TABLE OF CONTENTS

Abstract

The digital sheet music reader and player is primarily designed to serve as a preview device for musicians, musical instructors, or music students interested in hearing how a particular piece of music is intended to sound. The device is designed to be a portable stand alone desktop unit. Ideally it will set on a desktop next to a musicians playing station so that the musician can listen and compare as they organize parts or play the piece. Functionally, the user inputs image files to the device using a standard USB thumb drive. Image files can then be converted to MIDI and saved back on the USB drive or played on the device. The technical design aspects and professional components are discussed and outlined in the remainder of this report.

1.0 Project Overview and Block Diagram

The Digi-Sheet Music Reader and Player (DSMRP) is a portable music teaching aid. The purpose of this project is to provide musical students and teachers a way of hearing a score before or during practice of a piece. When people purchase sheet music there is often not an accompanying CD containing the recordings. In other cases the CD is lost and it can be hard to find the same version of the recordings. The DSMRP can give musicians the ability to still listen to that musical score that they are practicing.

The DSMRP is controlled by an ADSP-21262 SHARC DSP microcontroller which interfaces with the Winbond 64 Note Polyphonic Ring Tone Generator, Crystal Fontz LCD, Cypress SL811HS Integrated USB host, standard pushbuttons, and a rotary pulse generator.

Using a home PC setup with attached scanner, the user just has to scan the sheet music onto the PC and store it on an industry standard USB thumb drive. By plugging the drive into the DSMRP the user has the ability to convert any bit map image contained on that device into a playable MIDI file. In addition to conversion of an image file, the DSMRP allows the user to combine multiple scores and chose the type of instrument that will play each piece. After the user has defined their parameters the DSMRP will store the MIDI file onto the external USB drive. The user can then play the file on their home PC.

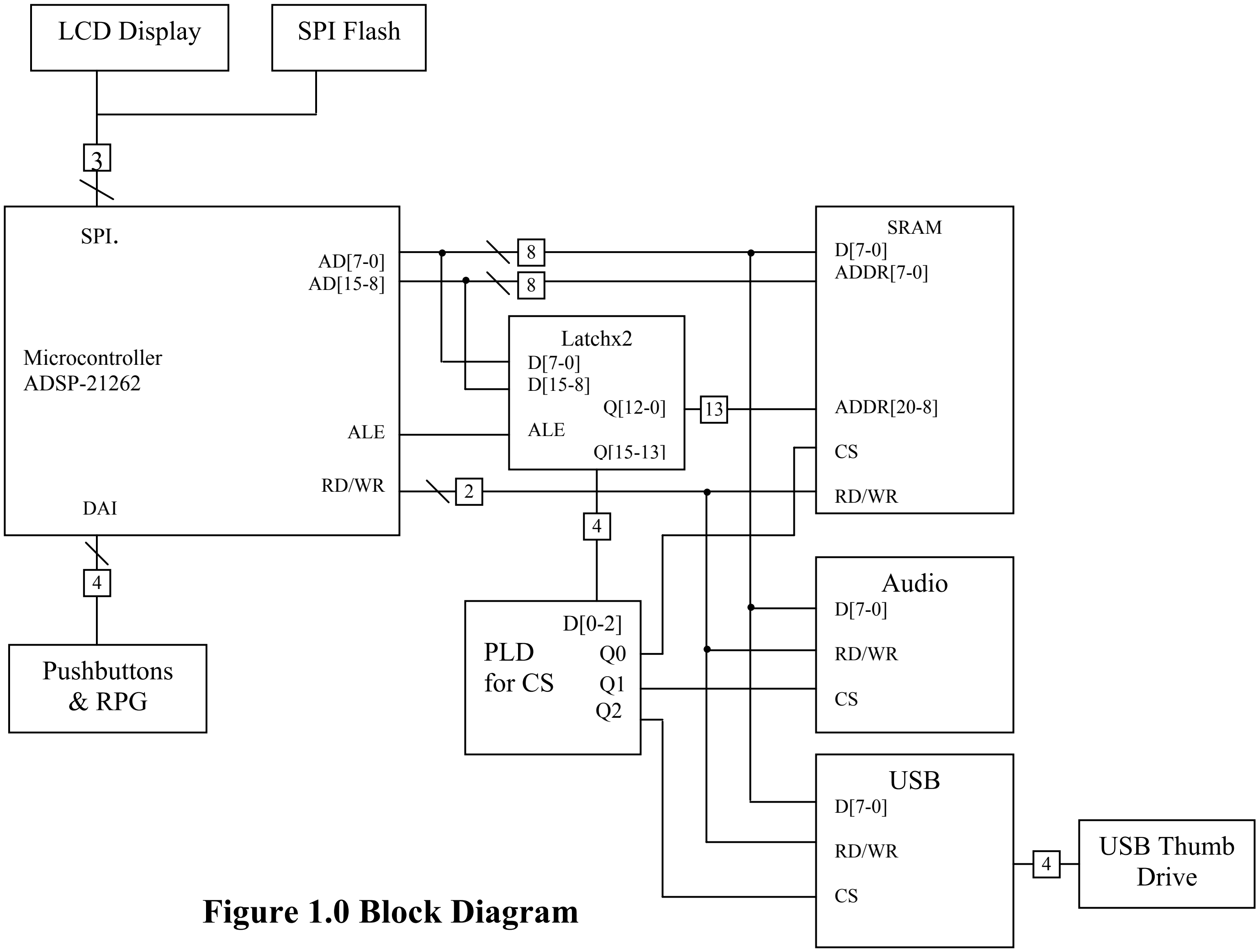


Figure 1.0 Block Diagram

2.0 Team Success Criteria and Fulfillment

1) An ability to parse an image file, to determine note type (quarter, half, whole) and position on staff given a clef and key, into intermediate musical symbols.

2) An ability to manipulate intermediate musical symbols into a MIDI file.

3) An ability to play a MIDI file through a MIDI audio processing chip.

4) An ability to act as a USB host and interface to a thumb drive.

5) An ability to display and navigate menu on a character LCD.

3.0 Constraint Analysis and Component Selection

3.1 Computation Requirements

We can expect that the processing of the image will require a good deal of computations, so it was important to choose a DSP that will be able to perform these calculations in a reasonable timeframe. Using an 8.5 x 11 inch piece of music scanned at 200 DPI gives us a 1700 by 2200 pixel image. A resolution of this size is required to accurately distinguish between different note types. A grayscale bitmap image of this size can be as large as 4MB while a black and white image is only around 500 KB. The external RAM chip will need to be large enough to hold the entire picture plus histograms while it's being processed. Based on current algorithms, it's estimated to take as many as 152 million simple arithmetic cycles to convert a grayscale image to black and white, detect the staffs, measures and filter out erroneous lines. More accurate results can be obtained by removing noise in the image using a Gaussian filter convoluted with the histogram of the image. A DSP with specialized commands for this is ideal.

3.2 Interface Requirements

The nature of this project is focused on software development concerning the image processing, thus the number of general-purpose I/O functions will be limited. We will require 8 general purpose I/O pins: 2 for a rotary pulse and 2 pins for pushbuttons, all required to navigate the menus. 2 more I/O's will control the reset pins of the USB host and MIDI chip while the last 2 I/O's will route interrupts from the USB and MIDI to the microprocessor.

The decision to use an 8-bit bus is derived from the choice of USB controller and audio chips, both of these chips require an 8-bit data bus. A multiplexed bus will require 19 pins[1] and will cut down the number of pins by as much as 36 pins[2]. The multiplexed bus will require a latch to hold the address lines while the data is being transferred. All pins connected to the address/data bus must be three-state so that there is no possibility of bus contention. Our design requires 3 chip selects for IC's on the parallel port and will need to be generated by a PLD if they are not generated by the microprocessor.

3.3 On-Chip Peripheral Requirements

- External Bus Interface: 8-bit data, address at least 8Mb, 3 Chip Selects available
- SPI/USART: 1 serial port for LCD communication.

- General Purpose I/O: A total of 8 I/O pins; 4 pins for rotary pulse generator and navigation buttons and 4 pins for reset and IRQ on the USB and MIDI chip.
- RAM: enough to store the image being processed, 600+KB would be ideal

3.4 Off-Chip Peripheral Requirements

This device will require a few peripherals outside of the DSP controller. The main components are the USB controller, memory, audio chip and LCD. All chips on the address/data bus will be 3.3V parts, thus a level translator will not be necessary. The LCD does use 5V TTL levels but will not require level translation of the SPI signals (data, clock and chip select) because the 3.3V bus drives the signal above the threshold for a 'high'.

To make the audio processing relatively easy, we chose to use the MIDI standard. MIDI will allow our device to choose between 128 instruments and play up to 16 different instruments at the same time. A separate audio chip that can process MIDI commands and is able to interface with an 8-bit parallel will be required. A chip with a small built-in amplifier as well as an external headphone output will eliminate the need for an external amplifier.

3.5 Power Constraints

This device is not intended to be hand held, so it will be powered using an A.C wall-wart. A regulated 5V wall-wart will provide power for the LCD and audio amplifier. Two separate LDO regulators will be used to obtain the 3.3V and 1.2V required by the rest of the circuit. By keeping the audio amplifier at 5V, the noise on the 3.3V and 1.2V power lines should be minimal. Current considerations are as follows: [1], [4], [5], [7]

5.0V: 160mA (LCD)

3.3V: 160mA (DSP) + 25mA (USB Host) + 200mA (Memory)
+ 20mA (Audio chip) = 405mA

1.2V: 500mA (DSP Core)

3.6 Cost Constraints

Currently there are no products like ours on the market, so we aren't competing with an existing product. The targeted consumer is primarily educators and music instructors but individuals would also benefit from the features of this device. A projected cost would be more than a simple CD player but the Digital Sheet Music Reader and Player provides much more functionality and a greater learning experience than simply playing one musical track. An initial retail cost is estimated around $90 but has the ability to be significantly reduced by optimizing two pieces of the project. Buying components in mass production quantities will reduce the cost per part, but the most obvious cost optimization would be the selection of the LCD. A serial CrystalFontz LCD was chosen for ease of development but a more basic parallel LCD would cost much less.

3.7 Component Selection Rationale

Microprocessor

There were many things to look at while considering a microprocessor, first and foremost was that we needed an external bus interface. This requirement adds many pins to the package, so we knew that we'd be working with a larger package. The nature of this project is more intensive on array and vector calculations because of the image processing. The optimization for convolution and histograms would be ideal and is integrated into many DSP architectures, while a RISC architecture would not be efficient at these calculations.

	ADSP-21262	AT91R40008	PIC24F	Constraints
Architecture	32-bit Floating-Point SHARC DSP	32-bit RISC ARM7	16-bit Microcontroller	DSP Not RISC
On-chip RAM	250KB RAM	256 KB RAM	8 KB RAM 128 KB Flash	200+ KB
External Bus	8/16-bit bus 4 flags/chip selects Multiplexed	8/16-bit bus 64 MB addressable 8 chip selects	8/16-bit bus 2 chip selects multiplexed	8-bit bus 3 chip selects
Serial Interface	1 SPI	2 USARTS	1 SPI & 1 UART while bus is enabled	2 SPI/UART
Package	144-LQFP	144-LQFP	80 & 100 LQFP	Non-BGA
Price	$19.26	$9.91	$5.71	

[1], [2], [3]

The PIC24F is a smaller package size but doesn't have enough on-chip RAM nor does it have enough chip selects. The AT91R40008 has a very limited instruction set, the external bus isn't multiplexed and 64MB of addressable RAM is overkill. The SHARC is the most appealing because it is a DSP and meets all other constraints of the project. It also has extensive development tools available, which will prove invaluable as this project is very code intensive. These reasons led us to choose the ADSP-21262 (SHARC) to power the Digital Sheet Music Reader and Player.

Memory

Our device will operate such that the data on the external memory will not need to be saved once the device is turned off. The DSP will copy the image from the thumb drive onto the memory chip. It will then process the image and store that processed information back on the thumb drive to save for later. This means that we can use a run-time SRAM memory instead of a storage Flash memory. The SRAM should be arranged in 8 bit words to match the 8-bit data bus. 3.3V SRAM chips above 16Mb are very rare unless they use words larger than 8 bits. So I found a few chips with the following specifications: SRAM, 16Mb (2Mx8), 3.3V, parallel interface.

Cypress: CY7C1069AV33 (36-pin SOJ or 44-pin TSOP II)

Renesas: R1RW0408D (36-pin SOJ)

The preferred package type is the TSOP package and it's relatively easy to obtain samples of Cypress parts. So we choose to use the Cypress CY7C1069AV33. [5], [6]

Audio Chip

Once MIDI was chosen as the method of audio playback we began looking for chips that would interface with an 8-bit data bus, accept MIDI commands and output an audio signal. All of our options came from the cell phone market:

Winbond W56964: 64-Note Polyphonic ringtone chip, 32-pin QFN

Fangtek FT1770N: Compatible with MP3 and MIDI, 32-pin LPCC

NEC uPD9992: Ring Tone Generator LSI, 65-pin FBGA

All three chips perform very similar functions so the main distinguishing factors are the package types and documentation available. Most chips had limited documentation, but Winbond was quick to respond via email and supplied us with five sample chips, an evaluation board, numerous documents, and sample code. The QFN package of the Winbond was a challenge to solder, but it is our chip of choice. [7]

4.0 Patent Liability Analysis

4.1 Results of Patent and Product Search

The Digi-Sheet Music Reader and Player (DSMRP) may infringe on a patent for a music interpreter apparatus, filed under patent number 5,202,526 on December 17, 1991. It was assigned to Casio Computer Company based in Tokyo, Japan. The device described in the patent is intended as a musical interpretation device. It would scan a piece of written music for musical cues converted from standard written notes and symbols to a "string of coded music notational symbols" [1]. These notation symbols are intended to be more complex than the standard midi file format saying that the "notation symbol string describes composer's instructions in terms of music notation" [1]. A "musical interpreter" within the system is designed in a hierarchical control structure where each aspect of a note is handled by a separate portion of the interpreter. So, loudness of sound could be handled by a section that compares all notes around it, while the pitch of the note is handled by a different section that only deals with the written markings for that note [1]. The DSMRP infringes on the device in the patent in one way. This infringement is the scanning of a musical image into coded symbols.

The second patent describing an apparatus similar to the DSMRP device was filed under patent number 5,825,905 on September 28, 1994, to Yamaha Corporation. It describes an apparatus that is used to display the image of a written piece of music as well as an interpreted version of the written image side by side. An interpreted version is described as an image where each note was scanned off of the original document, converted to "score code data", and an associated symbol displayed on the interpreted version of the written image [2]. If the symbol on the original piece of music does not match the one in the interpreted version, the device would allow for user input to change the interpreted symbol, changing the "score code data" [2]. The DSMRP device may infringe on the patent's claims to allow user input to change aspects of the music that was interpreted and the conversion of the scanned music into what the patent refers to as "score code data" [2].

A patent for a device that the DSMRP substantially infringes on can be found under patent application number 20060150803, which was filed July 15, 2005. This handheld or desktop device interprets a paper copy of a musical score and processes the aspects of the music such as notes, level of sound, pace of music, etc. into MIDI format. The device can also accept pre-scanned images from wireless and wired networks as well as USB memory devices such as a flash drive. These midi files are then played in the form of synthesized sound via a speaker or headphone. The whole device may be contained in a "handheld device" [3]. The DSMRP infringes on the claim referring to conversion of a sheet music image stored on a USB flash drive into a MIDI format and playing that MIDI information. Furthermore, the DSMRP will be packaged as a portable device, which may conflict with the patent's reference to a "handheld device" [3]. Also, the DSMRP may incorporate the ability to choose which instruments play during an audio output session, in the future. This may conflict with the "music minus one" feature claimed in the patent [3]. In essence, every part of the DSMRP will conflict on some claim within the patent.

4.2 Analysis of Patent Liability

Based on the doctrine of equivalents, the device infringes on both patents 5,825,905 and 5,202,526 because of the ability to convert sheet music to a set of symbols that the DSMRP can interpret [1-2]. The production of the DSMRP is limited to the capabilities of it to convert a written piece of sheet music to a computer readable symbol referred to as an Intermediate Music File, or IMF. This file is intended to have enough information for a microprocessor to produce a complete MIDI file and an audio chip to play the file. The processes the DSMRP uses to interpret the music file can be found in many published scholarly journals such as [4]. In fact, much research has been done on the topic of Optical Music Recognition (OMR) [4-5]. In the patent assigned to Casio Computer Company, the writer implies that the MIDI language is too primitive for the apparatus' intended function. The writer states that, "these electronic musical instruments do not handle musical interpretation" [1]. The device that is being created by Casio will produce symbols that have much more meaning than what the IMS will do and is not intended to be played through MIDI format, where as the DSMRP's program symbols are written with the sole intention of being played through MIDI format. Based on these facts it can be concluded that the idea of converting a scanned image into a computer readable symbol format does not infringe on Casio's symbol format because these two symbol formats do not perform the same function. The patent filed for Yamaha states that the created symbols would be used for playing of music utilizing the MIDI format [2]. The DSMRP creation of intermediate symbols with the intention of being converted to MIDI format is an obvious reason for creating these symbols. What needs to be addressed is not the idea of creating symbols from a piece of written music that will be then converted into MIDI commands, but the symbols themselves. The device will create computer readable symbols that are not the same as those used by Yamaha in this patent. Hence, the device does not conflict with the either of the patent's claims of parsing a piece of written sheet music into computer readable symbols.

Patent number 5,825,905 filed by Yamaha claims the ability to edit musical symbols that are created from a scanned image [2]. This process involves displaying an interpreted image of the scanned original document and allowing the user to edit the interpreted symbols by means of an LCD and buttons. The primary function of the LCD and buttons are to allow a user to change the symbols that were already created by the apparatus. The device will also have buttons and an LCD on it. They are present to allow the user to add extra information to the piece. This information could be the clef or speed of the piece. Upon closer examination, the DSMRP does not conflict with the present patent because the patent claims only to edit previously created symbols through the use of an screen and buttons, where as the LCD and buttons will be used for creation of new symbols that will aid in the playing of the device.

Finally, the DSMRP described in patent application 20060150803 claims many features the DSMRP will be incorporating. Again, the DSMRP will be able to convert a file, read from a USB flash drive, into an IMF. A MIDI file will then be created based on the IMF and the user input information via an LCD and buttons. This MIDI file will then be sent to an audio processor, which will play the MIDI file in the form of audible sound though a speaker or headphones. One claim within the patent refers to the processing of a scanned piece of written music taken off either a network or flash device [3]. Clearly, the DSMRP performs the same function as that of the device stated in the patent. Yet, it has already been established that the processing of an image into a computer readable file format can be found in research done prior to the publishing of the patent and therefore we cannot be sued on these grounds. Furthermore, the makers claim the "producing [of] an audio rendition of the music score data … by means of a recording medium inserted into a reader incorporated into the computing device" [3]. The idea of processing an image off of an external recording device can be found throughout the computer industry. The claim is invalid because images are loaded from a memory device onto a computer and edited in standard software before this patent was filed. The patent next claims that the apparatus will be hand held [3]. On the other hand, the DSMRP is intended to be “portable”. When referring to the device being portable it is implied that it would be easy to carry the device from place to place, but not so small that it could be held in a person’s hand or easily attached via belt clip. To see the distinction between these two concepts it is helpful to look at the difference between a PDA and a laptop. Both are devices that perform similar functions, but a laptop can not be held in one’s hand based on the consumer industry’s implied definition of handheld. As a result it is classified a portable device. On the other hand a PDA fits in a person’s hand easily and is classified as handheld. Therefore, the DSMRP does not conflict with this claim in the patent because they claim that the device can be handheld, where as the device is only portable. Given that the DSMRP will be able to change instruments that will be playing a musical piece in an audio session, it will be infringing on the “music minus one” feature claimed in the patent [3]. The “music minus one” feature allows the user to remove parts (instruments) from being played during an audio session. Given that a piece of music has multiple parts on it when it is scanned, it is hard to argue that the DSMRP does not conflict with the patent’s claim. Yet, MIDI standard integrates the “music minus one” feature. While most of the claims within the patent are proven to not conflict with the device when looked at individually, it is difficult to prove that all the claims making up the device is either obvious to anyone in the field or is not a novel idea. The conclusion is that the device does conflict with the patent’s claims and it would be hard to prove that the device is different, as a whole, from the patent’s device.

4.3 Action Recommended

While the patents filed for Yamaha and Casio contain features conflicting with the device, the features pertaining to the device were developed prior to the filing of the patents [4-5]. On the other hand, the last patent filed July 15, 2005 describes a device that, while containing features that are not new or novel individually, is as a whole a novel device. It would not be obvious to create a handheld device that scans and processes music as well as plays it in the form of synthesized audio [3]. Therefore, I feel that further research must be done to either find a portable device that processes an image of sheet music and plays it in the form of audible sound, or find a research publication that mentions such a device prior to the filing of this patent to prove that the apparatus in the patent is not novel. Another option is to purchase the patent from the owners or attempt to form a contract with the filers whereupon we pay them a royalty fee for production and selling the device. Finally, we can hope that the patent application never becomes a patent. Otherwise it is unlikely that it can be proven that the device does not conflict with the patent and we would most likely be prevented from producing the device.

5.0 Reliability and Safety Analysis .

5.1 Reliability Analysis

The overall success and reliability of the digital sheet music reader and player depends upon the successful interactions of all of the components in the device. If any single one of these components fails, the device as a whole will fail. Failure includes not only a physical device failure, but also a failure to perform the needed conversions and calculations as intended. The digital sheet music reader has four components that have been identified as the most likely to fail: 1. ADSP- 21262 SHARC Microcontroller, 2. KA278R05C Fairchild 5V and 3.3V LDO Regulators, 3. W56964 MIDI Chip, 4. SL811HS USB Controller. These components were selected because of their complexity and possibility of operating at temperatures greater than ambient.

The Military Handbook for Probability Prediction of Electronic Equipment [1] was used to perform a calculation of the number of failures per 10^6 hours, λ_p, and the Mean Time to Failure (MTTF), $1/\lambda_p$, for each component. The operating environment of all of the components was assumed to be ground benign because the device will always be operated inside in a non-mobile manner and in a temperature controlled environment. Also the quality factor for each component was taken to be 10, because the amount of screening was unknown. Definitions of the parameters used for each model are explained in Table 2.1. Specific assumptions for the separate components are elaborated upon in each section.

Parameter	Explanation
C_1	Die Complexity Failure Rate
C_2	Package failure rate
N_P	Number of functional pins
π_T	Temperature factor
π_E	Environmental factor
π_Q	Quality Factor
π_L	Learning Factor

Table 5.1 λ_p Parameters

5.1.1 ADSP- 21262 SHARC Microcontroller [2]

$$\lambda_p = (C_1\pi_T + C_2\pi_E) \times \pi_Q \times \pi_L \textbf{ failures per } \mathbf{10^6} \textbf{ hours}$$

Parameter	Value	Justification
C_1	0.56	32 bit MOS microprocessor
C_2	0.0771	144-pin nonhermetic SMT package $C_2 = 3.6 \times 10^{-4}(N_p)^{1.08}$, $N_p = 144$
π_T	3.1	Digital MOS device, maximum junction temperature under bias, $T_J = 125°$ C
π_E	0.50	Assuming ground benign environment, G_B
π_Q	10	Assuming unknown screening level
π_L	1.0	Generic device in production for ≥ 2 years

λ_p = 18.52 failures per 10^6 hours
MTTF = 54004.4 hours $\approx$ 6.165 years

The SHARC microcontroller could be viewed as the brain of the digital sheet music reader and player. The SHARC is used to calculate the various signal processing calculations as well as to communicate to the USB controller, LCD, Audio chip, and external memory. All of these constraints considered, the reliability of the device relies heavily on the successful operation of the SHARC. The 32-bit processor on the SHARC drives the failures per 10^6 hours up to 18.53 and MTTF down to 6.165 years. There is an obvious trade off for this component in that a much smaller and simpler microcontroller could be used for the component integration, however due to the aggressive signal processing, a simpler chip would likely be loaded down and perform much above normal operating temperature which would likely cause more failure. So despite the fact that the more complex processor has a shorter MTTF, it will be operating at much more normal temperatures despite the signal processing. Fortunately, while the failure of the microcontroller would render the digital sheet music reader and player unusable, it is of low criticality because no harm could be done to the user in the event of failure. Along with the fact that failures of the SHARC are repairable, the higher rate of failure per 10^6 hours is acceptable for this component.

5.1.2 KA278R05C Fairchild 5V and 3.3V LDO Regulators [3]

$$\lambda_p = (C_1\pi_T + C_2\pi_E) \times \pi_Q \times \pi_L \textbf{ failures per } \mathbf{10^6} \textbf{ hours}$$

Parameter	Value	Justification
C_1	0.01	Assume 1-100 transistors
C_2	0.0016	4 pin TO-220F full plastic mold package, assume equivalent to DIP package $C_2 = 3.6 \times 10^{-4} \left(N_p\right)^{1.08}$, $N_p = 4$
π_T	180	Linear MOS device, maximum junction temperature, $T_J = 150°$ C
π_E	0.5	Assuming ground benign environment, G_B
π_Q	10	Assuming unknown screening level
π_L	1.0	Generic device in production for ≥ 2 years

λ_p = 18.01 failures per 10^6 hours
MTTF = 55530.9 hours $\approx$ 6.339 years

The LDO regulators were included in the reliability analysis because by the very natural of the component, they operate at temperatures above normal. This capacity to fail only increases as the input voltage increases and the regulators have to dissipate more heat. Assuming no more than 100 transistors that the TO-220 full plastic mold package is closer to a DIP package than it is to a leaded can package, the failure rate is 18.01 failures per 10^6 hours. Unfortunately in the case that the regulator fails by causing a short to ground and leads to over heating of the device, the criticality is high because this poses a serious danger to the user. With this criticality level, the rate of 18.01 failures per 10^6 hours is unacceptable. In the opposite event, however, the regulator simply fails to produce any voltage or a voltage out of tolerance and the criticality is low as nothing will happen in the device and the user will not be in any danger

5.1.3 W56964 Winbond MIDI Chip [4]

$$\lambda_p = \left(C_1 \pi_T + C_2 \pi_E\right) \times \pi_Q \times \pi_L \text{ failures per } 10^6 \text{ hours}$$

Parameter	Value	Justification
C_1	0.29	Assume Digital MOS device with 30,001 to 60,000 gates
C_2	0.0152	32-pin SMT package, assume nonhermetic $C_2 = 3.6 \times 10^{-4} \left(N_p\right)^{1.08}$, $N_p = 32$
π_T	1.5	Digital MOS device, Assume max junction temperature, $T_J = 100°$ C
π_E	0.50	Assume ground benign environment, G_B
π_Q	10	Assume unknown screening level
π_L	1.0	Generic device in production for ≥ 2 years

λ_p = 4.426 failures per 10^6 hours
MTTF = 225937.6 hours $\approx$ 25.8 years

The Winbond audio chip is the final step in the signal processing of the digital sheet music reader and player. Once the image file has been converted to MIDI the Winbond chip actually processes and plays the MIDI file using wave table synthesis. The primary reason the Winbond chip was included in the reliability analysis was its processing complexity and timing critical operation. It was assumed that the chip was the most complex digital MOS device possible with up to 60,000 gates. This is a worst case assumption, because no information was available for the actual die complexity of the Winbond chip. Also the maximum junction temperature was assumed to be 100° C which is well above the maximum operating temperature and no actual junction temperature data was available. While these assumptions would be unsuitable for actual product manufacturing, they are worst case scenarios and are acceptable for this analysis. The MTTF of the Winbond chip is 25.8 years which corresponds to 4.426 failures per 10^6 hours. This failure rate is more than acceptable seeing as how failures are of low criticality because the effects only result in device malfunction and could not cause personal harm to the user. Along with the fact that no harm can be done to the user, a failure could be replaced and thus drives the criticality of this failure a bit lower.

5.1.4 SL811HS Cypress USB Controller [5]

$$\lambda_p = \left(C_1 \pi_T + C_2 \pi_E\right) \times \pi_Q \times \pi_L \text{ failures per } 10^6 \text{ hours}$$

Parameter	Value	Justification
C_1	0.29	Assume Digital MOS device with 30,001 to 60,000 gates
C_2	0.0132	28-pin SMT package, assume nonhermetic $C_2 = 3.6\times10^{-4}\left(N_p\right)^{1.08}$, N_p = 28
π_T	1.5	Digital MOS device, assume max junction temperature, T_J = 100° C
π_E	0.50	Assume ground benign environment, G_B
π_Q	10	Assume unknown screening level
π_L	1.0	Generic device in production for ≥ 2 years

λ_p = 4.416 failures per 10^6 hours
MTTF = 226449.3 hours $\approx$ 25.9 years

The Cypress USB controller will act as a host so that the user can interface to the device using a portable USB thumb drive. This is the very first step in the process of converting the image file to a usable audio sample. Without the proper function of the USB controller, the device will do nothing of any significance no processing can begin until a valid image file is recognized on the USB drive. The USB controller was selected for reliability analysis because it has an external interface connector and is handling large amounts of information in a very time sensitive manner making it a likely candidate to fail. For this digital MOS device, it was assumed that it has up to 60,000 gates and a maximum junction temperature of 100° C. There was no information available for die complexity or junction temperature, thus both assumptions are pessimistic estimates and prove suitable for this analysis. Using these assumptions the part failure rate of 4.416 failures per 10^6 hours and a MTTF of 25.9 years was calculated. In this device, the failure of the USB controller would simply render the device useless but cause no physical harm to the user. As such, the calculated MTTF is acceptable. If the device was being analyzed for actual production, more information would definitely be needed, but seeing as this is a low criticality component and that they are worst case assumptions, the assumptions are justified and the MTTF is acceptable.

5.1.5 Reliability Analysis Conclusions

Component	**Part Failure Rate, λ_P (per 10^6 hrs)**	**MTTF (hrs)**
ADSP- 21262 SHARC Microcontroller	**18.52**	**54004.4**
KA278R05C Fairchild 5V and 3.3V LDO Regulators	**18.01**	**55530.9**
W56964 Winbond MIDI Chip	**4.426**	**225937.6**
SL811HS Cypress USB Controller	**4.416**	**227449.3**

The results of the reliability analysis for each component are summarized in Table 2.5. The shortest MTTF are found for the SHARC microcontroller and the LDO regulators. While the criticality of failure for the microcontroller is low, if the voltage regulator causes a short and leads to overheating, a serious risk is posed to the user. This high criticality is reason enough to consider alternate designs to improve reliability or decrease potential hazards to the user.

Perhaps a more sophisticated voltage regulator that can protect against shorts or an oversized heat sink mounted near the regulators. The MTTF of the USB controller and Winbond chip are similar and more than acceptable considering the low criticality of failure for either one of the components. Overall the MTTF of the device seems acceptable and within acceptable values considering out of all possible failures, really only one case could be potentially harmful to the user.

5.2 Failure Mode, Effects, and Criticality Analysis (FMECA)

In order to analyze failure modes, the schematic of the digital sheet music reader and player was broken into functional blocks. The power circuit (A), the USB controller block(B), the microcontroller (C), the external memory (D), the audio circuit (E), and the LCD and associated pushbuttons (F). The schematics for each block can be found in Appendix A, and the FMECA chart for each block in Appendix B.

Each of these blocks represents a unique function in the device, and as such has unique modes, effects, and criticality of failure. Two levels of criticality are considered for this analysis. For a failure that could potentially cause injury or personal harm to the user, a criticality of high was assigned. The failure rate for this level should be on the order of $\lambda_p < 10^{-9}$. For failures that simply render the device useless and cause only user dissatisfaction, a criticality level of low was assigned. The failure rate for this level should be on the order of $\lambda_p < 10^{-5}$

6 Ethical and Environmental Impact Analysis

Although this device by the nature of its intent to be used does not pose any great threat to the user there are a number of ethical and environmental concerns that this product will bring to the user. Ethical issues to consider are that the device could destroy the user's personal data contained in the USB drive, cause hearing loss through the headphone jack, or have a relatively short life span due to poor design. The device also can cause many environmental issues in manufacturing, normal operational use, as well as improper disposal.

6.1 Protection of USB Data

The DSMRP uses a USB flash drive that is provided by the user. This means that the USB flash drive will more than likely have other data besides the images of sheet music on them. We have tried to take every precaution so that we do not write over the other files on the USB flash drive or corrupt the drive. Microsoft Windows XP had to fix a data corruption problem earlier windows had had when the drive is not removed safely. Windows XP writes changes to the file on the USB drive as changes are made instead of leaving the file open on the drive.[1]

The team plans to use this same technique to overcome data loss problems to the user's USB drive. The device will need to import portions of the image at a time since the microcontroller only has 2 megabits of SRAM on chip [2] and the bitmap images will be around 500 kilobytes. The portions will be compiled together on the external memory. The image will never be left open in the USB drive while waiting to send more of the image, but instead one bulk write will be made at a time until the entire image has been moved onto the external memory. All intermediate symbol (IMS) files will be written on external SRAM first and then written to the USB drive once the entire file is done being made. USB drives are hot-swappable, which means they can be plugged and unplugged from the device at anytime [3], but to be safe we will have in the manual a caution against unplugging the USB drive from the device until the user is completely done using the device. This way all files will be finished being written to or being read from the USB drive when the user unplugs the USB drive.

6.2 Hearing Loss from Sound Considerations

Conversation occurs at 60 dB, and sound at 85 dB will cause hearing loss after eight hours of exposure to it [4]. The DSMRP has the capability of being developed to play sound in later upgrades, so for now there is no concern. The audio processor does have a volume control that we will be able to limit to the output of our stereo out volume, but headphones are so close to the ear that extended listening at sound levels that don't seem that loud can still be damaging. When an upgrade is made with music, a warning will be included in the manual on this issue. If the user needs the sound to be louder so that a class room can hear, the user can hook the device up to external speakers through the stereo out connection on the device.

6.3 Lifespan Considerations

By the nature of the application of the DSMRP, the device should not suffer from many extreme external circumstances. For normal application the device will be kept in classrooms, which is out of the sun and the winter's cold. The device is not intended to withstand very much physical abuse since it is meant to be rested on a desk. From the Military Handbook for Probability Prediction of Electronic Equipment, it was found that the LDO regulators will fail 18.01 times per 10^6 hours [5], which translates to the product fails after just over 6 years. This would not be too bad of a problem except the LDO regulators can short to ground and cause overheating and potentially burning. This product does not need to last that long for a few reasons. Technology outdates present technology at a growing rate. The modern computer lasts a couple of years. It is also not cost effective to make the lifespan longer. LDO regulators will only improve in time, and they can be integrated in future designs.

6.4 Manufacturing Considerations

The DSMRP has a PCB for a main board as well as one inside of the LCD screen. The LCD screen and the PCB manufacturing causes many waste products including industrial wastewater and treatment residues, spent process baths, e/acids used for cleaning equipment, copper sulfate crystals, and re-flow oil [6]. The first step the team will take will be to find a manufacturer that will grant our request that the present environmental regulations are met or exceeded. We will also try to reduce waste by keeping the PCB as small as possible. All surface mount parts will be used as much as possible, and both sides of the PCB will be used as efficiently as possible.

6.5 Normal Use Considerations

The team is designing this device so that it takes the least amount of power possible. Lowering power will lower demand and waste like CO_2 from power generation plants [7]. The internal clock on the microprocessor will be clocked down when the device is first turned on until a USB drive is inserted into it. All IMS files made from processing images of sheet music will be saved on the USB so that the heavy DSP calculations that draw a lot of current to the microprocessor do not have to be done every time the device is power cycled. This will also shorten the time that the user has to keep the device on since less time will be spent on preparing the MIDI files.

6.6 Disposal and Recycle Considerations

The DSMRP packaging will be made out of aluminum. This means that the only parts of this device that will need to be recycled using special means are the PCB in the LCD and the main PCB. Several companies offer recycling services that recover the raw materials from PCBs [8]. A label will be put inside the packaging next to the main board as well as a note in the device's manual suggesting that the PCB be recycled with a company that offers the proper services.

Although he DSMRP does have some ethical and environmental issues the device is not any different than most electronic devices on the market. If the device is correctly designed, the user will not have to worry about hearing loss or data loss. Probability of the device burning up is also very small especially since it does not have a battery. Unfortunately there is not a way to make an embedded system without some hazardous materials being produced in the PCB and the manufacturing of the PCB, so there will be documentation that will point the user to where the user can recycle the board.

7 Packaging Design Considerations

7.1 Commercial Product Packaging

When considering the desired packaging for the digital music sheet reader, two commercial products were found to have both similar function and meet several of the set design criteria for the device.

7.1.2 SONY Model TCM-929 Standard Cassette Recorder

This SONY device is the typical desktop cassette player and recorder. Specifically, the recorder is 5.7w x 8.4d x 2.3h (inches) and weighs 1.6 lbs [4]. Some of the positives of this package include the size and portability of the device. This package is small enough to be transported without difficulty and the handle makes carrying it from place to place trivial. Functionally this recorder is very similar to the digital sheet music reader which proves beneficial because the package has a speaker built into the housing, similar to our device. Several of the drawbacks of this device are the fact that it is a tape recorder which in fact plays no role in the function of the sheet music reader. Also the panel of buttons is a disadvantage in the sense that we are looking to utilize a RPG for digital menu navigation.

To facilitate more involved user interfacing; the digital sheet music package will include an LCD for menu display, an obvious feature that this recorder is lacking. Also the major data input of the sheet music reader is intended to be a USB flash drive which facilitates the need for an onboard USB host controller. The necessary integration of the LCD and USB elements of our design are the major places where our package will differ from this SONY package. Also, this package is designed to operate on its “back”, which will differ from the sheet reader that is designed to sit upright to allow the user a good view of the LCD display and easy navigation on the menu.

7.1.2JVC RC-EZ38 USB Host Portable Boom box

This JVC boom box appears to be the futuristic version of the timeless jam boxes or ghetto blasters of the early 80’s. This package is a stand alone portable desktop model with dimensions 18.6w x 7.8h x 10.7d (inches) and weighs 8 lbs. [5]. Some of the positive aspects of this package design include the LCD screen, USB host controller, carrying handle, and the two vented speaker bays. These few extra external features give the JVC a much more broad interfacing capability. The weight is a major concern for this device. While it is portable, the JVC boom box weighs 8lbs which is heavy enough to cause some discomfort if it is to be transported by hand for an extended amount of time. A majority of this weight comes from the batteries used to power this device.

The digital sheet music reader will incorporate most of the external interface elements of this package. The LCD screen, USB connector, external speakers, and handle are all elements that will be present in the sheet music reader package. Also the JVC is oriented facing “forward” so that the display is easy to see while on a desk. The digital sheet music reader package will model this orientation in order to maximize user interfacing. While these elements will be modeled, the overall size and weight of the device will differ from the digital sheet music reader dramatically. In order to facilitate this, the sheet music reader will only employ a single speaker namely because the device is intended for more preview quality audio. Also the digital sheet music reader will not have large on board batteries to power the device.

7.2 Project Packaging Specifications

The digital sheet music reader package is a simple project box with some modifications for the DSMRP. Dimensionally the box is 7w x 7h x 2.5d (inches) [6] and is oriented in a portrait manner with the 7 x 2 inch face acting as the bottom and the 7 x 7 inch face acting as the front. The major components that are housed in or on the box are the PCB, the LCD, two pushbuttons, a RPG, and several external sockets for USB, power, headphone out and line out. The LCD, RPG, and the push buttons will all be installed on the front of the box in order to maximize user interface and usability. The package has room for an external speaker but on the current version of the DSMRP none is installed. The speaker could be mounted just below the RPG and push buttons on the front of the enclosure.

The PCB is mounted on the lower level of the box leaving room for the LCD in the upper half. The PCB orientation is set up so that an easy distinction can be made between the external interfaces on the package. The USB socket is mounted on the right side of the box, while the power, headphone and stereo outs are on the left side of the box. The specific position of these inputs and outputs on the actual enclosure is more of a function of the PCB layout than user interface or functionality. The reason for the PCB layout constraining the position of these components is their need to be attached to the PCB and to have an external interface.

7.3 PCB Footprint Layout

There are several design constraints regarding the sheet readers PCB. To begin with, there are several external peripherals with external connectors. These external interfaces include the USB connector, power connector, headphone out, and the speaker out. Because of these requirements, the chips corresponding to each of the connectors must be placed relatively near the edges of the board. Knowing this, it follows that components are placed on the board similar to the pipeline of data processing occurring on the board. The pipeline consists of the image coming in on the USB, moving to the main microcontroller, being processed into music information, moving to the audio processing chip, and then being played through the speaker or headphone jack.

The major components of the sheet music player were selected based on the overall design constraints, and in most cases the packages were selected because the option was typically between a BGA package and QFN or PLCC packages. For our application the BGA package will not work, so in most cases the QFN or PLCC was selected. Although is not the most desirable it was necessary to choose the QFN over the BGA given those were the only options for some of the components. The USB controller is a Cypress SL811HS Embedded USB Host/Slave Controller which comes in a 28pin PLCC with dimensions 12.45 x 12.45 x 4.58d mm [1]. The microcontroller is a Sharc ADSP-21262 which comes in a 144 pin LQFP package with dimensions 22 x 22 x 1.6d mm [7]. The Audio chip is a Winbond W56964 which is packaged in a 32 pin QFN with dimensions 5.2 x 6.2 x 1 mm [8]. From these major components, the expectation of other passive and non-major components, and the constraints set by the actual enclosure, the initial size of the PCB will be 4 x 6 in. This not only is a suitable size for all of the components, but also is a result of the package size and the ability to interface several external elements.

8 Schematic Design Considerations

8.1 Theory of Operation

USB Controller: Cypress - SL811HS

The Digital Sheet Music Reader and Player starts up and waits for a USB drive to be connected. According to USB standards, one device on a USB bus must act as the host. A USB thumb drive is incapable of being the host, so the USB controller operates in Host mode. A USB thumb drive is considered part of the Mass Storage Class and communicates via Bulk-only Transport. [] Every bulk transfer needs to start with a 31-byte Command Block Wrapper(CBW) telling the device what type of data transaction to expect. The requested data is either sent or received then the host requests a 13 byte Command Status Wrapper (CSW) from the USB device. The CSW contains the result of the data transfer from the perspective of the USB device, this is compared with the USB host to determine if the transfer was successful or not. Support for the FAT filesystem on the USB drive will be included to achieve file I/O which is compatible with Windows.

Microcontroller: ADSP-21262 (SHARC)

The SHARC operates as the central processing unit; all other major subsections are directly linked to it. The USB controller, external memory, and MIDI processing audio chip are connected to an 8-bit data bus. All ICs on the bus have tri-state pins which are driven only after the appropriate chip select signal. Chip selects are not generated by the SHARC, so a PLD uses the three most significant address lines to generate the three required chip selects. Each device is mapped to a specific address so that the three most significant address lines address only one IC at a time.

The SHARC can be set in five different computational modes; the default being 32-bit floating-point format. We have no extenuating circumstance to choose any other computational mode, so it is sufficient to use the default mode.

<u>Memory: Cypress – CY7C1069AV33 (2M x 8) SRAM</u>

The operating mode of the memory is rather straight forward, it gets complicated when considering the multiplexed bus and the mapping of the memory in the SHARC. One of the major concerns will be timing. There are 21 address lines and eight data lines in addition to three control lines and all of these will be operating at speeds up to 66 MHz; these address and data signals are time multiplexed to 16 pins on the SHARC.[] The generation of the chip select by the PLD will be crucial because the chip select will need to go high immediately after the data transfer so that the data pins become high impedance to avoid bus contention. A supply voltage of 3.3V was chosen because the SHARC address and data pins use 3.3V logic levels. Using this supply voltage prevents the need of using a 32-bit level translator for the entire bus.

<u>Power</u>

The main power for the board was chosen to be 6V and will be used to power the 3 voltage regulators at 5.0V, 3.3V and 1.2V. The LCD and USB bus expect 5.0V and the digital components require 3.3V and 1.2V, all of which will be obtained from three LDO linear regulators connected to the main 6V power source. There is come concern that digital switching noise will affect the audio amplifier on the MIDI chip. To remedy this we designed a low pass filter using a ferrite bead chip for analog-digital power separation.

8.2 Hardware Design Narrative

<u>Microcontroller: ADSP-21262 (SHARC)</u>

The parallel port on the SHARC plays an integral role in interfacing the major subsections together. All devices on the parallel port are 8-bits wide. This 8-bit bus is multiplexed with 16 pins that provide both the data and address information. The USB controller and MIDI processor are connected directly to the first 8 data pins, while the memory needs 21 address lines and 8 data lines. A 16-bit external latch is used to hold addresses 8-20 while the processor drives the lower 8 bits of the address.

The Serial Peripheral Interface (SPI) port is enabled and used for the LCD display. The baud rate is set significantly lower than the core clock speed because the LCD doesn't transfer large amounts of data and it isn't able to support fast baud rates. Three pins are required for the clock, serial data and chip select; but they are at 3.3V logic levels and the LCD requires 5V. A 'high' bit of 3.3V is above the threshold for a 'high' on the LCD, so level shifting of the three signals will no be necessary.

The SHARC has a Digital Audio Interface (DAI) that this project does not require, but several DAI pins are configured as general purpose I/O pins for menu navigation buttons. Two buttons and one Rotary Pulse Generator (RPG) are used and they require four I/O pins. These pins have the option of being interrupt driven or polling/status driven. The pushbuttons and RPG are interrupt driven and routed using the Signal Routing Unit (SRU) to the specified pins on the DAI. The main loop continually polls the navigations flags which are set upon detection of an interrupt. The previous state of the RPG is saved as this is required to determine the direction of rotation. I/O pins not being used utilize the internal pull-up resistors.

<u>USB Controller: Cypress - SL811HS</u>

An external clock of 12-MHz is required by the SL811HS for host mode. The reset pin is driven by a GPIO on the DAI of the SHARC because this IC needs to be reset on certain events. The interrupt pin currently isn't being utilized but is routed to a GPIO with the capability of triggering interrupts for various events. The interrupt wasn't used because the polling nature of our main program made it easy to query the SL811HS periodically. The series resistor values of 24 Ohms and 15K Ohms pull-down resistors were recommended by the SL811HS Interfacing document and are common to the USB standard.

<u>Audio: Winbond - W56964</u>

This MIDI chip includes a built-in amplifier that drives an 8-ohm speaker at 550mW. A simple low pass filter is implemented using a ferrite bead to separate the digital switching noise from the analog amplifier. The chip provides a 3-band equalizer that uses the recommended values to produce a flat equalization. The address/data bus interface will be similar to the other chips on the bus; 12 lines for the data and control lines with a chip select that is to be generated via PLD. Although support for this MIDI chip is currently not supported in our code, future revisions fully intend to utilize the features of this chip.

9 PCB Layout Design Considerations

9.1 PCB Layout Design Considerations – Overall

The PCB will be divided into 5 sections based on the major modules of the device. These sections are the USB module, power module, microprocessor module, user interface module, and audio Module. The USB module will be involve the SL811HS Integrated USB Host, a standard USB Type A receptacle connector, a 12 MHz clock and filtering for the clocking signal . The Microcontroller Module will encompass the ADSP-212602 DSP Microprocessor,74VCXH16373 16- bit D-type Latch, CY7C1069 BV33- (2M x8- bit) Static RAM, AT25F4096 4Mb SPI Flash, 16V8 PLD, 12 MHz Crystal, and Low pass filtering for the clocking signal. The Power Module consists of the 6 volt Regulated Wall Wart,5 volt LDO regulator 3.3 volt LDO regulator, 1.2 volt LDO regulator and low pass digital to analog circuit separation filtering and the power connector. The Audio Module has a MIDI interpreting chip and headphone jack. Finally, The User Interface Module allows a person to communicate via a LCD screen, two pushbuttons, and a rotary pulse generator. A technician will be able to communicate with the device using a JTAG connection.

The PCB will have dimensions of about 153.7 mm x 143.51 mm. The digital circuitry includes the components for the USB, microprocessor, and user interface modules. The analog circuitry consists of the audio module. The layout of the board has physical separation between the digital circuitry, the audio module and power module. The separation between digital circuits and audio module is present to reduce digital switching noise on the analog audio outputs. This is achieved by placing the power and audio outputs on the right side of the board and the digital components on the left side of the board. The only analog traces within the circuit, excluding power traces, are the output from the audio chip to the headphone jack. Therefore, by confining the analog circuitry to the bottom right side of the board the analog traces can be separated from digital traces, thus preventing cross coupling which can degrade audio performance and digital signal quality.

A large amount of current is needed by the audio module to drive a speaker connected to the PCB. As a result the audio module is placed as close to the power supply as possible, without being so close that signal quality is degraded by noise from the power traces. While the audio module consists mostly of analog signals, there are still digital signals passing into the module through the data bus connected to the audio chip. The cross coupling of capacitors prevented by not having an analog signal crossing a digital signal trace.

The project provides a user interface through push buttons a LCD and various jacks away from the PCB. All of these interface connectors are connected using standard friction lock connectors. The USB jack is mounted on board and is connected using traces. The power connector is connected using a clip lock connector to prevent accidental disconnection.

9.2 PCB Layout Design Considerations – Microcontroller

The microcontroller being used for the project is a 144-pin DSP chip. This chip requires a 3.3 volt source that will supply 10 pins interspersed around the chip and a 1.2 volt source supplying 34 pins scattered across the chip [1]. Coupled with 39 ground pins around the board, the power routing is a significant challenge with a two layer PCB. The best course of action was to run all power and ground traces from the regulators to the microcontroller area through the bottom of the PCB and bring them up through vias to the top layer of the PCB before connecting to pins. Ground traces were also run parallel to the power to the DSP from vias brought to the top layer next to the microcontroller. If power had been routed around the outside of the microcontroller they would run parallel to the data bus lines. Therefore, the power is routed from under the microcontroller because this will cause the least amount of cross coupling disturbance on the address/data bus which has pins located right next to power pins.

Eight coupling capacitors were required for the microcontroller in the form of six 0.1μF capacitors and two 0.01μF capacitors. All of them are located on the bottom side of the PCB under the microcontroller in a star pattern to make them as close as possible to the pins for an instantaneous supply of current. [2]

All address/data pins are routed on the top side of the PCB in all cases except those were the individual address lines were in the way of other address lines. The timing for data packets sent between the microcontroller and other chips are crucial, so routing between the bottom and top of the board through the use of vias was kept to a minimum. This reduced the trace capacitance.

Finally, an external clocking signal was placed next to the clock input pins of the DSP to provide the most accurate clocking signal possible. In addition, filtering was used to ensure that the power lines had the cleanest signal possible. This filtering utilized a ferrite bead and capacitors, laid out according to device specifications. [3]

9.3 PCB Layout Design Considerations - Power Supply

The power module of the project consists of a 6 volt wall wart power supply and two regulators. These components provide 3 voltages for the system. The 6 volt supply is used to power 3 regulators. At the point where the power supply enters the PCB, coupling capacitors and a ferrite bead were attached [4]. The bulk capacitor helps provide instantaneous current to the regulators and LCD and the smaller capacitor helps smooth out high frequency noise from the outlet. From the 6 volt supply, one regulator provides a 1.2 volt signal to the microcontroller and the other regulator provides a 3.3 volt signal to the microcontroller, audio chip, USB host, and external memory. All of these components are located on the left side of the board to isolate them from the rest of the components and reduce signal noise from coupling between traces.

The three voltages are distributed into two separate power circuits. The 3.3 volt and 1.2 volt source are routed to the digital power circuit and the 3.3 volt and 5 volt signal are routed to the analog power circuit. By having two semi-separate power circuits the effects of digital noise are more easily isolated from the analog circuit because traces are separated from each other as soon as they leave the regulators to prevent cross coupling. In addition all ground traces should be run in parallel to the power traces when possible to cancel EMI [4].

All of these power traces are being run on the bottom of the board to separate them from clutter of the other circuitry and reduce the use of vias. The power traces will mostly be routed inside of the board and along the right side isolate them from critical data signals. This will also allow room for the traces from the power supplies to begin as wide as possible and gradually decrease in width as they approach the pins for powering of the chips. By keeping the traces at 40 mils in width for as long as possible, the resistance will be kept to a minimum and the power dissipation in the traces will be lower. [4]

10 Software Design Considerations

This is a very heavily software dependant device using a SHARC ADSP-1262 [1] embedded processor to convert the images into code stored as intermediate music symbols (IMS) files that represents the notes' pitch, length, and location in the song. The code will then need to be converted into the MIDI format to be played through a MIDI processor. Additional peripheral units include the LCD screen and the user interface consisting of pushbuttons and a rotary pulse generator (RPG). Software to access all of the IMS files and selected images on a FAT [2] formatted USB drive must be created so that the files can be saved on our device's external memory and files created by our device can be saved back on the USB flash drive.

10.1 Memory Mapping:

0x0000 0000 – 0x0003 FFFF	Registers
0x0004 0000 – 0x0007 FFFF	Bank 1 (long word)
0x0008 0000 – 0x000F FFFF	Bank 2 (normal word)
0x0010 0000 – 0x001F FFFF	Bank 3 (short word)
0x0020 0000 – 0x00FF FFFF	Reserved
0x0100 0000 – 0x011F FFFF	Unmapped memory addresses
0x0120 0000 – 0x013F FFFF	External SRAM
0x0140 0000 – 0x0140 0001	USB controller
0x0140 0002 – 0x017F FFFF	Unmapped memory addresses
0x0180 0000 – 0x0180 0001	MIDI processor
0x0180 0002 – 0x02FF FFFF	Unmapped memory addresses
0x0300 0000 – 0x3FFF FFFF	Reserved

Bank 1

0x0004 0000 – 0x0004 3FFF	Block 0 SRAM (Code)
0x0004 4000 – 0x0005 7FFF	Reserved
0x0005 8000 – 0x0005 FFFF	Block 0 ROM (Mask Programmable)
0x0006 0000 – 0x0006 3FFF	Block 1 SRAM (Static Data, Stack, and Variables)

0x0006 4000 – 0x0007 7FFF	Reserved
0x0007 8000 – 0x0007 FFFF	Block 1 ROM (Mask Programmable)

The structure of Banks 2 and 3 are arranged similar to Bank 1. The SRAM in Banks 2 and 3 are reserved for an image being processed. The mask programmable ROM can be used when the device is mass produced for program code instead of the SPI Flash.

10.2 External Interfacing Mapping:

The device uses SPI flash on chip select 0 so that on boot up the processor loads the program code into internal SRAM. The LCD uses chip select 1. The push buttons each uses one pin from the Digital Applications Interface (DAI), and the rotary pulse generator, rpg, uses two pins. These pins are mapped through the Signal Routing Unit (SRU) as low priority interrupts. The USB controller, external SRAM, and the Winbond chip use the external bus uses the parallel port [AD0:AD15]. It receives data at 0x000 1809, and it transmits data at 0x0000 1808. The external bus has three control lines [ALE, Read, Write].

10.3 Utilization of Integrated Peripherals:

The SPI and the Parallel Port are core driven and are used to interface with all external ICs. The DAI is used for the push buttons as well as resets for the Winbond chip and the USB controller.

10.4 Organization of Application Code:

The device has two states that it can be in during operation. One state is where the user has a menu to select different options. The other state the user will have no control of the device, and the device will be processing data to create an IMS file from an image or a MIDI file from IMS files. In the first state the processor will need to be looking for when push buttons are set or a USB drive is inserted by the user when in the menu, so the processor will be acting in a polling loop at this time. In the second state the processor will need to process data and create files without user intervention, so the processor will be acting as command driven. Therefore, the device will be a hybrid of command driven and polling loop.

10.5 Flowchart: (see Appendix H)

10.6 Provisions Made for Debugging

There are 6 pins on the microprocessor that are used for JTAG/ICE debugging. The image processing algorithms were developed on Matlab. SourceUSB software was used to analyze standard USB bus traffic. Self testing signals, voltage levels, and pin outs were done extensively to ensure the units were properly placed on the PCB and within operation limits before using them.

10.2Software Design Narrative (see Appendix I for hierarchical diagram)

Mayhem: This is the main polling loop and menu module. Each time through the loop it checks if an interrupt has occurred corresponding to either of the push buttons or either of the directions for the rpg. If any of interrupt flags are set, an event occurs to modify the menu. If the select push button is pressed, the menu moves into the next sub-menu or selects the desired choice if it is the bottom sub menu. The back push button moves control to the previous menu unless the current menu is the main menu, in which back is invalid. The RPG interrupts allow current menu options to be scrolled through in a circular manner. The Mayhem module calls functions from all of the other modules in order to carry out the user desired functions.

Initialize: The SPI for the LCD screen is initialized here, and the SRU is used to route interrupts to the DAI for the push buttons and the rpg. The SPI is set to 32-bit words and MSB sent first. This module has been successfully tested and integrated into our software.

menu: The image processing set up sub menu is more complex than the main menu because of the various choices of MIDI instruments and musical notation. This module is similar in structure to the main menu, but contains all of the image set-up options and writes them to a string that eventually becomes the header for a given IMS file. Also the menu module contains helper functions for formatting the LCD menu and other displays.

USB_Driver: The complexity of the USB standard requires drivers to allow us to communicate with a USB device. Our module contains functions that meet the specifications of timing, error handling, and packet size and content. Some basic initialization functions were adapted for the 32-bit architecture of the SHARC, other functions were developed to interface with a mass storage device such as a USB drive. Higher level functions from a development board's software for the Intel 8051 interfaced with our chip was used in creating this module [3]. This module has been successfully tested and integrated into our software.

fatFxns: This USB drives use the FAT file system to store data. This module contains the drivers that enable FAT file reading and writing. The higher level functions were found from a project that interfaced with a compact flash card. The lower level functions were written using the USB driver, and enable the use of the higher level functions with minimal modification.

image_processing: The Optical Character Recognition (OCR) for sheet music was centered around creating histograms of the amount of black pixels in rows or columns of different segments of the music [4]. A histogram function is declared that can be adapted to be used by different functions to find staves, notes on a staff, and the location and value of each note. From the OCR of the sheet music, an intermediate symbol file (IMS file) is made to represent the sheet music in numerical values. This module has been successfully prototyped in Matlab and written in C, but it has not been tested and integrated into the software.

FileFunctions: The file conversion takes place on the SHARC, however the actual files need to be stored on the external SRAM. These functions allow writing to the SRAM one byte at a time by packing each byte into a buffer then sending 4 byte chunks in order to minimize the space taken up on the SRAM.

IMS_Build: This module is called by the image processing module to create an IMS file from the data specified by the user and the image processing. The IMS files will store a byte that represents the note's length and another byte that represents the note's pitch for each note for that instrument. It also will store the song's tempo, key signature, type of clef, and time signature. All of this data is necessary to create a MIDI file. This software has been successfully tested and integrated into the software.

IMS_MIDI: This module will take an IMS file and convert the stored data into a MIDI file. This module will look at the first byte of the IMS file for the song's tempo, the second byte for the key signature, the third for clef, the fourth for the time signature, and then it will look at two bytes at a time for each note's length and pitch [5]. The function then stores the MIDI file onto the USB drive. This module has been successfully tested and integrated into the software except that the code to store the MIDI file onto the USB drive has not been completed.

ISR: The menu polls flags to see if push buttons or the rpg was used. The pushbuttons and the rpg have interrupts routed via the SRU so that when an interrupt is generated the DAI interrupt subroutine sets the corresponding flag. This module has been successfully tested and integrated into our software.

11 Version 2 Changes.

If we were to develop a second version of the DSMRP, there are several changes and additions that would be made. A very easy change would be to add the capability to compile multiple IMS files together to make one MIDI file. This addition would probably make the device more appealing for musicians who play in bands and orchrestras. We would find a better audio processor for playing the music. The tech support for the processor was not very helpful, and lacked communication skills. Along with the chip we would want to get a larger speaker and an amplifier so that a classroom could hear this device play music. The image processing has a lot of room for improvement. There are still several characters that are not recognized by the software. The software is also not entirely reliable with what it does recognize. A great feature would be to add a graphical LCD screen that would display the notes that the Optical Character Recognition (OCR) found on staves just like it was originally on the sheet music. This way the user look through the image on the screen for missing and miss read notes. There is a lot of room to expand on the USB driver. One simple addition that would be nice would be if it supported FAT32 USB mass storage drives as well. We are happy with where this device is at, but these additions would make this device much more marketable and exciting to us as well.

12 Summary and Conclusions

The group was able to achieve 4 of the 5 PSSCs that it had set. The DSMRP is able to receive user input through the use of a LCD screen, push buttons, and a rotary pulse generator. It is also able to act as a USB host by uploading files from a USB device and saving them to memory. The DMSRP is also ableto accept either .ims files or bit map files and convert an uploaded .ims file to a MIDI file or convert a bitmap file to an .ims file. The PSSC that was not achieved was the ability to play a MIDI file from the device.

The group has learned about the MIDI standard and successfully utilized that knowledge to create software that converts an .ims file to a MIDI file. A firm understanding of the USB protocol was essential in producing a device that acts as a USB host because the host must control all traffic on the USB network. As a result the software that was edited and created had to properly initialize packet transfers in order to read information from a mass storage device. When packets of data were finally sent, a FAT driver had to written specific to the processor used in order to move data from USB to memory and memory to the DSP for processing. The group performed a lot of research in the methods of OCR in order parse an image and extract meaningful musical data out of it. This project has allowed each and every member of the group to expand their knowledge above and beyond what has been taught in the classes at Purdue. Furthermore, the group can now see that Purdue has taught each member to grow and adapt to any project presented to engineer a design that works.

13 References

13.1 Constraint Analysis and Component Selection References: Section 3

[1] Analog Devices, "ADSP-2126x SHARC Processor Peripherals Manual", **http://www.analog.com/UploadedFiles/Associated_Docs/62441403275532638 7ADSP_2126x_HRM.pdf**, December 2005.

[2] Atmel, "AT91 ARM Thumb Microcontrollers," **http://www.atmel.com/dyn/resources/prod_documents/doc1354.pdf**, August 2002.

[3] Microchip, "PIC24FJ128GA Family Data Sheet", **http://ww1.microchip.com/downloads/en/DeviceDoc/39747C.pdf**, June 2006.

[4] Cypress, "SL811HS Embedded USB Host/Slave Controller," **http://download.cypress.com/publishedcontent/publish/design_resources/datasheets/contents/sl811hs_8.pdf**, June 2005.

[5] Cypress, "CY7C1049BV33: 512K x 8 Static RAM," **http://download.cypress.com/publishedcontent/publish/design_resources/datasheets/contents/cy7c1049bv33_8.pdf**, September 2002.

[6] Renesas, "R1RW0408D Series: 4M High Speed SRAM (512-kword x 8-bit)," **http://documentation.renesas.com/eng/products/memory/rej03c0111_r1rw0408d.pdf**, March 2004.

[7] Winbond Electronics Corp., "64-Note Polyphonic Ringtone Chip," June 2006. (Obtained via email with a Winbond employee)

[8] MIDI Manufacturers Association, **http://www.midi.org**, September 2006.

13.2 Patent Liability Analysis References: Section 4

[1] Ohya, M., " Apparatus for interpreting written music for its performance,"U.S. Patent 5,202,526, April 13, 1993. Available HTTP: **http://patft.uspto.gov/netacgi/nph-Parser?Sect1=PTO1&Sect2=HITOFF&d=PALL&p=1&u=%2Fnetahtml%2FPTO%2Fsrchnum.htm&r=1&f=G&l=50&s1=5202526.PN.&OS=PN/5202526&RS=PN/5202526**

[2] Kikuchi,T. "Musical Score Recognition Apparatus with Visual Scanning and Correction." U.S. Patent 5,825,905, October 20, 1998. Available HTTP: **http://patft.uspto.gov/netacgi/nph-Parser?Sect1=PTO1&Sect2=HITOFF&d=PALL&p=1&u=%2Fnetahtml%2FPTO%2Fsrchnum.htm&r=1&f=G&l=50&s1=5825905.PN.&OS=PN/5825905&RS=PN/5825905**

[3] Taub, R. "System and method for musical score capture and synthesized audio performance with synchronized presentation." U. S. Patent Application 20060150803, July 13, 2006. Available HTTP: **http://appft1.uspto.gov/netacgi/nph-Parser?Sect1=PTO1&Sect2=HITOFF&d=PG01&p=1&u=%2Fnetahtml%2FPTO%2Fsrchnum.html&r=1&f=G&l=50&s1=%2220060150803%22.PGNR.&OS=DN/20060150803&RS=DN/20060150803**

[4] Bainbrindge, D. and Bell, T., "An extensible Optical Music Recognition System" Nineteenth Australasian Computer Conference. [Online], May 2001, Available : **http://www.cs.waikato.ac.nz/~davidb/publications/acsc96**. [Accessed September 23, 2006].

[5] Fujinaga, I. "Adaptive Optical Music Recognition."[Online document], June 1996. Available: **http://www.music.mcgill.ca/~ich/research/diss/.** [Accessed September 23, 2006].

13.3 Reliability and Safety Analysis Referances: Section 5

[1] Department of Defense, "Military Handbook Predicting Reliability of Electronic Equipment (Mil-Hdbk-217F),"[Online Document], 1991 December 2, Available at HTTP: http://cobweb.ecn.purdue.edu/~dsml/ece477/Homework/Fall2006/Mil-Hdbk-217F.pdf

[2] Analog Devices, "SHARC Embedded Processor ADSP-21262," [Online Document], Aug. 2005, Available HTTP: http://www.analog.com/UploadedFiles/Data_Sheets/462724356ADSP_21262_b.pdf#search=%22SHARC%20Embedded%20Processor%20ADSP-21262%22

[3] Fairchild Semiconductors, "KA278RXXC-Series 2A Output Low Dropout Voltage Regulators," [Online Document], 2005 February 18 (Rev 1.0.2), Available at Http: http://www.fairchildsemi.com/ds/KA%2FKA278R33C.pdf#search=%22KA278RXXC-Series%202A%20Output%20Low%20Dropout%20Voltage%20Regulators%22

[4] Winbond Electronics Corp., "64-Note Polyphonic Ringtone Chip," [Online Document], June 2006 (Rev A7), Available at HTTP: http://cobweb.ecn.purdue.edu/~477grp4/W56964%20data%20sheet%20rev.%20A7.pdf

[5] Cypress Semiconductor, SL811HS Embedded USB Host/Slave Controller," [Online Document], 2002 March 14, Available at HTTP: http://www.mscbp.hu/download/Cypress/SL811HS/SL811HS.pdf#search=%222.4%20SL811HS%20Cypress%20USB%20Controller%22

13.4 Ethical and Environmental Impact Analysis: Section 6

[1] Windows Hardware Developer Central, "USB Storage – FAQ for Driver and Hardware Developers" [Online Document], Oct. 2004, Available at HTTP: **http://www.microsoft.com/whdc/device/storage/usbfaq.mspx**

[2] Analog Devices, "SHARC Embedded Processor ADSP-21262" [Online Document], Aug. 2005, Available at HTTP: **http://www.analog.com/UploadedFiles/Data_Sheets/462724356ADSP_21262_b.pdf#search=%22data%20sheet%20ADSP-21262%22**

[3] Dangerous Decibels, "Information Center: Hearing Loss" [Online Document], 2006, Available at HTTP: **http://www.dangerousdecibels.org/hearingloss.cfm**

[4] Brain, Marshall, "How USB Ports Work" [Online Document], Available at HTTP: **http://computer.howstuffworks.com/usb.htm**

[5] McQuiston, Ben, "Homework 11: Reliability and Safety Analysis" [Online Document], Oct. 2006, Available at HTTP: **http://cobweb.ecn.purdue.edu/~477grp4/Homework/HW11.pdf**

[6] Apple Computer, INC, "Apple and Environment" [Online Document], 2006, Available at HTTP: **http://www.apple.com/environment/energyefficiency/**

[7] Virginia Waste Minimization Program, Vol. 1 Issue 9, "Fact Sheet: Printed Circuit Board Manufacturers" [Online Document], Oct. 1995, Available at HTTP: **http://es.epa.gov/techinfo/facts/vdwm/va-fs6.html**

[8] Joint Service Pollution Prevention Opportunity Handbook, "Printer Circuit Board Recycling" [Online Document], May 2003, Available at HTTP: http://p2library.nfesc.navy.mil/P2_Opportunity_Handbook/2_II_8.html

13.5 Packaging Design Considerations: Section 7

[1] Cypress Semiconductor Corporation. (2005, June 13) SL811HS Embedded USB Host/Slave Controller. Available: http://www.cypress.com/portal/server.pt?space=CommunityPage&control=SetCommunity&CommunityID=209&PageID=259&fid=10&rpn=SL811HS

[2] Crystalfontz 632 Serial LCD, http://www.crystalfontz.com/products/632/mech_mm.html\

[3] Grayhill Incorporated. Optical Encoders Series 61 C. Available: http://lgrws01.grayhill.com/web/images/ProductImages/I-21-22.pdf

[4] SONY Model TCM-929 Standard Cassette Recorder, http://www.staples.com/webapp/wcs/stores/servlet/StaplesProductDisplay?storeId=10001&catalogId=10051&langId=-1&productId=36637&cmArea=SC5:CG66:DP2140:CL159675

[5] JVC RC-EZ38 Portable USB Host Boom box, http://focuscamera.com/sc/main_item.asp?id=964672132&lhalf=allacc

[6] Cap Cover Chassis, http://www.action-electronics.com/lmcapcov.htm

[7] Analog Devices Inc. (2004, February) ADSP-2126x SHARC® DSPCore Manual. Available: http://www.analog.com/processors/sharc/technicalLibrary/manuals/index.html#Processor%20Manuals

[8] Winbond Electronics Corporation, "64-Note Polyphonic Ringtone Chip". June 2006.

[9] Square Ferrite Speaker 2.6 inch, http://www.jameco.com/webapp/wcs/stores/servlet/ProductDisplay?langId=-1&storeId=10001&catalogId=10001&productId=31958&pa=31958PS

13.6 Schematic Design Considerations: Section 8

[1] Analog Devices, "ADSP-2126x SHARC Processor Peripherals Manual", http://www.analog.com/UploadedFiles/Associated_Docs/624414032755326387ADSP_2126x_HRM.pdf, December 2005.

[2] Cypress, "SL811HS Embedded USB Host/Slave Controller," **http://download.cypress.com/publishedcontent/publish/design_resources/datasheets/contents/sl811hs_8.pdf**, June 2005.

[3] Cypress, "CY7C1069BV33: 16-Mbit (2M x 8) Static RAM," **http://download.cypress.com/publishedcontent/publish/design_resources/datasheets/contents/cy7c1069bv33_8.pdf**, August 17, 2006.

[4] Winbond Electronics Corp., "64-Note Polyphonic Ringtone Chip," June 2006. (Obtained via email from a Winbond employee)

13.7 PCB Layout Design Considerations: Section 9

[1] Analog Devices. "ADSP-21262 SHARC Embedded Processor Summary." **http://www.analog.com/UploadedFiles/Data_Sheets/462724356ADSP_21262_b.pdf** December 2005.

[2] Analog Devices. "ADSP-21262 SHARC Embedded Processor Peripherals Manual." **http://www.analog.com/UploadedFiles/Associated_Docs/624414032755326387ADSP_2126x_HRM.pdf**. December 2005.

[3] Citizen. "HC49US". **http://www.citizencrystal.com/images/pdf/ms-hc49us.pdf**

[4] Motorola. "System Design and Layout Techniques for Noise Reduction in MCU-Based Systems." **http://cobweb.ecn.purdue.edu/~dsml/ece477/Homework/Fall2006/AN1259.pdf**

13.8 Software Design Considerations: 10

[1] Analog Devices, "SHARC Embedded Processor ADSP-21262," [Online Document], Aug. 2005, Available at HTTP: **http://www.analog.com/UploadedFiles/Data_Sheets/462724356ADSP_21262_b.pdf#search=%22data%20sheet%20ADSP-21262%22**

[2] Microsoft, "NTFS vs. FAT: Which Is Right for You?" [Online Document], Oct. 2001, Available at HTTP:

http://www.microsoft.com/windowsxp/using/setup/expert/russel_october01.mspx

[3] Cypress Semiconductor Corporation, “CY3663-EZ-OTG/EZ-Host Development Kit” [Online Document], Jul. 2004, Available at HTTP: **http://www.cypress.com/portal/server.pt?space=CommunityPage&control=SetCommunity&CommunityID=285&PageID=552&shortlink=DA_246716&REF=SCH&CID=ILC-shortlinks&shk=DA_246716&REF=SCH**

[4] Prof. Paolo Nesi, “OMR: Optical Music Recognition” [Online Document], Available at HTTP: **http://www.dsi.unifi.it/~hpcn/wwwomr/le.html**

[5] “MIDI is the language of the gods.” [Online Document] Available at HTTP: **http://www.borg.com/~jglatt/**

Appendix A: Individual Contributions

A.1 Contributions of Daryl Sielaff:

Early in the semester we determined that my strengths were related to system design and developing concepts to resolve know or upcoming issues. I began by laying out the general architecture of our project and researching parts. I took a very aggressive approach to the design of our system and made it very specialized to our purpose. I placed the SHARC DSP at the core of the design and relied on the throughput of the 8-bit parallel bus for data access to the USB host and external SRAM. Because of the nature of the design it led to difficulties developing code towards the end of the semester; we ended up rewriting many device drivers for the USB host and the MIDI chip.

As soon as I finalized my decision to use the ADSP-21262 from Analog Devices in late September, I requested samples of the SHARC processor, development boards and JTAG debuggers. Because of problems with Sales Channel Marketing staff being out-of-office, we didn't receive any of these parts until the last week of October and first week of November.

As I started designing the schematic I spent a majority of my time reading the datasheets for our parts so I would know exactly how they would interface together. In retrospect, our group should have had more than one team member involved in the design process from the start. I feel like we would have avoided several issues that arose in November and December. As soon as I finished the schematic I worked with Bill to design and inspect the PCB. I wasn't familiar with the Layout program so before routing I helped layout a plan for routing the parallel bus and then I double checked the board before it was submitted. Inevitably there were still errors on the PCB that I found after the board was fabricated.

Once the development kit for the SHARC was delivered I started developing the interface for the LCD with Ben. I wrote some simple "print()" and "outchar()" functions that made the menu code much easier to write. After the LCD was working without a glitch I turned my attention to our PCB that had just arrived. I analyzed the board and added a few flywires around the power section due to faulty routing and missing vias. I continued to populate the board while testing the sections and developing high level functions for accessing the external

SRAM as well as writing low level functions for the MIDI middleware and USB drivers. I spent over a week attempting to get clear results from the MIDI chip and I was never confident that it was communicating with the SHARC successfully. I did get a series of tones from the speaker but it wasn't the correct tones and it wouldn't work every time. I decided to drop the MIDI for the time being and start work on the USB host. Bill had been researching the USB standard and FAT filesystem so initially we worked together then split apart. Bill continued his work on the FAT filesystem while I developed and debugged the driver code for the USB host. We had simple code that would initialize the USB drive and communicate with the setup endpoint on the device; from this point on I had to write functions from scratch to communicate with a Bulk-only endpoint on the USB drive as specified in the numerous USB standards documents.

A.2 Contributions of Ben McQuiston:

We decided early on that we were going to use MIDI as the audio portion of the project. So right from the start of the semester, I began researching MIDI and looking for a MIDI processing audio chip. This research led to my development of the IMS to MIDI converter. The first version was a gcc compiled ansi C program that used file pointers to create actual MIDI files that could be played on a normal media player. Later in the semester I converted this code so that it would write to the external SRAM of our project board. Along with the converter, I had found an audio chip and got an evaluation board to start testing. I spent quite awhile trying to test it before we got our SHARC development board and after.

I was responsible for the packaging homework, which entailed me learning to use autocad in order to generate some designs for potential package ideas. I also was responsible for the reliability analysis homework. At around the same time, I worked with Dave to try to get a feel for the overall flow of the project software. After the initial MIDI conversion and homework's were finished, I moved on to the development of the menu.

In order to have a menu that could be tested and viewed, we needed a working LCD, a RPG, and push buttons. Once we finally got our development board, I worked on setting up the push buttons and RPG so that they would trigger interrupts when pressed or rotated. Typically menus and pushbuttons are trivial; however with the SHARC it was a bit more complicated because of the SRU. Dave and I spent awhile figuring out how to route pins using the SRU and how we could use DAI pins as interrupts. When we finally had pushbuttons and a working RPG, we had figured out the SRU and how to use interrupts on top of the intended goal of being able to test the menu. Daryl and I also spent some time getting the LCD to work via the SPI. Once all of this was set up, I worked on getting a menu with options and sub-menus. Once I had the menu done, I started handling the integration of the various code that we had for the various functions of the project.

With the menu completed, I began developing code that would connect the IMS to MIDI converter to whatever data Dave was creating using the image processing. Dave and I came up with a method for passing the music image data and so I wrote an IMS build function that found the MIDI number for a given position on the staff and wrote it and the length of the note to the IMS file. Similar to the IMS to MIDI converter I tested this using ansi c on a gcc complier and then converted it into a version for the SHARC. With all of IMS and MIDI functions, I had all the code that would move the image data all the way to a MIDI file. I set it up so that Dave had to call the IMS build function on each note and it would build an IMS file. Then once an IMS file was made, the IMS to MIDI code could be called.

After this I helped Will with the code for the FAT file system. We found some code and just had to modify a few functions on the lowest level, and once Daryl got communication with the USB we spent some time debugging so that we could read and write with the FAT system. This allowed for the IMS and MIDI code to be run on the SHARC by simply writing from the external memory to the USB drive.

A.3 Contributions of William McKenna:

During the first half of the semester my role on the team was to create the PCB for our device. I began by familiarizing myself with Orcad Layout PCB software and aided in component selection. As components were selected, I looked up PCB footprints in Orcad's library. All of the discrete components and a few of the larger components had standard footprints, but I created about 70 % of the large component footprints based on descriptions in data sheets. I did a preliminary layout, and then two versions of the PCB. The first version only got to about 80% completion when I realized that there was no easy way to lay out the PCB without creating timing issues for the memory and USB modules of the device. At that point Daryl and I talked and he created a new schematic with new pin outs. I created a new PCB and did not see any areas where timing was going to be an issue. I submitted the PCB for design review multiple times and changed it to eliminate errors. Daryl also aided me in this process by finding errors that were not found in the error finding software.

During the first half of the semester I was also assigned the task of doing the Patent Liability Analysis Homework and PCB Design Narrative. While doing the Patent Liability Analysis I was able to find information about Optical Music Recognition and passed that information on to David for use in the writing of the DSP software.

After submission of the PCB, I was given the task of researching the USB protocol and FAT file system. During this time I obtained intimate knowledge of both subjects. I was able to find drivers for the SL811HS USB host chip used on our device and edited it's functionality to work with our microprocessor. I then passed the project on to Daryl and educated him on USB. My next focus was the FAT file system. I began by writing functions in an attempt to code my own FAT driver. This was because I could not find a FAT driver that was written outside of Linux. Daryl was able to find some software for me and I began breaking it down into parts and understanding its operation. Ben was moved onto the project a short time after and I educated him on the FAT system. We tested the program together and worked on acquiring confirming the information that was returned from the USB was correct. Given that he had more experience programming the processor and I had more experience with the FAT and time was at a premium, he wrote functions for the driver to interface to our microcontroller, while I did testing of the FAT to locate problems and possible fixes.

Finally, I worked on documentation by writing the Product Description of the User Manual, editing my homework for the Final Report, and writing the 2 page senior design semester report. I also aided in packaging with the proposal and purchasing of the materials to cover our box.

A.4 Contributions of David Hartman:

This project stemmed from my interest in image processing, so my main contribution was creating the software for the Optical Character Recognition (OCR) for the images of sheet music. At the beginning of the year I helped research what design constraints the image processing would put on our project. I was responsible for the Software Design Narrative, in which I developed the flowchart and code module hierarchy for the overall design of the device.

I worked closely with Ben on this, which was important in determining how the image processing would interface with the LCD menu and the conversion to MIDI music files.

I was able to start on developing the image processing software right at the beginning of the semester with Matlab. I had taken EE 438, digital signal processing, but the class only gave me the basis for what I did. A lot of research and trial and error helped me to develop this software completely from scratch. I created a histogram function that makes a histogram of any section of the sheet music. I use the histogram function to make a histogram of the entire page of sheet music to find the staves, to make histograms of all the staves to find the notes, and to make histograms of all the notes to analyze them. I found this method of OCR called segmentation the most effective out of all the different methods I researched and tried.

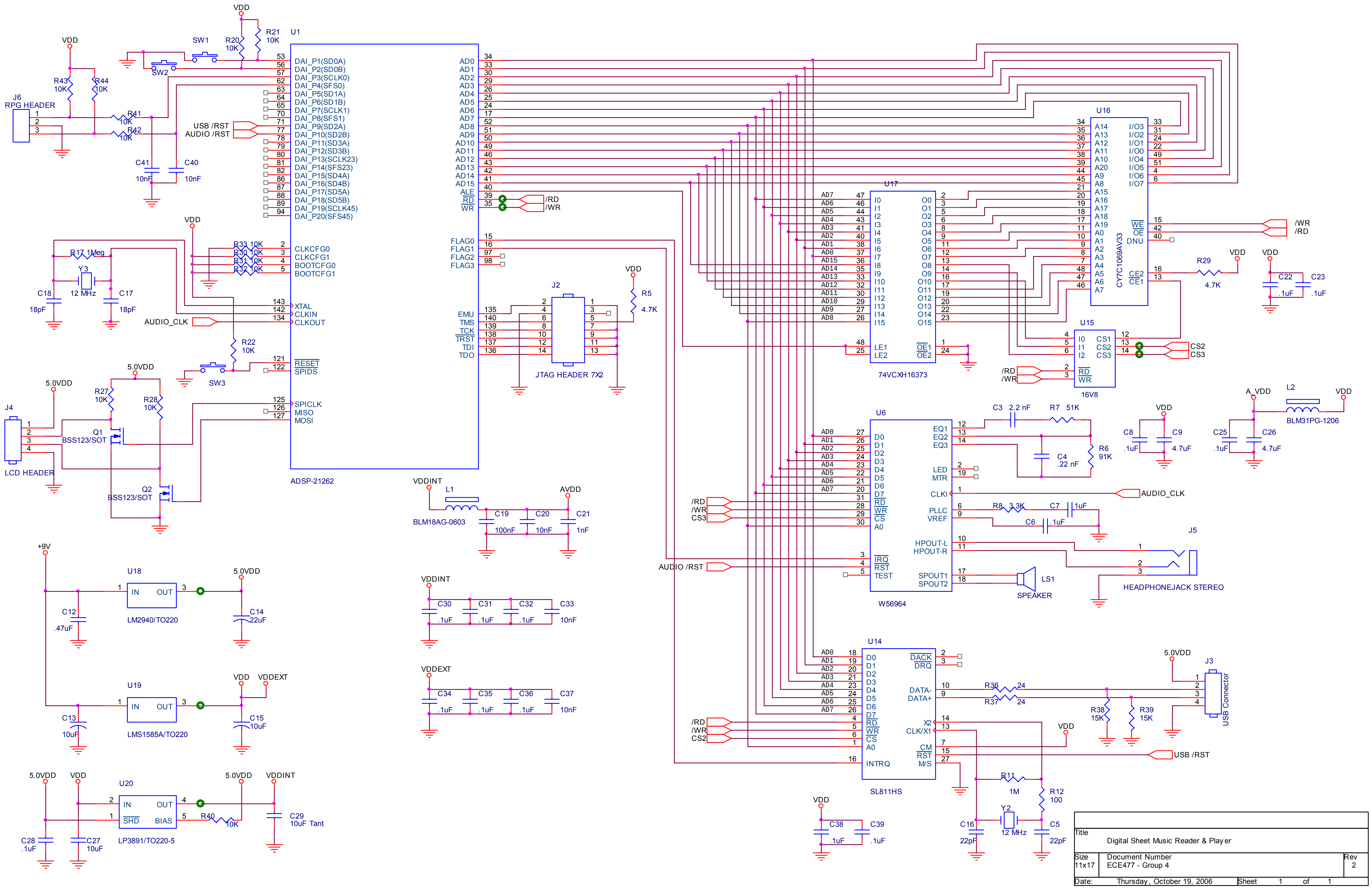
I created a note finder and a note analyzer function, which both call the histogram function. I made the software find all of the staff lines so that the note analyzer function can determine where the note is in relation to the staff lines. I also made the note analyzer function look for if the note has a stem and if the circle of the note is filled in or is hollow to determine the length of the note. I interfaced the image processing to the Ben's code that develops the MIDI file by returning numeric values to represent each note.

Once I completed developing the software in Matlab, I had to convert it into C to run on our microprocessor. This required that the code would work with the 32-bit architecture of the microprocessor. The software also needed to leave all large arrays on the external memory, and I could not take up too much space with lots of extra variables.

Outside of work on the image processing code, I made contributions with a few other software applications. I worked with Ben to develop the DAI interrupts, which the pushbuttons and rpg are routed to. Ben and I also developed the software for interpreting the signals from the rpg. The USB needs delays of a given time period, so I set up the timer interrupt subroutine and a function that Will could adapt into the USB code as a delay function.

The program code is stored on an external SPI flash chip, and I researched and learned how to adapt some software to our device to program our code to the flash. Our code takes up so much space that I had to make the flash programmer software store our code on the external memory, then bring it onto the microprocessor, and then program it onto the flash.

Appendix C: Schematic



C.1- Schematic for the Digi-Sheet Music Reader and Player

Appendix E: Parts List Spreadsheet

Bill Of Materials December 10,2006 14:17:47 Page1

Item	Reference	Quantity	Part
1	C3	1	2.2 nF
2	C4	1	.22 nF
3	C5,C16	2	22pF
4	C6,C8,C19,C22,C23,C25,C28, C30,C31,C32,C34,C35,C36, C38,C39,C43,C44	16	.1uF
5	C7	1	1uF
6	C9,C26	2	4.7uF
7	C12	1	.47uF
8	C13,C15,C27	3	10uF
9	C14,C42	2	22uF
10	C17,C18	2	18pF
12	C20,C33,C37,C40,C41	5	10nF
13	C21	1	1nF
14	C29	1	10uF Tant
	J1	1	3 Pin KK156 Power Connector
15	J2	1	JTAG HEADER 7X2 KK100
16	J3	1	USB Connector
17	J4	1	5 Pin KK100 LCD HEADER
18	J5	1	HEADPHONEJACK STEREO
19	J6	1	3 Pin KK100 RPG HEADER
21	L1	1	BLM18AG-0603
22	L2	1	BLM31PG-1206
23	R5,R29,R45,R46	4	4.7KΩ
24	R6	1	91KΩ
25	R7	1	51KΩ
26	R8	1	3.3KΩ
27	R11,R17	1	1MΩ
28	R12	1	100Ω
30	R20,R21,R22,R30,R31,R32, R33,R40,R41,R42,R43,R44	12	10KΩ
31	R36,R37	2	24Ω
32	R38,R39	2	15KΩ
33	SW1,SW2,SW3	3	SW PUSHBUTTON
34	U1	1	ADSP-21262 - 32-bit DSP SHARC
35	U6	1	W56964 - 64-Note Polyphonic Ringtone Chip
36	U14	1	SL811HS - USB Host Controller
37	U15	1	16V8
38	U16	1	CY7C1069AV33 - 2MB Static RAM
39	U17	1	74VCXH16373 - 16-bit D-type Latch
40	U18	1	LM2940/TO220 - 5V LDO Regulator
41	U19	1	LMS1585A/TO220 - 3.3V LDO Regulator
42	U20	1	LP3891/TO220-5 - 1.2V LDO Regulator
43	U21	1	AT25F4096 - 4Mb SPI Flash
44	Y2,Y3	2	12 MHz Crystal
45		1	CrystalFontz LCD