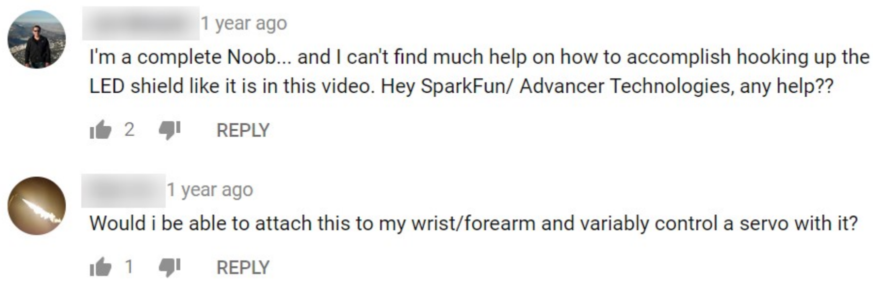
An Unofficial Introductory Tutorial to MyoWare Muscle Sensor Development Kit

Walter Lee Follow
Nov 6, 2018 · 14 min read

MyoWare Muscle Sensor Kit

As a beginner to Arduino, and having had some experiences with electromyography (or EMG; which I will explain soon), I was excited when I discovered the MyoWare Muscle Sensor which can hook up to your Arduino board to collect muscle movement signals. Most standard EMG measurement systems are expensive and not feasible for personal use; products from research equipment companies like BIOPAC, DelSys, or Biometrics Ltd cost well over $6,000 or even $10,000. For personal hobbyists, MyoWare Muscle Sensor therefore offers a good starting point, costing just below $40. Besides this main sensor board, there is also a complete development kit (costing a little over $80) with more accompanying components for DIY designers and researchers.

MyoWare Muscle Sensor (left) / MyoWare Muscle Sensor Development Kit (right) including the Muscle Sensor



It appears that I am not alone (from the official YouTube video).

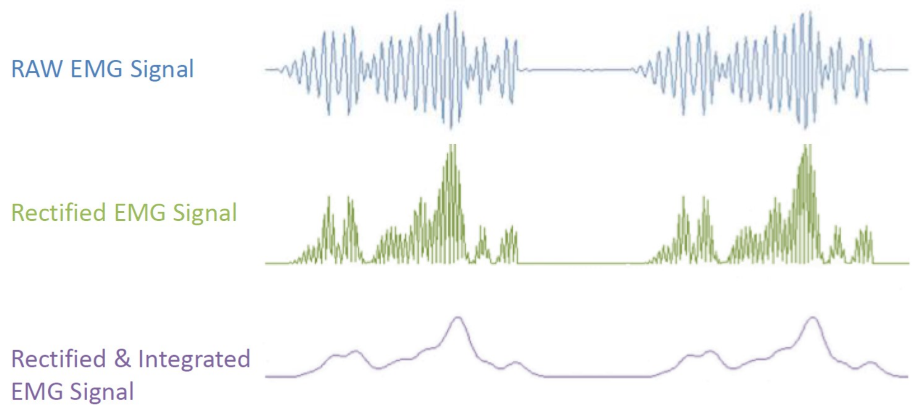
The Development Kit contains not only the Muscle Sensor board, but also a number of other shields that can be used with the sensor. When I got hold of the kit, however, I soon found out that there is no complete beginner's tutorial explaining all of these components in detail on the Internet, which motivated me to write this article. For example, the official SparkFun tutorial or the official video only briefly skims over these things, still leaving many questions.

Hence, this is the introduction to the components you can find in this kit, which I discovered through trial and error. (Note: In this tutorial, I am assuming that you already know the basics of how to use the Arduino IDE software and how to solder.)

What is Electromyography (EMG)?

> TL;DR: Electromyography (EMG) measures the electric signals from your muscle when it contracts.

Electromyography (EMG) is the method of measuring the electrical activity within muscle fibers during activation [1]. Each time when your muscle moves, your muscle fibers are activated with the signals passed down from the brain, through the central nervous system and motor neurons, to the muscle fibers. There are normally more negative charges inside the muscle cell than the outside, but a positive action potential travels along length of the muscle fiber when a muscle contracts [2]. EMG involves measuring this action potential travelling down the muscle fiber at a speed of 2-6 m/s [3]. Such momentary differences in the electric charges can be detected with two (or more) electrodes placed along this route, while amplifying the this difference and suppressing common voltages (with a process called the *differential amplification*). As such, EMG can show the intensity, duration, and order of muscle contractions [1–4].



From the official data sheet

It’s worthwhile to mention the types of signals you may come across with EMG. Raw signals of EMG are zigzag-like signals because these are the differences in electric potential between electrodes as the muscle is being activated (oscillates between V+ and V- back and forth). ‘Rectified’ signals are the absolute values of these raw signals, and ‘integrated’ signals are smoothed out to show the pattern of muscle activation magnitudes. MyoWare outputs ‘Rectified & Integrated’ signals as default, but also provides ‘Raw’ signals as well for those who wish.

From the official data sheet. Correct placement is on the midline of muscle belly, along the length of the muscle.

When collecting EMG, it is recommended to first remove any hair and clean the skin [3–4]. Two electrodes are placed 1-2 cm apart from each other parallel to the length of the muscle fibers on the midline of the muscle belly [4–5]. Then, another electrode, the reference electrode, is placed for subtracting off the signal noise and common voltages between the two muscle electrodes. It should be placed as far away from the main electrode as possible, ideally on a bony prominence [5].

What is in the Kit, and What Do I Need?

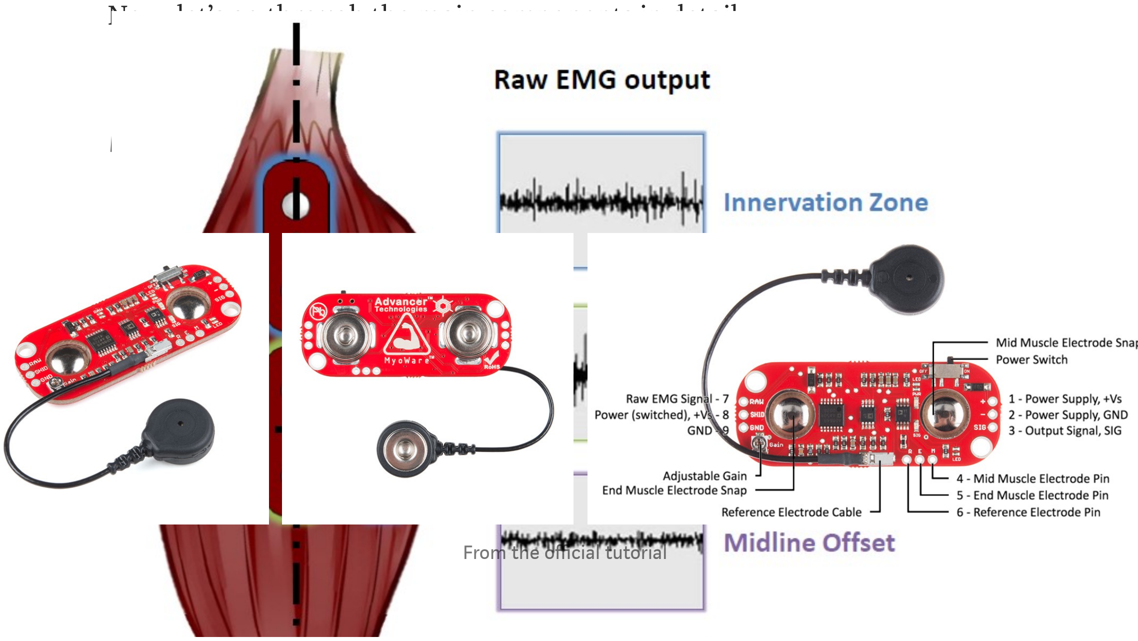
Included in the Development Kit are:

- MyoWare Muscle Sensor

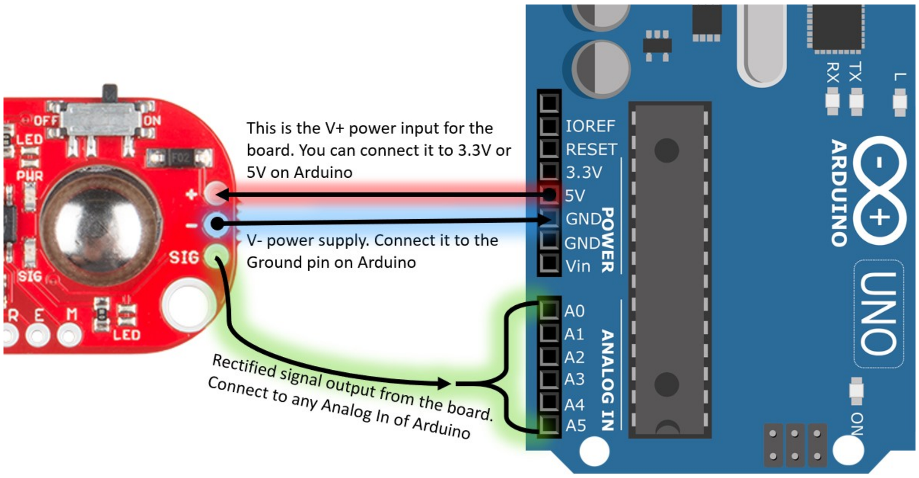
- MyoWare Cable Shield
- 3-Connector Sensor Cable
- MyoWare Power Shield
- MyoWare Proto Shield
- MyoWare LED Shield
- 10x EMG Electrodes
- 11x Stackable 3-pin Female PTH Headers
- 3x 3-pin Breakaway Headers

In addition, to follow this tutorial, you may need:

- Arduino UNO board
- Solder
- Soldering Iron
- Servo motor (generic)
- Breadboard
- 330-Ω Resistor
- Jumper Wires
- 4-leg, common-cathode RGB LED
- 2 CR2032 batteries



From the official tutorial

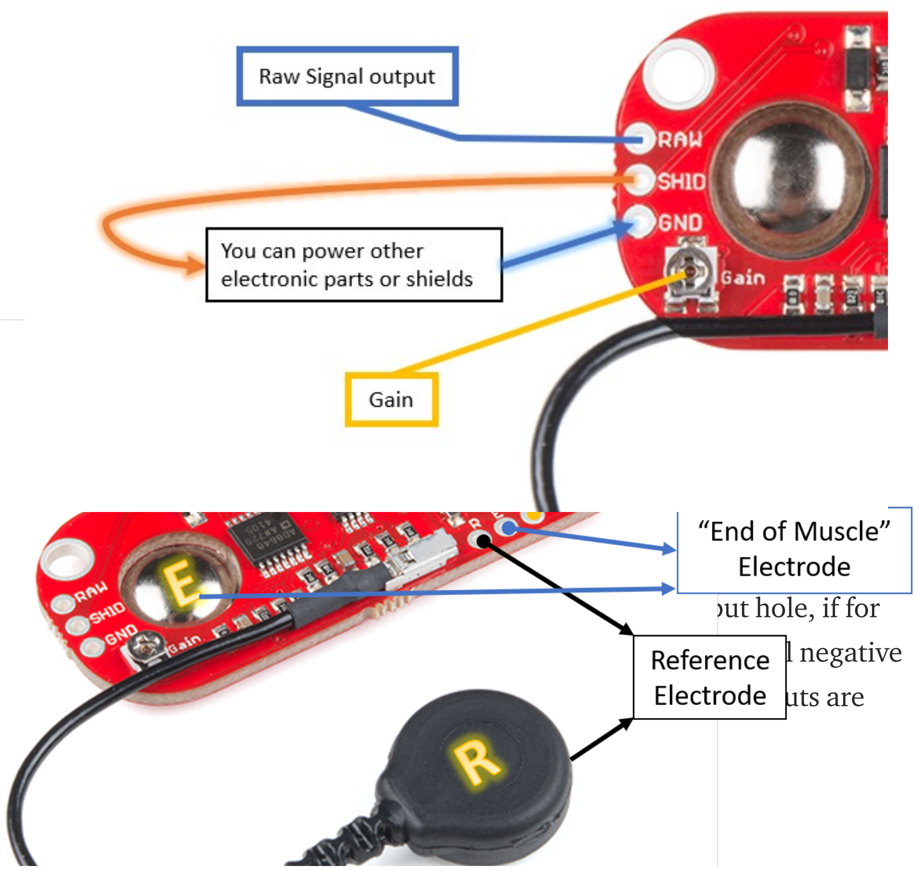


The three rightmost holes of the sensor are the most essential. Simply provide the voltage to the sensor with + and - holes, and collect the (rectified & integrated) EMG signals from the sensor from ‘SIG’. This output signal will be analog, ranging in values 0-1023. (Arduino takes analog signals that are essentially a scale of voltage from 0V to 5V of electric inputs, which is the maximum value of 5V divided into 1024 levels).

Electrode snaps and corresponding pinholes

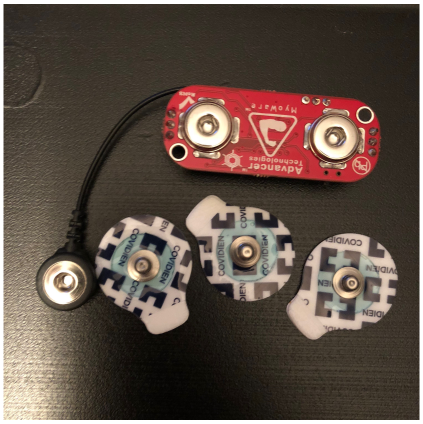
This is the part that I believe can be a little misleading. MyoWare designates which electrode should go to the “end” of the muscle and which should go to the “middle” of the muscle. In reality, as long as two electrodes are placed in the direction of the muscle fibers on the belly of a muscle, it does not matter which side goes where. Just make sure to provide the reference electrode, and you can place ‘M’ and ‘E’ electrodes in whichever orientation you like. In fact, it is not advised to place electrodes directly on or near the end of the muscle, as these areas tend to produce unstable signals [5].

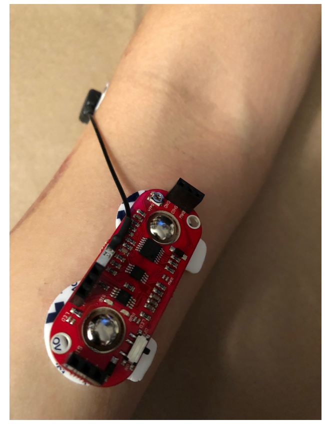
MyoWare also provides 3 pinholes (“R”/”E”/”M”) for each of these electrodes, in case you do not want to use the built-in electrode snaps. You would have to solder wires/connectors into those holes, of course, and provide your own electrodes, lead wires, and snaps instead. I suggest you play with these only when you first get the hang of MyoWare. For now, you can ignore these holes.



›ut hole, if for
l negative
uts are

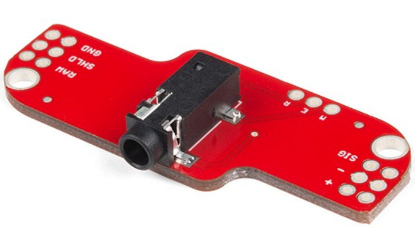
...ic parts, if you need. Lastly, do not worry about "Gain", it controls how much you want to amplify the EMG signal, but you probably won't need to do it as a beginner.

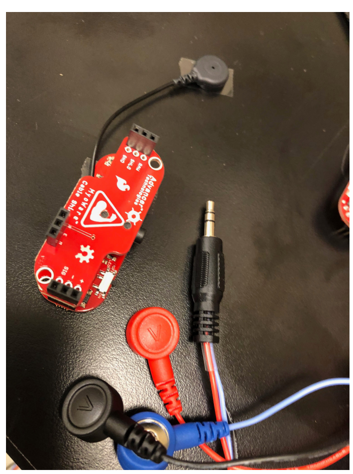


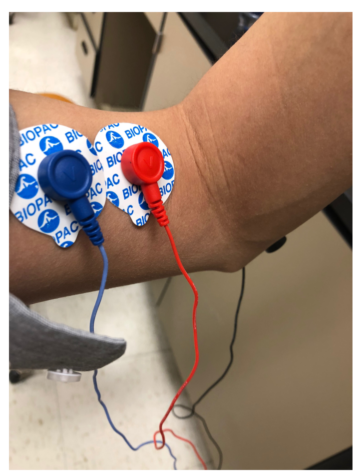


Before moving on, I just want to make sure that you grasped how to use the built-in electrode snaps of the sensor, in case it wasn't clear. You simply snap the electrodes onto the three metal snaps, and then place the two main electrodes along the belly of the muscle you want to measure. The reference electrode can then be placed as far away from the muscle as possible. Depending on the muscle, it may not be long enough to find a body protrusion (which is ideal), but just make sure to avoid the muscle that you are measuring, when placing the reference electrode.

MyoWare Cable Shield



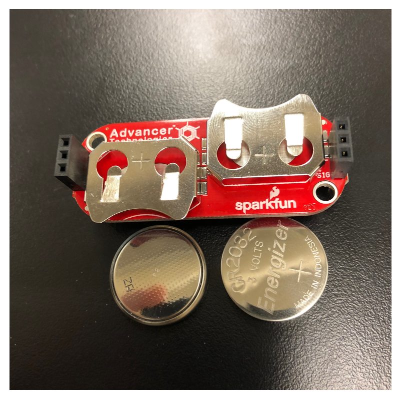




The Cable Shield can be used when you do not want to use the sensor's built-in electrode snaps. The Development Kit comes with the 3-connector sensor cable to connect to the Cable Shield.

Note: In my experience, this Cable Shield is not recommended for signal collection, since I've found that it tends to attract signal noise. EMG is a very weak signal, and is susceptible to noise from various sources, including wire motion, unstable electrode-skin contact, ambient electromagnetic fields, etc. [6]. The fact that the raw signal has to travel all the way down the wire and through the pinholes before it is amplified in the MyoWare Sensor board appears to be an open invitation to noise artifacts. Researchers tend to use more advanced (and infinitely more expensive) devices to combat such signal noise.

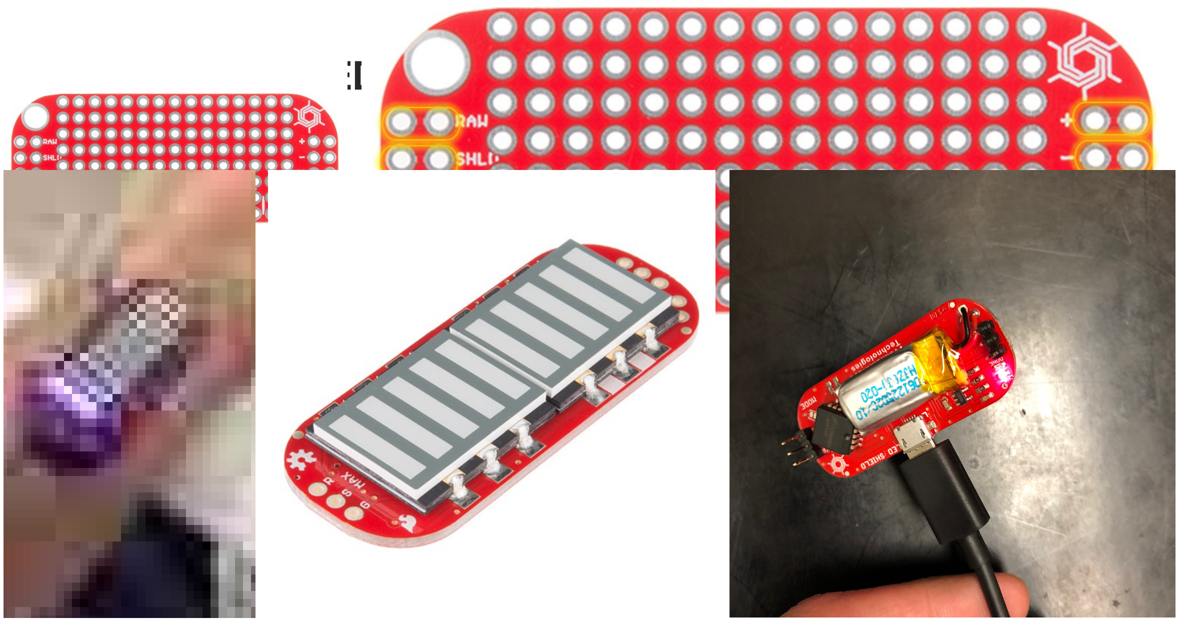
MyoWare Power Shield

As the name suggests, this shield is used to power the sensor with two CR2032 batteries, if you do not want to use Arduino/wires to provide power to the sensor.

MyoWare Proto Shield

The Proto Shield is used for prototyping, hence the name. Most holes are not connected to any other holes except for the circled pairs (since one of the pair needs to connect to the sensor or other shields, while the other hole can be used for prototyping). I will provide an example later on how to use this.



This one is also demonstrated in the official video. I don't see much practical value in this shield, but it looks cool. All you need is to **simply stack this on the sensor** and the LED bar lights will indicate how much muscle activation is going on (you don't need to do any programming). It also comes with a built-in rechargeable battery, which powers both the MyoWare Sensor board and LED shield itself when stacked. This battery can also be recharged with a micro-USB cable port (but USB cable is not included in the kit).

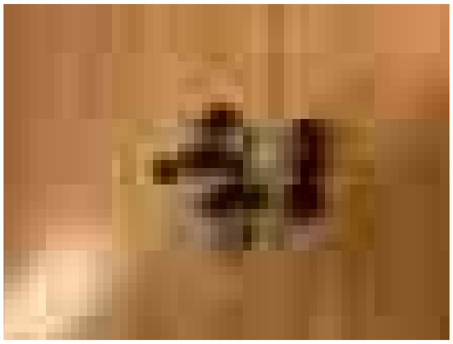
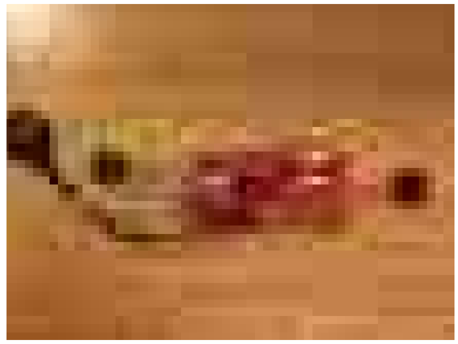
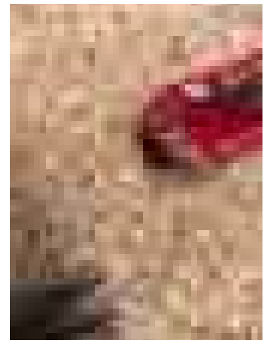
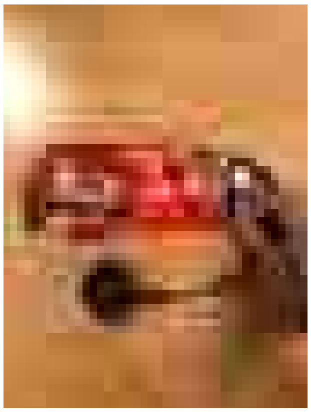
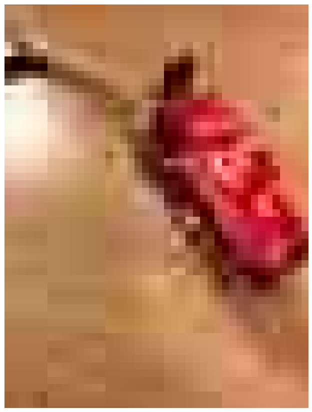
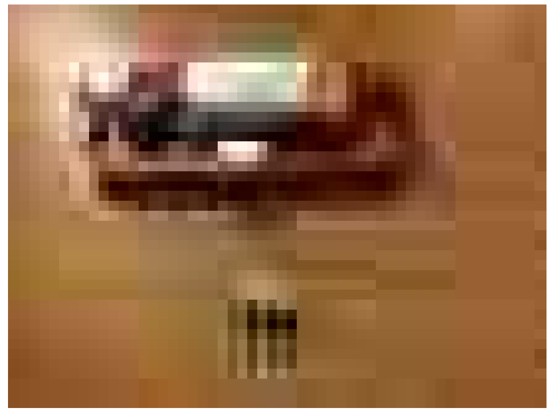
How to Stack Them

The first rule of stacking is that **pinholes with the same labels are meant to be connected together**; "+" hole of the Sensor board should be connected to the "+" hole of other shields, and the "RAW" hole of the Sensor should be connected to the "RAW" hole of other shields, and so on.

Now, you might be wondering, “Which shields can be combined with which other shields?” I’ve made a diagram of which of these can be combined:

The Sensor board is a must-have, and is recommended to be stacked bottom-most (because of the built-in electrode snaps to apply directly onto the muscle). Everything else is optional; the shields can be combined however you like, depending on the project. Also it does not matter which one goes on top of which, and only the LED shield is recommended to be on the very top in any case (because obviously you would want to see the LED lights without any shield on top of it). **However, one caveat is not to use the Power Shield and LED Shield together in any case**; they both provide power with batteries of different voltages, and should not be used together.

In doing so, it can be also confusing which connector pieces to solder onto which; because the kit comes with a bunch of stackable 3 pin female PTH headers and breakaway headers. While there are no technical rules regarding this, there is a way that makes the most sense:

My recommendation is to:

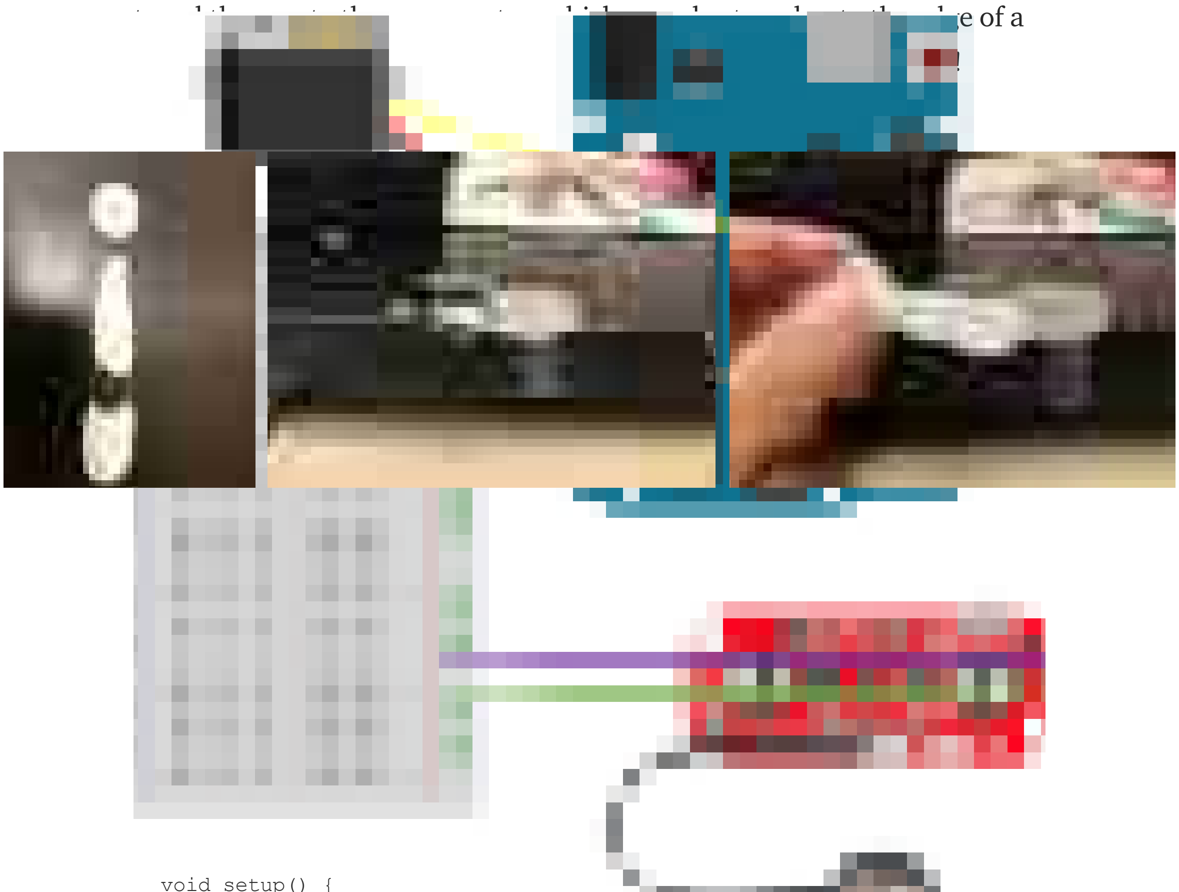
- Use the PTH headers for everything except for the LED shield. When soldering, make sure that the female part goes onto the top of the board, while the male parts stick out towards the bottom of the board.
- Cut away to long pins from the bottom of the sensor board, so that no pin can poke the muscle when using the built-in electrode snaps.
- For the LED shield, use the breakaway headers, so that no female part is connected to the shield (as the LED shield is on the top). The longer parts go towards the bottom.

Example 1 — Controlling a Servo Motor

Moving a servo motor according to your muscle movement, I believe, would be a classic example to get you inspired. So I’ve made a simple Arduino example to show you how to move a servo motor with EMG signals.

Wiring diagram for this example controlling a servo motor with MyoWare sensor

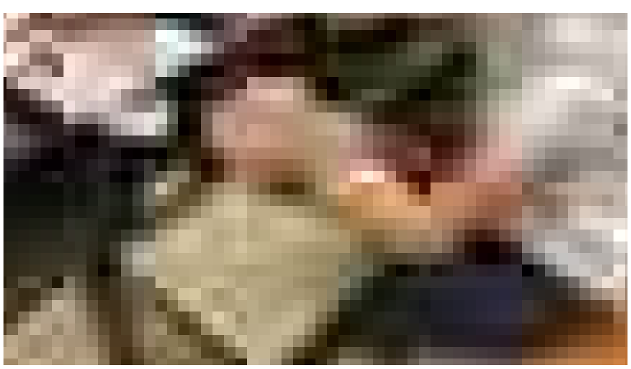
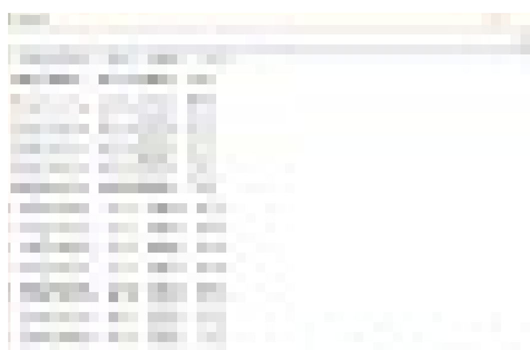
For this simple demonstration, I also made a fake arm with paper and
… e of a



```
void setup() {
 Serial.begin(9600); // Starting the communication with the computer
```

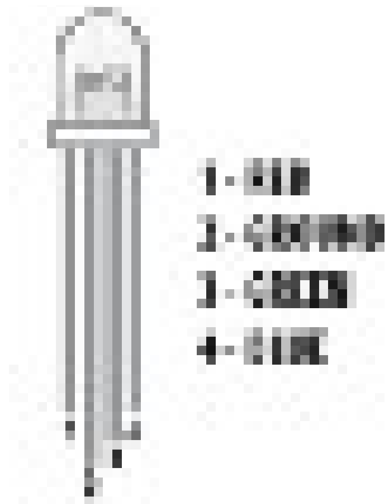
```
 elbowJoint.attach(9); // Tell the servo it is plugged into pin 9
}

void loop() {
 EMGsig = analogRead(A0); // Read the analog values of the
rectified+integrated EMG signal (0–1023)

if (EMGsig < threshold){     // If EMG signal is below the threshold
    servoPosition = 20;       // Servo will remain at 20 degrees.
  } else{            // If the EMG signal is above the threshold,
   servoPosition=map(EMGsig,threshold,1023,20,160);
  // The servo angle will be mapped with the EMG signal,
  // changing the range of 300(our threshold)-1023 into the range of
20-160 degrees.
  // 20 and 160 can be switched depending on which direction of
rotation you want.
}

 elbowJoint.write(servoPosition);
  // Move the servo to the 'servoPosition' degree
 Serial.print(servoPosition);Serial.print(" degrees, with EMG:
");Serial.println(EMGsig);
  // Display the servo and EMG values.
 delay(1000);
  // 1 second (1000ms) delay to not cause it to move as frantically.
But this can be adjusted as you like.
}
```

Connect the Arduino UNO board to the computer, and upload this code. Arduino will then take analog EMG values (0–1023), and will convert to the range of angles between 20–160, if the EMG signal is above the threshold. (instead of using 0–180 degrees, 20–160 degrees are used here, to prevent the servo from twitching.) In the resting stage, the servo motor will stay at the 20-degree position, until your muscle signal is high enough to move the servo motor. Serial monitor values will also update you on the status.

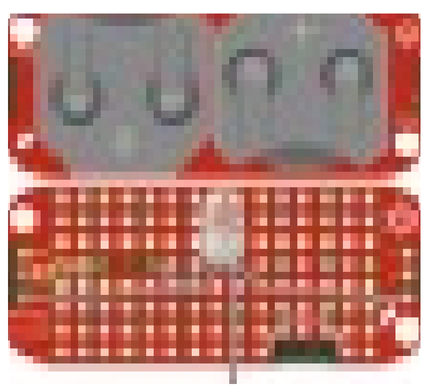
I intentionally set the servo reaction time for 1-second intervals, in order to keep the servo motion infrequent. Try changing the delay time or servo angle range and see how it works differently!

Example 2 — Using the Proto Shield



4-leg common-cathode RGB LED pins (Image source)

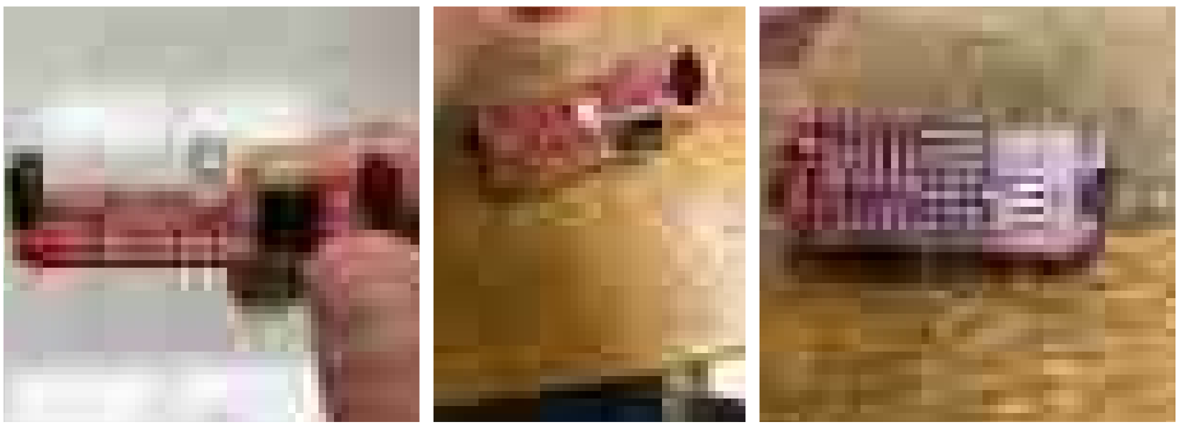
For those who want to use the Proto Shield for prototyping, I'm providing a simple example of prototyping without any other breadboard or Arduino. In this example, I'm using an RGB LED light and a resistor, to control the color of the LED depending on the amount of muscle activity detected.



Wiring diagram of the Proto Shield example (also using the Power Shield for the battery). The brown line shows the soldered connection on the back side. Notice that one of the RGB LED's legs is not used.

As you may notice from the wiring diagram, this is a simple system using the constant voltage provided by the 'SHID' pin to power one color of the RGB LED, while the variable voltage from the 'SIG' pin powers another color of the RGB LED. (And don't forget the ground pin.) I am using the red light here as the constant light (with the resistor keeping the LED from receiving too high constant voltage) and the blue light as the one activated by the EMG signal. Of course, you can easily switch the pins to create a different color combo; you may need extra wires as the LED legs may not be long enough.

If you've never used any proto shield for Arduino before (like me), it may be a little confusing at first how to actually connect different components on the proto shield. Unlike breadboards, most of the holes on the proto shield are not conductive with one another in any row or column. The answer is to basically solder the components together on the back side. (More info)



How front and back sides of the Proto Shield are soldered. I didn't want to cut the wires but you can.

When done, simply stack the Power Shield and the Proto Shield on top of the sensor and turn the sensor on with the electrodes attached on the muscle. What you should see:

Red light is on as a default, which is powered by the 'SHID' pin that provides a constant voltage. The blue light comes on with the 'SIG' pin providing a variable voltage depending on the magnitude of muscle contraction signal. Hence the mixed LED light looks purple when your muscle contracts.

Conclusion

EMG data can provide electric signals that show when and how much a muscle is being contracted. DIY products like MyoWare offers an inexpensive entry into the world of EMG, either for hobby or for research. I hope you learned to appreciate how EMG works, and how to use the MyoWare Development Kit to derive muscle activation signals for your projects.

Further Resources

- MyoWare official tutorial
- MyoWare official data sheet
- MyoWare GitHub repository
- Arduino Example Code for MyoWare from SparkFun
- Another good Medium article on MyoWare
- A really cool tutorial on making an Iron Man glove with MyoWare

References

1. N. Hamilton, W. Weimar, and K. Luttgens, Kinesiology: Scientific basis of human motion. New York, NY: McGraw-Hill, 2012.
2. P. Konrad, "The ABC of EMG: A Practical Introduction to Kinesiological Electromyography," 2005.
3. R. Baker and A. Shortland, "Electromyography," in Measuring Walking: A Handbook of Clinical Gait Analysis, R. Baker, Ed. London: Mac Keith Press, 2013, pp. 71–87.
4. M. Z. Jamal, "Signal Acquisition Using Surface EMG and Circuit Design Considerations for Robotic Prosthesis," in Computational Intelligence in Electromyography Analysis — A Perspective on Current Applications and Future Challenges, InTech, 2012.
5. C. J. De Luca, "Surface Electromyography: Detection and Recording," 2002.

6. E. A. Clancy, E. L. Morin, and R. Merletti, “Sampling, noise-reduction and amplitude estimation issues in surface electromyography,” J. Electromyogr. Kinesiol., vol. 12, no. 1, pp. 1–16, 2002.

Arduino Myoware Emg Electromyography Muscles