1984

Macintosh 128K

Redesigned, Remodeled, Reengineered, Re-everythinged

1984

Macintosh 128K

Redesigned, Remodeled, Reengineered, Re-everythinged

IN MEMORY OF STEVE JOBS

Table of Contents

List of Components

Active:

Arduino Uno

- Product page
- Schematics
- Dimension Drawing

uLCD-43-PT-AR Resistive Touch LCD

- Product page @ Sparkfun
- Product page @ 4D Systems

RN-52 Bluetooth Breakout

- Product page
- Datasheet
- Schematics
- Command Reference Guide
- Hookup Guide

Speakers x2

Stereo 2.8W Class D Audio Amplifier - I2C Control AGC - TPA2016

- Product page

Inactive:

IR Receiver Breakout - TSOP85

- Product page

DS1307 Real Time Clock breakout board kit

- Product page

Breakout Board for VS1053 MP3 and MIDI

Transcend Micro SD with Adaptor

Miscellaneous

Jumper Wire - 0.1", 5-pin, 6" - connect LCD to the Shield

- Product page

USB Cable A to B - 6 Foot

Rocker Switch - SPST (round)

- Product page

Intro

Being a passionate Apple fan, I came across this idea when I saw a link to a Mac 128K teardown by ifixit, celebrating the 30th aniversity of Macintosh.

EE 47 gave me enough knowledge to recreate this brilliant personal computer and Product Realization Lab gave me access to 3D printers and other design tools. First I would like to thank David Sirkin and the rest of the teaching team, Nik, Jane, Samyuktha, who helped me in my labs and offered me many suggestions on my final projects. Next, I would like to thank Marlo, Nick, Michael, and Dan for helping me with every detail on designing this mac.

Setup. . .

Well, I had no knowledge of any kind in electrical engineering prior to my arrival to Stanford. All I knew is some basics about voltage, current, resistance, and V=IR. With each lab, I gained more and more knowledge. Starting with lighting LEDs, analyze digital and analog components, and ultimately led to this final project.

Without furthur ado, let’s dive in to the birth of the new Macintosh 128K.

Arduino

The new Macintosh 128K uses Arduino as the micro processor. Because the Resistive Touch LCD comes with a shield, I bought an Arduino Uno instead of using the Arduino Micro.

Serial Communication

Although looks bigger, it does have some disadvantages compares to the Micro. First, USB and RX+TX share the same Serial Communication Interface, which means that if you want to print stuff to Arduino IDE Console (or in my case, the LCD uses default Serial Communication), you are not able to use RX+TX to communicate with another component. (e.g. Bluetooth Chip) Only option is to use SoftwareSerial library. Although claimed to be able to communicate with baud rate of 115200, the SoftwareSerial library is not perfect at communicating at this speed and most likely gives out garbage data when using mySerial.read(). I was lucky that the RN-52 bluetooth chip supports 115200 and 9600 baud rate so my program was working properly after I lower the baud rate.

Pin Interrupt

This is a great feature but unfortunately the Uno only has 2 interrupt pins. Micro has more… A lot more… Not a lot… But… Well… At least more… The LCD already took the 2 interrupt pins and I surely didn’t want to modify its library to accomadate my need. There is, however, a library to enable all pins to be interruptable. If money is not a problem, you could purchase an Arduino Due which “has powerful interrupt capabilities that allows you to attach an interrupt function on all available pins”. PinChangeInt allows any pins to become interruptable, but SoftwareSerial library also uses it that causes compatibility issue. It is, however, possible to change the library to make them play nice, but nah…

Problems/Limitations

- SoftwareSerial not as fast as it states, use with caution to avoid garbage data.
- Uno only has one Serial Interface (No Serial1).
- Pin Interrupt only availables on two pins for Uno.
- To solve the problem with lack of available Pin Interrupt pins, if your loop() function goes through pretty fast each time, check the state of pin in there and here is an example.

```
int prevGPIO2 = HIGH;
...
void loop() {
   detectBluetoothChange();
}
```

```
void detectBluetoothChange() {
  int bluetoothInterruptState = digitalRead(8);
  if (prevGPIO2==HIGH&&bluetoothInterruptState==LOW){
    prevGPIO2=LOW;
    digitalWrite(13,HIGH);
  }
  else if (prevGPIO2==LOW&&bluetoothInterruptState==HIGH) {
    prevGPIO2=HIGH;
    digitalWrite(13,LOW);
    detectBluetoothChangeHelper();
  }
}

void detectBluetothChangeHelper() {
  // Handle interrupt
}
```

4D Systems uLCD-43-PT-AR Resistive Touch LCD

I was looking for a LCD that is just the right size because it would determine the new Macintosh 128K's size. Not too big that I won't be able to carry it home, and not too small that I won't be able to fit all my components within. 4.3" colored display is a great choice. While browsing through Adafruit and Sparkfun, I found this. 4.3", colored, and it's a TOUCH SCREEN! Why not? The LCD comes with a shield, a jumper cable, and 4D System's own IDE Workshop.

Workshop

The IDE offers 4 kind of programming environment, Designer, Visi, Visi-Genie, and Serial. The LCD itself is a powerful component with its own processor, speaker, RX+TX pins, and many more. But for it to communicate with Arduino, Visi-Genie is the only option. Visi-Genie is a drag-and-drop fast graphic design environment. It is, indeed, fast and easy. Although having limitation, objects such as images, buttons, text, and strings on the LCD can be manipulated by Arduino through Serial Communication. Also, Arduino is able to capture the user interaction such as a button press. Each state of screen is called a form, that can be changed at runtime.

Shield

The LCD comes with a shield that can be easily attached to an Arduino Uno. On top of the shield, there are 5 pin male connectors to which the jumper wire can connect to.

Visi-Genie Library

It is super easy to communicate with the LCD using its own library for Arduino. The first thing is to include the library `#include <genieArduino.h>`. Define the genie object with `Genie genie;`

Interaction

Then setup Serial Communication with

```
Serial.begin(115200); // Serial0 @ 115200 (115K) Baud
genie.Begin(Serial);
```

To change the form, simply use

```
genie.WriteObject (GENIE_OBJ_FORM, FORM_NUMBER, 0);
```

where FORM_NUMBER is the form number set in the Workshop IDE.

To change the string, use

```
genie.WriteStr(STRING_NUMBER, buf);
```

where STRING_NUMBER is the ID for the string set in the Workshop and buf is a char[].

Capture User Event

The first thing to do is to put `genie.DoEvents();` in the `loop()` function. Then in the `setup()` function put `genie.AttachEventHandler(myGenieEventHandler);` where myGenieEventHandler is the function to be called every loop when a message is reported from LCD.

In myGenieEventHandler, define these:

```
genieFrame Event;
genie.DequeueEvent(&Event); // Remove this event from the queue
```

Next we want to check if the event is a report event from LCD, we don't usually need to handle this situation but I'll put here for completeness.

```
if (Event.reportObject.cmd != GENIE_REPORT_EVENT)
{
```

```
            // If this event is NOT a Reported Message
}
```

After it, we check which object triggered the event.

```
if (Event.reportObject.object == GENIE_OBJ_XXX) {
}
```

GENIE_OBJ_XXX could be an object that has output ability, such as a button. An example would be GENIE_OBJ_ANIBUTTON. If you want to take a look at the list of objects available, go to the Arduino Genie Library and find the .h and .cpp file.

Inside the previous if statement, we need to check which obj exactly fired the event. To check the object at index 0, use

```
if (Event.reportObject.index == 0) {
}
```

Great! That's all I had to use for the LCD. Take a look at App Notes and Data Sheets on 4D Systems' Website for more information.

Problems/Limitations

There are some limitations with this LCD and some problems that took me a while to solve.

- Do not use genie.WriteStr to write to a string object many times within a short period. It would cause the LCD to freeze and only thing I could do was to wait or to reboot the Arduino.
- The LCD stores is resources in a micro sd card which has to be manually taken off and plugged into a computer to download new resources when updating the LCD.
- When using a button object, or anything that reports an event, always make sure the value for OnChange in the Event tab is set to "Report message" so that events are sent to Arduino.
- When using a Anibutton, which is an animated button, 3 images must be set for it to work even if you don't want the button to change if user presses it. The first image is for normal state, the second (or more) image is for when pressed down, and last image is for when user releases the button and then the button animates back to the first image.
- There is absolutely no control over positioning things on the display and no control over writing an image object or manipulate a complex object such as a button during run time.
- The LCD has to be disconnected from Arduino when uploading sketch to the Arduino
- To fix the fore mentioned problem, Workshop has another environment which gives you entire control over the LCD.

RN-52 Audio Bluetooth Breakout

This bluetooth chip supports audio streaming without additional configuration. It also offers a variety of commands to query the state of bluetooth and set its properties. Sparkfun has a detailed hook up guide for this but I'll still state things here.

Serial Communication

On the guide it tells you that you could change the track, volume, and play/pause by pulling pins high or low on the bluetooth chip. It seems very pin-consuming on Arduino side. After I had gotten the chip and read over the entire data sheet, I find that Serial Communication can handle all the aforementioned things. Also as mentioned before, I had to use Software Serial to communicate with it, so I pulled down the baud rate to 9600 by ground the GPIO 7 pin. Through Set commands, which start with S, I customized the name, passcode, authentication methods, and many other properties of the chip. There are also some get commands to retrieve the state/firmware version of the bluetooth. Finally, there are action commands such as AT+ for next track, AT- for previous one, AV + - for volume, and so on.

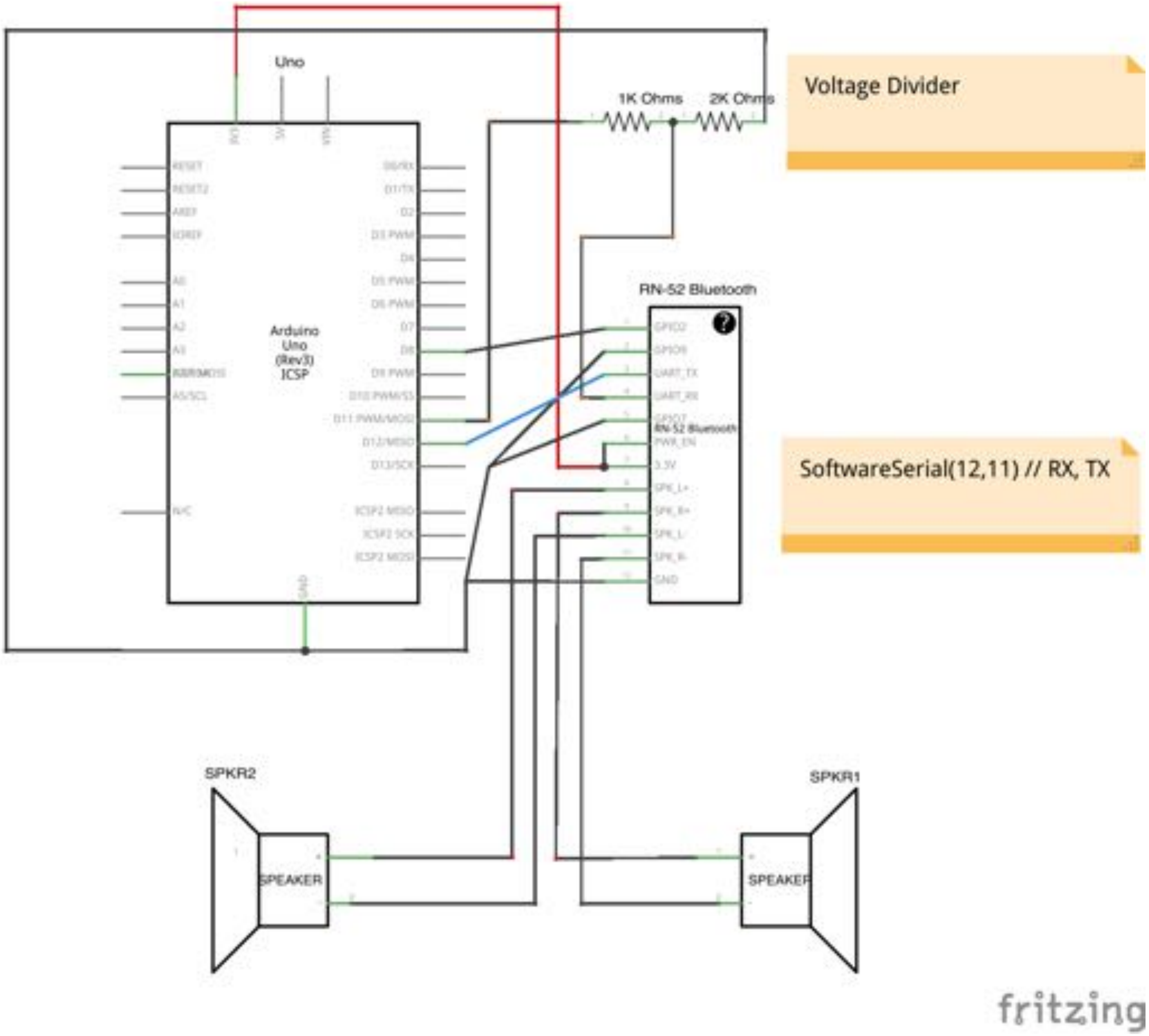
Pair w/ a Phone

It’s not an Audio Bluetooth Breakout that streams music only. It handles phone calls. With Serial Command, the bluetooth can accept an call, deny one, initialize one, and even has the ability to mute. This can be made to a bluetooth headset!

Pin Interrupt

This bluetooth chip pulls GPIO2 low for 100ms when the state of bluetooth changes. States include discoverable, connected, call active, audio streaming, etc. It is especially useful to detect an incoming phone call. But as I mentioned before, my Uno doesn’t have more interrupt pins, so I had to do something similar to pin interrupt to handle the situation.

Schematics



Voltage Divider

SoftwareSerial(12,11) // RX, TX

Problems/Limitations

- No amplifier on board, only able to drive headphones, not speakers
- AV+, AV- for volumn control not working when paired with iDevices, reason unknown.
- Modify ATCommands to work with SoftwareSerial.
- No command available if directly connected to the module.
- I recommand using Tera Term.
- State reporting not accurate. The immediately queries state after GPIO2 has been pulled low and then high is not the current state, but the previous one.

VS1053 MP3 Decoder

The same decoder chip we use in lab 6. No furthur details needed to be mentioned here. All information used are on wiki page. No significant problems found.

Transcend Micro SD with Adaptor

The same Micro SD used in the lab, nothing fancy. No significant problems found.

Speakers x2

The same Jameco speaker given out in the lab. They are not so bad.

Audio Amplifier

Stereo 2.8W Class D Audio Amplifier - I2C Control AGC - TPA2016. It does work.

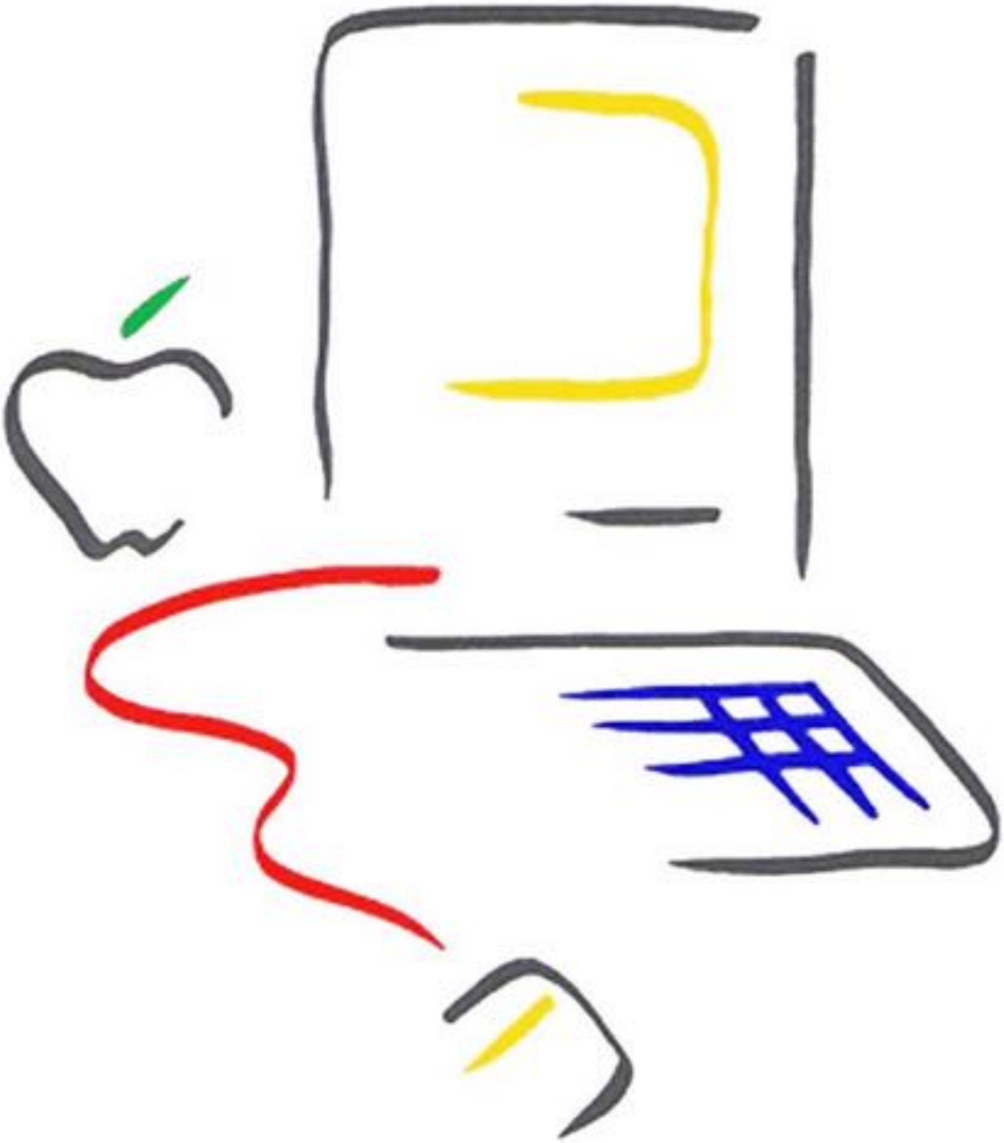
Real Time Control

No time to investigate it at all thanks to USPS's ultra-slow-2-day-express-shipping-that-took-a-week.

Notes

- SD and MP3 function has been taken out on the second last day due to either software or hardware issue with SPI communication with Arduino Uno. (Everything worked fine during labs with Arduino Micro)
- Incoming Call Handling has been taken out a week before the due date because of weird output from the Bluetooth Module when communicating directly with the Arduino.

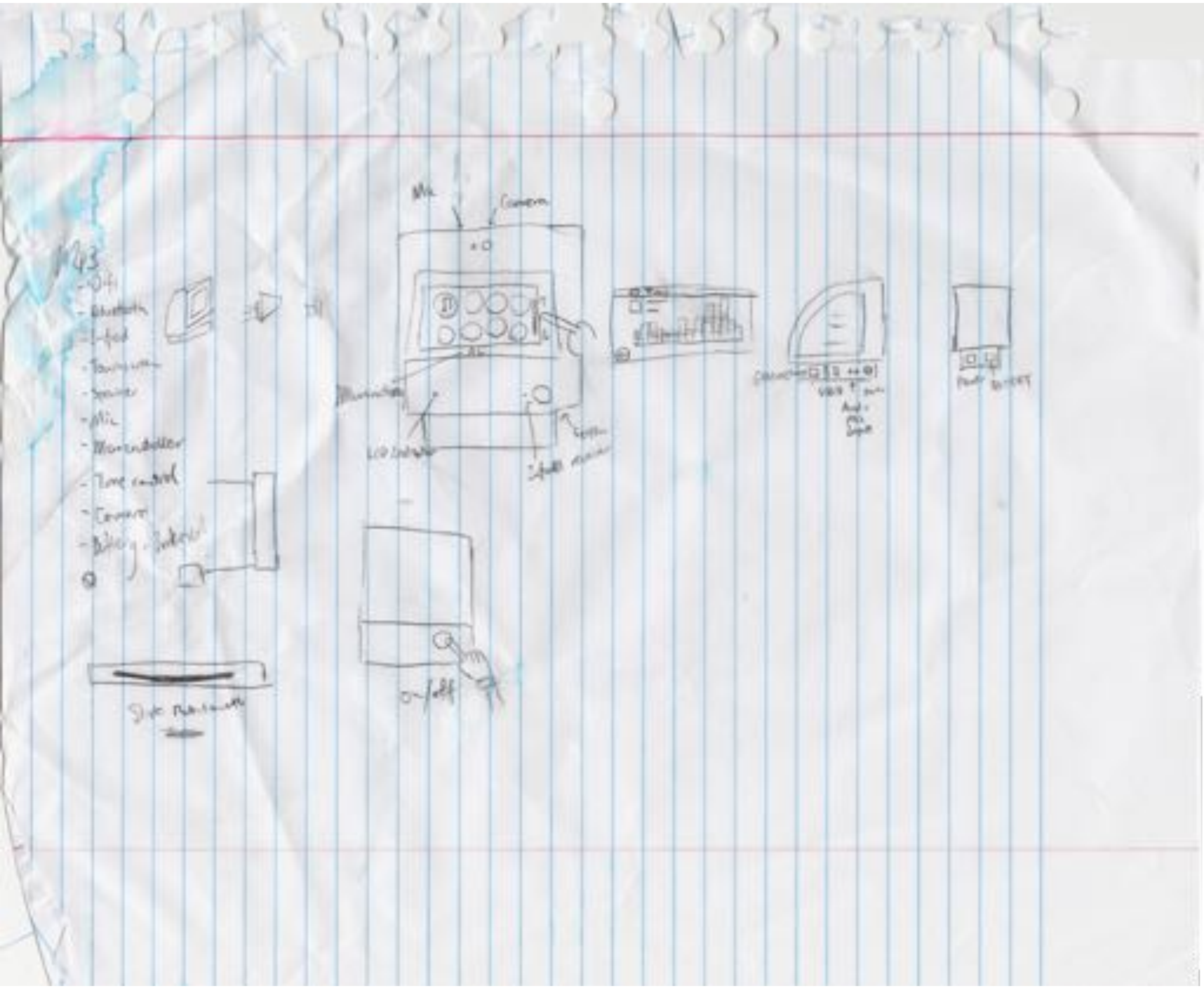
Design…

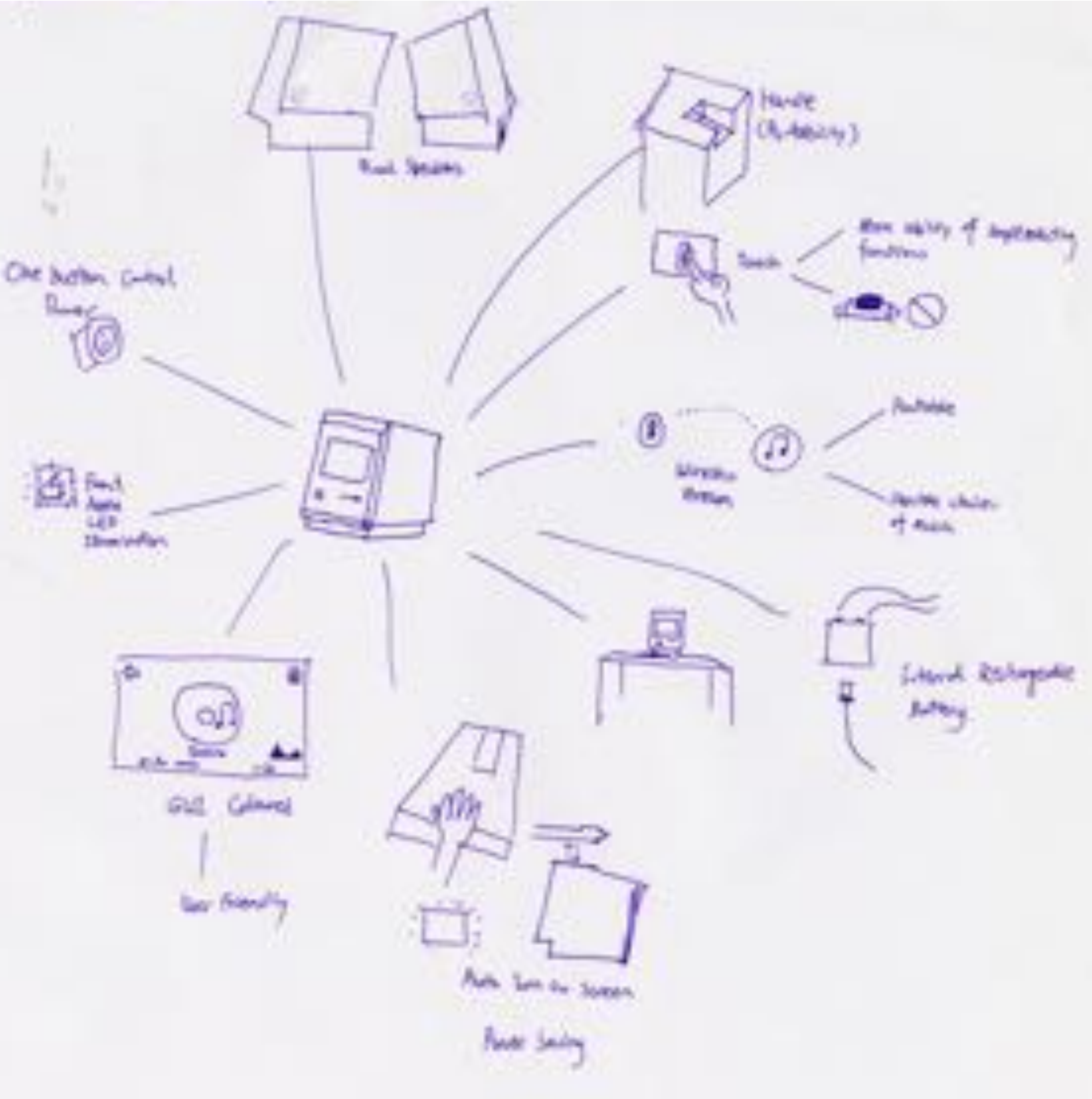


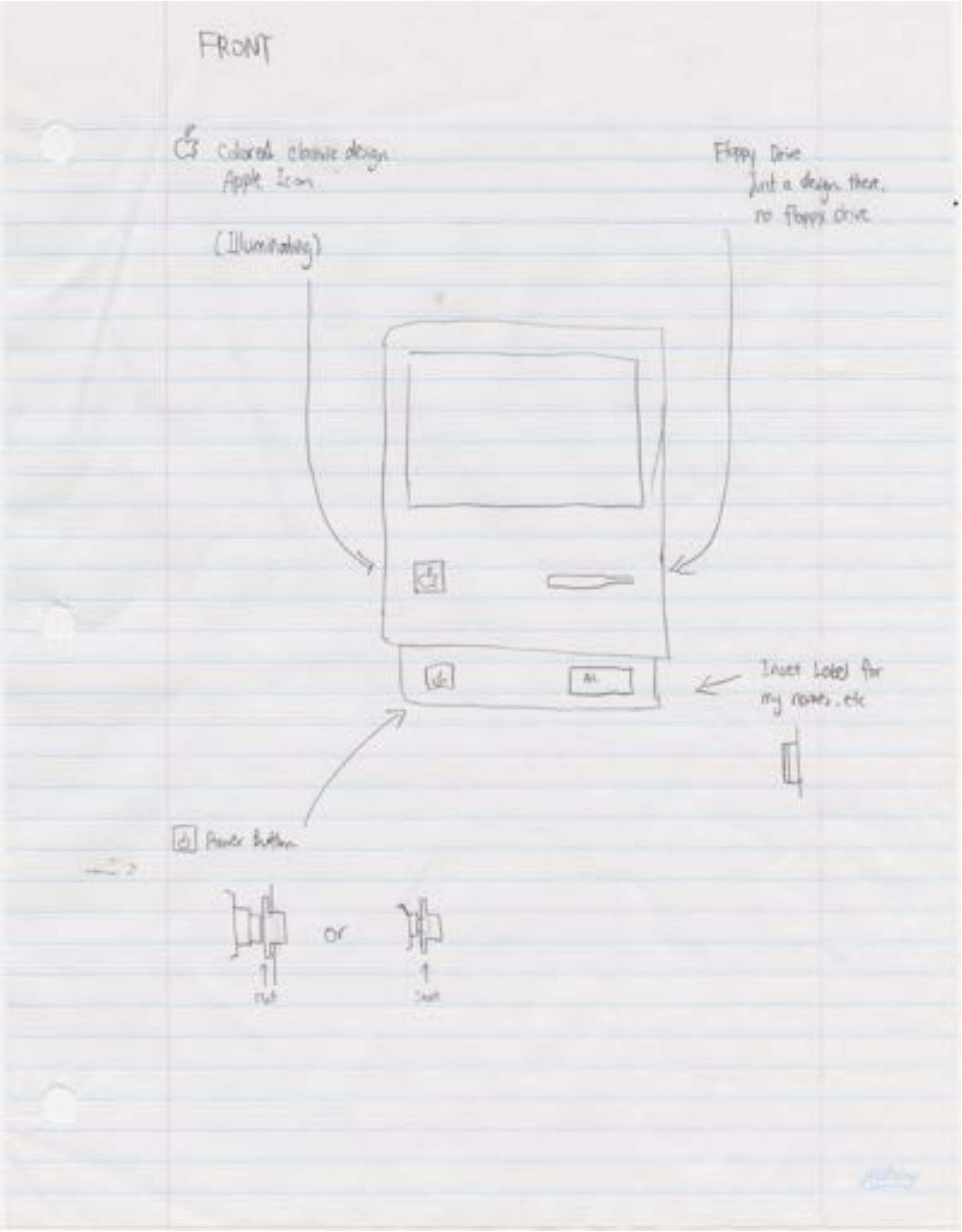
The New Macintosh 128K

Preliminary Sketch

The first set of sketches determined the general idea of the final product.

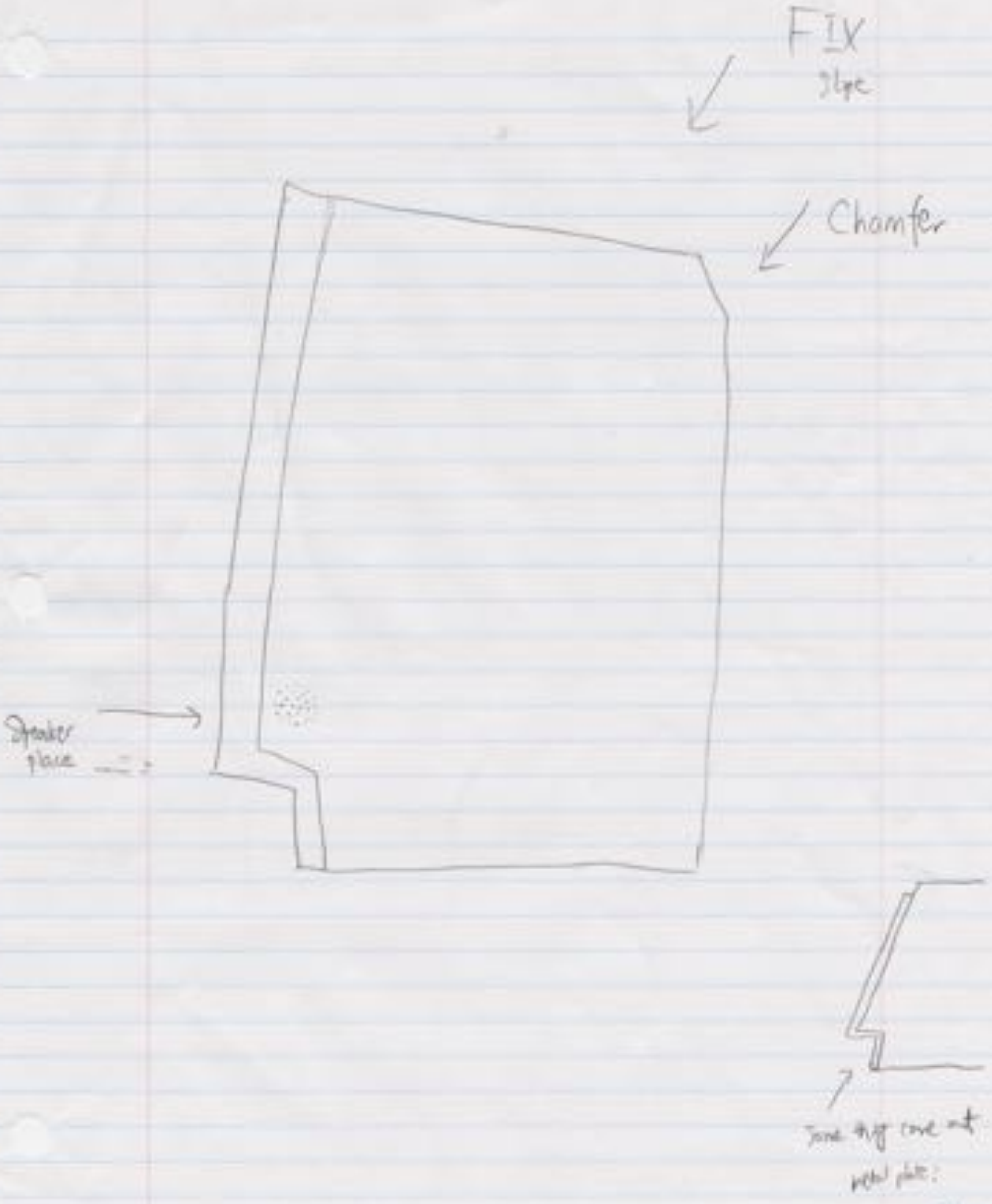



Front Speakers
Handle
(Portability)
One button control
Power
Touch
More ability of implementing
functions
Front
Apple
LED
Illumination
Wireless
Stream
Portable
Flexible choice
of music
External Rechargeable
Battery
GUI Coloured
User friendly
Auto turn on screen
Power Saving



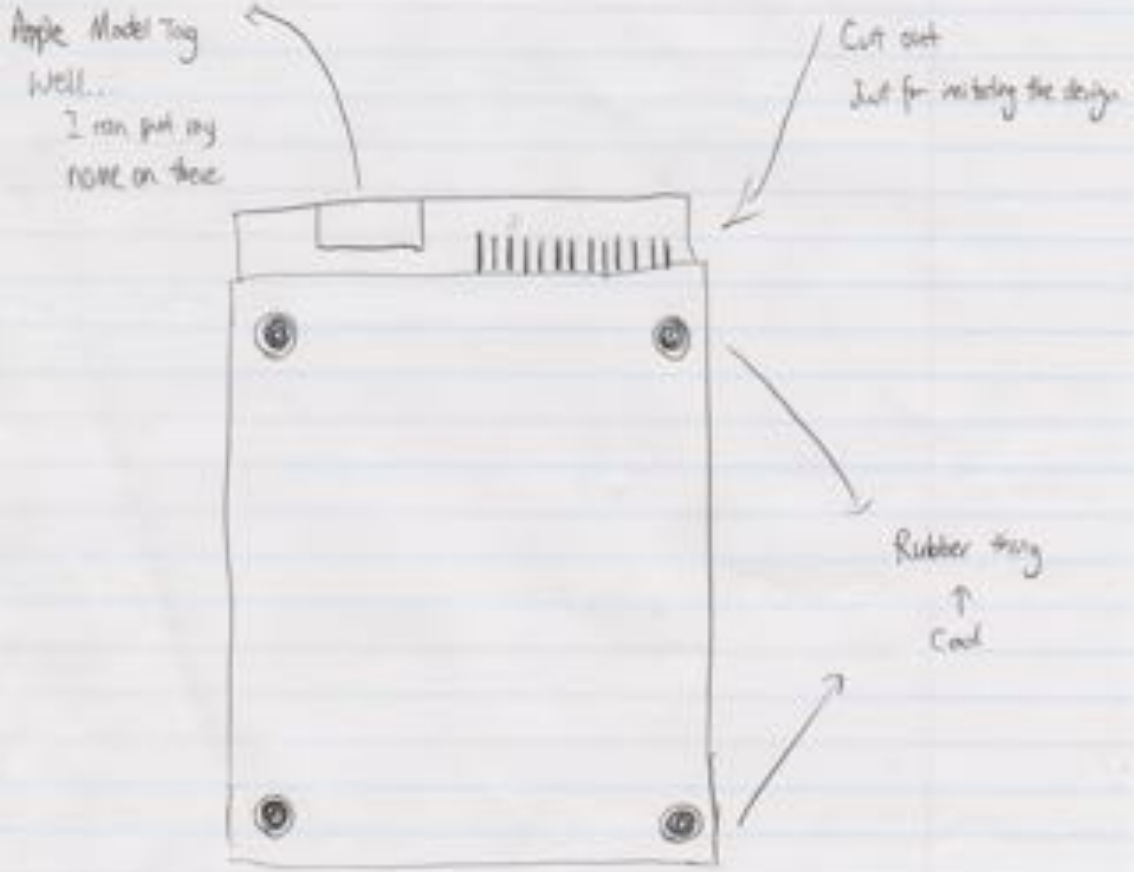
The New Macintosh 128K

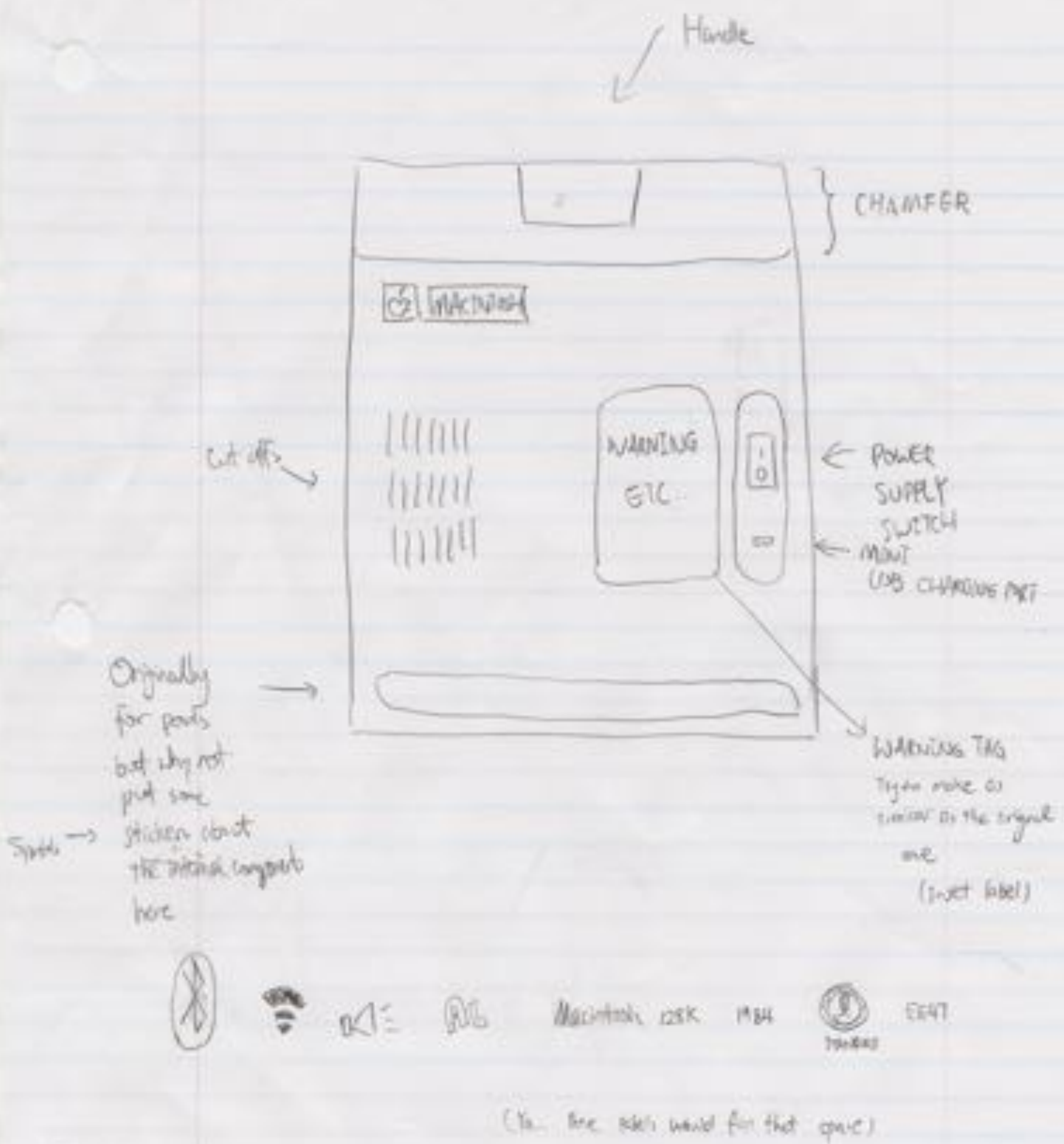
LEFT & RIGHT



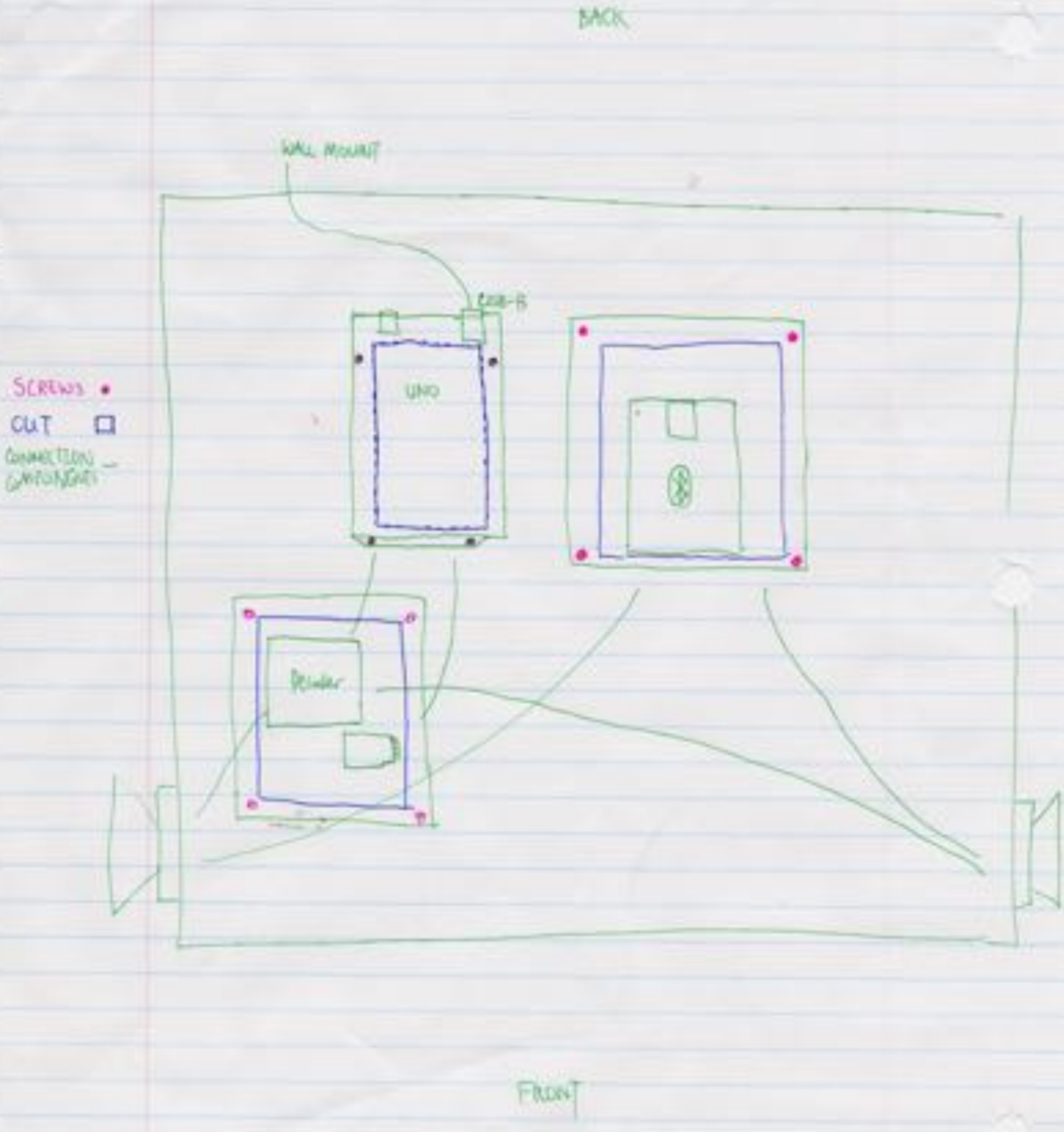
The New Macintosh 128K

BOTTOM


Apple Model Tag
Well...
I can put my
name on these
Cut out
Just for imitating the design
Rubber thing
↑
Cool



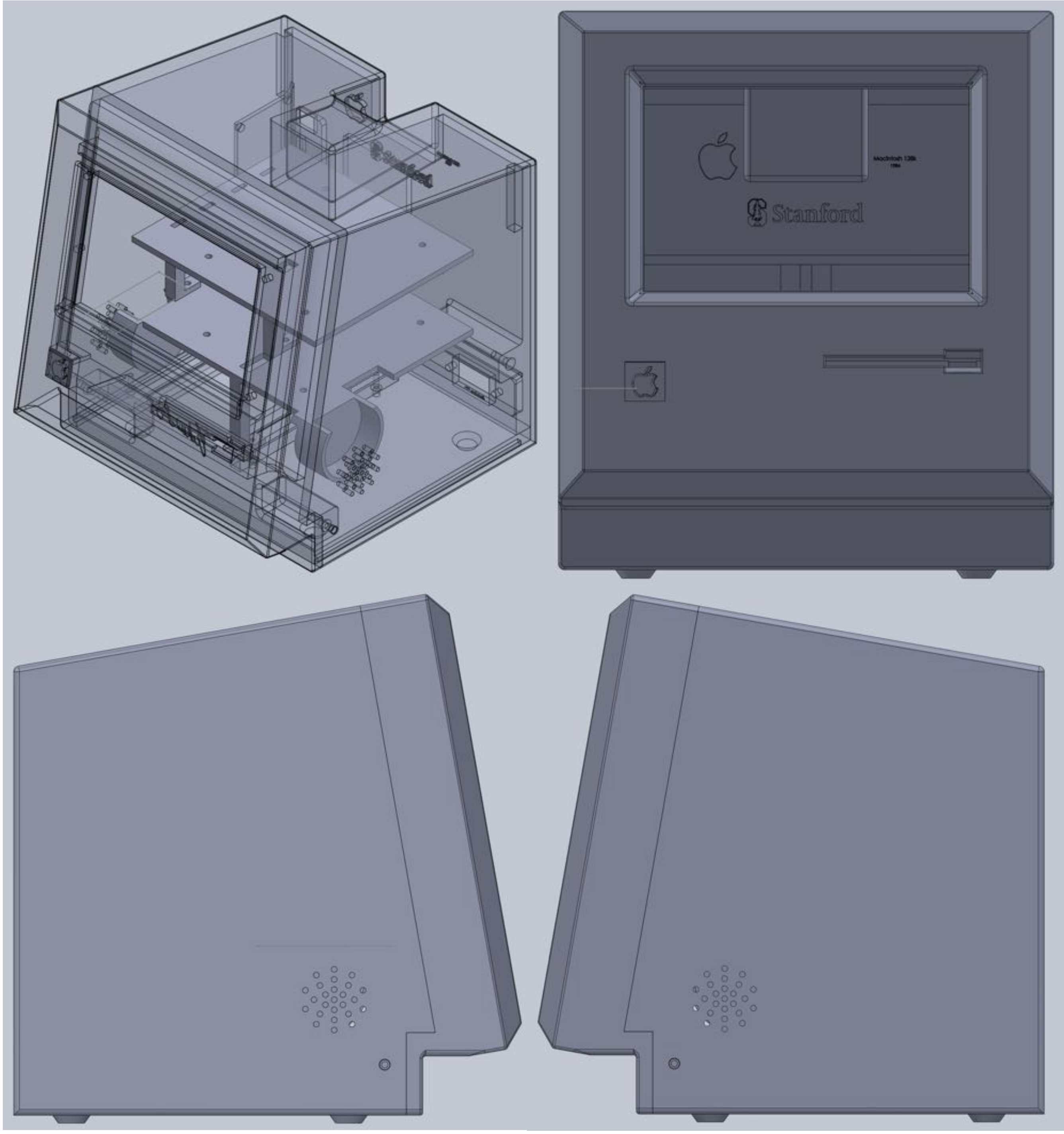
The New Macintosh 128K

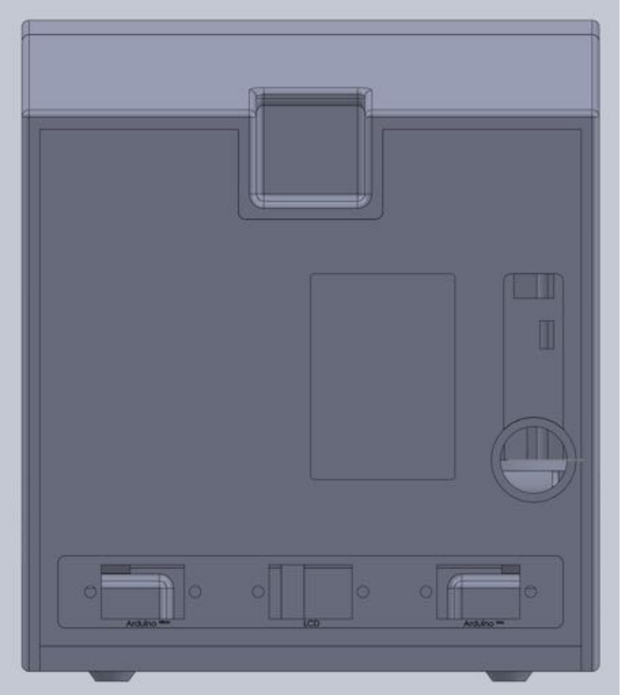


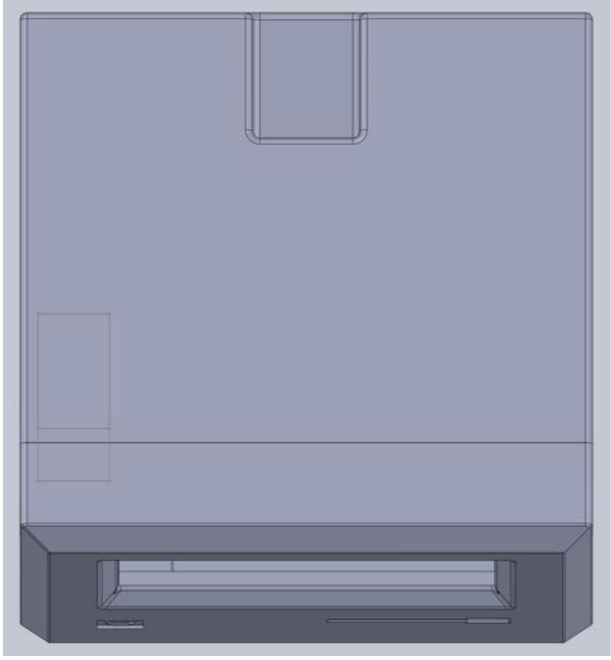
The New Macintosh 128K

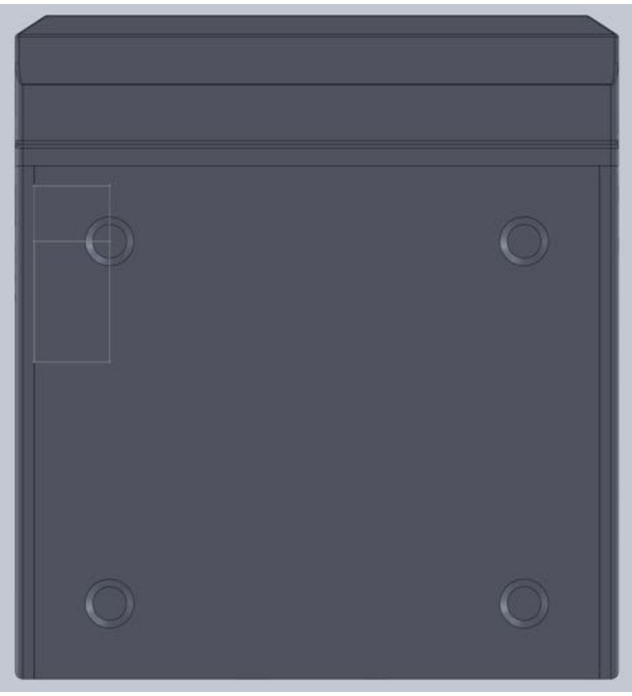
3D Modelling

The entire model is made from ground up by myself. All design is done in Solidworks. The use of calliper made my model very precise, especially where screws would go. Here are some views in Solidworks



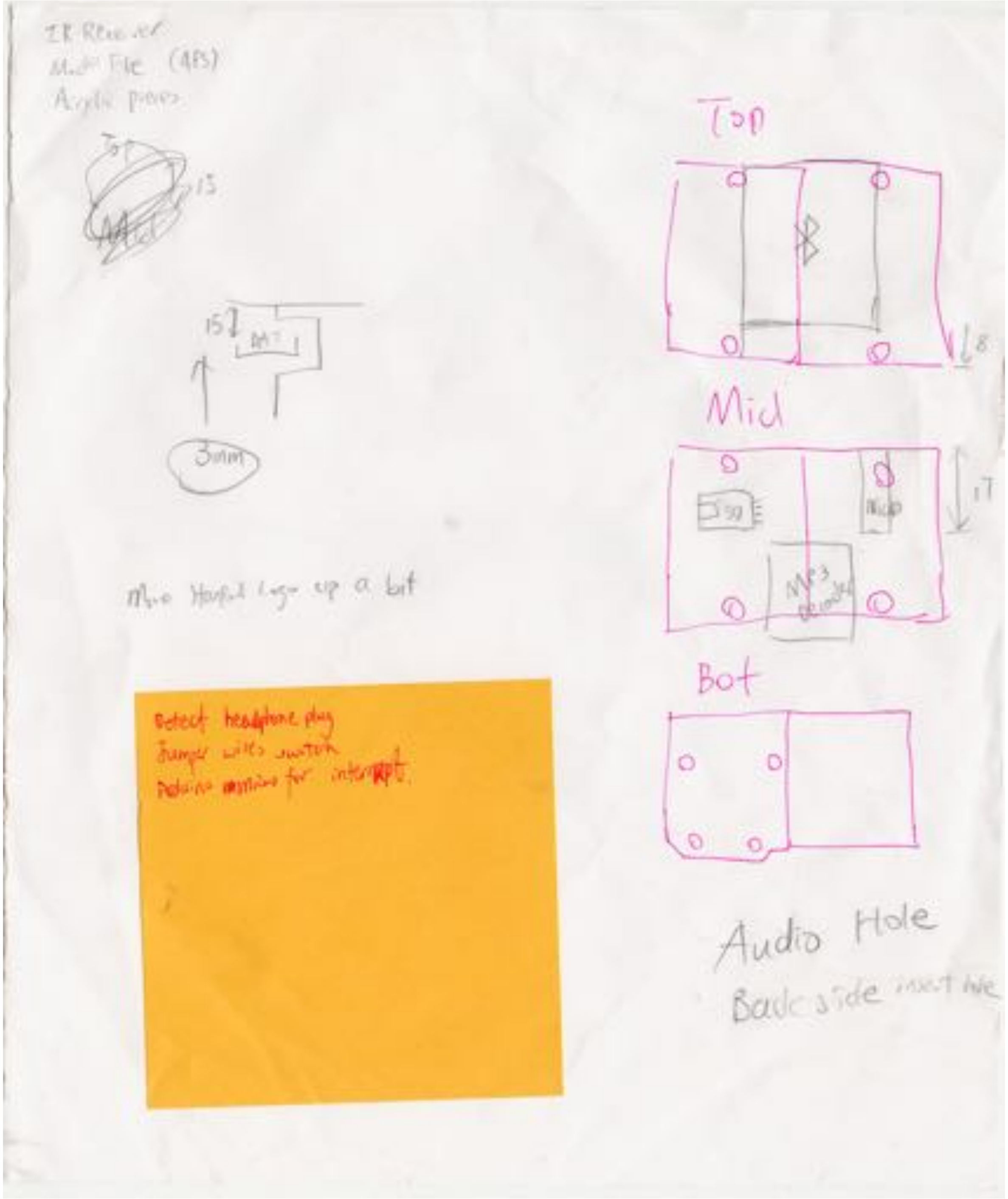

Arduino
LCD
Arduino





Editing

I went through many editing and most of changes I just wrote them on Post-It's and they were gone. But here is one that I wrote it down on real paper.



IR Receiver
Midi File (ABS)
Acrylic pieces

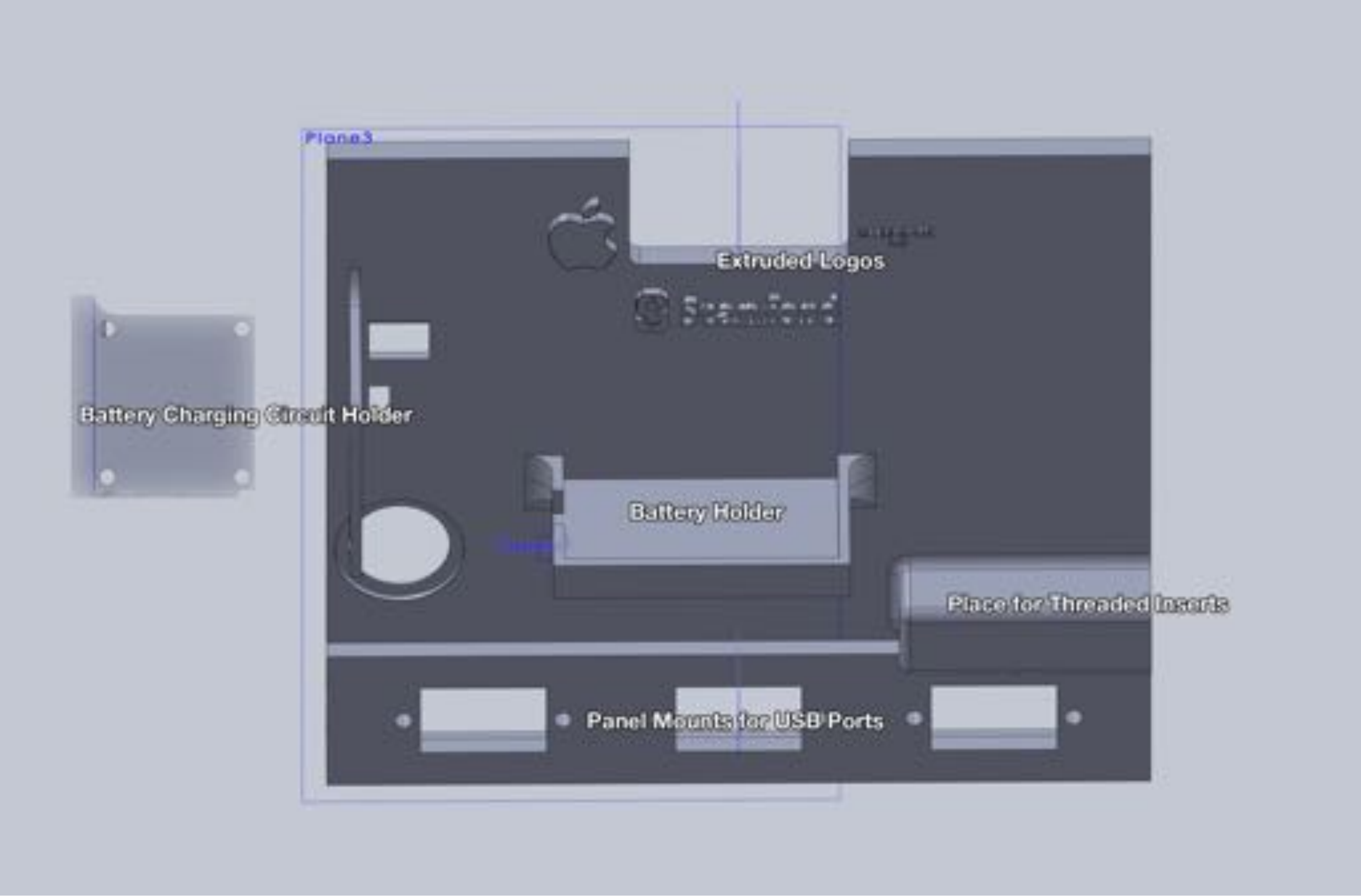
Move [illegible] logo up a bit

Detect headphone plug
Jumper wires switch
Arduino [illegible] for interrupt.

Audio Hole
Back side insert hole

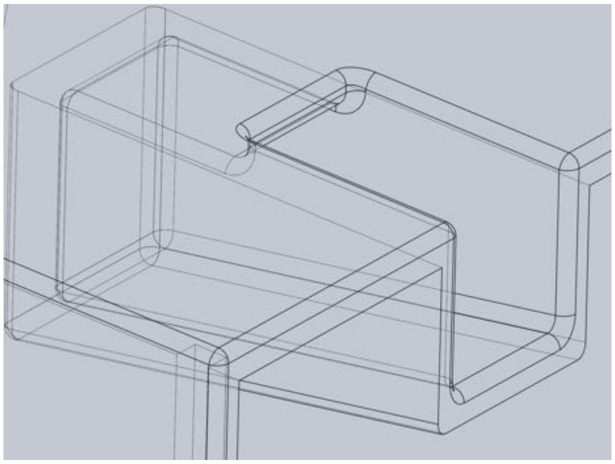
Details - Back

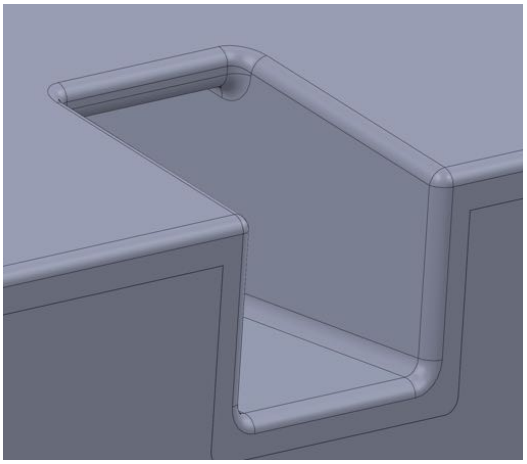
Battery Holder: It is a little bigger than the size of LiPo battery. It would be fixed in place by having a thing pressing on top of it. (You'll see it in the next few pictures)
Battery Charging Circuit: An extruded wall was created for holding the charging circuit.
Panel Mounts: Adafruit has these panel mounts for attaching a female USB port on the outside wall of your project. There are many types including micro, mini, and regular USB to panel mount.


Plane3
Extruded Logos
Battery Charging Circuit Holder
Battery Holder
Place for Threaded Inserts
Panel Mounts for USB Ports

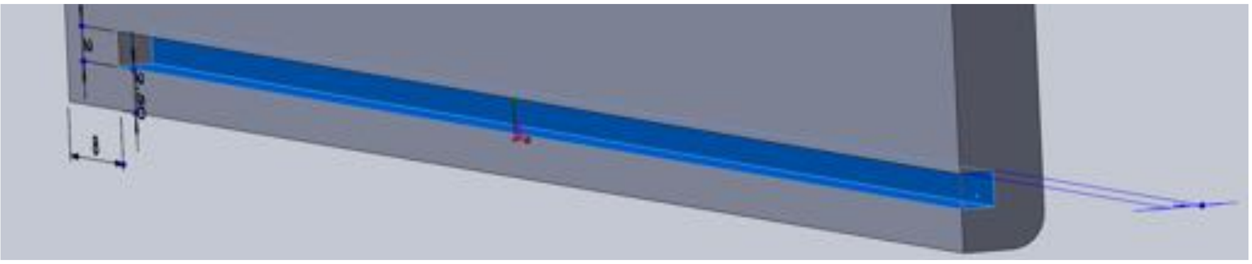
Details - Box

Handle: The handle on the top is made for portability.

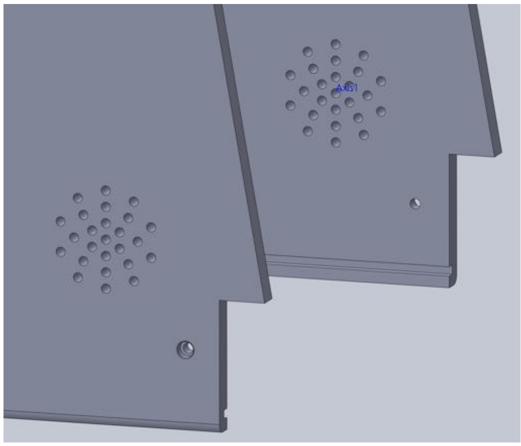




Grooves: They are located on the lower part of both sides of the box. It allows the bottom part to be attached without any screws.

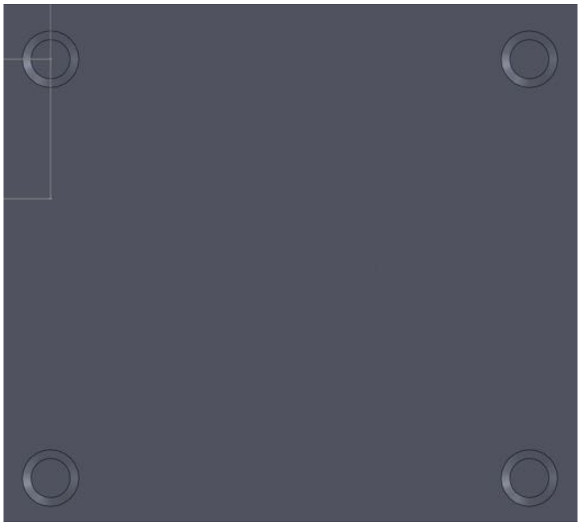


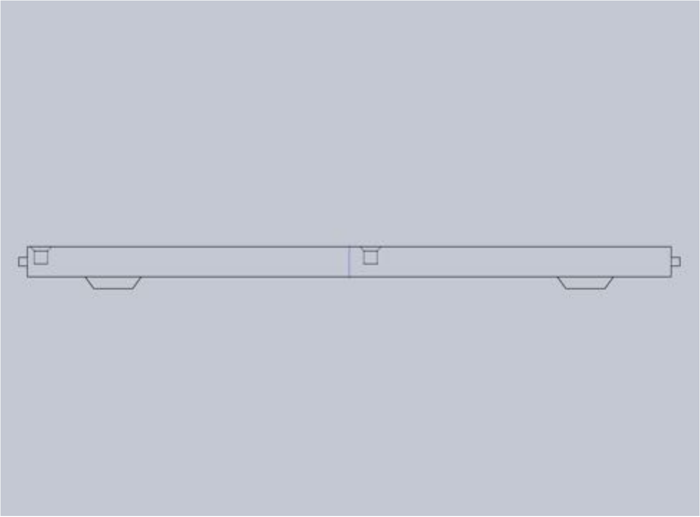
Symmetric Holes for Speakers: They are also located near the lower part of both sides of the box. It allows the sound from the speaker to come out of the box.



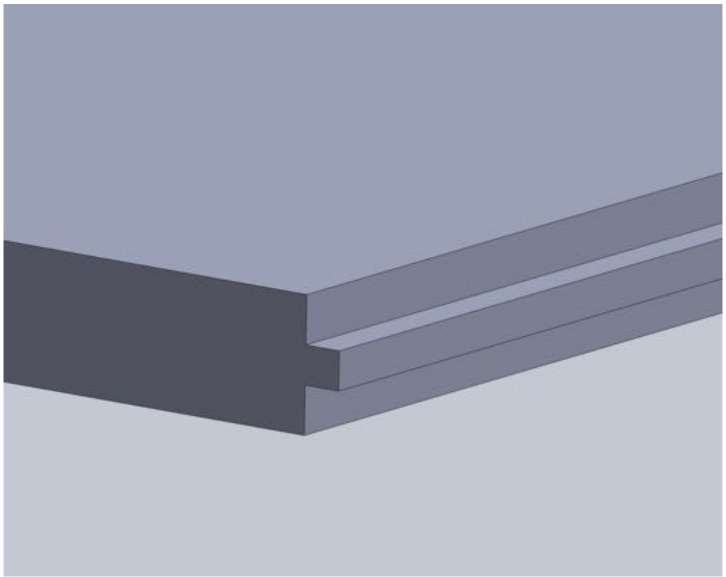
Detail - Bottom

Extruded Legs: There is no real purpose for this. The only factor that made me create them was that the original mac had them.





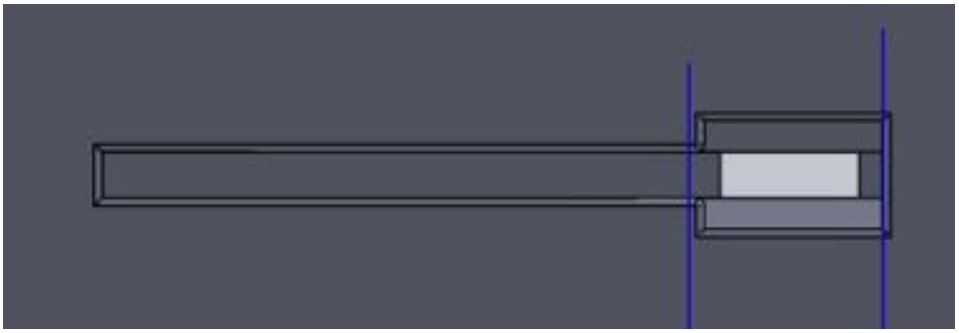
Extruded Ridges: The ridges that would fit exactly into the grooves on the box.



Details - Front

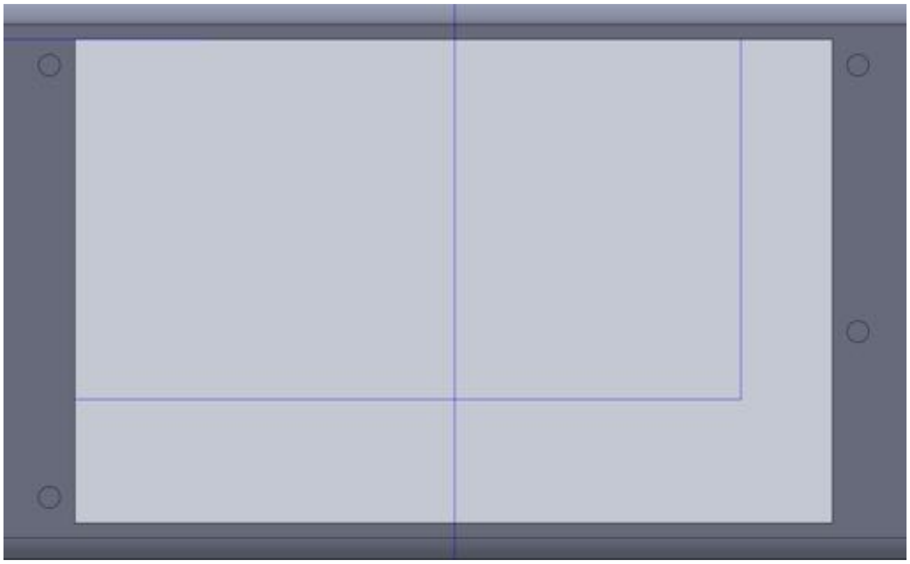
Apple Logo Cut: An open hole through that a semi transparent Apple Logo would be inserted and an LED would be placed behind it.

Floppy Disk Slot: A fake slot for Floppy Disk. The real purpose of making this was so that I could place an IR receiver in it.

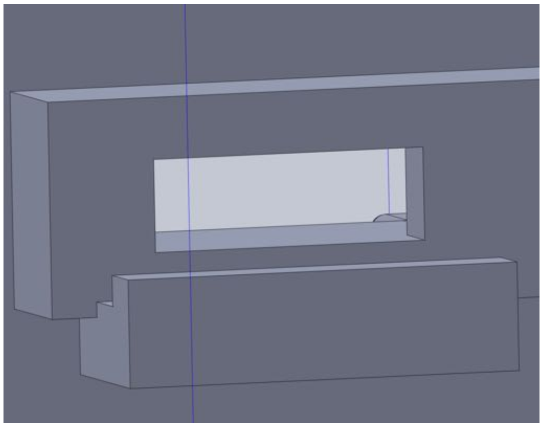


Yuhui Signed Project: My autograph is located inside the front part of Macintosh 128K

LCD Slot: An exact opening for the LCD to come out and 4 exact screw locations.

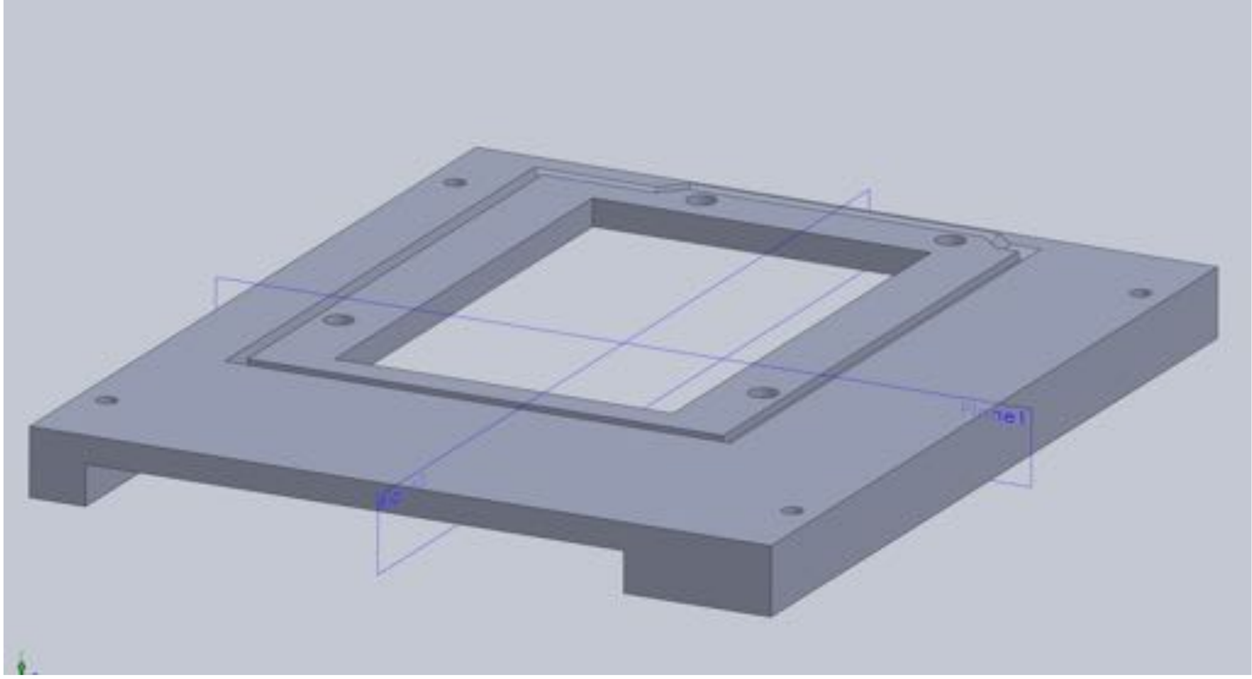


IR Receiver Holder: A support that would (partially) hold the IR Receiver in place.

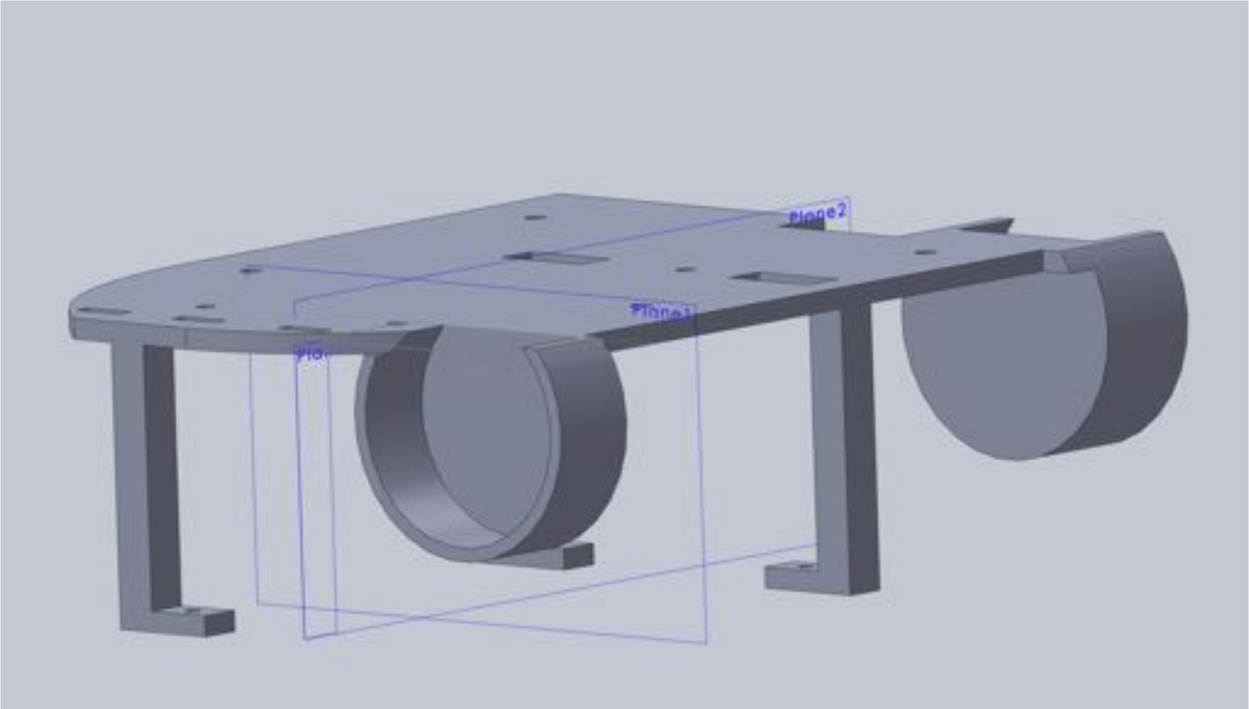


Details - H1 (First layer of the inner housing)

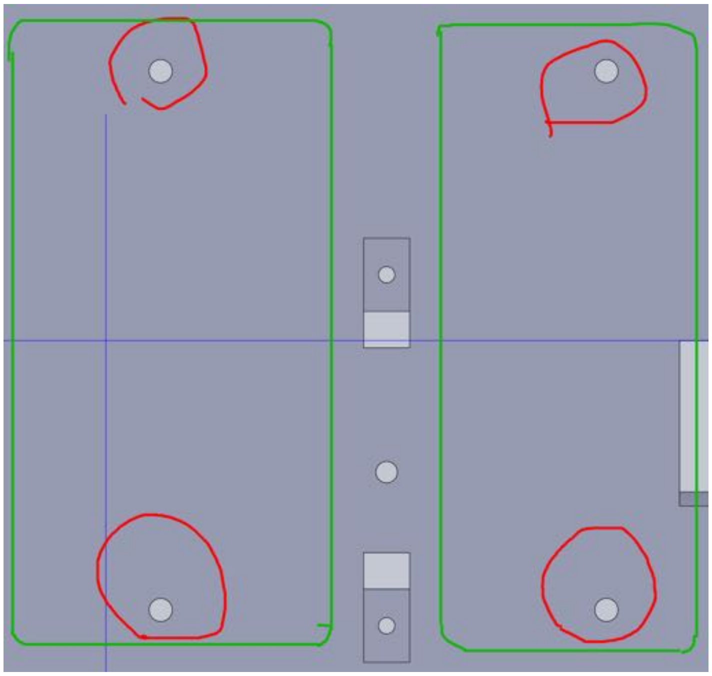
Arduino Uno Holder: The screw holes are exactly drewed according to the dimension data sheet.



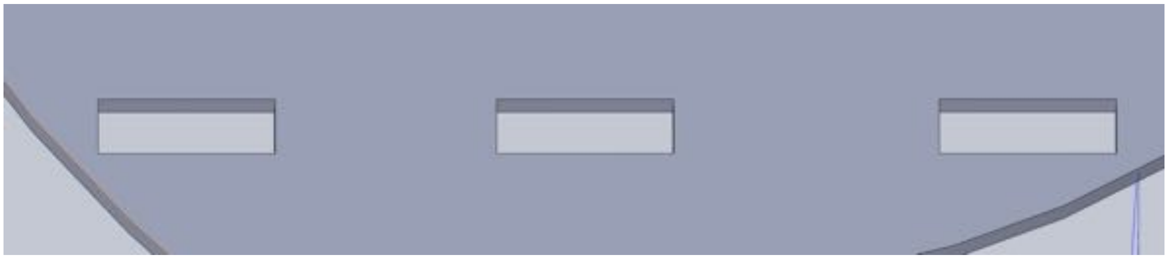
Details - H2 (Second layer of the inner housing)



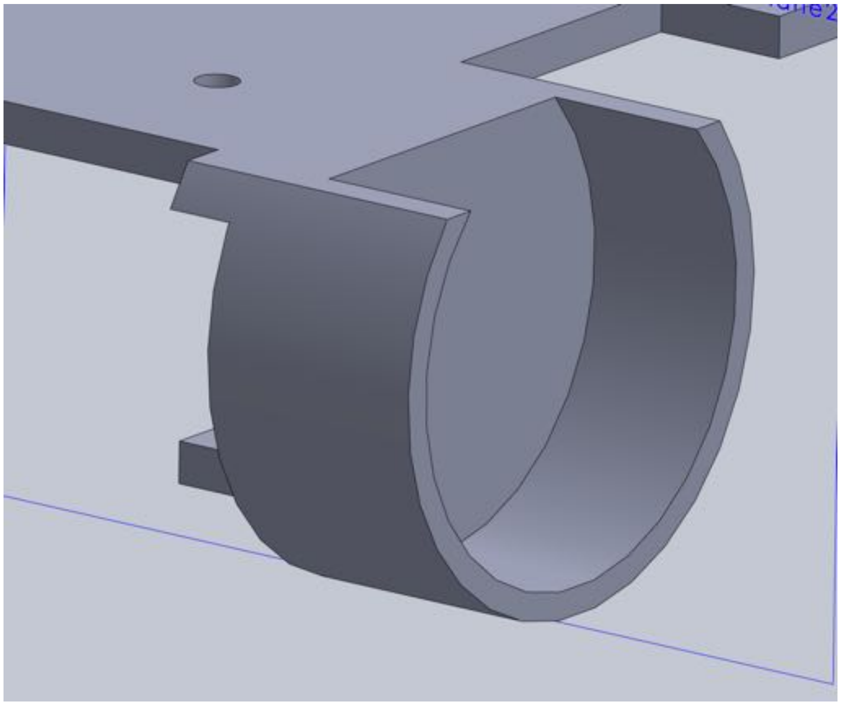
Adafruit Perfboard Holders: This platform would hold 2 Adafruit Perma-Proto 1/2 Size Perfboards. They will not be sticking together, but rather, having 2.54mm*4 distance between them. (Image not to scale)



Wire Organizers: Here are some holes that allow wires to go through. Each hole represent wires that would go to a single component.

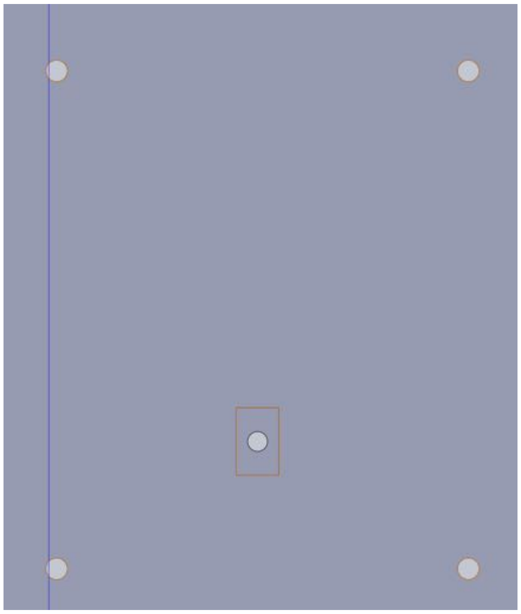


Speaker Holders: The cylinder's inner round face is larger than that of the end of speaker. Foam sheets would be used to fill the space.

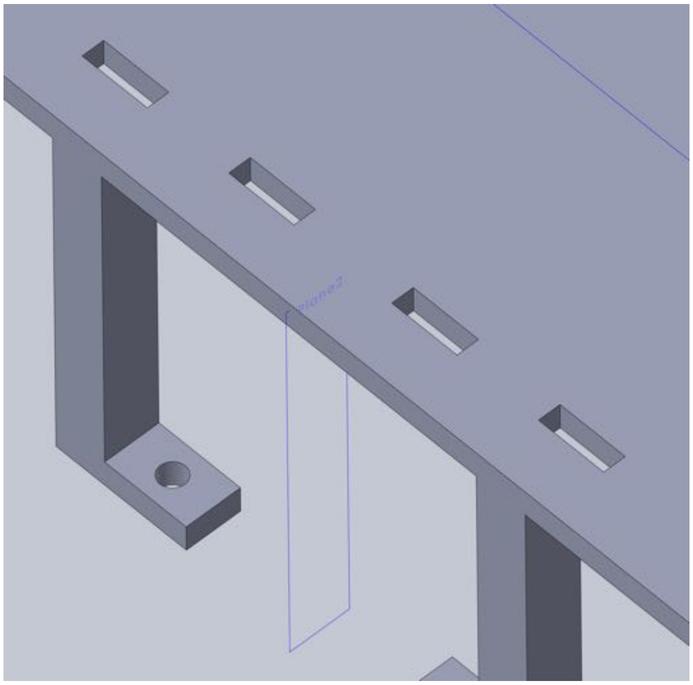


Details - H3 (Third layer of the inner housing)

Adafruit Perfboard Holders: This platform would hold 2 Adafruit Perma-Proto 1/2 Size Perfboards. They will not be sticking together, but rather, having 2.54mm*4 distance between them. (Image not to scale)



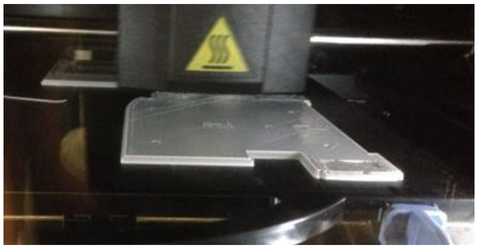
Wire Organizers: Here are some holes that allow wires to go through. Each hole represent wires that would go to a single component.



3D Printing in Progress…

Bottom

H1+H2



Assembly…

Threaded Inserts

For many places in the New Mac 128K, it is impossible to put the parts together using just screws and nuts. Using threaded inserts allows me to use only a screw on the outside when I need to put two parts together.

McMaster Carr has a variety of them. Measure all your holes before you purchase. Double check the insert length, drill size, etc properties on the website. Don't buy the press fit if possible, because base on my experience they are hard to insert into place and harder to keep them in there. Try the heat one. They ship from Los Angeles and with UPS Ground it would arrive by one day or two.

Nuts

Excellent choice to secure perfboard onto the structure I made for the inner part of the New Mac 128K. Room 36 sells many of them but not the metric ones. If you insist on using a metric screw, you need to purchase the nuts on your own. However, there is always a way to *(kinda)* work around the limitation. For M3 screws, you can *(kinda)* use a 4-40 nut with rubber on one side, and screw it in from the other side. Of course you won't get far, so put *(a lot of)* washers in between to secrew the things.

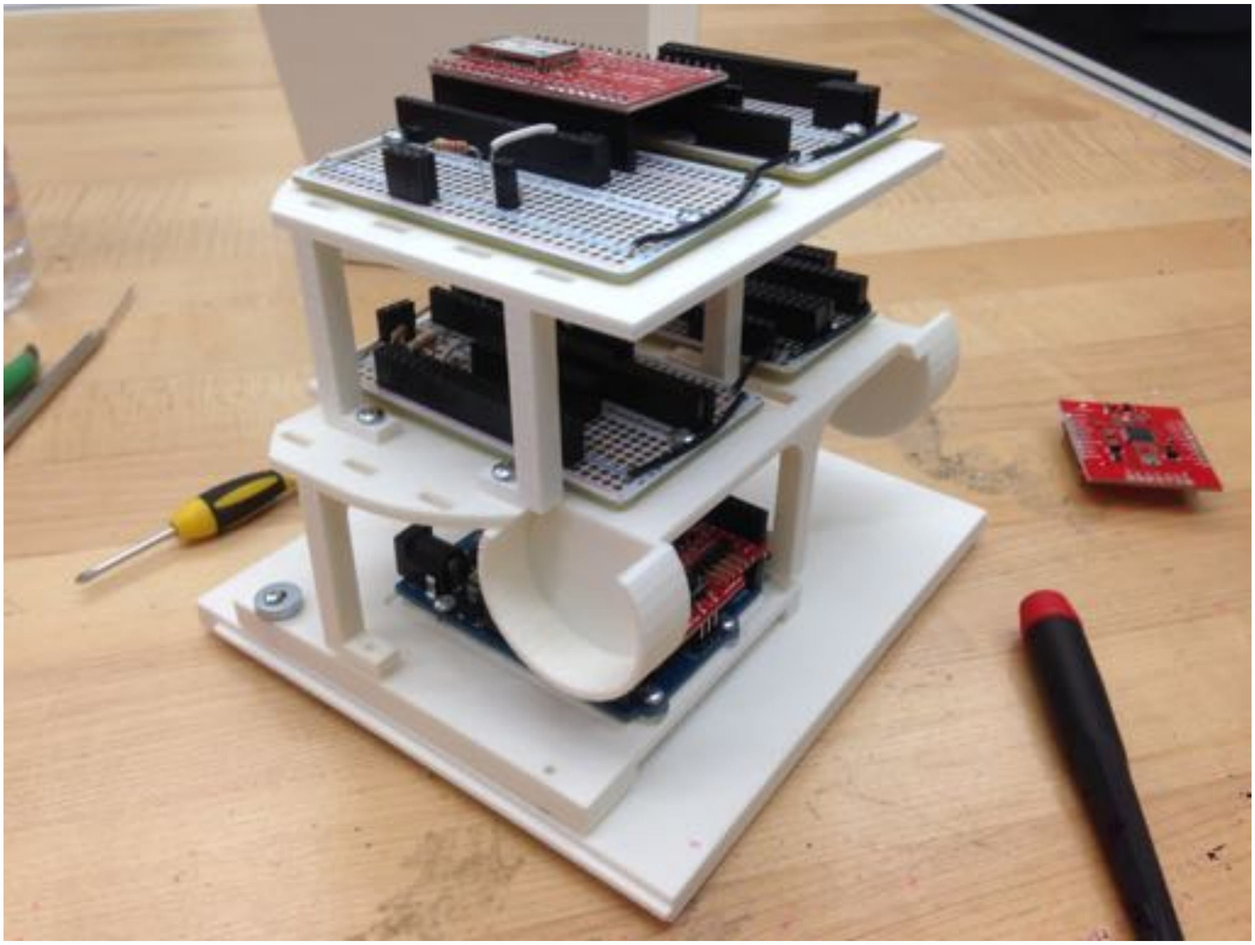
Washers

As mentioned above, they sometimes (in this case) helps to fill the gap between nuts and screws or other regions of the screws if needed.

Foam Sheets

Excellent choice for filling the gap between the speakers to absorb some vibrations.

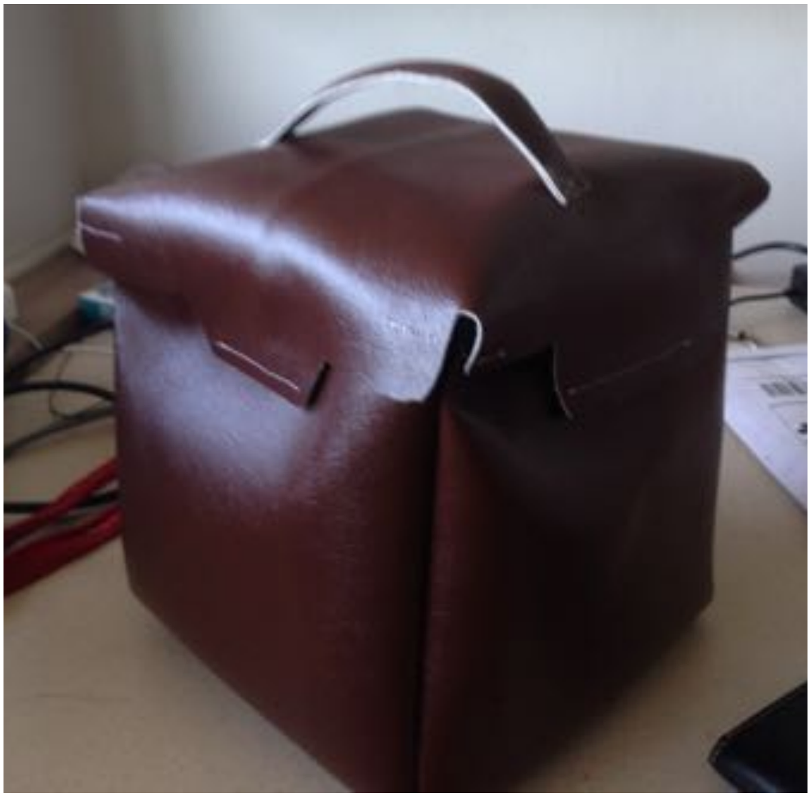
Preliminary Assembly

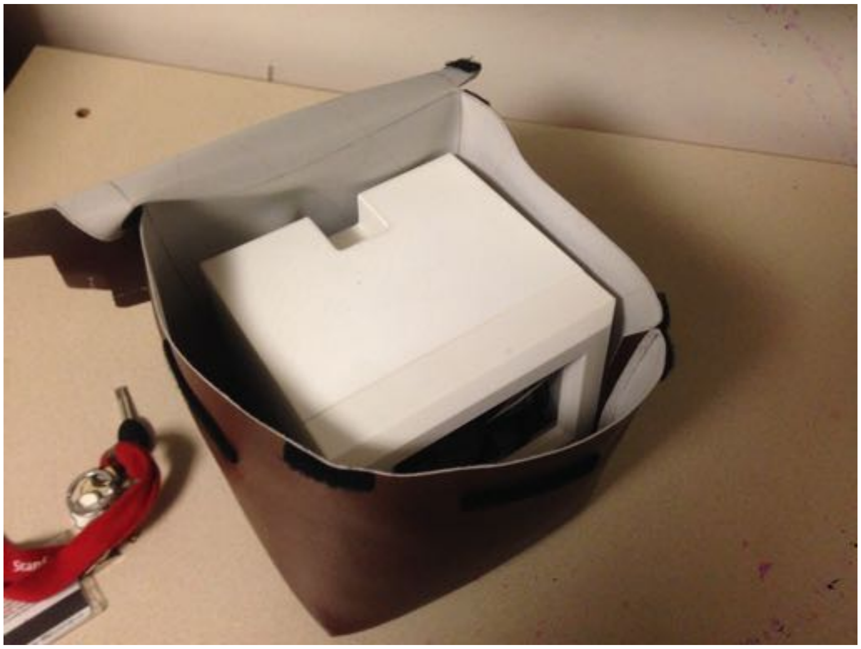


There are imperfections and things uncounted of that can only be seen after the parts are printed out. Examples include the height of Arduino which would include shield+wire+itself, height of the perfboard which would include female header+component+itself. I was forced to give up some space because other things need spaces as the parts are assembled.

The Bag

To recreate the feel of the keynote during which Steve presented the Mac 128K, I spent an entire afternoon making this bag so that during the presentation it would come out of a small bag.





Almost There - 24 Hrs Countdown till Presentation

Reality

The New Macintosh 128K

General

After working in labs 10 hours a day for almost three weeks, I was able to finish this project. However, because I spent a lot of time working on the design aspect of it, I had to cut off some functions that I wanted to accomplish such as Internal MP3 Playing from SD Card, Pictures Viewing, Finder functions.

This project is essentially a Bluetooth music player. It could pair with a device (such as an iPhone) and control the music playback wirelessly. Also it has the ability to make phone calls right on the player but unfortunately I did not have a microphone connected to it so all it can do is to let you listen to what the other person is talking to you.

GUI

The GUI of this music player simulates the Mac Plus interface. It is not running an OS but rather just a simulation of it by displaying series of images on LCD. I did not have to draw them manually because the LCD has its own IDE so all I needed to do is to drag and drop objects to it. The LCD has its own processor and SD storage so it would not add extra work load to Arduino.

Problems

Tradeoffs with Designs and Functionality

As mentioned above, not all aspects could be taken care of within the two months time. I spent a great amount of time working on the case itself (making models, 3D printing, etc), so at the end I did not have time to finish the functionality I intended to include. A valuable lesson is learned.

3D Printing with Dimension… Not so Easy

Dimension uses Break-Away support, which in essence is simply another kind of plastic material. For supports that are big, this material is used in very very low density, making it super easy to break it off the real model. However, for the small holes or grooves, the material is used very densely, It would take a long time to break it away. I would probably explore other printing services online to compare pros and cons in the future.

It is Impossible to Make Things Right at First Try

I thought that the models I created in Solidworks would work perfectly with the components I want to put in, for example, space. I did not counted on the extra height created by the female header so at the end I had to give up a region on the prefboard that is blocked by the battery holder.

Space Issue

I initially thought that I had a lot of spaces in the project that I would fit everything in including 3 usb cables. However, as i started to put everything in, my life became more and more miserable because to attach or detach a wire that is not on the top layer, I needed to use some tools to pull them out or plug them back in, which was not convenient at all either in debugging, or wiring.

Moving Forward

First I would like to finish all the functions that I wanted to put into this project including Finder, Internal Music Playing, Picture Viewer, and IR control (TV Remote). It would make this project much more interesting that just a simple Bluetooth music player.

Next, after looking through everyone else's projects in the demo, I would like to explore how some amazing projects are created, including the Air Drum, Mouse thing, LeapMotion with Monitor Output, and the one that uses Raspberry Pi.

After that, I'll probably be a college student by then so I'll continue to explore EE courses offered in the school.

Yuhui Li