**Musical Piano: Team 19**



|  |  |  |
| --- | --- | --- |
| *Team Members (left-to-right on picture, above)* | *Class No.* | *Lab Div* |
| Sid Parida | 7152-P | 4 |
| Kylie Broton | 8154-B | 6 |
| Mark Brooks | 8852-B | 7 |
|  |  |  |

|  |  |
| --- | --- |
| *Report/Functionality Grading Criteria* | *Points* |
| Originality, creativity, level of project difficulty | 20 |
| Technical content, succinctness of report | 10 |
| Writing style, professionalism, references/citations | 10 |
| Project functionality demonstration | 20 |
| Overall quality/integration of finished product | 10 |
| Effective utilization of microcontroller resources | 10 |
| Significance of individual contributions\* | 20 |
| *Bonus Credit Opportunities* | *Bonus* |
| Early completion | 0.5% |
| PCB for interface logic | 2% |
| Poster (required for Design Showcase participation) | 1% |
| Demo video (required for Design Showcase participation) | 1% |
| Design Showcase participation (attendance required)\* | 1% |

##### \**scores assigned to individual team members may vary*

|  |  |
| --- | --- |
| *Grading Rubric for all Criteria (Including Bonus)* | *Multiplier* |
| *Excellent* – among the very best projects/reports completed this semester | 1.0 - 1.1 |
| *Good* – all requirements were amply satisfied | 0.8 - 0.9 |
| *Average* – some areas for improvement, but all basic requirements were satisfied | 0.6 - 0.7 |
| *Below average* – some basic requirements were not satisfied | 0.4 - 0.5 |
| *Poor* – very few of the project requirements were satisfied | 0.1 - 0.3 |

**TABLE OF CONTENTS**

|  |  |
| --- | --- |
| 1.0 Introduction | 1 |
| 2.0 Interface Design | 2 |
| 3.0 Peripheral Utilization | 3 |
| 4.0 Software Narrative | 4 |
| 5.0 Packaging Design | 5 |
| 6.0 Summary and Conclusions | 6 |
| 7.0 References | 7 |
| Appendix A: Individual Contributions and Activity Logs | 8 |
| Appendix B: Interface Schematic and PCB Layout Design | 17 |
| Appendix C: Software Flowcharts | 19 |
| Appendix D: Packaging Design | 21 |

1. **Introduction**

The Musical Piano was a culmination of our efforts for the ECE 362 Mini-Project for Fall 2015. Inspired by a desire to create a music related device, we agreed on building a 13-key, one octave piano with recording and playback features.

# When the piano starts up, it starts in freeplay mode with the pitch set to 4 which is the normal pitch. The LCD in freeplay mode displays the phrase, “Free Play” followed by a pitch number, tempo number, and the note(s) being played whenever they are being played. From there you can start playing notes, up to two at a time, one for each speaker. Anytime at least one note is pressed, the green led is lit up, and whenever a note isn’t played, the green led is no longer lit. If more than two notes are played, the highest two notes are the ones the speakers play. You can change the pitch with the pitch potentiometer to make the notes higher or lower. You can also switch to the other two modes using the mode select button. The next mode is record mode.

Once in record mode, the LCD display changes to display, “Record” and the tempo value is changed with the song value which is the location in memory where the song is recorded. From here, notes can still be played, the notes will still appear on the lcd and the green light will still light up when the notes are played, however once the start button is hit, the message will change to “recording”, and a display of “L” followed by a count of how long the song is will occur. A song can go up to a count of 9 and will stop recording automatically once 9 is hit. If the recording is stopped before 9 is hit, and start is hit again, it will start where it left off. The yellow led is lit while recording.

Once in playback mode, the LCD display changes to display “Play” and the tempo option comes back. From there, using the song option, you can choose which song to play, and also change the pitch and tempo while playing it back. Press the start button to start the song, where the display changes to “Playin” and the L counter starts up. From there it plays the song recorded, and the red led lights up while playing, and the green led lights whenever a song is played.

Sid Parida was the team leader and Software Lead, responsible for most of the coding. Mark Brooks was the Interfacing Lead responsible for circuit assembly, while Kylie Broton was the Packaging and Documentation Lead, responsible for circuit debugging and final packaging.

1. **Interface Design**

Our Pianos interface consists of the following external interfaces. Thirteen push buttons for the 13 notes in an octave, 3 push buttons for mode, song option, and start/stop, 2 potentiometers to vary between the pitch and tempo, a reset button, an on/off switch, three LEDs for record mode, playback mode, and freeplay mode, an LCD for various information, and two speakers for up to two notes played at a time.

Our microcontroller is connected to our GAL22v10 which is connected to our LCD in a series in parallel out fashion. The SPI’s Clock pm, (PM5), and the MOSI pm (PM4) are the clock and input to the GAL respectively while pins 14 through 21 of the GAL connect to pins 0 through 7 of the LCD. PTT2, 3, and 4 of the microcontroller are connected to the LCD register select, read/write, and clock respectively.

We’ve got 13 push buttons chained to two 74HC165N configured in a parallel in series out fashion, where the output of 1 is an input of the other and the output of the other goes to PORTAD pin 2. Both shift registers are clocked using PTT7, strobed using PTT6, and the clock enabled with PTT5. We also have 2 potentiometers where ATD conversions are performed on them and 3 push buttons are connected to PORTAD pins 3, 4, and 5 respectively as normal inputs.

PWM pins 0 and 1 are connected to the speakers, PTT pins 5, 6, and 7 are used to clock the 74HC165N shift registers. PM pins 0, 1, and 3 are configured as normal output pins that vary based on software and are used to light the record, play, and note LEDs. PTT pin 3 is used for the R/W’ pin on the LCD which is active low so that way it’s always in write state. PTT pin 4 is used for LCDCLK which is pulled high so pin 4 is always 1. PTT pin 2 is used for the register select line which is used to send instructions to the LCD such as CURMOV, LCDON, LCDCLR, etc.

# **3.0** **Microcontroller Resource Utilization**

Our pitch and tempo switches utilized our ATD peripheral. Both options can go from 0 to 9 so we felt that potentiometers would be the perfect use for the ATD. We used a conversion length of 2, non-scan, non-fifo mode with 8 bit resolution, nominal sampling aperture (2 ATD clocks), and a max ATD clock frequency of 2 MHz. When writing to ATDCTL5, we would sample multiple channels starting with input channel 0. Anytime an ATD conversion occurred and the status register was ready, the pitch and tempo would be the results divided by 26 giving a value from 0 to 9.

For our SPI interface, we use a half-duplex mode with the LCD to display various information such as the mode being used, the notes being played, etc. We use master mode, active high, odd edges, no slave select, most significant bit shifted first. We don’t use mode fault or bi-directional mode, and we use a 6 Mbs baud rate. Half duplex made the most sense since it was simpler, we had a lot of experience with it from lab, and we had available input on the 9s12c32 for the shift registers from the push buttons.

In order to play notes at certain pitches, we used the PWM. We used channel 0 and 1 with active high polarity, 8 bit left aligned mode, max period, Clock A with prescalar of 2 resulting in a PWMPRCLK of 12 MHz, and an initially clear duty period. In the TIM\_ISR, both PWMDTY values are changed based on a freq\_mod variable where the duties are alternated between 2 and -2 halfway between the freq\_mod variable. Therefore the duty changes are the frequency variable changes. The freq\_mod variable is set to specific values that correspond to specific notes such that when a certain button is pressed, that freq\_mod value will be chosen, which will in turn affect the duty value and be output from the microcontroller into the speakers and you will be able to hear the sound played. The pitch change option will also affect the freq\_mod variable. Our project utilizes the TIM peripheral for a lot of different things such as the different modes, tempo, pitch, record change, etc. We use CH7 along with output compare and set TCNT to reset when OC7 occurs to generate interrupts . Our prescalar is 16 and TC7 is 150 resulting in a 0.1 ms interrupt. Changes in tempo, pitch and recording are utilized by checking flags and doing appropriate actions based on the flags set in the timer ISR.

Our project utilizes the RTI for the mode, start and option push buttons. Our RTI has a 2.048ms interrupt rate, and we use PORTAD pins 3, 4, and 5 for the mode, start and option push buttons respectively. We did not utilize any SRAM or Flash memory in our project.

**4.0 Software Narrative**

The musical piano project used PWM for actual note generation. Based on a interrupt driven system, the code for generating the musical notes was placed in the interrupt handling function for the TIM module which is triggered every 0.1 ms. A counter was set up, which increased by certain number (pitch factor) in every cycle of the interrupt. The counter was moded by a number called freq\_mod every cycle too For half the length of the cycle, the duty of the PWM pin was set to 2, while for the other half the duty was set to -2. This resulted in the generation of an oscillating square wave whose frequency could be changed by varying pitch factor or freq\_mod. Increasing freq\_mod decreased the frequency, while increasing pitch factor increased the frequency.

The values of freq\_mod for different notes in an octave were determined by experiment and calculations based on frequencies of actual piano notes. The value of the 8 bit representation of the pitch knob divided by 26 was used as the pitch factor. Therefore if the pitch knob value was increased it increased the frequency and hence the pitch. The buttons were scanned in the main loop and the display was updated in the main loop too. The piano control pushbuttons (Start, Stop, Song Selection) were sampled and debounced in the RTI interrupt function (every 2.048 ms) and flags were set to indicate button presses. These flags were handled in the main loop before sampling the push buttons.

To handle multiple key presses an algorithm was developed to scan through all 13 buttons and look for two key presses, then two freq\_mods were assigned values based on the specific key presses. If no keys were pressed, the freq\_mods were assigned to 1 hence effectively eliminating oscillation. This was essentially the algorithm for the free play mode which basically played a note based on user input on the pushbutton protoboard.

For record mode, two arrays were used to record each song, as the piano allowed two key presses at the same time. The piano also allowed for two different song locations. The record mode worked in a similar way to free play, except the addition of a new record note flag. Every Time a fixed number of interrupts had accumulated, the current notes being played were recorded into memory. The playback mode worked in a similar manner and played the notes back at evenly spaced periods. The period was made small enough to eliminate granularity in recording and playback. The displays and LEDs were updated based on various button states and mode options. However, the period could be changed during playback, using the tempo knob.

**5.0 Packaging Design**

The basic design starts out with a black fancy electric piano. The black box is actually a scrapbooking box that we purchased from Hobby Lobby. We then thought of how we could display keys while keeping all the wiring inside the one console. We thought about using a wooden box, but everything Hobby Lobby sold was too small and did not have enough surface area to design an appealing face with plenty of room for as many push buttons as needed and to house our multitude of breadboards and components.

To start out, we first poked holes for all the components (pushbuttons, potentiometers, LCD screen, switch, LEDs). Each of the piano keys or pushbuttons, we poked 13 holes to set it up but we quickly found out that it was difficult to attach the face of the piano to the inner components if each of the face components were attached to the top of the box first. We decided to solder the keys (pushbuttons) to a board and leave it with an open display, which really shows the integrity of the product.

The facial design of the actual piano and not the inside components basically came down to simplicity. We decided to space everything out and segmented meaning the buttons were distributed by type. The red push buttons with functionality were to the right of the LCD because they needed to be easily accessible, as well as the potentiometers for tempo and pitch. They are all equally important in terms of usage. The ON/OFF button and reset are not used as often as the other buttons on the board, so they are further away from the user. Lastly, the speakers and LEDs, and LCD screen and are the furthest because those are not to be touched by the user at all. The LCD is the most important for the user to know what he/she is doing, thus is it located at the center of the piano. The speakers are only to be heard and the LEDs are only to be briefly seen, thus those are located the absolute furthest from the user’s touch.

**6.0 Summary and Conclusions**

Working in a group is probably the most important skill we learned while completing the project. While there are a lot of technical skills to be mastered through the creation of this project, the amount of logistics, planning, and timing was the more time consuming part of the project. If we were to start all over again, we would first, plan the ordering of the printed circuit board earlier because we could have saved a significant amount of money. Also, each of the technical parts was split up based on availability and preferences. All of this was decided logistically.

We also learned how to really code and think about creating project in its entirety, from the beginning of ideas to the coding and interfacing to the end with documentation. We learned that coding and interfacing go hand in hand with a microcontroller project design. It made our class curriculum more concrete for future use in senior design or future classes.

If we had more time to improve the design, we would not want to change the code, besides maybe adding in a couple secret sequences of keys that would create an LCD display that said the name of the song or other phrases. We would really like to redesign the exterior or face of the piano. We would like to have actual keys that look like pianos rather than push buttons. Also, if we had unlimited money, we would have actually liked to buy a mini children’s piano and put our components inside to replace the manufacturers electronics. We would also add more keys to fit an entire children’s piano. If we had unlimited amount of time as well as money, we would like to recreate and reorder the printed circuit board to essentially replace all of the breadboards. Overall, we are very proud of our project and excited to show it off at Spark Challenge.

**7.0** **References**

[1]ATD Converter Block User Guide, 1st ed. Motorola, 2003 [Online]. Available: *https://engineering.purdue.edu/ece362/Refs/9S12C\_Refs/S12ATD10B8CV2.pdf*. [Accessed: 11- Dec- 2015]

[2]SPI Block Guide, 1st ed. Motorola, 2003 [Online]. Available: *https://engineering.purdue.edu/ece362/Refs/9S12C\_Refs/S12SPIV3.pdf*. [Accessed: 11- Dec- 2015]

[3]TIM\_16B8C Block User Guide, 1st ed. Motorola, 2003 [Online]. Available: *https://engineering.purdue.edu/ece362/Refs/9S12C\_Refs/S12TIM16B8CV1.pdf*. [Accessed: 11- Dec- 2015]

[4]PWM\_8B6C Block User Guide, 1st ed. Motorola, 2003 [Online]. Available: *https://engineering.purdue.edu/ece362/Refs/9S12C\_Refs/S12PWM8B6CV1.pdf*. [Accessed: 11- Dec- 2015]

[5]Module Mapping Control User Guide, 1st ed. Motorola, 2003 [Online]. Available: *https://engineering.purdue.edu/ece362/Refs/9S12C\_Refs/S12MMCV4.pdf*. [Accessed: 11- Dec- 2015]

[6]CRG Block User Guide, 1st ed. Motorola, 2003 [Online]. Available: *https://engineering.purdue.edu/ece362/Refs/9S12C\_Refs/S12CRGV4.pdf*. [Accessed: 11- Dec- 2015]

[7]S12CPUV2 Reference Manual, 1st ed. Motorola, 2003 [Online]. Available: *https://engineering.purdue.edu/ece362/Refs/9S12C\_Refs/S12CPUV2.pdf*. [Accessed: 11- Dec- 2015]

[8]GAL22V10 Data Sheet, 1st ed. Lattice Semiconductor Corporation, 1997 [Online]. Available: [*http://web.mit.edu/6.115/www/document/gal22v10.pdf*](http://web.mit.edu/6.115/www/document/gal22v10.pdf)*.*  [Accessed: 11- Dec- 2015]

[9]74HC165N Data Sheet, 1st ed. NXP (Philips), 2008 [Online]. Available: [*http://www.nxp.com/documents/data\_sheet/74HC\_HCT165.pdf*](http://www.nxp.com/documents/data_sheet/74HC_HCT165.pdf). [Accessed: 11- Dec- 2015]

**Appendix A:**

**Individual Contributions**

**and**

**Activity Logs**

**Activity Log for:** Kylie Broton **Role:** Packaging & Documentation Leader

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Activity*** | ***Date*** | ***Start Time*** | ***End Time*** | ***Time Spent*** |
| Purchasing packaging/components | dec 1 | 3 pm | 5 pm | 2 hrs |
| Creating packaging | dec 2 | 5 pm | 6 pm | 1 hr |
| Creating packaging | dec 3 | 3 pm | 5 pm | 2 hrs |
| Packaging (components into package) and realized it will not work | dec 3 | 7 pm | 12 pm | 5 hrs |
|  |  |  |  |  |
|  |  |  |  |  |
| Soldering | dec 3 | 2 pm | 8 pm | 6 hrs |
| Packaging (components into package) | dec 3 | 8 pm | 9 pm | 1 hr |
| Packaging (components into package) | dec 4 | 3 pm | 6 pm | 3 hrs |
| PCB design help | dec 4 | 6 pm | 11 pm | 5 hrs |
| Ordering the pcb online/ rewiring without PCB | dec 4 | 7pm | 8pm | 1 hr |
| Powerpoint/poster | dec 5 | 12 pm | 4 pm | 5 hrs |
| Software flowchart help | dec 5 | 2 pm | 3 pm | 1 hr |
| Soldering | dec 6 | 3 pm | 4pm | 1 hr |
| Soldering PCB | dec 9 | 3 pm | 6 pm | 3 hrs |
| Packaging with PCB | dec 10 | 3 pm | 4 pm | 1 hr |
| Video | dec10 | 6 pm | 8 pm | 2 hrs |
| Paper | dec 10 | 8pm | 11 pm | 3 hrs |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**Written Summary of Technical Contributions: Kylie Broton**

I did not think I would be able to participate in the technical aspects of this project as I really did. I was able to do most of the soldering and packaging. I also was able help with the printed circuit board design and installation. I would say I did most of the soldering, especially on the printed circuit board. I could have never done this all by myself, my teammates were incredible helpful and compliant on everything we completed together.

I was the brain behind debugging as well. I was able to point out a lot of the hardware bugs including, but not limited to, the battery dying, unplugged wires and unfinished soldering. I soldered the printed circuit board, random wires together, a lot of little wire components that we did not actually end up using.

For the design on the face or surface of the box, I was able to punch holes in the box. But like stated above the punched holes for the pushbutton keys were then changed into a rectangle cut in the black box for the entire circuit board to be displayed. I had to figure out how there could be pressure from the users touch on the keys or other pushbuttons without it breaking, so that was one of the solutions to that issue. The other push buttons were bigger so they are stabilized on the box by the nuts and bolts on the original push button. Everything else is held by gravity and insulated electrical tape.

The last thing I helped with was software flow chart. I was able to put the chart together with the help of the software design lead, Sid. While I was not contributing much to the coding, I understand it and spent time on designing the packaging and helped with a lot of the documentation as well as making sure the project is done on time and extra credit opportunities are achieved on time. I would say my most important contribution would be making sure the printed circuit board design was finished and ordered on time to not rake up a bill larger than $140 (without shipping). Timing is everything.

**Activity Log for:** Mark Brooks **Role:** Interface Leader

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Activity*** | ***Date*** | ***Start Time*** | ***End Time*** | ***Time Spent*** |
| Buying parts, intial setup, getting sound to work | 11/21 | 4 pm | 11 pm | 7 hours |
| getting push buttons setup, debugging | 11/23 | 2pm | 7 pm | 5 hours |
| more debugging, pcb rough draft | 11/26 | 2pm | 11 pm | 9 hours |
|  |  |  |  |  |
| purchasing packaging/components | 12/1 | 3 pm | 5 pm | 2 hrs |
| creating packaging/ rewiring | 12/2 | 5 pm | 6 pm | 1 hr |
| creating packaging/rewiring | 12/3 | 3 pm | 5 pm | 2 hrs |
| more packaging | 12/3 | 7 pm | 12 pm | 5 hrs |
| wiring help / pcb design | 12/3 | 2 pm | 8 pm | 6 hrs |
| packaging help | 12/3 | 8 pm | 9 pm | 1 hr |
| pcb design | 12/4 | 3 pm | 6 pm | 3 hrs |
| pcb design | 12/4 | 6 pm | 7 pm | 1 hr |
| ordering the pcb online | 12/4 | 7pm | 8pm | 1 hr |
| powerpoint/poster | 12/5 | 12 pm | 2 pm | 5 hrs |
| packaging with pcb help | 12/10 | 3 pm | 4 pm | 1 hr |
| video | 12/10 | 6 pm | 8 pm | 2 hrs |
| Final Report | 12/10 | 8pm | 11 pm | 3 hrs |
| Spark Challenge | 12/11 | 3 pm | 6 pm | 3 hours |

**Written Summary of Technical Contributions:** Mark Brooks

I was able to contribute a lot during the early stages of the project. Sid and I went out and bought

all of the materials needed to get started such as speakers, extra breadboards, wires, buttons, etc.

I helped wire the first prototype of the board going from one push button to the PTT5 input, and

then going out as a specific frequency on the pwm output. I researched the various frequencies

that corresponded to specific notes so that way we could define them in code as A, B flat, C etc. I

helped debug a few issues such as that we weren’t getting sound because our TIM was

initialized incorrectly, and another time we thought we weren’t getting sound when we were but

the pitch was too low to hear. I was able to figure out that we were getting two false inputs, one

from an incorrect wiring and another from a faulty pin on our breadboard that required us to shift

our shift register over on the breadboard. My main contribution was designing the PCB. During

the first draft that Sid and I worked on, I realized that there was too many things on our PCB and

that it would either require more layers, or shorter traces which wouldn’t be very practical. We

also had the issue of the ground layout being very poor so on the second draft, I put less things

on there so that way we would have plenty of space and no issues. Sid and I consulted one of the

TA’s and he recommended we start from scratch. So I redid the schematic and the board with

less on it. I followed recommendations such as using 135 degree angles instead of 90 degrees,

having wide ground traces, using decoupling capacitors, putting the 9s12c32 in the center, etc. I

made sure that same layer paths didn’t cross, triple checked the connections were right, then Sid

and I printed out the circuit in 1x1 scaling to make sure it was the right size, and he uploaded it

on a checker site to make sure there weren’t any showstoppers, and after that, let Kylie handle

the ordering. I made suggestions on the packaging such as which box to get, what material to

use, etc. I helped out with cutting and stripping wires as well as testing the connections to make

sure they were sturdy during the packaging phase.

**Activity Log for:** Sid Parida **Role:** Software Lead (Team Leader)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Activity*** | ***Date*** | ***Start Time*** | ***End Time*** | ***Time Spent*** |
| Initial parts shopping, setting up pwm to generate notes, Fur Elise | 11/21 | 4 pm | 11 pm | 7 hours |
| Initial attempt at making 74HC165N shift registers to work with 13 pushbuttons | 11/23 | 4 pm | 11 pm | 7 hours |
| Code free play mode along with two button simultaneous play | 11/27 | 2 pm | 11 pm | 9 hours |
| Code record mode & playback mode along with circuit completion | 11/28 | 6 pm | 12 pm | 6 hours |
| Complete coding along with display updates | 12/2 | 3 pm | 9 pm | 6 hours |
| PCB Design | 12/4 | 3 pm | 7 pm | 3 hrs |
| Ordering the PCB online | 12/4 | 7pm | 8pm | 1 hr |
| Creating packaging/ rewiring | 12/4 | 5 pm | 11 pm | 6 hours |
| Creating packaging/rewiring | 12/5 | 1 pm | 11 pm | 9 hours |
| Final coding changes along with led updates | 12/5 | 11 pm | 3 am | 4 hours |
| Soldering note pushbuttons protoboard | 12/6 | 12 pm | 6 pm | 6 hours |
| Final packaging and rewiring without PCB | 12/6 | 6 pm | 12 am | 6 hours |
| Final packaging with PCB | 12/10 | 1 pm | 4 pm | 3 hours |
| Poster/Report/Video | 12/10 | 6 pm | 11 pm | 5 hours |
| Spark Challenge | 12/11 | 3 pm | 6 pm | 3 hours |

**Written Summary of Technical Contributions:** Sid Parida

On the first night, Mark and I went out and bought materials like speakers, shift registers,

breadboards, wires, buttons, etc. from Radioshack. Mark and I worked on the initial wiring for

the first prototype with one push button going to the PTT5 input, and coming out as a certain

frequency on the PWM. Mark researched the different frequencies for different notes and once we

had the notes defined in code as well as lengths (half note, quarter note, etc), we wrote the song

“Fur elise” in main.

After debugging a few issues in software, I worked on wiring up the thirteen

push buttons so that we could play thirteen keys as well as work on the code that would be able

to distinguish which notes were being played. Once the circuit was debugged some more, I wrote

the remainder of the code which included the free play record, and playback modes along with

updates to the LCD. Mark and I worked on the first draft of the PCB design and after Mark finished

the final draft, I checked it by printing it in 1x1 scale and uploading it to an online checker site.

During the packaging stage, I worked on transferring certain things from the circuit to the top of

the box and rewiring and debugging to make sure the circuit still worked in the transition. I

worked on soldering the 13 notes to a protoboard as well as making some final changes to the

code.

Before the PCB arrived, Kylie, Mark and I worked on the final packaging of the piano

connecting everything more cleanly, making sure everything worked without any bugs, there

were no loose connections, etc. After demonstrating it to Professor Meyer, once our PCB came

in, we rewired certain parts to accommodate for the PCB and once again made sure that

everything was working fine. Once the piano was working, we demonstrated it again with the

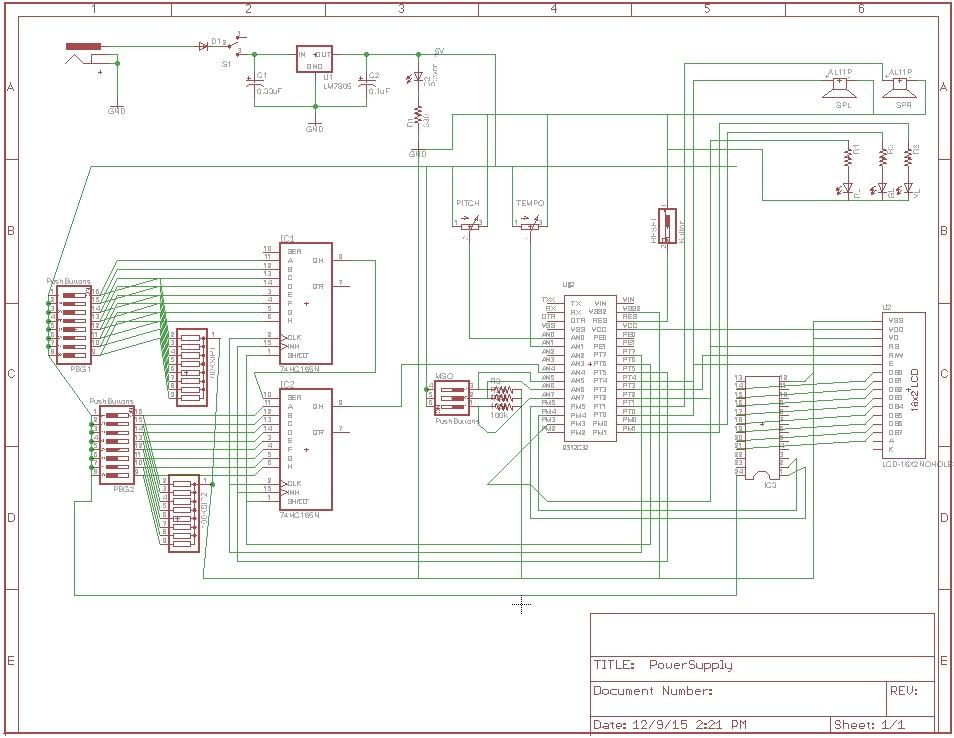
PCB and then afterwards, Kylie and I worked on the software flowchart for the presentation.

**Appendix B:**

**Interface Schematic**

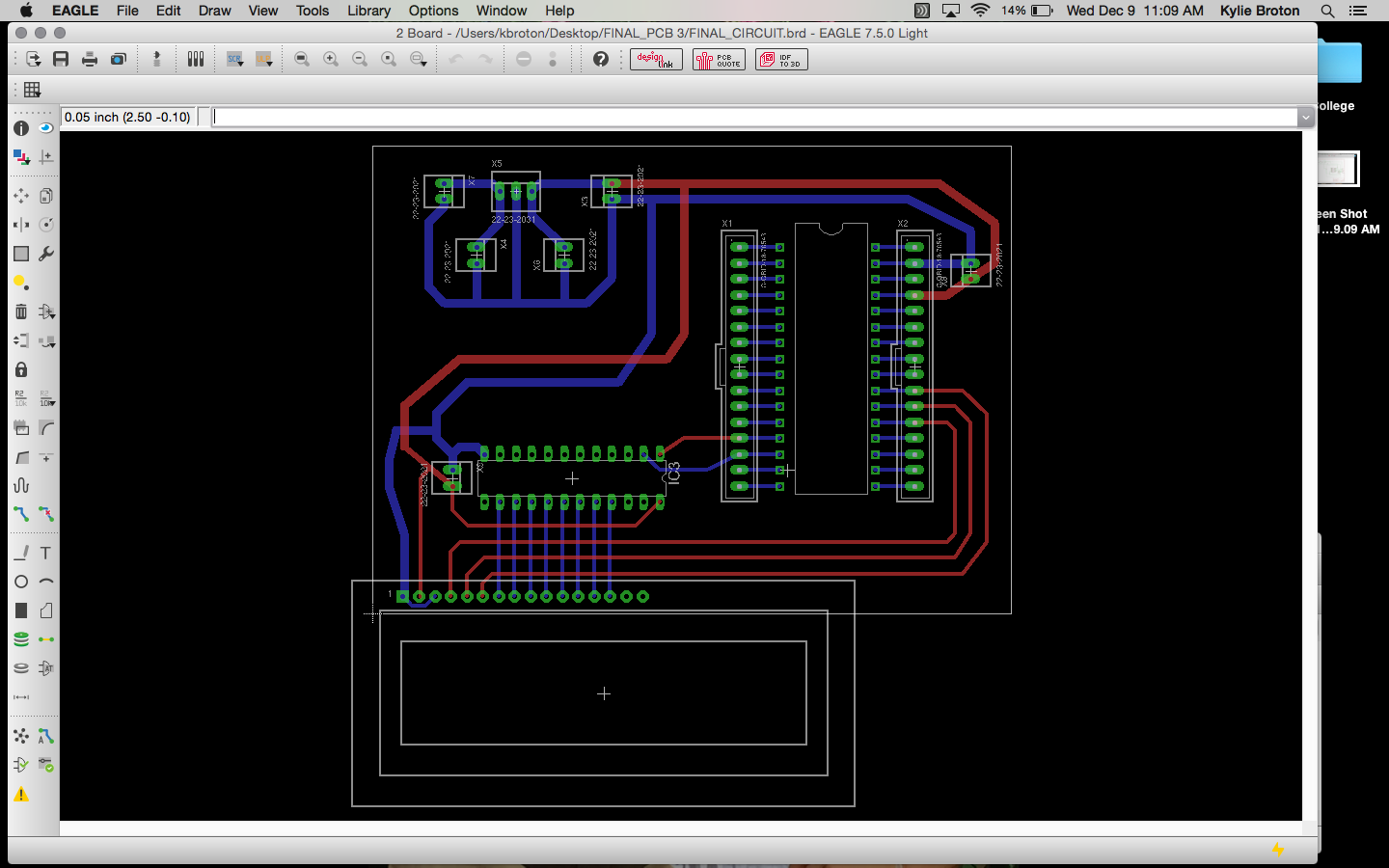
**and**

**PCB Layout Design**



*Eagle Schematic for Project*

*(Team Members Responsible: Sid Parida, Mark Brooks)*

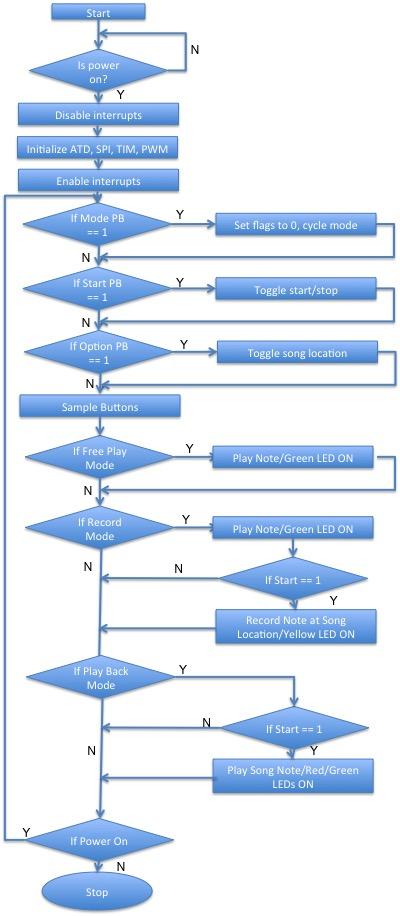


*Printed Circuit Board Design*

*(Team Members Responsible: Sid Parida, Kylie Broton, Mark Brooks)*

**Appendix C:**

**Software Flowcharts**



*Software Flowchart*

*(Team Members Responsible: Kylie Broton, Sid Parida)*

**Appendix D:**

**Packaging Design**



*Packaging*

*(Team Member Responsible: Kylie Broton)*