

## Lab #2 – Music Day

### *How to work as a group*

This lab involves building hardware, coding, and writing a short report. You only need to turn in one report per group.

The goal of having only one written report per group is to save you time writing and save me time reading. However, my expectation is that everyone in the group does the work together and discusses the questions together. The goal is *not* to have only one person learn the material 😊.

### *Overview:*

In this lab, we'll

- get more practice using the PyBoard. We'll use it to synthesize and play music
- get more practice using variables and functions
- learn to physically hook up peripherals (e.g., headphones) to the PyBoard

### *Background:*

The book *Think Python*, Chapter 3 (all about functions)

### *Lab setup – software*

The PC has a folder already set up for our course materials. In addition, you each have a folder set up for your own use:

- The class folder is Q → en1EIY → 2022f → public\_html → labs → code
- Your own folder is the Z drive. In a Windows File Explorer, it will be named (e.g., for me) “joelg’s home directory.”

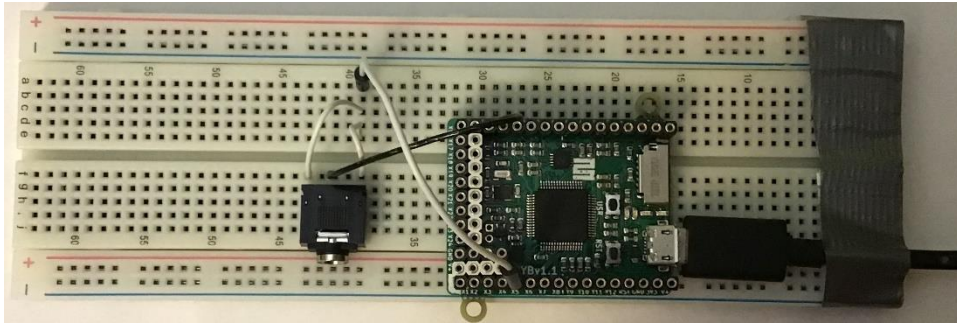
For this lab, you'll copy a Python program from the class shared folder to your own folder and then modify your copy to add your own code.

Use the standard Windows copy/paste to copy the file `2_pyboard_audio.py` from the course folder and paste it into your own folder. If you've not used Windows much, you can find a Windows File Explorer by clicking on the Windows “Start” icon in the lower left of the screen (it looks like a little Window with four panes), scrolling down to “Windows system” and clicking on it, and then selecting “File Explorer.”

### *Lab setup – hardware*

To get audio for this lab, you must hook up a headphone jack to the PyBoard's audio output. The PyBoard creates audio with a device called a **D**igital to **A**nalog **C**onverter (DAC). The program `2_pyboard_audio.py` (which will be running on the PyBoard) builds digital signals, sends them to the PyBoard DAC, and voila – you have audio output.

The audio output will appear at the PyBoard's pin X5. So you must hook up the pins X5 and ground to drive an earphone jack, and then insert your own earphone plug into the jack. Here's what the final hookup looks like:

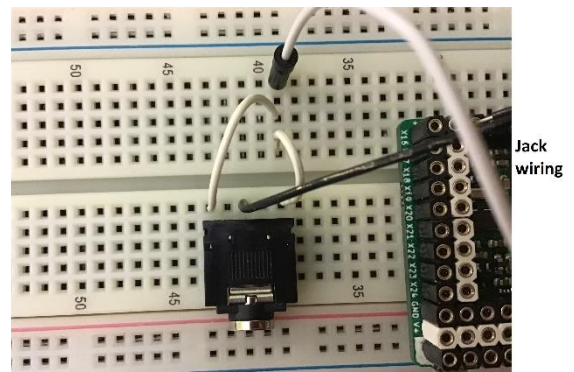
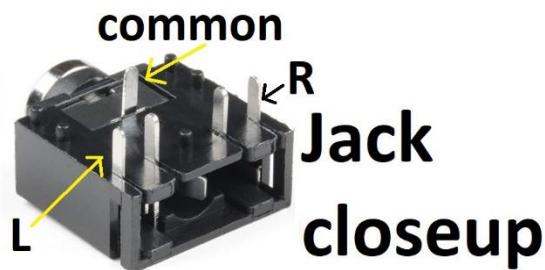


**Final assembled board**

You can see several things in the picture:

- the PyBoard and breadboard (which you've already assembled).
- a headphone jack (the black thing to the left of the PyBoard).
- a few wires interconnecting them. Specifically, the black wire connects the PyBoard's ground (marked GND) to the headphone's "common" terminal (shared by the left & right earpiece). The white wires connect the PyBoard's DAC output (pin X5) to drive the headphone's left and right channels.

The first part of your assembly will be mounting the headphone jack.



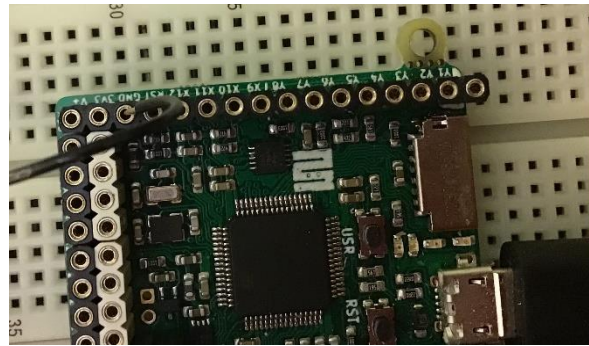
The picture on the left shows the bottom view of the jack; only three of its 5 pins are actually connected. When you insert the jack into the board (as per the right-side picture), those three pins each get their own column on the breadboard (so that we can connect to them independently). If the jack were inserted any lower on the breadboard, the "common" pin wouldn't have a hole to go into; any higher and there wouldn't be a spare row above the jack to use for wires.

The white wires drive the PyBoard DAC into the L and R headphone channels; the black wire drives GND from the PyBoard to the headphones. The breadboard internally connects the three white wires to each other since they are all in column 41.

The PyBoard connections are shown next:



Pin X5;  
DAC  
output



GND  
wiring

With that, just plug in your headphones and you should be all ready to go! Log into your Halligan account, plug in the USB just as last time and open up Thonny.

In Thonny, click on File/open, tell Thonny that your file is on the host, navigate to your personal folder and double-click on `2_pyboard_audio.py`.

### *Programming simple sound*

Scroll through `2_pyboard_audio.py` in the Thonny program window. You'll see some various setup code at the top of the file. We'll go over it in our next lecture, but feel free to try and figure it out now if you like, or ask for help with it.

Near the bottom, you'll see the big comment

```
#####
### Your code starts here
#####
play_note (1000, 10)
pyb.delay(500)
play_note (1000*1.0595, 5)
pyb.delay(500)
play_note (0,0)
```

What does this code do?

- **Play\_note()** is a function that gets defined near the top of `2_pyboard_audio.py`. In this case, it first plays a 1000Hz tone at a volume level of 10 (the volume can range from 0 to 10).
- You've already seen the code **pyb.delay()**; in this case, it delays for 500ms, or 0.5 seconds. Without this, the second note would happen so quickly after the first note that you wouldn't really hear the first note at all.
- The final **play\_note(0,0)** plays a note at a volume level of 0 – i.e., not at all. Without this, the second note would last forever.

### *What to do next*

Experiment. Change the notes and durations. Add new notes to your song. The middle C on a piano is 231Hz, and every successive half note has a frequency 1.0595 times higher; so C# is  $231 * 1.0595 = 244.7\text{Hz}$ . In the code above, one half note higher than 1000Hz is 1059.5 Hz.

You've learned a bit about variables. So next, use variables to make it easier to program more complex songs:

- Use variables to assign each of the twelve notes for an octave notes to 12 different variables.
- Then, using only the variables (not the numerical frequencies), write a nice little song.

Finally, you learned last week how to turn LEDs on and off. You might combine LEDs with music to create your own mini-disco. Here's one suggestion (but of course feel free to do your own):

- G (.25 seconds, .05 second pause) and LED #4. Repeat three times
- D# (2 seconds, .05 second pause) and LED #2
- F (.25 seconds, .05 second pause) and LED #3. Repeat three times
- D (2 seconds, .05 second pause) and LED #1

### ***In-class checkout***

- Grab Neelofar, Owen or Joel. Explain the hardware; what are all of the wires doing? Could you wire the board any differently and still have it function?
- And, of course, show off your music!

### ***What to turn in:***

A lab report that contains your code and a short description of what it does. You need only turn in one report for each group.

If you built the speaker hardware (see below), then show it to us during your in-class checkout and include a picture in your lab report.

### ***Bonus***

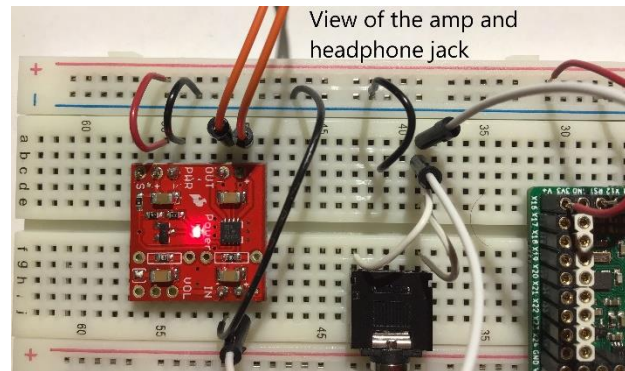
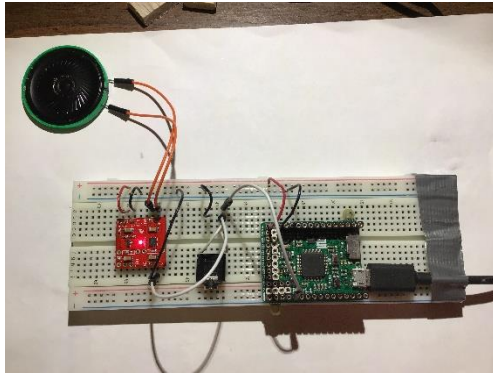
We expect that students will have a wide variety of programming experience. For those who already know a bit more, feel free to experiment with more complex programs. For example, you might

- use loops to, e.g., play the same song several times.
- modify the code in **play\_note()** to not merely use sawtooth notes; you could use square waves or sine waves instead.
- modify **play\_note()** to make it easier to use; e.g., add the note length and any pauses as extra arguments, or even let it take the note names instead of frequencies.
- if you're really clever, you could modify **play\_note()** to play chords by creating two notes and adding them together before sending them to the DAC.

If you do some of the bonus work, please show it off to us during class! And include it in your lab report if you like.

### ***Speakers and amplifiers***

If you like, we also have some speakers and amplifiers. The speakers are, of course, substantially louder than the headphones. However, the PyBoard is not powerful enough to drive a speaker directly, so we must use an amplifier. Here's a picture of the amplifier and speaker hooked up:

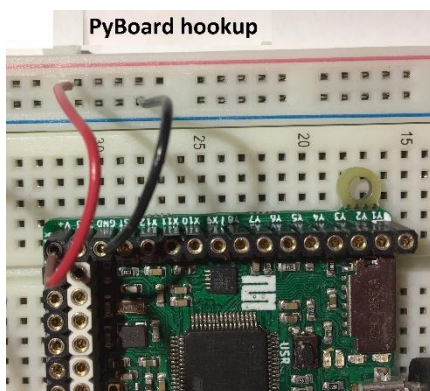


The leftmost picture above shows the entire circuit. In addition to the existing PyBoard and headphone jack, we've now added an amplifier (the small red board) and a small speaker. The rightmost picture above is a closer view of the amp and the headphone jack.

We see that the wiring is now different in several ways:

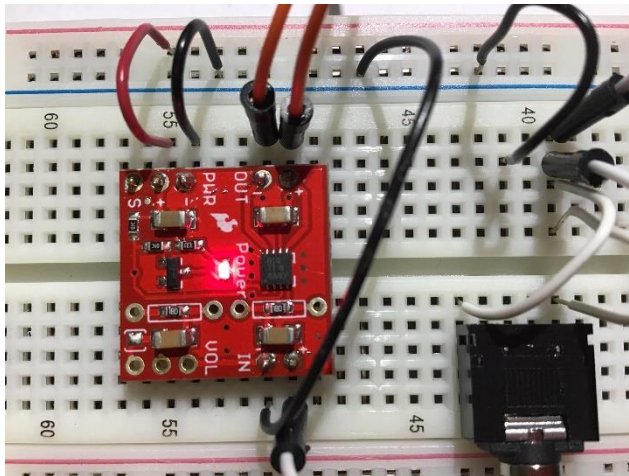
- Obviously, the amp and speaker have been added.
- With the addition of the amp, our system now has multiple power and ground connections. We'll thus dedicate the top two rows of the breadboard for power (top red "+" row) and ground (blue "-" row just below it).
- We've rewired the headphone jack slightly because of that.

Now for some details. In the picture below, you can see that the PyBoard is now driving its power (the V+ pin) into the top red power row via the red wire. It is driving its ground (the GND pin) into the blue row just below via the black wire. The white wire where the PyBoard drives out its DAC output is unchanged and hence not shown.



Next, a detail of the amplifier board and the headphone jack.





We have:

- The red and black wires at the upper left power the amp's power and ground (columns 55 and 54 respectively).
- The orange speaker wires are in columns 50 and 51. (They don't look very orange in this picture but they are!)
- The PyBoard DAC output (again, the white wires) now also drives the amplifier "+" input (column 50).
- The amplifier "-" input, like the headphone "common" input, connects to ground. Now, though, that is the black wires connecting them to the 2<sup>nd</sup>-row blue strip.