Mini-Project Final Technical Memo

Morse code keyer Design

Team member 1: Saaif Ahmed

Team member 2: Leon Li

IED Section: 6

Date: 6/18/2020

Introduction

The mini project we chose for our group is the Morse Code Keyer. This project in particular is abundant in electrical, computer, and systems engineering (ECSE) principles. Since both group members were ECSE students, and were familiar with the concepts needed for this project, the Morse Code Keyer was chosen so that the members were able to develop a refined and robust final product.

The goal of the project was to build a device that can output sound based Morse Code. Morse Code, the language of "shorts" and "longs", or "dots" and "dashes", is able to represent all letters of the English alphabet and the digits 1 through 9. According to the international standard for Morse Code, the length of a "dot" signal is equal to 1 time unit, and the "dash" signal is 3 times that at 3 time units. In addition the time between each Morse Code signal is defined as 1 time unit, and the time between each word is 7 time units.

This project had the added requirement of building a device that was able to output 5 to 40 words per minute. Since sound based Morse Code and the added requirement were all time based, we chose to design a circuit whose primary function was timing output signals.

Furthermore, the complete Morse code keyer should also include the keyer case model design. The case should include two keys enabling users to input the two different signals of Morse Code. The ideal size of such a keyer case would be roughly the size of a computer mouse and should have some friction pads so that it does not slide around when inputting the Morse Code. This device itself should also remain portable and retain all of its functionality.

Morse Code is one of the recognized international languages held as a baseline for any form of long distance communication. If a system cannot operate with Morse Code it does not pass the most fundamental standard of communication. With the Morse Code Keyer, we are attempting to develop a robust, lightweight, and reliable device that can produce sound based morse code. Given our approach of combining standard circuit concepts with materials that are used for class, we believe that the final product will meet all of these goals.

Concepts and Benchmarking

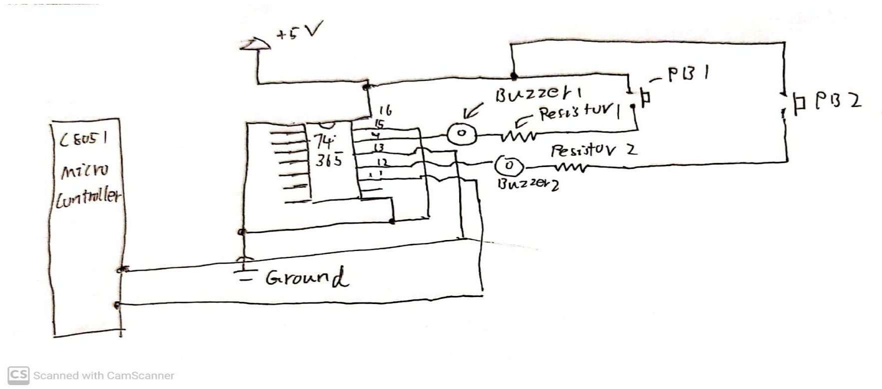


Figure 1: Solution 1, Use of a microcontroller.
PB = Push Button

The C8051 microcontroller has a timer built into the chip, this feature can be utilized to provide accurate time signals for the buzzer to turn on and off. When accounting for an accurately debounced Push Button, the expected error of the mircocontroller's timing would be on the scale of micro seconds. The downside to this approach is that microcontrollers in general heavily rely on a computer program. If the microcontroller was separated from the computer loading the program onto it, users would have an incredibly difficult time modifying the timing of the Morse Code Keyer. Due to this reliance on a computer, should any problems arise with this device, both the microcontroller and the computer would need to interact in order to resolve software problems. However, because the product will be shipped with the microcontroller separated from the computer it is impossible to resolve issues that arise with the device. Thus this approach is not practical.

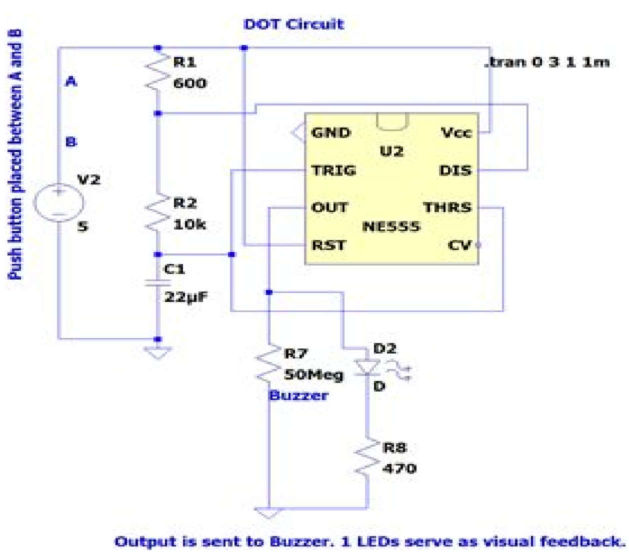


Figure 2: Solution 2, Use of a 555 timer

555 Timers are a commonly used timing circuit. They provide relatively accurate timing, though a few factors can affect the output. Namely internal resistance of the circuit board, the tolerance of other components, and the quality of the push button. With all these factors in mind the error range of the 555 timer is roughly between 0.001s and 0.01s. When compared to manually timed inputs this error range is effectively insignificant, but, when compared to the microcontroller of the previous design this error is quite large almost to the point of inaccurate.. However, the error mentioned prior is actually correctable by changing resistor and capacitor values. This is very easy to do unlike with the microcontroller. Furthermore, the 555 timer is completely independent of any other system meaning that it is easily adjustable and can allow for the development of open-source features of the Morse Code Keyer. In addition, when considering manufacturing costs, the 555 timer circuit design has a huge advantage over the microcontroller circuit design.

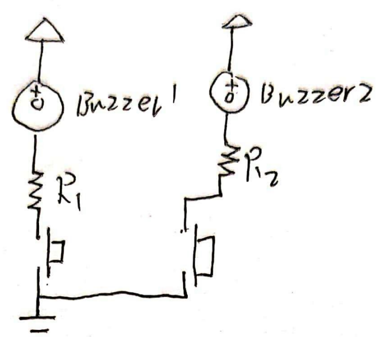


Figure 3: Solution 3, Basic circuit design

The design shown in Figure 3 above is the lowest price and simplest circuit design. This design was the very first idea we came up with. The main idea of the circuit is to use two different voltage dividers to separate between a DOT signal and a DASH signal. Because the buzzers would receive a different signal from the resistors they would output 2 different sounds, one for a DOT and one for a DASH. There is one critical flaw with this design. To the Keyer themselves, they are able to easily identify which signal represents a DOT and which signal represents a DASH. However, to the receiver of the signal the correspondence of each sound is undefined. While the requirement document for the mini project does not mention anything about timing, the standard for sound based morse code is based on duration of sound signal rather than the type of sound signal. This design, while it is the cheapest and easiest to manufacture, fails in complying with the international standard of sound based Morse Code.

Final Design Decision:

From observing our Concept Selection Table below (Table 1) we can see that based off of our criteria, the 555 timer circuit is clearly the best choice. It combines the perfect balance of modularity, cost efficiency, and accuracy that we are looking for in a final project. In terms of how the 555 timer circuit will come together we look to the Concept Combination Table (Table 2). We can see that with the elements highlighter in that table we have all the fundamentals we need to design our final product solution.

Selection criteria	Micro controller	555 timer	Basic circuit
Cost	-1	1	1
Energy efficient	1	1	-1
Timing accurate	1	1	-1
Able to provide time difference	1	1	-1
Build non-virtually	0	1	1
Send clear signal	1	1	-1
Easy to use	1	1	-1
Customizable	-1	1	-1
Circuit complexity	0	-1	1
Sum of 1	**5**	**8**	**3**
Sum of 0	**2**	**0**	**0**
Sum of -1	**2**	**1**	**6**
Total	**3**	**7**	**-4**

Table 1: Concept Selection Table

Power	Timing	Input	Sound Feedback
Microcontroller	Microcontroller	Paddle Toggle	Computer Audio
5V supply	***555 Timer***	1 Button	***Buzzer***
AA Batteries	Human Timing	***2 Buttons***	Large Speaker

Table 2: Concept Combination Table

Solution

Circuit Design:

The final solution for the Morse code keyer design project is using 2 555 timers to provide the adequate timing for both DOT and DASH signals by connecting different resistors and capacitors.

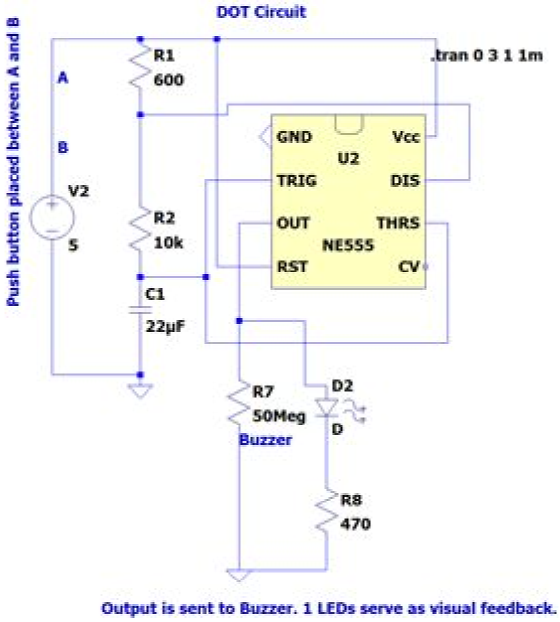


Figure 4: "Dot" signal circuit

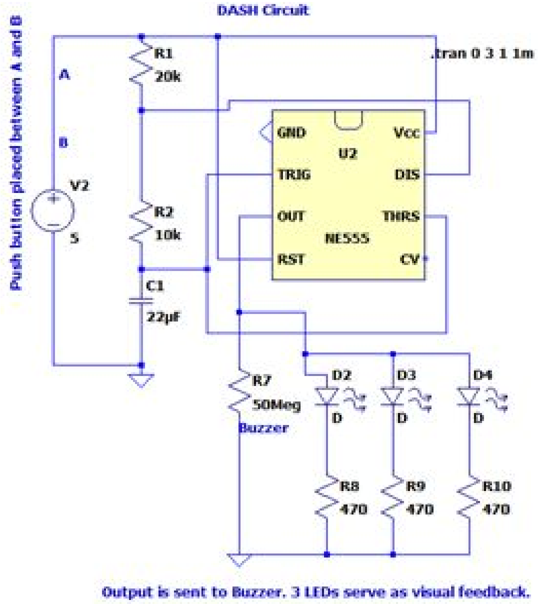


Figure 5: "Dash" signal circuit

The equation for the 555 timer timing circuit is:
High time (Output circuit turns on): T(s) = 0.69*(R1[Ohm] +R2[Ohm])*C1[Farad]
Low time (Output circuit turns off): T(s) = 0.69*R2[Ohm]*C[Farad]

According to the project requirement, the design of the Morse code keyer must output 5-40 words per minute. Our design for one time unit is 0.17 seconds. The circuit component for the DOT circuit is R1 = 600 Ohm, R2 = 10k Ohm, C1 = 22 uF, for the DASH circuit is R1 = 600 Ohm, R2 = 10k Ohm, C1 = 22 uF. The resistors and capacitors we chose are all included in the Analog Devices parts kit, which will allow us to build and test the physical circuit.

The additional features added during the circuit design is 1 LED connected in parallel with the buzzer at the DOT signal circuit and 3 LEDs are connected in parallel with the buzzer in the DASH signal circuit. These LEDs serve as a visual feedback to the user, in addition to the audio feedback from the buzzer. This creates an opportunity for those with hearing challenges to effectively utilize our circuit. In addition, these LEDs have the added benefit of serving as a timer for the keyer (when LED turns off, release the push button).

Keyer Case Model Design:

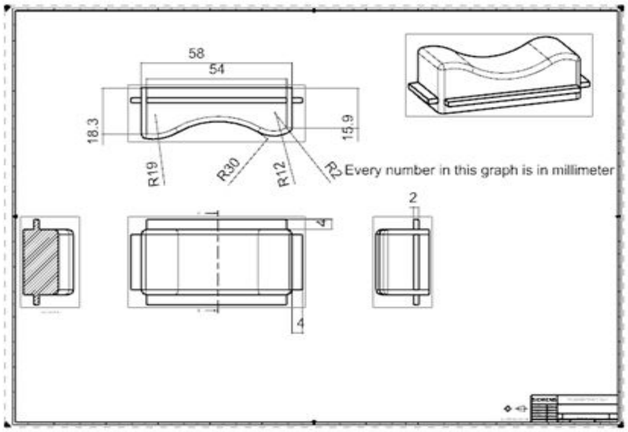


Figure 6: Keyer key

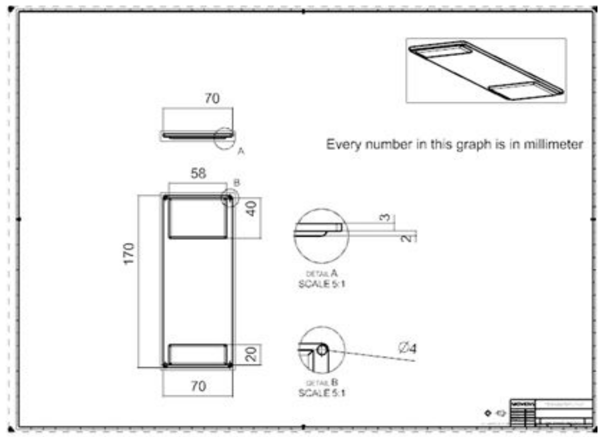


Figure 7: Keyer bottom

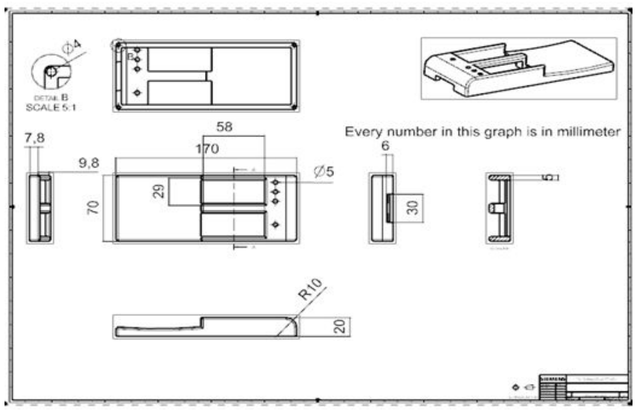


Figure 8: Keyer main part

The design of the Keyer case includes 4 different parts: 2 x Keyer keys, 1 x Keyer bottom, 1 x keyer main part. Overall size is 170 x 70 x 20 mm (length x width x height), or about the size of a computer mouse. This lets an adult user operate our Morse Code Keyer with one hand.

The keyer keys will be directly assembled at the keyer main part, above the circuit board and each key connected with each push button. Because the keyer key is connected to the push button there is a possibility to make contact with the circuit. Therefore the material for the keyer key should be non-conductive carbon fiber. This lightweight material has the added benefit of preventing mistyping due to its low weight.

The keyer bottom and keyer main part are connected together with 4 x 4mm screws, for easy assembly and disassembly. The material for both parts are carbon steel, which provides the sturdy structure and right amount of weight (3 kg). With the rubber pads attached to the bottom part creating friction, combined with the weight of the device, a relatively stable foundation is made for user to input Morse Code

Test Plan and Test Results

Test 1:

To test our device we attempted to output certain basic Morse Code phrases with our device. We chose “SOS”, a phrase that is essential for emergency protocol, to test out. The procedure for doing so is as follows:

1. Hold the DOT button for 3 codes
2. Hold the DASH button for 3 codes
3. Hold the DOT button for 3 codes

Test 1 Result:

Below we have the output collected by the oscilloscope of the Analog Discovery board.

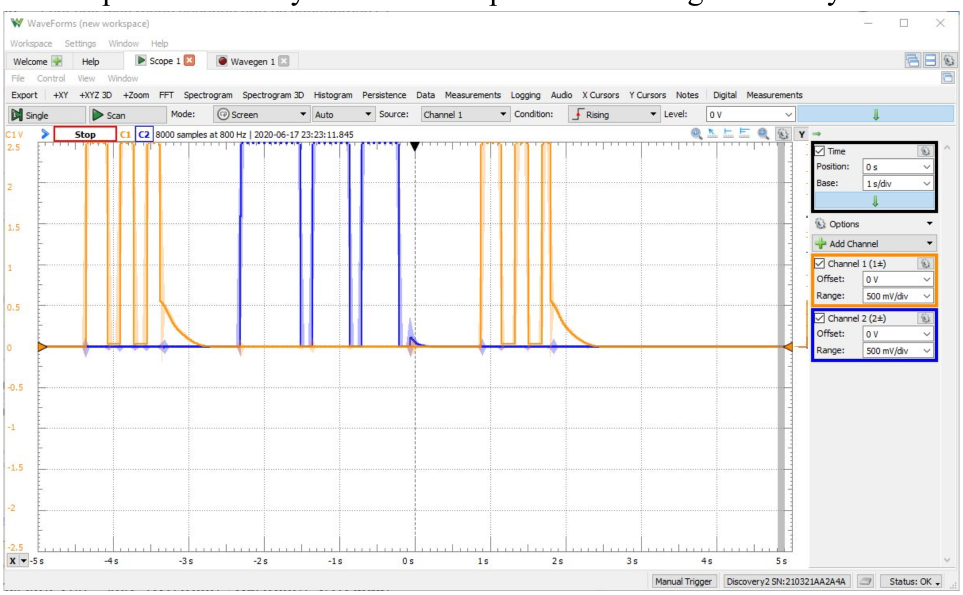


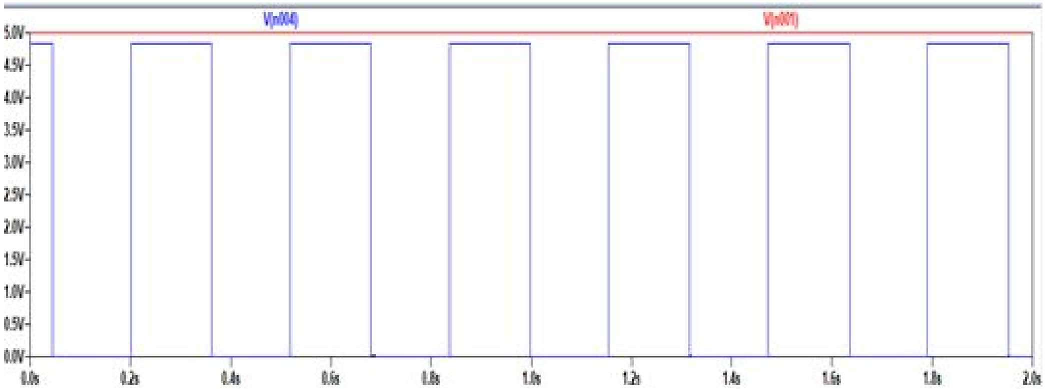
Figure 6: Test 1 Morse code signal “SOS” with physical circuit (.../---/...)

The orange graph is the output of our DOT circuit and the blue graph is the output of our DASH circuit. From this we can see that signal time of the DASH is, as we expected, 3 times longer than the DOT circuit. These when sent to the buzzers will output sound for as long as the output time of our circuit. Thus the final output is exactly as we expected it to be.

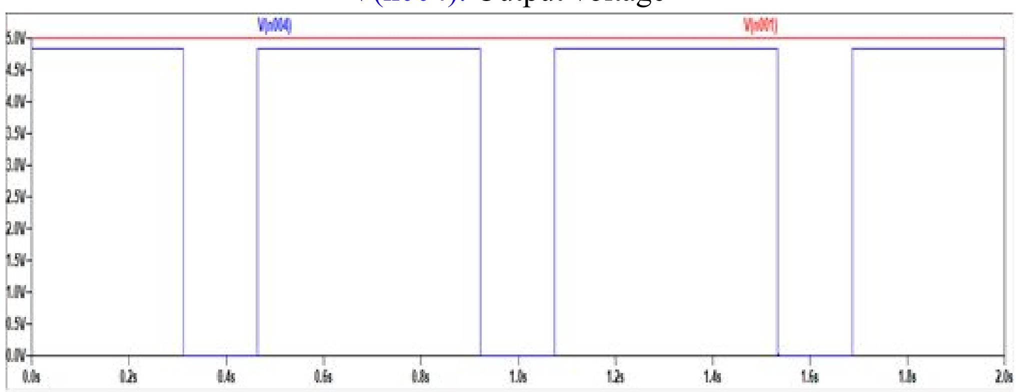
Test 2:

Our final circuit has the added feature of being able to hold down a button, DOT or DASH, and send out a repeated signal of the specified Morse code. To test this we run a simulation of the circuit using LTSpice. We show the repeated output of the DASH circuit and the repeated output of the DOT circuit below.

Test 2 result:



LTSpice simulation for DOT circuit
V(n001): Input 5V DC voltage
V(n004): Output voltage



LTSpice simulation for DASH circuit
V(n001): Input 5V DC voltage
V(n004): Output voltage

From the tables above we can see that the simulation results match perfectly with our definitions and other calculations. The output time of the DOT circuit is 0.17s with the off time being 0.17s as stated in the sound based Morse code standard. The output time of the DASH circuit is 3x the output time of the DOT circuit at 0.51s with an off time of 0.17s again as stated in the Morse code standard. With these outputs, when fed to the buzzer, will output the Morse code inputs perfectly.

Test Results Analysis:

Our circuit works exactly as predicted and operates perfectly in line with Morse Code standard. The 555 timer is able to control both the DASH and DOT circuits incredibly precisely and we are able to reap a clean output signal from that. The buzzer we used is incredibly responsive as well, being able to detect changes in signal that occur in less that a quarter of a second. From the combination of everything mentioned above we are able to meet the requirement of 5-40 words per minute easily.

Conclusion

The choice of the 555 timer circuit design includes knowledge of multiple components, the utilization of the integrated circuits, and understanding of the circuit timing. We initially began the design of a 555 timer based circuit by utilizing the equations of 555 timer duty cycles obtained from the previous Electric Circuits class and electronics available from the Analog Devices parts kit to decide the specific value of every component.

After the values of the electronics were determined, the entire circuit design for both signals was simulated in LTSpice. This was to ensure that the circuit design was functioning properly and that our timing results were matching our mathematical calculations. With the simulation functioning properly the next step was to observe the technical specification of any electronics that might be used in the physical circuit and build the circuit according to the schematic of the successfully simulated circuit.

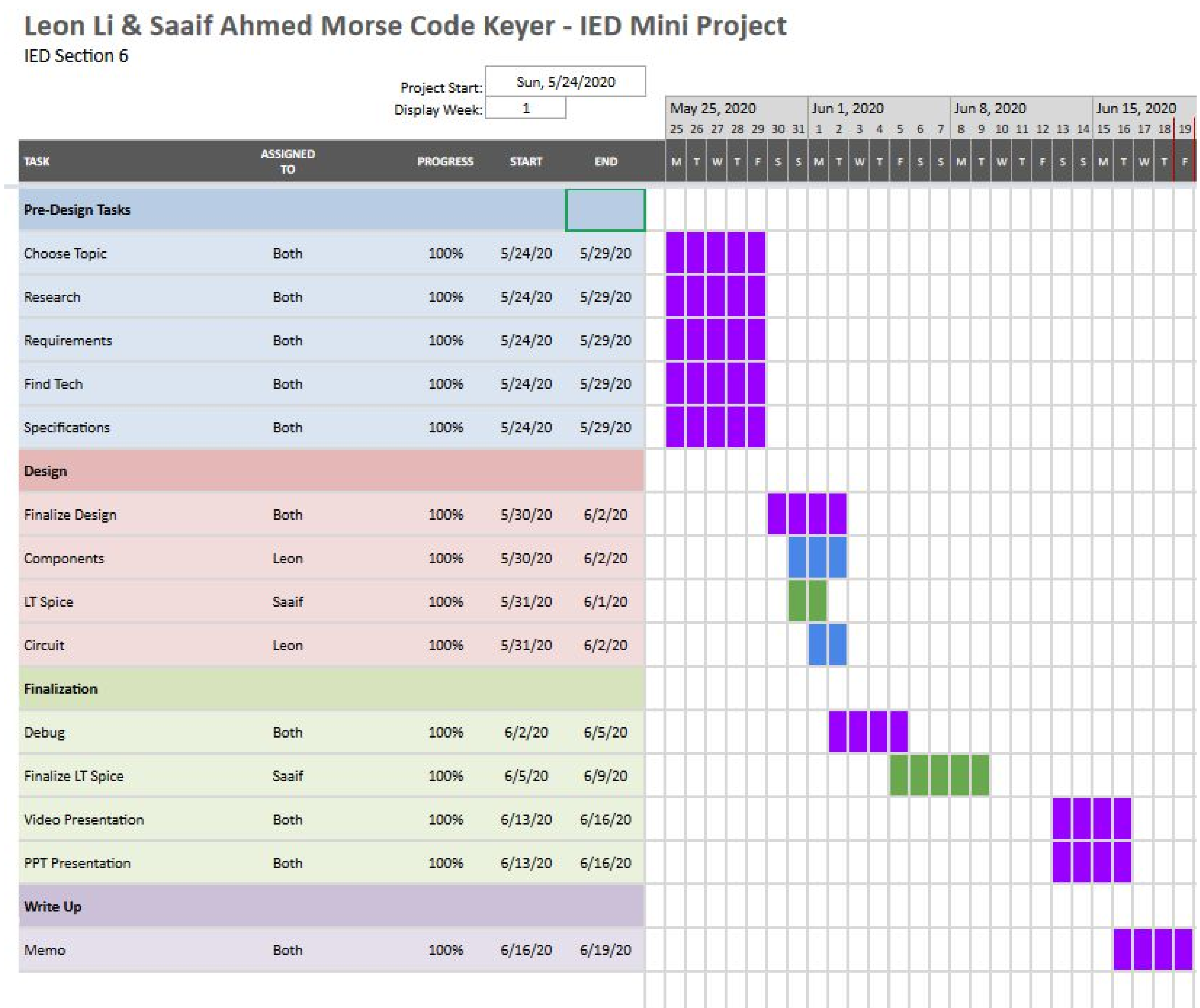
At our first performance test, our design was to have 1 time unit = 0.5 seconds. This allowed the users on average to output 5.6 words per minute. This was on par with the requirement but for very long words, this timing was too slow. Therefore we had to redefine our time unit as 1 time unit = 0.17 seconds. This allows the user to input 16.8 words per minute, which is well within the range of the requirement.

When the 5V output of the 555 Timer was sent directly to the buzzer, a 96dB sound was generated (measured by Physics Toolbox Suite Android Version). This sound was incredibly loud and had to be changed. With the addition of 2 47 Kilo-ohm resistors in series with the buzzer, the sound level was reduced to 56dB which was still loud and below the safety level of 70dB.

The final product of the Morse code keyer timing circuit functions as expected, providing the accurate time for both DOT and DASH signals. The device is comfortable to use as users can continuously input single letters, words, and complete sentences. The Morse code keyer does not appear to have any malfunctions during the 20 minute final test.

Lessons Learned

The most important lesson learned was that consistency is critical to success. From our experience, many teams begin very strong but begin to falter as time goes on and other projects. Our group made a point to remain consistent in goals and expectations as the project went on. Every meeting there was a clear conversation about what we had accomplished, any edits that needed to be made, and what we would aim to accomplish by the next meeting. We divided up work evenly and played to each other's strengths.



TASK	ASSIGNED TO	PROGRESS	START	END
Pre-Design Tasks				
Choose Topic	Both	100%	5/24/20	5/29/20
Research	Both	100%	5/24/20	5/29/20
Requirements	Both	100%	5/24/20	5/29/20
Find Tech	Both	100%	5/24/20	5/29/20
Specifications	Both	100%	5/24/20	5/29/20
Design				
Finalize Design	Both	100%	5/30/20	6/2/20
Components	Leon	100%	5/30/20	6/2/20
LT Spice	Saaif	100%	5/31/20	6/1/20
Circuit	Leon	100%	5/31/20	6/2/20
Finalization				
Debug	Both	100%	6/2/20	6/5/20
Finalize LT Spice	Saaif	100%	6/5/20	6/9/20
Video Presentation	Both	100%	6/13/20	6/16/20
PPT Presentation	Both	100%	6/13/20	6/16/20
Write Up				
Memo	Both	100%	6/16/20	6/19/20

Figure 7: Gantt Chart

The Gantt Chart above shows the vital lesson of timing and organization. In order to keep with the consistency lesson mentioned above, we decided to organize our tasks, roles, and steps of design with a Gantt Chart. Prior to this tool we were taking sub-par notes during meetings, but the inclusion of it allowed us to direct our focus on any specific task we chose.

The next important lesson is the importance of setting definitions. When building from the ground up, we found that there were too many variables we had to consider. When trying to narrow down a design we found we were comparing every single value of each design to the others. To save time and regain focus we had to set our own standards of what we wanted to build. From setting those definitions we were able to eliminate many other designs that would have broken those definitions. At that point we were able to use tools like the concept selection matrix to finalize our design.

There were a few smaller lessons that weren't as impactful as those listed above but still made a change to our perspective of the project. One of the more notable ones was that, working in the same time-zone is something no team should take for granted. It made communication incredibly easy and neither member was worried about the other teammate not seeing the message for multiple hours. Another lesson learned was that it's not necessarily better to make the most effective yet complex solution if the idea cannot be communicated. If only one member is capable of building the circuit, communicating a complicated design is incredibly difficult. With the consideration of the deadline both members had to comply with, it was decided that simple yet effective designs were valued more.

The final lesson that both team members learned was the incredible value in letting people play to their strengths. Even in the scenario of this project which arguably had quite more circuit elements than most other engineering projects, it was critical to have both team members come forward with their strengths to offer for the project. Not only did this have the advantage of certain project design steps being accomplished much faster, but it also enhanced the learning experience. The members weaker in circuit design received a crash course on design principles and those unfamiliar with professional presentation were able to greatly improve upon that.