Lab #4: Semiconductor Devices for PPG circuit
ENGN/PHYS 207—Fall 2020

Prelude

The circuits you build today will be used for the final PPG build. In this light two things to consider:

1. Leave them on you breadboard–you're gonna continue to use these in a real life design!
2. As such, build and wire carefully. This is going to get a bit complicated
3. The Feather can and should be removed from your breadboard to free up a big chunk of circuits real estate. Use male-female quick connect wires instead.

Circuits You'll Build

1. Leaky Peak Detector
2. LEDs driven by MOSFET switches

Lab Skills You'll Learn

1. Working with Diodes (both silicon and LEDs)
2. Working with MOSFETs as digital switches
3. More Feather/Arduino coding!

1 When Diodes meet RC circuits

They got together, held circuits hands, and made a Leaky Peak Detector. (What circuits components do in the lab after dark is anyone's guess...) We'll build to the leaky peak detector his up pretty quickly, in 4 stages. The (almost) final result will be used to detect individual pulses in an electrical signal associated with the heartbeat. More on that in the weeks ahead. For now, let's get lab cookin'...

1.1 Home brew voltage source using potentiometer

For your voltage source in this experiment, you'll use a hand-turned potentiometer per Figure 1(a). Wiggling the pot knob back and forth creates a time varying voltage. This is really just thinly disguised voltage divider circuit. Turning the knob changes the ratio of $R1$ and $R2$ per Figure 1(b).

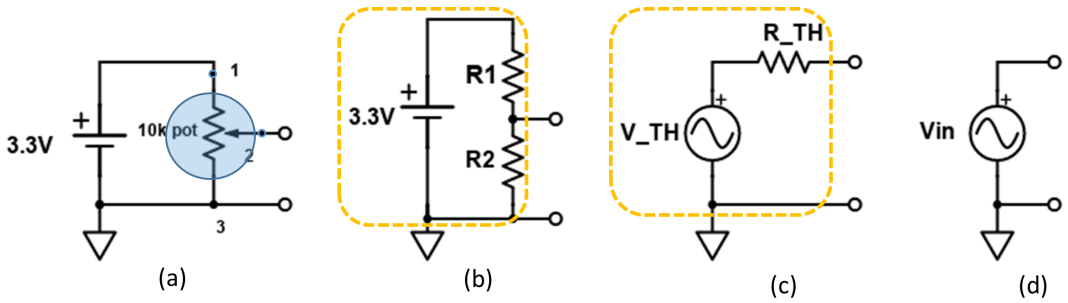

Figure 1: Making a time-varying voltage source based on hand-turned potentiometer. (a) Schematic of 3.3V source and 10 kΩ potentiometer. Note all 3 terminals of pot are important here. The output is measured across terminal 2 and 3 aka the resistance of the "bottom" part of the pot; (b) schematic showing pot drawn as a voltage divider. The "top" part of the pot is $R1$; the "bottom" is represented by $R2$. The dotted box box indicates what's "inside the box", emphasizing the two contact terminals serving as the output voltage (c) Thevenin equivalent circuit. Again, the dotted orange box serves to illustrate what's in the box and the two terminals to connect to the outside world (d) Ideal voltage source, which of course neglects the Thevenin source resistance R_{TH}.

Answer/compute the following:

1. The range of output voltage you can achieve with this configuration is: _____ Volts
2. Real voltage sources have real source impedance. This is modeled as R_{TH} in Figure 1(c). Of course, the lower the source resistance (impedance), the better—we don't want to drop voltage just getting out of the source and into the rest of the circuit. Given a 10kΩ pot, the range of the source impedance is: _____ kΩ.

1.2 "Leaky" Peak Detector Circuit with Diode and RC discharge

Let's build diode some circuits and see them in action!

1. ***Rectifier*** Build **Figure 2 (a)**. Use your "wiggly" function generator per Section 1.1. Using your Feather 32u4 and Serial Plotter functionality, make simultaneous measurements/visualization of the input and output signals. Recall the very helpful, spot-on Adafruit tutorial for Serial Plotting in the Arduino IDE (scroll toward bottom).

   To submit: Screenshot of representative data with 1-2 sentence caption explaining the input-output relationship noted.

2. Build **Figure 2 (b)**. As we discussed in class, this is a peak Detector that remembers the highest voltage it has ever seen forever (minus the diode drop). Similar to what you just did

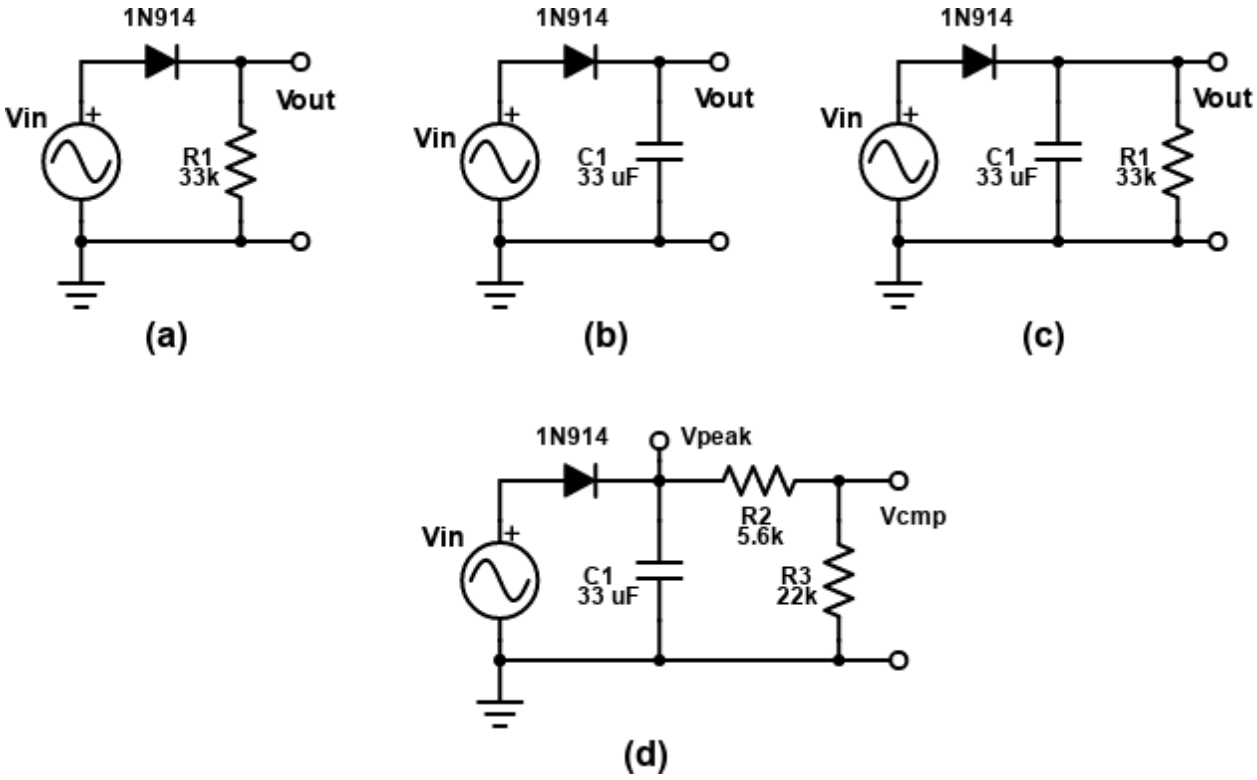

Figure 2: Diode circuits. (a) diode rectifier; (b) peak detector; (c) leaky peak detector aka RC smoother; (d) leaky peak detector with voltage divider. The "wiggly" input waveform is based on a hand-turned potentiometer circuit (see Figure 1.)

with the rectifier, make simultaneous plots of the input and output. The *delay()* function may be smartly used to control the rate at which Feather makes measurements and plots points.

Make sure you start with your capacitor discharged. You can easily do this by placing the cap in parallel with a small valued resistor (it discharges on a time scale of $\tau = RC$. The resistor safely dissipates the stored energy. Alternatively, but not recommended: You can also use just a bare wire, but generally ill-advised as sparks can fly.

To submit: Screenshot of representative data with 1-2 sentence caption explaining the input-output relationship noted.

3. Build **Figure 2 (c)**. You already know what is supposed to happen from our discussion in class. This is a leaky peak detector that "forgets" the highest voltage it has seen on a timescale of $\tau = RC$. You know what to do–same as above viewing/plotting input and output.

   To submit: 1) Screenshot of representative data with 1-2 sentence caption explaining the input-output relationship noted; 2) Computation of time constant τ, experimental estimate of discharge ("forgetting time") and comparison of these two values. (Do they reasonably agree?)

4. Build **Figure 2 (d)**. This time you need to make ***three*** simultaneous measurements: 1) input signal; 2) V_{peak} measured at node indicated relative to ground; 3) V_{cmp}, at node indicated.

Here, "cmp" is short hand for compare. We'll see soon, that this is the voltage against which another voltage is compared to identify individual heart beat waveforms in the final PPG circuit. *When you are done, leave this circuit intact–you'll use it for your final project!*

To submit: 1) Screenshot of representative data with 1-2 sentence caption explaining the relation ships noted between all 3 signals. 2) 1 sentence about the ratio you would expect to see for $\frac{V_{cmp}}{V_{peak}}$ (compute it theoretically!) compared to what you actually observed. Does it make sense?

2 MOSFETS and LEDs

Welcome to the circuits acronym buffet, where you are always welcome to come back for seconds!

Here we are going to build MOSFET drivers for LEDs. As we discussed in class, the reason for doing so is that more current can flow through the LED, brightly illuminating them. Before we start, we need to quickly substantiate this claim. Take a look at the pinout for the Feather 32u4 board. Look for the "3V3 output from regulator" designation.

To submit:

1. The max current that can be supplied by the Feather 32u4 "3V" pin is _____ mA.
2. The max current that can typically be supplied by a digital output pin, e.g D11, is _____ mA.
3. In view of Figure 3, left panel, the current I expect to flow through the red LED is computed to be _____ mA, and for the IR LED is computed to be _____ mA. (Hint: In the "on" state, the MOSFET acts like a $\approx 5\ \Omega$ resistor, specified as "$R_{DS}(ON)$" in the data sheet.)

Now let's get building real circuits. Build the MOSFET-switched IR and red LED circuits in Figure 3. Here, the MOSFET acts as a digital switch. If the digital output driving the gate is high (3.3V), then the MOSFET turns "on" and the LED is illuminates. If the digital output is low (0V), then the MOSFET turns "off" and so does the LED.

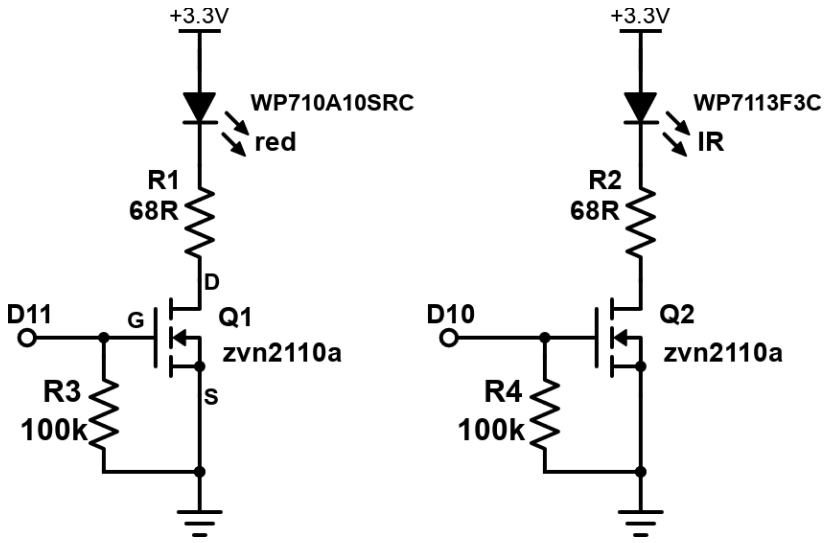

Figure 3: Red (left) and IR (right) LEDs switches by MOSFETs driven digital output of Feather board. The figure shows the MOSFET gate driven by D10 and D11, but you can choose any two you like. The value of the ***pull down resistors*** R_3 and R_4 is not critical for this application, and 100kΩ should work well.

The key here will be to sequentially time the LEDs turning on and off. For the PPG final circuits, we'll turn one on at a time.

1. Set both of your digital outputs driving the MOSFET gates to blink on some human timescale, e.g on 1 s, off 1 s.

2. For the PPG circuit, we'll need a bit more sophisticated timing of the LEDs successively turning on and off, per Figure 4. Program your Feather 32u4 to do exactly this timing sequence, first turning on the red LED, then the IR led, followed by a relatively long pause. Note the time between successive samples for each red and IR light is 2000 μs = 2 ms. The samples of red and IR light are offset by a mere 320μs. That's a very fast time-scale relative to heart beats and blood flow. So the blood flow scene is essentially constant during this short time interval.

   Coding tip: You will almost certainly want to make use of the *delayMicroseconds()* function to appropriately set such fast timings!

   To confirm your timings, use a Picoscope or Lecroy oscillscope making simultaneous measurements of

   (a) The digital output pin voltages **Submit screenshot to prove it works**! Annotate the timing on your screenshot and compare to the desired timing indicated in Figure 4.

   (b) The voltages at the node of the LED and resistor for both red and IR. You should see an ***inverted logic*** here: When current flows, the voltage you measure will actually decrease (input signal "high", output signal "low"). **Submit a screenshot to prove it works**!
   The reason for the inverted logic is a bit subtle/tricky—inverting logic often is. IN this case, it fundamentally derives from the IV characteristic curve of a diode. To help you think through the reason for inverted logic: How much voltage drops across a diode with subsatantial current flowing vs. little to no current flowing? If you start at 3.3 V at the "top" of the circuit, how much voltage remains by the time you get to the node where you are measuring? (Think KVL here)

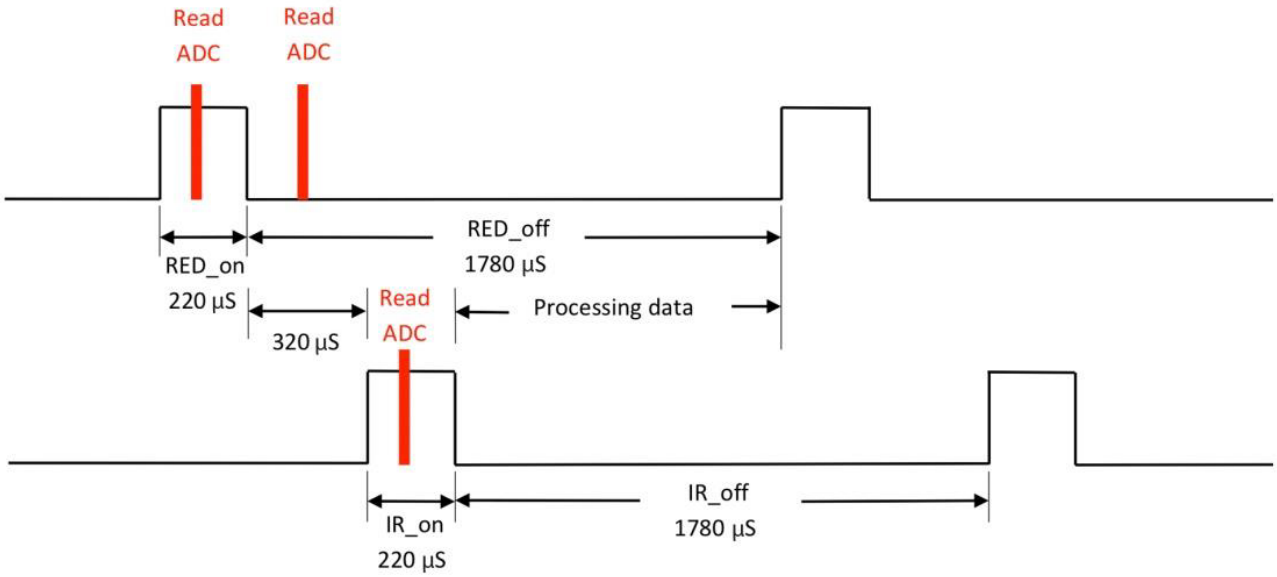

Figure 4: Timing of LEDs illuminating. Image from Zhang Feng, Microchip Technology Inc. "Measuring Heart Rate and Blood Oxygen Levels for Portable and Wearable Devices".

3 What to Turn In

Please submit what is asked throughout the lab report. In general, look for the "To submit" tags. ***Please include section headers*** to organize your submission and make it readable. Basically, you just need to fill in the blanks and submit screenshots/figures with a brief caption. Essentially, there is no real writing—the figures do the talking here.