Lab Report 2

Prepared by: Heidi Eren

Date: 10/19/2025

Executive summary (**5 pts**):

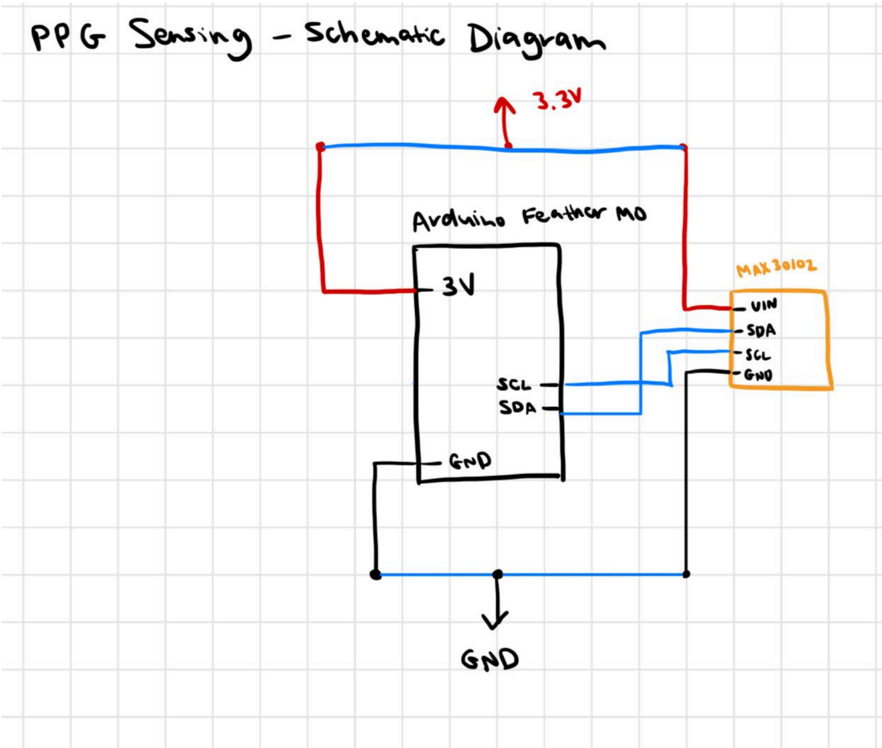
Describe what you are trying to accomplish/demonstrate. What is the purpose of this test?

The purpose of this experiment is to demonstrate an understanding of photoplethysmography (PPG) sensors to measure blood oxygenation, pulse rate (heart rate), and respiratory rate. The red and infrared lights can be measured to estimate the hemoglobin oxygen saturation of arterial blood, represented as an AC signal that is superimposed on a DC signal. By taking the derivative and thresholding, SPO2 and heart rate can be calculated for event detection. By removing the DC component from the infrared signal and obtaining the inter-beat interval through the lag (shift), the autocorrelation can be calculated. Lastly, the Fourier transform can transform the infrared signal into a frequency domain to detect peaks, resulting in another method of computing heart rate. When comparing the library-computed heart rate, the rapid response, noise, and accuracy of the manually calculated heart rates can be evaluated.

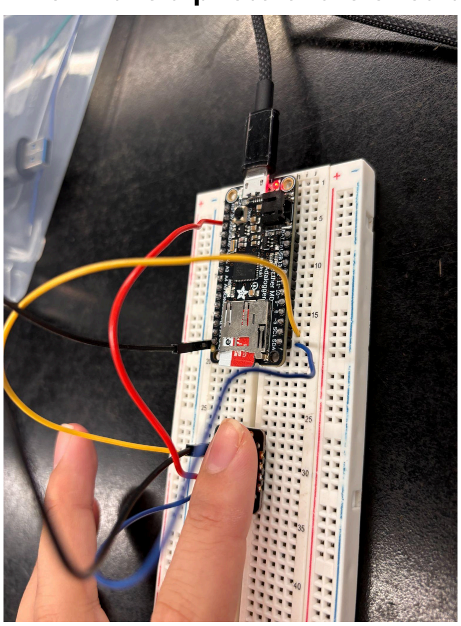
Procedure:

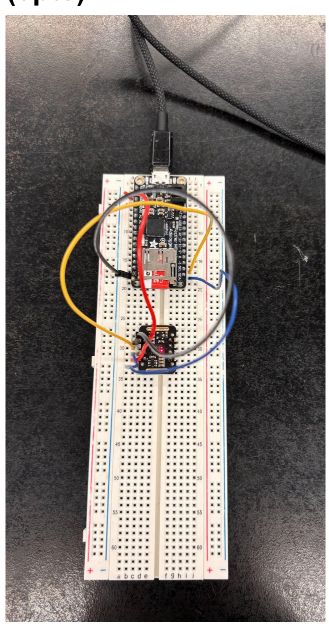
1. Build a circuit to measure pulse rate and SpO_2. Use an Arduino for data acquisition.

2. **Sketch a circuit diagram of the circuit (5pts):**

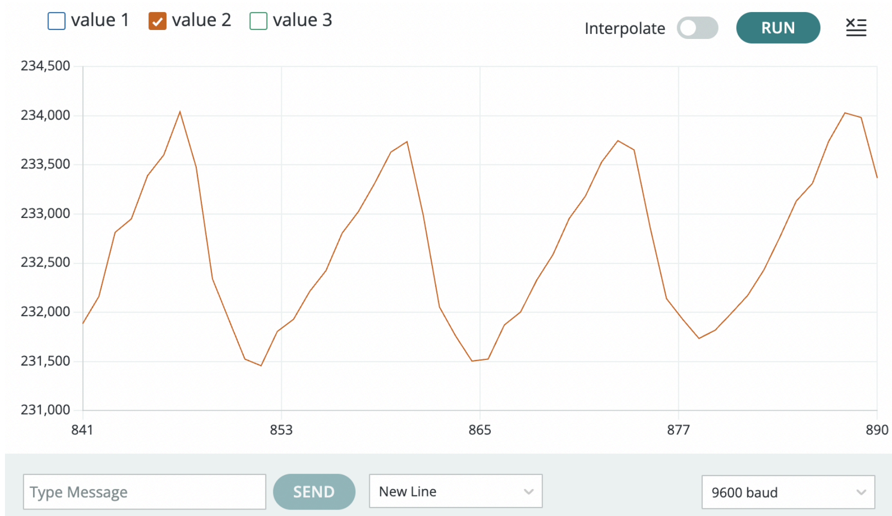


3. **Take a photo of the circuit (3pts):**





4. Confirm that you can see periodic changes in signal that roughly correspond to HR



5. In the Arduino IDE, calculate the AC and DC components of the IR and red signals and confirm that your detector matches what you see in the data (**You will upload your final code to Canvas**).

6. In the Arduino IDE, add the SpO_2 equation. Confirm that your calculated SpO_2 matches the sensor output (**You will upload your final code to Canvas**).

7. In the Arduino IDE, identify the "events" and calculate pulse rate. Compare against the pulse rate measured using the provided library.

8. **Bonus (5pts - in your code submitted to Canvas):** calculate heart rate variability.

9. Calculate the FFT of the PPG signal. Identify peak associated with pulse rate. Compare against the pulse rate measured using the provided library.

10. Use autocorrelation to calculate pulse rate. Compare against the pulse rate measured using the provided library.

11. Record the values of all the calculations of pulse rate, each calculated using event detection, FFT, and autocorrelation) in a .txt file **to be submitted with this report (5 pts for each)**.

Notes and recommendations:

1. Describe the circuit you used to measure PPG. How does the sensor communicate with the microcontroller (2 pts)?

   The circuit used for measuring PPG consists of the PPG sensor (MAX30102 pulse oximeter) that connects to the Adafruit Feather microcontroller. The pulse oximeter uses a photo detector to measure the light absorbed from the pulsating (arterial) blood by emitting two wavelengths of light (red and infrared). The PPG sensor communicates with the Adafruit Feather microcontroller with I2C, which only uses two pins, the serial clock (SCL) to generate the timing of the data bits and serial data (SDA) to send and receive the data. The microcontroller sends a 7-bit address of the sensor and then a 1-bit address for the read or write operation. 0 indicates to write, while 1 indicates to read. Then, the microcontroller recognizes this address and responds with ACK, short for acknowledge. Then it sends a repeated 7-bit address starting with a read bit, along with a 1-bit address read/write and response with ACK by sending the stop condition. Hence, the microcontroller reads the infrared and red signals, which then the PPG sensor digitizes and stores them in FIFO registers. This goes back to the microcontroller to retrieve the samples and processes them as heart rate and SPO2.

2. Discuss the different possible methods to calculate PR from a **raw data signal.**
   a. What are the advantages and disadvantages of each (which is the most straightforward from a signal processing perspective, which is the easiest to implement, which is most robust against noise, etc.. Include any issues you had in collecting your own data) (12pts - 4 each for each technique)?

   Event detection

   The event detection algorithm uses the IR thresholding and the derivative to detect the peaks as a pulse rate. The advantages of this algorithm include rapid response, ease of implementation (especially when the baseline is variable or the event is small), and its ability to calculate other parameters such as HRV (heart rate variability). HRV can be computed by taking the moving average of the inter-beat interval (IBI) and then calculating its squared root of the variance, which is the standard deviation. However, the derivative is prone to noise through false positives or false “events” where there are multiple peaks, which can lead to inaccurate detections of peaks which affects the calculations for pulse rate. One significant issue when collecting data for event detection is obtaining the correct threshold values for observed infrared light and its derivative. With large fluctuations in the readings, it was difficult to determine the optimal threshold for each parameter, as a value outside the threshold would not calculate a pulse rate at all. The rapid response made it easier to troubleshoot such issues.

   Autocorrelation

Using autocorrelation, the method removes the DC component from the IR signal then compares the signal to itself to detect the peaks and their corresponding locations. This method is advantageous in noise reduction and accuracy because each signal is compared with its delayed version, which amplifies the detection in a period more accurately. Because autocorrelation excels in detecting heart rate, it can be used for diagnosis. However, autocorrelation is not ideal for other parameters such as SPO2 and HRV since the amplitude or shape of a peak is not captured, only the periodicity and frequency. When measuring heart rate using autocorrelation, one issue I encountered was the detection of false events or peaks. Because the IR signal was being observed in the script, there were times when the peaks had multiple peaks, or jittered around the actual peak location, due to the nature of the PPG sensors. It was difficult to ensure that the true peak location was detected, so troubleshooting included the implementation of the next best peak if the current peak was not accurate (in the calculated heart rate). This ensured that calculated heart rates that were outside the true range were not outputted, and the peak locations were adjusted accordingly.

FFT

Using FFT, the method removes the DC component from the IR signal, then uses the Fourier transform to measure the strength of different frequencies of the peaks (in the frequency domain) in order to calculate the heart rate. After inputting the optimal windowing and sampling frequency, the FFT library makes computationally complex calculations to convert the frequency into the heart rate. FFT is advantageous in being used for other parameters such as HRV and respiratory rate. The frequency bands from FFT conveys variation in heart rate, which directly correlates to HRV (measure of variability frequencies in IBI) and respiration (frequency peak in high frequency band). When clean, stationary signals are detected, the computations from FFT produce relatively accurate data with ample noise rejection. However, FFT has limitations in windowing and speed. A stable, stationary window is required to produce accurate readings, which can range from 30 seconds to 5 minutes. As a result, the response time is slower as it needs a full window to proceed in addition to its processing time and integration. Because FFT computes the IR signal over the entire window (sampling size), if there were a sudden change in heart rate, it would be difficult to observe when such a change occurred. The frequency is computed as the average over a window, so if a heart rate experiences rapid variation or irregularity, FFT smooths the data over. This became an issue when collecting data. Making sure that the IR signal readings were reliable and consistent by minimizing all movements in the finger was challenging, and coupled with the easily detected noise from the PPG sensor itself, the fluctuations in the calculated heart rate were inevitable.

b. Which one should be preferred for a continuous monitoring device (2 pts)? Which one should be preferred for an intermittent monitoring device (2 pts)?

Event detection should be preferred for a continuous monitoring device because of its rapid (instantaneous) response rate with every heart beat and minimal complexity and power in calculating inter-beat intervals (for HRV). This makes peak detection easy to measure continuously and in real-time at every beat—capturing any sudden variation or irregularity—making it ideal for continuous monitoring devices.

Autocorrelation should be preferred for an intermittent monitoring device because its shorter periodic readings require less computational time and power for processing than FFT, producing stable and clean heart rate detection. Instead of beat-to-beat detection, a narrow window can ensure a clean and robust estimate of the heart rate within seconds. In addition, its resistance to noise makes autocorrelation suitable for short, reliable heart rate detection through intermittent monitoring.

c. Try to imagine edge cases where each might break down (2 pts).

Event detection has many edge cases. Any small movements in the finger can cause false peaks or missing heart beats, which can interfere with the threshold for the IR signal and its derivative. A heart beat may not be detected at all or can jitter in the peaks, resulting in inaccurate heart rate calculations.

The autocorrelation algorithm can break when there is irregularity in the heart rate or too short of a window to capture peaks. Autocorrelation depends on stable periodicity, so if there is a sudden change in heart rate, the frequency and amplitude would be affected. If the window is too short, there may not be sufficient cycles for true peaks, resulting in inaccurate peak detection. Autocorrelation may be detected across two windows, so the peak may be more ambiguous as well.

For FFT, the algorithm can break with a narrow window or when there are sudden changes in heart rate. There may be multiple peaks or jitters within a window, which affects the resulting frequency magnitudes of the peaks and leads to inaccurate heart rate calculations. A short window can result in inaccurate frequency resolutions since it can cut off a peak abruptly and capture significant fluctuations within the window rather than the true peaks.

3. Describe any other challenges or tricks that you want to remember (2 pts).

I think the most important trick to remember when debugging these scripts is to serial print each variable as it is being calculated or measured. By evaluating what values are being inputted or measured at each step, I had a better understanding of what changes needed to be made.

For the event detection script, a significant amount of troubleshooting was required to account for all the noise and variability in the PPG sensor readings. When out-of-range readings were observed, I had to implement if statements to avoid calculating the heart rate at these inaccurate peak locations. For example, if the detected heart rate from the inter-beat interval was too high or too small for a true heart rate, I excluded it from the final average heart rate. That way, the averaged heart rate values would not include such inaccurate readings in its range and output more reliable values. I also implemented an if statement for the interbeat interval because initially, the interval (from when the IR derivative changes sign) was too narrow for a true peak to be detected, or the window detected multiple peaks consecutively. By filtering the IBI to be within a specific range, this ensured that double detection of a heart beat would be excluded prior to any calculations.