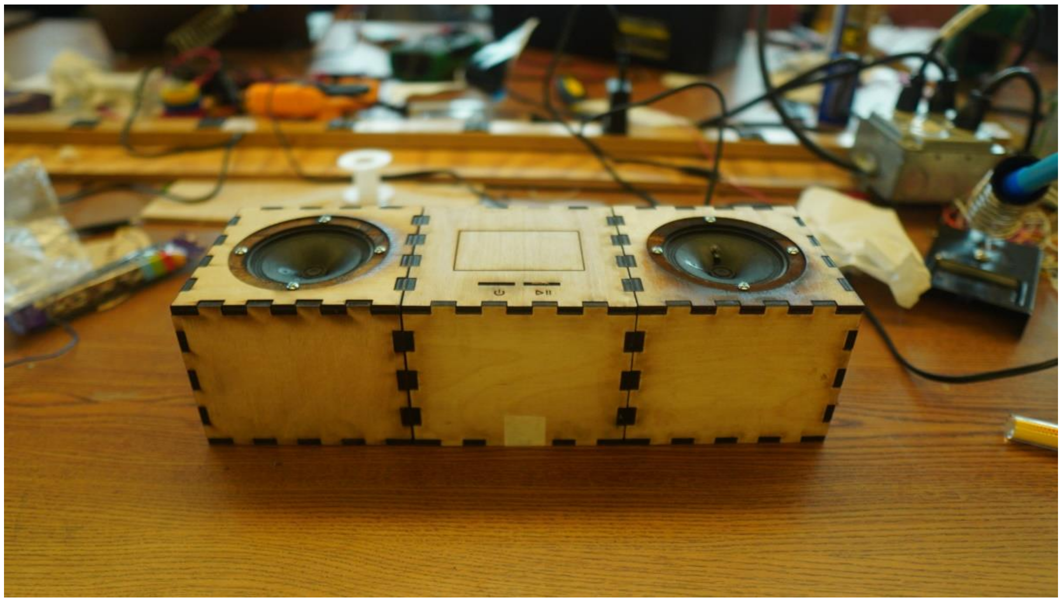
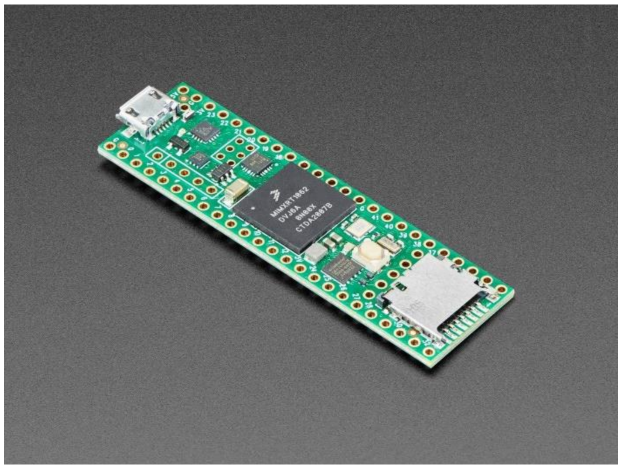
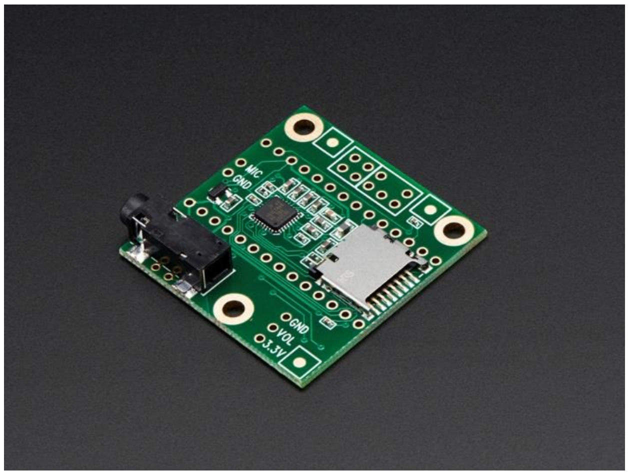
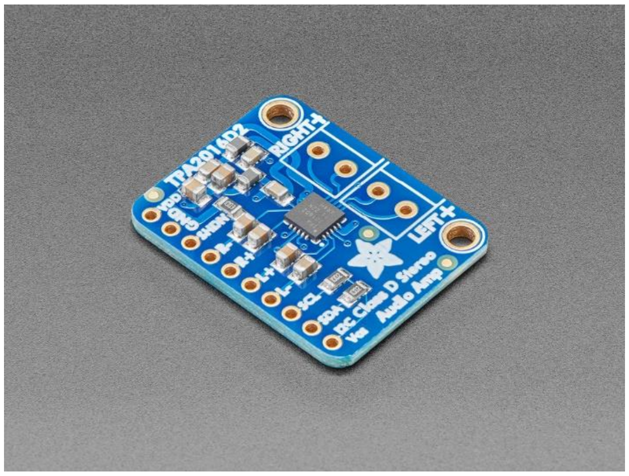


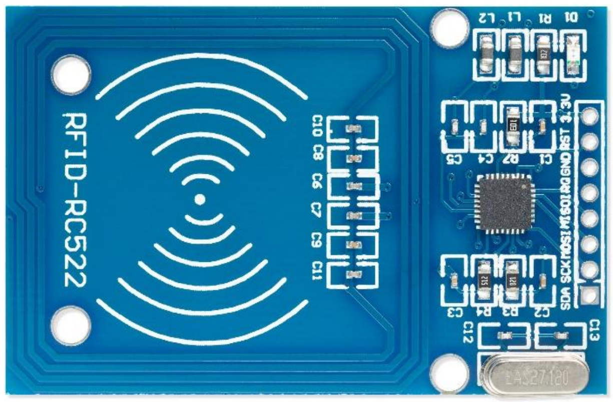
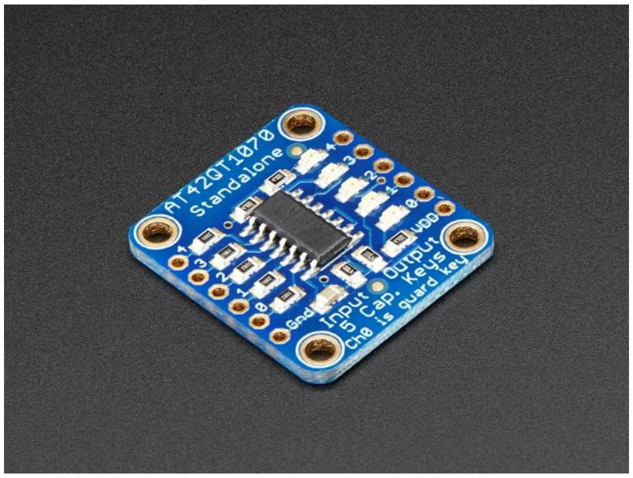
RFID Sound box Documentation

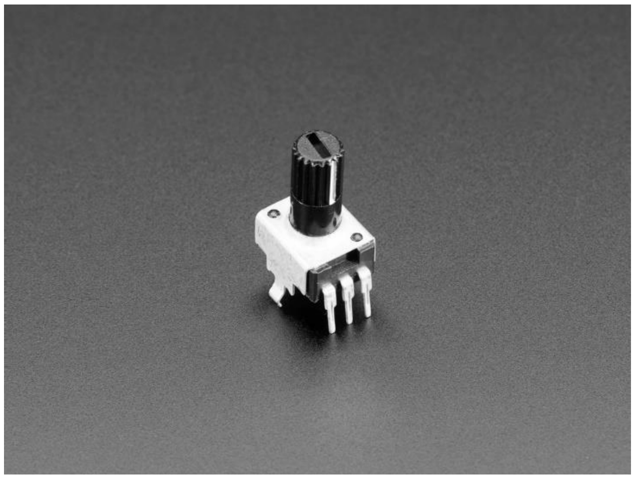
Courage Agabi

Section A

Part 1 – Hardware components

Teensy 4.1	
Teensy 16bit audio shield	
Adafruit 2.8W amplifier	

3W 4-ohm impedance speakers	
RFID reader	
Capacitive touch sensor	

10k potentiometer	

Part 2 – Libraries

MFRC512 library for RFID functionality (https://github.com/miguelbalboa/rfid.git)

TPA2016 for controlling amplifier (https://github.com/adafruit/Adafruit-TPA2016-Library.git)

Teensy audio library for controlling sgtl5000 chip in audio shield (https://www.pjrc.com/teensy/gui/)

Bounce library for touch input debouncing (https://github.com/thomasfredericks/Bounce2.git)

Wire library for I2C and SPI library for RFID reader

Part 3 – Custom components

RFID read and write

The MFRC library had functions to read and write hex values to a card but I needed to specify what part of the tag the data was written into and also add delimiters to know where to start and stop reading. I also needed to automatically convert the file name into hex and back into text when read. It would have been possible to do all this with the given functions but using this custom approach gave me a better understanding of the RFID tag system and how it can be used for projects that require higher encryption. It also helped with debugging read/write issues.

Audio amp node for volume control

The sgtl5000 chip allows you to change the voltage levels for the line out outputs. However, the result is a discrete step in volume change. I wanted continuous, smooth transitions in volume so I added an audio amplifier between the SD card wav node and the audio shield I2S node (see image below)

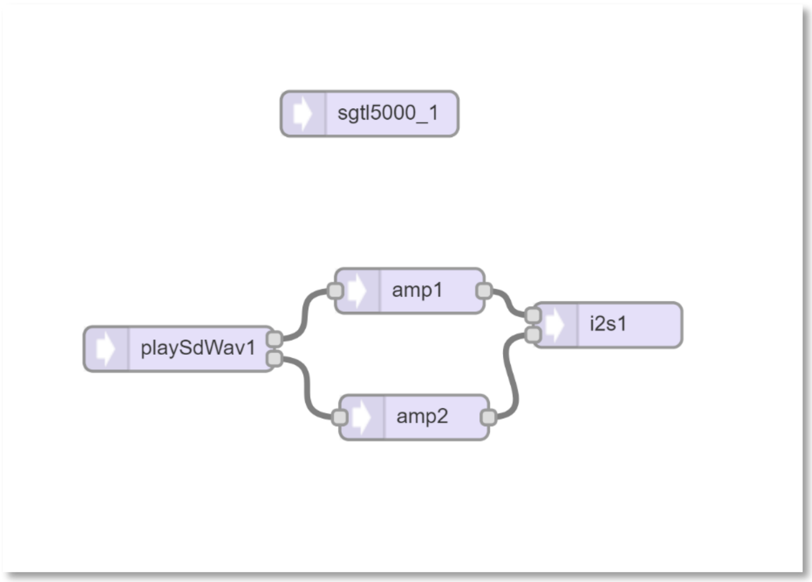


Figure 1. Sample audio configuration from PJRC graphic interface.

Hardware mounts

Finally, I designed hardware mounts for the audio shield, amplifier and power source. I did not want to glue these pieces to the plywood in case I needed them for other projects so I 3D printed these plates with side rails to hold the components. There are also notches on the side of the rails so that rubber bands can be wrapped around and placed in the notches. Lastly, the plates have extruding legs underneath that insert into holes in the plywood.

Part 4 – Skill learned

RFID data manipulation

I learned more about RFID tags than I care to admit. I learned potential ways to create more secure RFID tags such that it would be a reliable option for a household door locking system. Basically, you can change the key that the RFID tags have by default. The key is what you need to read data off the card. All tags have a default key but if you have a custom key it would be harder to use any key to open a door lock. And combined with the actual data on the key, the number of possible combinations increases exponentially (although there are still ways to get around this if you have unrestricted access to a person's key).

3D printing best practices

I learned that when trying to fit or mount 3D printed pieces together, it is best to make more generic designs rather that hyper specific designs since the degree of accuracy of the 3D print is not always perfect. So, rather than have prongs sticking out from a 3D printed plate that fit specific mounting holes on a circuit board, you can just print rails that hold on to the board from the sides.

Part 5 – Iterative process

There were many moving parts in the assignment, so I worked on incremental features…

1. Getting the RFID to read and write the file name – this was fairly easy given that there is existing code to do this. I just had to understand where the data was being written and in what format
2. Getting the audio shield to play the song – this was a bit more difficult because of a few unknown bugs. The filename had to be in 8.3 format (excluding the directory) and it had to be all capital letters. So, for a song like "Stevie Wonder – Sir Duke" in the folder called "Stevie", the data on the tag would be "STEVIE/SIRDUKE.WAV". It was little things like this that made play back hard at first.
3. Once the tag had the right file format and the audio shield could read that file from the SD card, then the audio would play from the headphone jack
4. Getting the touch input – the capacitive touch sensor was also tricky because unbeknownst to me, it did some super quick calibration when power was turned on so the range would differ every time. However, getting it to work was straightforward with the Bounce library. The audio shield also had an updated play and pause method that was added recently so that made playback control easier.
5. Finally, I had to get the amp and speakers connected. The issue was with the wiring of the amp and potential ground loop which cause a lot of noise in the music. I fixed this issue by giving the amp and independent ground reference from the rest of the components (audio shield LR outputs and ground connections). Also, the design of the speakers was modular so the amp would struggle with properly powering the speakers if they were attached at different times. I have not found a fix or done enough testing to figure this part out

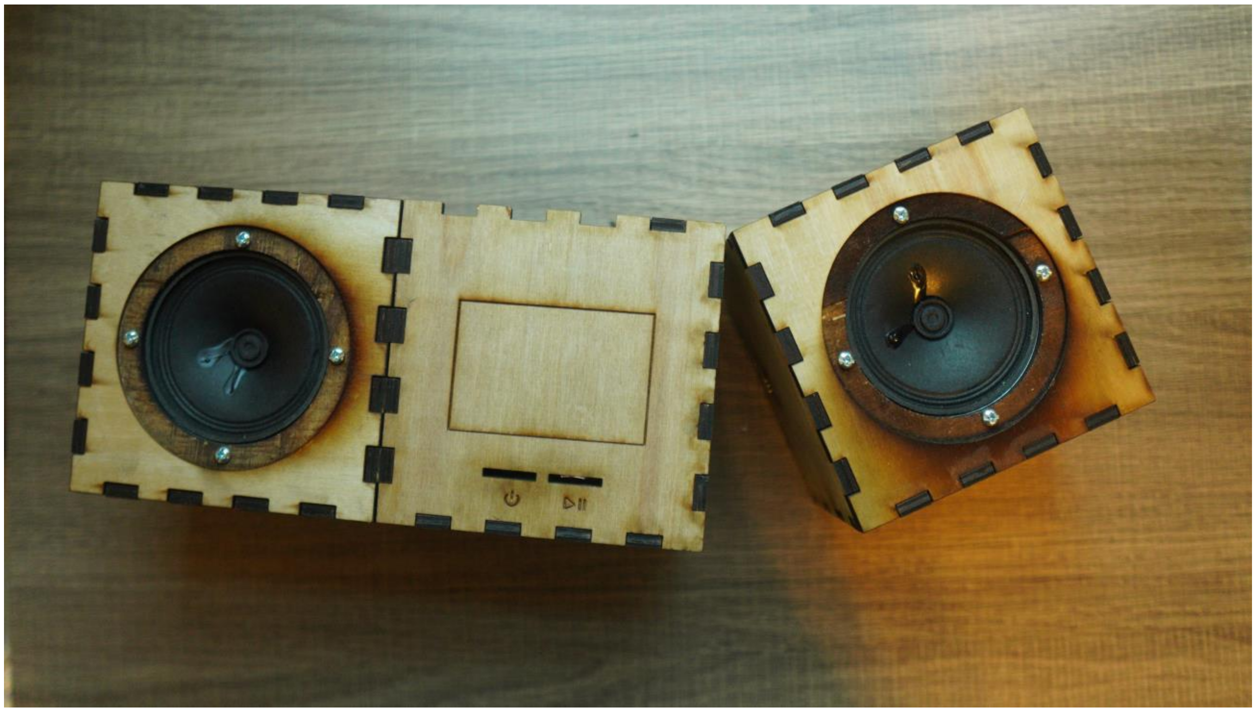
Ideas that didn't pan out…

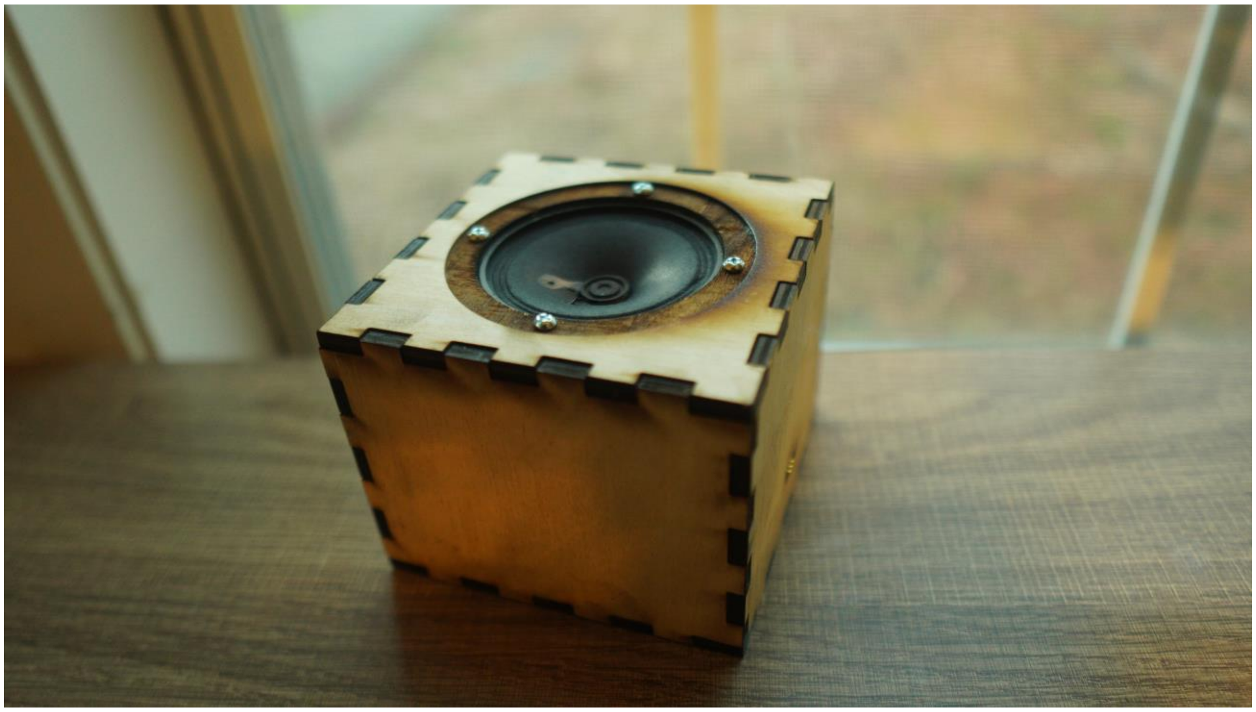
1. Connecting to Spotify – the audio amplifier issue was such a big problem for me that I was more focused on getting the best sound out of the speakers before adding the extra functionality. Also, the internet requirement would have meant that I had to buy another board (ESP-32) to connect the Teensy to Wi-Fi which would have been more expensive
2. Adjusting playback – I wanted to be able to rewind a song or just have all the songs play on shuffle but there was no straightforward way to adjust playback and I didn't have time to implement the latter
3. Low power mode – The teensy has an option for low power mode where it basically shuts off when you connect the power ON/OFF pin to ground. I wanted to implement this with a transistor such that current flowed when the capacitive touch input drove the base high. I wasn't sure if this would work and I didn't want to try anything that might damage the Teensy so I just manually shut down the power supply.

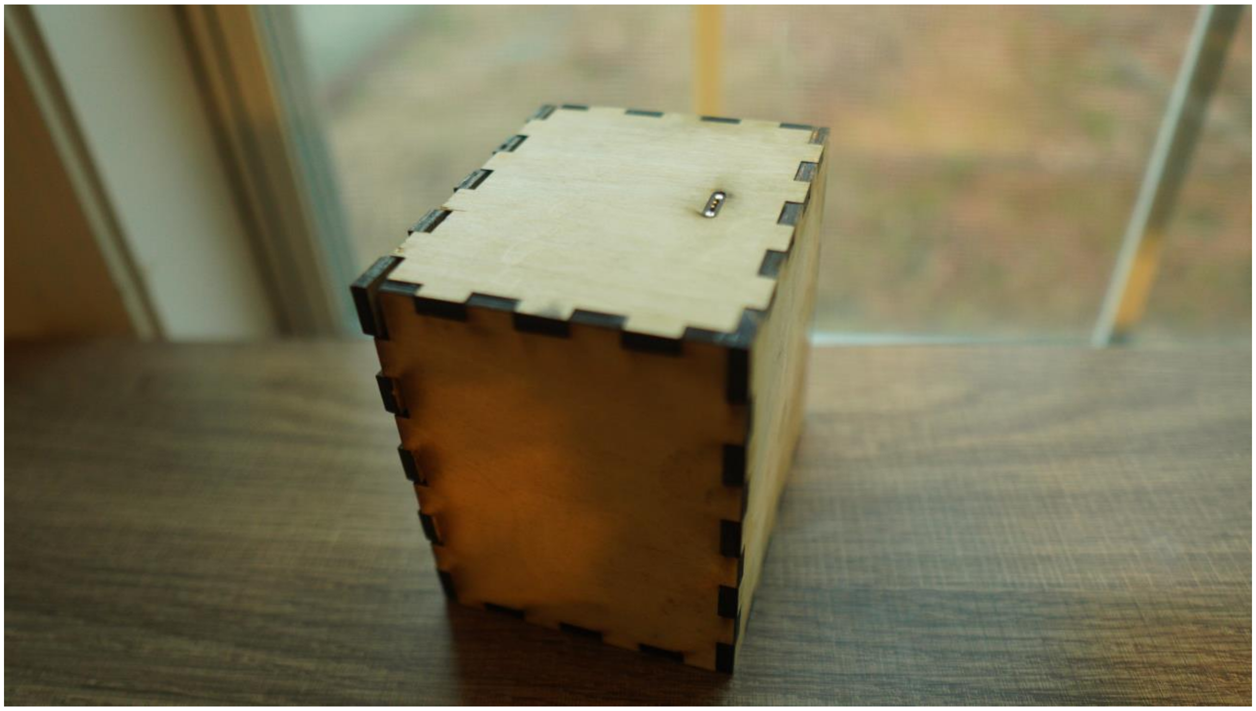
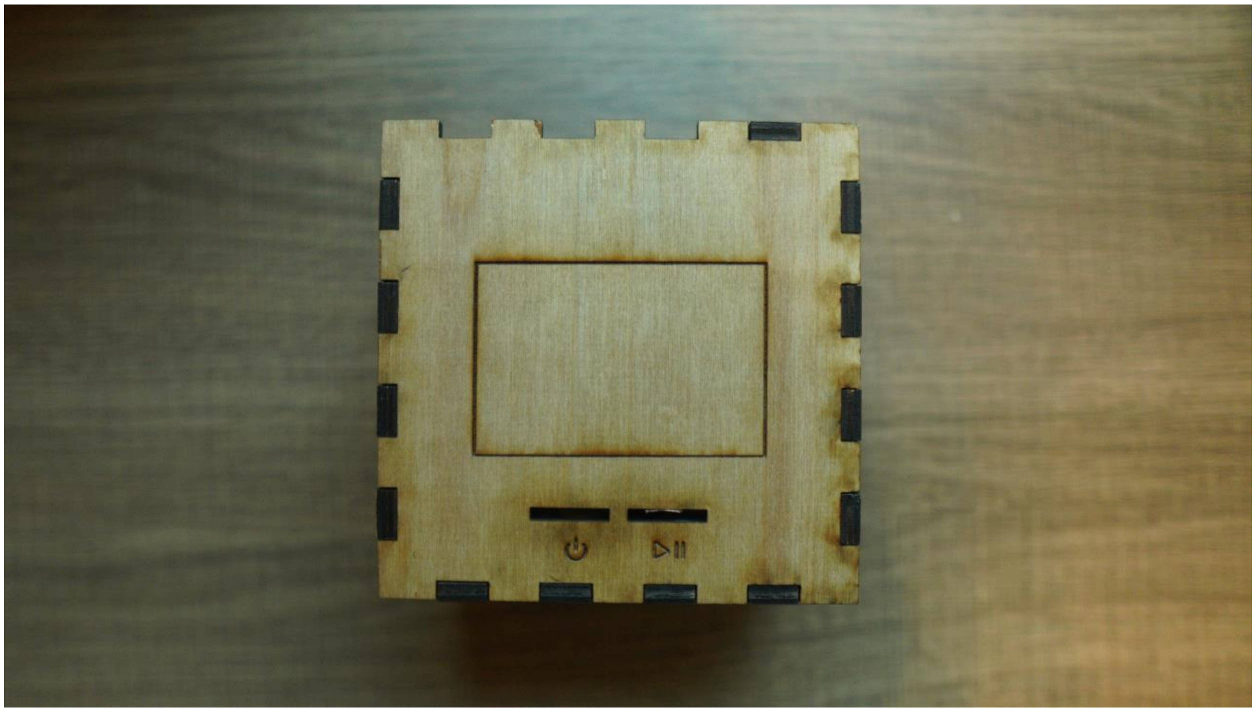
Part 6 – Lessons learned

1. The most important lesson is probably that when working with audio, you have to make sure that you ground pins are properly connected to avoid audible noise problems. I thing a soldering solution would have been better and I think I should have started with just the amplifier and speaker and the DAC on the teensy before introducing the audio shield and the multiple ground connections.
2. When 3D printing, leave room for error and design more generally so that all the measurements don't have to be ultra-specific for the piece to fit

Part 7 – Demo video and images





*** see zip file for demo video*

Part 8 – Code

***check zip folder for .cpp files (under the src folder)*