

# Using multiple files on a PyBoard

## *What's the problem?*

Most of our labs only require you to run software that is all in one file. This is relatively easy; you just open the file in Thonny, set Thonny to use “MicroPython (generic)” as its compute engine, and you're good to go.

Sometimes, however, you want to use code in more than one file. For example, you might be using *5\_emg\_process\_pyboard.py* to analyze an sEMG signal, and want to use *2\_pyboard\_audio.py* to play music as a result. If you try to just have your code in *5\_emg\_process\_pyboard.py* call functions from *2\_pyboard\_audio.py*, you'll probably find that it doesn't work unless you know the tricks.

This document is to teach you those tricks.

## *Solution #1 – just don't do that*

Perhaps the easiest solution is –stick with one file! In the example above, you could just merge the two files into one. I.e., copy/paste most of the code from *2\_pyboard\_audio.py* into *5\_emg\_process\_pyboard.py*. This is a nice end run around the issue, and usually works fine. There is, however, one issue to beware of.

If you look closely at *2\_pyboard\_audio*, you'll notice that much of the code defines functions, such as **play\_note()**, that you want to call. That part is good! However, the code at the end of the file may contain actual calls to **play\_note()**, perhaps left over from lab #2 . If you include this code in your merged file, then it will execute your songs from lab #2, which is unlikely to still be useful. The solution, of course, is easy – just don't copy that code into your merged file.

## *Solution #2 – imports*

Large software systems are built from *many* files, and Python absolutely has the ability to work with code in more than one file. The tricks to do so are slightly more complex than the file-merge technique above, but not too much more complex. Here are the steps involved if you would like to try:

- Start with *2\_pyboard\_audio.py* and, just as in solution #1, rip out the unneeded code.
- Using Thonny's File Display, copy your stripped-down version of *2\_pyboard\_audio.py* to the PyBoard. This is important – since your main program *5\_emg\_process\_pyboard.py* will be running on the PyBoard, it won't be able to find *2\_pyboard\_audio.py* unless that file is on the PyBoard also.
- You will notice that *5\_emg\_process\_pyboard.py* starts with the line

```
import pyb, array
```

You must modify this line slightly to become

```
import pyb, array, 2_pyboard_audio
```

This is how your code in *5\_emg\_process\_pyboard.py* knows to load *2\_pyboard\_audio.py* and thus get access to the code inside of it.

- Finally, when you want to use a function such as `play_note()` that is in `2_pyboard_audio.py`, you must refer to it as `2_pyboard_audio.play_note()`.

That's it for solution #2!