Pulse Oximeter

First build a hysteric oscillator circuit shown in Figure 1 (just the circuit on the left). We discussed this circuit in class. Build the circuit and test that it works by hooking the oscillator output (wire that says square wave) into one of your DAQs analog inputs. The output square wave should be from 0 to 5 volts at about 75 Hz. You do not need to record any data for your lab report; just confirm that it is working before moving forward.

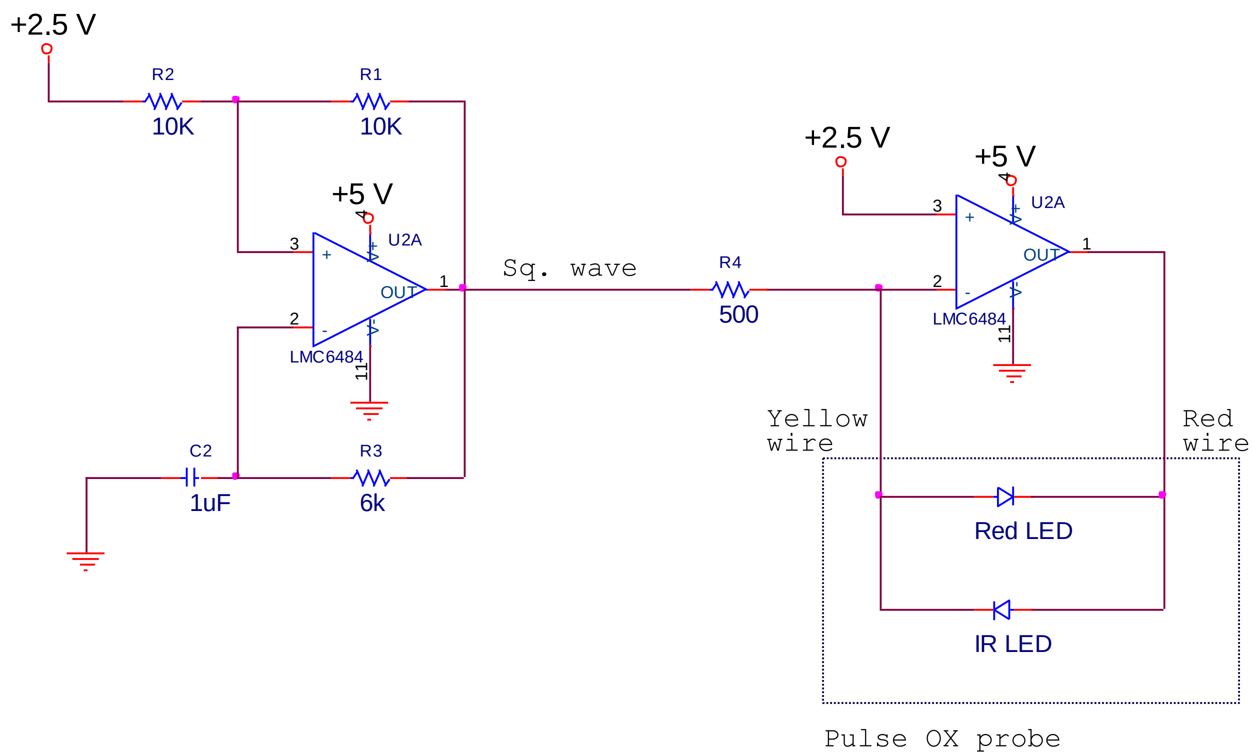

Figure 1: On the left is the hysteric oscillator circuit. The output of the circuit is a square wave between 0 and 5 V. The period of oscillation is 2.2RC, where the RC used in the calculation is the 1 microfarad capacitor and 6K resistor. The frequency is about 75 Hz. On the right is the LED circuit. This circuit drives the LEDs in the pulse oximeter probe. The LED inside the probe is actually two LEDs with a single pair of wires. The LEDs are switched in direction as shown in the diagram. Both circuits can be put on one quad LMC6484.

Once you have built the oscillator and have confirmed that it is working, build the LED driver shown in Figure 1 on the right. Once this circuit is built the red LED should be on. Since it is blinking faster than your eye can sense, it will appear on. You can confirm the IR LED is working by looking at it with a cell phone camera, though it may be hard to see with the red on as well. You can change the 500 ohm resistor if needed. A smaller resistor will result in a brighter light, though the current limit on the op-amp won't allow you to get too much brighter than you are at 500 ohms.

Now build the detector circuit. The black wires which come from the probe are attached to the cable shielding; these two black wires should be grounded to your breadboard. Once you have the detector built, modify one of your MATLAB programs to acquire data from the circuit at 20,000 samples per second. Use one of the analog input channels to measure the voltage across the 500 ohm resistor in the driving circuit. The voltage across the 500 ohm resistor will let you know the current going through the LEDs. Use another channel to measure the output of the detector circuit with respect to ground. First test the detector by aiming the photodiode up at the room lights, point it towards and away from the blinking LED. You should get an output signal that makes some sense.

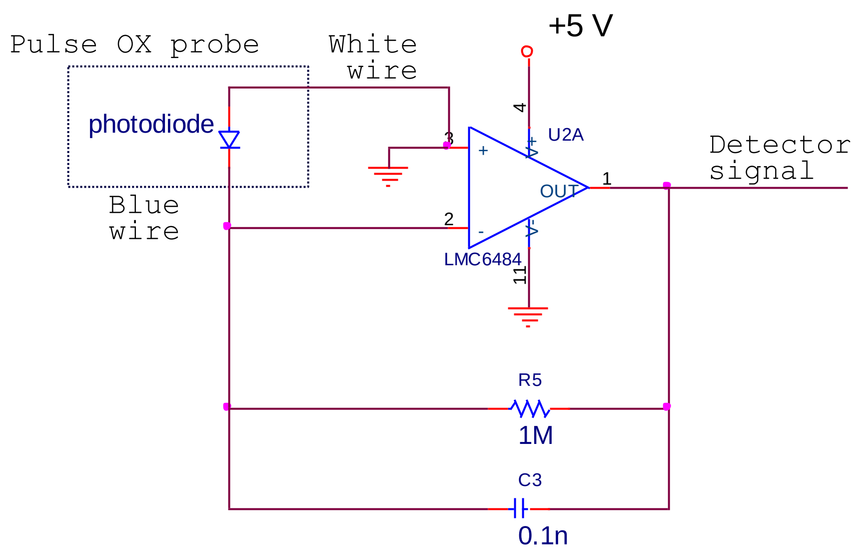

Figure 2: Photodetector circuit. The capacitor acts as a low pass filter to smooth the output just a little bit. You can remove the capacitor if you want just to see the difference. It just cleans up the detector output from having a lot of spikes at each transition of the square wave.

Once everything seems to be working, then attach the pulse oximeter probe to your finger. You can use a little tape to hold it in place. It seems to work best if you don't attach it too tight and if the LED goes through the fleshy part of the finger and the sensor goes over the nail. You can try different things. Collect about 10 seconds of data so you can make sure you get a reasonable looking signal. You should be able to see your pulse in the raw detector data as a slight modulation of the maximum in the detector output.

Some sample data is shown in Figure 3. The upper figure shows the data on a 10 second time scale where we can see the pulse as the ~1 Hz "wave" on the upper envelope of the data. If we zoom into the figure in time (the lower figure) we can see the signature of the square wave driving. Once you get good data, save it to a file. If you have trouble getting a good signal you can try acquiring your data with the room lights off.

Once you believe everything is working, you can try a long test holding your breath. Don't push yourself too hard, we don't want anyone to pass out in the lab. You will get better data if you hold still. As you record the data with you holding your breath, note the time in the data when you start to feel uncomfortable and the time when you take a breath. If you want to use one of the course instructors as a test subject, just ask! Once the test is over, SAVE THE DATA with the "save" command.

Once you have recorded the data, then you will need to write a MATLAB program to process the data. The first command of your processing program should "load" the save data into the MATLAB workspace. You can look through the class notes from Wednesday, but to process the data you need to:

1. Convert the current through the LEDs to a ±1 signal to denote which LED was on.
2. Multiply the raw data from the detector by the ±1 signal.
3. Put this result through a software low-pass filter (you should write a function to do this in MATLAB). A 2 Hz time constant seemed to work well as a first cut. We wrote this function in class on Wed., hopefully you took good notes!
4. Low pass the data again. Maybe a third time? You can play with time constant for the filter and such to get a result that you like. A "good" result should clearly show the pulse. You should also be able to see the signal "crash" when you hold your breath around the time that you started to feel uncomfortable and it should recover when you take a breath.

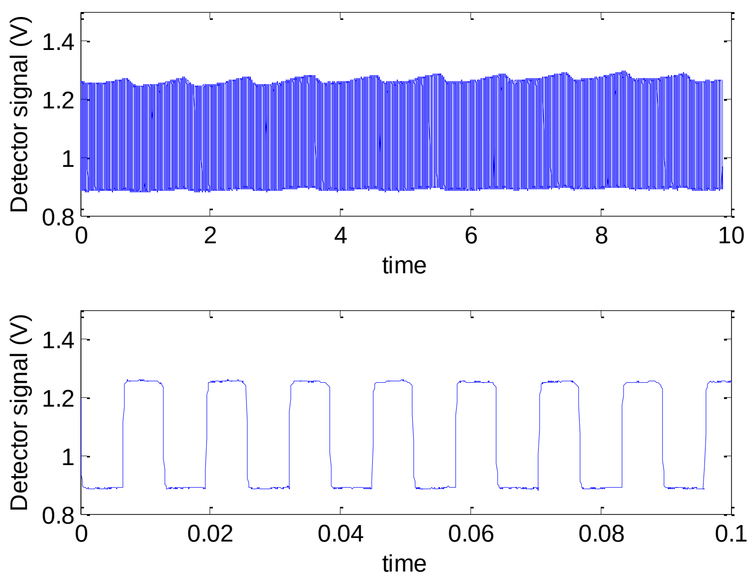

Figure 3: Sample output. Same data, only the time scale is different in the two plots. On the long time scale you should be able to see your pulse. On the shorter scale, you can see the on-off behavior of the LED.

Deliverables

1) Show a plot of raw detector data versus time. Show one plot for a time scale of about 10 seconds where we can clearly see a pulse and another for a longer breath test where we can see change as you hold your breath.
2) Show a plot of the processed data for the two cases described above in 1. Explain exactly how you filtered the data (i.e. time constants, how many times, etc.)
3) Include your code for processing (but not acquiring the data). Your code should be commented.

Grading

1) Deliverable 1: 3 points total for everything done right. 1point off for each sloppy/unlabeled graph, 2 points off for incorrect/poor data.
2) Deliverable 2: 5 points total for everything done right. 1 point off for sloppy/unlabeled graphs, 2 points off if processed data does not look correct (i.e. cannot see pulse). 3 points off if it appears completely bogus but you tried something. 1 point off if explanation of processing technique is wrong, unclear, or has many errors.
3) Deliverable 3: 2 points total. Full credit for correct and well commented code. 1 point off if it is correct but your code is not commented or very sloppy.

Ninjas, as always, will use their best judgment to follow these guidelines, but can award partial credit or deduct for other significant errors at their discretion.