

## Lab #3 – sEMG on an oscilloscope

### *How to work as a group*

This lab involves lots of lab work, thinking about the questions and writing a 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 lab 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

- learn how to connect electrodes to your body to capture your *sEMG* signals
- use our breadboard to connect the electrodes to an AD8232 pre-amp that takes the weak sEMG signals and makes them much stronger
- learn how to use an *oscilloscope* (a standard piece of lab equipment) to view the amplified signals
- use the PyBoard to digitize and capture the signals.

This will prepare us for next week's lab, where we analyze the signals using the host computer.

### *Legal reminder:*

This lab involves capturing your sEMG. Legally, this signal does represent medical data. As such, federal law says that you have the right to keep it private, and we will of course respect that law. Please refer to our separate legal sheet (which you should have read and signed).

The quick summary:

- Each group need turn in only one person's sEMG, and does *not* need to say whose it is. Any group member is free to simply tell their partner(s) that they prefer not to turn in (or indeed to even measure) their own signal.
- If no members of the group want to turn in their sEMG, you can download the instructor's sEMG (and there is no need to even state that you did so).

### *Background: what is an sEMG*

As we discussed in class, your brain activates your muscles by sending them electrical signals. An *EMG* (**electromyogram**) uses electrodes to capture those signals. A clinician or researcher may sometimes use a *needle EMG*, sticking small needles through your skin to make direct contact with the signals. More commonly (and less dangerously!) we use stick-on pads at the skin surface to capture the signals with a *surface EMG* (sEMG).

### *Learn to use the oscilloscopes*

You'll be using an oscilloscope in this lab. The course web page has a document on how to use it, in the *Resources for the labs* section. You may as well do that now.

### *Lab setup – software*

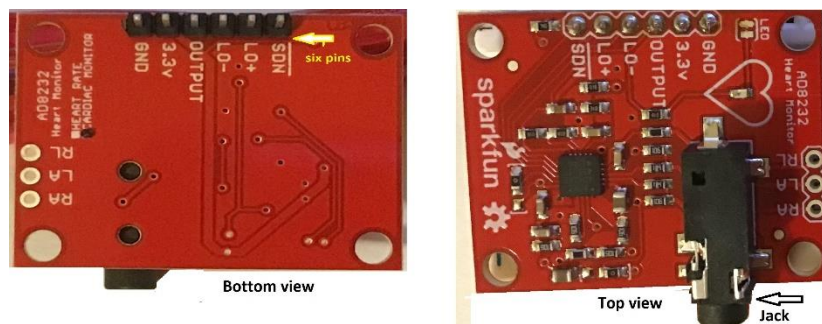
The setup for this lab involves first copying a Python program from the class shared folder to your own folder (where you can modify your copy).

The file locations are the same as last week; the class folder is Q → en1EIY → 2022f → public\_html → labs → code. Use the standard Windows copy/paste to grab the file *adc\_samples.py* and paste it into your own Z-drive folder (i.e., your own home directory).

### **Lab setup – hardware**

Electrical signals coming from your body are typically quite weak and noisy; the AD8232 board is a hobbyist-grade commercial preamplifier that is designed to amplify a weak signal and filter away much of the noise. It is specifically designed for an ECG (the signal from your heartbeat), but works reasonably well for sEMG signals also.

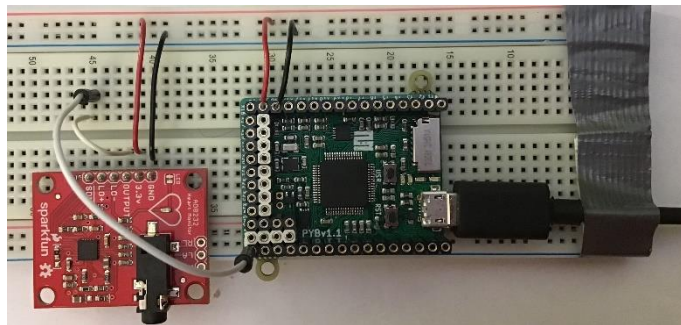
Here are pictures of the AD8232 board.



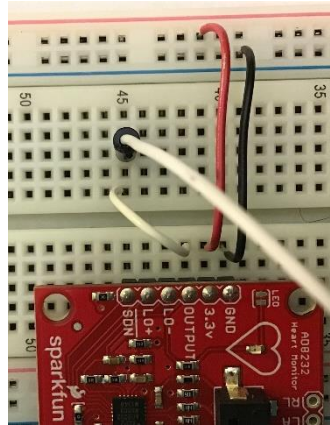
Things to note:

- It has a headphone jack (top view, lower right). This is not for plugging in headphones, though! Instead, you have a three-electrode set with a headphone plug. You plug it into the jack to connect up the electrodes.
- There is a six-pin connector on the AD8232 board (top of the bottom-view picture above). You plug this into the breadboard and it then lets you connect up power (3.3v), ground (GND) and the AD8232 output signal (OUTPUT).

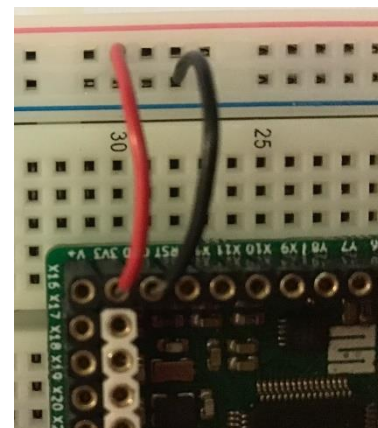
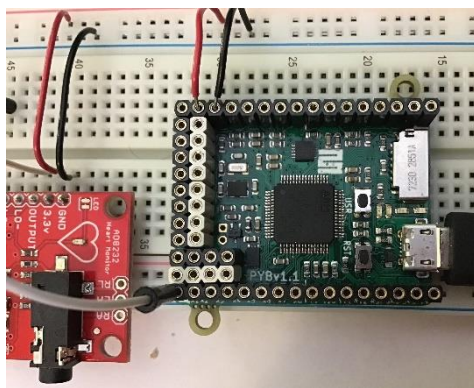
Here is a picture of what your final assembly will look like (but without the scope probe):



Here is a close-up of the AD8232 hookup:



You'll note that its power (i.e., 3.3v. column 41) and ground (GND, column 40) are driven from the top two horizontal rows (the top red one for power and the bottom blue one for ground). Its output (column 42 on the bottom half of the breadboard) gets wired to column 45 on the top half to give us some more space to play with; then it will drive the PyBoard's X1

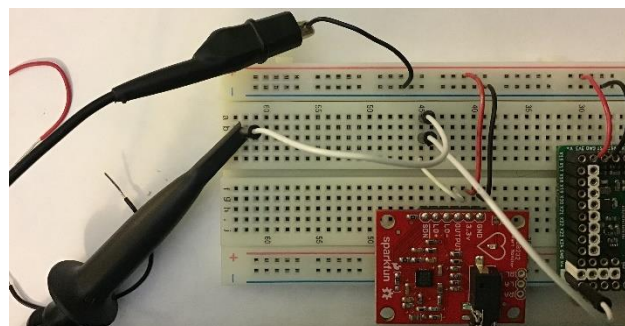


pin.

The X1 pin is all the way on the bottom left of the PyBoard. We then hook up the PyBoard to drive power (its 3V3 pin) and ground (its GND pin) with red and black wires respectively.

**Note that you must *not* use the PyBoard's V+ connector – that pin drives 5V, which can fry the AD8232 board!**

Finally, hook up the scope probe as per this picture:



The scope's ground (i.e., the black alligator clip) attaches to our common ground using a short black wire; the scope's signal probe clips onto a white wire that plugs into column 45 with the other white wires.

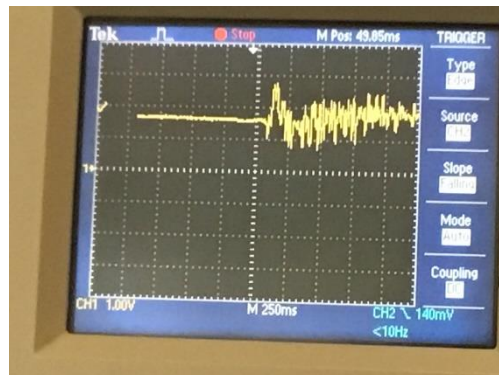
### ***Lab setup – electrodes***

The course website has a short video on how to attach the sEMG electrodes. A few things add to it and/or repeat:

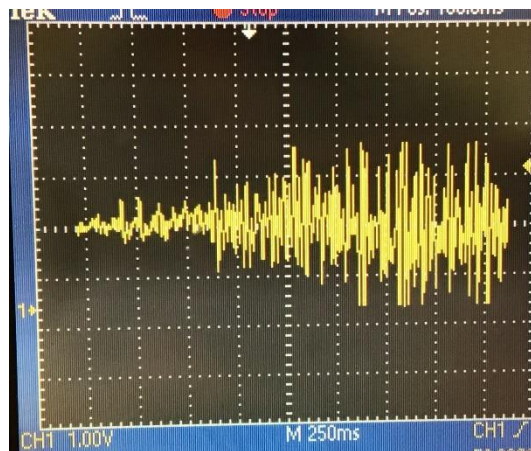
- we have skin-prep tape available. It is essentially very fine sandpaper; if you scrape your skin *lightly* with it (don't scrape yourself red!) then you get a better connection and stronger signals. Most people do fine without it until cold weather comes and your skin becomes drier.
- attach the biceps (blue) electrode right over the belly of the muscle
- attach the reference (black) electrode over fat or bone (not muscle) near your elbow
- avoid hair if possible
- attach the RLD (red) electrode to your wrist on the same arm as the other electrodes.

### ***Viewing your sEMG on the scope***

You should see a fairly obvious difference between the waveform when your muscle is relaxed (left side of the screen) vs. when it is active (right side):



Capturing the actual force of a contraction may be more challenging. Here's an effort I made to capture my biceps gradually contracting more forcefully:



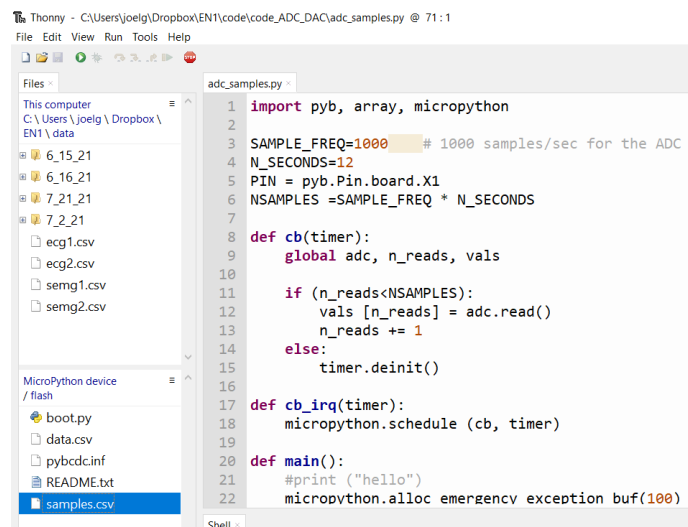
Take a picture (e.g., with your phone) of some of your own waveforms.

Once you've done that, the next step is to use the PyBoard to capture a waveform and upload it to the host PC to use next week.

### ***Capturing your sEMG signal with the PyBoard using `adc_samples.py`***

`Adc_samples.py` is a Python program that runs on the PyBoard. It looks at the signal coming into the PyBoard on pin X1 and digitizes that signal. It does so by sampling the signal at 1kHz (i.e., 1000 samples/second) for 10 seconds, turning each signal into a 12-bit number (i.e., in the range [0,4095]), and then saving all 10000 numbers into the file `samples.csv`.

When you press Thonny's green "run" arrow to run `adc_samples.py` on the PyBoard, it will collect 10 seconds of data. It will then save the data as a file `data.csv` on the PyBoard. The green LED will be on during capture, and the red LED will be on while the file is saving (which may take up to a minute or so). When it's done, upload the file to your home folder



on the host PC using Thonny as below:

- First make sure that the Files pane is visible on the left side of the screen. If not, use View → Files to make it visible.
- Next, navigate to your Windows home directory (Z drive) in the top half of the Files pane.
- Finally, find `data.csv` near the bottom of the Files pane, right-click on it and choose the *Upload* option to move it to your home directory.

Once it's uploaded to the host, you'll turn it in for this week's lab. And don't delete it on the host – you'll use it in next week's lab too.

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

Make sure to do your in-class checkout (see below) before you leave. Then turn in

- A lab report that contains your scope pictures (for off-then-on and for gradual increase). You need only turn in one report for each group. Feel free to add pictures from multiple people if you choose.
- The digitized capture file `data.csv` for the on-then-off capture.



There are no questions to turn in and no bonus section this week.

***In-class checkout:***

- “You can set the scope’s voltage and time scaling controls to get fine detail or to see the big picture, but not both” – explain what that means and why it’s mostly true. But can you think of a clever workaround so that you can get multiple pictures of the *same* waveform at different levels of detail?
- Can you explain as much of the code as possible (i.e., the parts we discussed in class)?