School of Engineering Science – Burnaby, BC – V5A 1S6
aaa75@sfu.ca

March 11, 2010

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

Re: ENSC 440 Design Specifications for a Musical Carpet

Dear Dr. Rawicz:

The enclose document, *Design Specification for a Musical Carpet*, describes the technical design requirements for the educational device we are developing. The musical carpet will replicate one octave of a piano embedded into a carpet and will contain additional features that will act as educational aids. This device allows the user to grasp the fundamentals of music in an entertaining fashion.

This design specification is used to simplify the implementation process by creating a comprehensive set of guidelines, which can be followed throughout the implementation process. These specifications apply to the proof-of-concept model but many are also applicable to the production model.

MusEd Technologies is a team of four driven and skilled individuals: Anthony Tsang, Anton Ayzikovsky, Danny Jiang and Payam Norouzi. If you would like to contact us with questions or comments, please contact Anton Ayzikovsky via e-mail at aaa75@sfu.ca.

Sincerely,

Anthony Tsang

Anthony Tsang

MusEd Technologies Inc.

Enclosure: Design Specification for a Musical Carpet

DESIGN SPECIFICATION FOR A MUSICAL CARPET

Project Team:	Anthony Tsang Anton Ayzikovsky Danny Jiang Payam Norouzi
Contact Person:	Anton Ayzikovsky aaa75@sfu.ca
Submitted to:	Dr. Andrew Rawicz – ENSC 440 Steve Whitmore – ENSC 305 School of Engineering Science Simon Fraser University

Issued date:	March 11, 2010
Revision:	1.0

EXECUTIVE SUMMARY

Many studies have been performed linking the healthy development of children and music. There has been some controversy over the legitimacy of these studies. But even if those studies are in question, anecdotal evidence suggests that music is beneficial regardless of whether it can make your child smarter or not. Music is often a highly influential factor in people's lives and we hope to be able to continue this in future generations.

We have provided detailed technical requirements that can be used to streamline development. This can also be use the promote uniformity between separate modules and will be critical to streamlining integration. The document also contains our system test plans, whose function is two-fold. Firstly, the test plans can be implemented for testing when development is well underway or nearing completion. Secondly, the system test plan helps paint a picture of desired functionality which complements the requirements listed in this document.

Development has been divided into two teams. One team will program the microcontroller on a development board. This consists of the main controlling unit, the audio processing unit and pitch detection sub-unit. The second team will work on the controls and system outputs, testing their functionality separately. As soon as the parts allow for integration, we will begin integrating the separate components so that testing can be performed.

Table of Content

List of Figures

List of Tables

Glossary

ADC	Analog-to-Digital Converter
APU	Audio Processing Unit
DAC	Digital-to-Analog Converter
DIP	Dual In-Line Package
EEPROM	Electrically Erasable Programmable Read-Only Memory
EMC	Electromagnetic Compatibility
EMI	Electromagnetic Interference
ESD	Electrostatic Discharge
FFT	Fast Fourier Transform
FCC	Federal Communication Commission
I2C	Inter-Integrated Circuit
LED	Light-Emitting Diode
MCU	Micro-Controller Unit
OpAmp	Operational Amplifier
PCB	Printed Circuit Board
SRAM	Static Random Access Memory
SPI	Serial Peripheral Interface
TWI	Two Wire Interface
USART	Universal Asynchronous Receiver/Transmitter

1. INTRODUCTION

The musical carpet is an educational device aimed at younger children. The goal is to create a product that can both teach and entertain. The device consists of one octave of piano keys that can be played intuitively primarily with your feet, though the device will also be able to synthesize other instruments. This product also includes a play and sing mode, which will indicate whether the singer's voice is in tune with the note played. Also implemented is a record and playback mode, which will capture the played notes and play them back on command. This design specification document provides the technical implementation plans for each part of the musical carpet and serves to complement the functional specification.

The design specification document is intended for internal use within MusEd Technologies Inc. as well as for external reviewers. The project manager will use this document to ensure that development is progressing as planned. The design engineers will refer to this document to guarantee product compliance while the test engineers will use the test plans to conduct unit tests throughout the development process.

2. SYSTEM OVERVIEW

2.1 MECHANICAL DESIGN

The system consists of two main modules. Sensor module is embedded into the carpet, which is connected to the control module, an on-wall art installation with control buttons, status/feedback LEDs, microphone and speakers. A wire bus connects the two modules. The information about the carpet structure can be found in "Sensing Module" section and the information about control/display board can be found in "Control/Display Module" section.

2.2 HIGH LEVEL BLOCK DIAGRAM

Figure 2-1 describes the high-level block diagram for the musical carpet. It includes a sensor module, control module, main control unit, audio processing unit, display module, speaker and microphone.

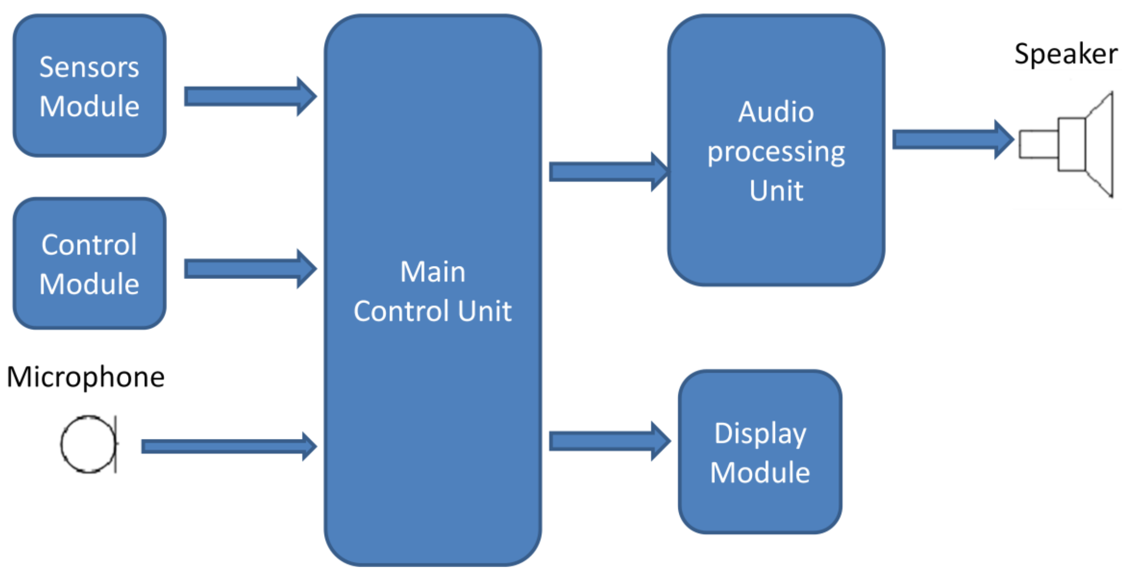

Figure 2-1. High Level Design Block Diagram

2.3 MODE OF OPERATIONS

The musical carpet has three modes of operation, the basic mode, the vocal training mode and the record/playback mode. These modes are described in Figure 2-2.

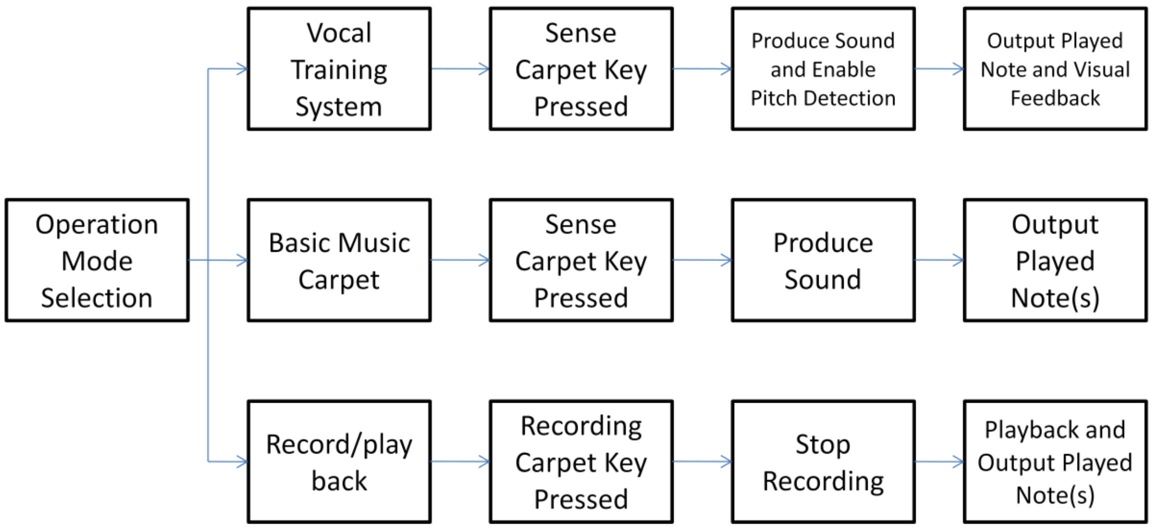

Figure 2-2. Operational Block Diagram

The system consists of a set of step-operated pushbuttons embedded into a carpet, which are connected to the main control unit. The processor sends pre-recorded sound samples of the corresponding frequency out to the speaker system. During the vocal training mode, the pitch detection system is enabled after a note has been played. The system compares the user's voice frequency with the frequency of the note that was played. The LED indicator shows if the user sings too low, too high or in perfect pitch, providing a perfect voice-training assistant. The system design is shown in Figure 2-3. The record and playback mode will allow the user to begin recording, play notes then stop recording and then play back what was just recorded.

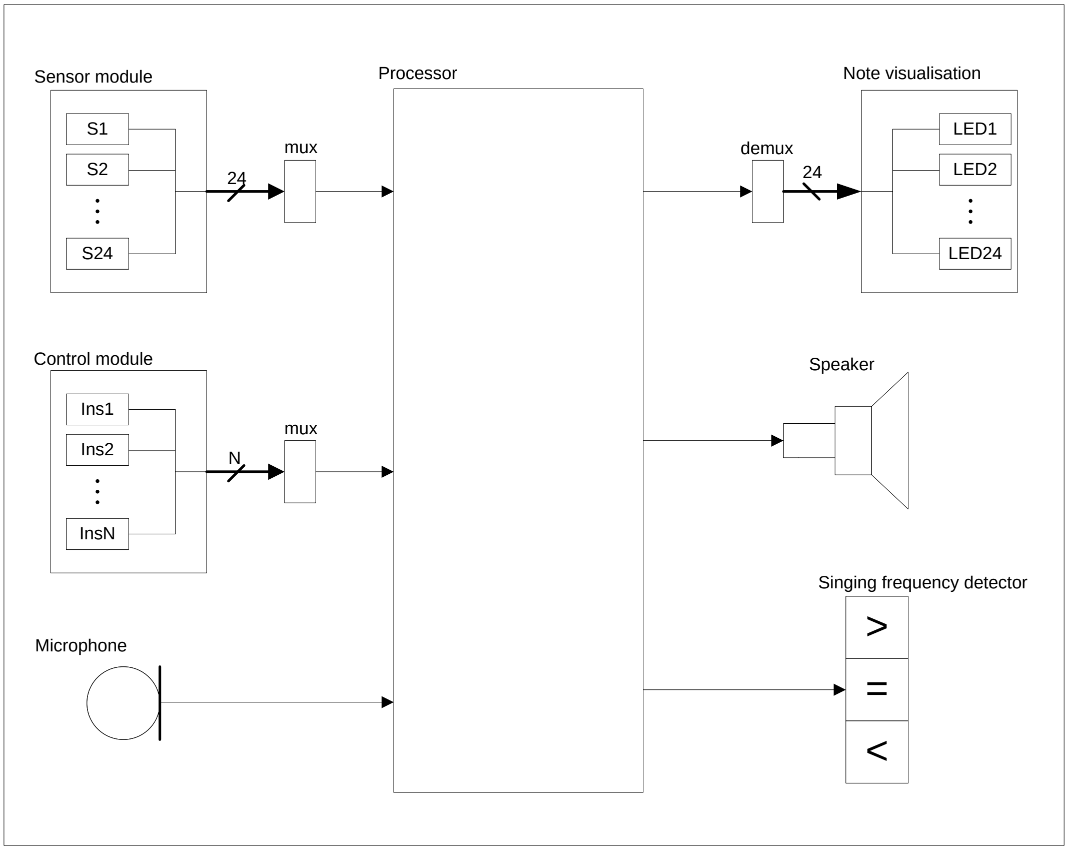

Figure 2-3. System Schematic diagram

The above diagram shows the main features of the product. The twelve sensors and corresponding twelve LEDs represent an octave of notes – white (Do Re Mi Fa So La Ti) and black keys on the piano. The control module is a set of pushbuttons located on the control board and used for selection of the instrument sound desired by the user. The microphone (when enabled) is used to sense the user's voice, which is compared to the note played and the result is the output from the singing frequency detection module. Power to the system is supplied through a regular AC outlet to avoid battery changing/recharging.

3. SENSING MODULE

The sensing module is implemented in form of the carpet. The outer frame, black and white keys have different thickness creating even more resemblance of the real instrument (see figure Figure 3-1).

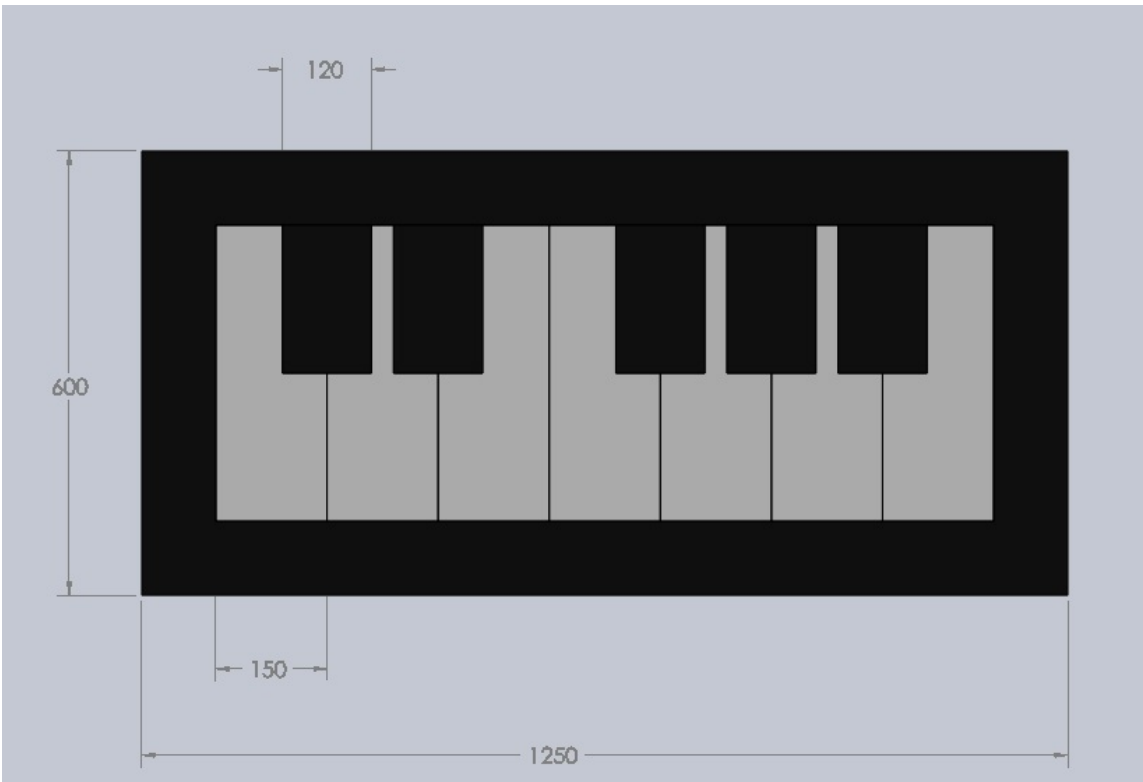

Figure 3-1. Top view of the carpet

The sensors layer is located under the keys section, thus the whole unit has three levels of thickness.

The contacts are implemented using shock-absorbing foam with apertures. Upon pressure application, the contact plates create the short circuit, bringing the voltage sent to the processor high or low, depending on the chosen configuration. Figure 3-2 shows the blow-up drawing of one of the keys.

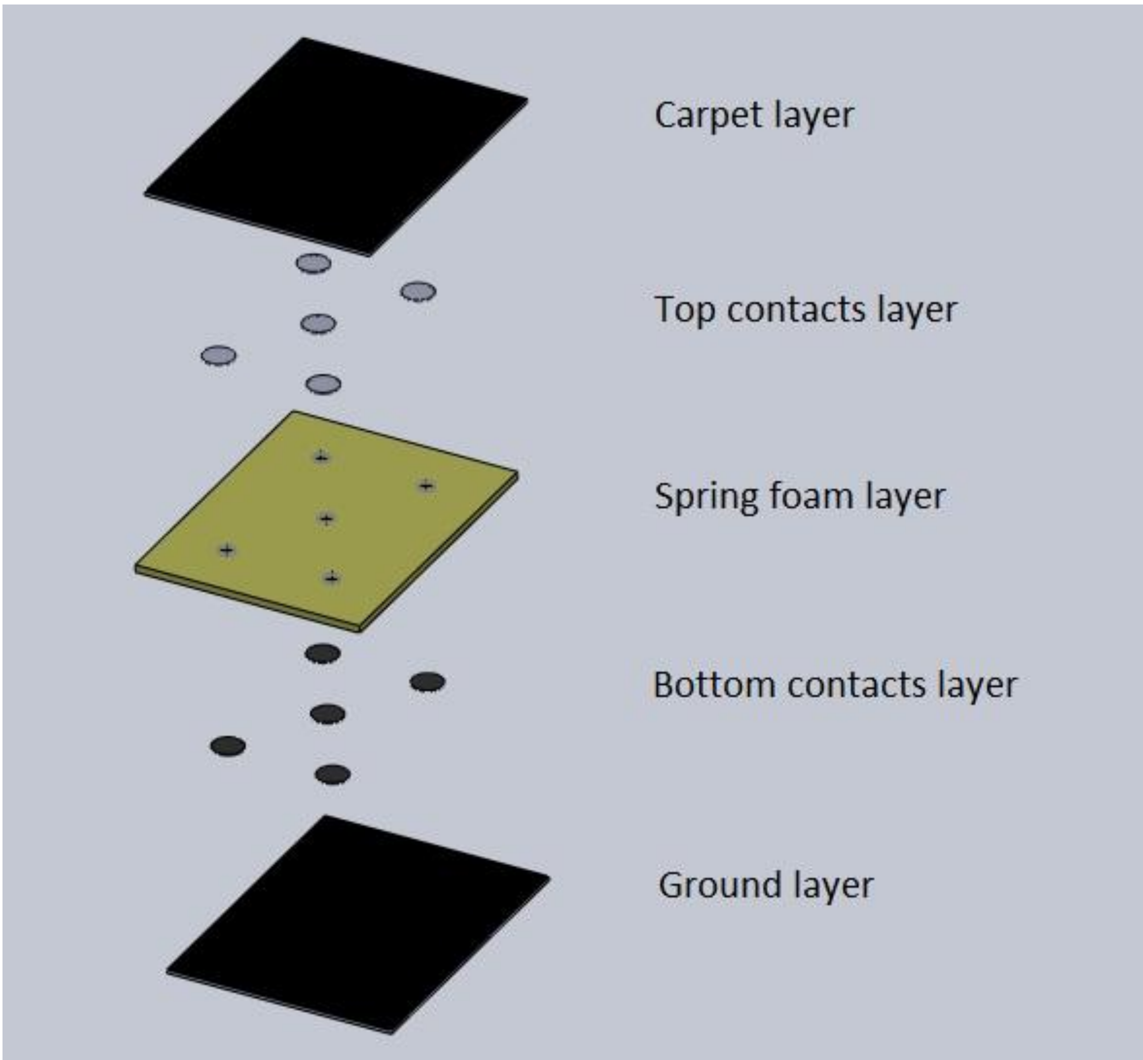

Figure 3-2. Assembly schematic of the single sensor

Since the range of user's weight is wide, the sensor has to be both easy to engage, rigid and durable.

4. MAIN CONTROL UNIT

The main control unit's primary responsibility is to sample sensors and microphone input and deliver control signals to the audio processing unit so that it plays the corresponding sound. This controlling unit consists of a microcontroller (MCU) with an internal ADC. The functions of the main control unit can be broken down into the following:

- Sample sensor switches using polling method
- Sample voice data from microphone to perform Fast Fourier Transform for pitch detection
- Communicate with Audio Processing Unit.
- Interaction with display unit
- Recording and playback feature

4.1 MICROCONTROLLER SELECTION

Atmel ATMega128 was considered for our application. The main control unit needs control 12 sensors, 18 LEDs and 4 buttons. The ATMega128 has 53 general purpose I/Os, which we can use for sampling sensors and controlling display unit. The high speed 10-bit ADC makes idea for sampling data from a microphone. The ATMega128 also features 16-bit timers with separate pre-scaler. It also features 4KB of internal SRAM, which is needed to store voice data samples from the ADC to perform FFT calculations. Atmega128 can achieve the maximum of 16 Million Instructions per Second (MIPS), which will give enough speed needed to perform fixed-point calculations.

Figure 10-1 shows the ATMega128 development board we will be using for our prototype. It has standard pins and JTAG, so it can easily be mounted on a breadboard for testing and debugging.

Figure 10-2 shows the pins on the ATMega128 Microcontroller and Table 4-1 summarizes functions for each I/O pin.

Port	Function
PB0 ~ PB7	Sensor 1-8 inputs
PC0 ~ PC3	Sensor 9-12 inputs
PA0~PA7	LEDs
PC4~PC7	LEDs
PD0~PD3	LEDs
PD4~PD7	Buttons
PF4~PF7(TCK, TMS, TDO, TDI)	JTAG Debug
PF0(ADC0)	Microphone input

Table 4-1. Functions for I/O pins.

4.2 SAMPLING SENSORS

The sensors described in Section 3 are basically simple switches. The Microcontroller will sample the sensors using a polling method, which will read one sensor at a time, then move on to the next sensor and it keeps reading in a loop. A flow chart of the sensor sampling is shown in Figure 4-1.

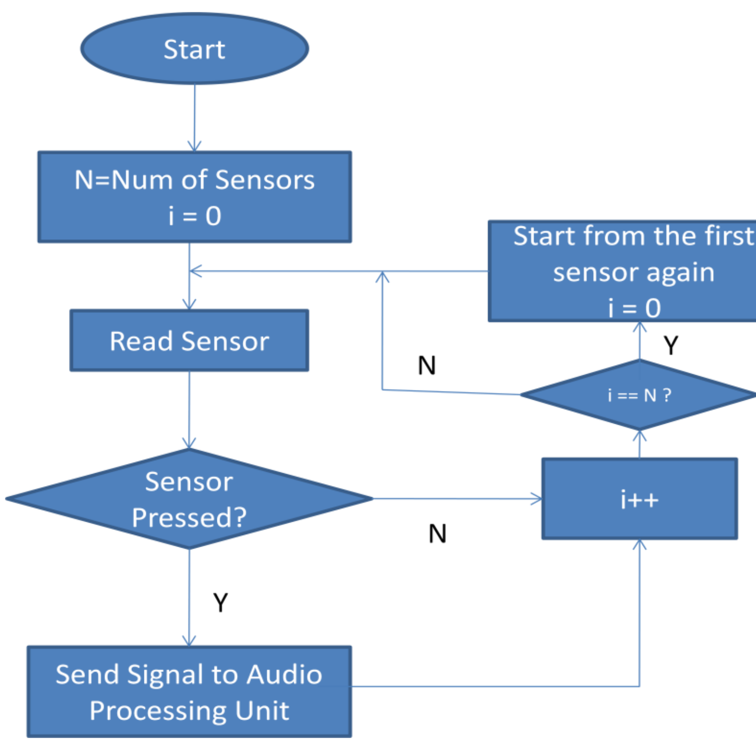

Figure 4-1. Algorithm for Sampling Data Inputs.

Due to mechanical properties of sensors, when the sensor is pressed, there is a period of time in which the electrical connection "bounces" between open and closed. However, the microcontroller will interpret this behavior as the sensor being pressed multiple times. To avoid this "bouncing" problem, a simple algorithm is introduced:

1. Read the sensor
2. If the sensor is pressed, wait for bounce time (From our testing, the bounce time is around 30 ms) and read the sensor again.
3. If the second reading also shows the sensor is pressed, then the sensor is pressed otherwise it is not being pressed.

4.3 RECORD/PLAYBACK

The Musical Carpet can record the pattern the user plays and can reproduce it later. An8-bit timer is used to measure the timing between two different notes being pressed. Figure 10-3 shows the 8-bit timer block diagram. The timer is set to run on FOSC/1024. The MCU is running at 16MHz, therefore the clock rate for the timer is 15KHz. Figure 4-2 shows the flow chart for the recording algorithm.

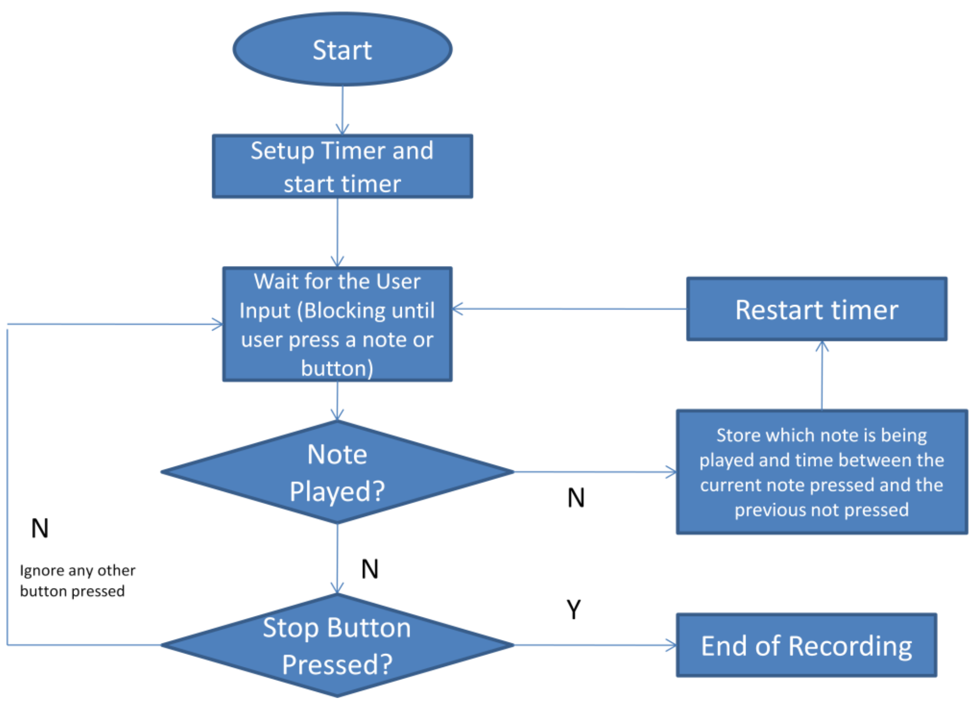

Figure 4-2. Flow Chart of Recording Algorithm

The playback would be the reversed operation of recording. There will be three types of data stored in the memory. The first type is the "note" data, which is the information of the note being played. The second type is "time" data, which is the information of the duration between two consecutive notes being played. The third type is to indicate that is end of recording. Figure 4-3 shows the flow chart for the recording Algorithm.

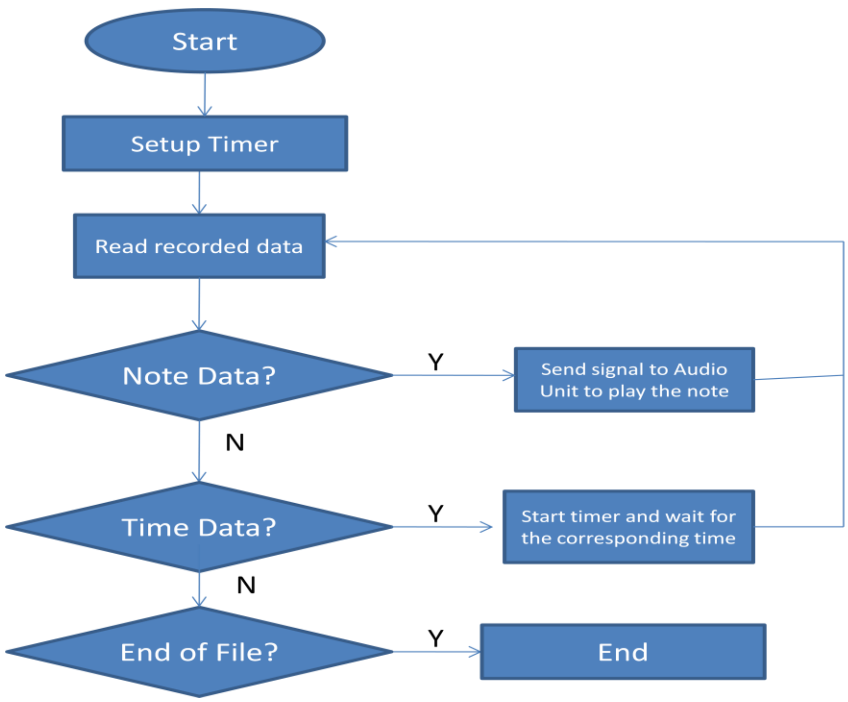

Figure 4-3. Flow Chart of Playback Algorithm

4.4 PITCH DETECTION

The pitch detection unit provides the vocal training functionality as described in the functional specification. This software-based unit compares a human voice input with a played note input and outputs a less than, equal to or greater than comparison.

4.4.1 ELECTRICAL DESIGN

A microphone input is connected to a pair of differential signal, analog-to-digital convertor (ADC) pins on the microcontroller. Three digital output pins on the microcontroller are used to provide feedback to the input. Each of the three pins is connected to the anode of an LED. The LED's cathode is connected to a resistor, which in turn is connected to the common ground. These components will be mounted into the control board as described in the appropriate section.

4.4.2 SOFTWAREDESIGN

The pitch detection unit is a sub-unit that is controlled by the main controlling unit and is located on the same microcontroller. The main controlling unit polls the ADC when in the vocal training mode. The ADC data as well as the last note to be played are provided as inputs to the pitch detection unit. The pitch detection unit processes the ADC data and compares it to the note played then returns the comparison. The main controlling unit then takes the results and turns on the appropriate LED. The main controlling unit can ignore the results provided by the pitch detection unit if another note has been pressed after the comparison has started but before it has finished. This is an unlikely scenario but is considered to avoid erroneous results.

The pitch detection unit itself consists of two modules, a fast Fourier transform (FFT) module and a pitch detection algorithm model. FFT is used as opposed to discrete Fourier transforms (DFT) due to computational ease. It is easily seen in Figure 4-4 that the FFT is faster than the DFT even when techniques are used to speed up the DFT. The FFT is also a weaker function of the number of data points used. The Fourier transform is used to covert the time domain data provided by the ADC to the frequency domain. This transformation is necessary because performing pitch detection in the frequency domain is faster.

The second module of the pitch detection unit is the pitch detection algorithm. The algorithm takes the transformed data finds the fundamental frequency (f_0). This can be found by taking the period of the harmonic frequencies and extrapolating back to the fundamental frequency, whether or not it is missing from the spectrum analysis. The calculated fundamental frequency of the inputted voice can be compared to the known fundamental frequency of the note that was played. This comparison is provided as the output of the pitch detection unit.

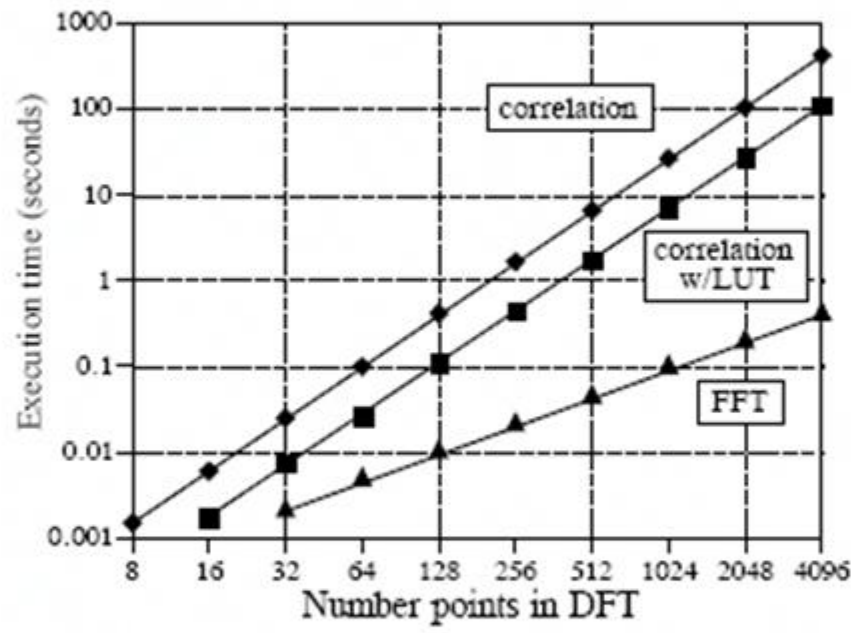

Figure 4-4. DFT and FFT Execution Times

4.4.3 LIMITATIONS

The main limitation of the system is due to the required amount of samples required as stated in the Nyquist-Shannon sampling theorem. The theorem requires a sampling period of $\frac{1}{2B}$ seconds for a maximum frequency of B. Given that the human voice has a maximum frequency of about 1.1KHz, the sampling period required is 4.545×10^{-4} seconds. The ADC provides 8-bit integers and using fixed point to give increased precision to the FFT calculations will require a minimum of 16-bit numbers. At 2KB of RAM, 512 samples can be taken. This means that only slightly over one fifth of a second can be analyzed at a time. At 4KB of RAM, a little less than half a second can be analyzed and at 8KB, over nine tenths of a second can be analyzed. This is only if that entire portion of RAM can be dedicated to the storage of samples. The actual calculations and comparison will require more RAM as will the other units that share the same microcontroller and will only lower the period that can be analyzed at once.

5. AUDIO PROCESSING UNIT

The main function of the Audio Processing Unit (APU) is to receive the signals provided by the Main Control Unit and to play the corresponding sound. The Audio Processing Unit will primarily consist of a microcontroller with an external digital to analog convertor (DAC), external SD card and two-stage voltage follower. Figure 5-1 shows the high-level block diagram for the Audio Processing Unit. All the pre-recorded sound samples are stored in the SD card. The MCU will access the data from the SD card through the SPI bus and it will send data to the DAC at a specified frequency. Then it will go through a two-stage voltage follower that acts as a high current operation amplifier (OpAmp), so it is enough to drive an 8 ohms speaker without external power.

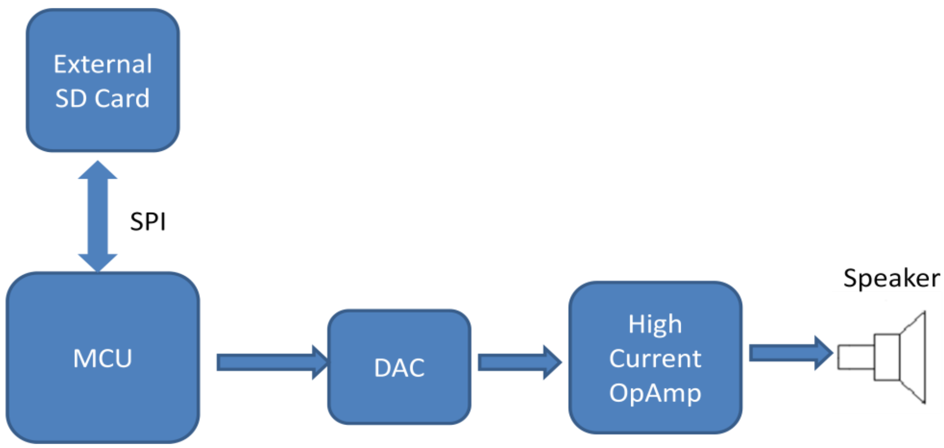

Figure 5-1. Audio Processing Unit High Level Block Diagram

The ATMega328 is chosen for this APU. The ATMega328 has almost the same features as ATMega128, but less I/O ports as the APU does not require many I/O pins to control the SD card and DAC. Figure 10-4 shows the pinout of ATMega328.Figure 5-2 shows the complete schematic for the APU.

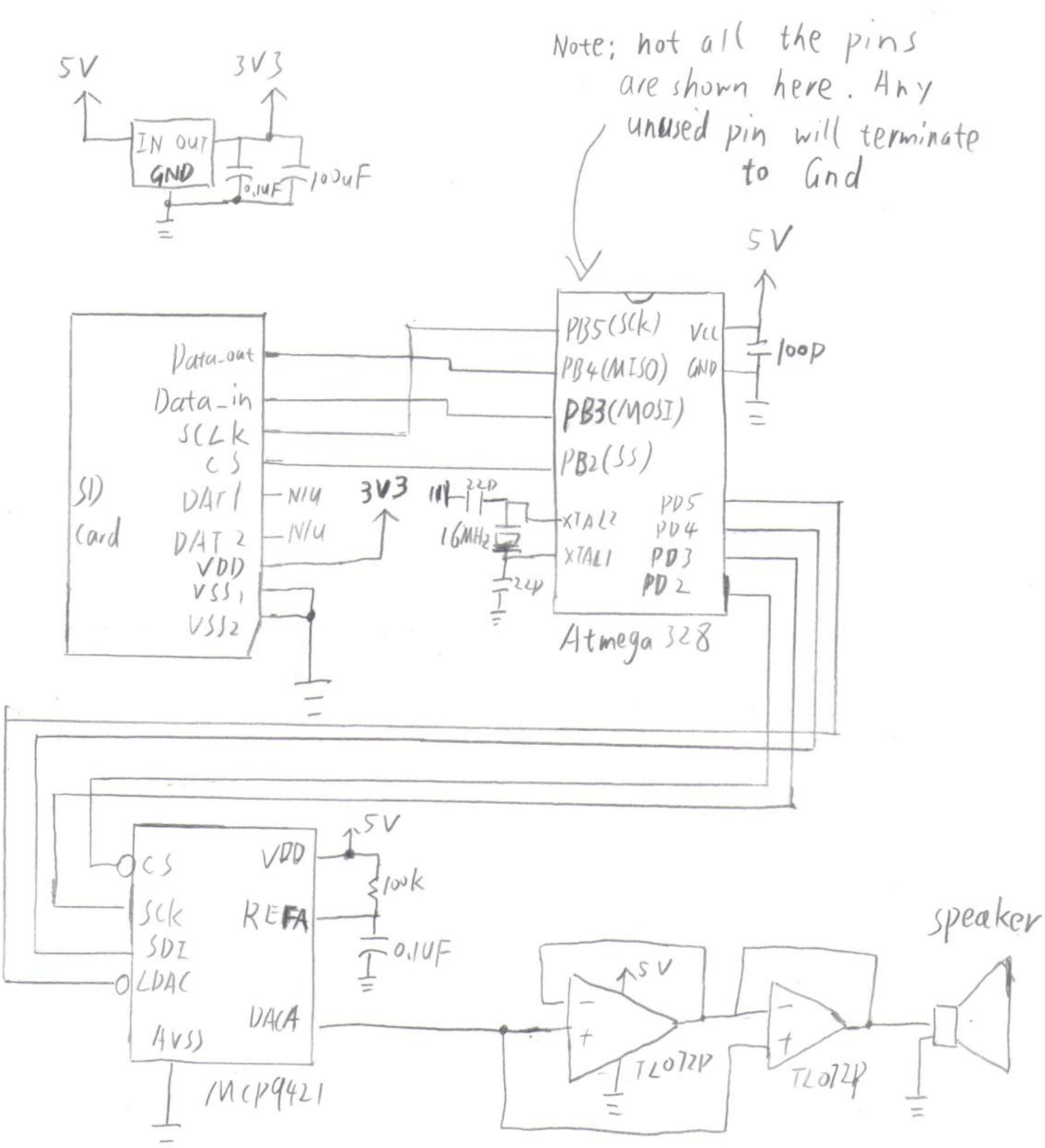

Figure 5-2. Design of Audio Processing Unit

5.1 MICROCONTROLLER TO SD CARD INTERFACE

The MCU communicates with the SD card through a high speed Serial Peripheral Interface (SPI). In SPI, devices are communicated in master/slave mode. The MCU will act as the master while the SD card will act as the slave in the APU. The master device (MCU in this case) initiates the data transaction. The SPI clock rate is configured to FOSC/4. The MCU is running at 16MHz. Therefore, the SPI clock rate is 4MHz.

The transmissions normally involve two shift registers in 8 bits size, one in the master device and other one in the slave device. The two registers are connected in a ring topology. The master device shifts out the most significant bit on the MOSI line while reading in the least significant bit on MISO line [11]. The slave device shifts out the most significant bit on the MISO line while reading in the least significant bit on the MOSI line [11]. Figure 5-3 clearly shows this operation between the master and slave device.

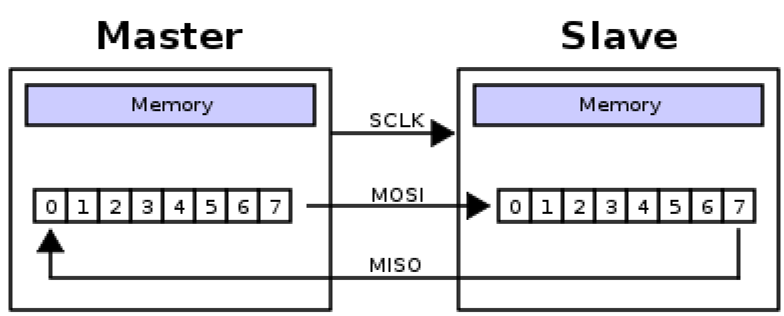

Figure 5-3. SPI Mater/Slave Shift Register [11]

5.2 DIGITAL ANALOG CONVERTER

MCP4921 is the perfect DAC for the Musical Carpet because of the following reasons:

- 12-bit resolution
- rail-to-rail output
- Selectable between unity and 2x Gain output
- SPI Interface with 20MHz Clock support

MCP4921 can be easily controlled by any four general purpose I/Os in the MCU. Figure 5-2 shows the connection between the MCU (ATMega328) and the DAC (MCP4921).

5.3 HIGH CURRENT AMPLIFIER

The amplifier stage of the APU is done using a two-stage voltage follower. This is on the bottom-right of Figure 5-2. A clearer version is shown in Figure 5-4. The part number for the OpAmp is TL072P. This OpAmp can provide up to 100mA. When connecting two of them in parallel, it can produce up to 200mA at 5V. Therefore, it provides enough power to drive an 8-ohm speaker.

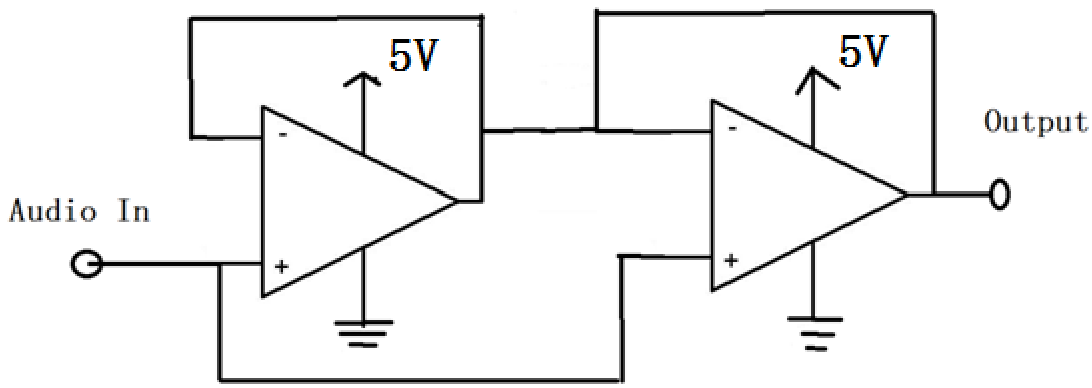

Figure 5-4. Two-Stages Voltage Follower (High Current Amplifier)

5.4 SOFTWARE DESIGN

All the sound samples are pre-recorded in WAV format and they are stored in the SD Card. When the APU receives a signal from the main control unit, it opens a wave file and first looks through the header section (first 44 bytes). The header section stores all sorts of information include channels (mono/stereo), bits-per-sample, sample rate. The detail wave file format is shown in Figure 10-5. The algorithm for playing wave file is done by using two timer-interrupts and a double buffer.

5.4.1 INTERRUPTS

Two interrupts will be used to play a wave file stored in the SD Card. The first interrupt is the timer interrupt. This interrupt will go off every $\frac{1}{samplerates}$ second. For example, if the frequency is 11KHz, then for every 1/11000 seconds, the timer interrupt picks a sample from the buffer and sends it to DAC. The second interrupt is used to fill up the buffer. Whenever the first interrupt reads to the end of the buffer, it starts the second interrupt, and then the second interrupt fills up the buffer by reading the SD Card.

However, the second interrupt will be running at a lower priority than the first interrupt. This way it won't affect the quality of the sound while filling buffer.

5.4.2 DOUBLE BUFFERING

To have a smooth playback, a double buffer is used. The size of each buffer is declared to be 256 bytes since the SD card can provide 512 bytes at a time. The timer interrupt starts with the first buffer. As soon as it reaches the end of the first buffer, it swaps to the second buffer and the second interrupt will fill up the first buffer in the mean time. The advantage of having two buffers is that the timer interrupt immediately has the backup buffer ready when it reaches to the end. This can avoid "stopping" during playback because of the time it takes to fill up the buffer.

5.4.3 COMPLETE ALGORITHM (ONE CHANNEL)

Figure 5-5 shows the interactions between the double buffer and interrupts and Figure 5-6 shows a flow chart of the wave file playback algorithm.

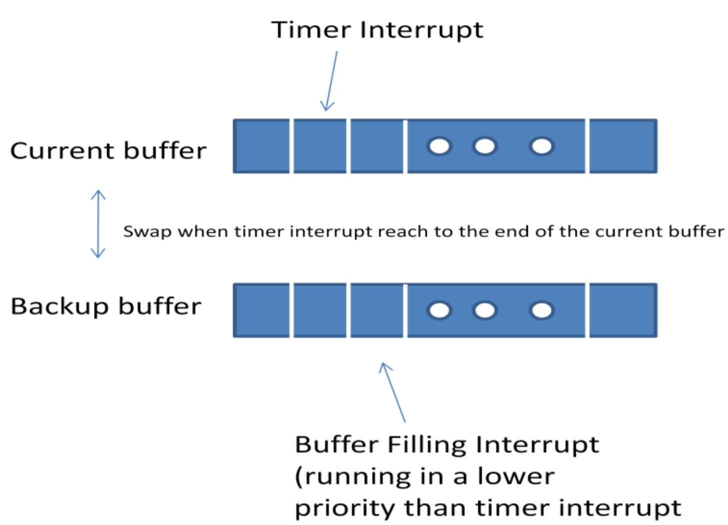

Figure 5-5. Interaction between buffers and interrupts

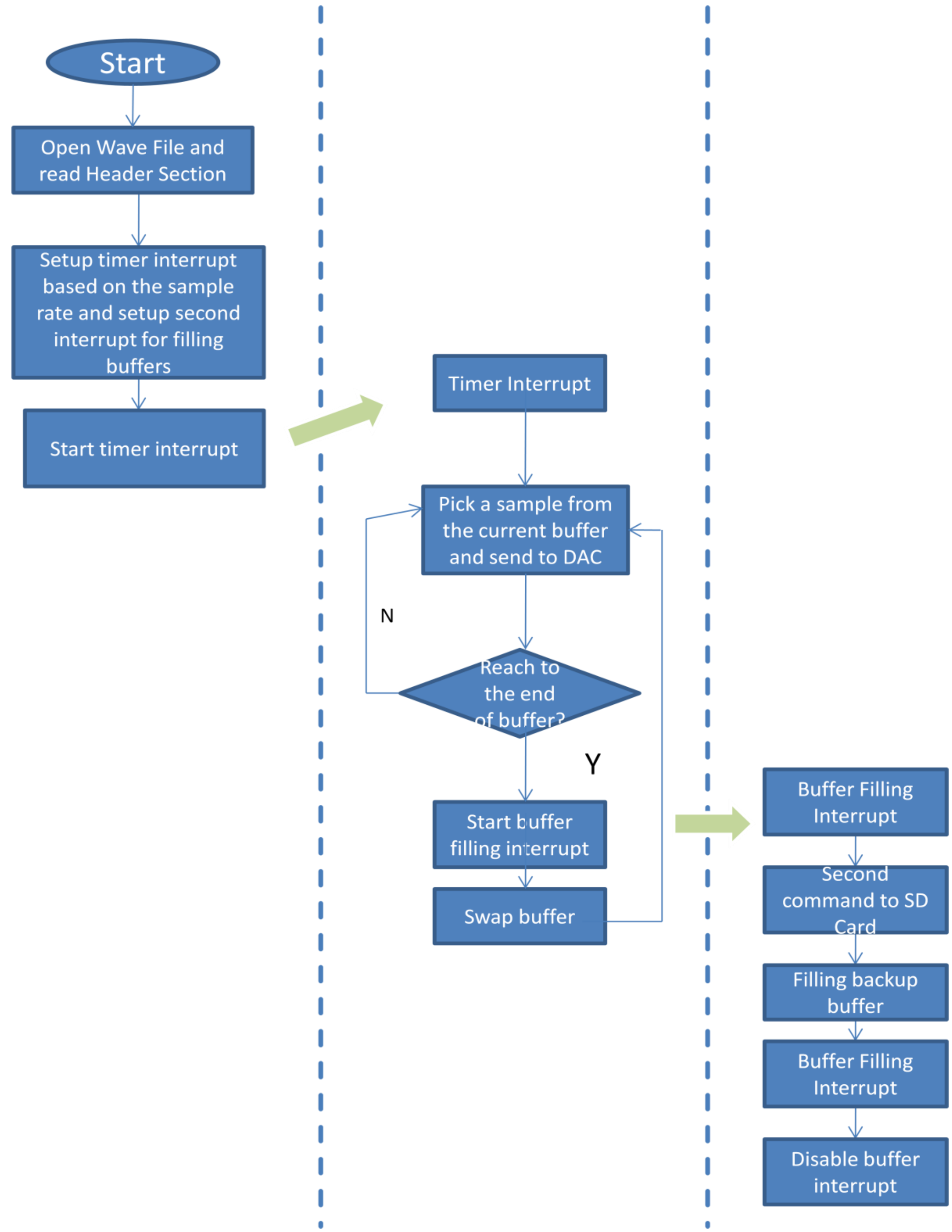

Figure 5-6. Flow Chart for Wave File Playback Algorithm

5.4.4 MULTI-CHANNEL AUDIO

Our audio processing unit should be capable of playing multiple wave files at the same time because it makes more sense when users press multiple notes altogether, multiple tones should be generated at the same time instead of one at a time.

For playing multiple wave files, the timer interrupt will pick each sample from the current playing waves and add them together. The adding is done in two's complement. Each 8-bit sample converts from 0~255 to -128~127. The result of summation will convert back to range 0~255 before it sends to DAC.

However, the number of channels we can achieve is limited by the hardware for the following reasons:

- Small capacity of internal SRAM
- Access speed to SD Card
- When many channels are playing at the same time, adding all samples together will lead to saturation.

In this project, we will aim for three channels.

6. MAIN CONTROL UNIT TO AUDIO PROCESSING UNIT INTERFACE

Figure 6-1 shows the connection between the Main Control Unit and the Audio Processing Unit. The GPIO pin in the Audio Processing Unit acts as the serial data input while the pin change interrupt (PCINT) pin acts as the serial clock input.

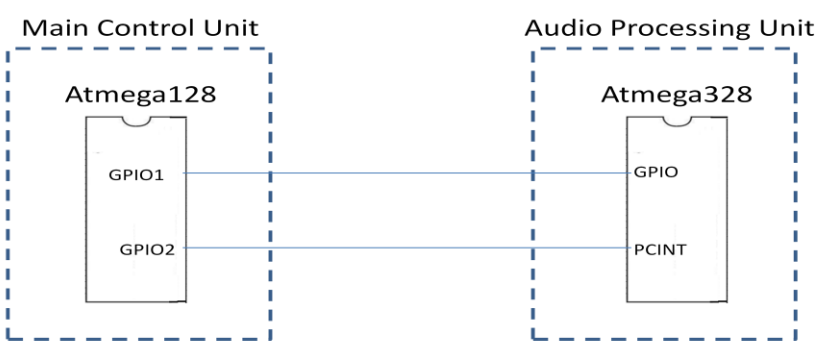

Figure 6-1. Main Control Unit to Audio Processing Unit Interface

The Main Control Unit changes the state of GPIO2 (either from low to high or high to low) to initiate the pin change interrupt handler in the Audio Processing Unit. The pin change interrupt handler will read-in the data from GPIO1 of the Main Control Unit.

Figure 6-2 shows the sequence when the Main Control Unit writes one byte of data to the Audio Processing.

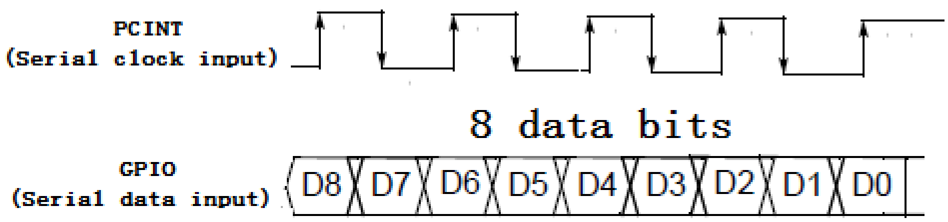

Figure 6-2. Write Command

7. CONTROL/DISPLAY MODULE

This unit is controlling and displaying the modes of operations. Several LEDs have been placed on this unit for the purposes of displaying, controlling, and giving feedback. Push buttons are also placed to control different modes. Figure 6-1 depicts the display unit.

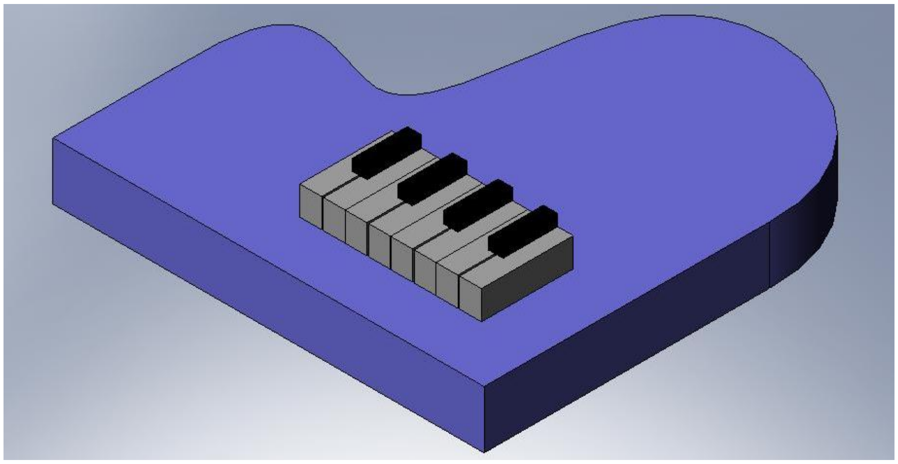

Figure 7-1. The Overall Design of Display Module, LEDs and Buttons are not Shown

7.1 LEDS CONNECTION

Sets of LEDs are put in place to show whether the corresponding function is working as proposed and to give our end user necessary feedback on the action they have performed. These LEDs are soldered to ribbon cables and the end connections from these ribbon cables are connected to our Microcontroller, which controls the on/off state of the LED. Since our microcontroller has internal resistors at the output pins, we do not need to use external resistors when connecting the LEDs to the output pins.

7.1.1 DISPLAY THE KEY PUSHED

The shape of one octave of piano keyboard is engraved on our display module covered by a transparent layer. Beneath this layer, LEDs are placed with proper distance to light the corresponding portion of transparent layer. These LEDs have a one to one relation with the keys pressed on our carpet piano.

7.1.2 LEDS CORRESPONDING TO PUSH BUTTONS

To show what mode is currently operational, a set of LEDs is placed next to each push button. As an example, a set of three LEDs is placed next to the instrument push button to show if we are in piano, guitar or organ mode.

7.2 PUSH BUTTONS

Normally open buttons are used for each control button. These buttons are connected to microcontroller input pins and are low when open and high when closed. These buttons control the instrument to be used and the modes of operation.

7.2.1 CHANGING THE INSTRUMENT

Using this button we can choose which instrument we are using. There are LEDs next to this button to show which instrument is currently chosen.

7.2.2 CHANGING THE MODES

This buttons is used to change the operating mode of our device. These modes are record and playback, vocal training, and normal operation. LEDs show the corresponding state.

8. SYSTEM TEST PLAN

To ensure the Musical Carpet is fully functional, we will thoroughly test each component, as well as the whole system with all the parts integrated. The component testing consists mainly of testing for the sensors, the main control unit and the audio processing unit.

8.1 SENSORS TESTING

The output of the sensors should simply be an on or off signal. We will first test the sensor under normal usage conditions. That is, the sensor will generate high/low signals when it is under a normal or compressed condition. Next, we will ensure the sensor is able to withstand the substantial force of a heavy adult without malfunctioning. After we run the same test for every sensor, we will combine all the sensors and perform similar tests as well as measure the power dissipation of the sensors.

8.2 MAIN CONTROL UNIT TESTING

The main control unit is to sample sensors inputs and deliver signals to the audio processing unit. The control unit acts as the interface module between the sensor module and the audio processing unit. Before the main control unit is connected to the sensor module or the audio processing unit, we will simulate sensor inputs using several pushbuttons and the control signals sent to the audio processing unit will be monitored using LEDs. For the microphone input, we will be feeding in a sinusoidal signal from the function generator. The microcontroller should be able to covert the signal to digital samples based on the magnitudes of the input signal. Then it will analyze the signal to provide user feedback.

8.3 AUDIO PROCESSING UNIT TESTING

The audio processing unit is the most challenging part of our system. The audio processing unit should be able to generate multiple sounds at the same time instead of simply playing one sound. Comprehensive combinations of sensor activations will be tested to ensure proper playback. Having accomplished this successfully, we will then need to process the voice of a singer and compare it with our sample within a pre-determined threshold. The result will be compared and shown, using the user interface.

8.4 INTEGRATED SYSTEM TESTING

Once everything is put together, it is time to test the final product. First, we will test the integrated product within our team. Then the final test would be done in public. We will let people come and play with our instrument so that we can spot any device malfunctions that we could not discover thus far.

9. CONCLUSION

The goal of the design specification is to provide guidelines to meet the requirements laid out in the functional specification. Development of the musical carpet will follow the technical specification proposed wherever possible. The test plans included in this document will serve to ensure correct adherence to the specifications and will ensure that we meet the goals we set out to achieve.

10. APPENDIX

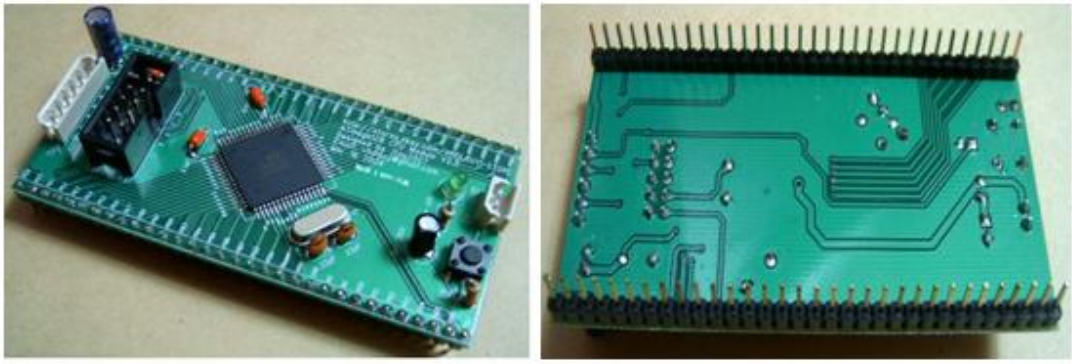

Figure 10-1. Atmega128 Development Board

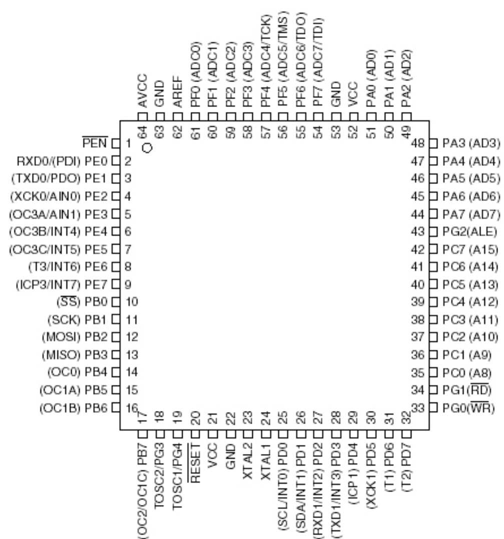

Figure 10-2. Pinout of Atmega128 [4]

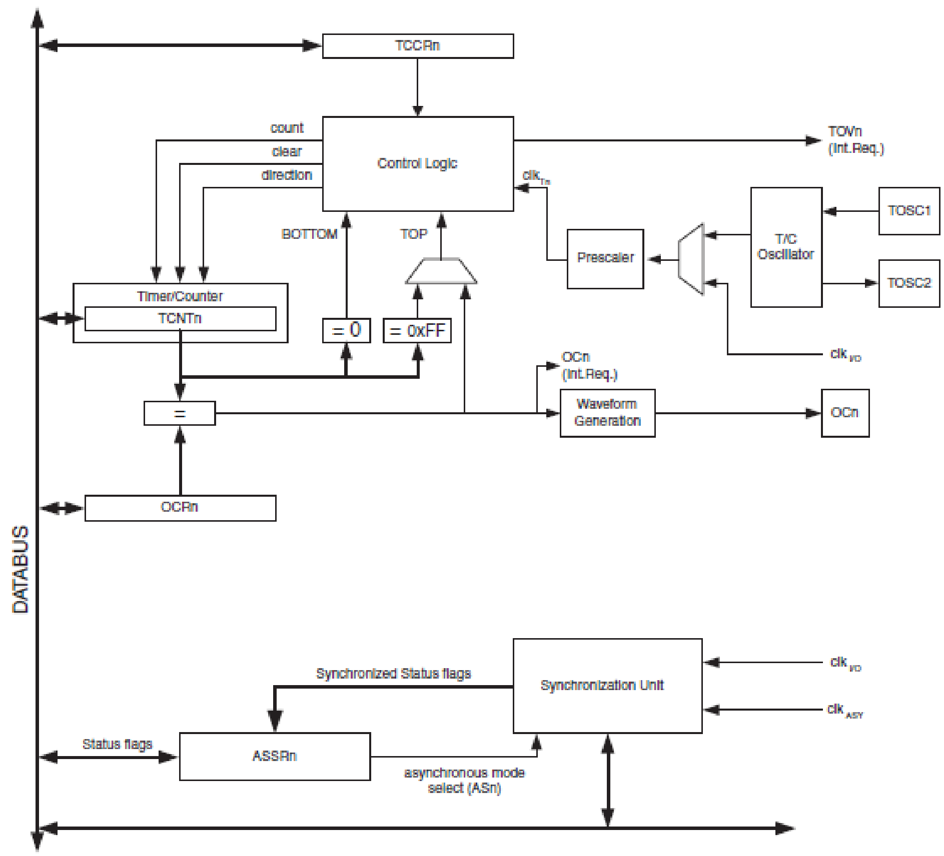

Figure 10-3. 8-bit Timer/Counter Block Diagram [5]

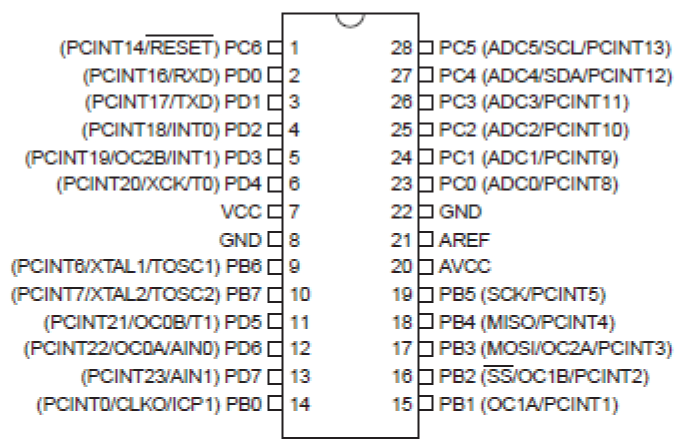

Figure 10-4. Pinout of Atmega328 [5]

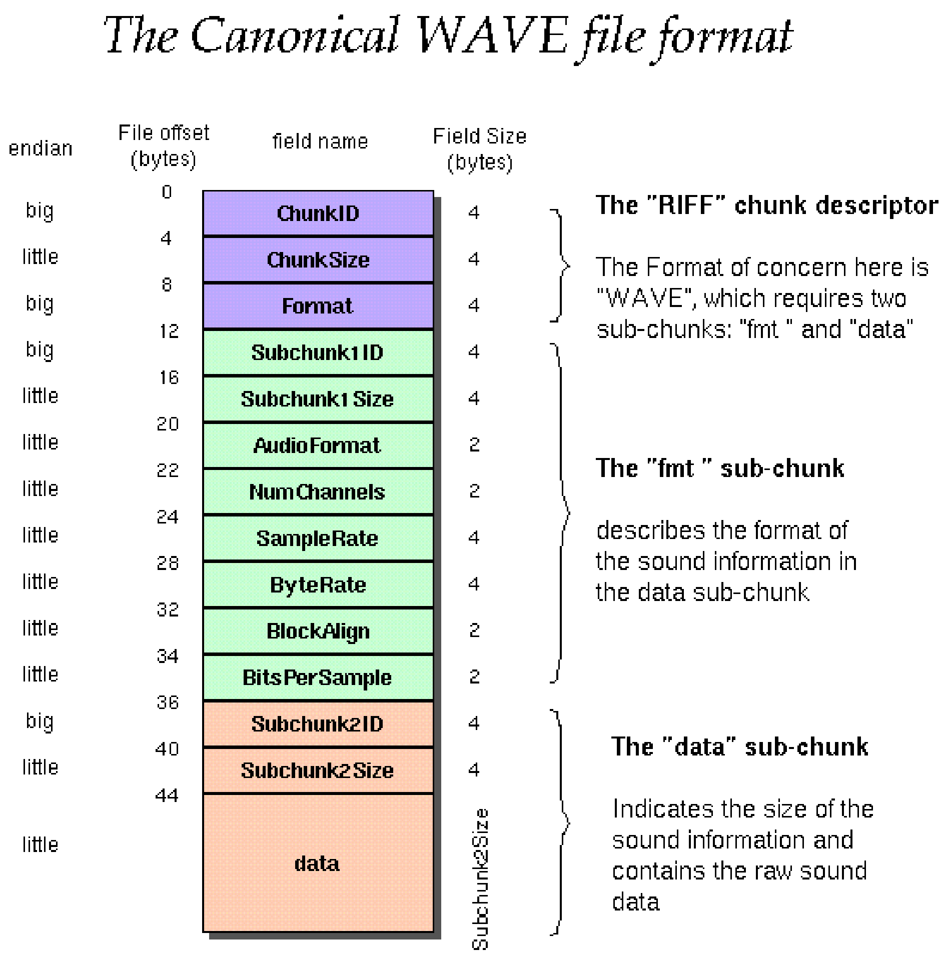

Figure 10-5. WAVE File Format [12]

11. SOURCES AND REFERENCES

[1]MicroController Pros Corporation, Accessed on Jan 13, 2010
http://www.microcontrollershop.com

[2] Digi-Key Corporation, Accessed on Jan 15, 2010
http://www.digikey.com

[3] Instructables, Accessed on Jan 18, 2010
http://www.instructables.com/id/Ghoul-Grabber/step1/The-step-sensor

[4] ATMega128 Datasheet, Accessed on Mar 4, 2010
http://www.atmel.com/dyn/resources/prod_documents/doc2467.pdf

[5] ATMega32 Datasheet, Accessed on Mar 4, 2010
http://www.atmel.com/dyn/resources/prod_documents/doc2503.pdf

[6] ATMega328 Datasheet, Accessed on Mar4 2010
http://www.atmel.com/dyn/resources/prod_documents/doc8025.pdf

[7] AGILE TEST GROUP, Accessed on Feb 3, 2010
http://www.agiletestgroup.com/site/

[8] Speed and Precision Comparisons, Accessed on March 3, 2010
http://www.dspguide.com/ch12/4.htm

[9] Arduino Board Schematic, Accessed on March 3, 2010
http://arduino.cc/en/uploads/Main/Arduino-Diecimila-schematic.pdf

[10] Wave Shield Schematic, Accessed on March 3, 2010
http://www.ladyada.net/media/wavshield/waveshield11schem.png

[11] Serial Peripheral Interface Bus, Accessed on March 4, 2010
http://en.wikipedia.org/wiki/Serial_Peripheral_Interface_Bus

[12] Wave File format, Accessed on March4, 2010
https://ccrma.stanford.edu/courses/422/projects/WaveFormat

[13] Speed and Precision Comparisons, Accessed March 3, 2010.
http://www.dspguide.com/ch12/4.htm

