**Is it feasible to use the arduino and biofeedback sensors to analyze pain in the body?**

**Abstract**: The arduino is cheap, easy to use, powerful and reliable - this is why it shows such promise as an integrative tool for medicine and health analysis. So many clinics and hospitals around the world look for the cheapest, and most efficient way to collect data about patients.

**Background**: Throughout the semester I explored the possibility of using an EEG sensor, a pulse sensor and a MyoWare muscle sensor to provide biofeedback - specifically using these sensors to analyze my body's response to pain while using a pavlok shocker to administer between 50 - 450 Volts at a time into my arm (starting at 0 V and incrementing by 50 V at a time). Some sensors showed more promise than others, and some proved to me more reliable. The EEG was very interesting to look at and was meant to show a peak in delta waves in the brain when in pain - but it was difficult extracting raw data that could be interpreted by the Arduino. The pulse sensor was very arduino compatible and had many libraries online. It was meant to observe the BMP at different shock values - but didn’t exactly give us that useful of information (at least, not by itself) when looking at pain. While I couldn’t quite figure out how to analyze how the brain interpreted pain, the next best thing I could do was analyze the body's response to pain. Using the pavlok to administer electric shocks, it made sense to look at the contractions in my arm in response to the voltages. For this reason, the MyoWare Muscle Sensor was by far the most useful of the sensors. While it was sometimes unreliable, I noticed that a large part of getting a successful outcome boiled down to making sure there was proper placement of the nodes on the arm and making sure that its gain potentiometer was calibrated. However, after tinkering and proper calibration - I was able to get adequate data and readings to observe a direct relationship between shocking at higher voltages and picking up higher readings for electrical activity in muscle contraction. In the end, I found that it was not feasible to use the MyoWare sensor to observe how the body reacts to pain, but it may be feasible to use the same technology(electromyography) to observe how the body reacts to shocks and electricity.

**Significance**: If we could find a more reliable EMG sensor or manage to get a MyoWare muscle sensor to give us a consistent reading, it would be interesting to conduct a feasibility study to observe the exchange of current through the arm at different points - perhaps by using multiple sensors, we could get a better understanding of how electricity is distributed when introducing voltages into your arm. Going forward, I also think there is a lot of promise in integrating multiple biofeedback sensors with the arduino. One day its very likely that we will develop a machine you can strap yourself into that will run checks on all of your biofeedback. This could help doctors figure out an exact source of pain, or muscle stress for example. An arduino could provide the foundation to develop such a product while having it be both portable and cheap.

**Design**:

Use an arduino and a MyoWare Muscle Sensor and a Pulse Sensor to read muscle contractions and stress. Thresholds from data will correspond to lighting up 10 different LED’s on arduino breadboard depending on % voltage shock from Pavlok.



**Pulse Sensor**

**Hardware**

- Mount sensor on Velcro

- Add Velcro to the back of the board, vinyl dots on the front

- Apply to end of fingertip or earlobe

- Header pins

- |-| goes to ground (Black)

- |+| goes to 5V (Red)

- |S| signal goes to A0 (Purple)

**Software**

We will use existing code for the Pulse Sensor:

- Download the Arduino Pulse Sensor library from github

- Source: PulseSensor\_Amped\_Arduino-master - sparkfun

- Allows us to visually see each pulse on the serial plotter

We will use our sensor to establish thresholds representing different pain values by shocking ourselves with the Pavlok at different voltages and recording the input sensor value

- the sensor takes in a value between 0 and 1000

- online sources thresholds use value of 550 to designate a pulse but we will make our own calibration depending on our recorded input value

- Pulse is felt due to expansion and contractions of blood vessels

- the sensor measures this value using an LED, a photoresistor and a microcontroller

- when the heart pumps, blood pressure rises and the amount of infrared light from the emitter that gets reflected back to the detector increases. The detector passes more current when it receives more light and causes a voltage drop. Two op-amps inside the circuit establish a baseline for the signal and emphasize each peak - as well as filter out noise.

We will use the input sensor values (measured in beats per minute, BPM) in an attempt to correlate the pulse sensor output to stress. We will write code to do this:

- higher BMP indicate higher levels of stress and discomfort

- Look at each peak and valley of each pulse.

- Use our values to determine flat portions to designate space between each beat

**MyoWare Muscle Sensor**

The MyoWare Muscle Sensor is an arduino compatible electromyography (EMG) sensor from Advancer Technologies that reads the voltage in your arm through your skin. It works by recording the electrical activity of the muscle tissue using three electrodes that attach to your skin.

**Hardware:**

Solder three header pins into the MyoWare Sensor

- |+| goes to 3.3V (Orange)

- |-| goes to ground (Yellow)

- |S|signal goes to A3 (Green)

Correctly place the electrodes on the skin

- The midline of the muscle belly between an innervation zone and a myotendon junction (a region located at the muscle-tendon that represents the primary site of force transmission -see setup below to see where on your arm there is a myotendon junction)

- The strength of the signal is dependent on the position and orientation of the muscle sensor.

- Clean intended area with soap to remove dirt and oil

- Snap electrodes to the sensors snap connectors

- Place sensor on desired muscle (either inner forearm or bicep)

Calibrate and get a baseline reading of the voltage the sensor is measuring when the arm is at rest

- put sensor on and record values when arm is at rest – either through serial monitor or serial plotter

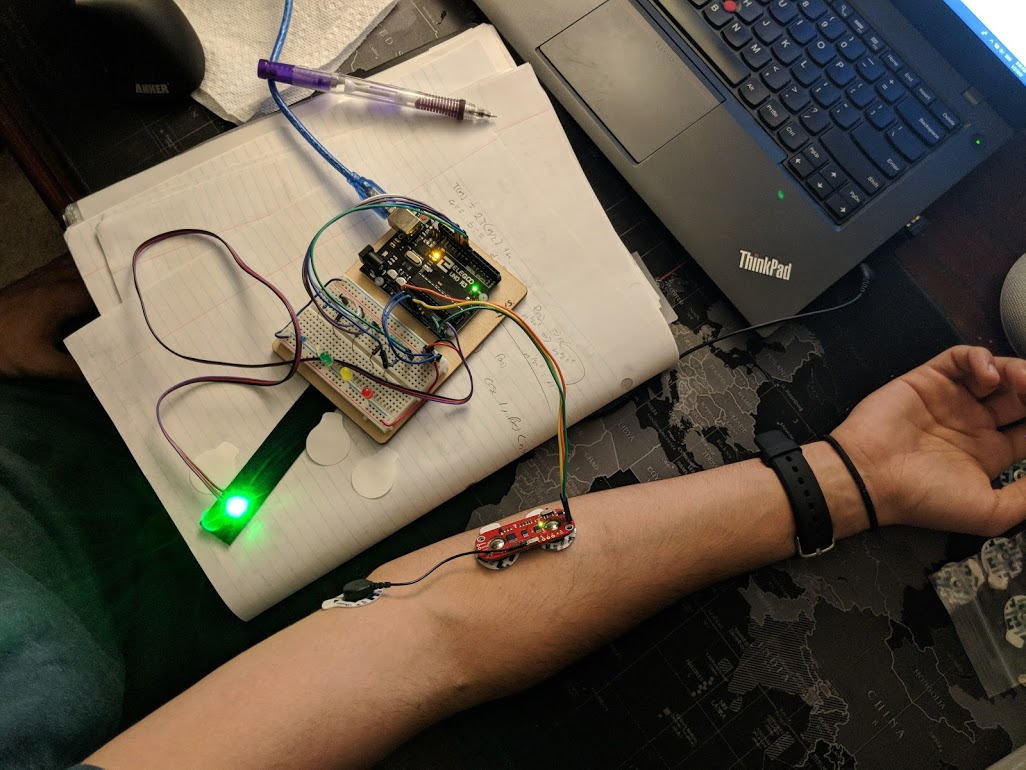
- We will use the gain potentiometer on the board to calibrate our values by getting a baseline reading

Find the appropriate range of the input voltage signal with the myoware sensor

- put sensor on and flex muscle to max capacity

- record values and locate the min and max

- observable range seems to be between 0 and 655



**Software:**

We will write code for MyoWare Sensor such that higher signals will light up more LEDs

- Threshold values for different stress levels will be determined

- Set a threshold value for each LED (LEDs will indicate levels of stress - 4 LEDs will indicate a higher level of stress than 2 LEDs, for example)

- Link each threshold value to a specific LED

- Use if statements to compare sensor value with thresh value

Pseudo Code:

- If (sensor value > thresh value){

Light up this LED} else {

**Pavlok**

**-** Connects to app on phone via bluetooth

- Once both of the sensors are in place, we will put the pavlok on the wrist

- we will start off by first sending a vibration to the pavlok - with zero voltage (control)

- we will shock the subject at 10% max voltage of the pavlok (the pavlok voltage options range from 10 - 100% with 10% increments) and record the Sensor values for both

the MyoWare sensor and the BMP

- we will also observe how many LED's light up with each shock value

- we will do this three times for each voltage from 10% to 100% max shock potential for the Pavlok

**Experiment:**

In the first set of trials, I shocked my arm for 10 minutes starting at 10% voltage on the first minute and incrementing by 10% at each time.

On the second go round, I performed 10 separate trials where I shocked myself 10 times at each voltage in order to get an average reading.

Note: It was important to do the following throughout the experiment

* Keep head straight and spine aligned
* Keep arm in the same place/orientation
* Have both feet planted on the floor
* Breathe steadily and clear your head
* Have someone else administer the shocks with the pavlok

**Technical Objectives**

We will solder pins on the MyoWare sensor to make them Arduino Compatible

We will put electrodes on the MyoWare Muscle Sensor

We will determine where to put the MyoWare muscle sensor on the arm

We will determine where to put the pavlok on the arm

We will get a reading from the MyoWare Sensor

We will determine the range that the MyoWare Muscle Sensor can read

We will shock ourselves with the pavlok and observe the data from the myoware muscle sensor

We will record data as we shock at different voltages

We must be able to isolate the peaks in our data

We must take our baseline when not being shocked with the pavlok

We will find the average peak reading for each 10 -100% voltage and plot them in order to find a relationship between pavlok voltage and myoware sensor peaks

We will determine the differences in the MyoWare data peak averages for each voltage

We will use this these differences to establish 10 thresholds – each corresponding to the change from the peak average from its baseline

We will use these thresholds in our code and allocate them to trigger an led on our arduino breadboard

We will set up an array of 10 leds – each in order corresponding to voltages 10 – 100% on the pavlok

**Technical Barriers**

The MyoWare Muscle Sensor sometimes produces a lot of noise - how will we reduce this noise?

- calculate a running average

- make sure our control node has full contact on the skin

The nodes on the myoware sensor sometimes peel off - the nodes on the MyoWare sensor need to be placed with maximum contact on the skin

- we will make sure to use new electrodes every time and wipe down the skin with alcohol first

- we will place a band aid over the sensor to secure it and the nodes in order to ensure maximum contact

The MyoWare baseline changes every time you move its positioning on your arm – we need to be able to correct for that in our code and be able to keep the differences in our calculated thresh values

* We will set a variable where the user can set their baseline in our code

Our first runthrough of the MyoWare data shows the change in peak voltage overtime, but we need to be able to calculate a function relating

* We will run complete trials at each % voltage rather than changing the voltage over time

**How We Will Measure Success:**

* We should be able to find a trendline that shows a direct relationship between shocking at higher voltages and the emg response to a shock in you arm through the MyoWare Muscle sensor data
* In order for us to claim success with our output, we should see that when we shock ourselves at 20%, 2 led’s light up. And when we shock ourselves at 30% strength with the pavlok, 3 leds light up, etc.