Experiment: Control Machines with your Brain

Now that you have your EMG Signal, can you use it to drive another system? Neural Engineering continues, using the popular Arduino microcontroller.

Time 30 minutes

Difficulty Intermediate

What will you learn?

Here you will learn how to interface an Muscle SpikerShield with external hardware. This is continuation of our Neuroprosthetics Experiment; only now, you will use an EMG Signal from a muscle of choice, paired with an Arduino Microcontroller, to control a bank of LED lights.

Prerequisite Labs

- Muscle SpikerBox - You should become familiar with what an EMG signal is.

Equipment

Muscle SpikerShield

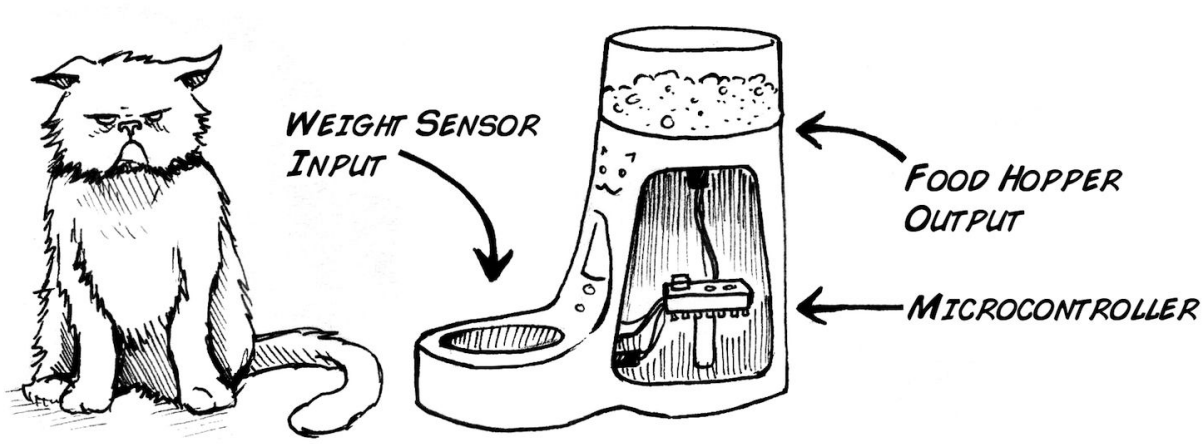
Arduino Uno

Background

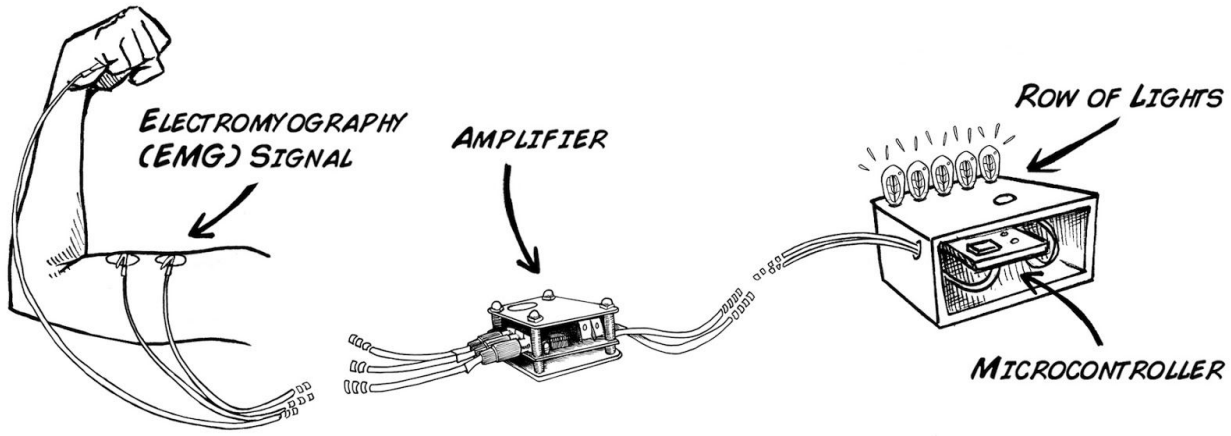
Written in collaboration with Lucas Alverez, a Chilean high school (colegio) student

The Muscle SpikerBox is a pretty cool device all on its own, but it only has 2 ways of communicating its output: by seeing it on SpikeRecorder running on your mobile device or by listening to the audio via the speaker.

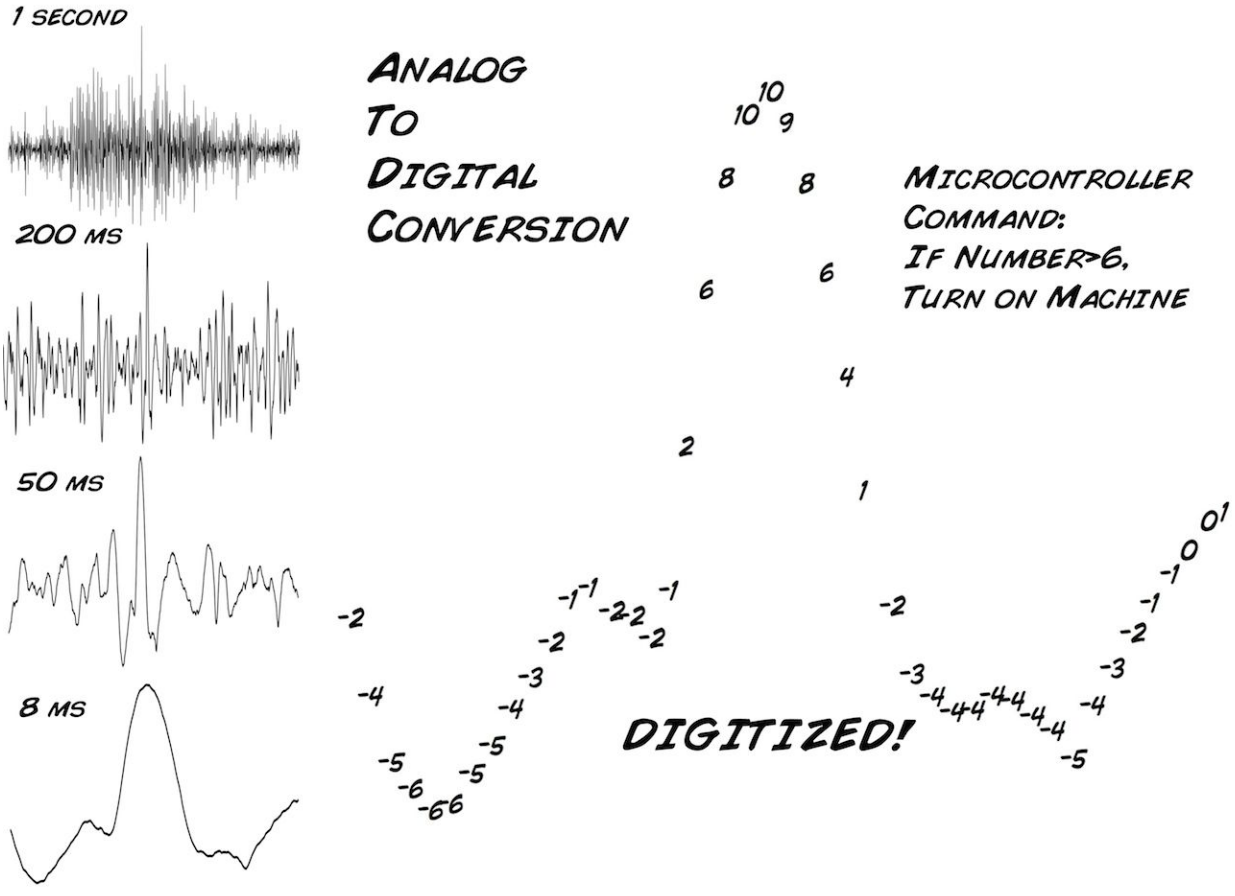
But wouldn't it be fun to use that signal to control things? In this experiment, we will do just that! For this we will use an Muscle SpikerShield, a custom Muscle SpikerBox we have designed that fits on an Arduino Uno board. What is this 'Arduino,' you may ask? An Arduino is a type of microcontroller, which allows you to take in input signals, do some math/analysis on them, and use this information to control other devices. A famous example is an automatic cat feeder. A sensor measures the weight of food in the bowl, this information is sent to the microcontroller, and the microcontroller commands the food hopper in real-time to maintain a set amount of food in the bowl.***Attn: Engineering Nerds - Yes we know you can do this without microcontrollers by using analog control circuits, but the microcontroller makes it much easier.***



But forget about cats, let's do something cooler. If we were to amplify your muscle's electrical activity with our Muscle SpikerBox, you could send this signal to a microcontroller, which can analyze the state of your muscle activity and use it to control a neuroprosthetic robot arm, a video game, or a bank of lights.



Importantly, the Arduino converts the EMG signal from an analog (continuously varying) into a digital one (broken into discrete "numbers"), making analysis and instructions on what to do with the signal much easier.



The arduino microcontroller was designed for artists, designers, and hobbyists - so it is really easy to program the software and setup the hardware. There are many "shields" available for the Arduino that extends the capabilities (for example, adding WiFi or Motors). Here we will walk you through how our Muscle SpikerShield works, which, in an 'introduction demo,' can display the transformed readout of your muscle biopotentials as a row of LEDs. This is only to get you started on utilizing your muscle activity as an interface. You can control whatever (and whoever) you want, your only limitation is creativity and work.

Downloads

This experiment assumes you have a basic understanding of how to use the Arduino Platform and upload code to your board. If you are new to arduino, or need a refresher, check out Getting Started with Arduino on Windows or Getting Started with Arduino on Mac OS X . Note: If you bought our Muscle SpikerShield Bundle, the led_strip2014 code is already preloaded.

Arduino Software

Download our Arduino(.ino) Sketches for this Experiment

Serial_test2014.ino

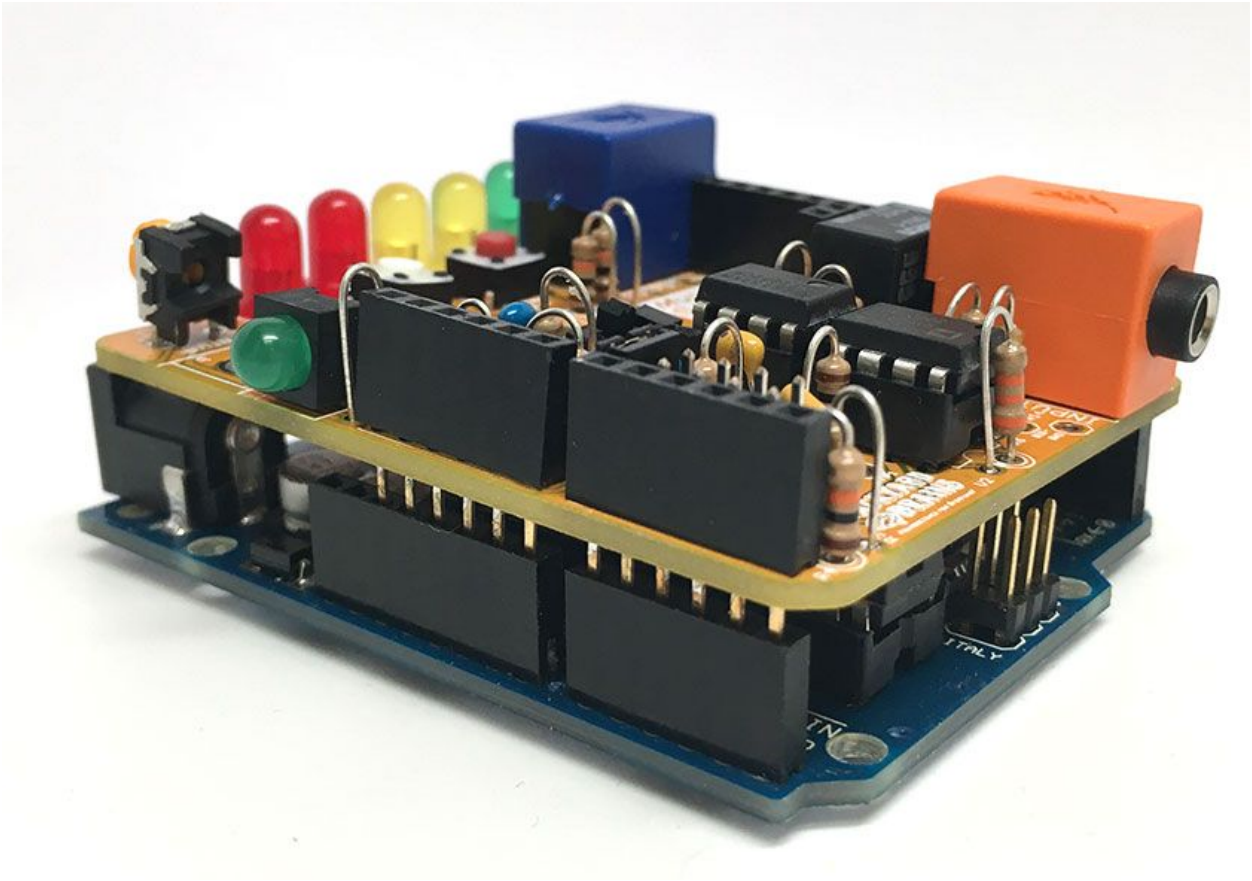
led_strip2014_highergain.ino

Video (Spring 2014)

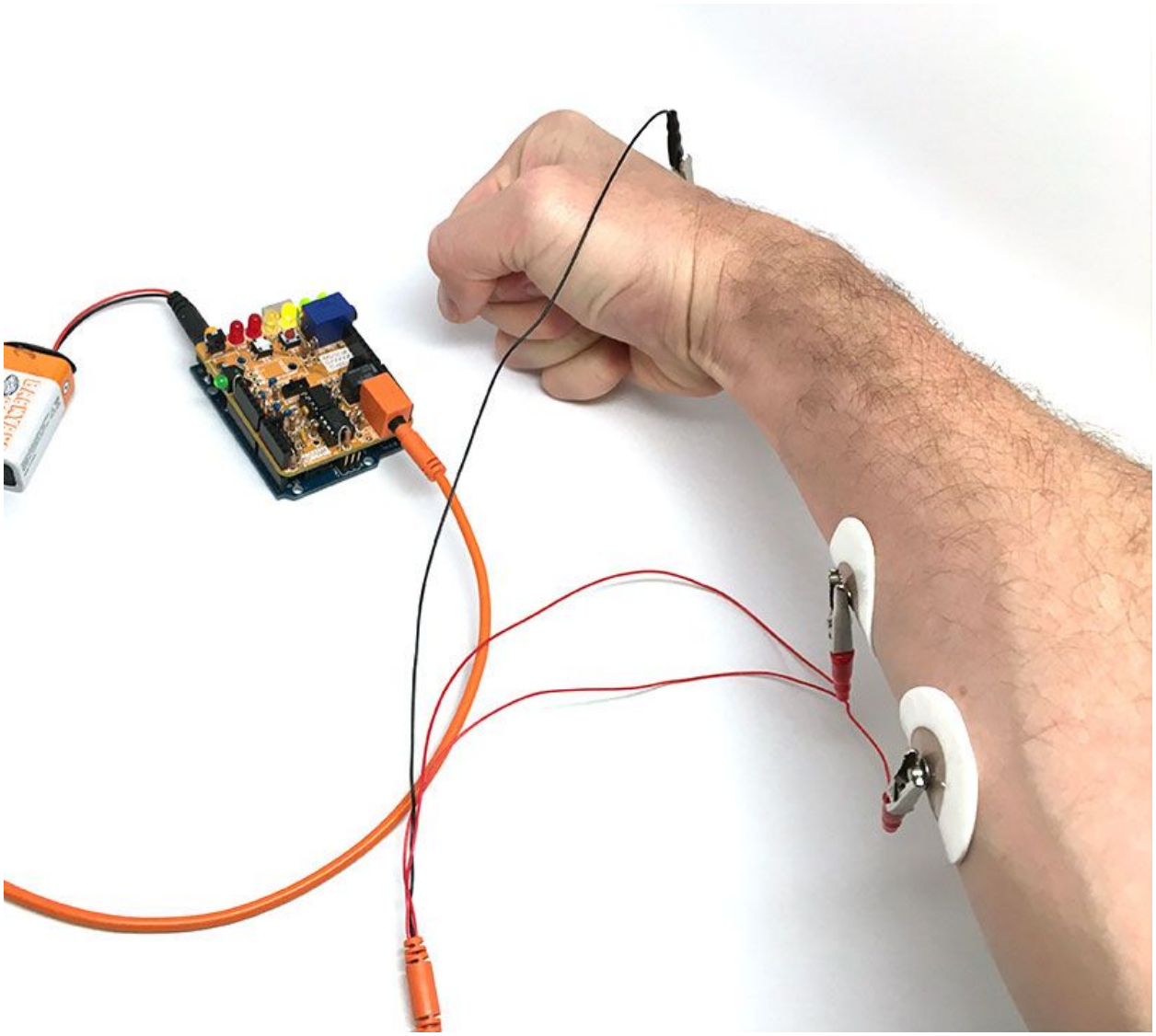
https://youtu.be/WHclLvr8BG8

Procedure (Fall 2015)

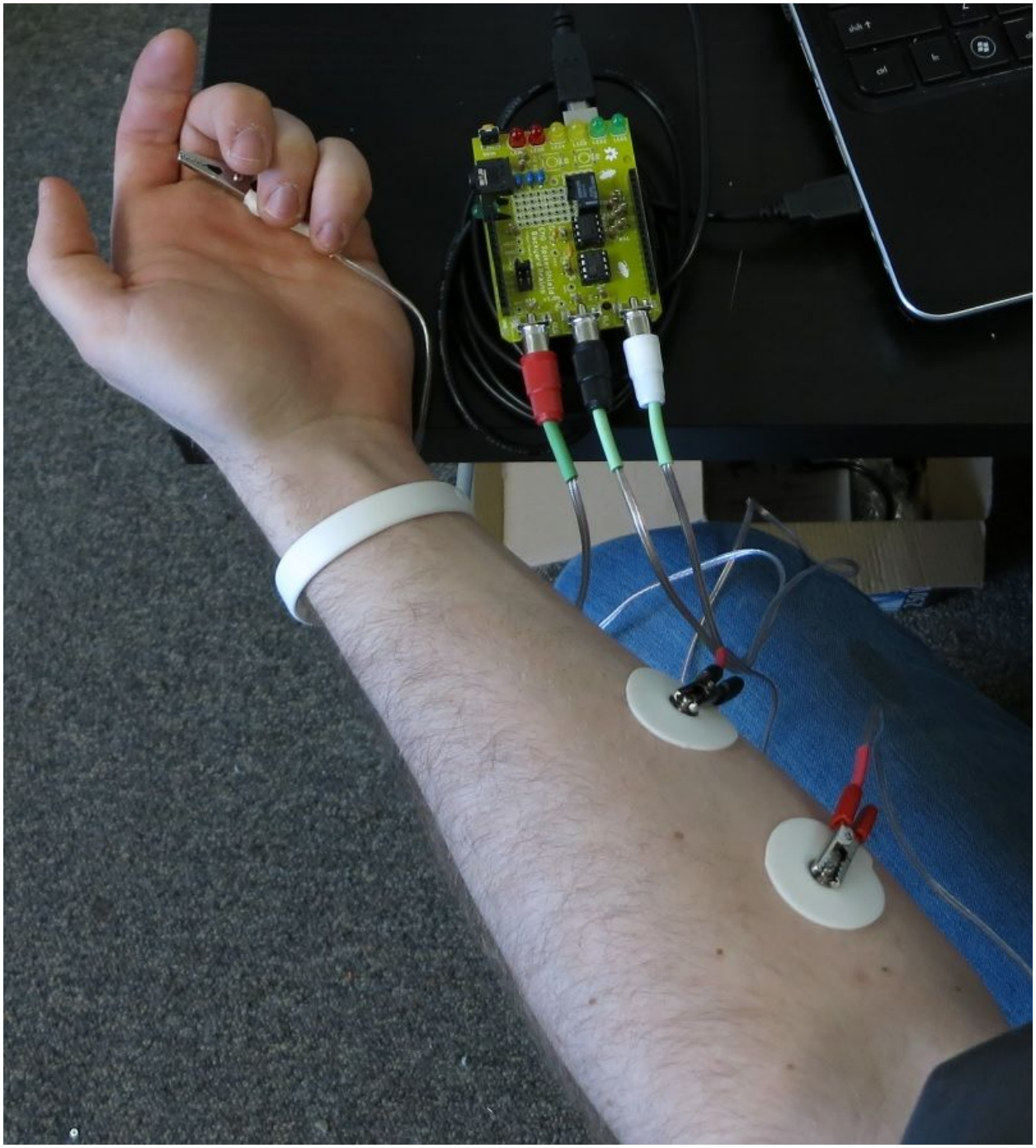
1. Let's get started! First, mate the Muscle SpikerShield on top of the Arduino, aligning all the pins on the shield to the female headers on the Arduino. In this experiment we are using the first Analog input (A0), so be sure to jumper the input select to 1. **Note:** your Arduino can run on an external USB power source once your code has been uploaded, but you need to use your computer to upload and make changes to the code. Plug it in to your computer now!



2. Download our serial_test2014.ino sketch (also mentioned above) and upload it to your arduino. Then connect the orange electrode cable wire from the Muscle SpikerShield to some electrodes on your arms. Hook the two red clips onto your arm and place the black one on the back of your wrist. You can also plug in a speaker to hear your muscle activity alongside the visual outputs!

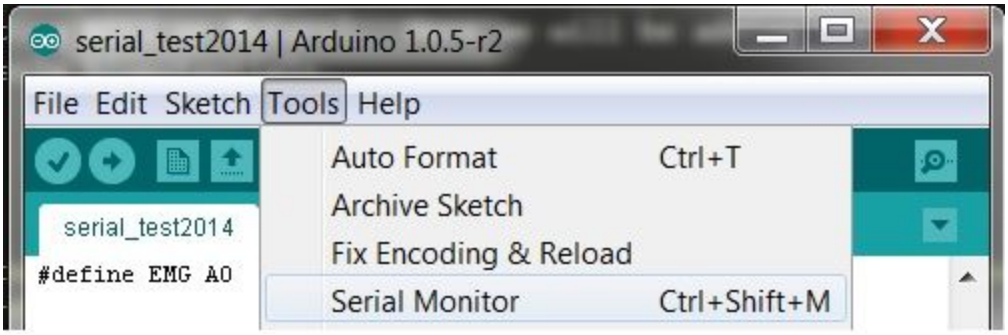


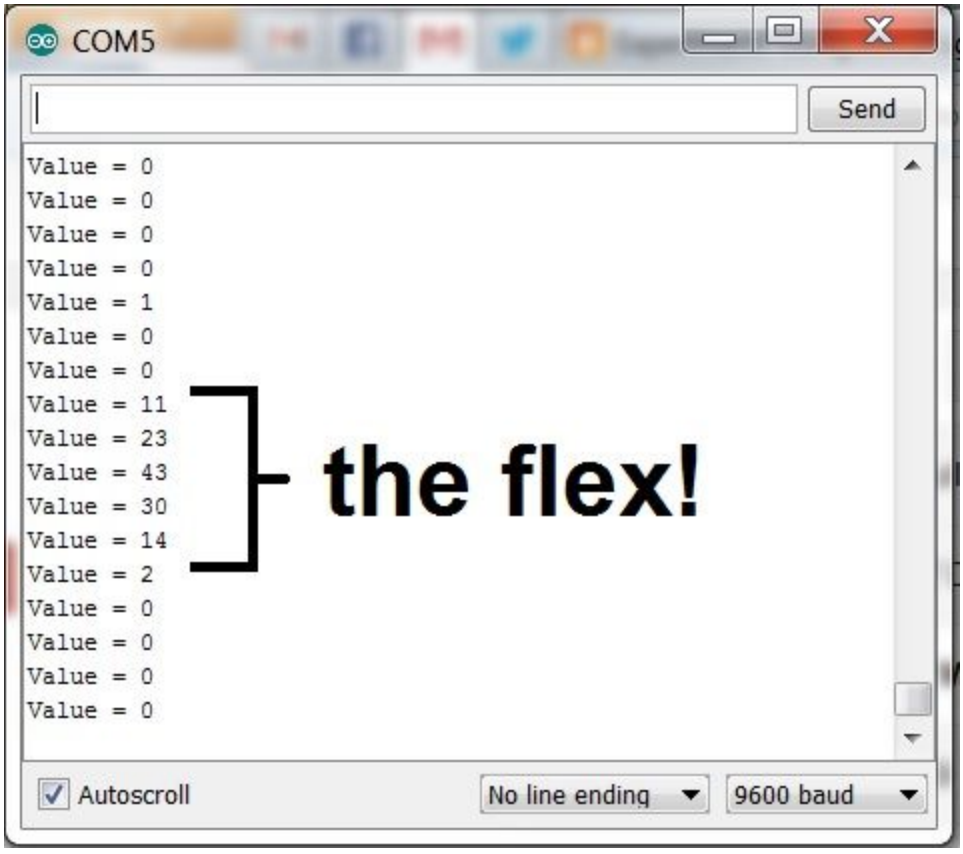
Note: Our older cables required black and red clips onto your arm and hold the white end in your fist.



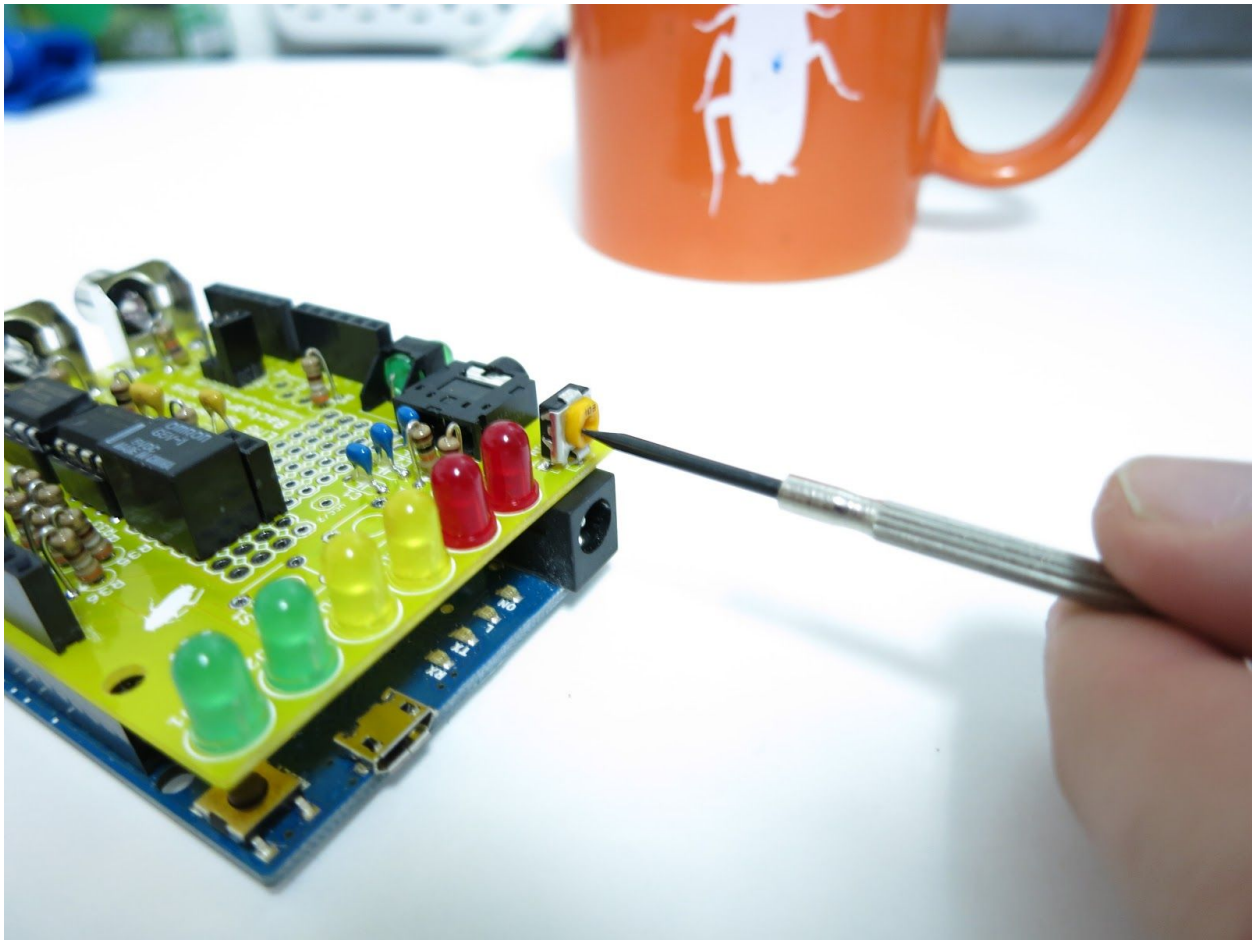
3. Open the Arduino serial monitor from the IDE to check the values. If you get larger values when you flex your muscles, and values near zero when you don't, then everything is working ok! If not, make sure all cables are connected correctly and nothing is loose. You may find connecting a portable speaker to the

Muscle SpikerShield, as we do in the video above, to be helpful so you can hear your muscle activity as well.





If this does not work, you may need to adjust your spikershield's potentiometer (pictured below - depending on the model number of your shield the potentiometer may be located somewhere else on the board). Give it a slight rotation clockwise to increase the gain and test it again in the serial monitor. Repeat this process until you receive results in the serial monitor similar to the ones pictured above.



4. Now that we've seen the kind of input that we can create for the arduino using the Spikershield, it's time to take advantage of it! Upload the second code (also mentioned above), led_strip2014_highergain.ino, and you should see the LEDs light up according to how hard you flex! If you feel the LEDs don't light up proportionately to your arm strength, then tweak the MAX constant from 4 to another value until the represented value is proportional to your strength. If the electrodes light up really easily, increase this value - if you are having a hard time making all the electrodes light up, lower it!

```
#define NUM_LED 6  //sets the maximum numbers of LEDs
#define MAX 254     //maximum posible reading. TWEAK THIS VALUE!!
#define Threshold 3 // this sets the light to activate TENS

int reading[10];
int finalReading;
byte litLeds = 0;
byte multiplier = 1;
```

```
byte leds[] = {8, 9, 10, 11, 12, 13};
```

This is just the first step into your new world of EMG Arduino design. Happy hacking! Let us know what you connect your EMG signals to! Let your creative mind flow and invent.

Science Fair Project Ideas

1. There are lots of potential devices you can interface with using the Muscle SpikerShield-what are some devices that you think would be interesting to control? Can you set up a gripper or a motor so that their strength/speed correspond to how much you're flexing? Can you get this to turn on something that delivers more force than you can?
2. The LEDs provide a way of showing if your muscle contraction has surpassed a certain threshold. How long can someone keep the last red LED on? How does the number of lights on correspond to the signal shown on the Spike Recorder App? Are different muscles more or less able to get all the LEDs on?
3. Most of what we've discussed so far are analog devices-basic motors, grippers, LEDs. As a more advanced project, can you interface and control any digital programs or devices? Can you get it to work with any other apps?

Final Note: If interested, you can read Lucas's Original Write-Up, which we have since updated into what you see now. But...Below is the video that started it all!

Video (Fall 2013)

https://youtu.be/-O1snHpl-qU