Building a Preset Piano Arduino

Computer Science 207

Brianna Hoff

April 10, 2020

**Table of Contents**

Introduction………………………………………………………………………………………..3

Inspirations………………………………………………………………………………………...4

Designing Process…………………………………………………………………………………5

Building Process…………………………………………………………………………………..7

User Manual for Preset Piano Arduino……………………………………………………………8

Setbacks and Failures……………………………………………………………………………...9

Milestones………………………………………………………………………………………..10

Conclusion……………………………………………………………………………………….11

References………………………………………………………………………………………..12

Appendix…………………………………………………………………………………………14

*Appendix A*: First Project Build………………………………………………………….14

*Appendix B*: Final Project Build…………………………………………………………15

*Appendix C*: Breadboard Diagram of Final Project Build ………………………………16

Appendix D: ………………………………………………………………………………

Appendix E: ………………………………………………………………………………

**Introduction**

The project that I worked on throughout this semester was to use a beginner’s project found on the Arduino Project Hub of a piano on the Arduino coded with a preset song. In the original project I found online, the song was an Anime theme song called “Unravel” from a show called *Tokyo Ghoul*. To make this project more of my own, I decided to choose a different song, which is “Another Brick in the Wall Pt. 2” by Pink Floyd. This song is one of my favourites from the album and the song essentially has only about five to six notes, so I thought it would be perfect for a simple project such as this one. The reason that I chose this project was because I am very much involved with music as I play a few instruments including the piano, so I wanted to incorporate something that I know very well already (music) with computer science, which is something that I don’t know very well. The idea is to combine the two together to make a new form of music, since I have never coded music on an Arduino before.

This project uses pushbuttons as “piano keys” and each pushbutton represents a different note, so it is replicating a piano. When a button is pressed, the piezo speaker on the breadboard plays a sound. One of the pushbuttons is different from the others because when it is pushed, a preprogrammed song plays in full, but does not loop.

I have seen a lot of different projects online that essentially use the same hardware and code, but each have different songs that play. I my project proposal, I mentioned that I would try to add LEDs to my project and make them turn on and off with the melody as it is playing. I tried several times and several different ways of coding to do this, but due to the restraints of being a beginner coder, I was not able to figure out how to get multiple LEDs to blink along with a melody. Instead, I decided to use my experience with music to create something different out of the melody that I coded, which I will explain in more detail later in this paper.

**Inspirations**

The main inspiration for this project was obviously the one that I based my project from. In the early research days of this project, I found a beginner project on the Arduino Project Hub website from a user named ExeCuteLi. This project caught my eye the most out of the other ones because it involved music, which is something that I am interested in and have experience in. The project is called “Unravel Preset Piano Easy Arduino (Even a Ghoul Can Make It),” created and published by ExeCuteLi in 2018 (ExeCuteLi, 2018). The code was fairly easy to follow as well, which I appreciated since I am not an experienced coder.

Another inspiration that I had that is not related to other Arduino projects is the film “The Wall,” which is basically a film-length music video for Pink Floyd’s album, “The Wall.” I have seen this film many times and listened to the album many times, so I know the song “Another Brick in the Wall Pt. 2” very well from beginning to end. This knowledge really helped me in coding the song on the Arduino because it is important to know how long each note is and which specific note is being played. I had to look up a few videos on YouTube of the specific notes for the song since I am not able to tell which note is played just by hearing it. One video especially helped me in this process and it is called “Pink Floyd – Another Brick in the Wall – EASY Piano Tutorial by PlutaX” (PlutaX, 2020). This video shows a virtual keyboard and shows each note being played on it as well as the duration of the note.

Along with a video of a virtual piano playing “Another Brick in the Wall Pt. 2,” sheet music was also very helpful for me to find out exactly what each note duration was. For example, I could see that the first note (note D) was an eighth note, so I would put a number in the code for that note that represented the same duration as an eighth note. The sheet music that I looked at for this project is from the website “Sheet Music Direct,” published by Hal Leonard (Sheet Music Direct, no date). I will go into more detail in the process of coding the song in the Setbacks and Failures section.

When I was trying to add LEDs to my circuit and trying to code them to blink when the melody is playing, I looked at sources online of code that does this in an Arduino circuit. Both sources of code were first found on YouTube, as I looked up projects that use LEDs along with melodies. In the descriptions of the YouTube videos were links to their code on GitHub, which I found useful to a certain point. The first video that I found (Li, 2013) plays the United States National Anthem on the Arduino and there are three LEDs that turn on with the song, one after the other. The code (Li, 2013) seemed fairly simple to use and to incorporate in my project, so I tried using it. This process will also be discussed further in Setbacks and Failures.

The second video (suskind7, 2013) that I found also involves multiple LEDs, but the creator of this project used LED patterns to perform while the melody is playing rather than the LEDs blinking with the beat of the melody. After trying the code from the first video, I thought that this process of making LEDs blink might be easier to try with my project. The code (suskind, 2013) is slightly similar to the code of the first project, but it is longer and uses some different functions. I tried to follow along with this code as well for my own project, but it proved to be a little too advanced for my skills in incorporating it into my project.

**Designing Process**

The design process was quite simple for this project. I followed the steps of the original project that I based mine from and used one piezo speaker, multiple pushbuttons (the project called for five pushbuttons, but I added one more for another note), 1kΩ resistors (I did not have this type of resistor, so I had to improvise and use 10kΩ and 2.2kΩ resistors), and jumper wires. The project uses the Arduino UNO and a breadboard for constructing the circuit. My build from the original project is shown in Appendix A on page 14.

As previously mentioned in my project proposal, I decided that I wanted to try to add LEDs to my circuit so that they light up as the melody is playing. When this did not go according to plan, I resorted to adding one LED to the circuit and coded it so that it turns on once a pushbutton is pressed and turns off once the button is not being pressed anymore. Although this LED does not light up when the song is playing, I would find it useful for if the pushbuttons were being used as an actual Arduino keyboard so that you can create your own song just by pushing the buttons and have a small light show as the buttons are being pressed.

After trying to design my project around having multiple LEDs with the help of examples of code from other projects, I just added one LED to pin 13 with an additional 560Ω resistor. With limited space on my breadboard due to six pushbuttons already on the board, the only space that I could really add a LED was right next to the piezo speaker.

Once my hardware and code were finished, I planned how I could make this project even better. I came up with the idea of using the song that I coded on the Arduino and playing along with it on an actual piano. I wanted to add one more instrument as well, so I asked my boyfriend to play along with it on his guitar. Once each instrument had played along with the Arduino, I edited the videos together on Adobe Premiere Pro so that it sounded like they were all playing at the same time, and the final product was essentially a mini orchestra that played “Another Brick in the Wall, Pt. 2.” Refer to Appendix D and E for the process of how I created the video of multiple instruments playing with the Arduino.

**Building Process**

The building process of this project was simple initially as I first used the build from ExeCuteLi’s project and built onto it from there. It was the process of adding the LEDs that I began running into trouble, but it was mostly with the code that I was not able to figure out the problems that occurred.

When I added the LEDs to my board, I replaced every pushbutton with an LED aside from the button that you press to play the preset song. I also switched around the wiring so that 5V and GND were connected at the bottom of the board rather than GND connected to the top of the board. I repositioned the resistors so that they were also at the bottom of the board and connected each resistor to the LED, which was then connected to a pin. For the code, at first I copied and pasted the code from the project that I found on YouTube (Li, 2013) to see if it would work on its own, but it was not working on my Arduino. Then, I tried to incorporate the LED part of the code into my own project code, but the piezo speaker was making weird buzzing sounds even without pressing any buttons. I tried to fix it by checking that the code was consistent with my own variables and in the correct place in the code, but nothing was really working for me as I am not great at problem solving when it comes to code.

The other code that I found from another YouTube video (suskind7, 2013) also did not work with my project. Once I had everything wired up according to this code, my preset song would not even play and no sound at all was coming out of the piezo speaker even when I pressed any button. This is when I resorted to using only one LED in my circuit because I already knew how to code it to at least turn on when a button is pressed since this is a skill we learned in the lab.

The final version of this project consists of one piezo speaker, three 10kΩ resistors and three 2.2kΩ resistors for the pushbuttons, one 560Ω resistor for the LED, one red LED, six pushbuttons, and ten jumper wires. It can play one preset song of a portion of the song “Another Brick in the Wall, Pt. 2.” Refer to Appendix B and C for the final build image and breadboard diagram.

**User Manual for Preset Piano Arduino**

The Preset Piano Arduino is very easy to set up if the breadboard diagram is constructed correctly. All that is needed to be done to replicate this project is to build the circuit according to the diagram and download the code that I have provided and uploading it to the Arduino. All of the required parts came with the Arduino kit that we were required to purchase for this course with the exception of the pushbuttons. There were more pushbuttons needed for this project than the number of pushbuttons that came with the kit, so I had to order some extra buttons online. However, the buttons are quite easy to find online and I had no problem with ordering them via Amazon. Also, pushbuttons are quite inexpensive, which is a plus.

If someone were to replicate this project but wanted it to play a different song, it is possible to do so with this code. All that would need to be done is to change the notes for each pushbutton to the desired notes of a different song in the int melody [ ] array as well as change the durations of each note in the int noteDurations [ ] array so that it sounds like the specific song. It can be a long process to do this if it is a more complicated song, such as the one that I chose, but it can be done.

Another element of the code to take note of is the “pitches.h” library that needs to be saved in the same folder as the .ino file in order for this code to work. I had troubles with this step at the beginning stages of my project as I tried to save the “pitches.h” library into the libraries of the Arduino program, but it is a much simpler process than this. Other than these minor steps to make this project work successfully, it is an easy process to follow and make this project into your own little sound system.

**Setbacks and Failures**

To get to the final stages of my project, I had to go through a lot of trial and error processes for things to work properly on my Arduino. Most of my problems stemmed from the code, which was not surprising to me because, as mentioned before, I am not an experienced coder in the slightest. My first setback was trying to get the initial code to work from the project on Project Hub. When I tried to upload the code to my Arduino, I kept getting an error message saying “the specified folder/zip file does not contain a valid library.” This error message was directed at the “pitches.h” file that was included in the code. At first, I thought that it was an ordinary library file and I saved it to the Arduino libraries, which is something that I learned in the lab. After several error messages, I resorted to getting help from the professor and I was told that the “pitches.h” file was just a bunch of definitions, so I just needed to save it into the same folder as my .ino file for my code. This process fixed my problem, so I was able to go to the next steps of my project.

Another setback that I had was coding the song. I had no troubles with changing the notes to match my song that I wanted to use, but it took quite a while and a lot of trial and error to get the note durations correct so that they sounded like the song I wanted to use. Written in the code from the creator of this project (ExeCuteLi) were instructions about the note durations and that specific numbers to use represented a whole note, half note, quarter note, etc., but these numbers did not work for me as some made the note go too fast and others made the note go way too slow. It took a lot of trial and error to find the right numbers that would make each note sound like how it was supposed to be played. This step in my project definitely took a lot longer than I was expecting and was a big setback in the process.

The major setback and failure that I had during this entire process was trying to change the code so that LEDs could light up as the song was playing. This process was previously explained in the Building Process section. I tried to use a few examples from other people’s code to incorporate into my project, but nothing worked well for me as I was not able to solve where the problem was occurring in the code. This setback also took some time to work through as this was the main change that I wanted to make to the original project that I found online. My only solution to this was to just add one LED to the board so that it would at least light up as a button is being pushed. I am disappointed that this addition of LEDs didn’t work properly for me as I would have really liked to have been able to code this part properly in my project. It would have made the project look visually pleasing when it is being played because you would have something to listen to and see at the same time.

**Milestones**

I would consider this project a partial success because I was able to achieve properly coding a different preset song onto the Arduino rather than leaving the same song in the code that already came with it. My main goal was to change the song, so this is why I consider it to be successful. The addition of the LEDs was one extra step that I wanted to achieve, but it was not done to my initial expectations.

The original milestones that I came up with in the project proposal were not all met at the date that I specified. All of the parts that were needed for this project were gathered in due time at the very early stages of this project, so that milestone was met properly. My initial assembly on the Arduino and breadboard milestone was also met at the date that I specified, but all other milestones included the code and LEDs, which were not met due to setbacks in the project. My stretch goal of adding another preset song that can be played on the Arduino was also not met or finished due to time constraints.

My idea of creating a video of the preset Arduino song being played along with a real piano and a guitar came from not being able to get the LEDs to work properly in the project, so this was a milestone that I created several stages into my project. However, the whole idea of this project revolves around using the Arduino as a medium for music, so I think this has been achieved because it shows that the Arduino can be played alongside other instruments as well.

**Conclusion**

The Preset Piano Arduino project that I worked on throughout this semester has been an eye-opening experience for me because I found that coding and solving problems with hardware is a difficult process to learn. I have gained knowledge throughout this process, such as learning that music can be made on an Arduino, but it just takes time and practice. This process can be thought of like learning how to play an actual instrument. It definitely takes practice to learn the basics and then to apply what you have learned to play a note or even a whole song. It can take weeks to learn a song, just like it can take weeks to learn how to code properly and how to wire hardware onto an Arduino and breadboard.

Before I saw this project on the Arduino Project Hub, I never would have thought that you could replicate a piano on an Arduino and have it play a whole song. Although my project presented a few problems for me in the process of creating it, I am happy with the final product as I find it to be a good integration of this course with my love of music.

**References**

ExeCuteLi (May 25, 2018). *Unravel Preset Piano Easy Arduino (Even a Ghoul Can Make It)*. [online] Arduino Project Hub. Available at: <https://create.arduino.cc/projecthub/executeli/unravel-preset-piano-easy-arduino-even-a-ghoul-can-make-it-17c472?ref=platform&ref_id=424_trending__beginner_&offset=217>. [Accessed 24 January, 2020].

Li, Jianan (March 21, 2013). *LED Music Player – My First Arduino Project.* [online] YouTube. Available at: <https://www.youtube.com/watch?v=mu7xcc0QeEo>. [Accessed 25 March, 2020].

Li, Jianan (October 4, 2013). *Misc-arduino-sketches/LEDMusicPlayer*. [online] GitHub. Available at: <https://github.com/Jianan-Li/misc-arduino-sketches/blob/master/LEDMusicPlayer/LEDMusicPlayer.ino>. [Accessed 25 March, 2020].

PlutaX, Peter (February 14, 2020.) *Pink Floyd – Another Brick in the Wall – EASY Piano Tutorial by PlutaX.* [online] YouTube. Available at: <https://www.youtube.com/watch?v=ABEz9qJWeiQ>. [Accessed 26 February, 2020].

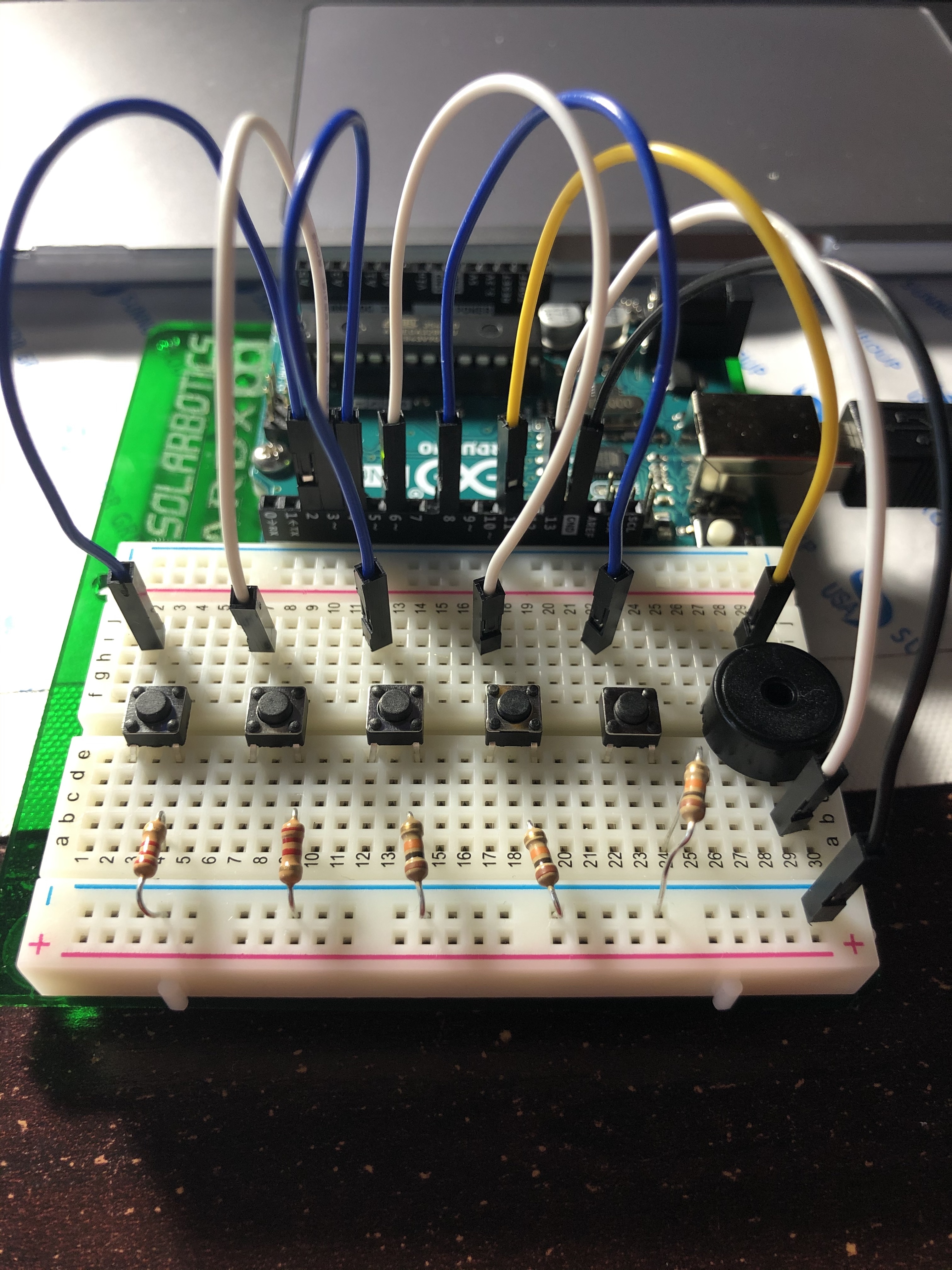
Sheet Music Direct. *Another Brick in the Wall, Part 2.* Words and Music by Roger Waters (1979). Published by Hal Leonard. [online: digital sheet music] Sheet Music Direct. Available at: <https://www.sheetmusicdirect.com/se/ID_No/96962/Product.aspx>. [Accessed 30 March, 2020].

Suskind (May 24, 2013). *Arduino-knight\_rider\_theme*. [online] GitHub. Available at: <https://github.com/suskind/arduino-knight_rider_theme/blob/master/knight_rider.ino>. [Accessed 25 March, 2020].

Suskind7 (May 24, 2013). *Arduino – Knight Rider – LEDs & Theme song*. [online] YouTube. Available at: <https://www.youtube.com/watch?v=lsbC4TEz_P0&list=WL&index=27&t=0s>. [Accessed 25 March, 2020].

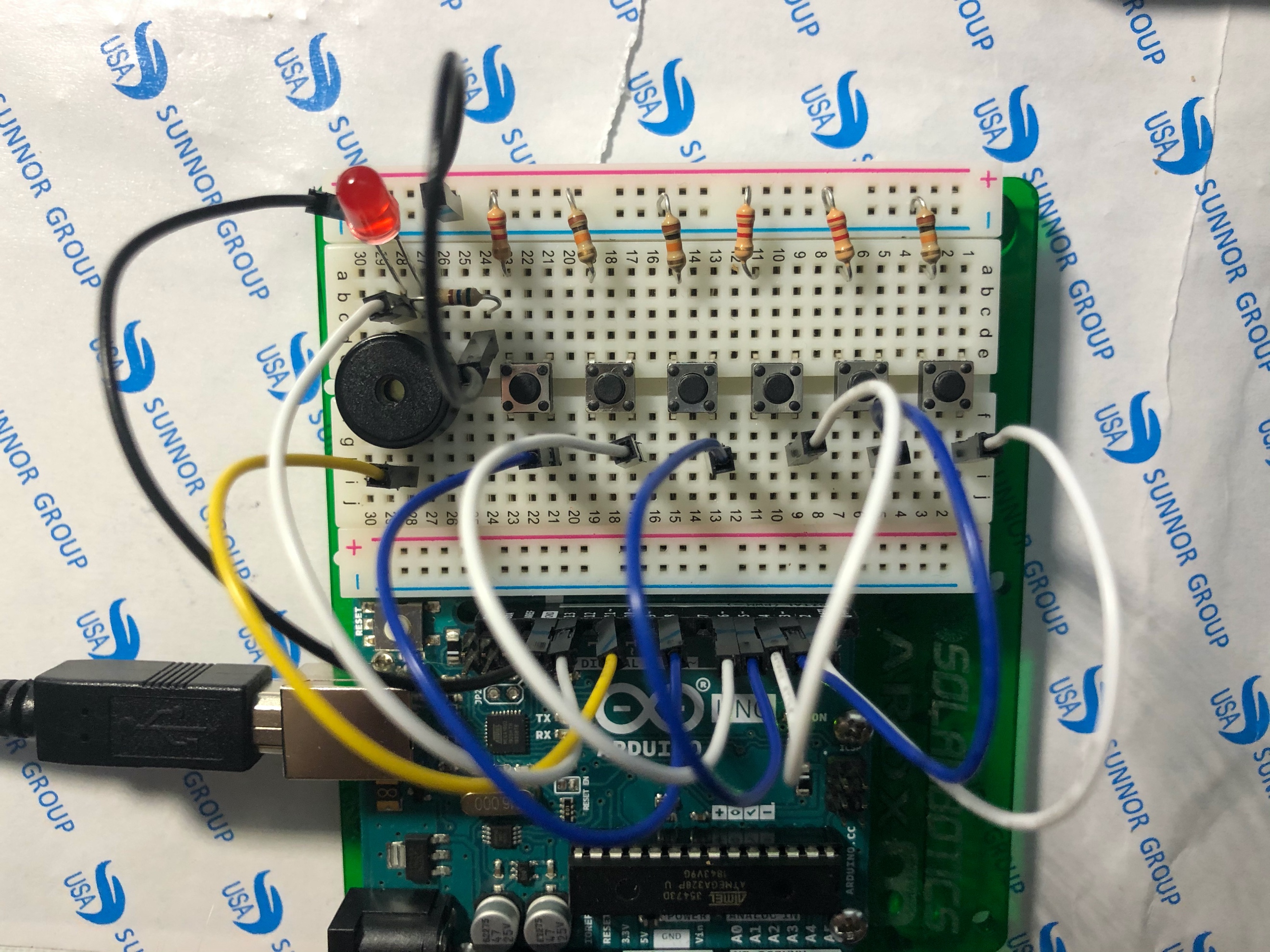
**Appendix**

*Appendix A*: First Project Build



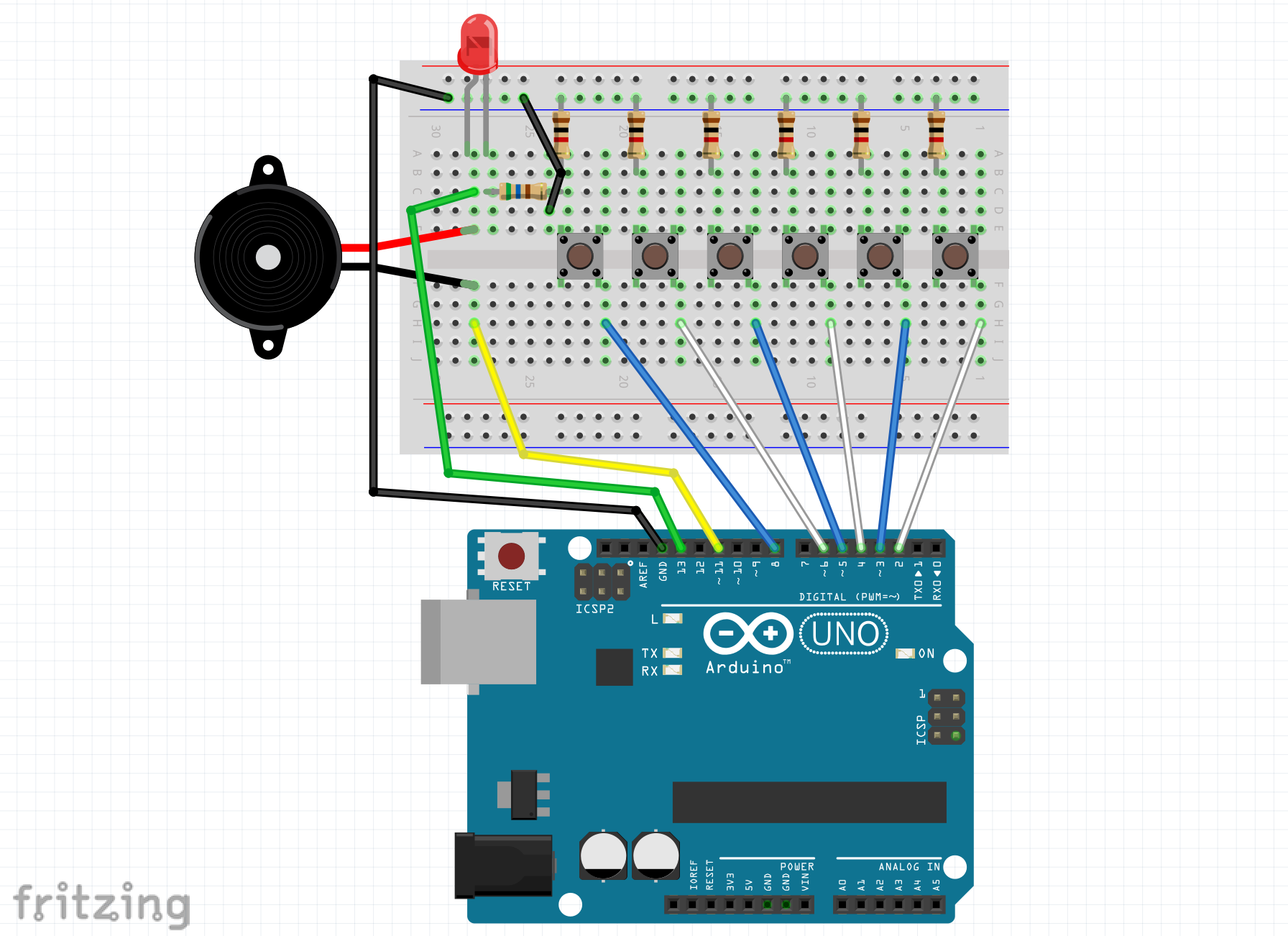
This is what my breadboard looked like when I first built it exactly like the project on Project Hub.

*Appendix B*: Final Project Build



This is what my breadboard looked like as the final project build.

*Appendix C*: Breadboard Diagram of Final Project Build



This is the breadboard diagram that I created on Fritzing of the final model of my project.

*Appendix D*: