ECE 362 *Microprocessor System Design and Interfacing* Fall 2016

# Project BEATS: Team 15


| *Team Members (left-to-right on picture, above)* | *Class No.* | *Lab Div* |
|---|---|---|
| Hengyi Lin | 9018-L | 2 |
| Chia-Hua Peng | 2220-P | 5 |
| Paul Swartz | 4017-S | 10 |
| Suprith Ramanan | 0536-R | 6 |

| *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

The designed product is a music-based tap-game. We implemented this on three columns of 8 LEDs and an LCD. A predetermined tune plays on a thirty-second loop as the game begins, and LEDs begin to light up according to the rhythm of the music at the start of each row. The LEDs light up in sequence down a row, and a pushbutton is placed to correspond to each row. If the button is pushed within a trigger period, the points are earned, and displayed on the LCD. The LCD also displays the time left in the game. Start/Stop and Reset pushbuttons are also defined, along with a Potentiometer that adjusts the speed of the game. Any game-speed changes also change the speed of the generated music, to play in harmony with the game. The total points earned by the player during their session will be totaled and displayed on an LCD at the top of the game, along with a game timer. As soon as the timer hits zero, the game is over, and the total points are shown. If the player correctly hit a light, five points are awarded. If the player miss a tap or mistakenly click a tap, one point is deducted.

In this project, despite all of us aiding each other, the group members focused on certain parts. Chia-Hua, as a Peripheral Leader, coordinated the peripheral layout and the microcontroller resource utilizations, programmed and tested the music generation algorithms as well as the LED/LCD shifting logics, integrated the software and hardware aspects of the design, and bug tracked software and hardware issues. He also designed the exterior layout of the final product, assembled the wiring connections, and worked as an editor/ producer for the video. Hengyi, as a Software Leader, programmed and tested the overall game logic, interpreted the music notes into digital representations, encoded the LED shifting patterns and the music rhyme, and interfaced a breadboard prototype for bug tracking. He also worked as an instrumentation connector technician, soldering the wires and the peripherals, to enhance the mechanical robustness of the final product. Paul, as an Interface Leader and the PCB Honor, designed the schematic as well as the PCB layout, bug tracked hardware issues, assembled the hardware components, and wired up the interior part of the final product. He also managed the budget and the shipping for the PCB and designed the poster. Suprith, as the Team Documentation Package Leader, worked on the technical write-up of the project, documented the game logic flowchart, and provided help on budgeting/ ordering the hardware and assembling the final product. He is also the director, script writer, and the narrator for the video.

## 2.0 Interface Design

The external interfaces utilized in this project included 24 LED's with inbuilt resistors, an LCD display, a speaker with inbuilt amplifier and low-pass filter, a potentiometer, 5 pushbuttons, and 4 GAL devices (originally we planned on using 5, but modifications were made after the PCB was ordered).

The five GAL devices were used as shift registers. One of the GALs received serial data from the SPI MOSI pin (PM4) and outputted the data to the LCD display in parallel. The other 4 GALs were daisy-chained together to create one long shift register. The first GAL in this multi-GAL shift register received serial data from the SPI MOSI pin and outputted the data, in parallel, to the first 8 LEDs. Each of the other GALs in this shift register received their serial input data from the most significant bit of the previous GALs output data. All 5 GAL devices were clocked by a single signal--the SPI SCK pin (PM5). A single slave select wire was connected to all 5 GAL devices, in order to allow the microcontroller to communicate whether the SPI is sending data to the multi-GAL shift register or the LCD GAL. If the slave select output is high, the multi-GAL shift register receives data from the SPI module. If the slave select output is low, the LCD GAL receives the data.

Five pushbuttons were included to allow user input. Each of these pushbuttons were connected in active-low arrangement. A 10Kohm SIP was used to connect pushbutton output to 5V. When a button was pressed, the pushbutton acted as a momentary closure to ground. The 10Kohm SIP was implemented after the PCB was ordered.

The speaker used to produce the music possessed a 5V input, a ground input, and an audio input. The audio input was optimized on the PCB to be of minimal length and to not cross paths with any other signals.

The 10Kohm potentiometer used to control the tempo of the music possessed a 5V input, a ground input, and an output. The potentiometer output trace was connected to the ATD module on the microcontroller (AN0), and its trace width on the PCB was made slightly larger than normal due to concerns that the potentiometer output may be of higher current.

Each of the 24 LEDs used were connected to their corresponding GAL shift registers in active high arrangement, in an effort to limit the amount of current they draw.

The LCD received data and instruction inputs from its corresponding GAL shift register and was connected directly to port T (pins PT2, PT3, and PT4) on the microcontroller for purposes such as register selection, selection of read mode or write mode, and LCD clocking. The LCD backlight was unused, due to concerns that it may draw too much current.

A few additional considerations were involved when designing the PCB. Power and ground planes were implemented to ensure that high currents would not burn individual power and ground traces. Trace widths for most of the electrical signals were set at the recommended 12 mil. High frequency traces were optimized to be of minimal length. Right angles were avoided in nearly all traces, especially with high frequency signals. Vias were used sparingly. 1µF decoupling capacitors were implemented at IC power terminals in order to help protect the ICs from changing voltage inputs.

## 3.0 Microcontroller Resource Utilization

The peripheral functions of the microcontroller utilized in the project were PWM, TIM, SPI, and ATD, in addition to other on-chip resources. Each worked in sequence, primarily for the function of music generation and the LED/ LCD display.

The PWM and TIM module were utilized to generate music signals in interrupt-driven mode. By changing the PWM duty cycle through the TIM interrupt routine, the amplitude of a PWM output waveform could be easily modified. Since the timer module was configured to reset when OC7 occurs, changing the TC7 value could precisely modify the PWM output frequency. Since both the amplitude and the frequency could be precisely adjusted, a combination of the PWM and TIM module could be used to construct a music signal. In this project, a simple square wave was used as a music signal to drive a speaker.

However, a signal at a constant frequency could only produce constant, repeated beats, not music. The on-chip RTI function was used to fix this. As it modified the rate of the timer interrupt routine, it altered the music notes. An accumulator was set up in the RTI interrupt. It would reset and send out a shifting flag once it reaches the upper limit indicating that the music note needs to be altered. If the signal is changed quickly, in set periods of time, music can be produced. However, the music produced was merely a long series of connected tones, there were no distinct sounds to it. For this, we utilized a wait time. After every tone, we reduced the frequency to a very low value for a single millisecond. This extremely fast drop produced distinct tones, due to the gap between each sound. And finally, to change the length of each tone, a parallel array was utilized to produce each tone exactly as long as we need it, by repeating certain tones for longer than a single cycle before the wait time was triggered. This finally produced the variable length tones of various distinct notes that we call “music”.

Thus having produced the music, we utilized a multiplexed SPI in program-driven mode for the LCD and LED shift registers. The clock enable for the LED registers was set to active high, with the LCD’s clock enable set to active low, making the two slaves mutually exclusive. This multi-slave configuration separated the registers needed by different peripherals and at the same time minimized the number of microcontroller pins required.

The rate of the game took place utilizing the ATD in program-driven mode. The 8-bit resolution, unsigned configuration was chosen to convert the potentiometer wiper voltage to a digital value. The output of the ATD was then used to adjust the upper limit of the music and the LED shifting rate accumulator. The music and LED shifting rate would change accordingly.

Besides controlling the music and LED shifting rate, the RTI routine was also utilized for pushbutton detection as well as the game timer. The prescaler was selected to induce an interrupt every 0.128 millisecond. It was chosen to ensure successful de-bouncing of the pushbuttons and to provide a more accurate timing flag. Choosing an RTICNT value of 781, the interrupt routine is able to generate a 0.1-second flag with only 0.032 % error.

## 4.0 Software Narrative

The Software written for this project was geared primarily towards peripheral control. The key functions written for the project included the main function, which looped through a variety of conditionals, an RTI interrupt subroutine, which probed pushbuttons, controlled the tempo of the music and updated the musical notes, a TIM interrupt subroutine, which controlled the duty cycle of the PWM output, a pulse_wait function, which provided a short delay after each musical note to help the user audibly differentiate between notes, and a number of SPI shift_out functions for outputting LCD data, LCD instructions, and LED data on the SPI MOSI pin.

The main function utilizes all other functions in the program. It checks the pushbuttons to determine the game state. If the game state is in run mode, the audio signal is outputted on the PWM, the user score is displayed on the LCD, the pushbuttons and LEDs are compared to update the user's score, and the LED shift-out data is updated. If the game state is in off mode, the audio signal is disabled, the LCD displays the difficulty bar, and the ATD is utilized to determine the user's difficulty input from the potentiometer. The main function also checks three flags that update after certain time intervals – one flag updates the LCD every second, another decreases the countdown timer every one millisecond, and the other updates the music notes, checks the tapping logic, and shifts the LED rows every time a shift-out flag is set. The Super Mario Bros theme song was hard-coded into two separate arrays – one of the arrays contained the frequency of the sound signal produced by each note and the other array contained the duration of each note. The music plays in a loop until the game is over. It is also important to note that the LED scrolling sequences were controlled using algorithms that illuminated certain LEDs based on the tone of the musical note being played. Once the countdown timer is set to zero, the main retains in a trap loop until a reset button is hit.

The RTI interrupt subroutine had three separate roles. It probes each of the pushbuttons, it updates the timer counters, and it determines which musical note is being played after a specific amount of accumulated time.

The TIM interrupt subroutine served to modify the PWM duty cycle (alternating it between 0% and 100%) to control the frequency of the audio output. We later realized an improvement could have been made here by using the PWM period and duty registers to control the frequency of the audio output without the timing module.

The pulse_wait function temporarily set the audio output to a non-audible frequency to provide a slight delay after each musical note. This delay allows the user to audibly differentiate between two sequential notes.

A variety of functions were used to control the SPI output and most of them were standard SPI shiftout functions. The key functions to mention here were the diff_display and game_display functions. The diff_display function displayed a bar on the LCD, whose length depends on the ATD input from the potentiometer. This bar's length corresponds to the tempo of the music and thus the difficulty of the game. The game_display function displays the user's score and the countdown timer by converting the decimal score and decimal countdown timer values to ascii.

## 5.0 Packaging Design

The packaging design for our project was a lengthy process that was continuously improved up to the deadline for our project. We took a decent sum of time to select and purchase the correct parts for our project--parts that would serve both the appearance of our project and the mechanical reliability of our project.

The project container itself was a repurposed cardboard box. Cardboard commonly does not have the structural integrity needed for this type of project, but the cardboard box that we selected was very firm and is certainly sturdy enough for our purposes. The cardboard box also was much easier on our budget than other alternatives. The dimensions of the box were based on estimates. We knew only the approximate number of LEDs we would need, and we could only guess how much space the electrical wiring would consume so our dimension estimates ended up being fairly generous. Holes were drilled through the top of the box with a slot for the LCD, a slot for the potentiometer and numerous slots for the pushbuttons and LEDs. Finally, in an effort to increase the amount of sound heard by the speaker, and to allow the power cable to enter the box, we drilled two holes through the side of the box.

Jumper cables were used to attach the many devices mounted on the top of the box to the breadboards within the box. To ensure mechanical strength, we soldered the jumper cables to their corresponding devices and used tape and glue gun to strengthen the LCD connection on the box. Breadboards were also attached inside the box using double sided foam rubber tape. Potentiometer and pushbuttons were also screwed on the box. To prevent any short in the circuit, we also used tape to wrap around the stripped wires near the soldering part.

We made a few cosmetic design choices. For instance, we covered the container with a black poster, as a means of making the LEDs appear brighter. We chose to use LEDs with inbuilt resistors to reduce the amount of clutter within the box and to reduce the possibility of loose connections. To match these red LEDs, we used red pushbuttons. Finally, to help improve the appearance of the inside of the box, we carefully taped jumper cables together in a clean, organized fashion.

## 6.0 Summary and Conclusions

This project brought about a great deal of new experiences for our team members. Prior to beginning this project, each of us had little to no experience with PCB design, audio generation or soldering. A great deal of effort went into researching and brainstorming how to perform these tasks.

The music output was one of our more challenging learning experiences because we had to determine not only how the correct sound frequency could be generated, but also how to make the music sound professional. The wave we used to generate the audio was a square wave, mostly because it produced a high volume and because it was fairly easy to implement; however, if we had more time, we would have preferred to use a sawtooth wave. We later realized that, despite being a little softer in volume, the sawtooth wave contains the potential to produce all of the harmonic sounds, making it the best and richest for generating music (Rise, 2012). We could have produced sounds at the same level as our current square wave given more time, but we settled for a square wave due to constraints.

There were a few ideas and learning experiences that we had involving the audio output for the project. We realized (a bit too late) that we could have improved our software efficiency by directly controlling the frequency of the PWM square wave using the PWM period and duty registers. In addition, we were interested in perhaps finding a way to automate our music generation so that we could avoid having to hardcode songs. We lacked the time to develop and implement that functionality, but it would probably have been the next logical improvement for our project.

We also had a few nice ideas involving the user interface that never quite made it to the final product. We were originally pretty keen about implementing a reaction timer that would display how quickly the user was able to react to an illuminated LED, but we eventually moved away from this idea as it would have meant a significant amount of work for an insignificant feature. For structural integrity and appearance, we originally wanted to 3D-print the container for our project. This idea was dropped due to lack of time and lack of experience with 3D-printing.

## 7.0 References

*LM386 Low Voltage Audio Power Amplifier Data Manual*, Texas Instruments Incorporated, Dallas, TX, 2011.

*Mill-Max 24-Position Dual In-Line Sockets Data Manual*, Mill-Max Mfg. Corp., Oyster Bay, NY, n.d.

*Phicomp Surface-Mount Ceramic Multilayer Capacitor Data Sheet*, YAGEO Corp., New Taipei City, Taiwan, 2016.

Rise, Scott. (2012, August 19) "Harmonics | The Synthesizer Academy." [online] Available: http://synthesizeracademy.com/harmonics/

*Series: ED Stamped DIP Socket*, Onshore Technologies, Torrance, CA, 2006.

# Appendix A:

# Individual Contributions and Activity Logs

**Activity Log for:** Chia-Hua Peng **Role:** Peripheral Leader

| *Activity* | *Date* | *Start Time* | *End Time* | *Time Spent* |
|---|---|---|---|---|
| Drafting Peripheral Layout | 11/1/2016 | 21:30 | 23:30 | 02:00 |
| Set up Google Drive/ Documents | 11/2/2016 | 14:50 | 16:00 | 01:10 |
| Drafting Peripheral Layout | 11/5/2016 | 14:40 | 15:20 | 00:40 |
| Team meeting - peripheral utilization discussion and ordering equipment | 11/6/2016 | 18:40 | 20:00 | 01:20 |
| Sorting Speaker Datasheet | 11/7/2016 | 10:00 | 10:16 | 00:16 |
| Discussing ideas with TA | 11/9/2016 | 19:30 | 19:50 | 00:20 |
| Testing equipment | 11/12/2016 | 02:00 | 02:30 | 00:30 |
| Testing equipment | 11/15/2016 | 15:30 | 17:00 | 01:30 |
| Discussing ideas with TA | 11/15/2016 | 21:40 | 22:00 | 00:20 |
| Drafting Proposal | 11/15/2016 | 23:40 | 00:15 | 00:35 |
| Coding - Use PWM to drive a speaker | 11/16/2016 | 15:00 | 16:30 | 01:30 |
| Drafting peripheral and interrupts logic flows | 11/17/2016 | 00:30 | 01:54 | 01:24 |
| Coding - Music Generation | 11/17/2016 | 17:30 | 00:37 | 07:07 |
| Coding - Rate of Music/ Shift LEDs based on the rhyme | 11/18/2016 | 15:30 | 01:40 | 10:10 |
| Updating Peripheral Layout Map | 11/19/2016 | 02:20 | 03:18 | 00:58 |
| Making images for poster and videos | 11/19/2016 | 16:00 | 18:00 | 02:00 |
| Team meeting - PCB discussion, Coding - ATD, Timer, and LCD display | 11/20/2016 | 14:30 | 01:06 | 10:36 |
| Team meeting - PCB discussion, Coding - debug LED problems | 11/21/2016 | 14:40 | 01:00 | 10:20 |
| Team meeting - Finalizing coding part and breadboard prototype | 11/22/2016 | 14:10 | 23:00 | 08:50 |
| Video Editing | 11/23/2016 | 17:00 | 19:00 | 02:00 |
| Packaging the final product - Designing Box Layout | 11/29/2016 | 12:00 | 18:30 | 06:30 |
| Packaging the final product - Interface and Debugging Circuit | 11/30/2016 | 19:00 | 02:00 | 07:00 |

| | | | | |
|---|---|---|---|---|
| Packaging the final product - Debug, Soldering circuits | 12/1/2016 | 18:30 | 04:30 | 10:00 |
| Packaging the final product - Debug, Finalize the Non-PCB part for demo | 12/2/2016 | 15:00 | 21:00 | 06:00 |
| Videotaping and connecting PCB | 12/6/2016 | 18:00 | 01:10 | 07:10 |
| Video Editing and re-wiring the final product | 12/7/2016 | 12:00 | 18:30 | 07:10 |
| Strengthen the mechanical robustness | 12/8/2016 | 16:30 | 20:00 | 03:30 |
| Editing Report – Introduction and Peripherals | 12/9/2016 | 08:00 | 09:00 | 01:00 |
| Editing Report – Peripherals and Software | 12/9/2016 | 13:00 | 13:50 | 00:50 |
| Finalizing the Report | 12/10/2016 | 02:00 | 03:00 | 01:00 |

## Written Summary of Technical Contributions: Chia-Hua Peng

I performed a variety of tasks that included both the software and hardware aspects of the project, and also provided guidance as a team leader and brought the teamwork to successful achievements.

One of my tasks is to coordinate the peripheral layout and the microcontroller resource utilizations. I drafted a rough circuit diagram indicating how the microcontroller and the peripherals are connected and how the multiplexed shift registers are functioning. I also determined the proper operating modes for the microcontroller features and decided how to utilize those resources to implement our design. To ensure the hardware components were suitable for our project, I also wrote a simple demo program to test the peripherals and informed the team whether or not an equipment need to be re-ordered.

Another of my tasks is to design and implement algorithms to make the music game using the on-chip resources. I designed the algorithms to generate a distinct sound and to alter a music note using the PWM and TIM module, to adjust the music rate using the ATD feature, and to shift out the LED display and the LCD message using a multiplexed SPI. With our Software Leader's help in translating music notes to digital representations, the algorithm was able to generate music with correct tones and tempos, and the LEDs were able to display a pattern that is based on the music rhyme.

Besides designing peripheral layout and coding the program, I also interfaced a breadboard prototype for bug tracking the program as well as hardware issues. The problems found included no distinct sounds in a series of connected tones, extensive current drain by the GAL devices, bad connections between the wires and hardware components, and many more. I troubleshoot the problems by modifying the programs and changing the hardware configurations with the Software Leader, and informed the Interface Leader the changes needed to be made in the PCB and the final product.

In addition, I also have a significant contribution to building the final product. I designed its exterior layout, drilled the holes on the box, mounted the hardware components, assembled the female to male jumpers neatly, assisted teammates soldering the circuits, and used a glue gun to strengthen the loose connections. I am also the producer/ editor for the promoting video.

As a team leader, I also need to ensure that the day-to-day operation of the team is maintained and that all the team members understand the ongoing plan of the project. To make sure the motivation and the performance level are maintained, I set up an activity log sheet on the Google Drive, requested team members to fill out their work activities day by day, and reminded

them to complete certain tasks on time. I also need to ensure all the team members receive enough help on their distributed work. I sent a rough game logic flow chart to the Team Documentation Leader to help him document the technical report. I drafted and updated a microcontroller-to-peripheral pin connection sheet to assist the Interface Leader designing the PCB. I even made a power point slide just to demonstrate how the peripherals were connected, how microcontroller resources were utilized, and how the software algorithms functioned, to my teammates and made sure they understand the ongoing plan of the project the while working on their own distributed work individually.

**Activity Log for:** Hengyi Lin **Role:** Software Leader

| *Activity* | *Date* | *Start Time* | *End Time* | *Time Spent* |
|---|---|---|---|---|
| Team meeting - peripheral utilization discussion and ordering equipment | 11/6/2016 | 18:40 | 20:00 | 01:20 |
| Discussing ideas with TA | 11/9/2016 | 19:30 | 19:50 | 00:20 |
| Testing equipment | 11/15/2016 | 16:30 | 17:00 | 00:30 |
| Discussing ideas with TA | 11/15/2016 | 21:40 | 22:00 | 00:20 |
| Ordering Additional equipment | 11/17/2016 | 02:30 | 02:45 | 00:15 |
| Coding - Music Generation | 11/17/2016 | 18:40 | 00:37 | 05:57 |
| Coding - Rate of Music/ Shift LEDs based on the rhyme | 11/18/2016 | 19:00 | 01:40 | 06:40 |
| Team meeting - PCB discussion, Coding - ATD, Timer, and LCD display | 11/20/2016 | 14:30 | 01:06 | 10:36 |
| Team meeting - PCB discussion, Coding - debug LED problems | 11/21/2016 | 16:30 | 01:00 | 08:30 |
| Team meeting - Finalizing coding part and breadboard prototype | 11/22/2016 | 14:10 | 23:00 | 08:50 |
| Packaging the final product - Designing Box Layout | 11/29/2016 | 18:30 | 01:00 | 06:30 |
| Packaging the final product - Interface and Debugging Circuit | 11/30/2016 | 19:00 | 02:00 | 07:00 |
| Packaging the final product - Debug, Soldering circuits | 12/1/2016 | 18:00 | 04:30 | 10:30 |
| Packaging the final product - Debug, Finalize the Non-PCB part for demo | 12/2/2016 | 15:00 | 18:50 | 03:50 |
| Videotaping and connecting PCB | 12/6/2016 | 18:30 | 03:10 | 08:40 |
| re-wiring the final product | 12/7/2016 | 17:00 | 18:30 | 01:30 |
| Strengthen the mechanical robustness | 12/8/2016 | 17:00 | 20:00 | 03:00 |

## Written Summary of Technical Contributions: Hengyi Lin

Although I was assigned as a Software Leader, I performed various tasks in this project. Besides coding and testing the overall game logic flow of the program, I also interfaced a breadboard prototype, helped the Peripheral Leader to test and debug the software/ hardware problems, soldered the loose connections to improve the strength of the final product, and ordered necessary equipment for the design.

Implementing the overall software logic flow was a significant task. To assure that the algorithms successfully worked on the hardware, I collaborated with the Peripheral Leader extensively. I translated the music notes to digital representations, encoded the music note frequency and durations in a parallel array, and utilized the PWM and TIM features set up by the Peripheral Leader to have the speaker play a perfect song. And then I designed the LED note representation for each music note, played as a player to test the reasonability and revised the combination until a balanced and reasonable LED note representation was generated. Last but not least, I also determined the most crucial part of the game, scoring and losing points. I decided that when the last row of LEDs lit up, the player had to press the pushbutton in order to score, or the player lost point. I worked with the Peripheral Leader to make sure that the peripherals of the microcontroller could perform these actions.

Besides implementing the software logic, I also helped building up the breadboard prototype of the hardware before packaging with the Peripheral Leader. I carefully selected the places to put GALs and microcontroller so that the wiring distance was shorter. I also kept track of each trace of each wire, especially some important traces such as LED clock, serial-in pin, and clock enable. According the coded peripheral settings, I placed the breadboard with the GALs shifting LEDs to the corresponding place so that the LEDs shifted from the top to the bottom. I also checked the display of the LCD to ensure that the speed select and in-game select were correctly displayed and the score was able to increase or decrease depending on the match/not match condition.

I also strengthened the connection of the loose wires using soldering. Soldering was a really new experience to me. With the help of the TAs, I didn’t solder very well at first, especially for the easily broken wires of the pushbutton and it became so difficult to solder to the same place again since I didn’t know how to remove the solder on that place. After working for several hours and being helped by the TAs, I soldered faster and faster and I learnt how to remove the solder using the desoldering gun.

In addition to programming and prototype interfacing tasks, I am also the one to order and manage the equipment for the design. I purchased the necessary components for the design, created an excel spreadsheet showing all the ordering information, and took over the ordering job whenever I was informed that an additional equipment was needed.

**Activity Log for:** Paul Swartz **Role:** Interface Leader

| *Activity* | *Date* | *Start Time* | *End Time* | *Time Spent* |
|---|---|---|---|---|
| Team meeting - peripheral utilization discussion and ordering equipment | 11/6/2016 | 18:40 | 20:00 | 01:20 |
| Finding appropriate PCB parts | 11/18/2016 | 19:30 | 23:30 | 04:00 |
| Developing EAGLE part libraries | 11/19/2016 | 23:30 | 03:30 | 04:00 |
| Developing EAGLE PCB schematic | 11/19/2016 | 12:30 | 17:30 | 05:00 |
| Developing EAGLE PCB layout | 11/19/2016 | 23:30 | 06:30 | 07:00 |
| Team meeting - PCB discussion, Developing PCB layout | 11/20/2016 | 13:00 | 23:00 | 10:00 |
| Team meeting - PCB finalization / Submission | 11/21/2016 | 06:15 | 12:45 | 06:30 |
| Team meeting - Finalizing coding part and breadboard prototype | 11/22/2016 | 14:10 | 17:30 | 03:20 |
| Packaging the final product - Interface and Debugging Circuit | 11/30/2016 | 19:00 | 23:00 | 04:00 |
| Packaging the final product - Debug, Soldering circuits | 12/1/2016 | 18:00 | 21:00 | 03:00 |
| Packaging the final product - Debug, Finalize the Non-PCB part for demo | 12/2/2016 | 17:30 | 21:00 | 03:30 |
| PCB soldering | 12/3/2016 | 20:30 | 21:30 | 01:00 |
| PCB soldering | 12/4/2016 | 10:30 | 12:30 | 02:00 |
| Designing Poster | 12/4/2016 | 20:00 | 21:00 | 01:00 |
| Creating Poster | 12/4/2016 | 21:30 | 23:10 | 01:40 |
| Videotaping, connecting PCB, debugging PCB | 12/6/2016 | 18:00 | 06:00 | 12:00 |
| re-wiring the final product | 12/7/2016 | 12:00 | 18:30 | 06:30 |
| Commenting code, writing interface description in report, writing individual contributions | 12/6/2016 | 23:00 | 03:00 | 04:00 |
| Finalizing Documentation | 12/18/2016 | 18:00 | 21:30 | 03:30 |

## Written Summary of Technical Contributions: Paul Swartz

I performed a variety of different tasks for this project, but the most significant tasks were designing, soldering, and implementing/testing/debugging the PCB.

I selected the necessary parts for attaching the GAL devices and the microcontroller to the PCB, as well as the surface-mount decoupling capacitors and the parts needed for connecting the LEDs and the LCD to the PCB. This mostly consisted of looking through datasheets to verify appropriate measurements, finding the least expensive parts, and finding parts that I could be certain would work. During this process, I considered what would be the most efficient and practical way of laying out the PCB and attaching parts to the surface of the PCB.

I created the EAGLE libraries for most of the parts I used on the PCB, as most of the parts that I selected did not have easy-to-find preexisting libraries. The one exception was that I used the 0603 Sparkfun capacitor library for the decoupling capacitors I selected. Creating the libraries was fairly straightforward and mostly just involved making careful measurements to ensure the parts would fit into the PCB.

I developed the EAGLE schematic and EAGLE board layout for the PCB. As I had very limited experience with EAGLE and PCB design prior to this project, much of my time spent on the board layout involved watching videos, reading tutorials, and looking through common tips and recommendations. While creating the board layout, I had to make a number of design considerations. The maximum PCB size allowed in the free version of EAGLE is 80mm x 100mm so the layout had to be fairly compact, given the large number of LEDs and other devices that had to be attached. I optimized the traces to be of minimal length—particularly higher frequency signal traces. I used the recommended trace width for most signals, but used significantly larger trace widths for signals likely to draw a lot of current. I used ground and voltage planes to maximize the size of the power traces. I minimized my use of right angles in trace paths, particularly with higher frequency signals. I spent time minimizing my use of vias. Finally, to ensure a consistent voltage at IC power terminals, I added decoupling capacitors to the board.

Soldering the PCB was another new experience for me. Though it did not require any significant design choices, I got to become pretty familiar with a number of soldering tools and soldering practices.

Implementing the PCB was a far more significant challenge. I disassembled and reassembled the wiring of the project several times in an effort to debug. The audio, LEDs, potentiometer, and pushbuttons all worked perfectly with the PCB implementation; however, the

LCD would work perfectly some of the time and would be blank the rest of the time. The LCD behavior seems unpredictable with the PCB implementation. After a great deal of troubleshooting, I was able to narrow down the cause of the LCD problem to likely be an issue with the power and ground terminals on the PCB. Given a lack of progress after a ridiculous amount of time spent debugging the PCB, and given the approaching deadline, I have decided to revert the project back to the breadboard.

My contributions to documenting the project – commenting the code, creating the poster, and writing the interface description – were all fairly straightforward.

**Activity Log for:** Suprith Ramanan **Role:** Team Documentation Leader

| *Activity* | *Date* | *Start Time* | *End Time* | *Time Spent* |
|---|---|---|---|---|
| Team meeting - peripheral utilization discussion and ordering equipment | 11/6/2016 | 18:40 | 20:00 | 01:20 |
| Editing Proposal | 11/16/2016 | 14:30 | 15:30 | 01:00 |
| Budgeting and Planning Requisite Materials | 11/18/2016 | 14:30 | 15:30 | 01:00 |
| Team meeting - PCB discussion, Planning, Analysis and Recording | 11/20/2015 | 13:00 | 19:00 | 06:00 |
| Team meeting - PCB finalization / Submission | 11/21/2016 | 06:15 | 10:30 | 04:15 |
| Code Analysis and Flowcharting + Report Part 1 | 11/23/16 | 13:00 | 18:00 | 05:00 |
| Code Flowcharting | 11/24/16 | 14:00 | 17:30 | 03:30 |
| Code Analysis + Report: Peripheral Layout | 11/25/16 | 13:00 | 16:00 | 03:00 |
| Code Analysis + Report: Software Narrative | 11/26/16 | 14:30 | 17:00 | 02:30 |
| Packaging the final product - Designing Box Layout | 11/29/2016 | 12:00 | 18:30 | 06:30 |
| Packaging the final product - Interface and Debugging Circuit | 11/30/2016 | 19:00 | 21:00 | 02:00 |
| Packaging the final product - Debug, Finalize the Non-PCB part for demo | 12/2/2016 | 14:00 | 19:30 | 05:30 |
| Video Scripting | 12/4/2016 | 23:00 | 02:00 | 03:00 |
| Videotaping and connecting PCB | 12/6/2016 | 18:00 | 23:00 | 05:00 |
| re-wiring the final product | 12/7/2016 | 18:00 | 19:00 | 01:00 |
| Report: Packaging Design | 12/7/2016 | 21:00 | 23:00 | 02:00 |
| Report: Summary, References + Finalizing Documentation | 12/8 | 18:00 | 21:30 | 03:30 |

## **Written Summary of Technical Contributions:** Suprith Ramanan

The functions I performed for the team could be neatly summarized as “whatever was needed at the time”. I took on writing the report as my primary, and served to assist everyone else with what they were trying to do. I helped Paul bug check and edit his PCB and helped Chia-Hua and Hengyi put the project together. I aided them in putting things together, taking them apart, and fixing them as needed. But my primary purposes were in the writing, analysis, and ordering of what we did.

I kept notes on the flow of the code, and analyzed what I was given to work the rest out, as this was before the code had been commented. This came in handy for the flowchart, as it was far easier to map out the code with Chia-Hua’s help. This also aided in writing the report, my primary job. The report demanded exacting detail and logic, something that was exceedingly difficult to accomplish, given that I needed to do it essentially on my own over Thanksgiving break. I reviewed the code multiple times in order to work out the flow, something that wasn’t easy as the code lacked comments and guiding points. The flowchart aided me in this way, replacing the guiding comments, as I could work out the general flow of the code’s logic and work from there. In this way, I mapped out the flow of the code. The Packaging Design was far easier to do, as I was also the one who found the necessary parts we needed for our project. I also worked with Paul to do our project’s budget sheet. Thus, as I was the one to figure out the logic for much of the parts we used, it was very simple section to do. In this way, I wrote the report with my group’s help. The Interface Design was done with heavy help from the group, but the summary was very simple to do, as I’d been taking notes from the beginning, and already had a list of our planned and implemented ideas. I also had aided with some of the research, so it wasn’t difficult to do.

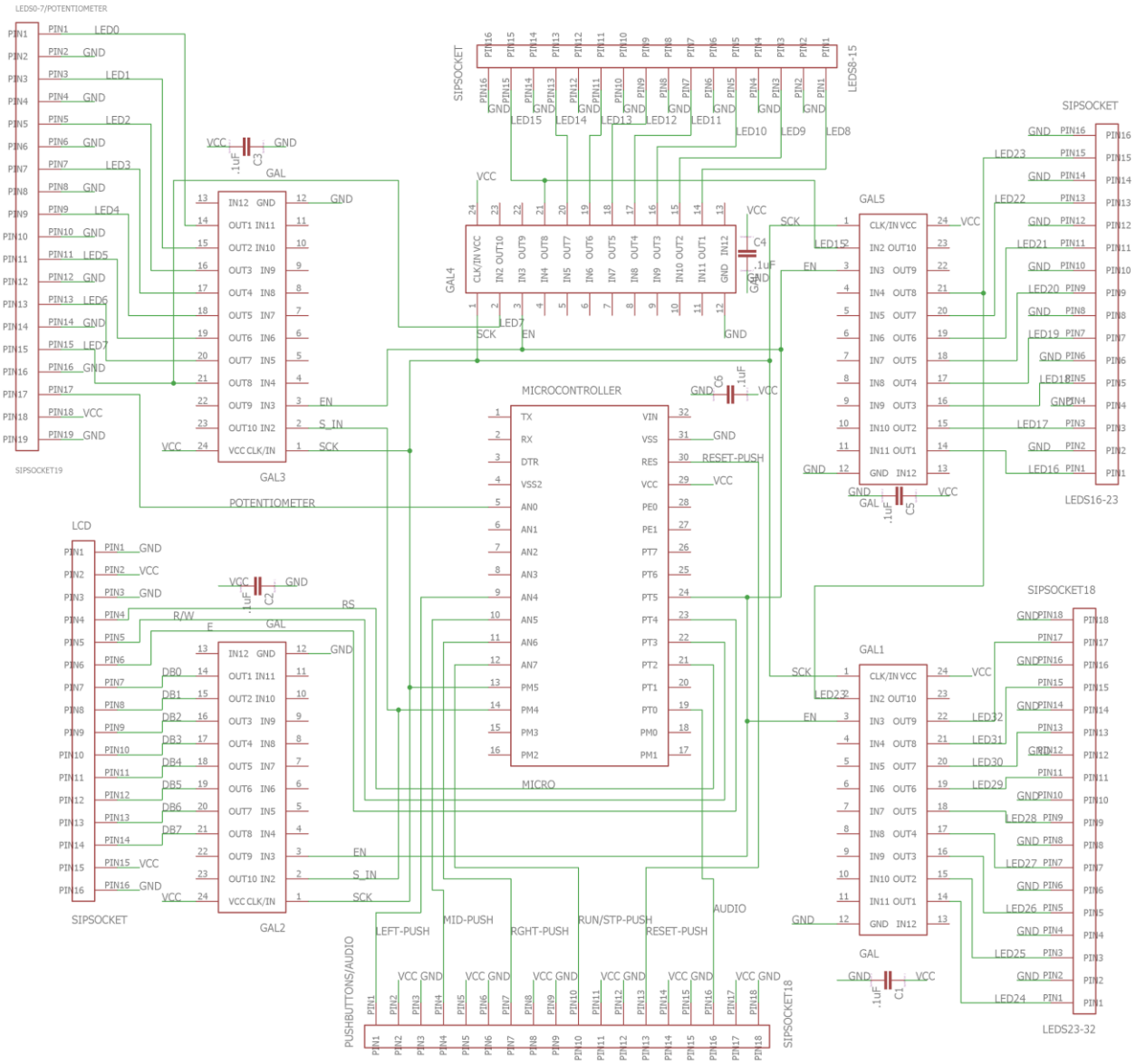
Following this, the next task I took on was planning, scripting, and directing the video. My group requested that I be the one to narrate the video, which made writing lines far easier. The project, of course, needed to be the focus, so I presented the video as an almost tutorial format. It took a while, but the project lent itself naturally to being displayed due to the lights and music, so the script focused on showing off how it sounded and played.

The other tasks I performed were fairly straightforward - I aided my teammates where I could, worked on what was necessary for the project, or the circuit, or the design, or the parts, and stayed as long as I could to give them support.

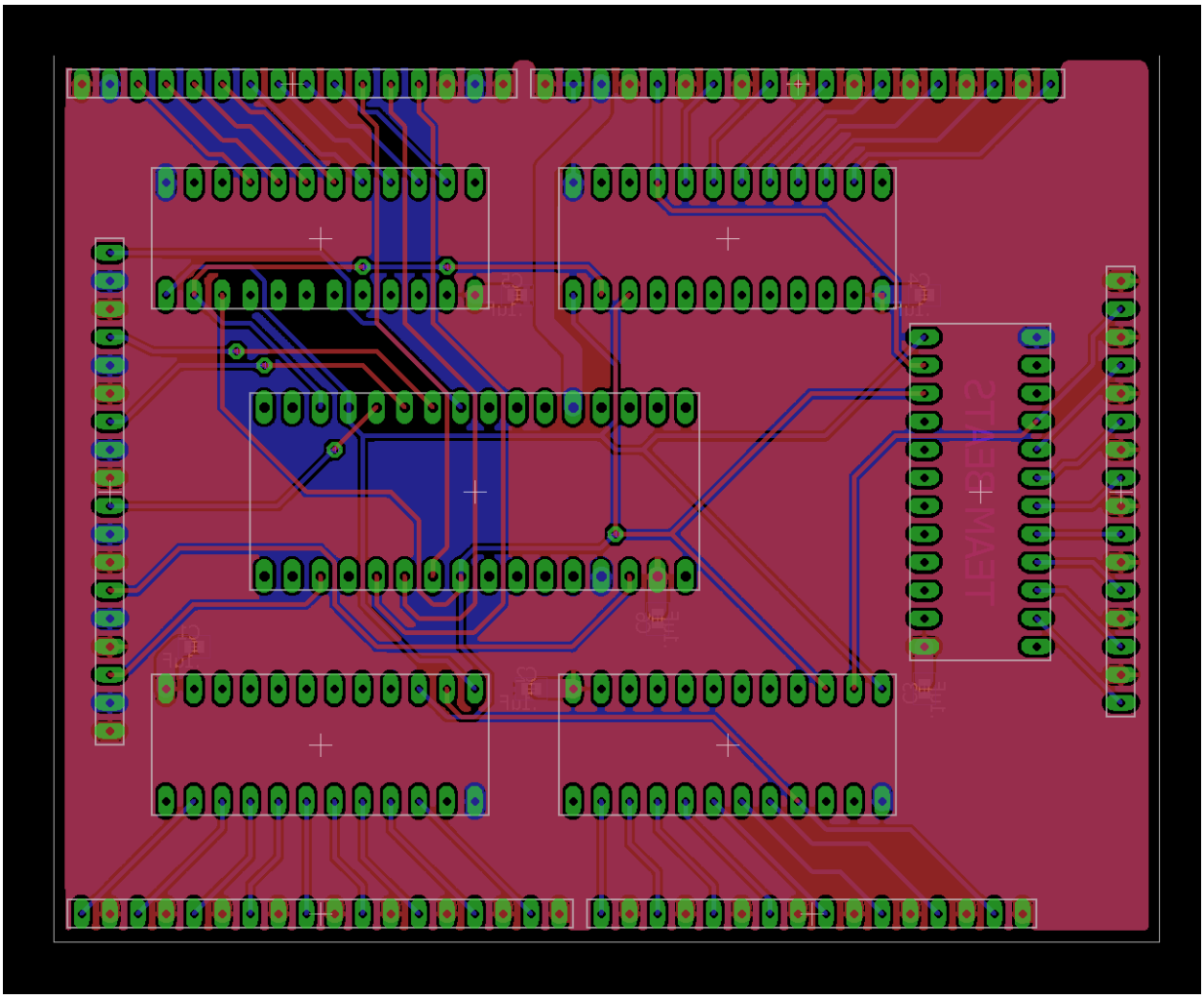
# Appendix B:

# Interface Schematic and PCB Layout Design

EAGLE Interface Schematic - Paul Swartz

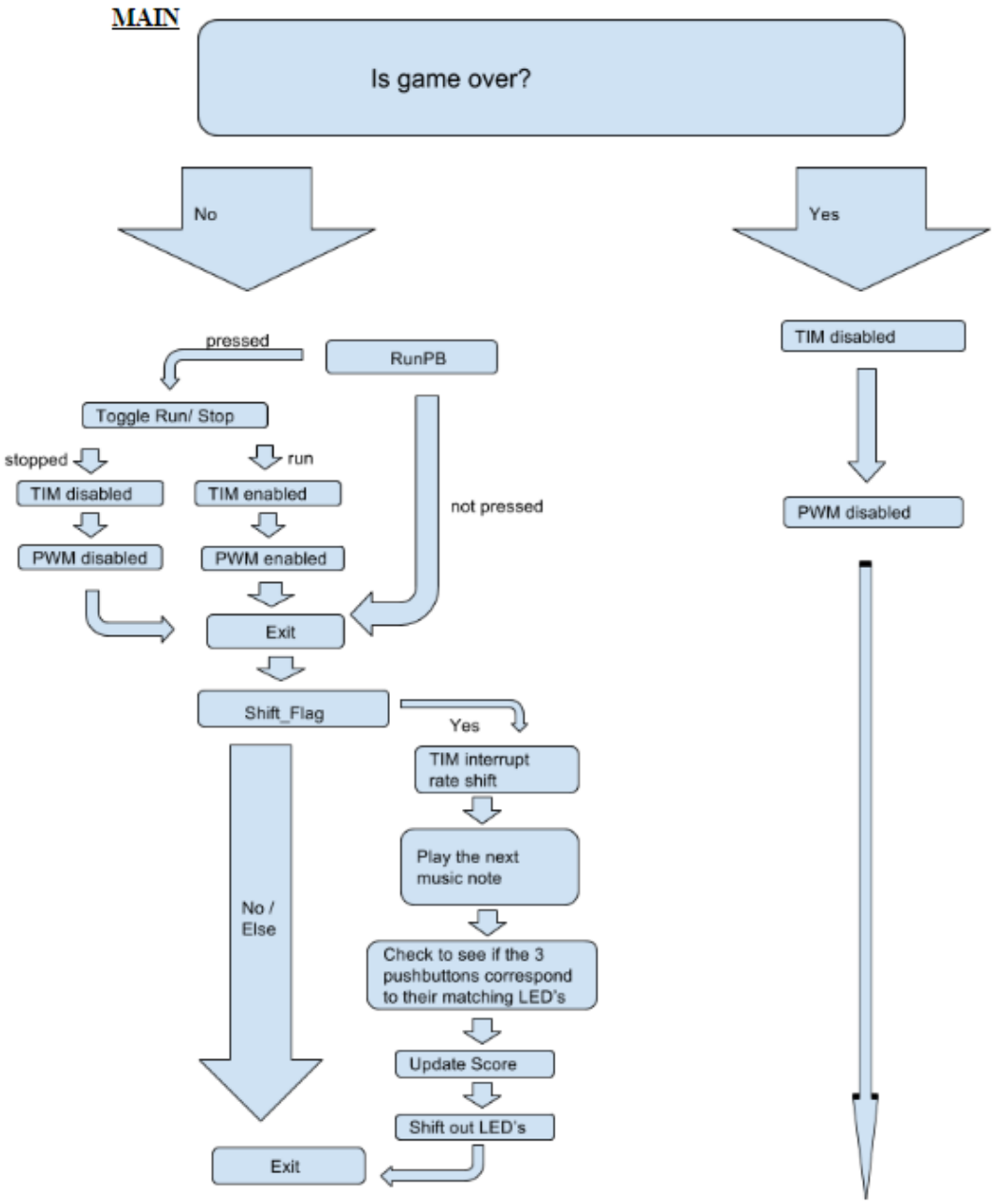


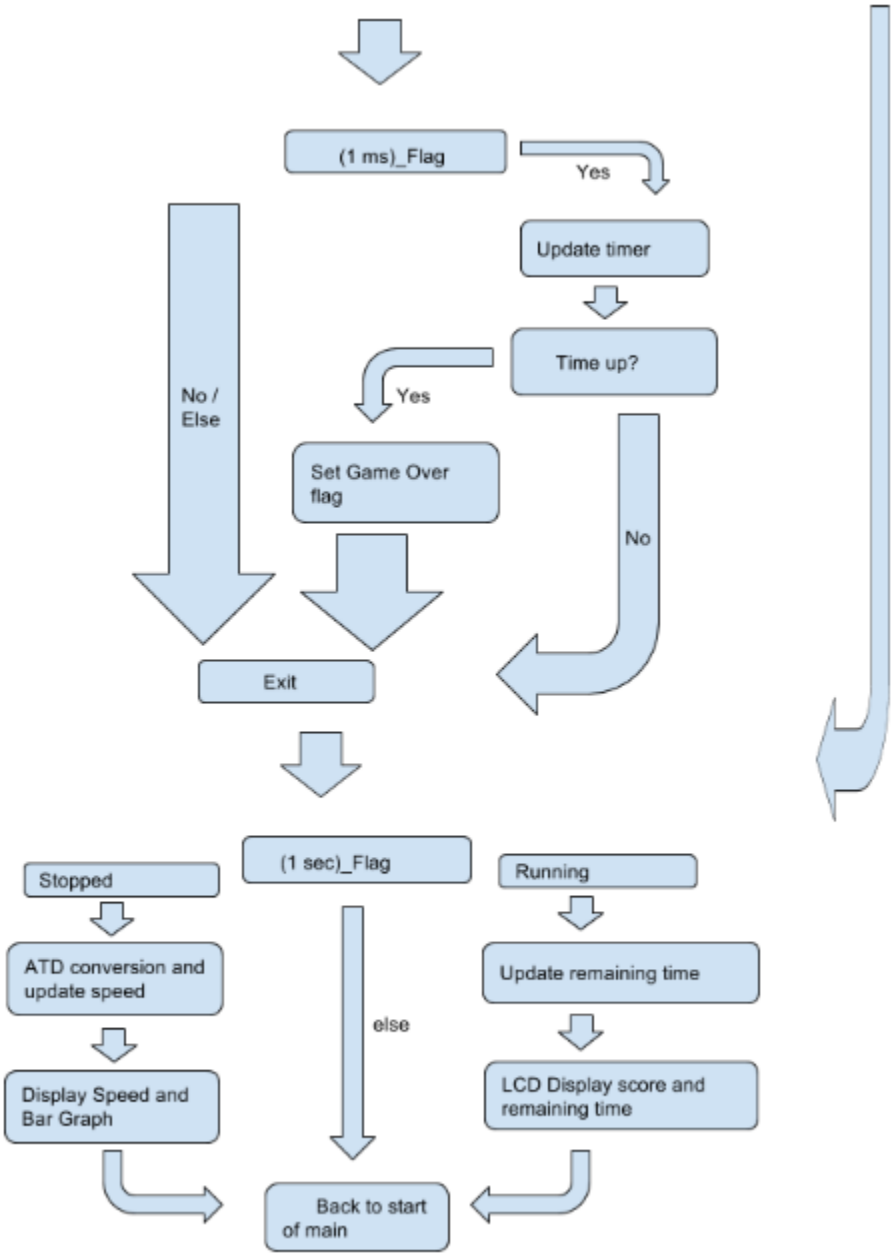
PCB Layout - Paul Swartz



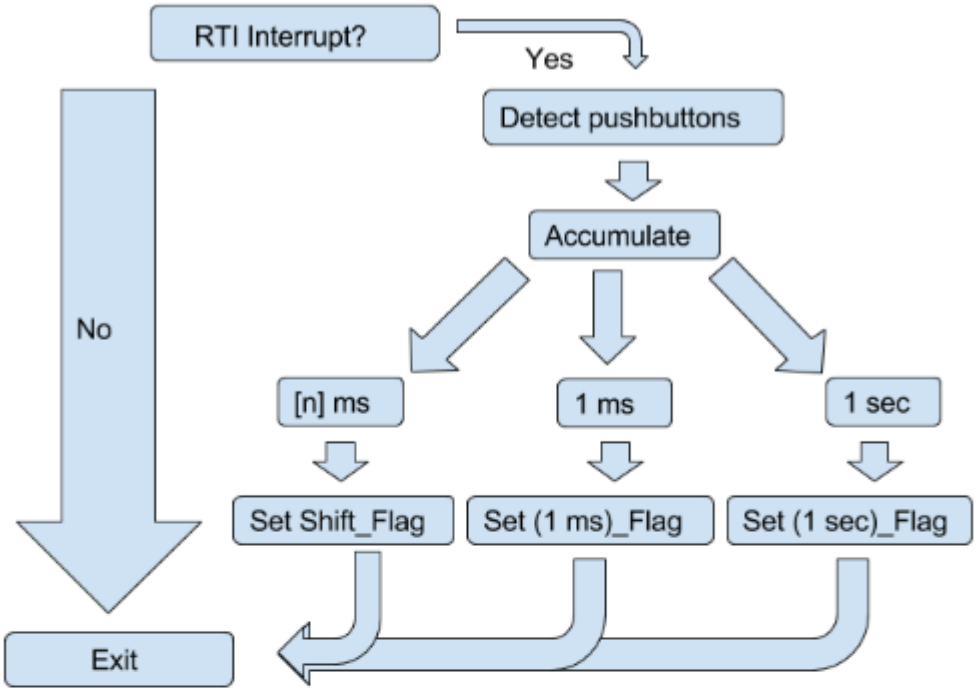
# Appendix C:
# Software Flowcharts

Software Flow Diagram - Suprith Ramanan

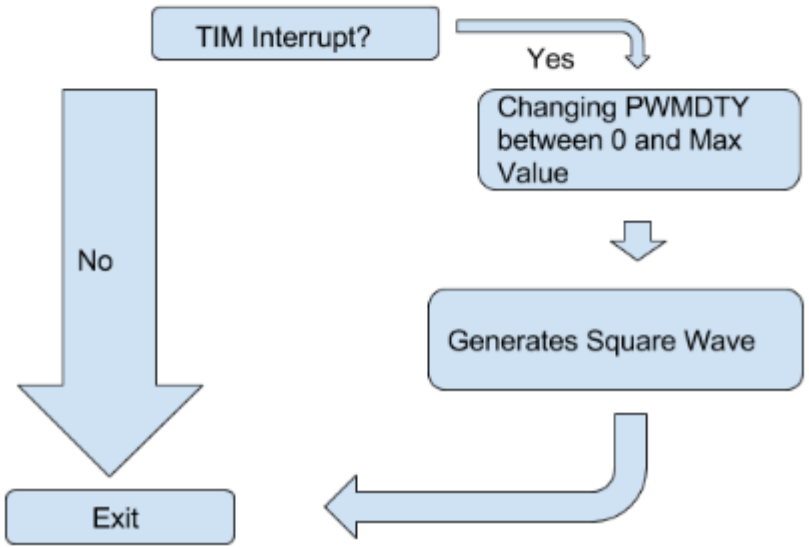



(1 ms)_Flag
Yes
Update timer
Time up?
Yes
No / Else
Set Game Over flag
No
Exit
(1 sec)_Flag
Stopped
ATD conversion and update speed
Display Speed and Bar Graph
else
Running
Update remaining time
LCD Display score and remaining time
Back to start of main

**RTI INTERRUPT**

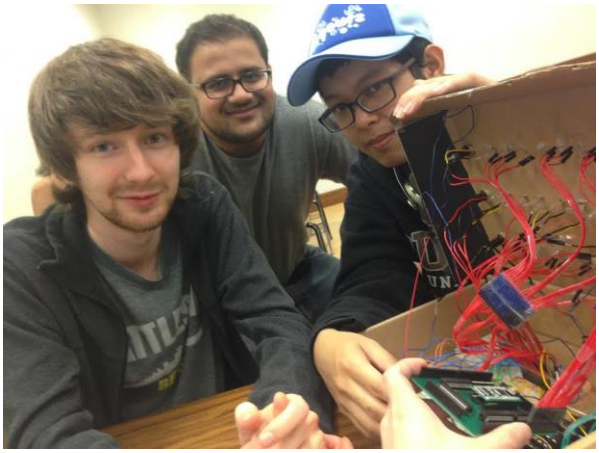


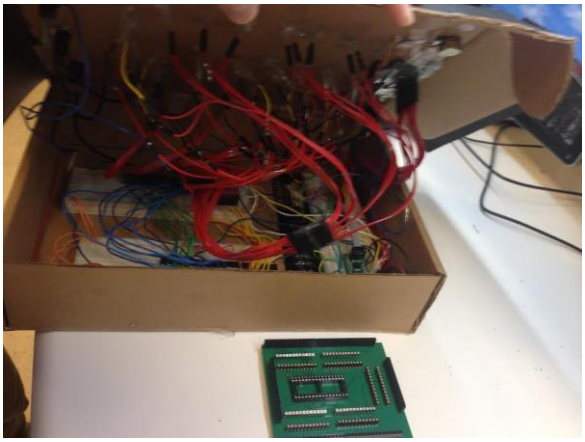

**TIM INTERRUPT**



# Appendix D:
# Packaging Design

Packaging Design Photos - Suprith Ramanan, Chia-Hua Peng





Video Link: https://youtu.be/reYrb8jZK_c