FIVEPOINT technologies

November 8, 2005

Dr Andrew Rawicz
School of Engineering Science
Simon Fraser University
Burnaby, British Columbia

Re: ENSC 340 Functional Specifications

Dear Dr. Rawicz,

The following document presents the design specifications for the Voiceture System as part of our project for ENSC 340. Our goal is to create a device that will translate sign language into speech using a commercially available glove to recognize hand gestures.

The system has already begun the development phase with sensor research and design, motion capture software programming, and algorithm development. The system will feature a hand mountable data input glove with mounted sensors interfaced through a java enabled PC featuring a graphical user interface.

This document details the sensor, electronic, and software design necessary to complete our prototype of the Voiceture for proof of core technology demonstration in December. Lastly, a proposed test plan to verify the project is presented.

Fivepoint Technologies is comprised of four motivated Engineering students from Simon Fraser University. These are Ganesh Swami (President and CEO), David Brayden (Vice President, Research and Development), Phoenix Yuan (Chief Operating Officer), and Kjell Eggen (Chief Technical Officer). Questions and concerns can be directed to 604-992-1404 or e-mail to ensc340-Fivepoint@sfu.ca.

Sincerely,

Ganesh Swami

Ganesh Swami
President and CEO
Fivepoint Technologies
Vancouver, BC

Enclosure: Design Specifications for the Voiceture System

Design Specifications for the Voiceture system

Translation of American Sign Language into Speech

November 8, 2005

Project Team:

David Brayden
Kjell Eggen
Phoenix Yuan
Ganesh Swami

Submitted to:

Andrew Rawicz
Steve Whitmore
Mike Sjoerdsma
Brad Oldham

Simon Fraser University
School of Engineering Science
ENSC 305/340

Executive Summary

Being deaf or mute in today's society is not easy. Something as normally trivial as ordering food in a restaurant can be a slow and frustrating task. In North America, few people know how to communicate properly with a deaf person, and even fewer understand American Sign Language (ASL), even though the number of people using ASL as their primary language in the United States is estimated to be up to 2 million.

The Voiceture system aims to fill this linguistic gap in basic human communication by being the first economical and reliable technological device to translate signs and gestures into speech. The first stage of Voiceture will be the development of the sensing and analysis system consisting of core technology features:

1. data input glove to sense hand orientation and positions
2. standard computer interfacing for data acquisition
3. software with the capability to sample and monitor hand positions
4. algorithms to analyze single character gestures
5. audible and visual translation outputs

Following this, the second stage of development would focus on the following requirements:

1. full-fledged word and grammar translation
2. device portability (user mobility)
3. quality and reliability

This document outlines the design for the Voiceture's hardware, hardware-software interface, data analysis and recognition algorithms, and graphic user interface.

The first phase of development is already underway, with a scheduled deadline to be completed and demonstrated by December 14, 2005. This first phase will be demonstrated and delivered as our ENSC 340 project.

Table of Contents

List of Figures

List of Tables

1. Introduction

Communicating with English speaking people has never been easy for a deaf or mute person as to date no economical and robust system exists outside of the traditional translator. This device will bridge that gap, proving to be a milestone towards independent real-time communication for the deaf and hard of hearing.

The Voiceture is a data input system to translate American Sign Language (ASL) signs and gestures into speech. The device consists of two components, the data input glove and the gesture capturing and translation software.

1.1. Scope

The purpose of this specification is to describe the Voiceture design requirements listed in the functional specification [3].

1.2. Acronyms

API	Application Programming Interface
ASL	American Sign Language
BOM	Bill of Materials
DAQ	Data Acquisition
FAQ	Frequently Asked Questions
GUI	Graphical User Interface
PC	Personal Computer
PCB	Printed Circtuit Board
USB	Universal Serial Bus
SDK	Software Development Kit

1.3. Objectives

These design requirements are intended to be used by the engineering team to develop the prototype Voiceture system. The design proposed herein will be delivered for December 2005. These specifications are based on the premise that the prototype will be a demonstration of the core innovative technology only and not representative of a fully packaged mobile final product for an end user.

2. System overview

The Voiceture system is composed of the motion capture device consisting of sensors and the corresponding electronics and a PC device with audio-visual multimedia capabilities. The motion capture device outputs hand and finger positional data that is sampled by the computer. The computer is then responsible for all data analysis and to give output to the user through the GUI. The system information flow is shown below.

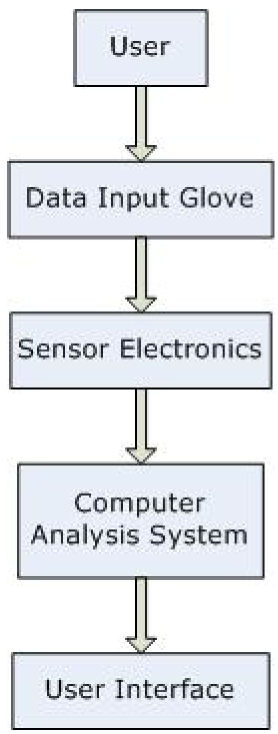


Figure 1: Voiceture System Overview

3. Hardware Overview

The diagram below shows the major component organization for the Voiceture system hardware.

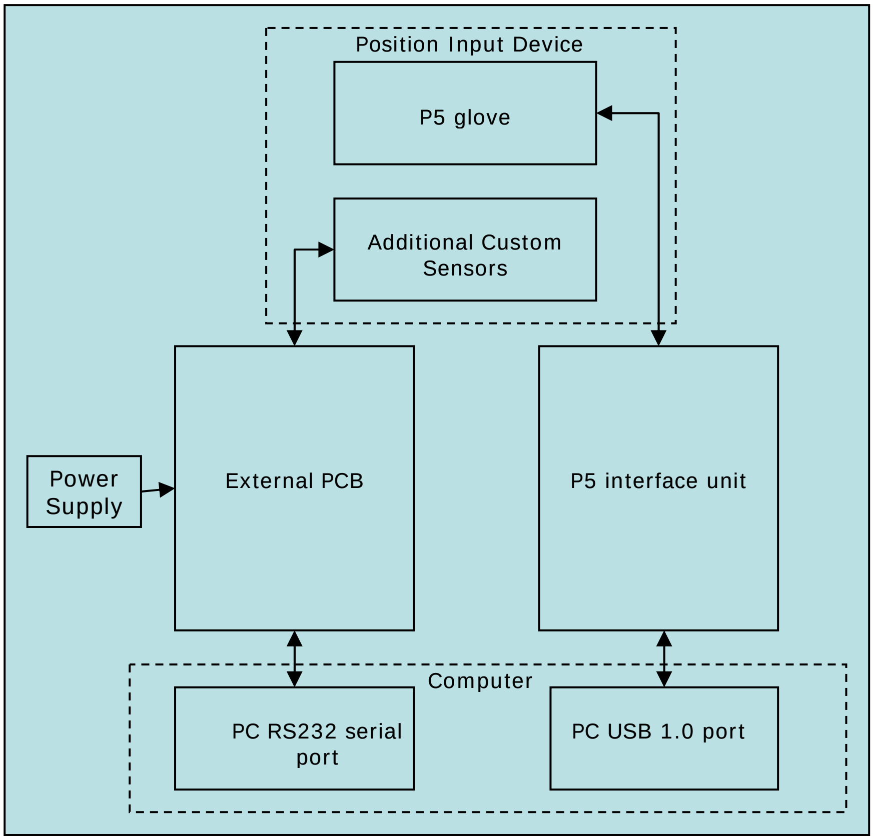


Figure 2: Hardware Overview

3.1. P5 glove

The P5 reality glove is a motion and position capture device by Essential Reality Inc. created for the consumer market. This device's functionality meets the sensor requirements stated in the functional specification [3]. The P5 Reality Glove has one bend sensor that attaches to each finger, giving finger bend positioning data accurate to 0.5 degrees. It also incorporates X, Y, Z translational position tracking sensors with yaw, pitch, and roll. These sensors are optical based and will not be used in the first prototype.

Figure 3: P5 Reality Glove

The interface for the glove is supplied via an external unit which doubles as the infrared tracking sensor housing and USB interface to a PC.

The glove comes with a standard driver for Microsoft Windows that is fully functional, and includes a free SDK for custom driver programming. The P5 Reality glove meets all functional requirements listed in *Section 3: Sensing Glove Requirements* of the Functional Specification [3].

While the glove outputs sensor bend data, it has trouble distinguishing between a sensor that is being stretched due to finger motion or bent due to finger curling. This produces a problem with the recognition of similar closed fist letters. This problem leads to the need for special software additions and the additional sensors described in Section 3.2.

3.2. Custom Sensors

As reasoned above, two additional sensors need to be added to the P5 glove. Metal contacts will be placed on both surfaces to model a switch. Since a clean signal is required by the PIC input (see Section 3.3), each switch will be debounced using an RC circuit (see Section 3.3). This will make the initial electronic development easier since physical contact will not need to be constantly maintained but will yield the desired electronic response.

The first switch operates between the index and middle fingers at around the last knuckle. The contacts are attached to the plastic rings, which connect the bend sensors to the fingers on the P5 glove. The second switch is slightly more complicated as it is installed on the tip of the fingers. An elastic holds the sensor in place on the end of the middle finger, while the metal contact on the thumb is attached directly to the bend sensor.

The first switch is used to:

- Separate the letter U from H, V, and K
- Separate the letter T from S.

The second switch is used to:

- Separate E from the closed fist letters,
- Distinguish D from P, Q, G, L, and possibly Z.
- Determine O easily and accurately

3.3. Additional Sensor PCB

This PCB is to form the interface for the custom sensors for the PC. A block diagram outlining the major components of the Sensor PCB is shown below. The detailed schematic is shown in Section 3.3.5.

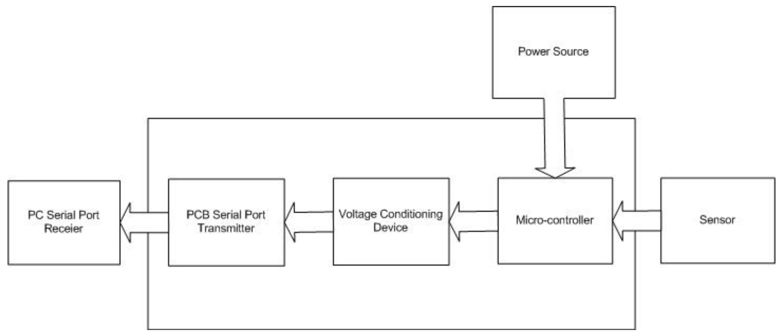


Figure 4: Block Diagram of the Sensor PCB

3.3.1. Power Supply

The input power required is 9V DC. Since a battery-powered device requires periodic maintenance, our circuit will make use of a standard power adapter that can be connected to any AC power outlet. However, a 9V DC battery will be interchangeable. The input circuit voltage is further conditioned by a voltage regulator that brings the voltage down to 5V DC.

3.3.2. Microcontroller

The Microchip Inc. manufactured PIC series micro-controllers are the most suitable device as the "brain" of the PCB. We chose the PIC16LF876 micro-controller for its attractive price, wide range of peripherals, and onboard flash memory. It is also one of the newer PIC designs that have low operating power due to inclusion of flash-based memory. A pin layout diagram of the PIC16LF876 is shown in the following diagram.

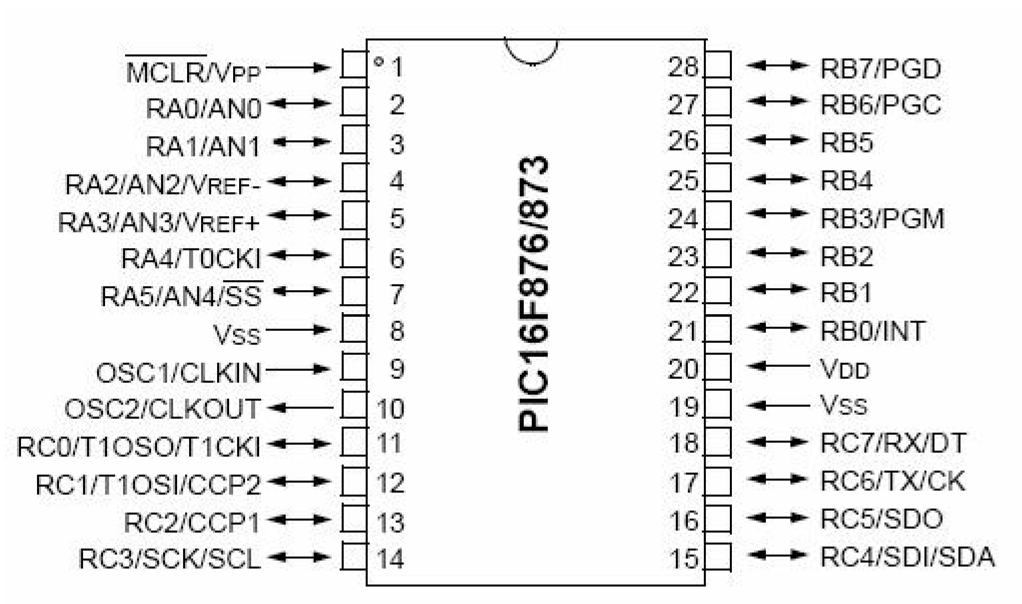


Figure 5: PIC Connection Diagram

Since this micro-controller does not have an on-board oscillator, an external 20Mhz crystal is connected to its oscillation input pins (Section 3.3.5).

As shown in the diagram, the micro-controller has three sets of I/O pins. Port A and Port B will both be used for sensor inputs. The PIC supports up to a maximum of 14 sensors.

The on-board UART peripheral that resides on Port C greatly simplifies the implementation of serial port transmission protocol. The UART system is responsible for handling transmission baud rate and start/stop bits. However, the built-in UART does not handle parity error correction natively, thus the transmission will contain pure data only.

The PIC input will be sampled based on edge-triggered interrupts. Every time an interrupt is detected, the signal will be sent to the PC in the next UART transmission. Thus, the input signal must be debounced so that it will not generate multiple false triggers.

3.3.3. Serial Port Voltage Conditioning Device

PIC micro-controller normally outputs a +5V DC signal as digital high and a 0V DC signal as digital low. However, PC serial port specification requires a high voltage of +12V DC and a low voltage of -12V DC. Hence, the PIC output signal must be conditioned in order to be suitable for the serial port. Fortunately, the MAX232 serial port interface chip is designed to handle the voltage conversion.

3.3.4. Serial Port

Standard serial port communication requires only 3 pins: voltage high, voltage low, and ground. However, the PC serial port specification has added a device handshake requirement. Since we cannot actually perform a proper handshake with the PC, we will permanently short the serial port handshake pins in order to fake a successful two-way handshake. Our custom serial port is shown in the diagram below.

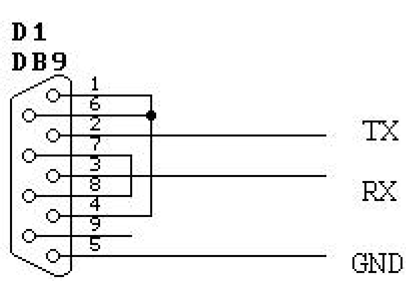


Figure 6: Serial Port

3.3.5. Overall Schematics

The Bill of Materials (BOM) and schematic for the Sensor PCB is shown below

Quantity	References	Value	Part Number	Description
9	R1-R9	1K	General stock	Resistors

2	C1, C2	1u	General stock	Electrolytic Capacitor
2	C5, C6	10u	General stock	Electrolytic Capacitor
1	C7	47u	General stock	Electrolytic Capacitor
2	C8, C9	22u	General stock	Electrolytic Capacitor
2	C3, C4	15pF	General stock	Ceramic Capacitor
1	U1	PIC16F876	05171PM	PIC16X Micro-controller
1	U2	MAX232	LT1181CN	Serial Port Converter
1	U3	LM7805	JRCM7008C	Voltage Regulator
1		20Mhz	Raltron 20.000	20Mhz oscillator
1	J1	DB9	DB15M HD	Male Serial Port Connector

Table 1: Sensor PCB BOM

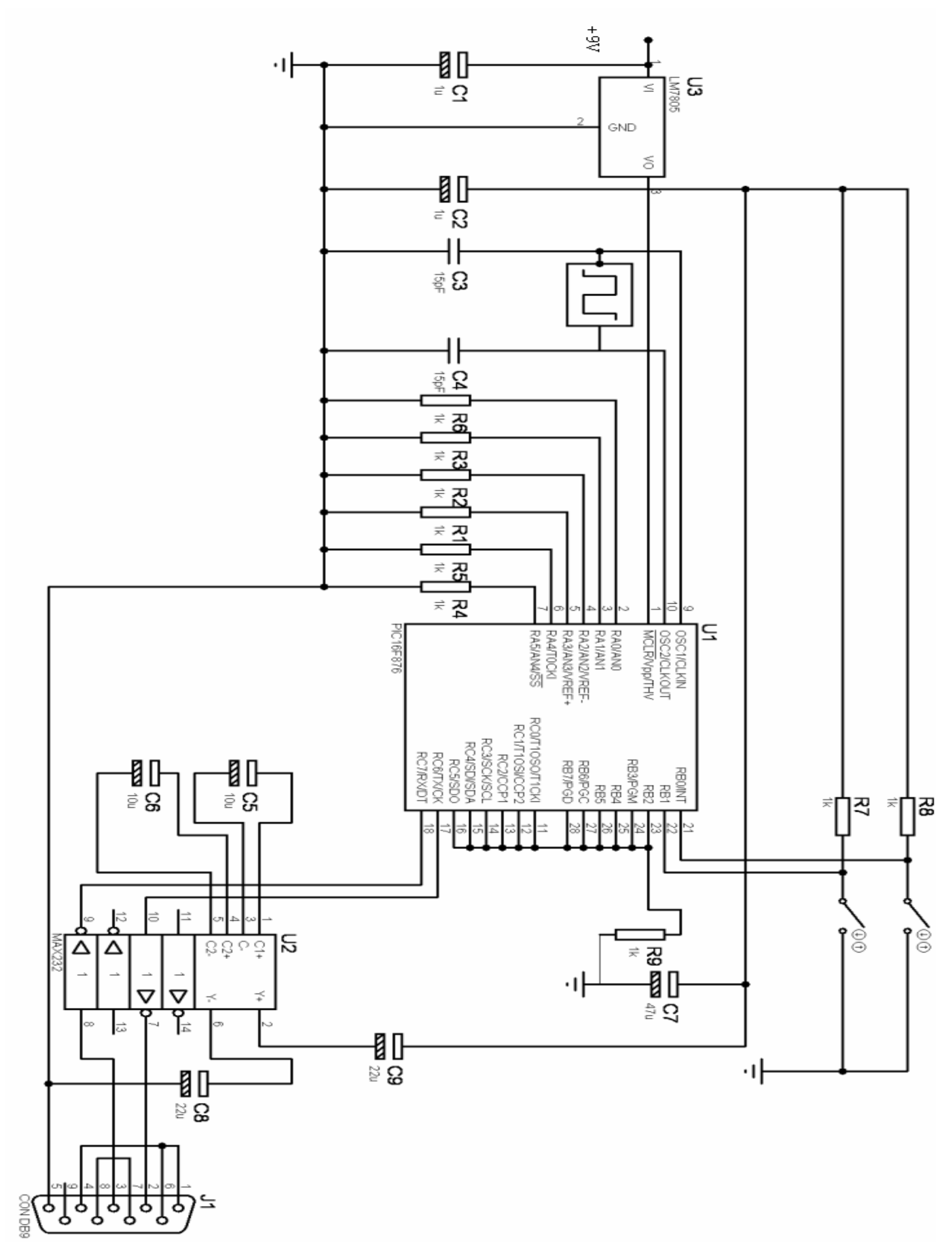


Figure 7: Sensor PCB Schematics

3.4. Computer system

To meet the functional requirements of the computer system, a standard commodity computer will be employed for the first prototype. Commodity Computers are far easier to develop for and work with than a custom built microcontroller, thereby reducing the time to a working prototype. This will also serve the long term purpose of having the ability to design on a well established consumer platform. Also, the software is implemented using Java which is a cross-platform development standard for computers. Therefore, no foreseeable issue will occur in porting the software to another type of computer (like a handheld PC).

4. Hardware-software interface

4.1. P5 glove

Since the P5 Glove's primary target audience is gaming industry developers, the official API is available to the public free of charge. The official driver interface, P5_C_Dll, is written in C. A Java P5 Glove API wrapper, P5_Java_Wrapper, is used to convert the API function call from C to Java. Thus, the P5_Java_Wrapper brings the glove signal into our Java software development environment. Further more, the P5Driver_Wrapper is created to simplify the interaction between the Glove API and the actual application. It takes care of glove initialization and automates updating of glove data. Ultimately, the interaction between the application and the driver is reduced down to one function call defined in P5Driver_Wrapper: p5FreadFingers()

Figure 8 shows the P5 glove interface layer.

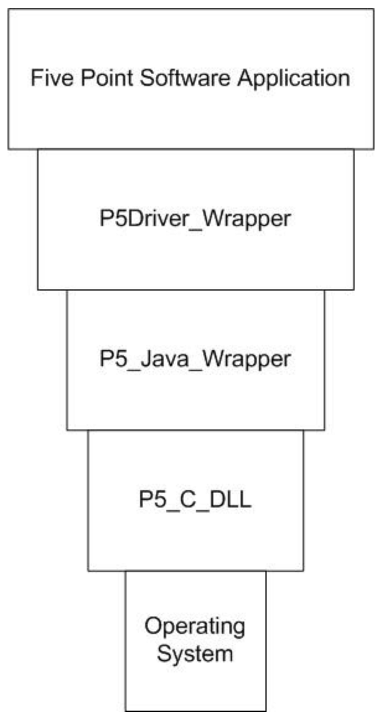


Figure 8: P5 Software Interface

4.2. Sensor PCB interface

` The UART signal transmitted by the Sensor PCB will be received by the PC serial port. The Java Communication API will be responsible for detecting the arrival of this signal, and trigger an internal interrupt to deal with the newly captured data. A Serial Port Communication Wrapper will interface with the Java Communication API to hide the internal complexity from the main software application. readSerialPort() is the method that will be used by the main software application to maintain an up-to-date status of the sensors. The hierarchy for this architecture is shown below.

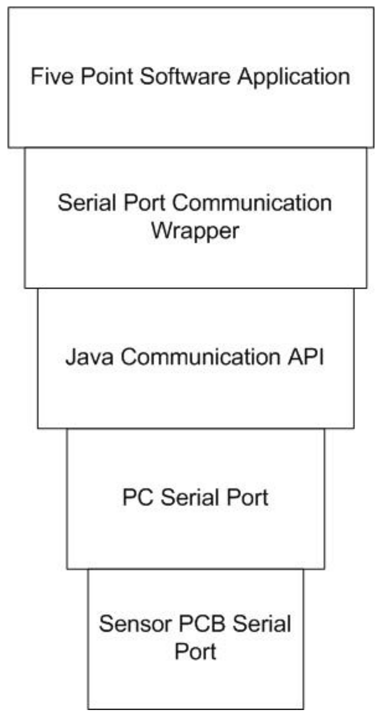


Figure 9: Serial Port Communication Software Interface

5. Data Analysis algorithms

The basis of the data analysis stems from a user calibration profile that each user must create in order to use the Voiceture. As part of this profile, the user signs each letter, from which a data template is created. The data entered into this template will be taken as the correct data for each letter. Many problems arise when attempting to match data with a template, especially from a constant stream. These issues include:

- User's inability to exactly match same hand and finger positioning for each letter
- The data may pass through trained points on the way to other letters
- Letters are hard to distinguish due to similarity

To best distinguish the separation of letters we will employ a combination of Gaussian statistics and counters shown in Figure 10.

Each time the data is within the radius of a letter, a Gaussian factor representing how close the letter is to its trained value is calculated and added to the corresponding letters bin. At the end of each cycle, each bin is decremented. This way the system will only output a letter if the data is within its radius for a long time and invalid intermediate data is filtered out. The time taken to recognize each letter will vary with the distance to the center of the letter. Before each letter is output the main program checks to see if it was repeated. Also during each cycle there are special case calculations (not shown in Figure 10).

This algorithm has several advantages. Firstly, it enables us to keep track of past data values without explicitly storing them. Tricky tasks like remembering how long the data has been within a letter space, or dealing with the data switching between letter spaces is reduced to simple calculations of current data and comparison of bin values. Also, using Gaussian statistics we don't have to worry about the data staying within a well defined letter space for a predefined amount of time. With this algorithm, instead of having an exact cutoff line to determine whether the data is within a letter space, we have a sloping edge.

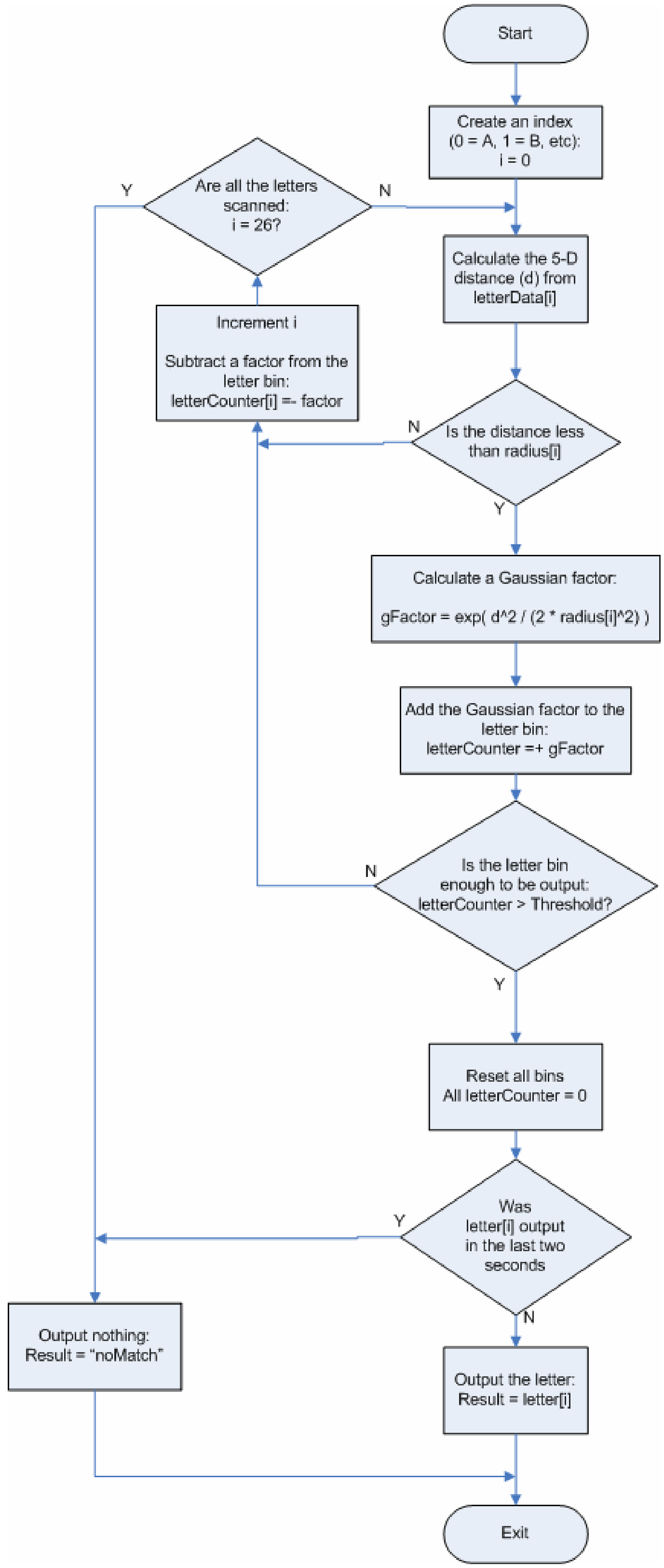


Figure 10: ASL Letter Selection Algorithm

Also incorporated is an absolute bend checking algorithm. This will scan the Letter Selection Algorithm for values of ± 5% of the absolute bend of all fingers. Varying the absolute bend will partly account for the glove position on the hand changing, caused by either the glove moving during operation, or being removed and put back on.

These algorithms will fit into the overall program as follows:

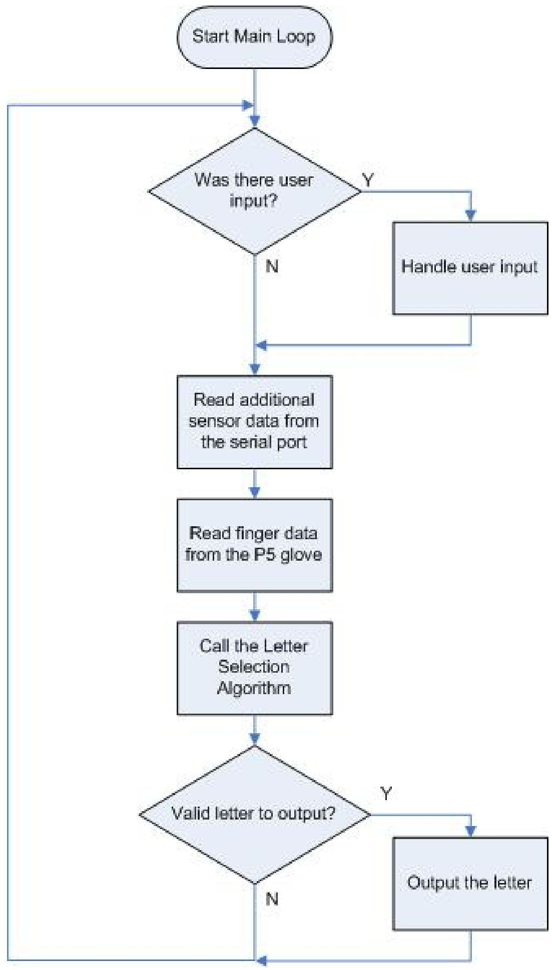


Figure 11: Main Loop Algorithm

6. User Interface software

Fivepoint's Voiceture shows the login screen when first started up (as shown in Figure 12). If Voiceture is being run for the first time, the list of existing profiles is empty, and therefore a new profile must be created. Profiles will be stored as USERNAME.txt with content being the calibration data.

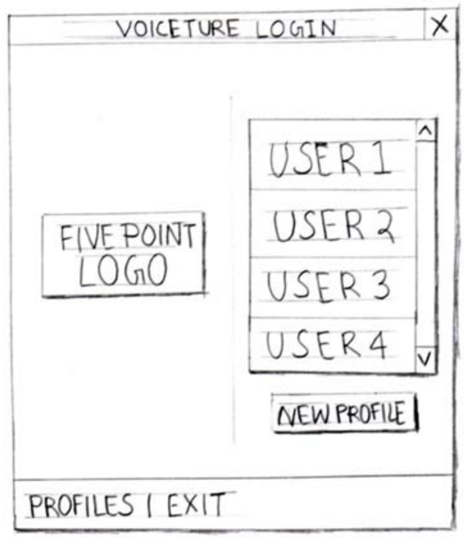


Figure 12: Voiceture Login Screen

When the "New Profile" command is chosen from the login screen, Voiceture prompts the user for a profile name. After this dialog, the user is shown the training screen (Figure 13) which runs through all the 26 characters individually. When finished the user is returned to the main login screen.

Figure 13: Voiceture Training Screen

In a normal run, the Voiceture login screen shows a list of existing profiles, sorted by most recently used profile. From the login screen, the user can also access dialogs to delete and rename profiles (Figure 14).

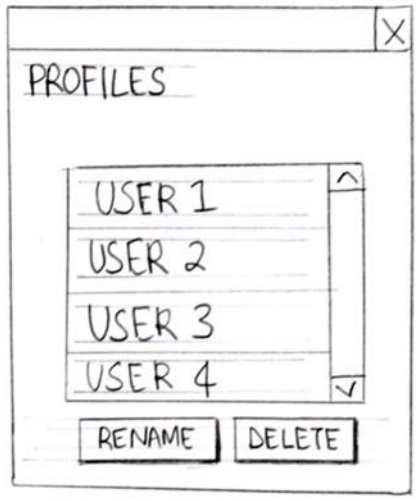


Figure 14: Profile Selection

When an existing profile is chosen on the main login screen, the user is shown the main Voiceture dialog (Figure 15). This is the dialog where the user is going to spend the bulk of their time, and therefore usability is key.

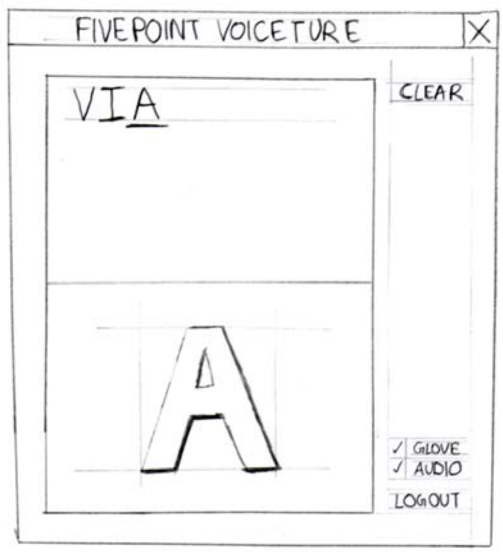


Figure 15: Main Voiceture Dialog

The main dialog is split into two main parts. The top half stores accumulated characters as they are recognized, and the bottom half shows the most recently recognized sign. On the right hand side, the user has the ability to clear the text accumulated and check the status of the glove. The user can also turn off the Voiceture audio output. At any time, the user can choose the “logout” command that takes them back to the login front screen.

The user interface will be created using the Java Foundation Classes (JFC) Swing packages. Swing is the latest Java standard for creating GUI's and feature portability between different architectures and operating systems.

The Java Speech API (JSAPI) [7] provides a cross-platform interface to support command and control recognizers, dictation systems and speech synthesizers. Unfortunately, the JSAPI is not a standard part of the Java Runtime Engine, and therefore we cannot guarantee that all installations will include the API.

We chose to tackle this difficulty in a far more elegant manner. We intend to manually record all of the 26 characters as WAV files and use the standard Java API [5] to play them out. The API is part of the javax.sound.* namespace. The Voiceture interface makes it easy to turn off this feature at any time.

7. Testplan

7.1. P5 glove

In order to ensure the quality of our prime input device, we verify the functionality of the P5 gloves we received by conducting several tests on them.

- Connectivity and glove usage by using the standard driver and bundled software
- Create simple programs that can poll the glove's hardware API for collection of real-time statistics. The test is then expanded to cover all available glove data, which in terms is used to create primitive gesture matching algorithm.
- The glove finger sensor profile is calibrated through the built-in calibration program system. This step allows the collected data to better reflect the human anatomy.

7.2. Sensor PCB

7.2.1. Software

The Java communication API serial port implementation will be test using two methods.

- Java JUnit test plan will be created for automated regression testing purpose. The test plan consists of a test routine that will stimulate interaction between the serial port and the API implementation.
- Interactions between humans and the glove while the data collected by the API is being monitored.

7.2.2. Firmware

A MPLAB assembly simulation program will be used to automate PIC assembly code testing. The success of the test will be used as the criteria of a correct PIC design.

7.2.3. Hardware

The hardware will be tested in 3 steps

- *Component connection verification:* During this stage, an engineering test will verify the connection of IC pins to board and component connections match the design schematics.
- *Verification of the DC voltage at all power nodes:* The PIC microcontroller's operation status will also be examined in this stage.
- *Capturing the sensor PCB's output data stream:* Doing so will guarantee that the sequence of data received by the PC is valid.

7.3. Analysis Algorithms

A program has been created to input the trained coordinates through the letter selection algorithm. This way, some errors can be easily tracked throughout the development process. The program inputs the data for the letters A through Z and checks that the subroutine outputs the corresponding letters.

To analyze the accuracy of the combined system, a large confusion matrix will be constructed with all the letters. Each letter will be tested 20 times to ensure accuracy. The percentage error of each letter will be calculated and tabulated as well.

7.4. User Interface & Integration

The user interface testing will incorporate the entire working system. These tests will be done by compiling the program and attempting to use the program as a typical user including:

- Create a new user
- Calibrate a new user
- Delete a profile
- Rename profile
- Checking ability of UI to show the translated characters properly
- Checking functionality of log out, sound on/off, and glove status indicators

8. References

[1] Communication Technology Laboratory, “American Sign Language Browser,” *Michigan State University*, 2000. [Online]. Available: http://commtechlab.msu.edu/sites/aslweb/browser.htm

[2] Craig Peacock, “Interfacing the Serial / RS232 Port,” *Beyond Logic*, June 2005. [Online]. Available: http://www.beyondlogic.org/serial/serial.htm

[3] D. Brayden, K. Eggen, G. Swami, P. Yuan, “Functional Specifications for the *Voiceture System”*. Fivepoint technologies, 2005

[4] D. Brayden, K. Eggen, G. Swami, P. Yuan, “*Project proposal for Voiceture System”*. Fivepoint technologies, 2005

[5] Java 2 Platform Std. Ed. V1.4.2, “Java API Package javax.sound.sampled,” *Sun Microsystems, Inc.* 2003. [Online]. Available: http://java.sun.com/products/java-media/sound/doc-sampled.html

[6] Java Software, “Java SDK 5.0 documentation. J2SE platform,” *Sun Microsystems, Inc.* 2004. [Online]. Available: http://java.sun.com/j2se/1.5.0/docs/

[7] Sun Developer Network, “Java Speech API,” *Sun Microsystems, Inc.* 2005. [Online]. Available: http://java.sun.com/products/java-media/speech/

9. ASL Reference [1]

A

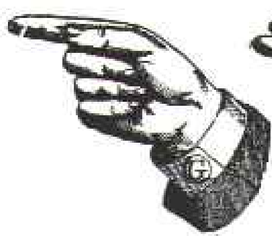
B

C

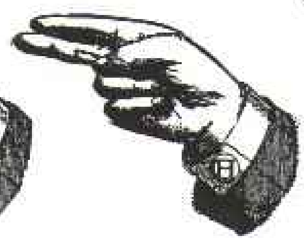
D

E

F

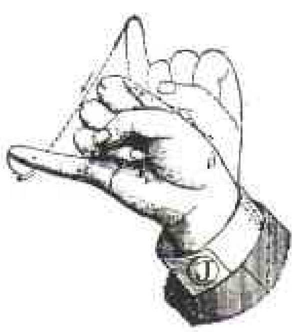


G



H

I



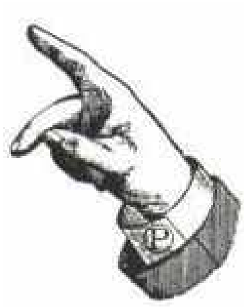
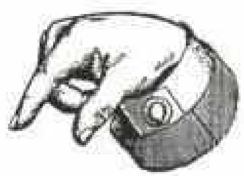
J

K

L



M **N** **O**

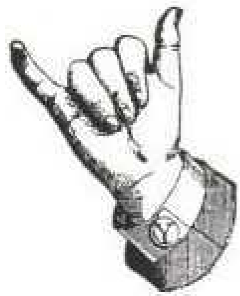

P **Q** **R**



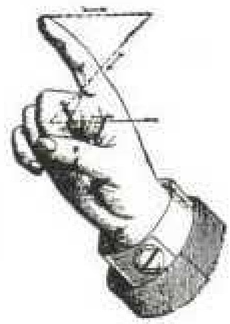
S **T** **U**



V **W** **X**



Y



Z