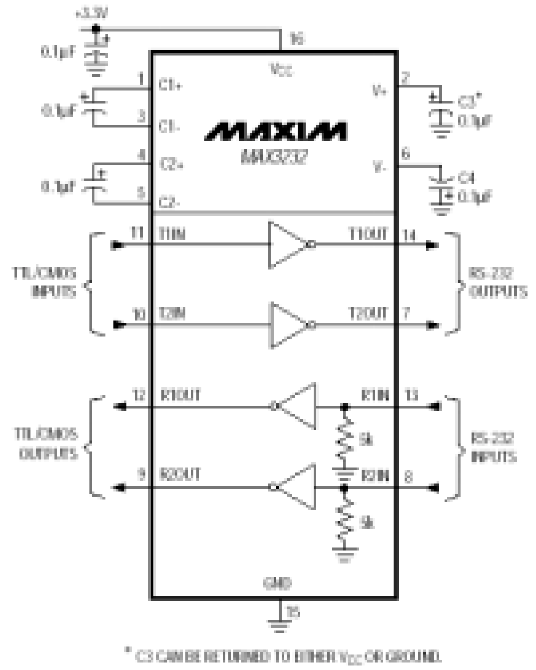


Figure 12: Operation Configuration of MAX 3232

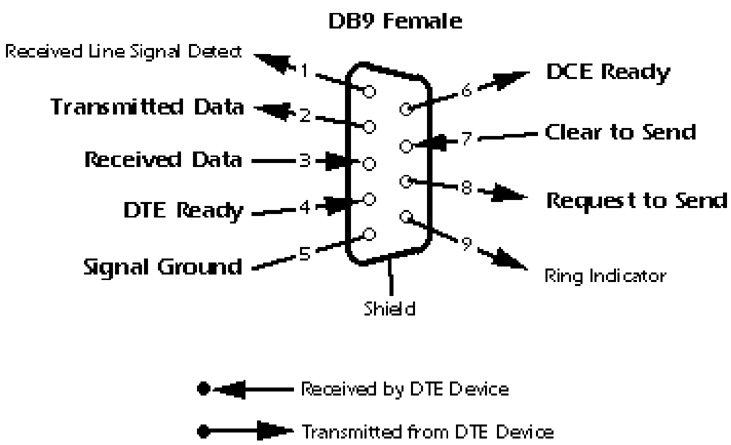


Figure 13: Pin Assignment of a DB9 Female Connector.

The RS232 will be operating on the following settings:

- Byte length: 8 bits
- Parity bit: none
- Stop bit: 1

- Baud rate: 115200
- Handshaking: hardware

The following is a description of a single data block that will be transferred on the RS232. The length of a data block is 15 bytes, and contains the names and times of receivers.

Table 3: Serial Byte Description

Byte	Name	Values	Detailed Description
0	Header	A0	Signifies beginning of transmission
1	Transmitter ID	0 to 7	The ID number of the transmitter that sent out the ultrasonic signals
2	Sensor ID	0 to 12	The ID number of one of the 4 stations that picked up the transmitter signal
3	Time Upper Byte	16-bit unsigned integer	The sound travel time in 0.1 μs
4	Time Lower Byte		
5	Sensor ID	0 to 12	
6	Time Upper Byte	16-bit unsigned integer	
7	Time Lower Byte		
8	Sensor ID	0 to 12	
9	Time Upper Byte	16-bit unsigned integer	
10	Time Lower Byte		
11	Sensor ID	0 to 12	
12	Time Upper Byte	16-bit unsigned integer	
13	Time Lower Byte		
14	Footer	F0	Signifies end of transmission

3.4.2. Firmware

A PIC 18F252 microcontroller is the central control unit of the hardware. It coordinates all operations in the hardware section and communicates with a PC through a serial port. The microcontroller firmware consists of 4 main sections as described below.

Main program

The main loop of the firmware is responsible for serial communication. Since data transmission is not timing critical and can be interrupted anytime without resulting in any error, this function is implemented in the main loop. Figure 14 is a program flowchart illustrating the operation of the main loop.

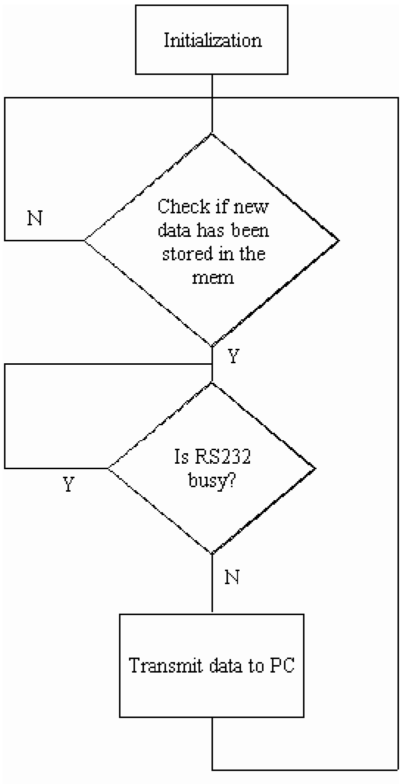


Figure 14: The Program Flowchart of the Main Program.

Input Capture Interrupt Subroutine

The input capture interrupt subroutine registered the time at which an ultrasonic signal is received. It calculates the difference between the time an RF signal is sent out to activate a certain ultrasonic transmitter and the time a response from the transmitter is received by the RAM. The time difference is then stored at the corresponding memory address. The ID of the ultrasonic transducer picking up the response is also registered during this interrupt, and, along with the calculated time delay, it will be sent to the serial port, at a later time. This subroutine also initializes A/D conversion function, which determines the magnitude of the input signal. Figure 15 illustrates the program flow of this subroutine.

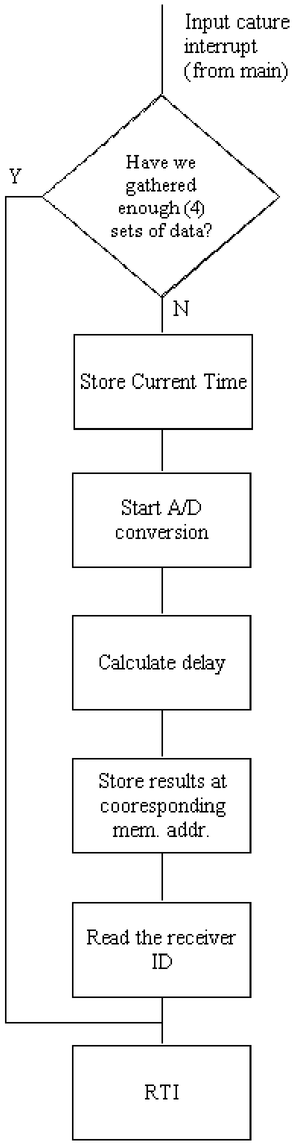


Figure 15: the input capture interrupt subroutine program flowchart

A/D conversion complete interrupt

This interrupt is triggered when an analog to digital conversion is completed. It stores the result of the conversion, which is to be transmitted at a later time. This subroutine also disables the receiver that triggers the interrupt till next update sequence is initiated by the timer 0 overflow interrupt. Figure 16 is the flowchart of this subroutine.

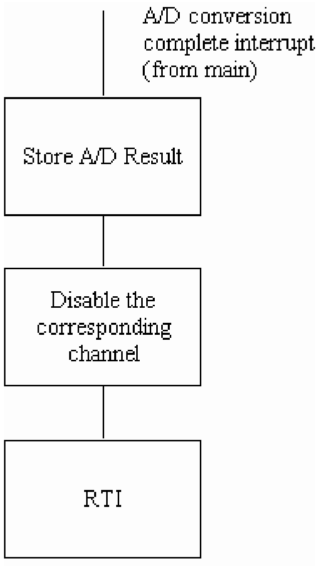


Figure 16: the program flowchart of the A/D conversion complete interrupt subroutine

Timer 0 overflow interrupt subroutine

Timer 0 is set to overflow every 10ms, the maximum time delay for a sound wave to travel from one end of the diagonal to the other of the RAM. This subroutine initializes an update cycle for a target ultrasonic transmitter. It first reset all external hardware (i.e. RS latches), identifies the transmitter to be updated, and then send a corresponding message to the RF module input. Figure 17 shows the program flow of the subroutine.

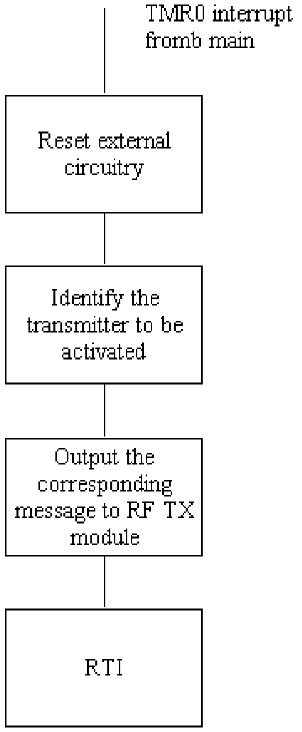


Figure 17: The Program Flowchart of the Timer 0 Overflow Interrupt Subroutine.

4. APPLICATION SOFTWARE DESIGN

4.1. High Level Design

The graphical user interface is an event driven interface based on a PC computer with the computer monitor as the output display medium. The user interface is comprised of various windows as described in the following sections. The design of the software is completed using Microsoft Visual C++ 6.0, using the Microsoft Foundation Class for GUI development.

The following table summarizes the classes used in this program.

Class Name	Base Class	Description	Main Member Functions
CSonitrakApp	CWinApp	This class creates the application. The initialization occurs here.	• InitInstance(): Initialization code, where a Multiple Document Interface template is used to create the main interface window. • OnAppAbout(): Specification of the "About" information for this application.
CMainFrame	CMDIFrameWnd	This class controls the main window of the program. It involves code to handle menu commands as well as a free running timer to continuously update the image of user.	• OnDoneSampling(): This is the handler function when sampling has completed. It will allow create the Data Plot windows, and allow the windows to generate the proper graphs. • OnCommandStart(): This function is used after the user has accessed the "Start" command from the menu, and specified their plotting options. • OnTimer(): This timer is used for sampling. • OnTimer2(): This timer is used to update the orthogonal image.
ChildFrame	CMDIChildWnd	This class is used as the basis for the child windows created.	• PreCreateWindow(): Used to specify size and location of window.

CsonitrakView	CView	This class handles the drawing of the 2D orthogonal figure	• OnDraw(): This function handles the drawing of the User
CsonitrakDoc	CDocument	This class handles the information that the drawing is based on	•
CGetData	CDialog	This class captures the plotting options specified by the user once the 'Start' command from the menu is selected	• OnOK(): Once the user presses ok, all the plotting options is relayed back to the main window.

Table 4: Graphical User Interface Overview

4.1.1. Main Window, 2D Figure Child Window and Application

In the main window, the user will be able to see the orthogonal 2D representation of the user on the left hand screen. This will continuously update and will thus track the position of the user visually. From this window, the user can start capturing data to plot, by going to the menu 'Command', and choosing 'Start'. This will pop up a new window, which will allow the user to control parameters of data capture. Please see next section for more details on the Data Capture window. The following is a screen capture of the Main Window.

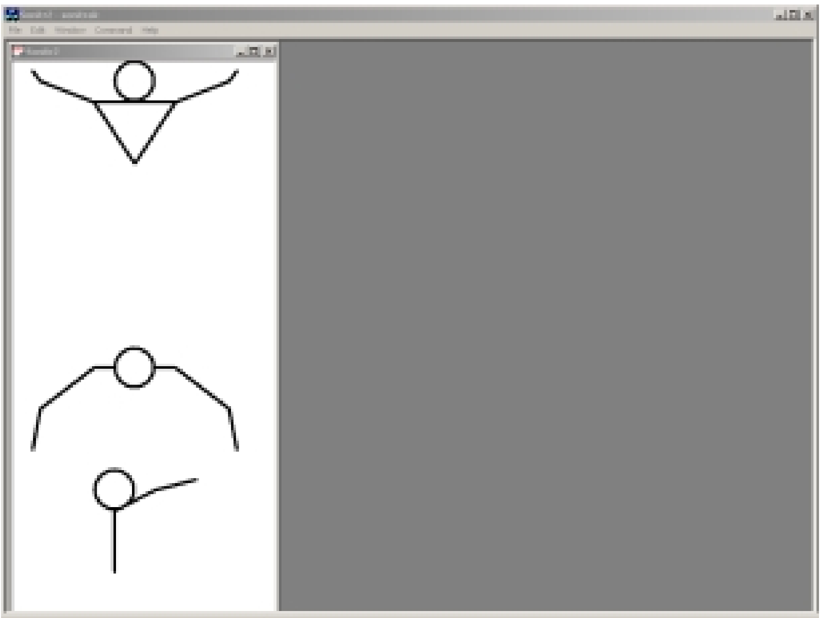


Figure 18: Screen Capture of Main Window

All windows are derived from the MFC classes. The main application is derived from CWinApp class, and the Main Window is derived from CMDIFrameWnd Class. In the InitInstance() function of the application class, the MainWindow as well as the Child Window which displays the 2D Orthogonal representation of the user are created. The 2D figure is updated with a free-running timer. The figure is drawn using built in MFC drawing functions.

4.1.2. Data Capture Window and Data Capture Sequence

The following is a screen capture of the Data Capture Window.

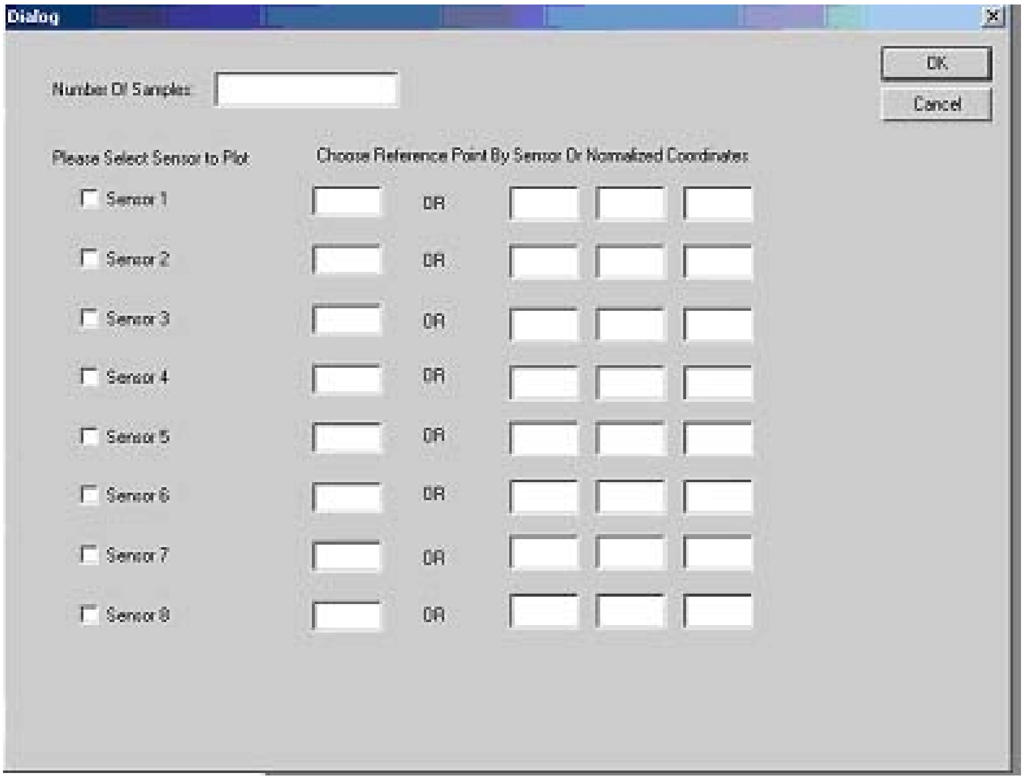


Figure 19: Screen Capture of Capture Window

First of all, the user is asked to answer the number of samples they wish to capture. Then, on the left column, they are asked to check the sensors that they wish to plot. For each sensor, the user is asked to enter in a point of reference to plot the data. This can be accomplished by 1) entering in a number corresponding to another sensor (ie. Sensor 2's position can be used as reference for Sensor 1's plotting) or 2) entering a specific normalized coordinate for the reference.

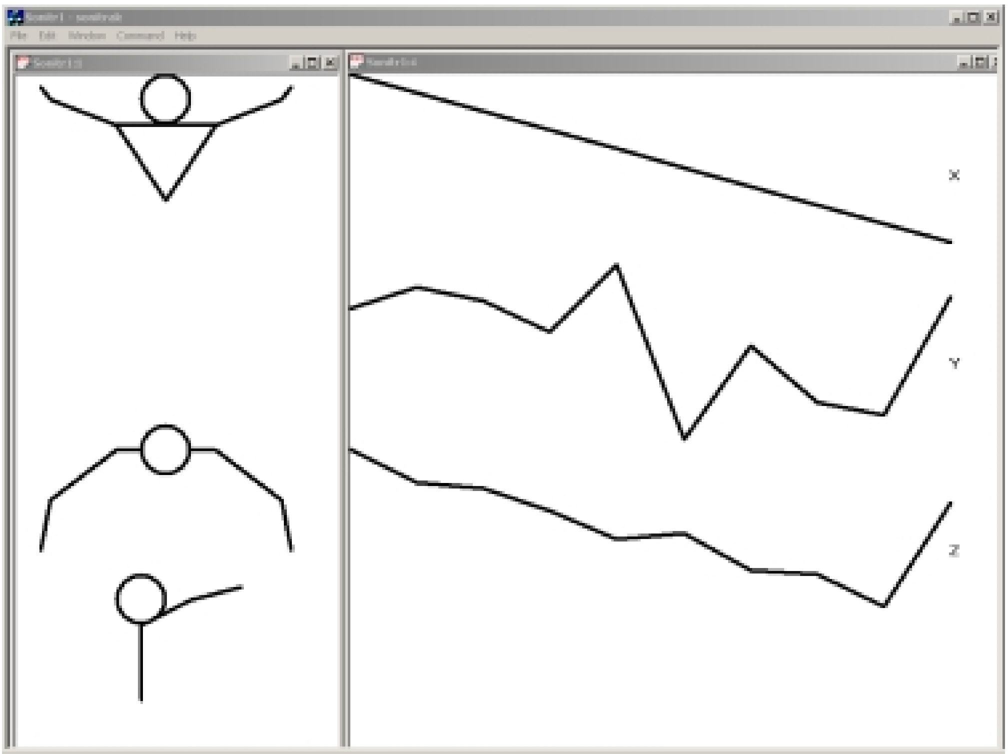
Once the user is complete their choice, they should press 'OK'. If they wish to cancel the data capturing sequence, they simply press 'Cancel'.

Once the user selects which sensor values to plot, the Main Window will receive these selection values, and will start saving the proper data. Once the number of samples have been captured, the corresponding Plot Windows will be created for each sensor. See next section on details of the Plot Windows.

4.1.3. Plot Window

Plot windows can range anywhere from 1 to 8. Each window consists of 3 graphs, one for each coordinate. They are plotted with reference to the reference point chosen for each sensor. The graph is generated using built in drawing functions of MFC, which are line graphs representing the position of a particular sensor. Below is an example of what a

Plot Window would look like. Once the user has finished looking at the graph, they can simply close the window by clicking on the ‘x’ button.



4.2. Low Level Design

4.2.1. Serial Interface

Interfacing the software application with the hardware is accomplished through the use of the serial port of a PC computer. The CSerial library from reference 6 is used to assist in manipulating the serial port and reading from the port’s buffer to obtain information from the hardware. The CSerialDriver implements the necessary functions needed for our particular application.

4.2.1.1 CSerialDriver::Init()

This method initializes the serial port by first opening the serial port for reading and writing. It then initializes the port by setting the baud rate, data bits, parity, handshaking, and timeouts. The specific settings used for the serial communication can be found in section 3.4.1. Various errors are defined in the CSerial library and are reported if any of them occurred during the initialization of the serial port.

4.2.1.2 CSerialDriver::Read()

This method calls the CSerial::waitEvent() function which waits for a particular event, such as an break, CTR, DTS, error, ring, RSLD/CD, and data receive events. The data receive event is the event necessary for the retrieval of data from the serial port. When this event occurs, the functions processes and forwards the data retrieved in the serial buffer.

4.2.2. Triangulation

The triangulation class is responsible for determining the coordinates of a transmitter in 3-dimensional space when given a set of times from varying receivers. The triangulation task has two major tasks:

1. The triangulation class must convert the serial data it retrieves from the serial interface into valid representatives of the time elapsed for various receivers to communicate with the transmitter. The transmitter number and receiver numbers are also retrieved through the serial data to allow the triangulation algorithm to properly determine the position of each transmitter.

2. The triangulation class's main purpose is to triangulate the position of the transmitter given the elapsed time of the first four receivers available. Using linear algebra and numerical approximations, the normalized coordinates in 3-space is calculated.

4.2.2.1 CTriangulate::Convert()

This method is the main function of the CTriangulate class which implements both of the major tasks including reading and converting the serial data into proper elapsed times and using the times to triangulate the position of the corresponding transmitter in 3-dimensional space. The triangulation algorithm can be described as follows:

The algorithm is based on the distance equation

$$X^2 + Y^2 + Z^2 = R^2, \qquad (1)$$

and Gaussian Elimination in order to solve a system of equations. Because Gaussian Elimination must be performed on a set of linear equations, the 4 distance equations from the 4 receivers must be translated into a set of 3 plane equations. The plane equation

$$aX + bY + cZ = N, \qquad (2)$$

can then be entered into a matrix.

The following is a detailed description of the steps required in the triangulation algorithm. Please refer to figure 20 for a geometrical interpretation of step 3.

1. Convert from raw byte data to floating point number
2. Adjust for hardware delay and convert signal travel time to distance
3. Repeat steps a to g 3 times to create a set of 3 linear plane equations to solve
    a. Select the data from 2 receivers
    b. Determine which receiver is farther from the transmitter to use as origin of new temporary coordinate system *(receiver A)*
    c. Find the distance between the 2 receivers from database *(distance D)*
    d. Use cosine rule to find the acute angle between the sensors and the transmitter (*angle theta*)
    e. Use the above angle for sine equation to find the vertical distance to the plane as a ratio of D (*ratio offset*)
    f. Use the above ratio to find a point on the plane in the old coordinate system. (*Point P*)
    g. Use P and D to form the equation of the plane and enter into a row on the matrix.
4. Use Gaussian elimination to solve for the intersection of the 3 planes:
    a. Repeat i to iii for each row of the 3x4 matrix:
        i. Check to see if we need to perform row exchange in order to acquire a non-zero entry on the diagonal
        ii. Perform a row exchange if necessary
        iii. Eliminate entry(s) by performing linear operation between two rows
5. Perform backwards substitution:
    a. Perform division on last row of matrix to find answer 3 (Z coordinate)
    b. For every other row, from 2 to 1:
        i. Substitute answer(s) from previously calculated rows to find answer for current row

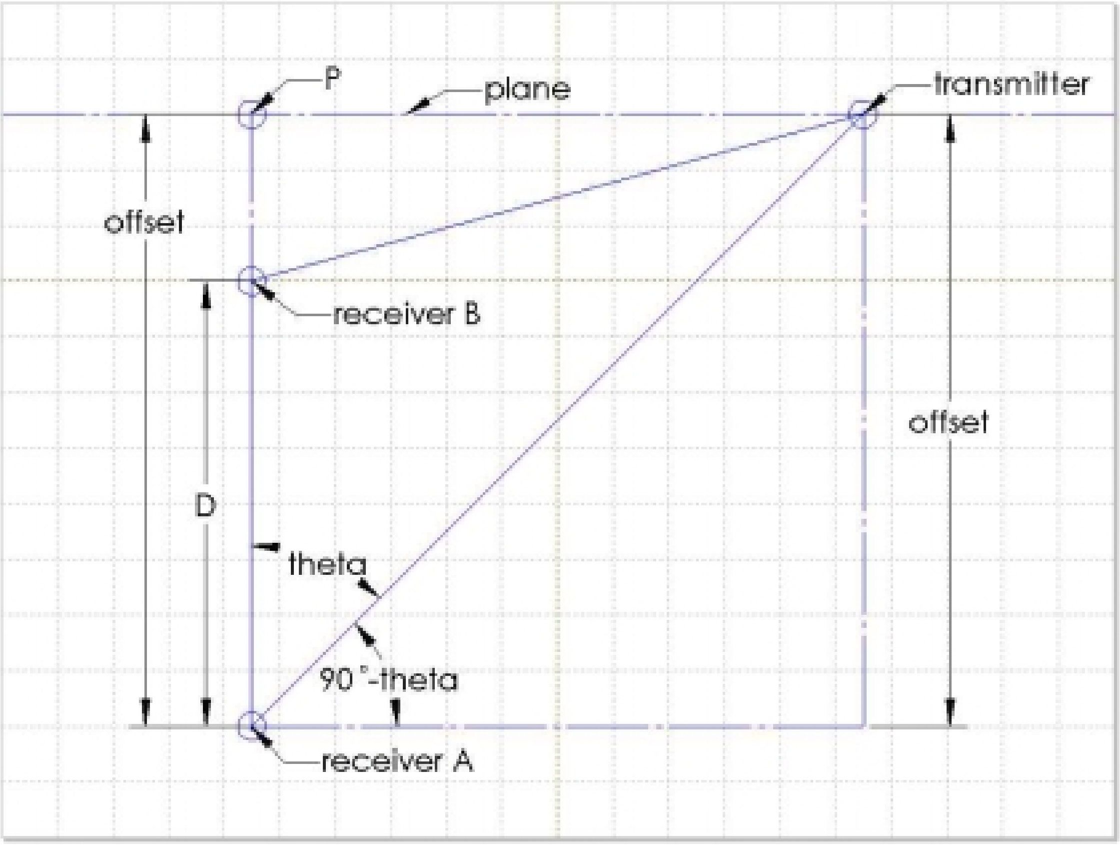


Figure 20: Geometrical Representation of Triangulation Algorithm

5. TEST PLAN

In order to ensure that we were successful in our implementation of the design specifications, we have devised several test plans. Several tests will be conducted at intermediate stages to examine the functionality of each module individually; other tests are developed to ascertain that Sonitrac is properly integrated and the system performance is above our standard. All these tests shall be passed for the system to function as expected.

5.1. TH.1 Ultrasonic Transducers

Tests on the transducers will concentrate on their sensitivity over distance and different orientations.

Procedure

A 40KHz, 20V peak-to-peak sinusoidal signal will be applied to the ultrasonic transmitter terminals. The output from the receiver terminals will be measured along with the relative position and orientation of the two units. The data will be collected for determining whether the transducer will be suitable for our application or whether any modification (i.e. physical changes and etc.) will be necessary.

Pass Condition

The receiver is able to receive the transmitted signal from the transmitter with peak-to-peak signal amplitude at least 20mV. The separation between the transmitter and the receiver should be at least 2.5 m with their relatively orientation 45 degrees apart.

5.2. TH.2 RS232 Communication

Tests will be conducted to ensure the reliability of hardware to PC communication.

Procedure

Test data will be sent from the base station to the PC through the serial communication hardware circuitry. A test program will be written to display the received data, which will be compared to the original test data. Modification on the transmission baud rate will be modified on the microcontroller side to achieve more reliable communication.

Pass Condition

The received data matches the transferred data with over 99% accuracy.

5.3. TH.3 RF Communication

Tests will be conducted to ensure proper data communication between the Receiver Module and the Transmitter Module.

Procedure

Pre-determined test data following the format described in section 3.3.2 will be generated by a PIC microcontroller and applied to the transmitter module. The output of the receiver module will be picked up by another microcontroller, which compares the received signal with the pre-determined data and counts the number of errors that occurs. The result will be used for modification on the proposed data format or other adjustments if necessary.

Pass Condition

The Received Data matches the Transferred Data with 99% accuracy

5.4. TH4 Power-stress Test

Since the sustainability and quality of the power circuitry affects the accuracy of the system (via the amplifier, peak-hold-detector of RAM and etc.) tests will be conducted to ensure the durability and output quality of the power supply circuitry.

Procedure

1) The power circuitry will be connected to a regular 120V 60Hz outlet with its switch turned on over 72 hours.
2) The output of the circuit will be monitored on an oscilloscope. The maximum ripple voltage will be recorded for evaluating the circuit.

Pass Condition

1) All units of the circuitry are still functional after the stress test.
2) The output ripple voltage does not exceed 0.2V.

5.5. TS1 Triangulation Algorithm

Tests will be conducted to determine if the triangulation algorithm correctly calculates the position of the transmitter.

Procedure

1) Select a random point in 3D space, calculate its distance to 4 receivers and attempt to calculate the original point using the algorithm.

2) Send a set of times from the base station to the PC and have the algorithm calculate the position of the original point.

Pass Condition

1) The algorithm must correctly calculate the original position for all of 1000 trials to accuracy within 0.5cm.
2) The software must correctly interpret raw data from serial port and correctly calculate the original position to accuracy within 0.5cm

5.6. TS2 Graphical User Interface

The graphical user interface must function as stated in order to maintain a user-friendly experience.

Procedure

1) Attempt to select every command available as documented by the design specifications.

Pass Condition

2) Every command must function as written in the design specifications.

5.7. Overall Performance

Tests will be conducted to determine the performance of the overall circuitry and modifications will be made to meet requirements as per the functional and design specifications.

Procedure

1) The mobile unit will be fixed in a 3D space defined by RAM and the hardware circuitry will be activated to triangulate the position of the unit. If the system functions properly, the results yielded each time should be constant or with very little discrepancy.
2) Two of the transmitters will be place at different positions will their distances measured, and their positions shall be estimated with our system. Again, the estimated result should agree with the measured values.

Pass Conditions

1) The triangulation results agree with each other with less than 0.5 cm difference.
2) The triangulation results does not differ from the measured result by over 0.5cm.

6. CONCLUSION

This concludes the Design Specifications of Sonitus Solution's Sonitrac. Stage one of R&D is scheduled to be completed by December, 2003. Scheduling for Stage two will be contingent on the results of our proof-of-concept module.

7. Referenced Materials:

1. http://www.cs.sfu.ca/~amulder/personal/vmi/HMTT.pub.html
2. http://www.cs.sfu.ca/~amulder/personal/vmi/HMTT.add.html#acoustic
3. http://www.wave-report.com/tutorials/MoTrak.htm
4. http://www.ensc.sfu.ca/users/whitmore/public_html/courses/305/305.htm
5. http://www.ensc.sfu.ca/undergrad/euss/essef/index.html
6. http://www.codeproject.com/system/serial.asp

Appendix A

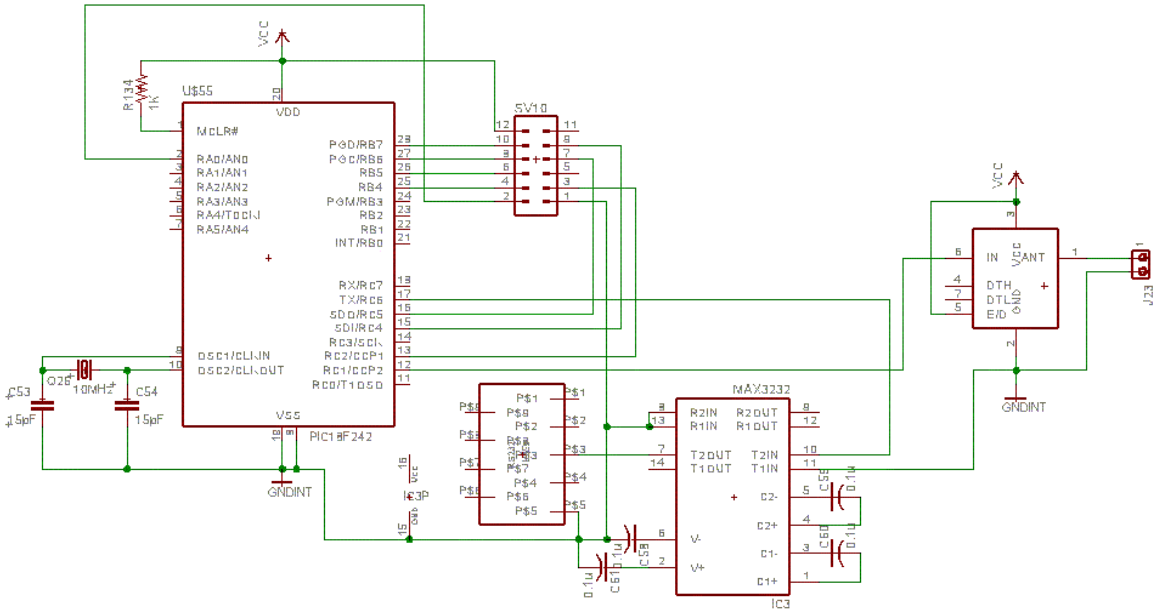


Figure 21: Base Station Circuit Diagram

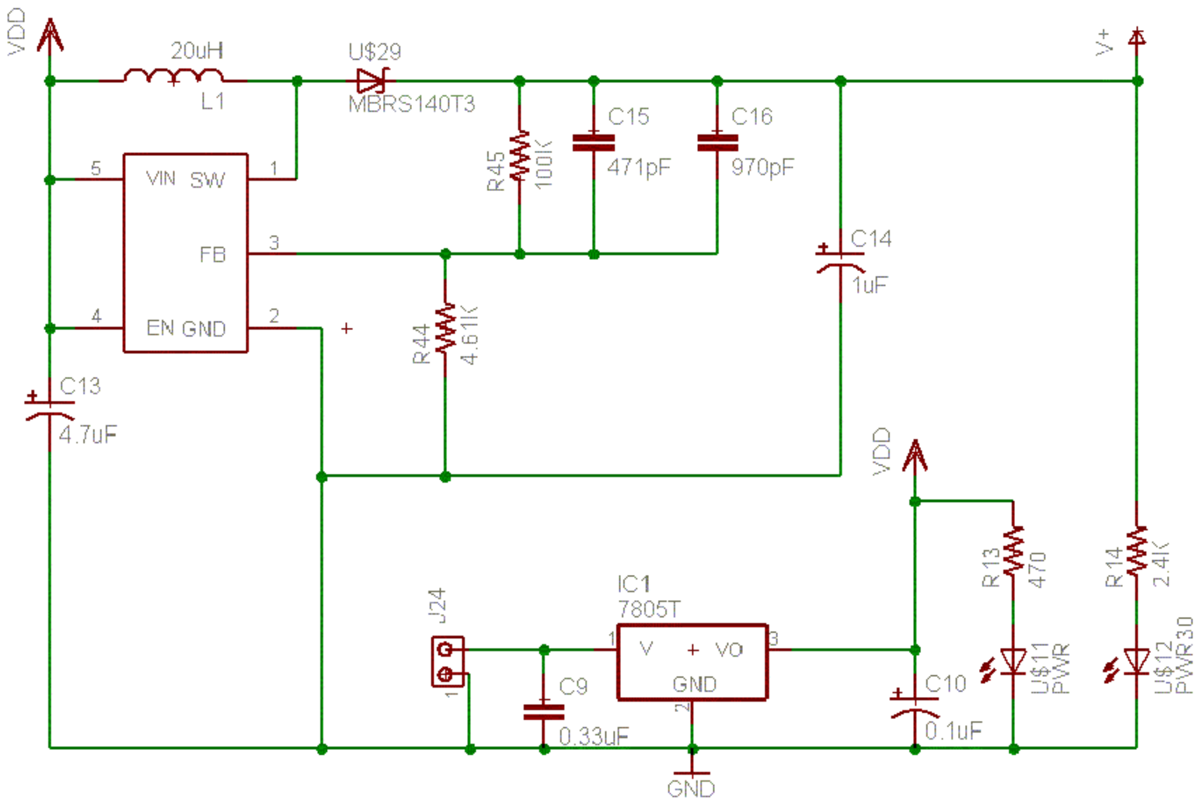


Figure 22: Boost Converter Circuit Diagram

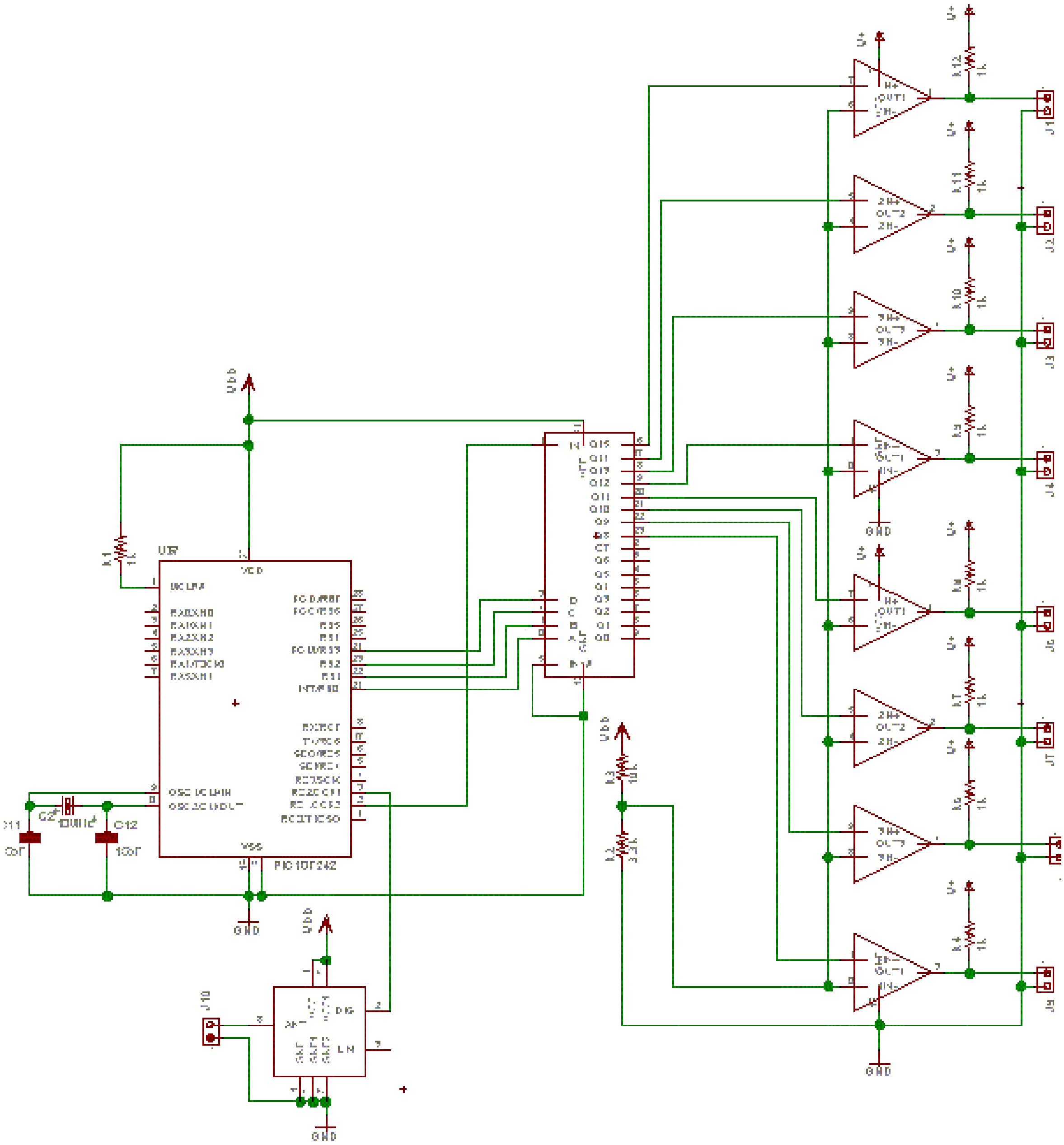


Figure 23: Mobile Suit Circuit Diagram

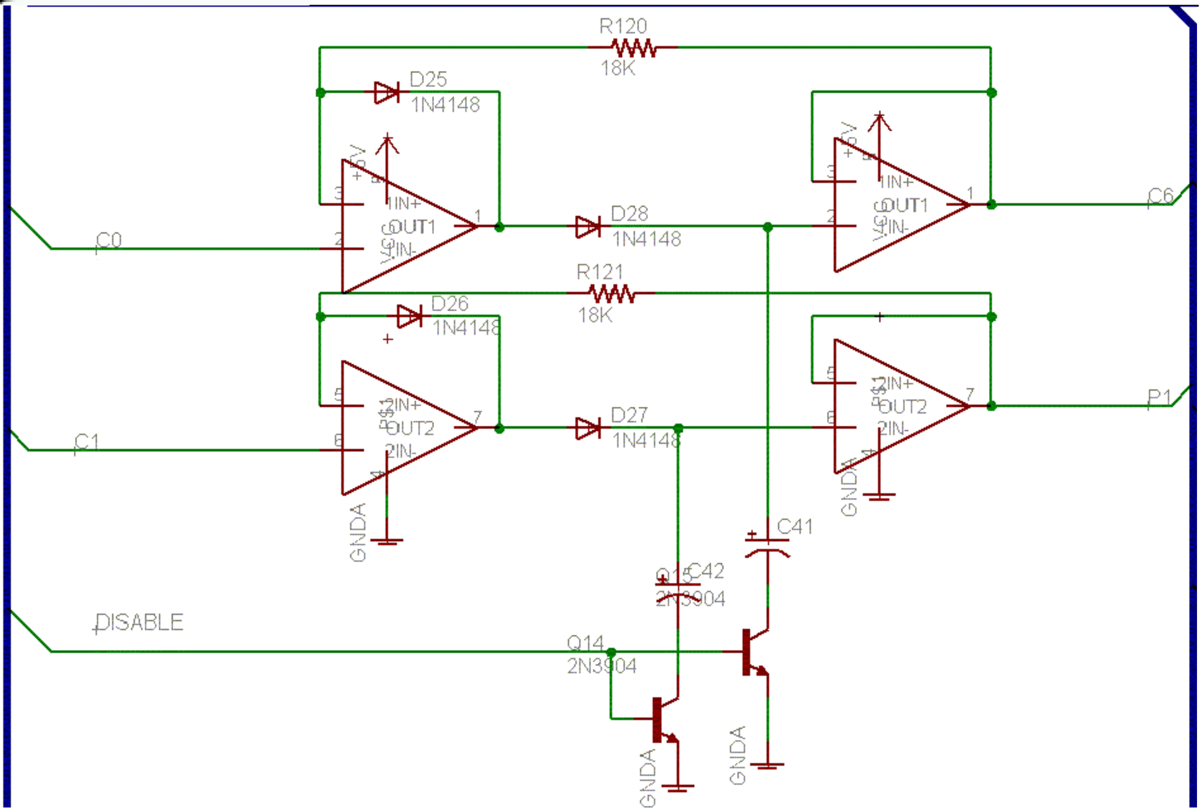


Figure 24: Peak-Hold Detector Circuit Diagram

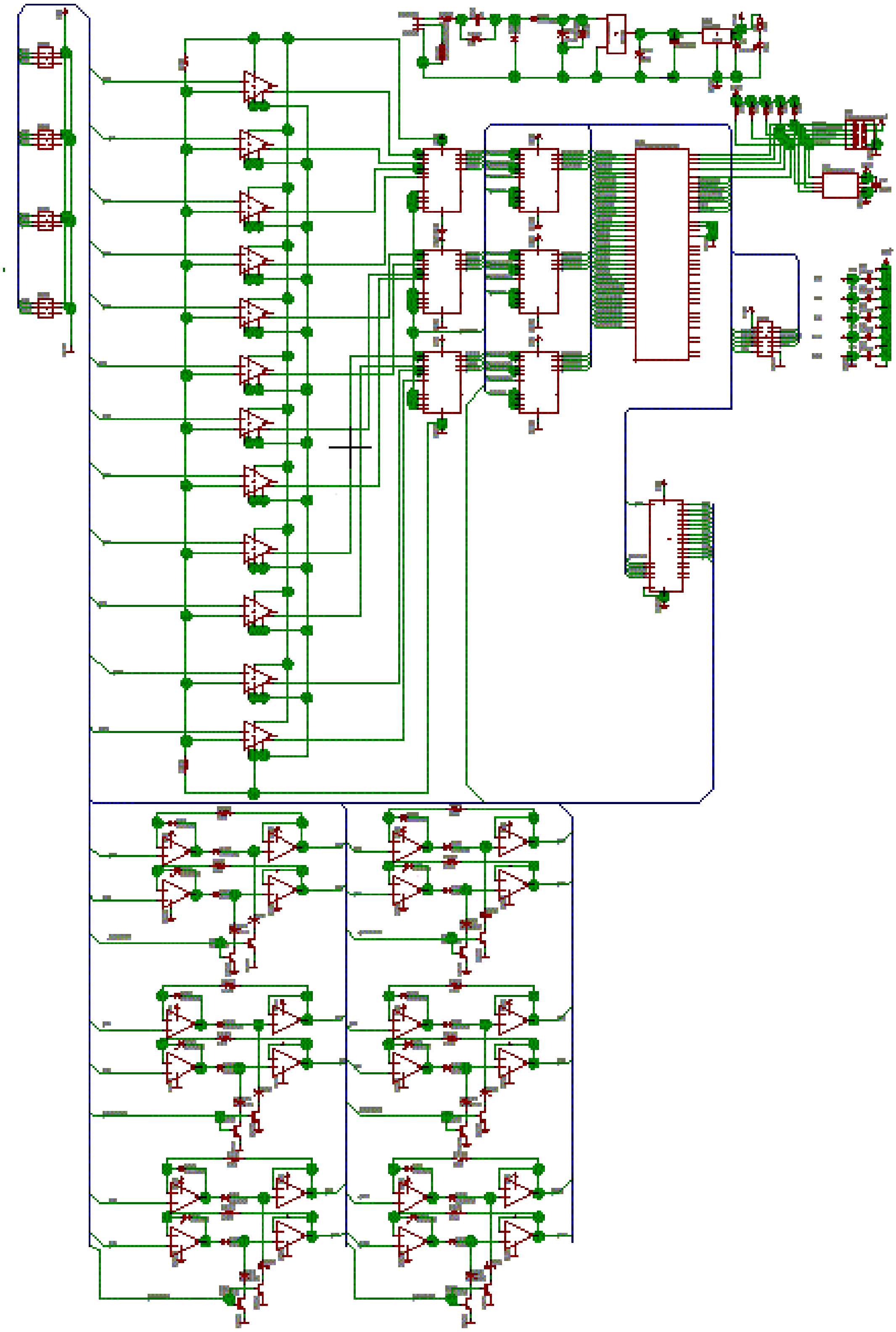


Figure 25: Receiver Array Matrix Circuit Diagram

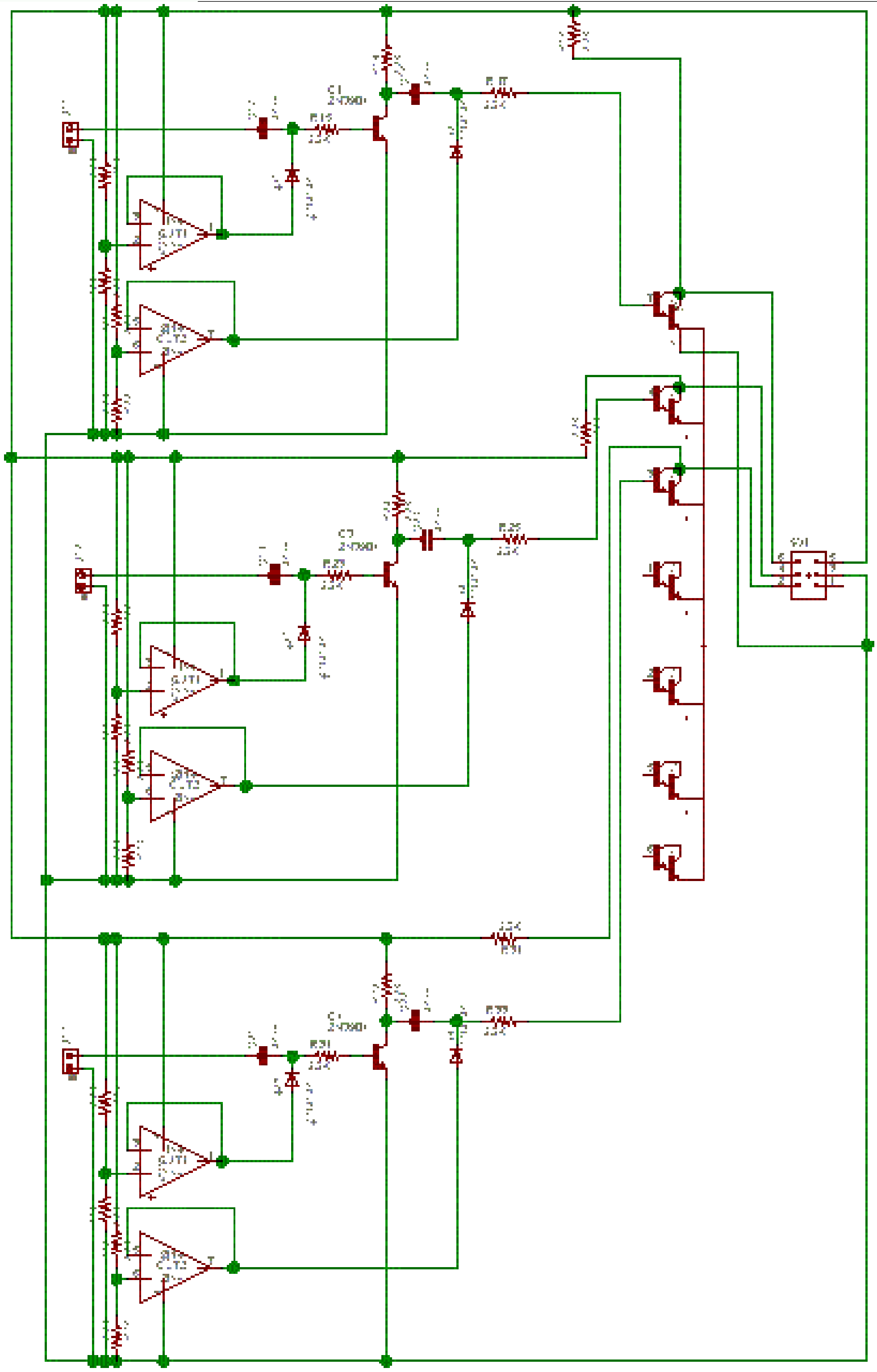


Figure 26: Selection Circuitry & Output Amplifier Circuit Diagram