Ricochet

Systems Ltd.

March 5, 2007

Steve Whitmore
School of Engineering Science
Simon Fraser University
Burnaby, British Columbia
V5A 1S6

Re: ENSC 440/305 Design Specification for the Ricochet Tracking System.

The attached document, *Design Specification for the Ricochet Tracking System*, lists detailed design specifications of our project for ENSC 440 (Capstone Project). Our goal is to design and implement a tracking system for the ice hockey puck.

The purpose of this design specifications document is to provide detailed design parameters and testing procedures that will be fulfilled by the Ricochet Tracking System (RTS). This document outlines design specifications, which will be completed by the project deadline for the proof-of-concept RTS.

Ricochet Systems Ltd. consists of three innovative senior Engineering Science students: Ashkan Tehrani-Nejad, Kelsey James Regan and Imran Dewji. If you have any questions please contact me at (778) 288-4274 or via e-mail at ensc440-group5@sfu.ca

Sincerely,

Ashkan Tehrani-Nejad

CEO
Ricochet Systems Ltd.

Enclosure: Design Specification for the Ricochet Tracking System

Design Specification for the Ricochet Tracking System

Team: Ashkan Tehrani-Nejad
Kelsey James Regan
Imran Dewji

Contact: Ashkan Tehrani-Nejad
ensc440-group5@sfu.ca

Date: March 5, 2007
Version: 1.0

Abstract

Every goal during the NHL hockey game is reviewed by a team of judges at the NHL headquarters. Questionable goal judgments during NHL hockey games are the topic of much debate in this sport. Many times during the season fans must wait through a long review process while the NHL officials determine whether or not a puck has crossed the goal line. This lengthy process not only takes away from the excitement of the game but more importantly causes frustration among fans and players with inconsistent calls.

This document outlines the design for a puck tracking system, which employs ultrasonic technology to take measurements of the puck to precisely determine its position and velocity and approximate the puck orientation.

The Ricochet Tracking System (RTS) consists of a hockey puck with an embedded microcontroller and a set of ultrasound transmitters, an array of ultrasound receivers, and a Windows PC based data capture and processing system. The design outlined in this document meets the requirements set out in the functional specification [1].

Table of Contents

Abstract ..... **ii**
Glossary ..... **v**
1 Introduction **1**
2 System Overview **2**
3 Puck / Transmitter Design **3**
3.1 Overview 3
3.2 PIC Microcontroller 3
3.2.1 System Settings 4
3.2.2 Ultrasonic Waveform 4
3.2.3 Output Enable Signal 4
3.3 Ultrasonic Buzzers 5
3.4 Power Supply 5
4 Receiver Design **6**
4.1 Overview 6
4.2 Ultrasonic Microphone 6
4.3 Amplifier 6
4.4 Band-Pass Filter 7
4.5 Envelope Detector 8
4.6 Power Supply 8
5 Software Design **9**
5.1 Overview 9
5.2 Interface Board 9
5.2.1 Calibration Routine 9
5.2.2 Interrupt Service Routines 10
5.2.3 Data Transfer Routine 10
5.3 Windows Software 11
5.3.1 Communication 11
5.3.1.1 Idle 12
5.3.1.2 Data Acquisition 13
5.3.1.3 Computation 13
5.3.2 Graphical User Interface 14
6 Test Plan **15**
6.1 Overview 15
6.2 Puck/Transmitter Unit Tests 15
6.2.1 Power Source 15
6.2.2 Ultrasonic Buzzers 15
6.2.3 PIC Microcontroller 15
6.3 Receiver Unit Tests 15
6.3.1 Power Supply 15
6.3.2 Ultrasonic Microphone 16
6.3.3 Instrumentation Amplifier 16

6.3.4 Band-Pass Filter 16
6.3.5 Envelope Detector 16
6.4 Interface Board Unit Tests 16
6.4.1 Calibration Test 16
6.4.2 Data Capture Test 16
6.5 Software Unit Tests 17
6.5.1 Startup Test 17
6.5.2 Tracking Test 17
6.5.3 Recording and Playback Test 17
6.6 Integration Tests 17
6.6.1 Setup Test 17
6.6.2 Positioning Test 17
6.6.3 Passing Test 17
6.6.4 Slapshot Test 17
6.6.5 Goal Judge Test 18
6.6.6 Long Term Test 18
7 Sources and References 19

List of Figures

Figure 2-1: Puck transmitting signal to sensors, adapted from [4]. 2
Figure 3-1: Block Diagram for the Transmitter Sub-System 3
Figure 3-2: PIC16F684 PDIP pin diagram [5] 3
Figure 3-3: System Algorithm. 4
Figure 4-1: Receiver Block Diagram 6
Figure 4-2: Amplifier Schematic 7
Figure 4-3: Band-Pass Filter Schematic 7
Figure 4-4: Envelope Detector Schematic 8
Figure 5-1: Interface Board/Windows PC Block Diagram 9
Figure 5-2: Interface Board Block Diagram 9
Figure 5-3: State diagram for retrieving data and computing the position 12
Figure 5-4: Picture showing the types of measurements being taken and sent 13

List of Tables

Table 5-1: Data Transfer Format 10
Table 6-1: Power Supply Voltages [11] 15

Glossary

A/D *Analog to Digital*.

ATX *Advanced Technology Extended.* A physical specification for computer hardware including the motherboard, case, and power supply.

Dipyramid Two pyramids symmetrically placed base-to-base [2]. A square dipyramid has a square base.

EEPROM *Electrically Erasable Programmable Read Only Memory.*

NHL *National Hockey League*. A North American league of professional hockey teams.

MCU *Micro Controller Unit.*

PWM *Pulse Width Modulator.*

RTS *Ricochet Tracking System*.

Ultrasound Sound that has a frequency above the threshold of human hearing of 22 kHz.

1 Introduction

The Ricochet Tracking System (RTS) is a system for tracking the movement of a hockey puck around the ice rink in three dimensions during the course of a game. It is born from the need to determine if the puck has crossed the goal line during a goal dispute, but is useful throughout the course of the game. The RTS can track the position of the puck, its orientation, its height above the ice surface, and its speed.

This document defines the design of the RTS in compliance with the functional requirements defined in the functional specification [1]. The whole system is divided into functional units each of which has its own set of requirements. The intended audience for this document is the designers who will build the RTS from the design specified within. Requirements are referenced by number to allow for tractability to the functional specification.

2 System Overview

The Ricochet Tracking System consists of as ultrasound-emitting hockey puck, an array of ultrasonic sensors around the perimeter of the play area, and a computer used to capture and process the data collected from the sensors. RTS uses time of flight position triangulation to calculate position of the puck around the play area. This design follows from our investigations in our proposal [3] and meets the requirements outlined in the functional specification. Figure 2-1 illustrates a scenario where the puck transmits a signal to the blue sensors around the perimeter of the ice surface.

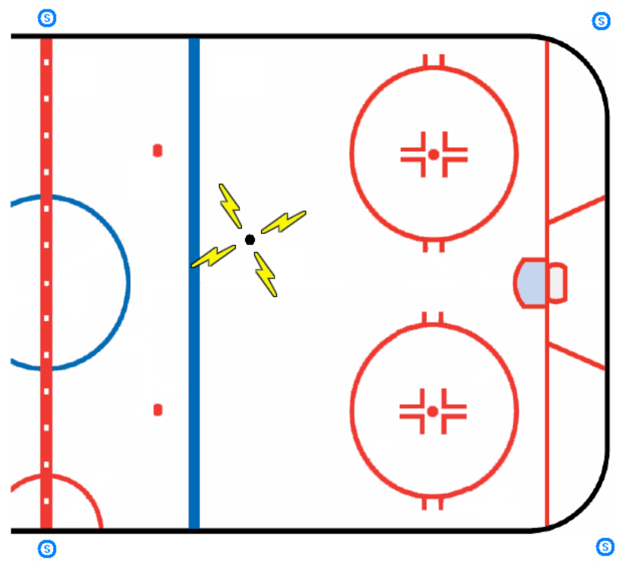

Figure 2-1: Puck transmitting signal to sensors, adapted from [4].

3 Puck / Transmitter Design

3.1 Overview

The puck subsystem will consist of six ultrasonic transmitters placed on vertices of a square dipyramid. The signal transmitted by the planar transmitters will be distinguishable from those on the vertical axis.

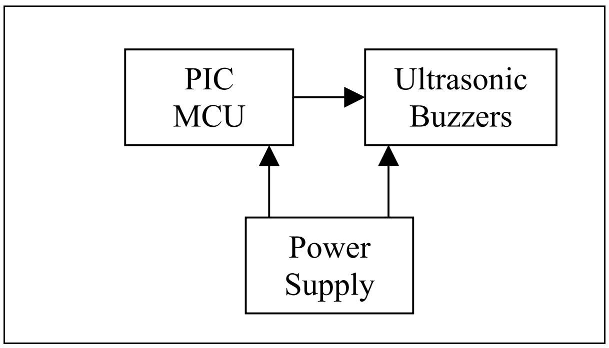

Figure 3-1: Block Diagram for the Transmitter Sub-System.

3.2 PIC Microcontroller

Requirements Satisfied: R[2], R[7], R[10], R[11], R[12], R[16]

The Microcontroller used is Microchip PIC16F684. This is a 14 pin PIC Flash microcontroller with a wide operating voltage of 2.0-5.5 VDC, on board EEPROM data memory, an Enhanced Capture/Compare/PWM (ECCP) [5]. In addition, this microcontroller features software selectable internal oscillator, ranging from 8 MHz to 32 kHz.

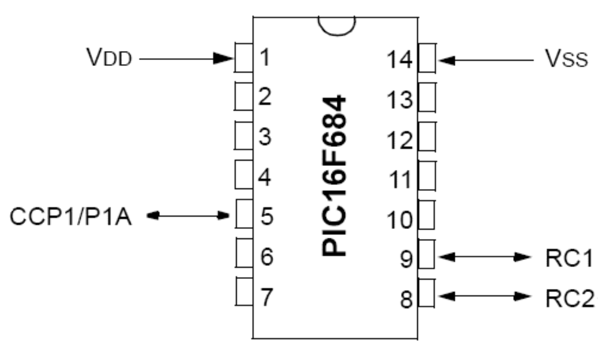

Figure 3-2: PIC16F684 PDIP pin diagram [5]

PIC16F684 features three timer modules, namely Timer0, Timer1 and Timer2. Timer1 and Timer2 provide the timing reference for required signals.

3.2.1 System Settings

The system clock, F_{osc}, is set to the high frequency internal oscillator at 8 MHz and interrupts are enabled. Two pins of Port C, namely *RC0* and *RC1* are set as digital out. In order to achieve the required effect for differentiable signal from planar and vertical buzzers, the microcontroller uses a table lookup algorithm to create two different Pulse Width Modulated (PWM) ultrasonic waveforms.

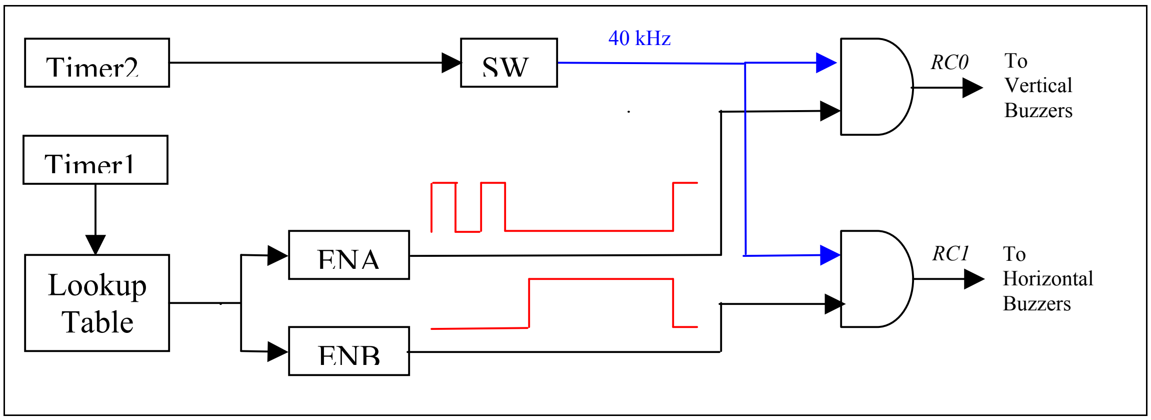

Figure 3-3: System Algorithm.

3.2.2 Ultrasonic Waveform

The Timer2 module, *TMR2*, of the microcontroller is used to create a 40 kHz square wave required by the ultrasonic buzzers. Timer2 module is an 8-bit timer with programmable prescaler and postscaler as well as an 8-bit period register, *PR2*. An interrupt flag bit is set on match between the *TMR2* and *PR2*. The clock input to the Timer2 module is the system instruction clock, $F_{osc}/4$.

3.2.3 Output Enable Signal

The Timer1 module, *TMR1*, of the microcontroller is used to create two enable signals for the ultrasonic waveforms. Timer1 module is a 16-bit timer with programmable internal or external clock source and a 3-bit prescaler. An interrupt flag bit is set on *TMR1* overflow. System instruction clock, $F_{osc}/4$, is chosen as the clock source for this timer as well.

Timer1 is set to create an interrupt every 10 ms. This interrupt is then used to create a software counter which counts from 0 to 7, creating 8 states. At each interrupt the software counter increments. When the counter reaches 7, the interrupt will cause the count to be reset to 0.

If the counter's count is 0 or 2, the ENA bit is set causing the vertical buzzers activation. If the counter's count is 4, 5, 6 or 7, the ENB bit is set causing the planar buzzers activation.

3.3 Ultrasonic Buzzers

Requirements Satisfied: R[1], R[3], R[5]

The ultrasonic buzzers used are the Jameco part number 139491 ultrasonic transmitter/receiver pair; the model number for this pair is 407R12B. This transmitter has a center frequency of 40 ± 1 kHz with bandwidth of 5 kHz [6].

3.4 Power Supply

Requirements Satisfied: R[12]

A lightweight portable power source is required for the transmitter. The power required for the ultrasonic buzzers and the microcontroller is provided with a 6V CR-P2 Lithium battery [7]. The PIC absolute maximum operating voltage is 6.5V. In order to prevent loading of the PIC micro controller, the circuit is design so that PIC sinks the current instead of supplying the current.

4 Receiver Design

4.1 Overview

Requirements satisfied: R[18], R[21],

The Receiver subsystem is divided into five components: the ultrasonic microphone, amplifier, band-pass filter, power supply and the envelope detector. The connection of these components is shown in Figure 4-1.

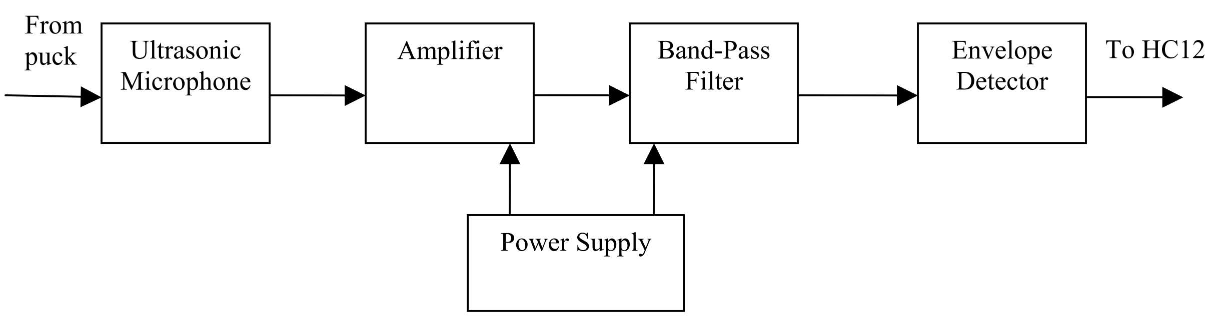

Figure 4-1: Receiver Block Diagram

4.2 Ultrasonic Microphone

Requirements satisfied: R[19], R[20], R[26]

The ultrasonic microphone used is from a Jameco part number 139491 ultrasonic transmitter/receiver pair. The model number for the pair is 40TR12B. This part has a 40.0±1.0 kHz center frequency with a bandwidth of 5.0 kHz [6].

4.3 Amplifier

Requirements satisfied: R[19], R[20], R[22], R[24], R[25], R[27], R[28]

The amplifier circuit is designed around a LM386 audio amplifier. The LM386 is operated from a single power supply, and provides a gain of 100 at 40 kHz in the circuit shown in Figure 4-2, derived from [8].

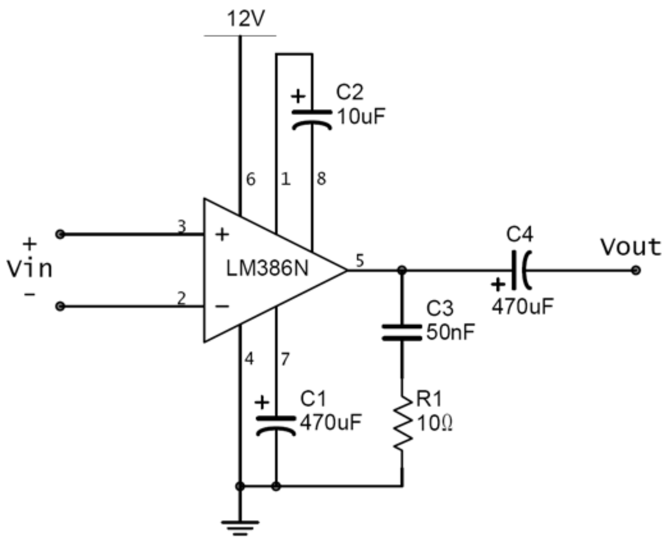

Figure 4-2: Amplifier Schematic

This circuit will be assembled with the microphone on a single circuit board which is connected to the band-pass filter with a length of wire. The wiring will send the signal to the filter, and also supply the +12V DC required to operate the amplifier.

4.4 Band-Pass Filter

Requirements satisfied: R[19], R[20], R[24], R[29]

The band-pass filter shown in Figure 4-3 has a centre frequency of 40 kHz and a bandwidth of 8 kHz. The filter is a second-order active filter designed around a LM324 op-amp using Analog Devices web-based filter designer [9].

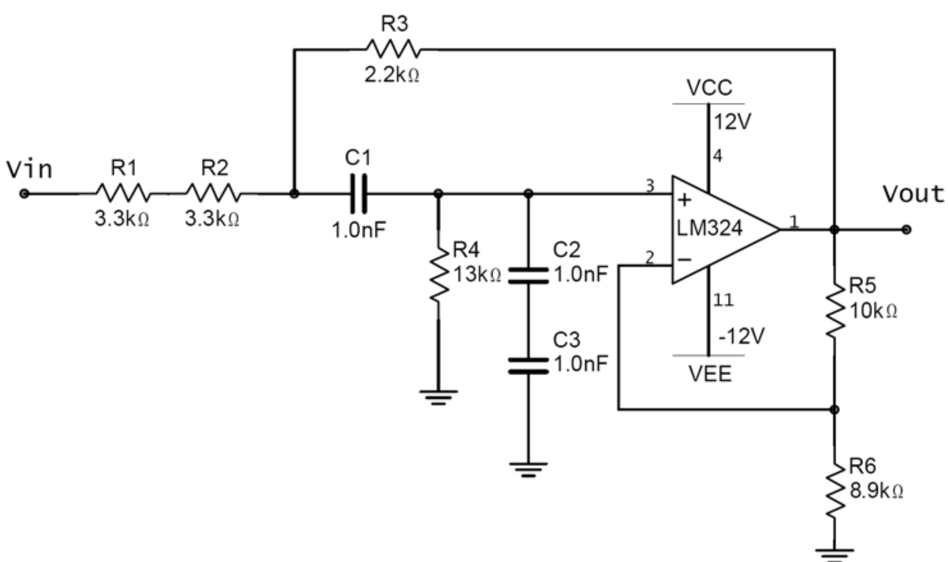

Figure 4-3: Band-Pass Filter Schematic

The filter requires dual power rails of ±12V DC which are supplied by the power supply. The overall gain of the filter is 8 at 40 kHz.

4.5 Envelope Detector

Requirements satisfied: R[30]

The envelope detector extracts the rectangular pulse originally transmitted by the puck from the received signal. An RC-circuit extracts the envelope and a Schmitt-trigger inverted converts the signal to a rectangular pulse. The circuit is shown in Figure 4-4.

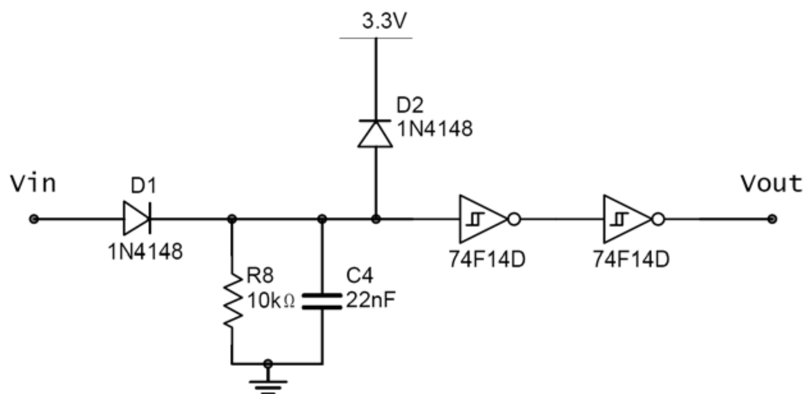

Figure 4-4: Envelope Detector Schematic

The signal voltage is clipped to prevent it from exceeding the 5V DC input of the inverter. The inverter output will be in the range 0-5V DC, precisely what it must be to be connected to the HC12 board.

4.6 Power Supply

Requirements satisfied: R[25]

A standard ATX power supply designed for a PC will supply the necessary +12V, -12V, +5V, -5V, and +3.3V DC supplies needed for the receiver circuitry. The power sense line (green wire) must be shorted to ground so that the power supply can be turned on and off by its rear power switch.

5 Software Design

5.1 Overview

The software system control consists of a PC and an interface board which connects the receiver to the PC. Signals received by the ultrasonic sensors are sent to and input on the interface board and are then communicated to the PC via the serial port.

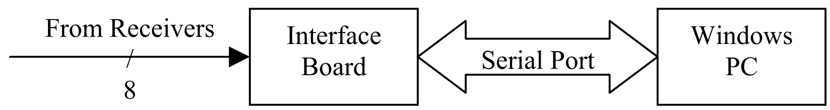

Figure 5-1: Interface Board/Windows PC Block Diagram

5.2 Interface Board

Requirements Satisfied: R[31], R[32], R[33], R[34]

A Motorola 68HC12 based microcontroller board ("HC12 board") is used to communicate the signals received by the ultrasonic receivers to the Windows PC. The particular board used is the M68EVB912B32 evaluation board supplied by Freescale Semiconductor. The HC12 board will be programmed with a calibration routine to initialize the board, an interrupt service routine (ISR) to do the data capture, and a data transfer routine to transmit the data to the PC, as depicted in Figure 5-2.

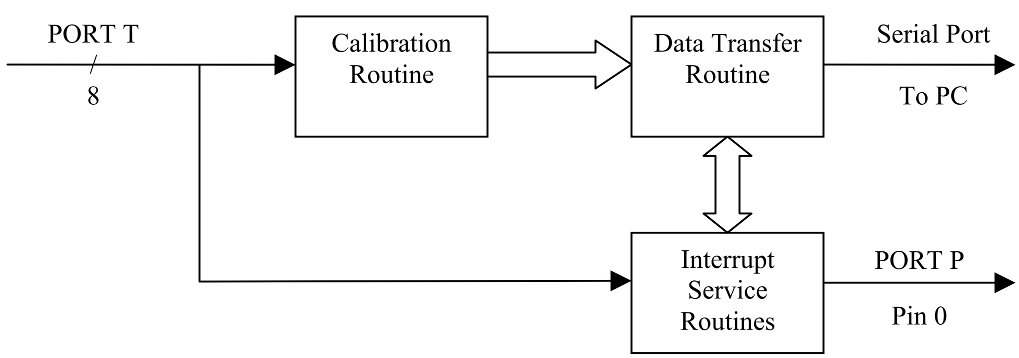

Figure 5-2: Interface Board Block Diagram

The eight receivers will connect to the eight pins of *PORT T*, which interfaces with the Input Capture function on the HC12 board [10]. Captured data is transferred to the PC via the serial port built in to the HC12 board.

5.2.1 Calibration Routine

The calibration routine initializes the Timer/Counter function to configure *PORT T* as an input port, capturing on both rising and falling edges of the input signal. The serial port is also configured for communication with the PC.

To calibrate the Timer/Counter, the routine listens for pulses on pin 0 of *PORT T*, and synchronizes the internal timer to its period. Once the timer is synchronized, the HC12 board sends a message to the PC via the serial port indicating that calibration is complete, and waits for a response before entering the main routine. The internal timer will also toggle *PORT P* pin 0 so that the period of the timer can be verified with an oscilloscope.

5.2.2 Interrupt Service Routines

Eight identical ISRs are created, one for each input pin, to capture the time that a pulse is received and measure its length. When a pulse edge is captured, the ISR stores the time of reception relative to the internal timer in main memory. A ninth ISR maintains the internal timer. Leading edge times are stored separately from trailing edge times.

5.2.3 Data Transfer Routine

Each time captured occupies 16 bits of memory. Two times are stored for each channel, one for the leading edge and one for the trailing edge. Thus a complete set of times requires (2 times)*(8 inputs)*(16 bits) = 256 bits, or 32 bytes. Data will be transferred one byte at a time in the order shown in Table 5-1.

Table 5-1: Data Transfer Format

Byte	Contents		Byte	Contents
0	Header byte ($FF)		17	Header byte ($FF)
1	Pin 0 leading edge time, low byte		18	Pin 0 trailing edge time, low byte
2	Pin 0 leading edge time, high byte		19	Pin 0 trailing edge time, high byte
3	Pin 1 leading edge time, low byte		20	Pin 1 trailing edge time, low byte
4	Pin 1 leading edge time, high byte		21	Pin 1 trailing edge time, high byte
5	Pin 2 leading edge time, low byte		22	Pin 2 trailing edge time, low byte
6	Pin 2 leading edge time, high byte		23	Pin 2 trailing edge time, high byte
7	Pin 3 leading edge time, low byte		24	Pin 3 trailing edge time, low byte
8	Pin 3 leading edge time, high byte		25	Pin 3 trailing edge time, high byte
9	Pin 4 leading edge time, low byte		26	Pin 4 trailing edge time, low byte
10	Pin 4 leading edge time, high byte		27	Pin 4 trailing edge time, high byte
11	Pin 5 leading edge time, low byte		28	Pin 5 trailing edge time, low byte
12	Pin 5 leading edge time, high byte		29	Pin 5 trailing edge time, high byte
13	Pin 6 leading edge time, low byte		30	Pin 6 trailing edge time, low byte
14	Pin 6 leading edge time, high byte		31	Pin 6 trailing edge time, high byte
15	Pin 7 leading edge time, low byte		32	Pin 7 trailing edge time, low byte
16	Pin 7 leading edge time, high byte		33	Pin 7 trailing edge time, high byte

This sequence repeats infinitely until the HC12 board is reset or powered off.

5.3 Windows Software

Requirements Satisfied: R[35], R[36], R[37], R[38], R[39], R[40], R[41], R[42], R[43], R[44], R[45]

In the Windows software portion of the Ricochet Tracking System, there are two main parts, including communication and graphical user interface (GUI). The first portion is where data is taken from the M68HC12 and from it the position of the object is calculated. Through the GUI portion, the result of the previous portion is interrogated and applied to produce a graphical representation of the object under scrutiny and its general environment. The software also requires the coordinates of the sensors to be stored in a text document in the main file. The format of the text document is very basic and consists of a listing of the sensors coordinates separated by tabs with each sensor separated by a return. For example if there were two sensors, one located at (1,2,3) and the other located at (4,5,6) then the text document would read

```
1	2	3
4	5	6
```

with an additional return at the end.

5.3.1 Communication

The communication side consists of three main states, which include idle, data acquisition and computation. Each state has its own algorithm to follow in order to complete its task. Seen below in Figure 5-3 is a state diagram of each of these states.

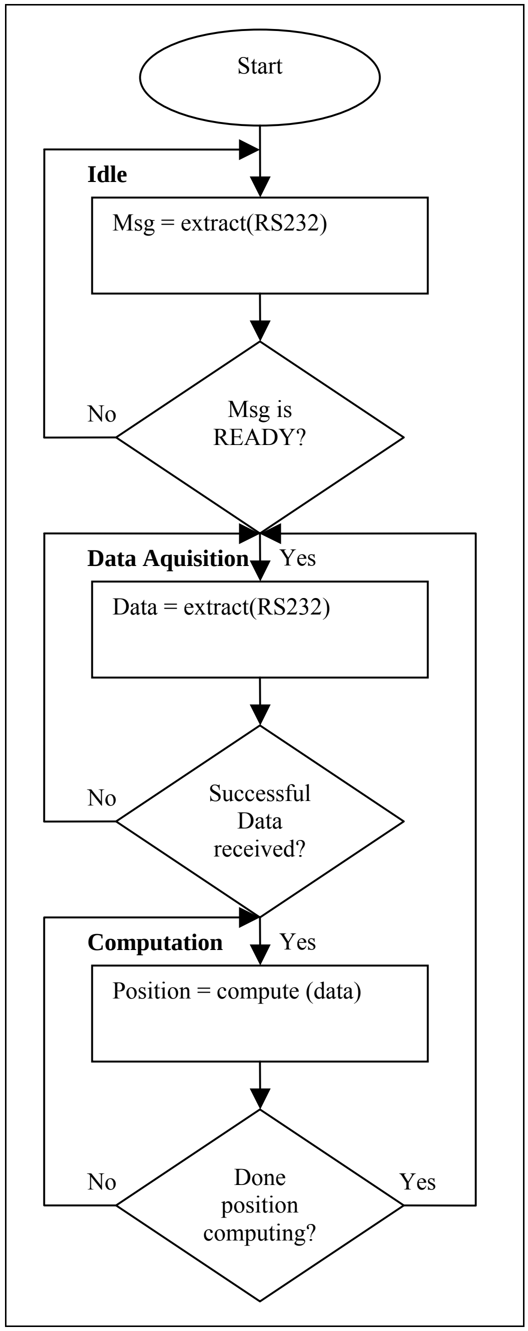

Figure 5-3: State diagram for retrieving data and computing the position

5.3.1.1 Idle

Throughout the idle state, the HC12 is being calibrated or waiting to be calibrated. During this time, the program is waiting for a signal from the HC12 received through the RS232 port signaling the calibration has been complete. This serial message is in the format of

$,READY,

where the "$" represents the beginning of a message and "READY" signals that the HC12 is calibrated and is now going to send data. At this time, the software continues to the next state of data acquisition.

5.3.1.2 Data Acquisition

During the data acquisition state, the computer is continuously listening to the RS232 port for a successful message. A successful message means that the message received is in the right format, which is

<header> ,<Δt_1>,<Δt_2>,…,<Δt_8>,

where the <header> is signaling that this is a data message and the <Δt_1> through <Δt_8> are the measurements the HC12 has taken, as shown in Figure 5-4.

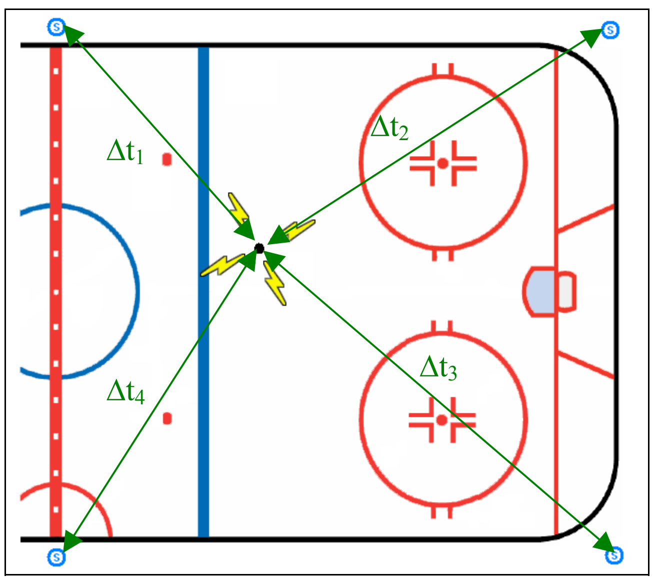

Figure 5-4: Picture showing the types of measurements being taken and sent

This data is parsed and sent through to the computation phase.

5.3.1.3 Computation

The computation phase is where the software analyzes the data from the previous part and extrapolates the position of the object. To successfully compute a three degree position calculation, the data <Δt_1> through <Δt_8> must be applied, and at least 3 data points are needed. Each data point is multiplied by the speed of sound to achieve a distance measurement as

$$d_i = 340.29\Delta t_i \, .$$

The coordinates of each sensor (x_i, y_i and z_i) are known and the puck's position, described by x_p, y_p and z_p, is the needed result. So, through triangulation, the position of the object is discovered, and the formulas which this state will be solving are

$$d_i + \varepsilon = \sqrt{(x_i - x_p)^2 + (y_i - y_p)^2 + (z_i - z_p)^2}$$

where ε represents an error in the distance measurement, which is minimized through the calibration process. However, this error is constant throughout each equation; therefore, even if the error is large, with a fourth measurement, this is just merely another variable to be solved for.

5.3.2 Graphical User Interface

The graphical user interface (GUI) consists mainly of OpenGL coded near real-time recreation of the environment of where the object is moving and the object itself. In this interface, the user is able to manipulate various sensors and watch the object move around the environment on screen. Various data is also available through this setup, such as the object's velocity and acceleration.

6 Test Plan

6.1 Overview

The test plan for the Ricochet Tracking System is comprised of four sets of unit tests for the separable components of the system, and an integration test to be completed once the unit tests have been passed.

6.2 Puck/Transmitter Unit Tests

6.2.1 Power Source

The battery must be able to provide the required current for the operation of the ultrasonic buzzers. Base on the Lab tests conducted on the buzzers the required current for each ultrasonic buzzer does not exceed 3mA. This current requirement is increased by 60% to 5mA for each ultrasonic buzzer resulting in total required current of 30mA.

6.2.2 Ultrasonic Buzzers

The Ultrasonic buzzers must be tested to provide a 40 kHz signal. The function generator should be configured to output a 40 kHz square wave with 5V peak-to-peak amplitude and zero offset. This signal may vary within 5% of the required 40 kHz. This standard is set based on the receiver's bandwidth.

6.2.3 PIC Microcontroller

The output of the Microcontroller must be tested. The PIC output must be a 40 ± 4 kHz square wave pulses. The RC0 output must be two 10 ms pulses with 10 ms time difference between the two pulses and 50 ms pause after the second pulse. The RC1 output must be a 40 ms pause followed by a 40ms pulse. The detail output of RC0 and RC1 is demonstrated in Figure 3-3.

6.3 Receiver Unit Tests

6.3.1 Power Supply

The outputs of the ATX power supply must be tested. Each output must be within 5% of the stated value, as listed in Table 6-1. Note that the power sense line (green wire) must be shorted to ground (black wire) on the large 20 pin connector for the power supply to turn operate.

Table 6-1: Power Supply Voltages [11]

Wire	Voltage (DC)
Orange	+3.3V
Red	+5.0V
White	-5.0V
Yellow	+12.0V
Blue	-12.0V

6.3.2 Ultrasonic Microphone

The sensitivity of the microphone is tested by connecting the receiver to an oscilloscope and a transmitter to a function generator. The function generator should be configured to output a 40 kHz square wave with 5V peak-to-peak amplitude and zero offset.

The microphone should receive a 40 kHz sine wave, with an amplitude proportional to the distance between the transmitter and receiver. Reception can be tested at varying distances, and angles between the transmitter and receiver.

6.3.3 Instrumentation Amplifier

Using a function generator and oscilloscope, verify that with an input of a 40 kHz sine wave with a 10mV peak-to-peak amplitude to the amplifier, the output has the same frequency, but greater amplitude. The amplitude at pin 6 of the AD623 must be 100mV. At pin 5 of the LM386 the amplitude must be 2.0V.

6.3.4 Band-Pass Filter

Using a function generator and oscilloscope, verify that with an input of a 40 kHz sine wave with a 1V peak-to-peak amplitude to the amplifier, the output has the same frequency, but an amplitude of 8V. Verify that the peak-to-peak amplitude is 5.5V at the upper and lower cutoff frequencies of 43 kHz and 35 kHz.

6.3.5 Envelope Detector

Using two function generators, generate a 40 kHz sine wave modulated with a 1 kHz square wave. Use a 5V peak-to-peak amplitude and 2.5V offset for the square wave and an 8V peak-to-peak amplitude and zero offset for the sine wave. Connect this signal to the input of the envelope detector, and verify with an oscilloscope that the output is a 1 kHz square wave swinging between 0 and 5V.

6.4 Interface Board Unit Tests

6.4.1 Calibration Test

Connect all eight inputs to ground. Connect the serial port to a PC and monitor its output with a terminal program. Power up the interface board and verify the initialization message in the terminal program. Use an oscilloscope to verify that there is a 6.25 Hz square wave present on Port P, pin 0.

Next apply a 12.5 Hz square wave to input 1 (Port T, pin 0). Verify that the square wave on Port P, pin 0 is still 6.25 Hz, but its phase is now the same as the input being applied. Verify the calibration message in the terminal program.

6.4.2 Data Capture Test

After completing the Calibration Test, press enter in the terminal to enter data capture mode. Verify in the terminal that the correct data format is being used, as described in Section 5.2.3. Also verify that when an input pin is grounded, the corresponding data bytes seen in the terminal are zero. When an input has a 12.5 Hz square wave applied, the corresponding data bytes seen in the terminal are non-zero.

6.5 Software Unit Tests

6.5.1 Startup Test

Without the interface board connected, verify that the software recognizes that the calibration signal has not been received. Using a second PC connected via the serial port, send the calibration signal and verify that the software receives the calibration signal and attempts to track the puck.

6.5.2 Tracking Test

Using a second PC and a set of "dummy" data, observe the puck being tracked. Verify that the position, speed and acceleration are calculated and shown.

6.5.3 Recording and Playback Test

Again using the dummy data, record the motion of the puck, then use the playback feature to review the recording.

6.6 Integration Tests

6.6.1 Setup Test

In a rectangular area, at least $3m^2$ in area, setup the RTS. Define the center of the room, and position the sensors around the room. Measure the locations of the sensors, and enter the data into the software. Power up and calibrate the system and verify that tracking begins.

6.6.2 Positioning Test

Move the puck around the room to various fixed locations. Measure each location, and verify the accuracy of the position shown by the software. Slide the puck along the floor at low speeds to verify it can be tracked in motion.

6.6.3 Passing Test

Two or more people with hockey sticks are required for this test. Pass the puck back and forth to and from different locations, and at different speeds. Observe and check the accuracy of the results. Test the recording and playback features. Verify that there is no accumulating error.

6.6.4 Slapshot Test

Use a soft target to avoid damaging the puck. Take increasingly harder slapshots at the target and monitor the results on the PC.

6.6.5 Goal Judge Test

Mark a "goal line" on the floor. Verify that the software can determine if the puck is in front of the line, behind the line, and on the line.

6.6.6 Long Term Test

Verify that with a fresh battery, the puck can be tracked continuously for at least 20 minutes.

7 Sources and References

[1] Tehrani-Nejad, Ashkan, et al, Functional Specification for a Hockey Puck Tracking System, 2007.

[2] Weinstein, Eric W, "Dipyramid" [Online document], 2002 Aug 29, [cited 2007 Feb 19], Available HTTP: http://mathworld.wolfram.com/Dipyramid.html

[3] Tehrani-Nejad, Ashkan, et al, Proposal for a Hockey Puck Tracking System, 2007.

[4] National Hockey League, Official Rules 2006-2007, Toronto: North 49 Books, 2006. Available HTTP: http://cdn.nhl.com/rules/20062007rulebook.pdf

[5] Microchip Technology Inc., PIC16F684 Data Sheet, 2006. Available HTTP: http://ww1.microchip.com/downloads/en/DeviceDoc/41202E.pdf

[6] Jameco Electronics, Jameco part number 139491, 2006. Available HTTP: http://www.jameco.com/Jameco/Products/ProdDS/139491.PDF

[7] EEMB Batteries, CR-P2 Brief Datasheet, 2007. Available HTTP: http://www.eemb.com/PDF/CR/CR-P2%20Brief.pdf

[8] National Semiconductor, LM386 Low Voltage Audio Power Amplifier, 2000. Available HTTP: http://cache.national.com/ds/LM/LM386.pdf

[9] Analog Devices, Inc., "Active Filter Synthesis" [Online document], 2007, [cited 2007 Feb 20], Available HTTP: http://www.analog.com/Analog_Root/static/techSupport/designTools/interactiveTools/filter/filter.html

[10] Motorola, Inc., M68HC912B32 Technical Summary, 1998. Document number MC68HC91232TS/D.

[11] Intel Corporation, ATX Specification - Version 2.01, 1997. Available HTTP: ftp://download.intel.com/design/motherbd/atx_201.pdf