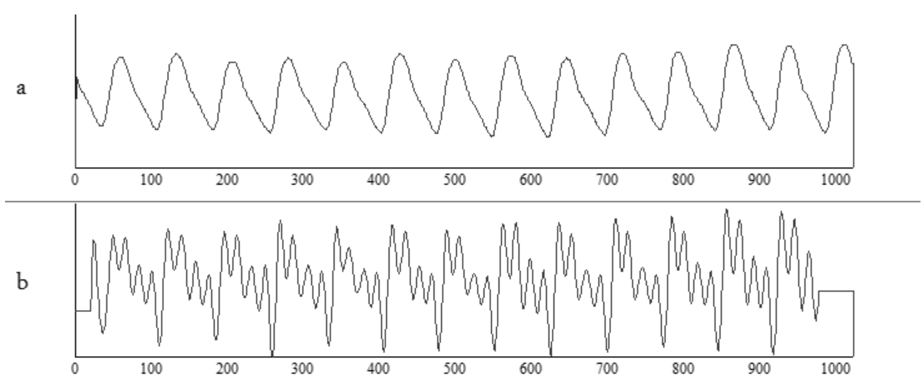
Work Plan: Pulse Oximeter

Matthew Mueller
James Pate

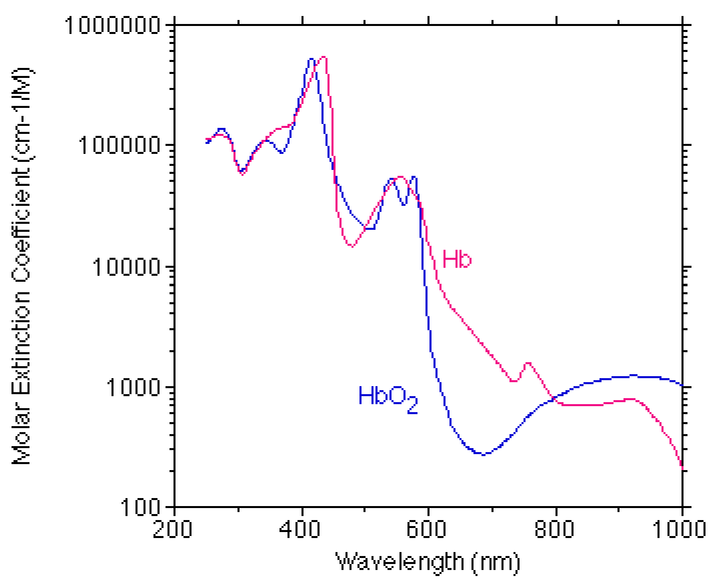
Introduction

Pulse oximetry is the process of noninvasively measuring an individual’s oxygen saturation, or the ratio of oxygen saturated hemoglobin to total hemoglobin. Our pulse oximeter will do transmissive pulse oximetry, which involves attaching a sensor to a thin part of the body and passing light through to a photodetector on the other side. The absorbance of the measurement location is determined by using the photodetector to measure the amount of transmitted light that made it through the measurement location. The absorbance of two different wavelengths is measured over time to create a photoplethysmogram. The collected data can be used to determine the oxygen saturation and other other metrics such as heart rate and blood pressure.

Background



In a red blood cell the hemoglobin molecule has two states: when it is carrying an oxygen molecule(O2), and when it is carrying carbon dioxide (CO2). The amount of light that is absorbed by hemoglobin depends on this state, and they have very distinct absorbance values when the light wavelength is around 600 to 1000 nm, which is crucial to detecting how much oxygen is in the blood. This is also why oxygenated blood is red, while deoxygenated blood is maroon. The LEDs we use to illuminate the blood fall within this range, a red LED with a wavelength around 650 nm, and an infrared LED with a wavelength of around 850 nm. According to the chart of molar extinction on the right (1),



at around 650 nm, the absorption of light for deoxygenated blood is about 10 times more than oxygenated blood, and at 850 nm, the oxygenated blood is about 2 to 3 times more than deoxygenated blood.

A difficulty with this process that we foresee as being an issue is that the skin tone can be an issue with the light as lighter and darker skin may affect the readings by the potential of a couple of percentage points which could be crucial for medical determinations. The light sensor can collect data about the amount of light that is allowed to pass through the finger at each of these wavelengths and estimate the ratio of oxygenated to non-oxygenated hemoglobin molecules in the blood. The absorbance of the finger will also vary based on the user's cardiac cycle. As blood is pushed out into the extremity, the absorbance will increase. This can be used to determine the user's heart rate. The graph we should get when plotting these values should be a plethysmogram, or more specifically since we are using light to do it, a photoplethysmogram. This graph represents the flow of blood in arteries and veins as new oxygenated blood is pumped in and is slowly depleted of its oxygen in the finger. This graph visually looks similar to a sawtooth wave, except not as sharp. Additional processing may be required for smoothing this data, see the pulse wave analysis paper below (4). This paper describes some additional processing that they did to view each of the chambers of the heart's pulses but for our purposes we only need the regular one that shows the overall pulse to determine the beats per minute.

Additional Information:

1. Optical Absorption of Hemoglobin
2. Tabulated Molar Extinction Coefficient for Hemoglobin in Water
3. Pulse oximetry: fundamentals and technology update - PMC
4. (PDF) Pulse Wave Shape Analysis of the Cardiovascular System Using High Signal Resolution

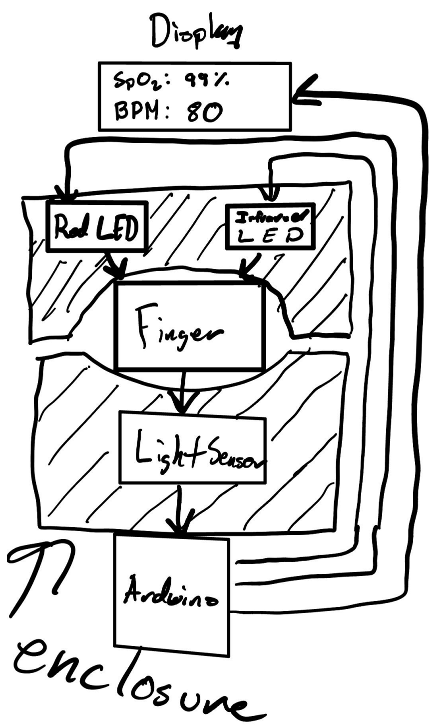
Design

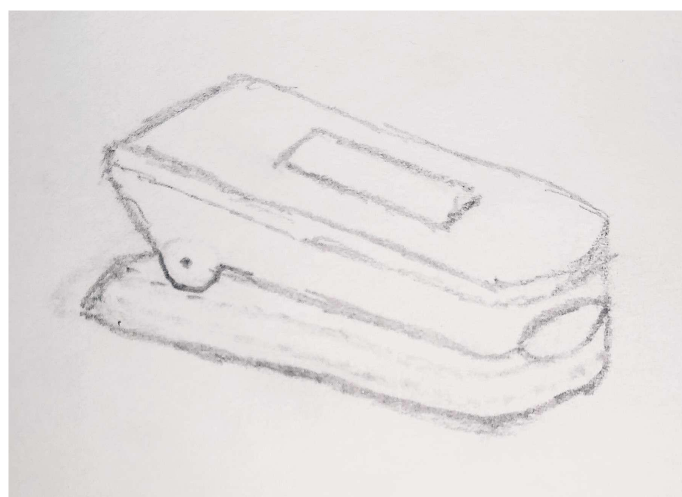
Hardware Design

Overview

The pulse oximeter will be a device that encloses or clamps onto the user's finger and sends light through the finger from one side of the enclosure to a sensor on the opposite side. The primary design requirements will be positioning the lights and sensors so that they are able to effectively send and receive the pulses of light through the user's finger, ensuring that the enclosure stays firmly attached to the finger to avoid any variation in the data based on movement, and blocking any external sources of light from interfering with the sensor readings. Some secondary design considerations will be ensuring that the enclosure is comfortable to wear for an extended period of time and aesthetically pleasing.

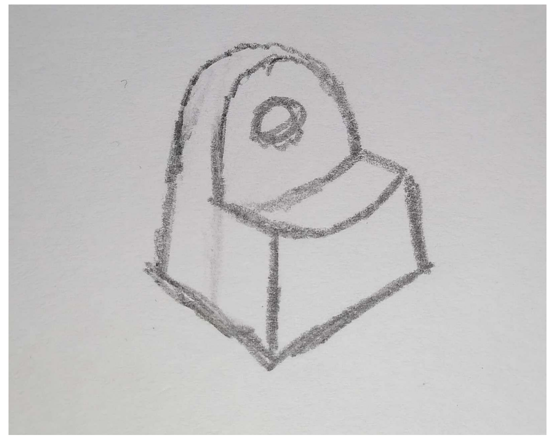
We have yet to determine the exact dimensions of the enclosure, but it will need to be large enough to accommodate a reasonable range of finger sizes and contain all of the hardware components. The display and light sensor will both connect to the arduino using the I2C protocol and can communicate along the same ports on the arduino as long as they have different I2C bus addresses. The LEDs will need to be connected to separate digital output pins on the arduino. Because the LEDs and display are both on opposite sides of the enclosure from the arduino, wires will need to be routed between the two sides. The most likely path for them between the hinge and where the end of the user's finger will rest. Below are two sketches that show the positioning of the different components within the enclosure and what the device may look like from an external view.





Hinge

The design of the hinge will be an important aspect of the enclosure for multiple reasons. It will ensure that the enclosure fits securely onto the user's finger and will move around as little as possible while taking measurements, needs to be durable, may have to be designed to help block light from reaching the interior of the enclosure, and must be 3D printable.



We've considered several different designs for the hinge. The most straightforward design is shown in a sketch to the right. It will be a simple barrel hinge that connects on the outside edge of the enclosure with the hinge pin running through the middle. There are several different possible variations on this design that depend on factors such as the type of spring used to create tension to close the device and whether the hinge needs to block out light. If the inner barrel of the hinge opens up in the middle, a torsion spring could be attached to the hinge pin and used to hold the device closed. This design would be simple, but would not allow the hinge to block light. The inner barrel could also extend across the entire length of the interior and a coil spring could be placed beyond the hinge to push the device closed.

Another possible design for the hinge would be to design the upper and lower portion of the enclosure to be a single compliant mechanism. This would remove the requirement for some type of spring, as the enclosure would create the required tension on its own, and would reduce the number of separate components in the design. This design does have some drawbacks though: it would require much more time to design and multiple print attempts and would probably not be able to block out any light.

Layers

The upper portion of the enclosure should require only two separate 3D printed layers. The inner layer would connect to the lower portion of the enclosure with a hinge and contain two appropriately sized holes for the LEDs to be installed in. This layer should also have standoffs with screw holes that the display can be attached to to keep it separate from the LEDs and provide space for the wires. The second layer of the upper part of the enclosure should act as a faceplate to conceal and protect the electronic components.

The lower portion may be designed with two or three layers. The inner layer will have a hole to expose the light sensor to the LEDs and holes or standoffs to mount the sensor to. Next, either longer standoffs or an additional layer to mount and separate the arduino from the sensor. The final layer will be the bottom faceplate to protect the electronics. These layers will need additional wire pass throughs in order to connect the components to the microcontroller and connect the microcontroller to a power source.

The final potential layer for the pulse oximeter will be a possible foam layer between the upper and lower portions of the device. This foam would serve to make the device more comfortable to wear, prevent the finger from shifting around, and to block light from entering the device and interfering with the sensor. If this layer is included it would remove the burden of blocking light from entering the device from other components like the hinge and possibly simplify the design.

Data Sheets

- https://ams.com/documents/20143/36005/AS7341_DS000504_3-00.pdf/

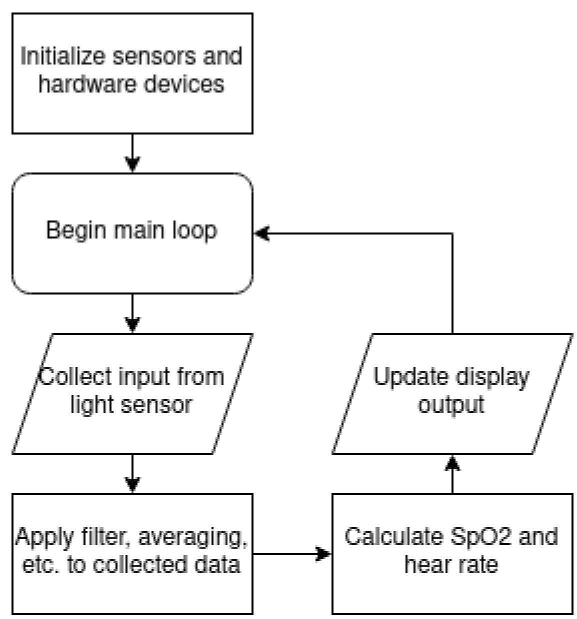
Software Design

Overview

The software for this project will be written using the arduino programming language because it is what our chosen microcontroller supports. The microcontroller was chosen due to availability of online information and library support for the other sensors we are using. The display we will be using and the light sensor (Adafruit AS7341) both operate through I2C and have official libraries that will work with the arduino and will simplify interfacing with the hardware.

Setup and Main Loop

Like many of the other Adafruit sensors, the AS7341 library allows interaction with the sensor as a class which handles the more detailed aspects of controlling the sensor in the back end. The AS7341 will need its integration time configured with appropriate values during the setup phase of the program. In this sensor the integration time refers to the amount of exposure time the photodiode receives before outputting a result. A longer exposure time will allow the sensor to get a more accurate reading in a low light environment but reduce the possible polling rate. It may be necessary to do some tuning of this value if the sensor has trouble picking up the light passing through the finger within the enclosure.



The display will use the SSD1306 library for monochrome OLED displays that are either 128x64 or 128x32. This library extends the capabilities of the [Adafruit GFX] library with specific support for the subcategory of displays. Like with the AS7341 library, the display is represented by a class which provides functions to simplify configuring and setting the output of the display. This class needs to be initialized with the resolution, I2C bus, and reset pin and provides functions for drawing shapes or writing text to the display.

The digital output pins that power the IR and red LEDs will also need to be configured during the setup phase. When configured as outputs, the digital pins can be set to a value of high or low to turn the LEDs on or off.

Because a pulse oximeter is a simple device in concept the program's main loop will only require repeating several basic steps. First data will be collected from the sensor. The details of this process are described in more detail in the next section. After a suitable set of data is collected, that data is filtered and refined. The exact methods used will require the collection of experimental data and testing, but may include zeroing out some of the least significant bits of data points and a moving average. The filtered data will then be put through the algorithms for calculating SpO2 and heart rate discussed below. Finally, those values will be displayed on the OLED and the process will repeat.

Data Collection

The light sensor is able to read multiple wavelengths simultaneously, so there are two ways the data collection may be implemented depending on results of testing: simultaneous or sequential reading. The light sensor reads all channels at once using a single function call. The values can then be read from the registers where they are stored after the previous function call based on the wavelength range you want to use. The values output by the sensor are 16 bit values representing the magnitude of the light within a given wavelength limited by the range of the sensor. With the simultaneous reading method, the red and IR LEDs would be powered on at the same time, values would be read from the light sensor, and the LEDs would be turned off. The values for the red light channel and IR channel could then be collected together. The sequential method will turn on one LED, read the value, turn off the LED, turn on the other LED, read the value, and turn it off. The sequential method would take more time than the simultaneous method because it would require twice as many readings from the light sensor. It may be necessary to use the sequential method, however, because the light output by the two LEDs may overlap or interfere with each other. Both methods will need to be tested, but the simultaneous method would be preferable in order to achieve a higher possible polling rate.

There are also two different data structures we are considering for storing the readings: either two separate lists for the IR and red light readings or as a list of tuples that pair the IR and red light readings together. This will be decided based on which is more useful in the final algorithms and which is easier to implement in the arduino programming language.

Algorithms

The algorithm for the heart rate will be based on the waveform generated by the graph readings. The absorbance of the finger will increase as the cardiac cycle pushes blood into it, creating a waveform that corresponds to the heart rate. There are multiple different methods that could be used to determine the average frequency of a set of collected data. You could potentially use a fourier transform to determine the frequency of the waveform, but this method may be difficult to implement or take too much processing time. Another possible method would be looking at local extrema or zeroes to determine the frequency. Using zeroes may also be made more difficult by the changing baseline of the measurements based on the SpO2 value and blood pressure and finding local extrema may take too much processing power for the microcontroller. Determining the best method will require testing with the microcontroller.

The algorithm for determining the blood oxygen percent (SpO2) will be the most complicated part of this project. The principle behind the algorithm is that one of the wavelengths is absorbed more strongly by oxygenated hemoglobin and the other is more strongly absorbed by non-oxygenated hemoglobin. The algorithm cannot be based on absolute values because each person's finger will vary in absorbance based on size and potentially other factors like skin tone. We cannot make a working algorithm to measure this until after we assemble a prototype because it will be based on the ratio between the absorption of the two wavelengths of light and will require some experimentation and tinkering. This algorithm will likely be the most difficult and time consuming part of this project.

Libraries

Light Sensor:

- https://github.com/adafruit/Adafruit_AS7341

OLED:

- https://github.com/adafruit/Adafruit-GFX-Library
- https://github.com/adafruit/Adafruit_SSD1306

BOM

Part	Link	Cost
Infrared LEDs	https://www.amazon.com/l	$7.99 for 100

	nfrared-Lighting-Electronics-Components-Emitting/dp/B01BVGIZGC/	($0.08 per LED)
Red LEDs	https://www.amazon.com/CHANZON-PC-59042-Emitting-Assorted-Arduino/dp/B01AUI4WC8/	$6.99 for 100 ($0.07 per LED)
Light sensor	https://www.adafruit.com/product/4698	$15.95
STEMMA QT Cable	https://www.adafruit.com/product/4209	$0.95
Arduino	https://www.amazon.com/Arduino-A000005-ARDUINO-Nano/dp/B0097AU5OU/	$24.50
Display	https://www.amazon.com/Pieces-Display-Module-SSD1306-3-3V-5V/dp/B08CDN5PSJ/?th=1	$14.99 for 5 ($3.00 per display)
Total		$71.37

Schedule

Week	Goals
1 (10/23 - 10/29)	Order parts, design and print prototype of 3D model for enclosure
2 (10/30 - 11/05)	Solder/Connect hardware and assemble enclosure suitable for taking measurements. Write software to take measurements. Refine enclosure model if necessary.
3 (11/06 - 11/12)	Tune software calculations to give accurate readings. Extend software to integrate display if not already done.
4 (11/13 - 11/19)	Thoroughly test and refine enclosure

	design and software, testing against multiple users.
5 (11/20 - 11-26)	Finalize hardware and software design based on testing during the previous week and continue to adjust as necessary.

Agreement

We agree to contribute our time and effort equally, to the best of our ability, in order to complete this project on the decided schedule.

Matthew Mueller
James Pate