ECE 441 Senior Design
Preliminary Design Specification

Cell Phone Audio Controlled Point of Sale

Alana Sweat
Maggie Watkins
Lin Fu

Group 9
November 3, 2008
# TABLE OF CONTENTS

1. REVISION HISTORY 1

2. INTRODUCTION 2

2.1. CUSTOMER REQUIREMENTS & PROJECT BACKGROUND 2
  2.1.1. TECHNOLOGY REVIEW ANALYSIS - SYSTEMS 3

2.2. PROJECT RESEARCH 5
  2.2.1. TECHNOLOGY REVIEW ANALYSIS – BLOCKS 5

2.3. FEATURE SET 12
  2.3.1. ABSOLUTE MINIMUM REQUIREMENTS 12
  2.3.2. DESIRED FEATURE SET 13

3. ARCHITECTURAL OVERVIEW 14

3.1. IMPLEMENTATION APPROACHES: 15
  3.1.1. MICROCONTROLLER VS. PC 15
  3.1.2. GENERAL PURPOSE MICROCONTROLLER VS. DSP MICROCONTROLLER 15
  3.1.3. AMPLIFIERS 16

4. TOP LEVEL DESCRIPTION 17

4.1. TOP LEVEL BLOCK DIAGRAM 18
  4.1.1. TOP LEVEL INTERFACE DEFINITION 18

6. TESTING 19

6.2. SYSTEM TEST 19

APPENDIX A: CELLULAR TECHNOLOGY 22

APPENDIX B: NAMING CONVENTIONS AND GLOSSARY 24

APPENDIX C: REFERENCES 25
# 1. Revision History

<table>
<thead>
<tr>
<th>Revision</th>
<th>Date</th>
<th>Author</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>*</td>
<td>Nov 2, 2008</td>
<td>L. Fu</td>
<td>Created document from technology reports, and added new sections</td>
</tr>
<tr>
<td>A</td>
<td>Nov 2, 2008</td>
<td>M. Watkins</td>
<td>Added Appendix A and updated the title page</td>
</tr>
<tr>
<td>B</td>
<td>Nov 3, 2008</td>
<td>A. Sweat</td>
<td>Added table of contents, modified 2.1, added paragraph to 3.1.3, changed to single spacing per course syllabus, reworded several requirements and tests per Adam’s suggestions</td>
</tr>
</tbody>
</table>
2. Introduction

Picture this: you’ve been searching forever to find a parking place downtown, and you finally see one. You park the car and get out to add money to the parking meter, only to realize that you don’t have any change. However, you do have your cell phone. You dial an 800 number, enter a few codes at the prompt of the voice on the phone, and hold your cell phone up to the parking meter. In no time at all, a tone comes out of the phone, the parking meter hears it and adds the amount of money you specified to the meter, and you’re free to go about your business.

The Cell Phone Audio Controlled system will consist of three main components—a cell phone, a tone generator, and a point of sale device. To use the system, a user places a call from a cell phone to an 800 number. The tone generator receives the call, uses caller ID to recognize the caller, and then asks the caller for a PIN, the ID number of the point of sale system, and dollar amount to add to the meter. Next, the tone generator uses a secure encryption algorithm to encode the dollar amount of the purchase into an audible sequence of tones and sends it to the cell phone. As the tones play out the cell phone’s speaker, the caller holds the speaker up to a microphone on the point of sale device (e.g. parking meter) to transfer the message. The microphone receives the message and passes it to a decoder algorithm. Once the message is decoded, the money is added to the parking meter. Since the data is sent as a sequence of tones over the phone, the system will work from any cell phone on any network.

2.1. Customer Requirements & Project Background

The project sponsor, GE Security, is primarily interested in the algorithms and necessary hardware to encode a message, send it over an audio voice call on any cell phone network, and receive and decode the message at the decoding unit. Once developed, the technology could be used in any unmanned security system as a way for a user to provide personal identification. Because the message transferred could be sensitive (i.e. credit card number, ID number, etc.), data encryption security and system reliability are the essential requirements for this project. Other requirements that are important to the project sponsor include a way to measure the success rate, and a functional prototype that works with any cell phone.

The target customer of this project would be anyone that carries a mobile phone in any country around the world. The cell phone was initially designed for calling only, then a digital camera was added to it, followed by 3G (3rd generation). Now, the cell phone is becoming a type of credit card that people can use widely. In this modern technology world, people want things to become as easy as possible. From cash to credit card, now it is time for mobile phone credit card. It is a great idea for people whom do not always have cash around. Customers would want to use this product only if it is easy to function and takes equal or less time than previous paying methods.
### 2.1.1. Technology Review Analysis - Systems

<table>
<thead>
<tr>
<th>Product</th>
<th>Description</th>
<th>Data Transfer Method</th>
<th>Direction of Data Transfer</th>
<th>Operating Distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>GE Security TRACkey</td>
<td>Electronic key that records data about what and when it opened a lock. The data is transmitted to a server via a touch-tone telephone.</td>
<td>Touch-tone telephone</td>
<td>User to server</td>
<td>1/4&quot; to 1/2&quot;</td>
</tr>
<tr>
<td>Acoustically Coupled Modem</td>
<td>Converts data (ones and zeros) into audible tones and converts audible tones into data.</td>
<td>Telephone</td>
<td>Bi-directional</td>
<td>As close as possible</td>
</tr>
</tbody>
</table>

GE Security’s TRACkey [1] is very similar to this project in that it transmits data over an audio phone call. A TRACkey user holds the TRACkey handheld device up to a telephone receiver, presses a sequence of keys, and waits for the audible message to be transmitted to a data server at the other end of the telephone (see figure 1 below). The most obvious difference is the direction of data transfer. The TRACkey sends data from the key through the phone to the server, but the audio POS system will send data from the server through the phone to the POS device. Since the data direction of the audio POS system is reversed compared to the TRACkey, the encoding and decoding responsibilities are also reversed. The TRACkey encodes the data and the server decodes the data. Whereas, the server will encode the data and the POS device will decode the data for the audio POS system.

![Figure 1: TRACkey in use](image-url)
The TRACkey is specifically designed to transmit data over a landline not over a cell phone network. Because cell phone networks apply audio compression to the signals sent over the networks, the signal encoding method and data transfer rate required for a successful message transfer will likely be different for the audio POS system than for the TRACkey.

Starting in the late 1960s, acoustically coupled modems were used to enable computers to transmit and receive data over a telephone line using a standard telephone handset [2]. The acoustic coupler would connect to the computer with a wire and the telephone handset would fit into two rubber seals (see figure 2). The computer would send data in the form of ones and zeros to the coupler. The coupler converted the data into audible tones that could be picked up by the telephone and transferred to another computer at the other end of the phone. The other computer could send data back using the same form of communication. Data transfer rates for acoustically coupled modems were usually about 300bps [3].

The acoustically coupled modem is very similar to the audio POS system in the way it transfers data. The main difference is that the acoustically coupled modem was designed to work with a landline not over a cellular network. Even though a data transfer rate of 300bps is slow relative to modern transfer rates, it may be too fast for a cellular network to transmit without losing some data.
2.2. Project Research

2.2.1. Technology Review Analysis – Blocks

Cell Phones

<table>
<thead>
<tr>
<th>Model</th>
<th>Manufacturer</th>
<th>Technology</th>
<th>Frequencies/ Bands</th>
<th>Speaker Phone</th>
<th>Antenna</th>
<th>Data</th>
<th>Voice Dialing</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>NOKIA 5165</td>
<td>NOKIA</td>
<td>TDMA</td>
<td>850, 1900</td>
<td>No</td>
<td>Stub</td>
<td>No</td>
<td>No</td>
<td>$22.99</td>
</tr>
<tr>
<td>LG CU515</td>
<td>LG</td>
<td>GSM</td>
<td>850, 900, 1800, 1900</td>
<td>Yes</td>
<td>Internal</td>
<td>Edge, HSDPA</td>
<td>No</td>
<td>$230.00</td>
</tr>
<tr>
<td>SONY ERICSSON S500</td>
<td>SONY</td>
<td>GSM</td>
<td>850, 900, 1800, 1900</td>
<td>Yes</td>
<td>Internal</td>
<td>Edge, GPRS</td>
<td>Yes</td>
<td>$300.00</td>
</tr>
</tbody>
</table>

The Nokia 5165 was an older model cell phone that was very popular around 10 years ago. Built-in speakerphones were not available yet at that time. But almost every cell phone out on the market right now has a speakerphone. Each carrier licenses a particular frequency from the FCC in certain geographical areas. In the United States, only the 800/850MHz, 1900MHz, and now 2100MHz frequency may be used for cellular communications. All phones sold by the carriers will support both the 800/850 and 1900 bands. Most carriers prefer to use the higher frequencies as they can deliver more call and data volume through the higher bands. The higher frequencies tend to be more active in densely populated areas while rural areas are likely covered by the 800 bands [4]. See Appendix A for more information about TDMA, GSM, and other cellular technologies.
The microphone is an important part of the system, as it has to be able to accurately pick up a sound coming from a cell phone speaker. The microphones reviewed in the table are all electret condenser microphones, but they all have different characteristics. They all have an adequate frequency range, but have differing frequency responses. The third microphone is the only one with a non-varying frequency response, and it is also waterproof, which would be useful for a system that theoretically would be operated outdoors. However, the noise-canceling aspect of the CUI microphone is also interesting. The final decision on which microphone to use will probably only come after several microphones have been tested for performance.
Amplifiers

<table>
<thead>
<tr>
<th>Part Number</th>
<th>Manufacturer</th>
<th>Power Supply</th>
<th>Load</th>
<th>Average Power Deliver</th>
<th>Features</th>
<th>Shutdown Current</th>
<th>Other Features</th>
<th>Applications</th>
</tr>
</thead>
<tbody>
<tr>
<td>LM4992</td>
<td>National Semiconductor</td>
<td>5 VDC</td>
<td>8 ohms BTL</td>
<td>1 watt</td>
<td>2.2 – 5.5 V operation</td>
<td>0.2uA</td>
<td>Unity gain stable</td>
<td>Portable electronic device</td>
</tr>
<tr>
<td>LM4917</td>
<td>National Semiconductor</td>
<td>3 V</td>
<td>16 ohms</td>
<td>95 mW</td>
<td>1.4 – 3.6 V operation</td>
<td>0.1uA</td>
<td>Ultra low current shutdown mode</td>
<td>Portable electronic device</td>
</tr>
</tbody>
</table>

The LM4992 is a stereo audio power amplifier primarily designed for demanding applications in mobile phones and other portable communications device applications. The LM4992 does not require output coupling capacitors or bootstrap capacitors, and therefore is ideally suited for mobile phone and other low voltage applications where minimal power consumption is a primary requirement. The LM4917 provides high quality audio reproduction with minimal external components. A ground referenced output eliminates the output coupling capacitors typically required by single-ended loads, reducing component count, cost and board space consumption. This makes the LM4917 ideal for mobile phones and other portable equipment where board space is at a premium. Eliminating the output coupling capacitors also improves low frequency response.
### A/D Converters

<table>
<thead>
<tr>
<th>Location</th>
<th># Channels</th>
<th>Sample Data Size</th>
<th>Max Sample Rate</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATmega128 [6]</td>
<td>8</td>
<td>10 bit</td>
<td>3.8 kHz</td>
<td>2 channels w/ optional gain</td>
</tr>
<tr>
<td>STM32 [12]</td>
<td>10</td>
<td>12 bit</td>
<td>unknown</td>
<td>Board also has D/A converter</td>
</tr>
<tr>
<td>AT32AP7000 [5]</td>
<td>2</td>
<td>16 bit</td>
<td>50 kHz</td>
<td>Board also has D/A converter</td>
</tr>
</tbody>
</table>

The three A/D converters in the table are all integral parts of their respective microcontroller. Of the categories compared, the number of channels is fairly irrelevant for the system, as there will only be one signal coming into the microcontroller at a time. The sample data size is basically the level of resolution that it is possible to achieve. More bits equal a more accurate translation from analog signal to digital signal. By that criterion, the third board definitely has the best accuracy, and it can also sample at very high speeds. However, for this application the frequencies used will probably be fairly low (<1 kHz), so any of the three boards would give a fairly accurate signal conversion.
### Code

<table>
<thead>
<tr>
<th>Language</th>
<th>Platform</th>
<th>Efficiency</th>
<th>Level of Difficulty</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>PC, AVR microcontrollers</td>
<td>Depends on compiler and code written</td>
<td>Fairly easy, a lot of pre-defined functions, easy to write own functions</td>
<td>Language have most experience with</td>
</tr>
<tr>
<td>Assembly</td>
<td>PC, any microcontroller</td>
<td>Compact code, doesn’t need to be translated from high-level language</td>
<td>Not many functions, difficult to write complex code</td>
<td>Least amount of experience</td>
</tr>
<tr>
<td>Matlab</td>
<td>PC</td>
<td>Unknown</td>
<td>Easy, many pre-defined functions</td>
<td>Very good with DSP</td>
</tr>
</tbody>
</table>

Of the three coding languages listed, assembly will probably not be used by itself, as it is difficult to write complex code, such as DSP algorithms, since there are not very many instructions built for assembly, and it requires a lot of practice and skill. Of the other two, both will probably be used. Matlab will be very useful for the signal generation side of the system, as well as for preliminary testing of the receiving end of the system. It has a lot of built-in DSP functions, such as FFT, and everyone in the group has at least some experience using Matlab. C will most likely be used to program the microcontroller, since the only coding languages compatible with the microcontrollers are C and assembly. It will take some experimentation to come up with DSP algorithms in C, but there is also open source code available on the web that can be adapted for this application.
### Microcontrollers

<table>
<thead>
<tr>
<th>Part Number</th>
<th>Manufacturer</th>
<th>Development Board Info</th>
<th>Speed</th>
<th>Supply Voltage</th>
<th>Flash Memory</th>
<th>SRAM</th>
<th>Data width</th>
<th>Price</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATmega128</td>
<td>Atmel</td>
<td>Tekbots kit (#4887368)</td>
<td>16 MHz</td>
<td>4.5V – 5.5 V</td>
<td>128K Bytes</td>
<td>4K</td>
<td>8-bit</td>
<td>$114.95</td>
<td>General usage</td>
</tr>
<tr>
<td>STM32 (F101RB)</td>
<td>STMicroelectronics</td>
<td>STM3210B-EVAL</td>
<td>36 MHz</td>
<td>2.0V – 3.6 V</td>
<td>128K Bytes</td>
<td>16K</td>
<td>32-bit</td>
<td>$210.00</td>
<td>Optimized for DSP</td>
</tr>
<tr>
<td>AT32A P7000</td>
<td>Atmel</td>
<td>ATNGW1000</td>
<td>150 MHz</td>
<td>3.0V – 3.6 V</td>
<td>External</td>
<td>32K</td>
<td>32-bit</td>
<td>$89.99</td>
<td>Optimized for DSP</td>
</tr>
</tbody>
</table>

The microcontroller will be used to decode the encrypted message from the cell phone. It needs to be powerful and fast enough to execute the decoding algorithm. Characteristics to consider are the clock, data width, memory size, and ease of use. The three microcontroller development boards reviewed in the table above have different combinations of these features. The ATmega128 is definitely the smallest and slowest, but it will also be the easiest to use. The other two are faster and larger, but will be more difficult to use.
## LCD Display

<table>
<thead>
<tr>
<th>Item Number</th>
<th>Manufacturer</th>
<th>Maximum Display Digits</th>
<th>Power Supply Voltage</th>
<th>Character Generator ROM (CGROM)</th>
<th>CPU bus timing</th>
<th>Double Data SDRAM (DDRAM)</th>
<th>CGRAM</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>HD44780S</td>
<td>Hitachi</td>
<td>16 digits (8 digits x 2 lines)</td>
<td>5V ±10%</td>
<td>7200 bits</td>
<td>1 MHz</td>
<td>80 bytes</td>
<td>64 bytes</td>
<td>$9.95</td>
</tr>
<tr>
<td>HD44780U</td>
<td>Hitachi</td>
<td>16 digits (8 digits x 2 lines)</td>
<td>2.7V – 5.5V</td>
<td>9920 bits</td>
<td>1 MHz (Vcc = 3V) 2 MHz (Vcc = 5V)</td>
<td>80 bytes</td>
<td>64 bytes</td>
<td>$9.95</td>
</tr>
</tbody>
</table>

There are no huge differences between the two different models. But HD44780U uses slightly less voltage. It is suitable for any portable battery-driven product requiring low power dissipation, which could be one of the considerations for the POS design. If we use a Vcc of 5 volt, the HD44780U model will have a high speed twice as fast as HD44780S for MPU bus interface. A single HD44780U can display up to one 8-character line or two 8-character lines, which is more than enough for our device. The CGROM is extended to generate 208 of 5 X 8 dot character fonts and 32 of 5 X 10 character fonts for a total of 240 different character fonts.
2.3. *Feature set*

2.3.1. **Absolute Minimum Requirements**

- **Customer Requirement #1**: A message is transmitted using audible tones over an audio voice call. (10/100)
- **Customer Requirement #2**: Send a message with an 80% success rate. (20/100)
  
  **Engineering Requirement**: Send a 15-bit message with an 80% probability of success.

- **Customer Requirement #3**: A message can be successfully sent at a distance of ½ inch. (10/100)
  
  **Engineering Requirement**: The system can successfully decode at least one message when the cell phone speaker is within ½ inch of the microphone without using a speakerphone.

- **Customer Requirement #4**: The system can successfully send 100-bits of data. (10/100)

- **Customer Requirement #5**: Create a working prototype of the system. (10/100)
  
  **Engineering Requirement**: Create a working prototype of the system that uses a microcontroller to decode the audio signal and provides a way to view the decoded message.

- **Customer Requirement #6**: Display the message received. (10/100)
  
  **Engineering Requirement**: Display the decoded message received on an LCD display.

- **Customer Requirement #7**: A message can successfully be sent in a variety of environments. (10/100)
  
  **Engineering Requirement**: The system meets minimum success requirements when a recording of a busy street or a loud room is playing near the system at the audio dB level that the noise was recorded.

- **Customer Requirement #8**: Suggest a way to protect against each of the following security attacks. (10/100)
  
  1) Recording and playing-back of the audio message
  2) Making a phone number appear to be a different number to a caller-ID system
  3) Using a tone generator to ‘trick’ the POS devise into thinking money has been added to it
Customer Requirement #10: Compare the performance of different frequency encoding algorithms and baud rates. (10/100)

Engineering Requirement: Prototype three encoding and decoding algorithms (DTFM, FSK, and OOK), test each at four different baud rates (10Hz, 20Hz, 50Kz, and 100Hz) using the same microphone that will be used in the prototype. Compare the bit-error-rate of each in a matrix.

2.3.2. Desired Feature Set

Customer Requirement #10: A message can be successfully sent at a distance of 3 inches. (15/50)

Engineering Requirement: The system can successfully decode at least one message when the cell phone speaker is within 3 inches of the microphone using either the standard speaker or a speakerphone.

Customer Requirement #11: Send a message with a 99% success rate. (10/50)

Engineering Requirement: Send a 100-bit message with a 99% success rate.

Customer Requirement #12: Implement an IVR system that auto-generates a sequence of tones based on user input from the phone. (10/50)

Customer Requirement #13: Confirmation tone that indicates successful data transmission. (15/50)

Engineering Requirement: A confirmation tone sounds when the message transfer is complete.
3. Architectural Overview

The audio controlled point of sale system will enable users to make purchases using a cell phone. The system will use audio signals to transfer data rather than RFID, Bluetooth, or infrared technology, thereby allowing users to use either a very basic cell phone or a full-featured smart phone. To make a purchase the user would dial an 800 number and follow the prompts. The computer that receives the call will use caller ID to detect the phone number the user is calling from, and then ask the user to enter a PIN. Upon successful PIN entry, the computer will ask the user to hold the cell phone speaker up to the microphone receiver on the POS device and wait for the message to transfer. When the message transfer is complete the user will hear a tone from the POS device and see the sale amount on a display indicating the purchase is complete.

An audio controlled POS system would have many uses. It will be small and could easily be designed to operate on battery power. It will be easy to use, and will not require a physical connection to the cell phone or any other system. These features make it versatile for many applications. An audio controlled POS system would increase parking meter revenue by eliminating the costs associated with collecting coins from the meters. It could also be used in vending machines, movie theaters, and convenience markets to sell a product to consumers without the need for a cashier.

The server that sends the data signal to the cell phone would also keep a record of all sales data and will transfer the data to the vendor upon request. Since the POS system does not need to transfer sales data, the maintenance needs for the system will be limited to an occasional battery replacement.

The POS system will be secure against 1) record and playback attacks where a person records the audio tones from the cell phone and plays them back to make an illegitimate purchase, 2) caller ID spoof attacks where a person makes his/her cell phone number appear as a different number to the caller ID system, and 3) random tone generation attacks where a person uses a tone generator to generate a sequence of random tones to “trick” the POS system into a false sale.

The design goal is to attain a 99% data transmission success rate when transferring the audio message from the computer, through the cell phone, and into the POS device. To help achieve the success rate goal, the design will utilize a quality microphone and pre-amp circuit so that the microphone and pre-amp will not contribute to data loss. The design will also incorporate a microcontroller with a decoding algorithm and baud rate that optimizes data transfer success.

The baud rate does not necessarily need to be large to complete a transaction in a reasonable amount of time. Since the message size does not need to be very large, a slow baud rate will not be a nuisance to the user. For example, a 100-bit message sent at 10 baud would take only 10 seconds to transmit. If the same message were sent at 300 baud (the minimum rate of most modems) it would take only 0.33 seconds to transmit.
To make the audio decoding robust enough to work in noisy environments, the design will include any necessary filters or noise gates between the amplifier and the microcontroller to ensure only the intended data is allowed into the microcontroller for processing.

The first step in building the audio controlled POS system is to build a prototype. The first prototype will not necessarily include every feature described above, but will include all of the features listed in the minimum requirements section 2.3.1. The goal of the first prototype is to determine the best encoding and decoding algorithm and baud rate for the system so that it will work in a noisy environment. Future prototypes will incorporate additional features.

3.1. Implementation Approaches:

3.1.1. Microcontroller vs. PC

The end result of the audio controlled POS system will need to use a microcontroller to decode the signal received from the cell phone, but trying to work with a microcontroller from the beginning of the project could be very difficult. Before figuring out the specific code needed for the microcontroller, a great deal of experimentation with different algorithms, baud rates, and frequencies to see what gets through the cell phone network, out the speaker of a cell phone, and into the microphone with the least error is going to have to take place. Trying to come up with multiple coding algorithms on a microcontroller purely for testing purposes would be adding unnecessary work to the project.

Instead, starting out working with a PC hooked up to a microphone makes more sense. This will enable the use of programs like Matlab, which has a lot of built-in DSP functions, to figure out which frequencies and baud rates will work best for POS project. It will also facilitate easily writing code to quickly measure success rates and generate data comparisons. Once exactly how the data is going to be transferred from the cell phone to the decoding device is determined, the code written for the PC can be used to help write the code needed for the microcontroller.

3.1.2. General Purpose Microcontroller vs. DSP Microcontroller

There are two main types of microcontrollers: general-purpose microcontrollers and DSP microcontrollers. General-purpose microcontrollers, such as the ATmega128, have a wide variety of I/O interfaces, such as SPI, USART, and general purpose I/O. They also tend to be designed to handle external events via interrupts. DSPs, on the other hand, are generally built to handle real-time events, such as ongoing A/D conversions, and usually have fewer I/O interfaces [14].

The microcontroller for the audio controlled POS system is going to have to handle a lot of A/D conversions, and it would appear that using a DSP would work better. However, one of the cons of DSP microcontrollers is the fact that there are not a lot of great development tools, such as C compilers and debuggers, which will work with
DSPs. This means that writing the code for the microcontroller could be very problematic and take a very long time to figure out.

The general-purpose microcontroller reviewed in the technology analysis, the ATmega128, has an A/D converter that can sample data at up to 3.8 kHz. This will most likely be fast enough for the POS application, since there will not be too much data transferred at one time (only up to 100 bits), and even at a fairly slow rate it should not take more than several seconds. Because of the way the ATmega128 is built, it might not be able to perform computations such as the Fast Fourier Transform (FFT) in real time, but again, since our data size is not very large, and the data transmission is only periodic, the microcontroller should be able to do the job. So, even though a DSP might be able to handle A/D conversions and calculations more efficiently, the ATmega128 will be much easier to program, and should be able to do the job required for this application.

3.1.3. Amplifiers

There are four different classes of amplifiers: class A, B, AB, and C. Two primary items determine the class of operation of an amplifier: (1) the amount of bias and (2) the amplitude of the input signal. Two things to keep in mind when choosing the best fit for a device: fidelity and efficiency.

Fidelity is the faithful reproduction of a signal. If the output of an amplifier is just like the input except in amplitude, the amplifier has a high degree of fidelity. Class A amplifier has a high degree of fidelity. Class AB amplifier has less fidelity, and class B and class C have poor fidelity. In this case for our design, fidelity is the most important thing, therefore class B and class C would be eliminated from our choice already.

Efficiency of an amplifier refers to the power usage. An amplifier has two input power sources: one from the signal, and one from the power supply. Since every device takes power to operate, by using more power, an amplifier has less power available for the output signal, thus the efficiency of the amplifier is low. This is the case with class A amplifier. Even with no input signal, the class A amplifier still uses power from the power supply. On the other hand, class AB in this case leads to better efficiency [15].

Depending on the power source for the POS device, class A may or may not be the right choice. If batteries are used, class A would not be a great choice, but if a power cord is used, class A should be okay.

To actually implement the amplifier block, a pre-made IC will most likely be used to simplify the circuit. Which chip is implemented will depend on the amount of gain required and the voltage levels gotten out of the microphone circuit.
4. Top Level Description

<table>
<thead>
<tr>
<th>Name</th>
<th>Interface Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Input</td>
<td>Input</td>
<td>Keypad and volume control of user’s cell phone, user also must position phone correctly</td>
</tr>
<tr>
<td>Audio Signal</td>
<td>Input</td>
<td>Within range of comfortable human hearing, will contain 15-100 bits of data</td>
</tr>
<tr>
<td>Environment</td>
<td>Input</td>
<td>Noise (most problematic at senior expo)</td>
</tr>
<tr>
<td>Power</td>
<td>Input</td>
<td>Voltage: 2 – 5.5V From plug-in transformer to development board</td>
</tr>
<tr>
<td>Desired Action</td>
<td>Output/Action</td>
<td>Display showing transmitted data Optional: real-time clock, ASCII characters</td>
</tr>
</tbody>
</table>
4.1. Top Level Block Diagram

![Block Diagram]

4.1.1. Top Level Interface Definition

<table>
<thead>
<tr>
<th>Name</th>
<th>Interface Type</th>
<th>Specifications</th>
</tr>
</thead>
<tbody>
<tr>
<td>User_Input</td>
<td>Input</td>
<td>Standard cell phone keypad and volume controls, user must manually position cell phone in correct location</td>
</tr>
</tbody>
</table>
| Audio_Signal       | Input          | Frequency: 20-20kHz
|                    |                | Contains 15-100 bits of data                                                  |
| Noise              | Input          | 0 - 100dB at frequencies from 20 – 20kHz                                      |
| Small_Analog_Signal| Internal Signal| Frequency: 20 – 20kHz
|                    |                | Voltage: 0 – 5V                                                                |
| Amplified_Analog_Signal | Internal Signal | Frequency: 20 – 20kHz |
|                    |                | Voltage: 0 – 5V                                                                |
| Digital_Signal     | Internal Signal| Sample Rate at least 80 – 80kHz
|                    |                | Voltage: 0 – 5V                                                                |
| Decoded_Data       | Output         | SPI interface, at most 8 MHz data transmission speed
|                    |                | Voltage: 0 - 5V                                                                |
| Results            | Output         | Visual Display
|                    |                | At least shows message received as a string of 1’s and 0’s
|                    |                | Optional: real-time clock, ASCII characters                                    |
| Power              | Input          | Voltage: 4.5 – 5.5V
|                    |                | From transformer plugged into wall to microcontroller board                   |
| V+                 | Internal Power  | Voltage: 5.0V from microcontroller
|                    |                | DC Current: 200-400 mA                                                        |
| USB_Cable          | Internal Connection | Code written in C compiled to assembly Programmed via USB programmer          |
6. Testing

6.2. System test

- #1: A message is transmitted using audible tones over an audio voice call.
  PASS: The system uses only audio to transmit a message.
  FAIL: The system does not use audio to transmit a message.

- #2: Send a message with an 80% success rate.
  1) Hold speaker up to microphone.
  2) Transmit 1000-bits of data.
  3) Compare decoded message with original message.
  4) Record encoding algorithm, frequency, and baud rate used.
  5) Calculate the bit-error-rate and use it to determine the probability that 15 consecutive bits are transmitted correctly.
  PASS: Probability of transmitting 15-bits correctly is 80% or higher.
  FAIL: Probability of transmitting 15-bits correctly is less than 80%.

- #3: A message can be successfully sent at a distance of ½ inch.
  1) Maximize the volume of the cell phone.
  2) Hold cell phone speaker over microphone at a distance of ½ inch.
  3) Send a message over the phone.
  4) Record encoding algorithm, frequency, and baud rate used.
  5) Compare the original message to the decoded message by using a display, serial output to a PC running a terminal emulation program, or an oscilloscope.
  PASS: At least one message is sent successfully in 10 attempts or fewer.
  FAIL: No messages are sent successfully in 10 attempts.

- #4: The system can successfully send 100-bits of data.
  1) Hold cell phone up to the microphone.
  2) Send a message containing 100-bits of data.
  3) Record encoding algorithm, frequency, and baud rate used.
4) Compare the original message to the decoded message by using a display, serial output to a PC running a terminal emulation program, or an oscilloscope.

PASS: At least one message is sent successfully in 10 attempts or fewer.
FAIL: No messages are sent successfully in 10 attempts.

● #5: Create a working prototype of the system.

1) Make a voice call from a cell phone to a system that is capable of playing tones.
2) Hold the phone up to the microphone.
3) View the decoded message by using an LCD display, serial output to a PC running a terminal emulation program, or an oscilloscope.

PASS: The original message is displayed as a direct result of the audio transmission.
FAIL: The original message is not displayed as a direct result of the audio transmission.

● #6: Display the message received.

1) Hold cell phone up to the microphone.
2) Send a message.
3) Look at the display.

PASS: The decoded message is displayed.
FAIL: The decoded message is not displayed.

● #7: A message can successfully be sent in a variety of environments.

1) Use an audio volume tester to play recorded sounds at the dB level they were recorded.
2) Hold cell phone up to the microphone.
3) Send a message.
4) Record encoding algorithm, frequency, and baud rate used.
5) Measure success rate.

PASS: Met the success rate and explained in a written document how we had to adjust the protocol and baud rate to meet the success rate.
FAIL: Did not meet the minimum success rate when recorded noise was playing in the background.

● #8: Suggest a way to protect against each of the following security attacks.
PASS: Suggested at least one way to protect against each security threat.
FAIL: Did not suggest at least one way to protect against each security threat.

- #9: Compare the bit-error-rates of different frequency encoding algorithms and baud rates.
  PASS: Completed a table with all of the results.
  FAIL: Did not complete a table will all of the results.
Appendix A: Cellular Technology

First generation cell phones use analog cellular technology. Digital cell phone technology is considered second generation and is usually referred to as 2G. The three common 2G network technologies are frequency division multiple access (FDMA), time division multiple access (TDMA), and code division multiple access (CDMA). Global system for mobile communications is based on TDMA.

FDMA separates the frequency spectrum into equal segments of bandwidth to get several unique voice channels. Each phone uses a different frequency. It can be used for digital transmission, but is more commonly used for analog transmission.

TDMA uses 30-kHz channel spacing and three time slots per channel. One channel is shared between three different cell phone conversations based on time. It divides a narrow band that is 6.7ms long and 30-kHz wide into three equal time slots. The Electronics Industry Alliance and the Telecommunications Industry Association use TDMA for Interim Standard 54 (IS-54) and Interim Standard 136 (IS-136). TDMA operates in the 800-MHz (IS-54) or 1900-MHz (IS-136) frequency band.

CDMA digitizes the data, divides it into data segments, and spreads the segments over the full bandwidth. With this approach, several calls can use the same channel. Each call's signal uses a unique spreading code to spread the data over the entire bandwidth section. The receiver uses the same unique code to recover the signal. Since the data segments are divided up on the bandwidth, they are not necessarily in chronological order. In order to put each segment back into the correct order once received, CDMA uses a GPS system to put an accurate time-stamp on each segment. CDMA operates in the 800-MHz and 1900-MHz frequency bands. IS-95 is based on CDMA technology. Up to 10 calls can be transmitted using CDMA in the same bandwidth that one analog call requires.

GSM uses TDMA, but in a way that is incompatible with IS-136. GSM systems implement encryption to secure the phone call. In the US, GSM operate in the 900-MHz and 1800-MHz bands, but in Europe and Asia, it uses the 850-MHz and 1900-MHz bands. Europe, Africa, Australia, and parts of Asia use GSM as an international standard. However, since the frequency bands used in the US are different from those used in other countries, the international standard is not compatible with US GSM phones. Integrated Digital Enhanced Network (IDEN), used by Nextel, is based on GSM.

Some cell phones support multiple bands or multiple modes or both. A phone that supports multiple bands can switch operating frequencies as needed. One example is a dual-band TDMA phone that can switch between 800-MHz and 900-MHz. A cell phone that supports multiple modes can switch between two different transmission technologies. An example is a phone that can switch from AMPS (analog) to TDMA (digital) when one technology is not available in a particular area.

Generation 2.5

Since smartphones demand more advanced protocols that 2G technology offers, network engineers of early smartphones developed protocols to fit the needs of the new
phones. Because the new protocols where not considered innovative enough to be called the third generation, they were dubbed generation 2.5. Some of the protocols in generation 2.5 are still in use. Generation 2.5 protocols include General Packet Radio Services (GPRS) and Enhanced Data GSM Environment (EDGE). GPRS can communicate data at a rate of 114 Kbps. EDGE can transmit data at a rate of 384 Kbps. [16]

**Third Generation**

Third generation cellular technology is designed for multimedia smartphones. They offer a larger bandwidth for higher volumes of data transfer. The three most common 3G cellular technologies are Wideband Code Division Multiple Access (WCDMA), Time-division Synchronous Code-division Multiple Access (TD-SCDMA), and CDMA2000 a technology that is based on 2G Code Division Multiple Access.

Although 3G technologies are available through some US providers, the 2.5G protocols are still very common for US smartphones.
Appendix B: Naming Conventions and Glossary

A/D – Analog to digital
Customer Requirement – A requirement that may or may not be able to be tested as is. A requirement supplied by the customer, sponsor, or mentor.
D/A – Digital to analog
DSP – Digital signal processing
DTMF – Dual tone multi frequency
Engineering Requirements – A requirement that can be tested and evaluated through a step-by-step process. Usually a numerical specification is included.
FSK – Frequency shift keying.
IC – Integrated circuit
IVR – Interactive voice response
Modem – A device that modulates and demodulates signals [2].
OOF – On-off frequency.
POS – Point of sale.
System – The complete system that you are designing. This includes all blocks in your design.
Appendix C: References


