

Lab #6 – collecting your ECG for analysis

Overview:

In this lab, we'll

- learn how to connect electrodes to your body to capture your ECG signals
- connect the electrodes to drive an AD8232 pre-amp that takes the weak sEMG signals and makes them much stronger
- use an oscilloscope to view the amplified ECG
- use a Nucleo 432 board to digitize and capture the signals. The digitization happens with an analog-to-digital converter (ADC) inside the STM32L432 chip. If you've never been exposed to an ADC before, the course web page has more info.

This will prepare us for our final labs, where we'll analyze your ECG to extract your heartrate.

Legal reminder:

This lab involves capturing your ECG. Legally, this signal represents 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 ECG, 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. Others should not try to make you, or even to ask why.
- If no members of the group want to turn in their ECG, you can use the instructor's ECG (and there is no need to even state that you did so). `Task_canned_ECG()` in our lab code drives the instructor's ECG out onto Nucleo pin A3, and you are welcome to use this signal as your own. There will be no penalty for choosing this option.

Oscilloscope setup

Not every one of your heartbeats will be the same, and it's a slowly-changing signal. It's thus often best to work in scan mode, with a horizontal sweep on the order of 250 ms per division and roughly 200 mV per vertical division.

ECG terminology: electrodes and leads

Each wire that you connect to your body is called an *electrode*. Basic physics says that voltage is defined as the difference between the electric potential at two different points, and so any particular reading that you see on a screen is the voltage difference between two electrodes. Each of these readings is called a *lead*. I don't know why it's called a lead – but that's standard medical terminology, even if it usually sounds a bit strange to engineers at first.

A typical ECG in a doctor's office attaches 10 electrodes to your body and displays 12 leads. With 10 electrodes, it would only take 9 leads to fully represent the data, and so we have some redundant readings – the redundant readings make it easier for a clinician to quickly diagnose issues.

The particular reading we will perform in this lab is called Lead I in cardiologist-speak.

Noise in ECGs

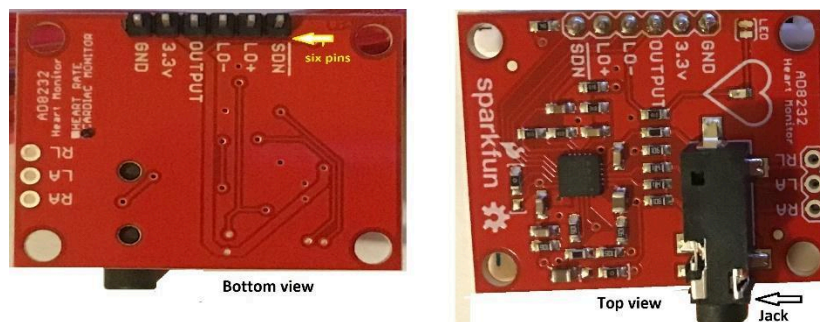
The signals that your heart produces are greater than 60mV in your heart, which is more than enough to spread across your heart and cause a contraction. By the time they travel through your body to our electrodes, though, the signal is only a few mV. It is easy for environmental noise to overwhelm an ECG, and the most common source of environmental noise is the 60Hz noise from power distribution and appliances in homes and buildings.

The ECG amplifier is a *differential amplifier*; if the same noise is applied to both electrodes of any lead, then the AD8232 (which measures the *difference* between its two electrodes) will not show the noise. Thus, the main source of noise that we see in our readouts is noise that affects the two electrodes differently.

You may wonder why, if a lead only involves two electrodes, our experimental setup uses three electrodes rather than two. The third electrode is used to minimize noise. It uses a system called *right-leg drive* (RLD), where the AD8232 board tries to dynamically find noise that is common to both electrodes and then drive it back to your body (via your right leg) to cancel it out.

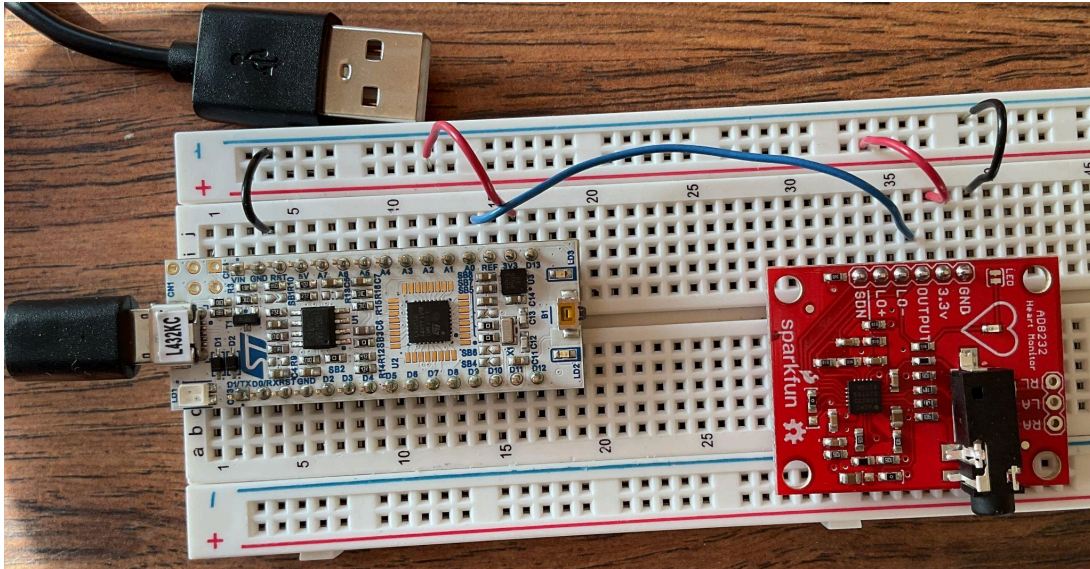
Lab setup – hardware

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 ECG and filter away much of the noise. Here's a picture of the AD8232 board..

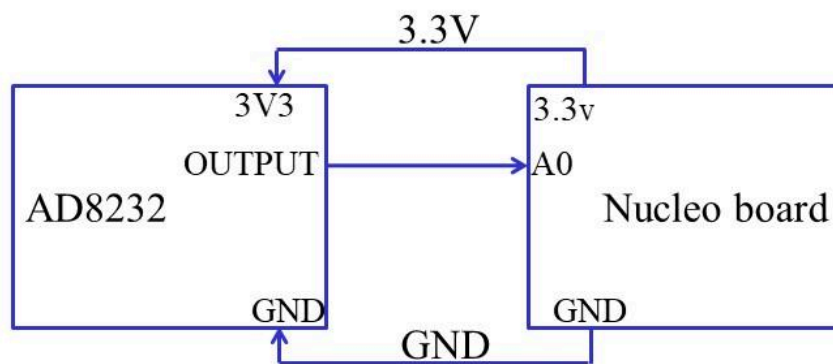


Note the headphone jack. This is not for plugging in headphones, though! Instead, you have a three-electrode set with a headphone plug. You plug it into the jack to connect up the electrodes.

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



Here's a schematic diagram:



How do the boards get powered? As usual, the USB cord supplies power to the Nucleo board (and allows the Nucleo board to talk to a host). USB supplies 5V; the Nucleo board has an internal voltage regulator that regulates 5V down to 3.3V and drives it out to supply the AD8232 board. Running a full 5V into the AD8232 would likely fry it.

The Nucleo board's A0 pin is an analog input to its ADC, which we drive with the AD8232 output. Note that the picture above does not show the wire hookup well, due to the camera angle – but the schematic is correct.

Lab setup – electrodes

Our lab setup has three electrodes:

- The red RLD electrode should always be attached to your right leg.
- The blue electrode is the positive input. For lead I, it connects to your left arm.
- The black electrode is the negative input. For lead I, it connects to your right arm.

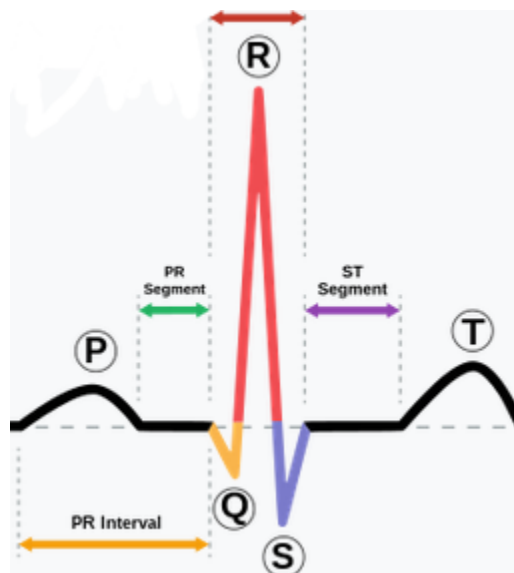
A few tips:

- Try to attach the electrodes where they sit over bone or fat, not muscle. If they sit over a muscle, they will read the signal that the muscle generates rather than the cardiac signal.
- Avoid attaching over bodily hair if possible. This will not only worsen the skin conductivity and make your signal weaker, but will hurt more when you take the pad off!
- The arm electrode attaches on the inner side of your arm, about halfway between your elbow and shoulder, right in-between your biceps and triceps so as to avoid being on top of either muscle.
- The leg electrodes attach a little bit above your ankle, being sure to avoid the calf muscle.
- If the air is dry, your skin resistance can be high, again worsening electrode contact. We have some skin-prep tape (essentially medical-grade sandpaper!) you can use to remove part of your dead-skin layer and improve contact. It's called 3M Prep Tape.

The course website has a short video on how to attach the electrodes.

Viewing your ECG on the scope

Hook up a scope probe to sample the AD8232 output (i.e., to view the wire from the AD8232 OUTPUT pin to the Nucleo A0 pin). Make sure you connect the scope's ground to the breadboard ground. Use scan mode, with about .5s per horizontal division and 1V per vertical division. Your ECG will probably have something close to the classic shape below



Again, to use the instructor's ECG, you would build the Nucleo project (as detailed below), and take the output from Nucleo pin A3. You would use `task_canned_ECG()`; note that you must learn how to use this code for the lab checkout in any case.

Debugging a weird or nonexistent scope picture

If you don't see a nice ECG signal on the scope, there are a few things you can try

- Double check that you have power everywhere. The AD8232 board has an onboard red LED that indicates power. Alternately, touch a scope probe directly to the

AD8232 board's 3.3V pin (i.e., not on the breadboard, to see if the breadboard's connection to the AD8232 is bad).

- Try swapping out a different AD8232 board.
- Try using a different scope channel or a different scope.

Using the STM32L432 Nucleo board ADC and DAC

The next step is to use the Nucleo board to digitize the waveform using the ADC.

First set up your VSCode project. You can simply copy it from clean PlatformIO/FreeRTOS link on the course home page under “Common files for the labs.” Then unzip and build as usual. You do need FreeRTOS for this lab.

Take a look at the code. Start with the function `task_ADC()`, and ignore the data-collection part. It's simple – it just reads samples from pin A0 with `analogRead()` and writes them out on pin A4 with `analogWrite()`. With scope channel 1 still monitoring your ECG at the ADC8282 output, use a second scope channel to look at pin A4.

Build and run the code, and you should now see your ECG twice – the original version on channel 1 and the digitized version on channel 2. Play with the scope to capture them both right on top of each other, so that you can compare them. Take a picture again with your phone.

Saving your ECG for future analysis

The next (and final) step is to save your digitized ECG for analysis over the next several weeks. We'll be using it as input for an algorithm that analyzes your ECG in real time and computes your beat-to-beat heart rate.

Take a look at the rest of the code; specifically, the code to save your ADC samples in a global array, print them out using a UART and blink the green LED for status.

Next, set up a serial terminal on the host with Moba and rerun the code.


You should get the prompt “type the letter 'g' to go”. When you're ready, do so. At that point, the green LED will turn on and you'll get 10 seconds of ECG sampling at 500 samples/second. You'll also see lots of numbers (your digitized ECG) appearing on the Moba terminal.

When the 10 seconds are over, the sampling stops. But since a 9600 baud UART typically cannot keep up with the 500 samples/second data, you're not done yet; and so the numbers will continue to stream across the terminal. To indicate this mode (printing but not sampling), the LED changes from solid green to 1Hz blinking.

Finally, when printing is done, the LED turns off. At this point, you can do a simple cut/paste from your serial terminal to save the text in your favorite editor. See the section on Moba below for details.

Once you save it to a text file on the host, be sure to put it somewhere you can reuse it for next week's lab too.

Capturing a diversity of signals

More signals, more fun :-).

There are several ways for you to capture multiple signal traces. First, we have two varieties of AD8232 boards. Most of the boards we have are the “regular” ones. However, there are also five “broadband” boards, marked by a small piece of a gray duct tape on the microphone jack. The regular boards have an onboard bandpass filter that only passes signals in the range .5 to 40 Hz; the broadband boards allow signals up to 150 Hz. You can collect traces with both boards if you like.

You can collect traces from multiple people in your group. Most college-age students have pretty similar ECGs, but who knows. You will likely at least have different heart rates. You can grab an ECG from a Designated Older Person if you ask nicely :-).

You could do a bunch of jumping jacks before (or during) your data collection, or even do a quick run around the block and then quickly record yourself.

Lab checkout

Show a TA your screen shot of both the ADC input and DAC output to be sure they look reasonable.

Also, if you haven’t already done so, look at the function `task_canned_ECG()`. This is the code to drive a sample ECG out to A3, either as a debugging aid or because you don’t want to divulge your own ECG. Explain how it works to a TA.

Using Moba

Moba is a PC-resident program to establish sessions. While we most often use it to establish remote-SSH sessions with a remote Linux host, it can also act as a serial terminal. To do set, click on Sessions/New Session and then choose Serial (the Serial icon resembles a plug and socket).

Choose your serial port. If the Nucleo board is already plugged into a host USB port, then Moba should be smart enough to only give you one option for your serial port, the one that the Nucleo is plugged into (typically COM3). Choose your serial speed as well, and you should be ready to go.

You can clear the screen with right-click/Clear Scrollback. You can put the entire screen into a copy buffer with right-click/Copy All, at which point you can paste the entire screen into your favorite Windows editor (e.g., Notepad).

What to turn in:

Save your digitized capture files in a location where you can easily find them. For your own sanity, each should be named in some useful manner. For example, the files collected with the broadband boards should have the suffix “_bb.”

Turn in your two screen shots as part of your lab report: first, of your ECG alone. Second, with the ADC-then-DAC version of it. Finally, turn in the answers to the following questions.

1. Look closely at your screen shot of both the original and the ADC-then-DAC waveforms. How closely do these match? If the match isn’t perfect, do you have any hypotheses for why not?
2. The code for this lab goes through a fairly elaborate process of using a global array to build a sort of FIFO, buffering the high-speed digitization of your ECG from the relatively-slow UART transmission to the host. It would be nice to remove all of this

code. How fast would the UART have to be in order to do this? Assume that each character is one byte with one stop bit and one parity bit, and ignore any other overhead in the UART transmission.