

PPG Code - Quick Start Manual

(JE 10 Nov 2020 – Circuits fall 2020)

This brief manual will hopefully get you up and running with the Adafruit feather 32u4 interfaced to your PPG circuit in 10 min or less (money back guarantee! Act now, and we'll throw in a pasta strainer, a food dehydrator and more!...Ok back to serious circuits land).

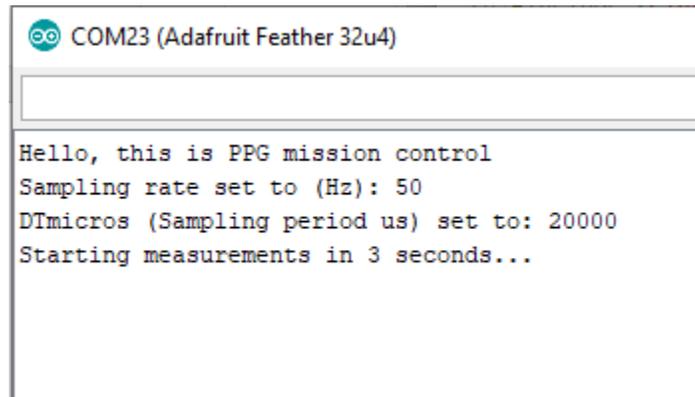
Programming your Feather board

1. Code is available [in this box folder](#) (PPGcontrol_circuits_f2020).
2. Note there are two files, both must in the same folder. These files are named; **PPGcontrol_circuits_f2020.ino** and **Pin_Timing_Defs.h**
3. The latter contains definitions for Input/Output (I/O) pins. **Update pin numbers** as needed. Hopefully the variable naming and comments make obvious what each variable and pin number combo does. One that merits special attention is the **DigInPin**. This is the Feather pin where you connect the output of your comparator that generates brief 3.3V square pulse, presumably on each pulse (heart beat).
4. Similarly, **update timing variables**:
 - a. **FS** (Hz): sampling frequency. 50 Hz is probably a good ballpark number.
 - b. **Trir** (s): period for which Red light stays illuminated then IR light. For example, a value of 5 (units = sec) means the red stays illuminated 5 s with IR off; then the lights toggle for the next 5 sec with IR on and red off. Choose a value that is sufficiently long to capture several successive pulses with a single light source. This is needed to accurately determine the AC and DC components of the waveform to compute the ratio rho, to ultimately compute Sp02.
 - c. **Twait** (ms). This pauses the system for just a brief moment in time immediately after toggling LEDs. The idea is that the circuitry may need to stabilize as it switches illumination sources. 20 ms is probably long enough. Recommend not adjusting for now.
5. **Compile and upload per usual**.

Usage

1. Your Feather will just be sitting there waiting for you to open the Serial Monitor. It won't do until the serial port connection is established (because that's what code line 66 says to do).
2. Upon opening the serial monitor, you should see a friendly message informing you of the system settings, namely the sampling frequency and corresponding sample period.

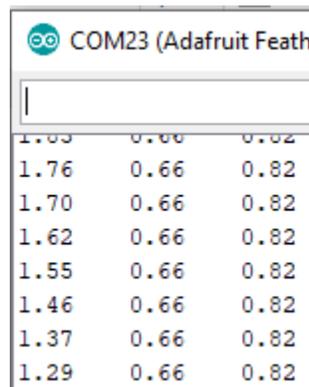
The system is programmed to pause 3 seconds before launching into making measurements.



```
COM23 (Adafruit Feather 32u4)
Hello, this is PPG mission control
Sampling rate set to (Hz): 50
DTmicros (Sampling period us) set to: 20000
Starting measurements in 3 seconds...
```

Figure 1. Friendly message from the PPG system upon startup.

3. Viewing data in serial monitor. If you leave the serial monitor open, you should see a stream of numbers appear in 3 columns. In order these are:
 - a. voltage reading based on analog input reading
 - b. Cardiac interbeat interval (units of s). For example 0.66 means there were 0.66 s = 600 ms between the last 2 successive beats. This corresponds to a heart rate of $1000/600 * 60 \text{ bpm} = 100 \text{ bpm}$. You can of course adjust your code to display the heart rate (bpm) instead of interbeat interval. I chose not to here, because I wanted to have all metrics fit on a nice display (see below)
 - c. SpO2 on a scale of 0-1. Basically dividing % saturation by 100. Here 1 = 100% blood oxygen saturation; 0.82= 82% for example (someone call a doctor!); and 0% means nary an oxygen molecule bound (that seems bad).



1.76	0.66	0.82
1.70	0.66	0.82
1.62	0.66	0.82
1.55	0.66	0.82
1.46	0.66	0.82
1.37	0.66	0.82
1.29	0.66	0.82

Figure 2. Serial monitor data display. Columns L to R are: voltage reading; cardiac interbeat interval (sec); and SpO2/100.

4. Viewing data in serial plotter. You can view the data in graphical format instead, per example below. The waveforms are as follows:
 - a. Blue = voltage reading. Here I'm using a function generator set to 2 Hz, 0.7V offset, and 0.5V amplitude to generate a PPG-like signal. Hopefully, you'll measure a similar-ish signal with your actual PPG system (not JE's function generator cop out...hey, gotta test the system somehow!)
 - b. Red = interbeat interval (described above). Here we can see the variation in timing beat-to-beat.
 - c. Green = SpO2 (0-1) scale. Again, I just divided out by 100 to have all 3 signals on a similar magnitude scale. You can adjust to taste, but this didn't seem too bad (I hope!).

Overall, this is just a quick visualization of the 'final result'. Admittedly, it is a bit difficult to read off physiologically relevant information in the red and green trace, so maybe the serial monitor is nicer in that regard. Meanwhile, the serial plotter does seem favorable for visualizing the PPG signal the Feather is recording in the first place.

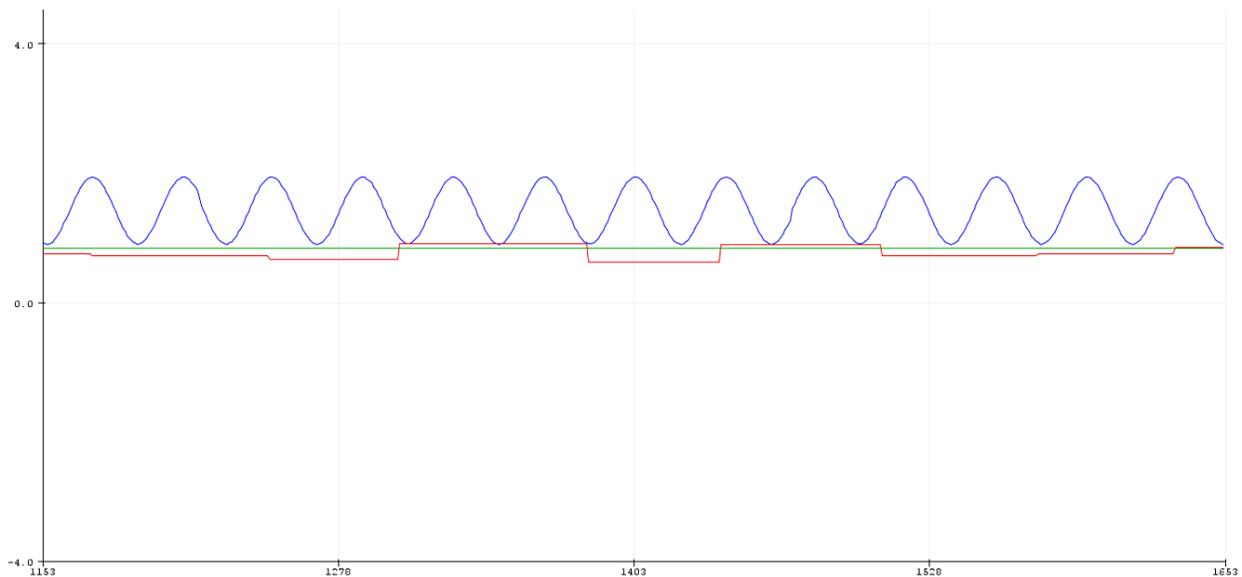


Figure 3. Serial monitor screenshot. See text above for description of blue, red, and green waveforms.