**CAPACITIVE KEYBOARD II DOCUMENTATION**

Contents

[Parts List 3](#_Toc17033772)

[Connections 3](#_Toc17033773)

[Suggested Wiring Approach 4](#_Toc17033774)

[Teensy 3.2 and Audio Board 4](#_Toc17033775)

[MPR121 Board 4](#_Toc17033776)

[MPR 121 Wires to Conductive Material: 4](#_Toc17033777)

[MPR 121 Board: 5](#_Toc17033778)

[Mic 5](#_Toc17033779)

[Battery Pack 6](#_Toc17033780)

[Connecting MPR121, Teensy 3.2, Buttons and Audio Board 7](#_Toc17033781)

[Box Construction 8](#_Toc17033782)

[Code 11](#_Toc17033783)

[General Set Up: 11](#_Toc17033784)

[Main Variables 11](#_Toc17033785)

[Setup Function 11](#_Toc17033786)

[createChord Function 11](#_Toc17033787)

[playChord function 12](#_Toc17033788)

[Recording Functions 12](#_Toc17033789)

[loop() 12](#_Toc17033790)

[Audio File Sources: 12](#_Toc17033791)

[Adding New Keyboards 13](#_Toc17033792)

[User Manual 17](#_Toc17033793)

[Materials/Supplies 17](#_Toc17033794)

[Necessary: 17](#_Toc17033795)

[Fun Extras: 17](#_Toc17033796)

[Set Up 17](#_Toc17033797)

[During Use and Trouble Shooting 17](#_Toc17033798)

[If a fruit stops working/playing a note when tapped: 17](#_Toc17033799)

[Other Issues: 17](#_Toc17033800)

[Clean Up 18](#_Toc17033801)

[Facilitation Fun! 18](#_Toc17033802)

# Parts List

* Teensy 3.2 Board
* Teensy Audio Board
* 2 Push Buttons
* Elect Mic
* Rocker Switch
* Battery Pack for 3 AA Batteries
* MircoUSB cable (other end does not matter, will be connected to rocker switch for power)
* 12 pieces of 22-gauge wire (length is decided based on how much distance between MPR121 board and conductive materials, suggested around 70cm)
* 2.54mm wire to terminal blocks (amount depends on how wiring is decided)
* 12 tin crimps
* 2 4.7k resistors
* #24 0.21 mm^2 ferules for terminal blocks
* Heat shrink
* Breakout board
* Proto-board
* Audio Extension Cable
* Nylon Female Quick Disconnect Flag Crimps
* If building box:
  + Hammond NFG 1550G box
  + M3 Standoffs and screws for the Audio Board
  + M2 Standoffs and screws for the MPR 121 Board
  + Thumbscrews (4mm)
  + Adhesive Rubber Feet
  + Grommet

# Connections

List of wiring connections for Capacitive Keyboard. When constructing wiring it is important that any connecting wires to the Teensy Audio Board are kept short, otherwise audio becomes unclear.

|  |  |
| --- | --- |
| From | To |
| Teensy Audio Board Mic | Elect Mic Output |
| Teensy Audio Board Ground | Elect Mic Ground |
| MPR121 VIN | Teensy 3.2 3.3V |
| MPR121 GND | Teensy 3.2 GND |
| MPR121 SCL | Teensy 3.2 19 |
| MPR121 SDA | Teensy 3.2 18 |
| MPR121 IRQ | Teensy 3.2 17 |
| Button A (Switch Keyboards) | Teensy 3.2 16 and GND |
| Button B (Record) | Teensy 3.2 15 and GND |

Teensy Audio Board and Teensy 3.2 need pins 9,11,13,18,19,22,23,7,10,12,14 connected together

# Suggested Wiring Approach

## Teensy 3.2 and Audio Board

The simplest way to connect the Teensy 3.2 and the Audio Board is by soldering headers onto the Teensy 3.2 and sockets onto the Audio Board.

## MPR121 Board

### MPR 121 Wires to Conductive Material:

* Cut 12 pieces of 22 gauge wire around 70 cm long.
* On one end of each wire add crimps that can be inserted into the terminal blocks
* On the other end of each wire add tin crimps. These tin crimps can easily be inserted into conductive materials
* Lastly heat shrink each of the ends with tin crimps. This gives a cleaner look and also prevents any scratching from the tin crimps



Other end of wire with tin crimp

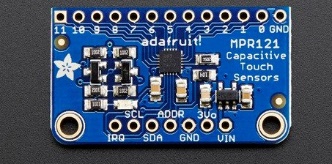
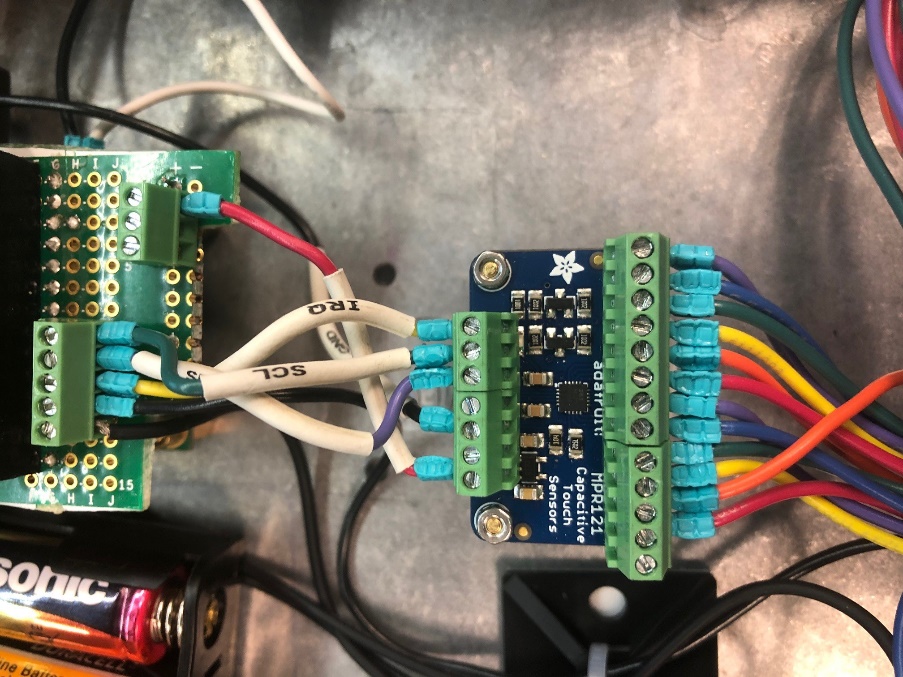
Tin crimped side of wire after having heat shrink added

End of wire that will go into terminal block



### MPR 121 Board:

* Solder Terminal Blocks onto the MPR121 board
* Cut 4 lengths of 22-gauge wire about 4 cm long each and 1 length of 22-gauge wire about 10 cm long
  + The 4 lengths of wire are for VIN, SCL, SDA, and IRQ
  + The longer 10cm length is for GND
* At the both ends of wires crimp on ferules for terminal blocks

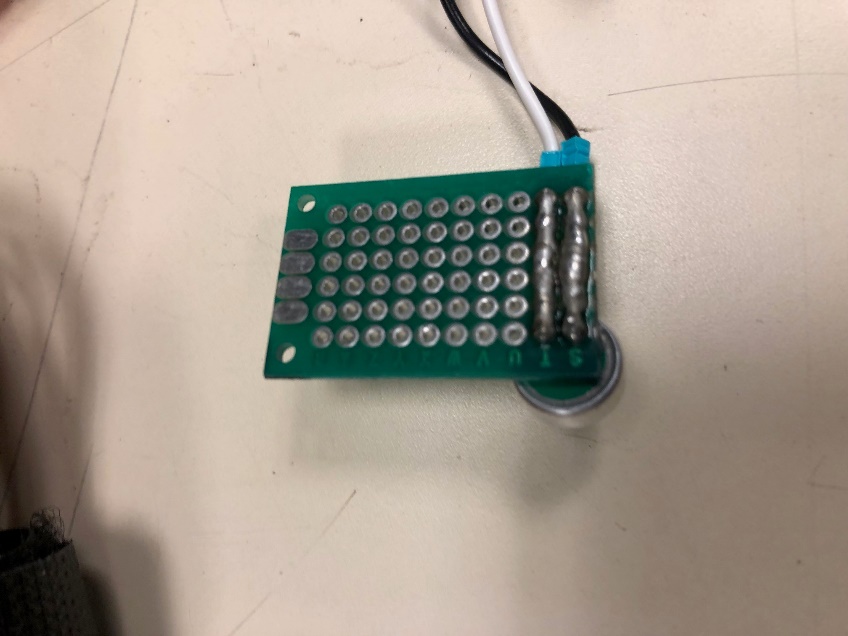
****

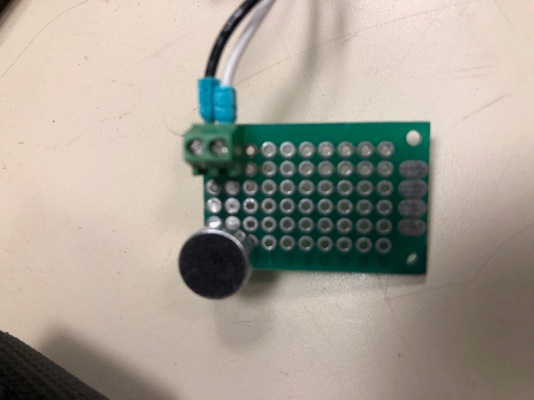
MPR 121 Board without Wires

Completed MPR 121 Wiring

## Mic

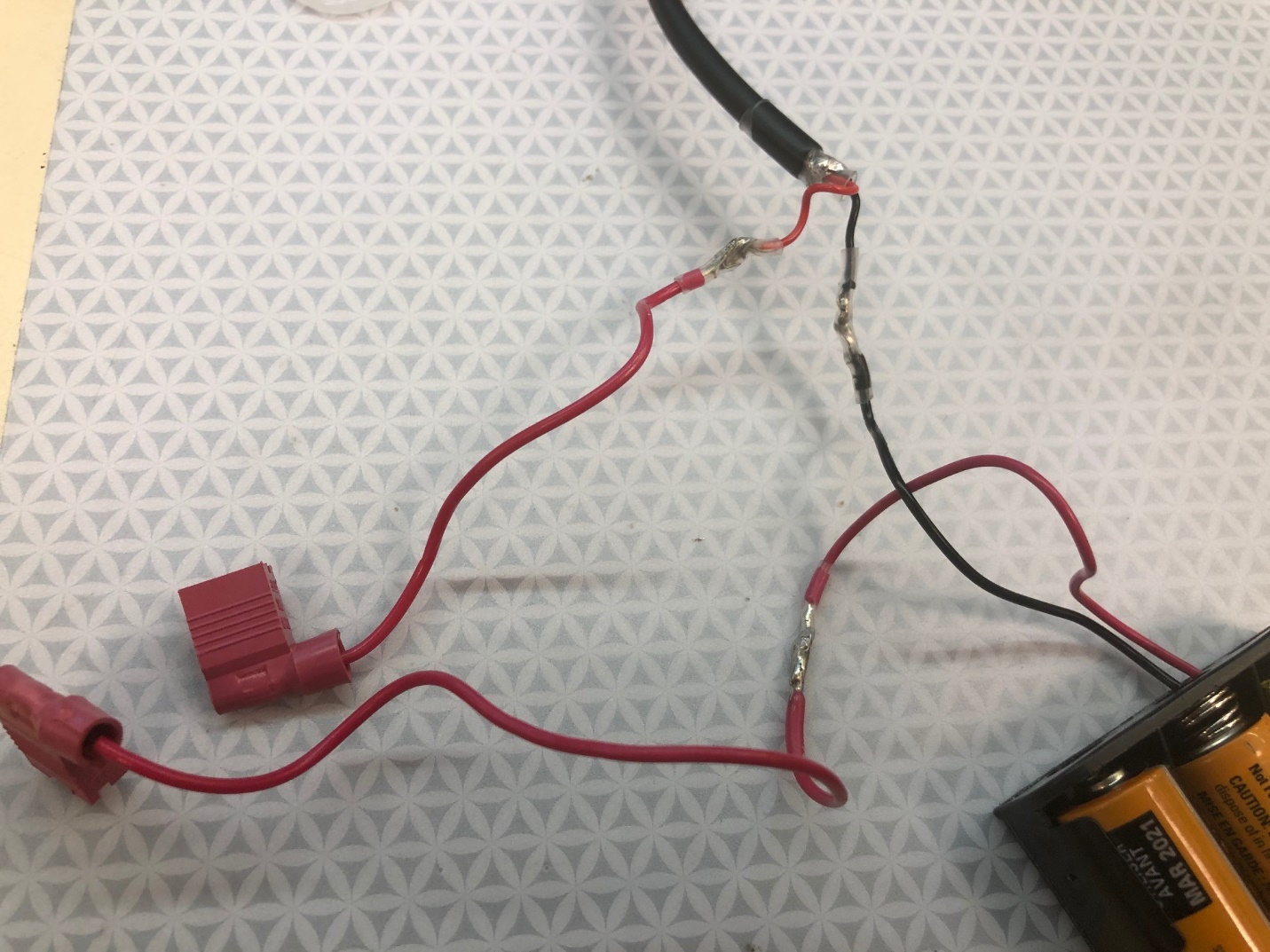
* Solder Mic onto a small breakout board
* Add a terminal block for 2 wires (GND and IN)
* Connect mic and terminal block using solder bridge
* Cut 2 short lengths of wire about 3 cm for GND and IN
* Crimp ferules onto both ends of each wire





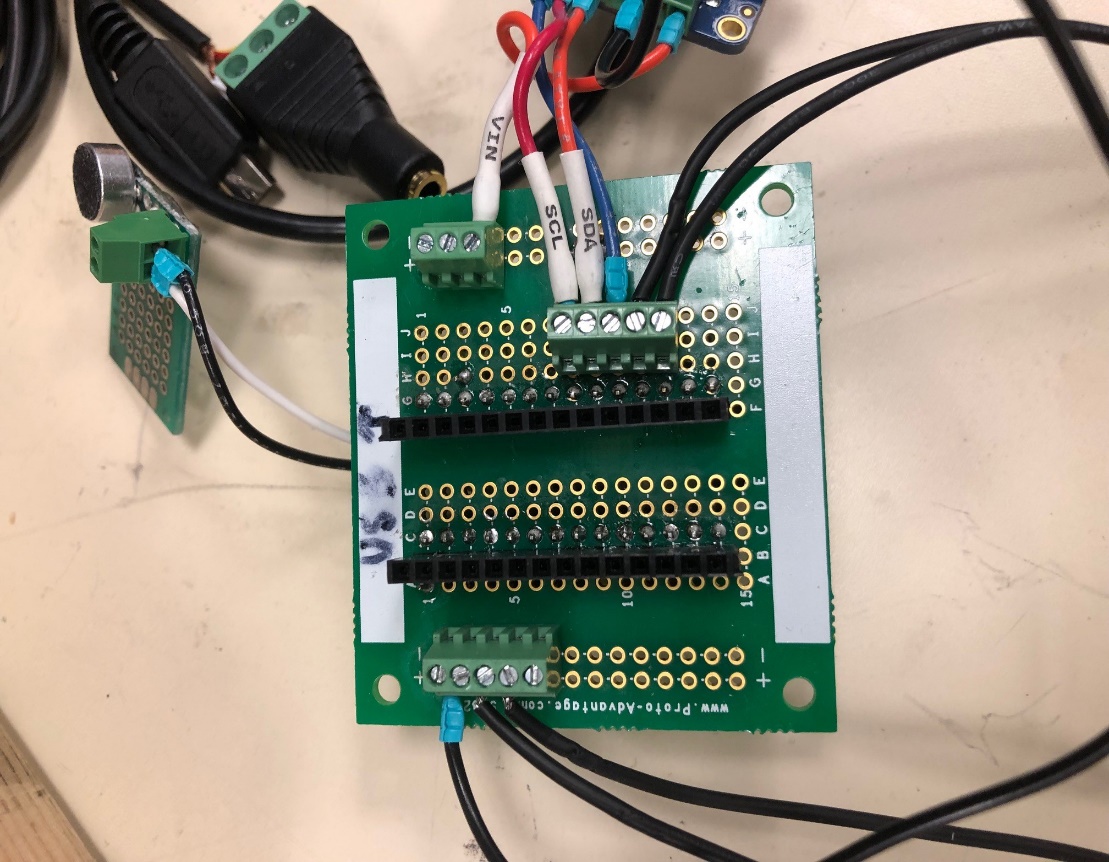
## Battery Pack

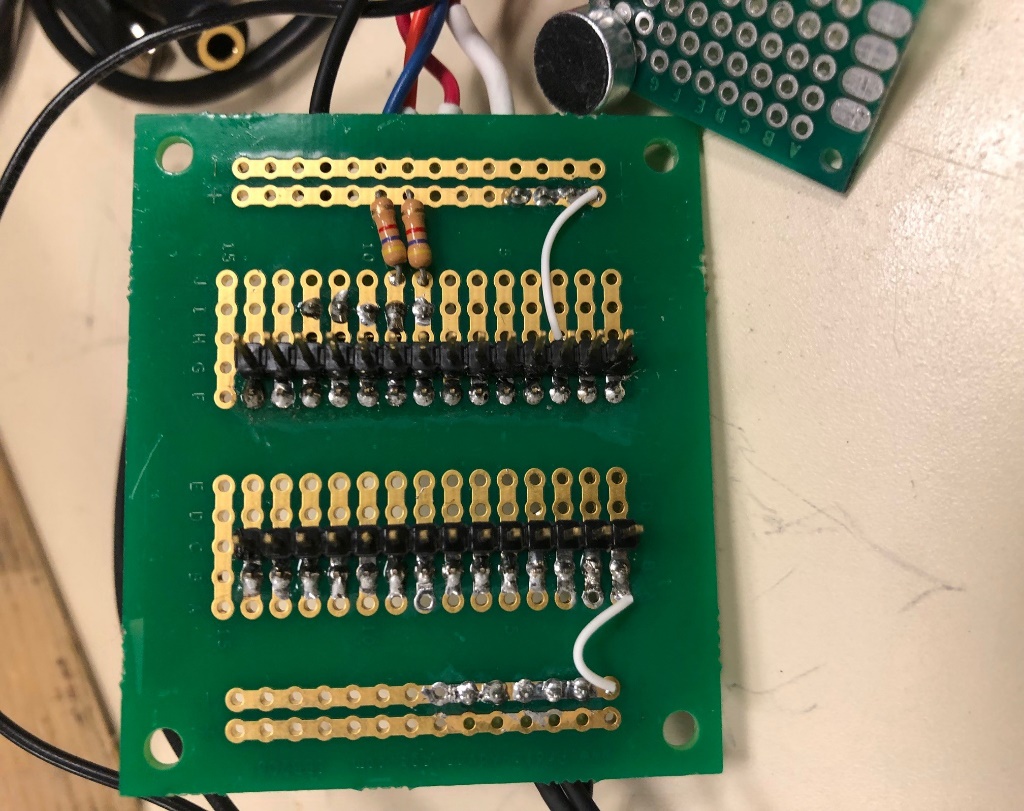
* Cut a microUSB cable, separate the GND and VCC wires.
  + As these wires are very thin, splice each wire onto a 5 cm length 22-gauge wire
  + Heat shrink the microUSB wire and the splices
* Extend the positive wire of the battery pack by splicing on a 5 cm length of 22-gauge wire
* Solder the GND of the battery pack together with the GND of the microUSB wire, heat shrink the connection
* Crimp a Quick Disconnect Flag Crimp onto the positive wire of the battery pack
* Crimp a Quick Disconnect Flag Crimp onto the VCC wire of the microUSB cable
* The Flag Crimps will fit onto the terminals of the rocker switch



Wiring of Battery Pack

## Connecting MPR121, Teensy 3.2, Buttons and Audio Board

* Onto a Proto Board
  + add headers for the Teensy 3.2 and sockets for the Audio Board
  + Add terminal blocks for GND onto negative side of power rail and Vin onto positive side of power rail
  + Add a 5 input wide terminal block for MPR121 inputs and buttons (should match Teensy 3.2 pins 15-19)
  + Add 4.7k resistor from Teensy 3.2 pin 19 to positive side power rail
  + Add 4.7k resistor from Teensy 3.2 pin 18 to positive side power rail
  + Connect wire from GND side of power rail to Teensy 3.2 GND
  + Connect wire from positive side of power rail to Teensy 3.2 3.3 V



Front side of Proto board

Opposite Side of Proto board

Connecting Wires:

* MPR121 Vin onto positive power rail
* MPR121 SCL to Teensy 3.2 19
* MPR121 SDA to Teensy 3.2 18
* MPR121 IRQ to Teensy 3.2 17
* Switch Keyboard button to Teensy 3.2 16
* Record Button to Teensy 3.2 15
* Other ends of Buttons to GND side of power rail
* MPR121 GND to GND side of power rail

For connecting Buttons to terminal blocks either use ferule crimps or tin the ends of the wires to fit in the terminal blocks.

Using a Proto board labelled A-J and 1-15:

* B1-B14 headers for Teensy 3.2
* C1-C14 sockets for Audio Board
* F1-F14 headers for Teensy 3.2
* G1-G14 sockets for Audio Board
* 5 input wide terminal block on I8-I13

Onto the Audio Board MIC and GND connections solder a 2 input wide terminal block

# Box Construction

The following modifications are made to a Hammond NFG 1550G box

* On base of box:
  + 3 holes for M3 standoffs for Teensy Audio Board
  + 2 holes for M2 standoffs for MPR121 board
* Forward face of box:
  + hole for MPR121 wires connecting to capacitive input (diameter depending on grommet, about 3cm)
* Backwards face of box:
  + Hole for rocker switch (about 22mm in diameter)
* Left face:
  + One hole for audio extension cable (depends on width of cable)
  + One hole for mic
* Right face:
  + 2 holes for buttons about 16mm wide
* Tap screw holes of box to fit M4 thumbscrews

Very Optional Step: Laser cut a clear acyclic lid for box



Hole for Rocker Switch



Hole for Grommet (MPR121 wires to capacitive inputs)

Holes for Buttons

Holes for Buttons



Holes for Mic and Audio Jack



Box Layout with all holes cut out

# Code

## General Set Up:

* The Capacitive Keyboard requires the Adafruit\_MPR121 library found here:

<https://github.com/adafruit/Adafruit_MPR121>

* + Make sure the .h file is named "Adafruit\_MPR121.h"
* To edit and change code make sure to have the Teensyduino add-on for Arduino <https://www.pjrc.com/teensy/td_download.html>
* Download main.ino from the Capacitive Keyboard Folder, (Community Engagement 🡪 On the Road 🡪 Demos-Shows-Supplies 🡪 Adafruit Musical Fruit 🡪 Capacitive Keyboard)
* Copy Audio files (in Capacitive Keyboard folder) to microSD card

## Main Variables

* Interval is how long the audio recording will be in ms
* touchedPins holds the output from the MPR121 board
* currentKeyboard is whichever keyboard is currently being used
* numNotes is how many notes are in the current chord
* cGain is the Gain for each mixer depending on how long the chord is
* chord[] holds notes in current array
* Audio Connections
  + 4 different AudioPlaySdWav because the max amount of notes in a chord is 4
  + 2 different AudioMixer4, need one for each side of output
  + audiooutputI2S for audio output
  + audioinputI2S ,audioRecordQueue, AudioPlaySdRAw, AudioAnalyzePeak, for audio recoding
  + patchChord1-patchCord10 set up audio out from playing SD .wavFiles
  + patchCord11-patchCord14 set up audio recording

## Setup Function

* sets up buttons are Inputs
* enables the audio board and SD card
* sets up MPR121 board as I2C device
* If the audio board or MPR121 board don’t set up currently, check connections and use the Wire library Scanner example code, include following lines to enable audio board:
* #include <Audio.h>
* AudioOutputI2S audioOutput;

## createChord Function

* Input: notes (output of MPR121 board), chord (array to store notes)
* Clears previous chord by replacing old chord with zeros
* Resets numNotes to 0
* Bitwise reads notes (the output of the MPR121 board is a 12 bit long binary number, if the position i has a 1 then input i of the board was pressed),
  + If it reads 1 then add currentNote to chord and increment numNotes by 1
* Returns numNotes

## playChord function

* Inputs: chord array, notes (number of notes returned from createChord function), cKeyboard (current keyboard)
* First changes gain for each mixer depending on how many notes in chord (notes variable)
* To find the corresponding .wav file to play in the 2D keyboards array, the first argument is cKeyboard and the second is chord[i], chord[i] stores the ith note in the chord
  + Ex, keyboards[cKeyboard][chord[2]] plays the second note in the chord

## Recording Functions

The following functions came from the Teensy Audio Example Recorder (<https://github.com/PaulStoffregen/Audio/tree/master/examples/Recorder>)

* startRecording()
* continueRecording()
* stopRecording()

The recordRaw() function calls the previous functions and records a sample that is as long as the inputted seconds variable

playRawFile() function plays a .RAW file in its entirety

## loop()

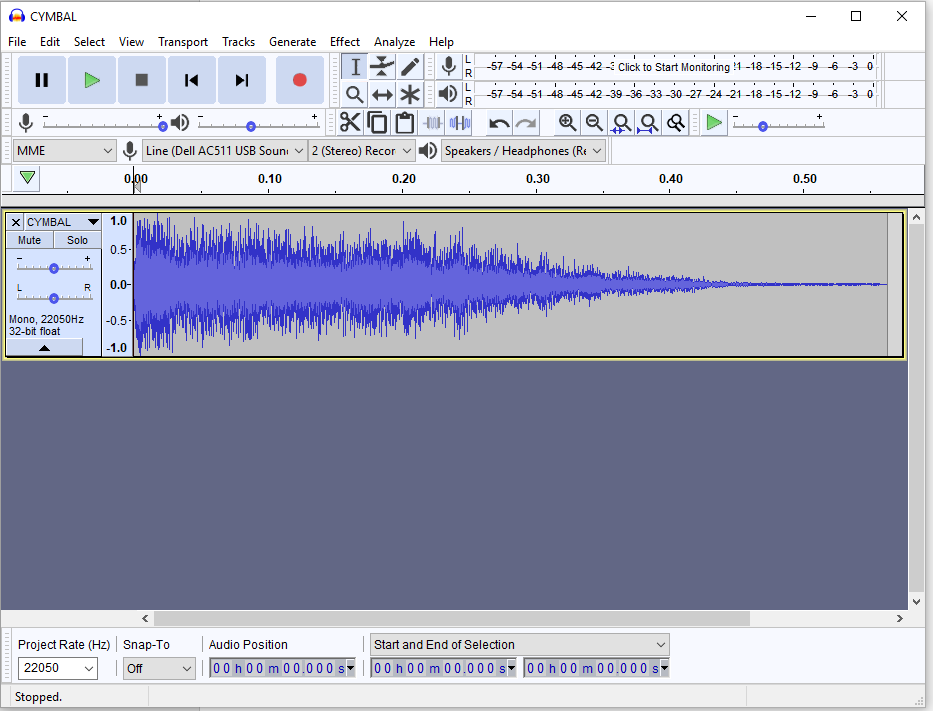
* checks if keyboard has been switched (the black keyboard\_switch button has been pressed)
  + changes keyboard by incrementing currentKeyboard by 1
* checks if the record button has been pressed and records a sample
  + changes currentKeyboard to keyboardLength (this means that the currentKeyboard will not be found in the keyboards 2-d array
* If the currentKeyboard equals keyboardLength then the current keyboard is the recorded sample one
  + As long as any pin has been pressed, play the recorded sample
* Otherwise call the createChord and playChord functions to play the current chord

# Audio File Sources:

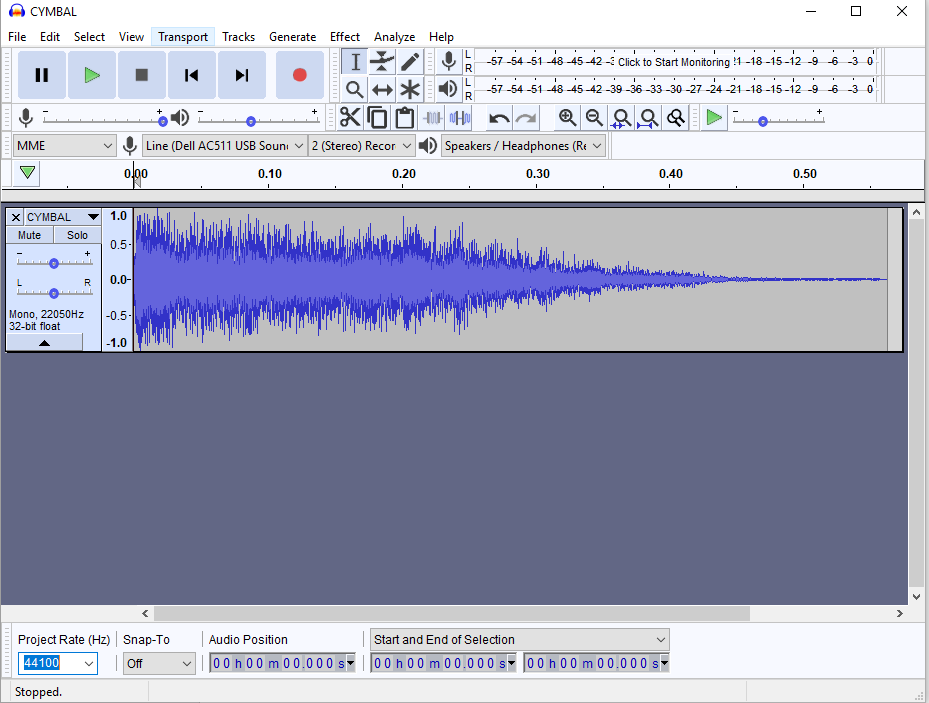
* Cats:: <https://freesound.org/people/tuberatanka/sounds/110011/>
  + The sample is then pitch shifted in Audacity to create new notes
* Piano Keyboard: auto generated tones in Audacity
* Drums: samples from Adafruit Playground Drum Machine (<https://learn.adafruit.com/adafruit-circuit-playground-express/playground-drum-machine>) and Tama Drum Kit by Snapper4298 (<https://freesound.org/people/Snapper4298/packs/11125/>)
  + Samples were adjusted and clipped to match microSD card requirements for Teensy Audio Board

# Adding New Keyboards

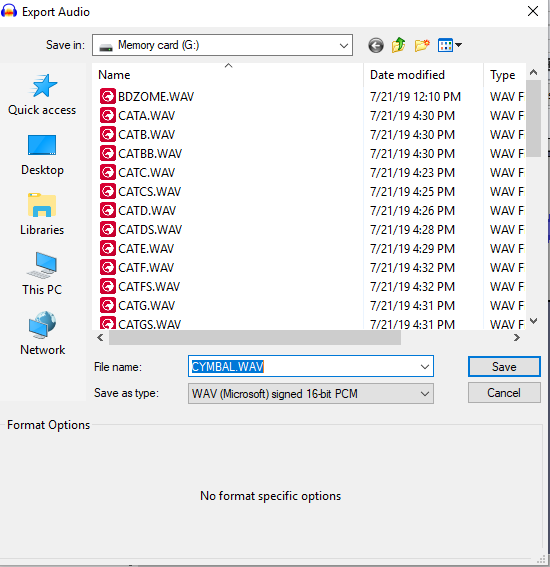
* To add a new keyboard you will need a mircoUSB cable, microSD card reader and 12 .wav files
* Make sure the .wav files are formatted as 16-bit 44.1kHz and the name is in 8.3 format
  + What does that mean and how do I make the file that if it’s not the case?
  + 8.3 format name means the name is in all caps and is no longer than 8 characters long and the file extension is up to 3 characters long EX. ELECBLIP.WAV is an acceptable/ideal name
  + It’s easy to fix the .wav format in Audacity:
    - Open file in Audacity, you can choose to import as a copy if you want to keep the original file as is
    - At the bottom left of the window there will be a Project Rate Hz, make sure it is 44100 Hz
    - export file as .wav, change the file name so the .wav is also capitalized (the file Save As Type should read “signed 16-bit PCM)
    - there’s no need to save the project or change the metadata of the file when saving



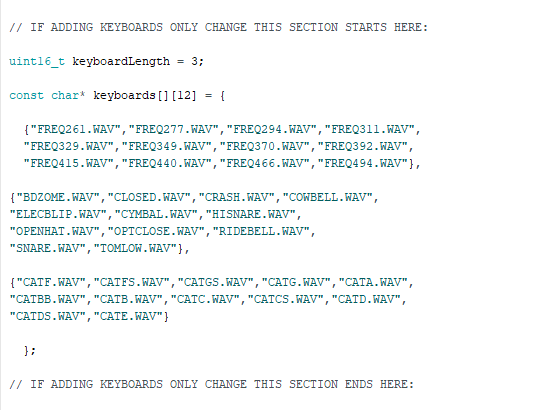
Original File with Incorrect Frequency



File with fixed frequency



* To access the microSD card on the Teensy Audio Board, unplug the microUSB cable connecting the Teensy 3.2 to the power supply, lift the battery pack up and remove the microSD card
* Using a microSD card reader, save the .wav files onto the microSD card and insert the microSD card back into the Audio Board
* Connect the Teensy 3.2 to a computer via microUSB
* Open up the Arduino software with Teensyduino and Adafruit\_MPR121 library installed (if not installed, follow instructions under General Set UP for Code
* Open the program main.ino from the Capacitive Keyboard Folder, (Community Engagement 🡪 On the Road 🡪 Demos-Shows-Supplies 🡪 Adafruit Musical Fruit 🡪 Capacitive Keyboard)
* Scroll to IF ADDING KEYBOARDS ONLY CHANGE THIS SECTION STARTS HERE
* Increase the keyboard Length variable by 1
* Add a comma after the } of the last keyboard
* Add the list of new .wav files, keep the keyboards spaced as pervious ones so it’s easy to read and modify



# User Manual

## Materials/Supplies

### Necessary:

* Capacitive Keyboard in box
* Speaker
* 12 fruits/ conductive materials + some extras for more fun times!
* Paper Towels

### Fun Extras:

* Wooden dowel
* Metal Cup 🡪 put water in the metal cup, can tap the cup or the water inside!
* Sodium Polyacrylate 🡪 turns the water into sludge, reduces chance of spills

## Set Up

* Stab the unused end of each pin into a conductive material
* Connect the audio speaker (insert into audio jack)
* Flip the battery rocker switch (back of box) from the OFF position to the ON position
* Make sure that you turn on the keyboard AFTER stabbing the unused end of each pin into the different materials
* Red LED on Teensy board should be on and flashing

## During Use and Trouble Shooting

* Each pin is its own unique note.
* When the keyboard is started it is in its first mode: piano
* To change modes, press the black button on the side of the box
* Whenever a pin is touched a LED on the MPR121 board should blink red and a sound should be heard
* About every half hour try to put the materials back into some kind of order / tidy up the space

### If a fruit stops working/playing a note when tapped:

* Make sure the end of the clip is fully inserted into the fruit
* If the issue is still not resolved flip the battery switch OFF and then ON again
* Tap each fruit and see if the problem is resolved
* Tap the problematic fruit and check if it now makes a sound
* This is normally a problem with lemons and limes after they have been heavily tapped

### Other Issues:

* If the red LED on the Teensy board does not turn on, the battery has run out, please replace the batteries
* **In general, the solution to most problems with this Demo is just to flip the battery switch OFF and then ON again**

## Clean Up

* First slide the battery switch to the OFF position and unplug the speaker
* Remove all the fruit from the pins
* Wipe down/clean the pins (removing any fruit gunk left over from the day’s use)

### Facilitation Fun!

* Use a wooden dowel to explore the difference between insulating and conducting materials
* Leave extra fruits out as explorers are very keen to move around pins
* Please don’t leave out just water in a cup as there are high chances of spills 🡪 mix in sodium polyacrylate to make sludge that won’t spill
* Let explorers tap the outside of the metal cup, the inside of the metal cup, and the sludge inside
* Grapes should not be used with this demo 🡪 many will stick them directly into their mouths. Conversely, brussel spouts are wonderful for this demo!
* Let explorers tap just the metal end of the wire to produce a sound
* A really great way to explain what’s happening is to mention how our hearts beat because of electrical impulses. The electricity travels from us through the conductive material, into the metal tip and up the wire.
* Encourage stabbing one material with two wires 🡪 this completes the circuit and makes a continuous sound
* Encourage touching the metal ends of the wires together
* Most explorers will want to move around pins, stab other fruit and all around push the demo as far as they can with repeated tapping. This is awesome! Encourage them to explore as they want to with some prompts. If the demo ever stops working just reset it with the battery switch!