

# Getting Started with the MyoWare® 2.0 Muscle Sensor Ecosystem

## Introduction

**Note:** This tutorial is for the MyoWare V2.0 Muscle Sensor and its ecosystem. If you are using the previous version, please refer to the older MyoWare Muscle Sensor V1 tutorial.

Using our muscles to control things is the way that most of us are accustomed to doing it. We push buttons, pull levers, move joysticks... but what if we could take the buttons, levers and joysticks out of the equation and control it with our muscles? The MyoWare 2.0 Muscle Sensor is an Arduino-compatible, all-in-one electromyography (EMG) sensor from Advancer Technologies that allows you to do just that! The MyoWare 2.0 Muscle Sensor has been redesigned from the ground up with a new, easy-to-use, compact design and upgraded with the latest and greatest chipset improving sensor performance and reliability. The innovative snap connector system eliminates the need to solder connections for the MyoWare 2.0 ecosystem. It's that easy: stick on a few electrodes (not included), read the output voltage and flex some muscles! In this tutorial, we will go over the features and related shields to connect the sensor to a muscle group.

**Note:** This sensor is not intended for use in the diagnosis of disease or other conditions, or in the cure, mitigation treatment, or prevention of disease, in a person or other animals.



MyoWare 2.0 Muscle Sensor

○ DEV-18977



MyoWare 2.0 LED Shield

● DEV-18387



MyoWare 2.0 Link Shield

● DEV-18425



MyoWare 2.0 Power Shield

● DEV-18427



MyoWare 2.0 Arduino Shield

● DEV-18426



MyoWare 2.0 Cable Shield

● DEV-18386

Product Showcase: MyoWare 2.0





**Note:** The MyoWare 2.0 ecosystem was designed by myoelectric experts at Advancer Technologies, LLC while SparkFun Electronics handles the manufacturing and sale of the boards.



## Required Materials

To follow along with this tutorial, you will need the following materials. You may not need everything though depending on what you have. Add it to your cart, read through the guide, and adjust the cart as necessary. You'll need at least three biomedical sensor pads per muscle sensor each time you connect to a muscle group.

**Note:** The EMG pads are single use. You'll want to make sure that you have spare pads each time that you connect to a muscle group!

## Minimum Parts

At a minimum, you will need the following parts to connect and power the muscle sensor. You can view the muscle signal from the ENV LED.



MyoWare 2.0 Muscle Sensor  
○ DEV-18977



Disposable Surface EMG/ECG/EKG Electrode -  
24mm (10 pack)  
● SEN-12969

★★★★☆ 4



MyoWare 2.0 Power Shield

● DEV-18427



Reversible USB A to C Cable - 2m

○ CAB-15424

**Note:** Click below for a wishlist of the minimum parts.

**MYOWARE 2.0 MINIMUM PARTS WISHLIST**

## Displaying the Magnitude of the Muscle Signal with LED Segments

You will need the following parts to connect and power the muscle sensor. You can view the magnitude of the muscle signal from the LED segments.



MyoWare 2.0 Muscle Sensor

○ DEV-18977



MyoWare 2.0 LED Shield

● DEV-18387



Disposable Surface EMG/ECG/EKG Electrode - 24mm (10 pack)

● SEN-12969



Reversible USB A to C Cable - 2m

○ CAB-15424

**Note:** Click below for a wishlist of the minimum parts.

**MYOWARE 2.0 MUSCLE SENSOR AND LED BAR GRAPH WISHLIST**

**Note:** We also have the parts available in the basic kit, which also includes the Reference Cable and a screw driver.



MyoWare 2.0 Muscle Sensor Basic Kit

○ KIT-19719

## Connecting to an Arduino

You will need the following to connect an 5V Arduino without the need to solder any wires between boards. Connecting to an Arduino allows you to process sensor data to control a device or viewing the signal on the Arduino Serial Plotter.



MyoWare 2.0 Muscle Sensor

○ DEV-18977



SparkFun RedBoard Plus

● DEV-18158



Disposable Surface EMG/ECG/EKG Electrode - 24mm (10 pack)

● SEN-12969



MyoWare 2.0 Link Shield

● DEV-18425



MyoWare 2.0 Arduino Shield

● DEV-18426



Reversible USB A to C Cable - 2m

○ CAB-15424



Audio Cable TRS - 1m

● CAB-18983

**Note:** You could also use an Arduino Uno as opposed to a RedBoard Plus. Just replace the RedBoard Plus and USB Cable.



Arduino Uno - R3

● DEV-11021



USB Cable A to B - 6 Foot

○ CAB-00512

**Note:** Click below for a wishlist of the minimum parts.

**MYOWARE 2.0 MUSCLE SENSOR AND ARDUINO WISHLIST**

**Note:** We also have the parts available in the development kit. Note that the development kits includes all the boards in the MyoWare 2.0 ecosystem, their associated cables, and a RedBoard Plus.



MyoWare 2.0 Muscle Sensor Development Kit

● KIT-18441

## Transmitting Wirelessly

You will need the following to transmit sensor data wirelessly through an Arduino. Of course, you will need 2x RedBoard Artemis boards and 2x USB cables. You'll need 1x battery pack and 4x AA batteries for each RedBoard Artemis that is battery powered.



MyoWare 2.0 Muscle Sensor  
○ DEV-18977



Disposable Surface EMG/ECG/EKG Electrode -  
24mm (10 pack)  
● SEN-12969



MyoWare 2.0 Link Shield  
● DEV-18425



MyoWare 2.0 Arduino Shield  
● DEV-18426



SparkFun RedBoard Artemis  
● DEV-15444



Battery Holder - 4xAA to Barrel Jack Connector  
● PRT-09835



Reversible USB A to C Cable - 2m



Audio Cable TRS - 1m





### Panasonic Alkaline Battery - AA

● PRT-15201

**Note:** For users interested in using a rechargeable battery, you could also use a NiMH instead of Alkaline.



### Tenergy T4s Intelligent Universal Charger - 4-Bay

● TOL-14457



### 2500 mAh NiMH Battery - AA

● PRT-00335

**Note:** Click below for a wishlist of the minimum parts.

**MYOWARE 2.0 MUSCLE SENSOR AND ARDUINO BLUETOOTH® WISHLIST**

## Accessories

Depending on your setup, you may need the following accessories to add an extension between the hardware and muscle group.



Sensor Cable - Electrode Pads (3 connector)

● CAB-12970



MyoWare 2.0 Cable Shield

● DEV-18386



MyoWare 2.0 Reference Cable

● CAB-19244

## Tools

To easily disconnect the boards, we recommend using a flathead screwdriver. The following screwdriver can be used to remove the stack of boards from each other. Depending on the Arduino that you choose, we recommend getting a hobby knife, solder, and soldering iron when changing the jumper pad for the logic levels.



Solder Lead Free - 15-gram Tube

● TOL-09163



SparkFun Mini Screwdriver

● TOL-09146



Soldering Iron - 30W (US, 110V)

● TOL-09507



Hobby Knife

● TOL-09200

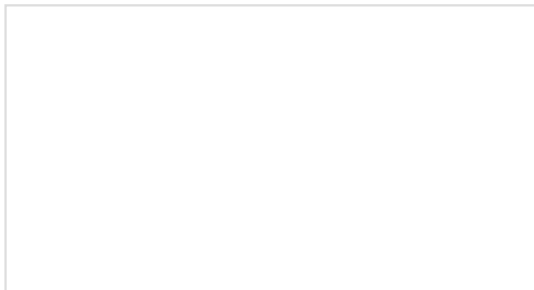
## You Will Also Need

You will also need the following.

- Alcohol Swab (Isopropyl alcohol and a cotton ball)
- USB isolator circuit (optional)

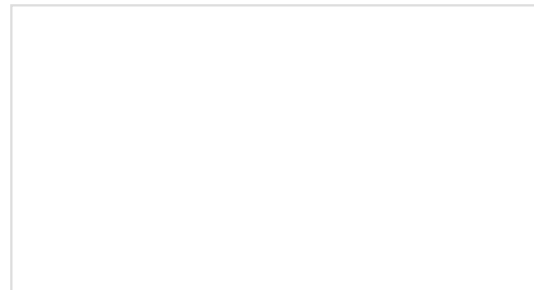
## Suggested Reading

If you aren't familiar with the following concepts, we recommend checking out these tutorials before continuing. Depending on the Arduino development board that you are using, you may need to install drivers. The RedBoards use a different USB-to-serial converter compared to the Arduino Uno. Both the Arduino Uno and the RedBoard Plus use 5V for the logic levels.



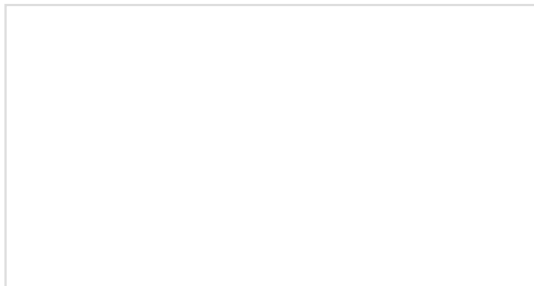
### Analog to Digital Conversion

The world is analog. Use analog to digital conversion to help digital devices interpret the world.

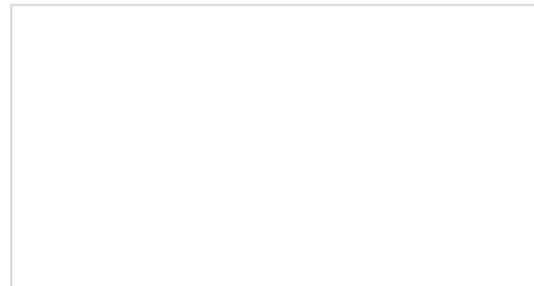


### Logic Levels

Learn the difference between 3.3V and 5V devices and logic levels.



### How to Install CH340 Drivers

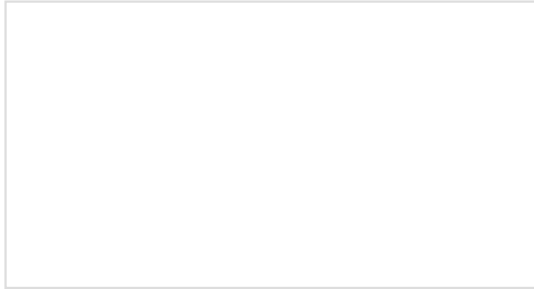


### RedBoard Plus Hookup Guide

How to install CH340 drivers (if you need them) on Windows, Mac OS X, and Linux.

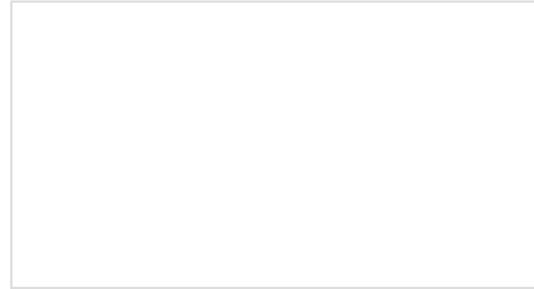
This tutorial covers the basic functionality of the RedBoard Plus. This tutorial also covers how to get started blinking an LED and using the Qwiic system.

For users interested in using a Bluetooth® connection to wirelessly transmit the sensor data, we recommend getting the RedBoard Artemis and installing the board definitions. Note that the RedBoard Artemis uses 3.3V for the logic levels.



### Hookup Guide for the SparkFun RedBoard Artemis

Get started with the RedBoard Artemis - all the functionality of the SparkFun Artemis module wrapped in the familiar Uno R3 footprint



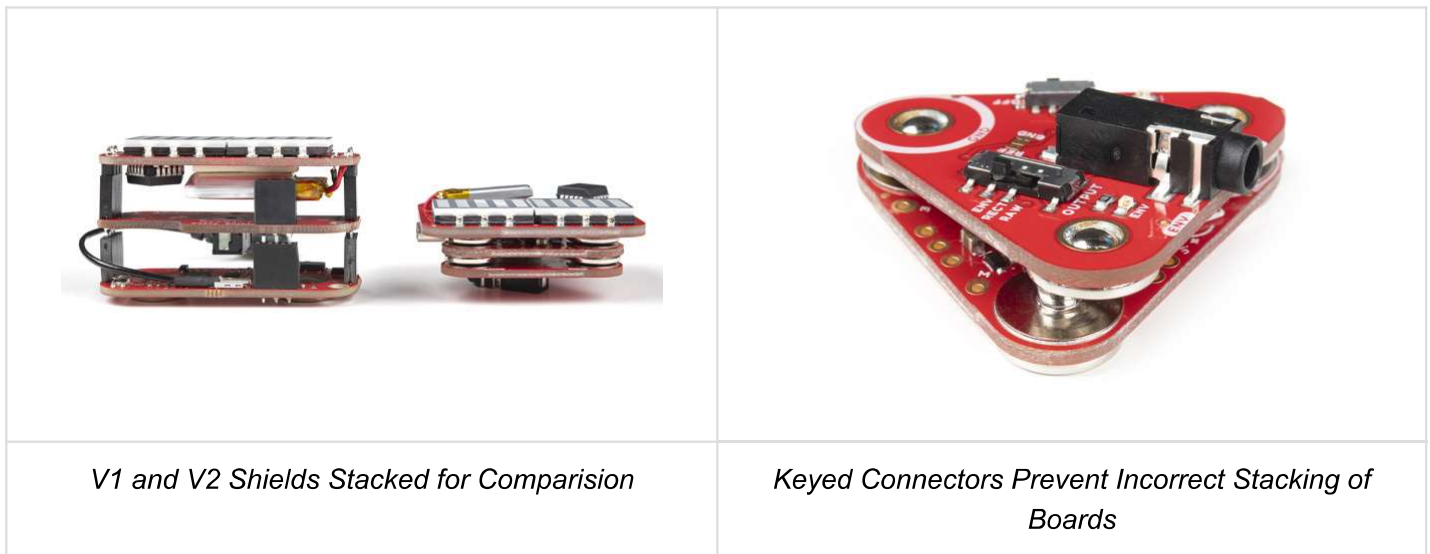
### Installing Board Definitions in the Arduino IDE

How do I install a custom Arduino board/core? It's easy! This tutorial will go over how to install an Arduino board definition using the Arduino Board Manager. We will also go over manually installing third-party cores, such as the board definitions required for many of the SparkFun development boards.

## New Features

The MyoWare 2.0 ecosystem builds upon the previous muscle sensor designs. The major changes to the ecosystem include:

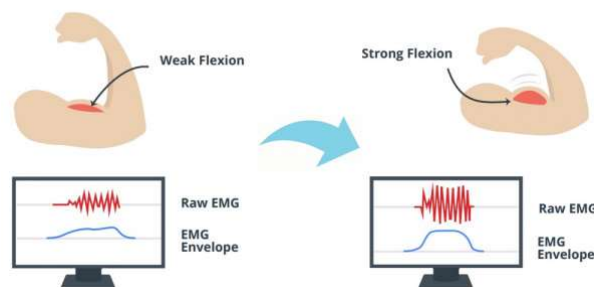
- **Updated Circuitry** - The MyoWare 2.0 Muscle Sensor has been redesigned from the ground up! The circuitry is updated with the latest and greatest chipset improving sensor performance and reliability.
- **Innovative Snap Connector System** - The built-in male and female snap connectors on the sensor and its shields eliminates the need to solder connections! Topside connectors link the shields to the power and the EMG envelope output of the sensor. Bottom side link to the input electrodes for the Cable Shield.
- **Compact** - The snap connectors also allow for a lower profile when stacking shields to the MyoWare 2.0 Muscle Sensor.
- **Keyed** - The ground and reference snap connectors on the boards are slightly offset and designed to be "keyed." Since the muscle sensor and shields are keyed, there's only one way to connect the boards together!
- **New Shields** - The MyoWare 2.0 Link and Arduino Shields were designed to easily connect up to six MyoWare 2.0 Muscle Sensors to a development board with the Arduino Uno R3 footprint. This also eliminates the need to solder connections when connecting the muscle sensors to an Arduino. You'll just need one TRS-to-TRS cable for each muscle sensor.
- **MyoWare Power Shield** - The previous version of the Power Shield used coin cell batteries. The updated MyoWare 2.0 Power Shield includes a built-in rechargeable LiPo battery.
- **MyoWare Proto Shield (Deprecated)** - With a solderless system in mind, the MyoWare Proto Shield is depreciated in the MyoWare 2.0 ecosystem.



## What is electromyography (EMG)?

"The MyoWare measures muscle activity through the electric potential of the muscle, commonly referred to as surface electromyography (*EMG* or *sEMG* for short). When your brain tells your muscle to flex, it sends an electrical signal to your muscle to start recruiting motor units (*the bundles of muscle fibers that generate the force behind your muscles*).

The harder you flex, the more motor units are recruited to generate greater muscle force. The greater the number of motor units, the more the electrical activity of your muscle increases. The MyoWare will analyze this electrical activity and output an analog signal that represents how hard the muscle is being flexed. The harder you flex, the higher the MyoWare output voltage will go." -Advancer Technologies



*Image Courtesy of Advancer Technologies taken from the MyoWare 2.0 Advanced Guide*

## MyoWare 2.0 Muscle Sensor

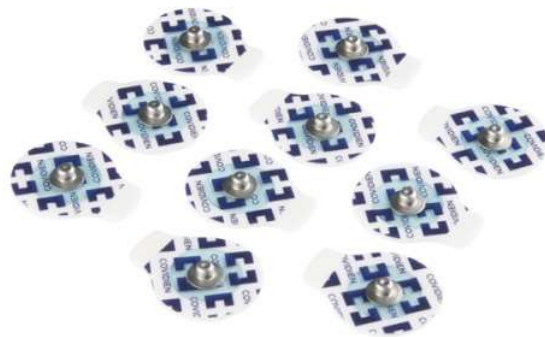
The MyoWare 2.0 Muscle Sensor is an Arduino-compatible, all-in-one electromyography (EMG) sensor from Advancer Technologies! The innovative snap connector system eliminates the need to solder connections for the MyoWare 2.0 ecosystem. It's that easy: stick on a few electrodes (not included), read the output voltage and flex some muscles! The muscle sensor's snap connector system makes it easier to stack shields together. The top

side connectors link to power and the sensor's EMG envelope output while the bottom side links to the input electrodes. Measuring muscle activity by detecting its electric potential has traditionally been used for medical research. However, with the advent of ever shrinking yet more powerful microcontrollers and integrated circuits, EMG circuits and sensors have found their way into all kinds of control systems such as video games, robotics, and prosthetics!



MyoWare 2.0 Muscle Sensor  
○ DEV-18977

The embedded snap connectors mate well with our biomedical sensor pad (10 pack).

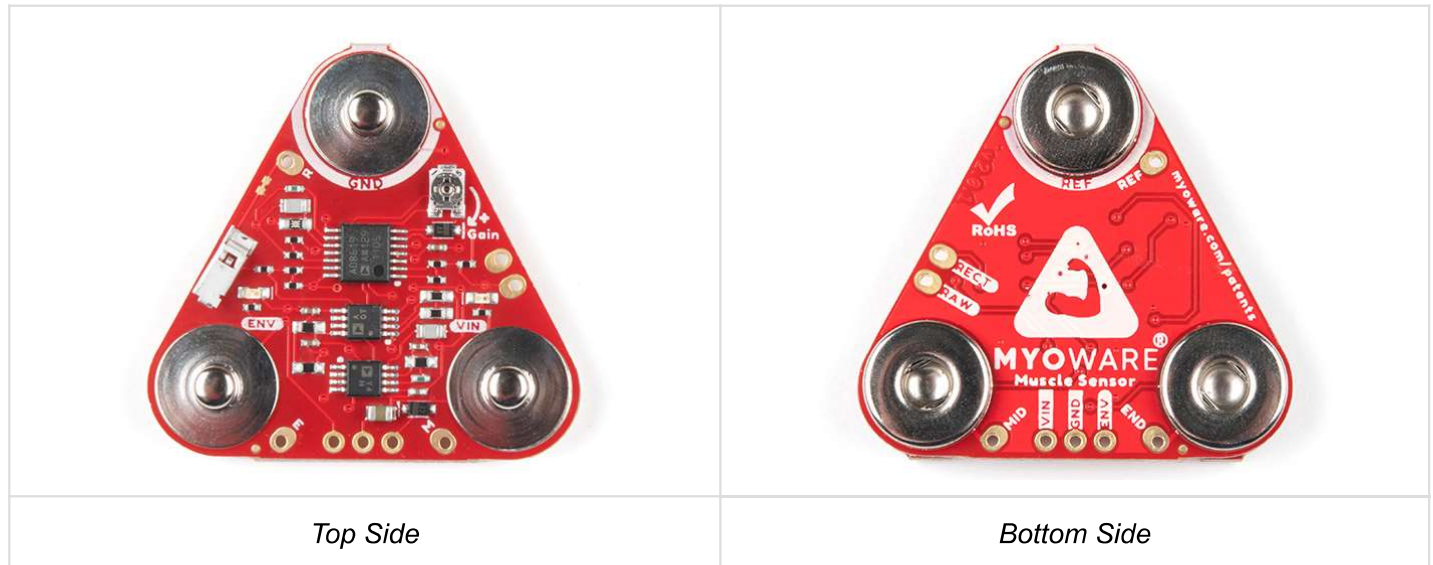


Disposable Surface EMG/ECG/EKG Electrode - 24mm  
(10 pack)  
● SEN-12969

## Hardware Overview

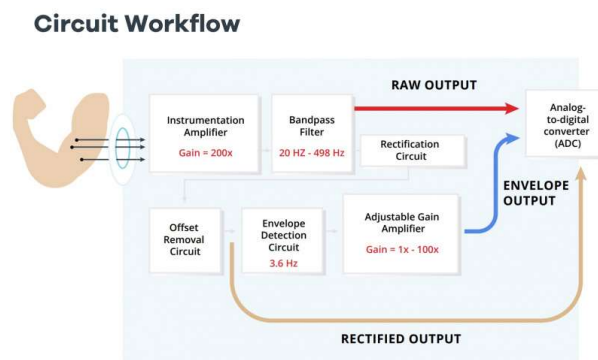
**Note:** The male snap connectors on the top side of the board are referred to as "stud" connectors while the female snap connectors on the bottom side of the board are referred to as "socket" connectors. You may notice this terminology in the technical specifications or guides that reference these snap connectors.

The top side of the board is populated with the circuit and male snap connectors. The bottom of the board has the female snap connectors.



## Amplifiers and Adjustable Gain

The MyoWare 2.0 Muscle Sensor measures a small EMG reading from the muscle group. This signal goes through an amplifier, bandpass filter, rectifier, and envelope detector so that users can easily read the muscle activity through a microcontroller's ADC pin.

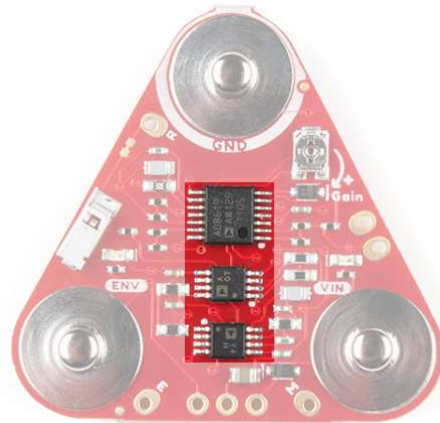


*Image Courtesy of Advancer Technologies taken from the MyoWare 2.0 Advanced Guide*

The ICs highlighted below are amplifiers used to help process the raw EMG signal. As stated from the MyoWare 2.0 Advanced Guide, some of the muscle sensor's technical specifications are as follows:

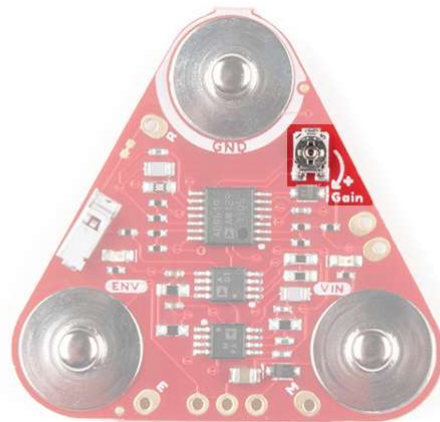
- **Voltage Input**
  - min. = +2.27V
  - typ. = +3.3V or +5V
  - max. = +5.47V
- **Input Bias Current**

- 250 pA, max 1 nA
- **Input Impedance**
  - 800
- **Common Mode Rejection Ratio (CMRR)**
  - 140dB
- **Filters**
  - **High-pass Filter:** Active 1st order,  $f_c = 20.8$  Hz, -20dB
  - **Low-pass Filter:** Active 1st order,  $f_c = 498.4$  Hz, -20dB
- **Rectification Method**
  - **Full-wave**
- **Envelope Detection**
  - **Linear:** Passive 1st order,  $f_c = 3.6$  Hz, -20 dB



The board includes a potentiometer (trim pot) to manually adjust the gain of the envelope signal. You will need a screwdriver to adjust. Of course the top shield will need to be disconnected to access the potentiometer. The gain (G) for the following signals are as follows:

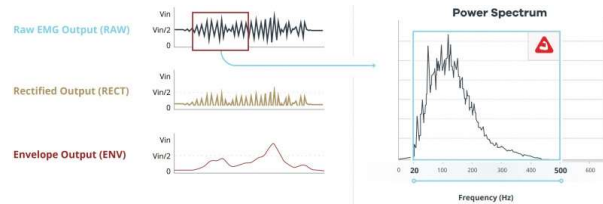
- **Raw (RAW):**  $G = 200$
- **Rectified (RECT):**  $G = 200$
- **Envelope (ENV):**  $G = 200 * (R / 1 \text{ k}\Omega)$ , where R is the resistance of the gain potentiometer in  $\text{k}\Omega$



**Note:** The default gain should be sufficient for most applications. However, advanced users can adjust the trim pot for the ENV's output to the desired gain setting using a screwdriver. Turn the trim pot clockwise to increase the gain. Turn the trim pot counterclockwise to reduce the gain. Take care when rotating the trim pot since it can be damaged by over rotating or applying too much pressure.



The image below shows the waveforms for the raw, rectified, and envelope outputs. The graph to the right of the outputs show the raw output's power spectrum. The MyoWare 2.0 Muscle Sensor has a first-order bandpass filter ranging between 20Hz - 500Hz. This range is ideal for capturing the the raw EMG signal and removing any unwanted signals such as motion artifacts. Note that the data shown is for illustration purposes only and not the actual data.



*Image Courtesy of Advancer Technologies taken from the MyoWare 2.0 Advanced Guide*

## PTH Pads

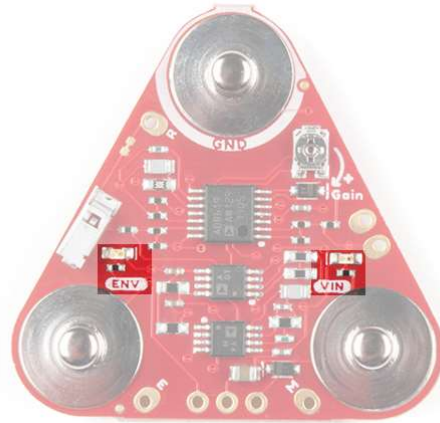
Flipping the board to the bottom side, you will see PTH pads labeled. For those that need to solder wire to connect VIN, GND, and ENV, we have included a 1x3 row of 0.1" spaced PTH pads on the board. Additionally, the RAW, RECT, REF, MID, and END pins are broken out throughout the board. When soldering wire to the REF, RECT, and RAW pins, care must be taken to avoid adding excessive solder to the pins as this may prevent the MyoWare 2.0 Link Shield's pogo pins from connecting properly to the PTH pads.



## LEDs

The board includes two status LEDs.

- **VIN** - The VIN LED lights up when the power switch is flipped to the ON position to indicate when the board is providing power for the MyoWare 2.0 Muscle Sensor through the VIN snap pin.
- **ENV** - The ENV pin lights up when there is activity from the MyoWare 2.0 Muscle Sensor's ENV pin.



## Snap Connectors

The board includes snap connectors to easily stack shields on the MyoWare 2.0 Muscle Sensor. The top of the board is slightly offset and designed to act like a key so there's only one way to connect the boards together. Align the GND or REF snap connectors before stacking the boards together. Otherwise, the three snap connectors will fail to connect.

The following male snap connectors are on the top side of the board. These are also broken out on the 1x3 header on the bottom of the board. Flip the board over to view the pin labels.

- **GND** - Ground for the MyoWare 2.0 Muscle Sensor.
- **VIN** - Voltage input for the MyoWare 2.0 Muscle Sensor. This voltage depends on what shield is stacked on top when using the snap connectors. Make sure to leave the VIN pin's PTH pad open when stacking a shield on top to avoid conflicting voltages from two power sources.
  - When using it with the MyoWare 2.0 Power Shield or LED Shield, it will depend on how much the LiPo battery is charged up. When plugged in to a power source for charging, the charge IC may start charging the LiPo battery and system voltage will be about 4.2V. For safety reasons, we recommend that users always disconnect this shield from the sensor when charging the battery.
  - When using it with the MyoWare 2.0 Link Shield and MyoWare 2.0 Arduino Shield, it depends on the position of the Power Jumper on the MyoWare 2.0 Arduino Shield. This sets VIN to 5V (default) or 3.3V.
- **ENV** - Envelope signal ranging between 0-VIN. Connect this to an ADC on your microcontroller.



**Note:** Note that the PTH pad labeled "E" in the above photo connects to the END muscle snap connector and *not* the ENV snap connector.

The following female snap connectors are on the bottom side of the board. These are broken out onto a PTH pad next to each snap connector.

- **REF** - Reference electrode snap connector. Connect this to a separate section of the body such as a bony portion of the elbow or a nonadjacent muscle near the muscle group.
- **MID** - Middle muscle electrode snap connector. Connect to the middle of the muscle group.
- **END** - End muscle electrode snap connector. Connect to the end of the muscle group.

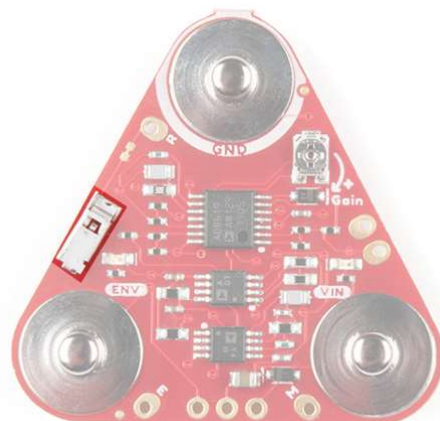


**Note:** Ensure that the REF electrode pad is making a solid, stable connection to the skin or it could cause performance issues.

## Reference Electrode Port

**Note:** The built-in reference snap connector should be sufficient for most applications. However, for users having a difficult time connecting to a muscle group, we recommend getting the reference cable. This is mainly intended for large muscle groups (such as the quad, hamstring, etc.).

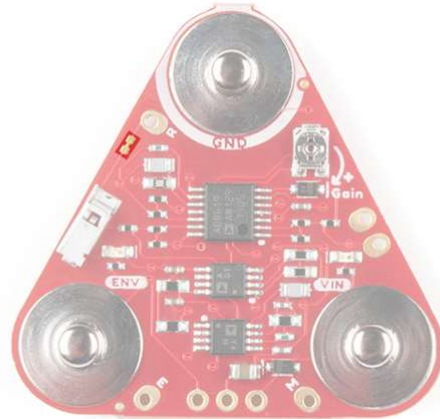
The board has a housing to insert the reference cable's crimped pin. By inserting the pin to its housing and placing a biomedical sensor pad into the electrode snap, you can stick the reference cable to a separate section of the body such as a bony portion of the elbow or a nonadjacent muscle near the targeted muscle where the built-in reference pin is unable to reach.



**Note:** The reference snap connector on the board remains active even with the cable inserted. Disable the socket by cutting the jumper trace on the top side of the sensor. Re-enable the socket by shorting the jumper pads with solder. For more information on modifying the jumpers, check out our tutorial on working with jumper pads and PCB traces.

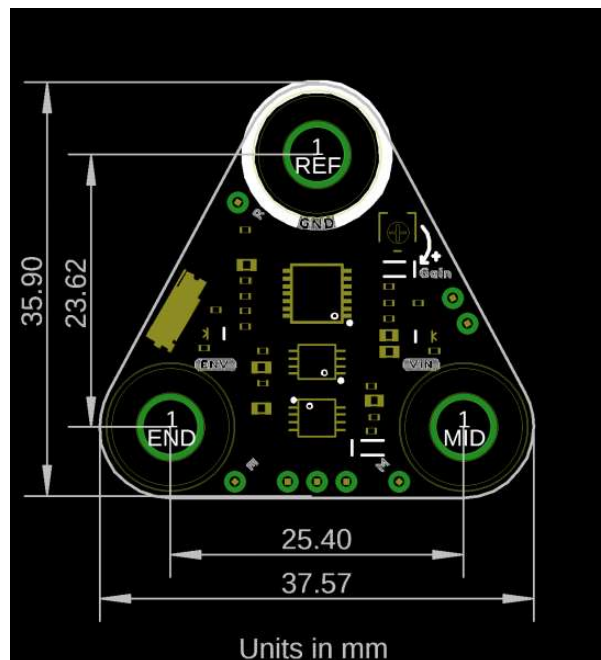
## Jumper

There is a small jumper beside the reference pin labeled "R". This jumper is closed by default and connects to the reference electrode connector. Make sure to cut this trace when using the reference cable. For more information on modifying the jumpers, check out our tutorial on working with jumper pads and PCB traces.



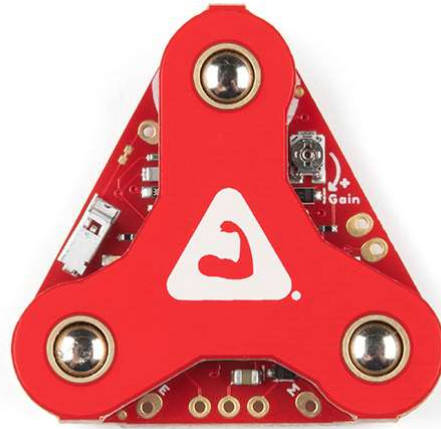
## Board Dimensions

The MyoWare 2.0 Muscle Sensor board dimension is 37.57mm x 35.90mm (1.48" x 1.41").



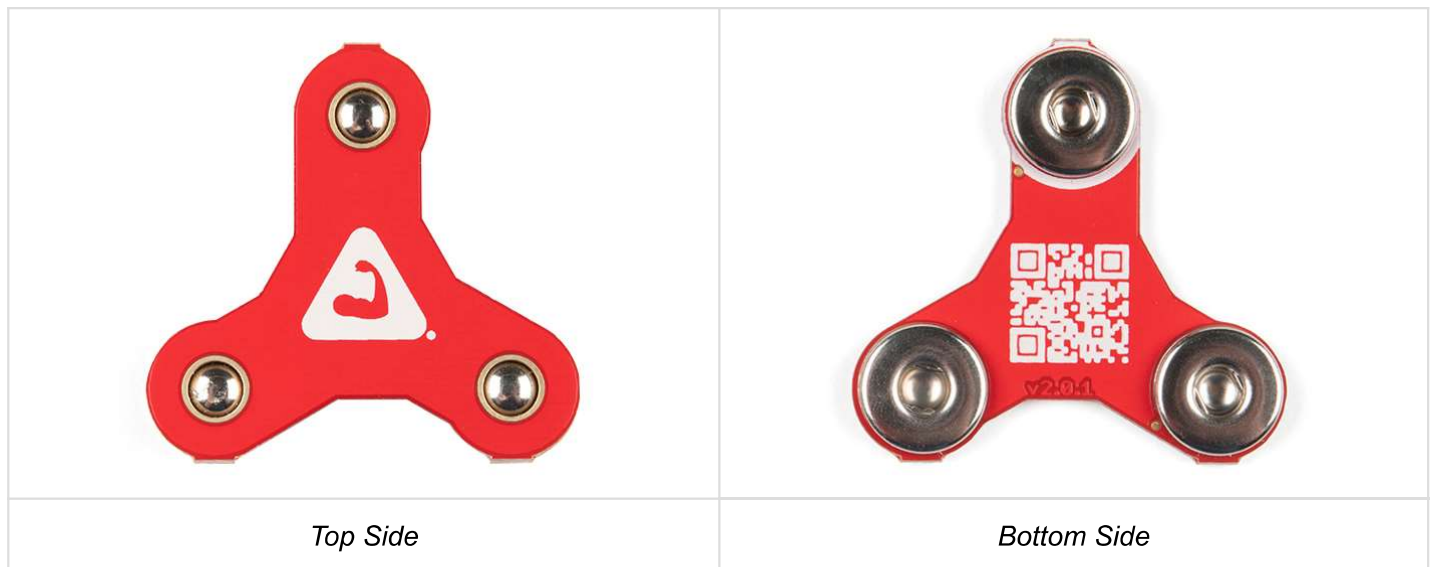
## MyoWare 2.0 Sensor Cover

The MyoWare 2.0 Sensor Cover comes pre-installed with each MyoWare 2.0 Muscle Sensor. The cover should remain on the sensor to protect the circuitry unless you decide to stack shields on top.



## Hardware Overview

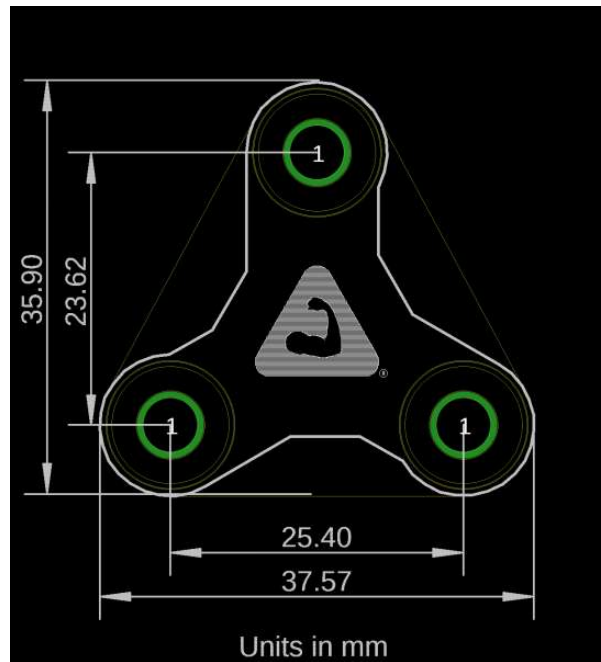
There's not much going on with this board besides the snazzy design. It looks similar to a fidget spinner... well, that is without the press fit skateboard bearings built into the board. There is no circuitry populated on the board. On the top of the board, you will find the MyoWare logo from Advancer Technologies. On the bottom, you will find a QR code on the bottom of the board. Feel free to take out your smartphone and scan the QR code to head on over MyoWare's official site!



## Board Dimensions

The sensor cover uses the MyoWare 2.0 Muscle Sensor form factor and has board dimensions of 37.57mm x 35.90mm (1.48" x 1.41").

---



## MyoWare 2.0 Reference Cable

The MyoWare 2.0 Reference Cable is meant as an extension for the MyoWare 2.0 Muscle Sensor's reference pin. By inserting the pin to its housing and placing a biomedical sensor pad into the electrode snap, you can stick the reference cable to a separate section of the body such as a bony portion of the elbow or a nonadjacent muscle near the targeted muscle where the built-in reference pin is unable to reach.



## MyoWare 2.0 Reference Cable

© CAB-19244



## MyoWare 2.0 Reference Cable - 10 pack

● KIT-19627

### Hardware Overview

**Note:** The built-in reference snap connector should be sufficient for most applications. However, for users having a difficult time connecting to a muscle group, we recommend getting the reference cable. This is mainly intended for large muscle groups (such as the quad, hamstring, etc.).

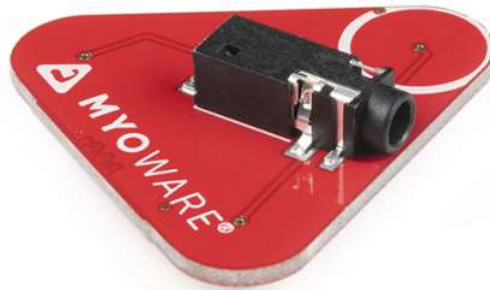
One end of the MyoWare 2.0 Reference Cable has a crimped pin with a tab. This is meant to be inserted into its respective pin housing on the MyoWare 2.0 Muscle Sensor. The tab will lock the pin in place. The other end has a snap pin to connect to a biomedical sensor pad. The cable is about 3 inches long.



**Note:** The reference snap connector on the board remains active even with the cable inserted. Disable the socket by cutting the jumper trace on the top side of the sensor. Re-enable the socket by shorting the jumper pads with solder. For more information on modifying the jumpers, check out our tutorial on working with jumper pads and PCB traces.

## MyoWare 2.0 Cable Shield

The MyoWare 2.0 Cable Shield is designed for cases where you would want to mount the sensor pads away from the MyoWare 2.0 Muscle Sensor. This allows the users to test and use the muscle sensor without actually attaching the board to the target muscle. The shield includes a 3.5mm TRS jack where you can attach a traditional three electrode sensor cable. The board is also equipped with snap connectors on the board so you can easily stack it. Unlike the other shields in the MyoWare 2.0 ecosystem, this board snaps to the bottom side of the MyoWare 2.0 Muscle Sensor. You'll want to use this shield for non-wearable projects that are housed in an enclosure or that target muscles that are too small or situated such that the on-board electrode sockets would not be suitable.



### MyoWare 2.0 Cable Shield

● DEV-18386

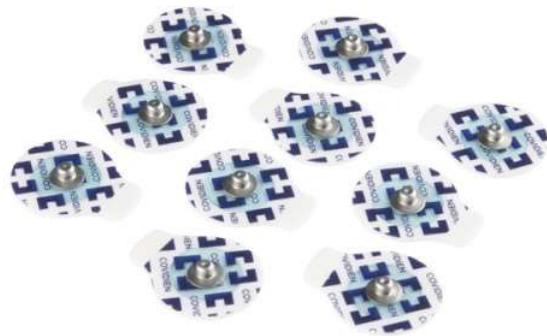
Of course, you will need a 3.5mm TRS connector to access the mid, end, and reference ground pins along with a few biomedical sensor pads. Make sure to grab the following products if you do not have them already.





Sensor Cable - Electrode Pads (3 connector)

● CAB-12970



Disposable Surface EMG/ECG/EKG Electrode - 24mm  
(10 pack)

● SEN-12969

Or you could grab a TRS plug to make an adapter using conductive material.



TRS Audio Plug - 3.5mm (Metal)  
● COM-15438

**Note:** To make an adapter, try looking at the following tutorial from Advancer Technologies to make a conductive fabric electrode.

## Hardware Overview

**Note:** The male snap connectors on the top side of the board are referred to as "stud" connectors. You may notice this terminology in the technical specifications or guides that reference these snap connectors.

While the default image for the MyoWare 2.0 Cable Shield's product page shows the side with the 3.5mm TRS connector as the first image, the "top" of this board is the side with male snap connectors. Compared to the other shields in the MyoWare 2.0 ecosystem, this is the only shield that stacks to the bottom side of the MyoWare 2.0 Muscle Sensor.



Top Side

Bottom Side

The cable color codes can vary depending on the manufacturer. If you are seeing unusual sensor readings and the MyoWare 2.0 Muscle Sensor is not responding to a muscle group, try testing the pinout by using a multimeter set to measure continuity. Below is a table that references the connections between the male snap pins, TRS connector, and cable. The male snap pins allow users to easily stack to the bottom of the MyoWare 2.0 Muscle Sensor

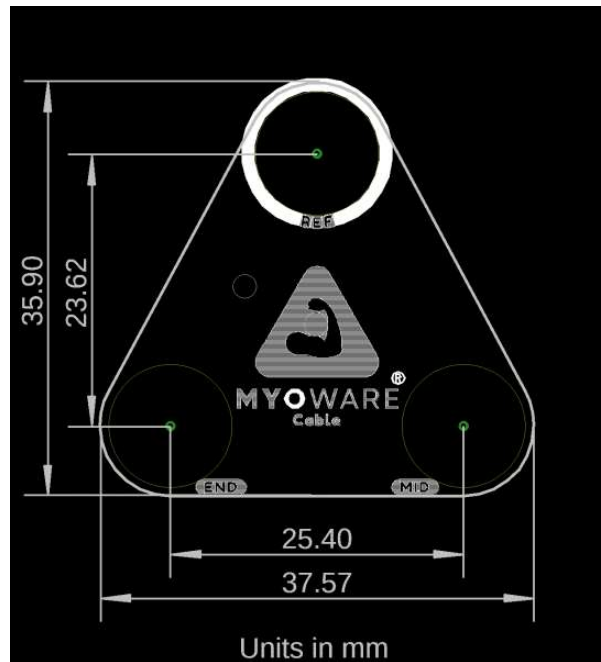
Snap Connector	TRS Pinout	Electrode Pin Color [CAB-12970]	Notes
Reference [REF]	Sleeve	Black	Reference electrode connection. Connect this to a separate section of the body such as a bony portion of the elbow or a nonadjacent muscle near the muscle group.
End [END]	Ring	Blue	End muscle electrode connection. Connect to the end of the muscle group.
Middle [MID]	Tip	Red	Middle muscle electrode connection. Connect to the middle of the muscle group.

**Note:** Ensure that the REF electrode pad is making a solid, stable connection to the skin or it could cause performance issues.

**⚠ Warning:** Do not insert a TRS cable into the MyoWare 2.0 Cable Shield from either the MyoWare 2.0 Arduino Shield or MyoWare 2.0 Link Shield!

### Board Dimensions

The shield uses the MyoWare 2.0 Muscle Sensor form factor and has board dimensions of 37.57mm x 35.90mm (1.48" x 1.41").



## MyoWare 2.0 Power Shield

The MyoWare 2.0 Power Shield is designed to easily power the MyoWare 2.0 Muscle Sensor for remote applications. The MyoWare Power Shield is equipped with snap connectors on the board so you can easily stack it to the top side of the MyoWare Muscle Sensor. Flip the switch to the ON position to give the sensor all the power it needs to work its myoelectric magic. Connecting the MyoWare 2.0 Muscle Sensor to battery power allows for a cleaner signal while also eliminating the possibility of creating a dangerous current path to the power grid. Use it to power your sensor in portable applications.



## MyoWare 2.0 Power Shield

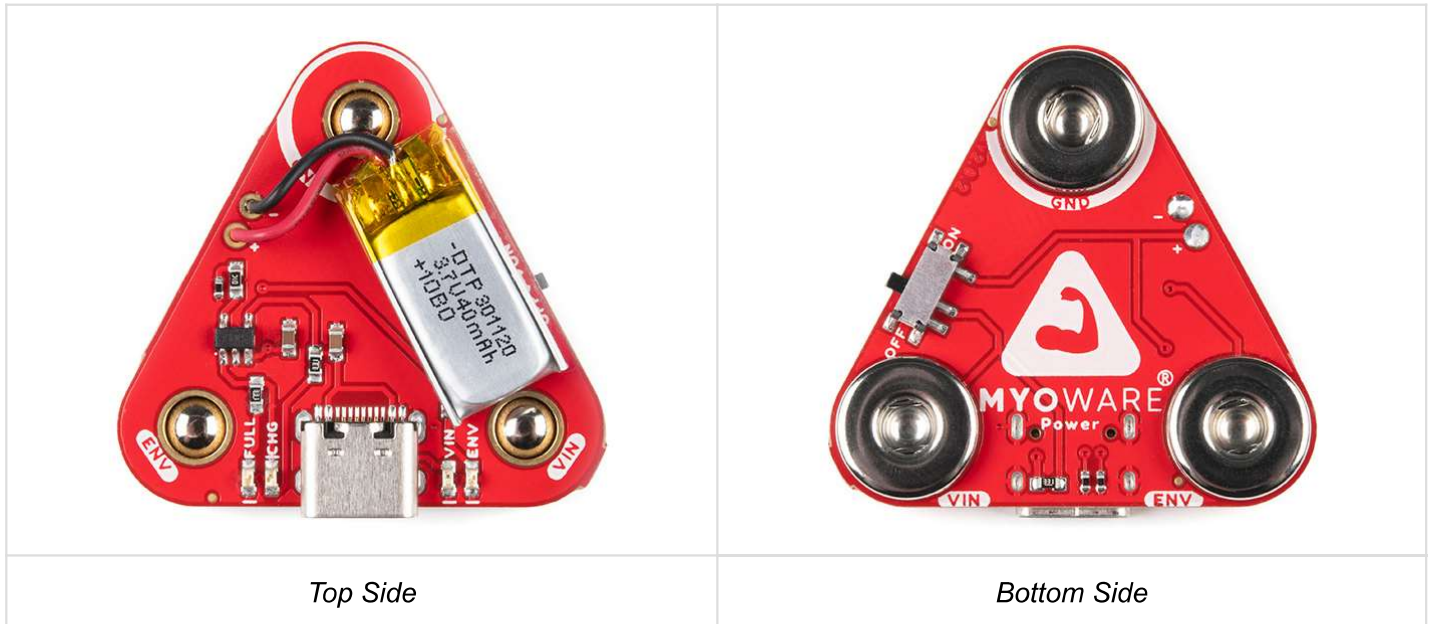
© DEV-18427

## Hardware Overview

---

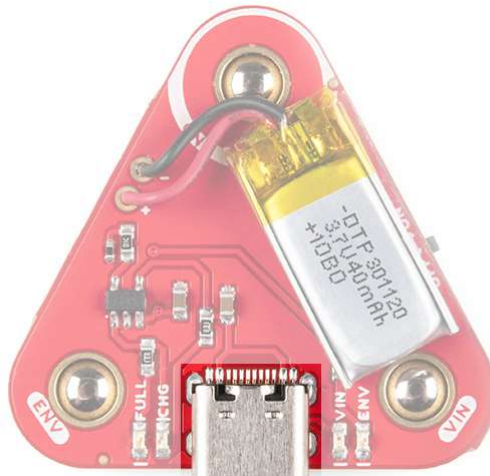
**Note:** The female snap connectors on the bottom side of the board are referred to as "socket" connectors. You may notice this terminology in the technical specifications or guides that reference these snap connectors.

The top side of the board has the LiPo battery and the USB connector. The bottom side has the female snap connectors and a power switch.



## USB C Connector

Included on the board is a USB Type C connector. Plug in a USB C cable to a 5V power source to charge the single cell, LiPo battery.



**⚡ Danger:** We recommend that you always remove the MyoWare 2.0 Power Shield from the MyoWare 2.0 Muscle Sensor whenever you are charging its LiPo battery with a computer or USB Charger. The MyoWare 2.0 Power Shield was not designed to be used with the MyoWare 2.0 Muscle Sensor while charging. If the power switch is in the ON position while you are charging, it will compromise the integrity of the output signal. Do not charge the LiPo battery when the shield is stacked on top of the sensor and attached to a user. This

will safeguard against electrical shock so that users are not connected to a power grid (such as a computer power supply and USB charger connected to a wall outlet).

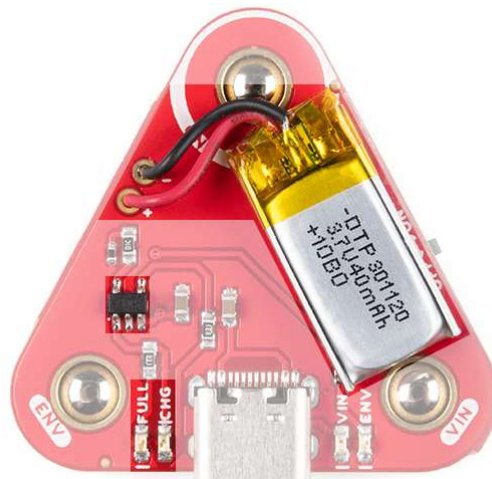
## Power Switch

The board includes a power switch. Flip the switch to the ON position to provide power to the MyoWare 2.0 Muscle Sensor via the VIN female snap connector. When not in use, flip the switch to the OFF position.

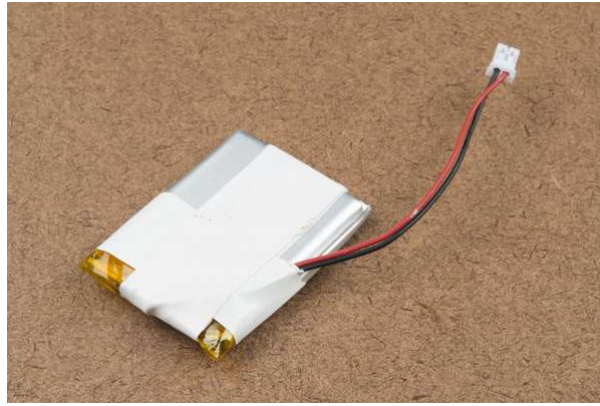


## LiPo Charger

The board includes the MCP73831 LiPo charger IC (the little black IC with 5 pins) to safely charge a single cell, LiPo battery. In this case, the **charge rate is set to 40mA** to charge the built-in 40mAh LiPo battery. The **charge status** of your battery is indicated by the charge (CHG) and full (FULL) LEDs.



**Note:** For more information on proper handling of LiPo batteries, check out the LilyPad Basics: Powering Your Project - LiPo Battery Safety Care.



## Snap Connectors

The board includes three female snap connectors to easily stack on top of the MyoWare 2.0 Muscle Sensor:

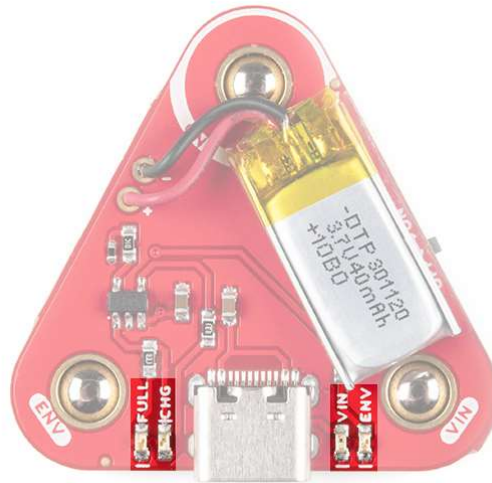
- **GND** - Ground for power.
- **VIN** - Voltage input for power. This voltage depends on how much the LiPo battery is charged up. When plugged in to a power source for charging, the charge IC may start charging the LiPo battery and will be about 4.2V. For safety reasons, we recommend that users always disconnect this shield from the sensor when charging the battery.
- **ENV** - Envelope signal ranging between 0-VIN. Connect this to an ADC on your microcontroller.



## LEDs

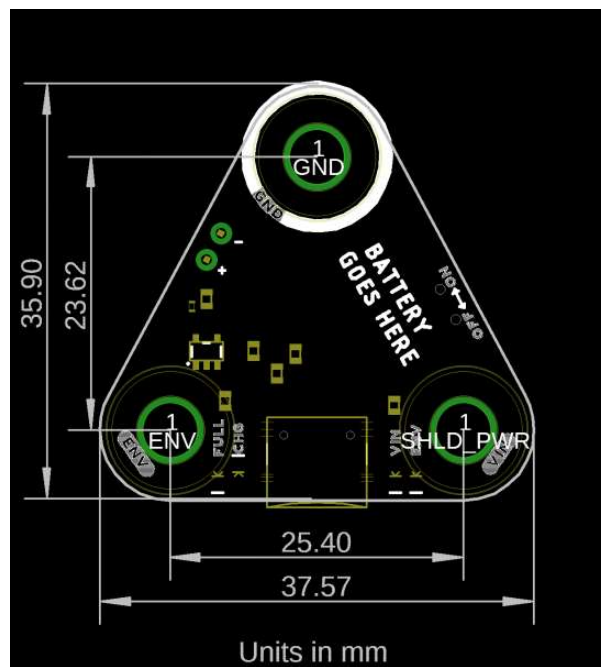
The board includes four LEDs on the top side of the board:

- **FULL** - The FULL LED lights up to indicate when the battery is fully charged. This will be off when the LiPo is not charging or when the charger is not connected to the board.
- **CHG** - The CHG LED lights up to indicate when the LiPo battery is charging. This will be off when the LiPo is fully charged or when the charger is not connected to the board.
- **VIN** - The VIN LED lights up when the power switch is flipped to the ON position to indicate when the board is providing power for the MyoWare 2.0 Muscle Sensor through the VIN snap pin.
- **ENV** - The ENV pin lights up when there is activity from the MyoWare 2.0 Muscle Sensor's ENV pin.



## Board Dimensions

The shield uses the MyoWare 2.0 Muscle Sensor form factor and has board dimensions of 37.57mm x 35.90mm (1.48" x 1.41").



## MyoWare 2.0 LED Shield

The MyoWare 2.0 LED Shield is designed to display the magnitude of a target muscle's signal and power the MyoWare 2.0 Muscle Sensor with its built-in battery. The blue 10-segment bar graph shows the magnitude of the measured signal. The more muscle activation measured, the higher up the board LEDs will go! With this shield, you will be provided with a visual representation of the signals provided by the MyoWare 2.0 Muscle Sensor. The LED shield is equipped with snap connectors on the board so you can easily stack it on the top side of the MyoWare 2.0 Muscle Sensor and flip the switch to the ON position to give the sensor all the power it needs to work its myoelectric magic. Connecting the MyoWare 2.0 Muscle Sensor to the battery power allows for a cleaner signal while also eliminating the possibility of creating a dangerous current path to the power grid. Use it to gauge how hard you're working a muscle during a workout, as a teaching tool, or add some myoelectric flair to your Halloween costume!

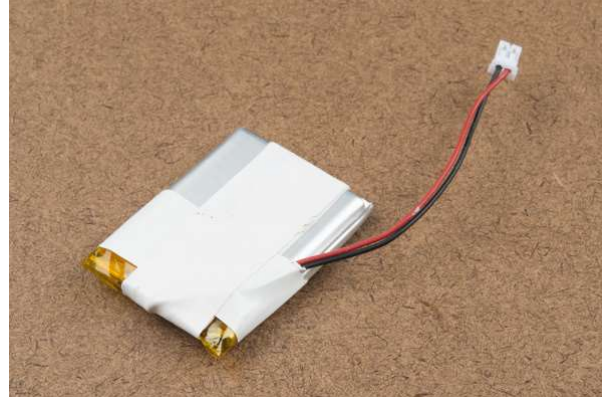




## MyoWare 2.0 LED Shield

© DEV-18387

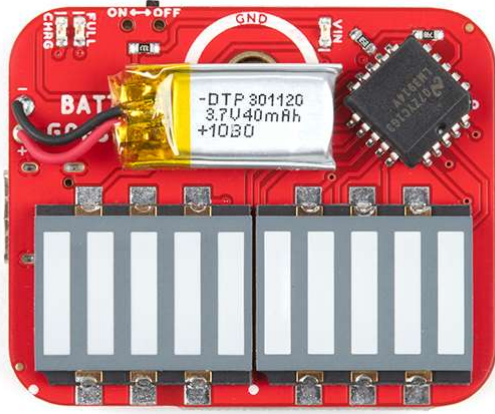
**Note:** For more information on proper handling of LiPo batteries, check out the LilyPad Basics: Powering Your Project - LiPo Battery Safety Care.



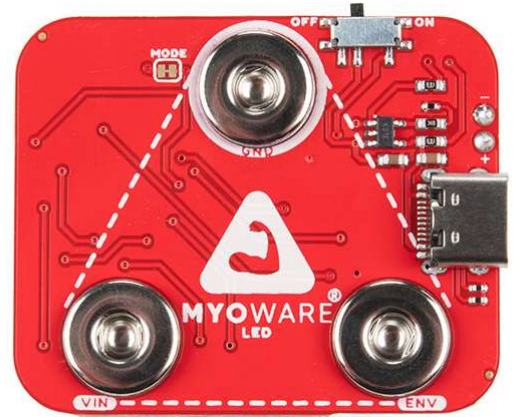
## Hardware Overview

**Note:** The female snap connectors on the bottom side of the board are referred to as "socket" connectors. You may notice this terminology in the technical specifications or guides that reference these snap connectors.

The top side of the board has the LEDs and LiPo battery. The bottom side has the female snap pins, power switch, and USB connector.



*Top Side*



*Bottom Side*

## USB C Connector

Included on the board is a USB Type C connector. Plug in a USB C cable to a 5V power source to charge the single cell, LiPo battery.



**⚡ Danger:** We recommend that you always remove the MyoWare 2.0 LED Shield from the MyoWare 2.0 Muscle Sensor whenever you are charging its LiPo battery with a computer or USB Charger. The MyoWare 2.0 LED Shield was not designed to be used with the MyoWare 2.0 Muscle Sensor while charging. If the power switch is in the ON position while you are charging, all of the LED segments will light up to the max value and compromise the integrity of the output signal. Do not charge the LiPo battery when the shield is stacked on top of the sensor and attached to a user. This will safeguard against electrical shock so that users are not connected to a power grid (such as a computer power supply and USB charger connected to a wall outlet).

## Power Switch

The board includes a power switch. Flip the switch to the ON position to provide power to the MyoWare 2.0 Muscle Sensor via the VIN's female snap connector. When not in use, flip the switch to the OFF position.



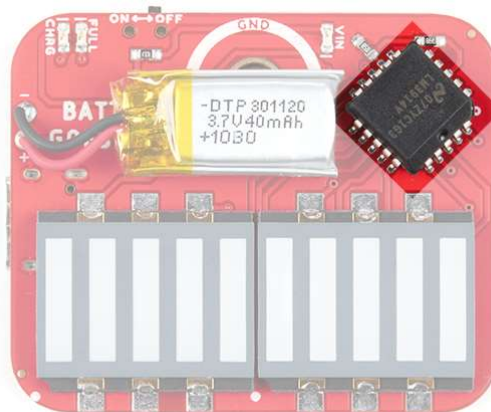
## LiPo Charger

The board includes the MCP73831 LiPo charger IC to safely charge a single cell LiPo battery. In this case, the **charge rate is set to 40mA** to charge the 40mAh LiPo battery on the board. The **charge status** of your battery is indicated by the charge (CHRG) and full (FULL) LEDs.



## LM3914 Dot/Bar Driver

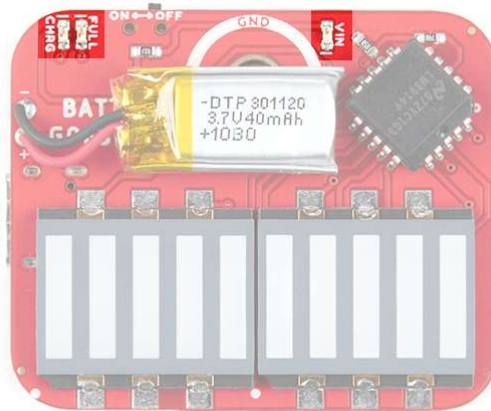
The black square IC is the LM3914 Dot/Bar Driver. This IC takes the output from the muscle sensor and outputs it to the 2x5 segment LEDs.



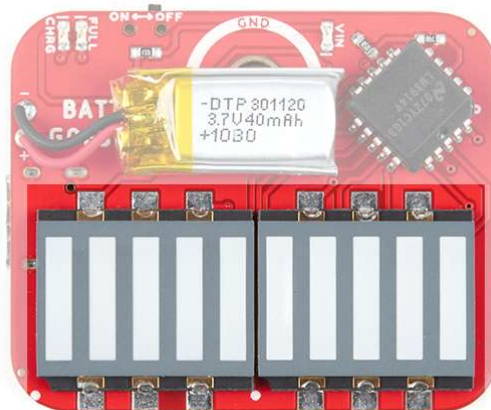
## LEDs

The board includes three status LEDs.

- **FULL** - The FULL LED lights up to indicate when the battery is fully charged. This will be off when the LiPo is not charging or when the charger is not connected to the board.
- **CHRG** - The CHRG LED lights up to indicate when the LiPo battery is charging. This will be off when the LiPo is fully charged or when the charger is not connected to the board.
- **VIN** - The VIN LED lights up when the power switch is flipped to the ON position to indicate when the board is providing power for the MyoWare 2.0 Muscle Sensor through the VIN snap pin.



Of course, this board would not be called the MyoWare 2.0 LED Shield if it did not have the large segment LEDs. The bar graph lights up corresponding to the level of the muscle signal measured. The minimum is closest to the dot bar driver IC. As there is more muscle activity, the voltage will increase from the ENV pin. As a result, the LEDs will continue to light up until the last one on the other side is lit. The LEDs are bright enough to be seen in full light, but really glow nice in low light.



## Snap Connectors

The board includes three female snap connectors to easily stack on top of the MyoWare 2.0 Muscle Sensor:

- **GND** - Ground for power.
- **VIN** - Voltage input for power. This voltage depends on how much the LiPo battery is charged up. When plugged in to a power source for charging, the charge IC may start charging the LiPo battery and will be about 4.2V. For safety reasons, we recommend that users always disconnect this shield from the sensor while charging the battery.
- **ENV** - Envelope signal ranging between 0-VIN. Connect this to an ADC on your microcontroller.



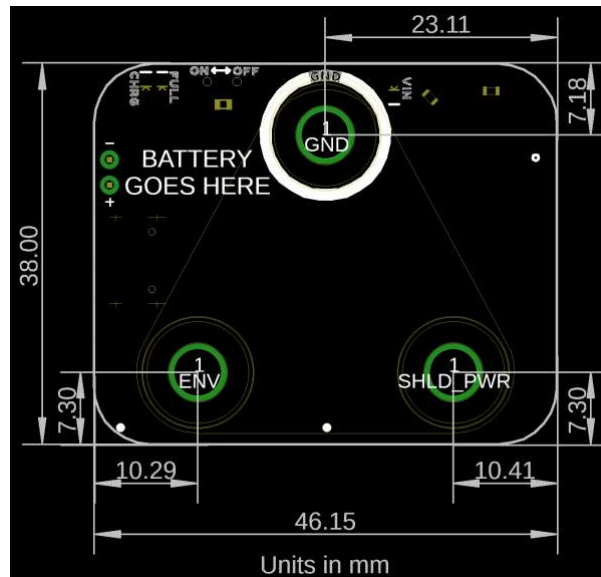
## Jumper

The board includes one jumper for the **MODE** that is closed by default. The MODE is wired directly to LM3914 Dot/Bar Driver's V+ pin for *bar graph mode*. In this mode, the LED segments will continue to light up based on the magnitude of the muscle signal until the last one on the other side is lit. Opening the MODE jumper will turn the board into *dot mode*. When dot mode is enabled, a leading LED bar (i.e. "dot") will light up and traverse the segments. Any trailing LEDs behind it will be turned off instead of filling in with light. For more information on modifying the jumpers, check out our tutorial on working with jumper pads and PCB traces.



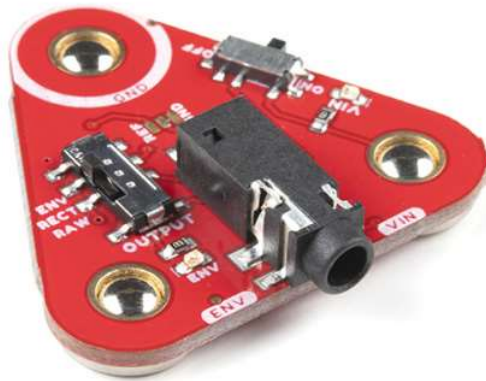
## Board Dimensions

The shield uses the MyoWare 2.0 Muscle Sensor form factor. The board dimension is different than the other boards: 46.15mm x 38.00mm (1.82" x 1.50").



## MyoWare 2.0 Link Shield

The MyoWare 2.0 Link Shield is designed to easily connect a MyoWare 2.0 Muscle Sensor to your Arduino microcontroller. The shield provides a 3.5mm TRS jack so that you can use a 3.5mm TRS-to-TRS cable to link the boards together. The MyoWare 2.0 Link Shield is equipped with snap connectors on the board so you can easily stack it to the top side of the MyoWare 2.0 Muscle Sensor. You'll need the MyoWare 2.0 Arduino Shield and a development board with the Arduino Uno R3 footprint to link each MyoWare 2.0 Muscle Sensor and Link Shield stack.



## MyoWare 2.0 Link Shield

© DEV-18425

## Hardware Overview

**Note:** The female snap connectors on the bottom side of the board are referred to as "socket" connectors. You may notice this terminology in the technical specifications or guides that reference these snap

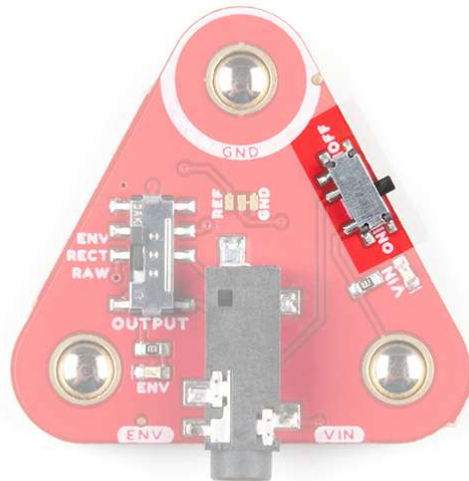
connectors.

The top side of the board has the TRS connector, power and output switches. The bottom side of the board includes the female snap connectors.



## Power Switch

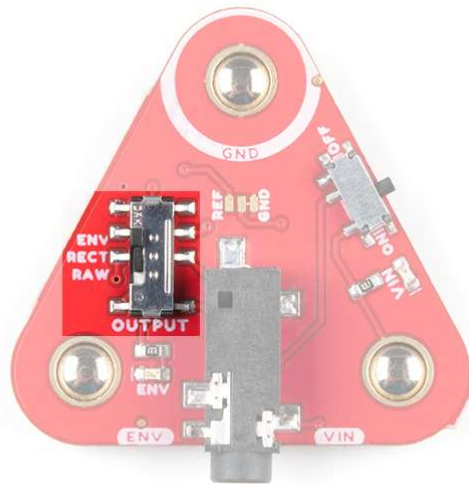
The power switch turns MyoWare 2.0 Muscle Sensor on and off. Power is provided by your Arduino and the Arduino Shield. By default, power is set to 5V on the Arduino Shield. The ring pin of your TRS-to-TRS cable will connect the 5V to VIN when the switch is flipped to the ON position.



## Output Switch

The OUTPUT switch selects the signal to provide for the Arduino's analog input.

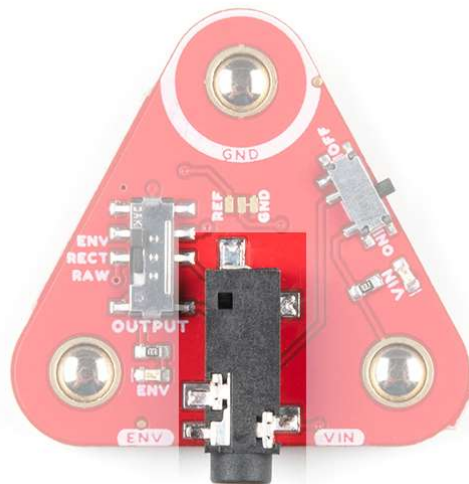
- **ENV** - Envelope output.
- **REC** - Rectified output.
- **RAW** - Raw output.



### 3.5mm TRS Connector

The 3.5mm TRS connector is connected to the following pins.

Pinout	TRS Pin
Ground [GND]	Sleeve
Voltage Input [VIN]	Ring
ENV/RECT/RAW [OUTPUT]	Tip

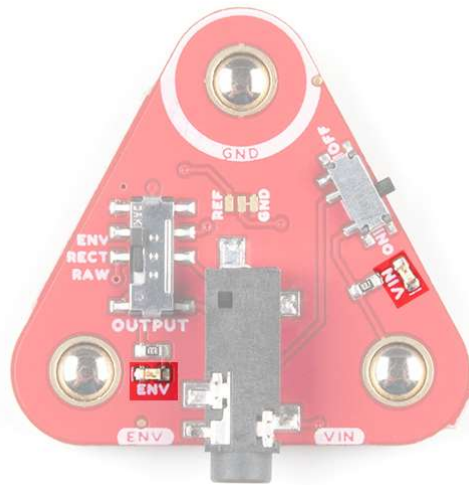


### LEDs

The board includes two LEDs.

- **VIN** - The VIN LED lights up when the power switch is flipped to the ON position to indicate when the board is providing power for the MyoWare 2.0 Muscle Sensor through the VIN snap pin.
- **ENV** - The ENV pin lights up when there is activity from the MyoWare 2.0 Muscle Sensor's ENV pin.





## Snap Connectors

The board includes three female snap connectors to easily stack on top of the MyoWare 2.0 Muscle Sensor:

- **GND** - Ground for power.
- **VIN** - Voltage input for power. This voltage depends on the the the jumper that was set on the MyoWare 2.0 Arduino Shield. This voltage can be 5V (default) or 3.3V depending on the position of the PWR jumper on the MyoWare 2.0 Arduino Shield.
- **ENV** - Envelope signal ranging between 0-VIN. Connect this to an ADC on your microcontroller.



## Pogo Pins

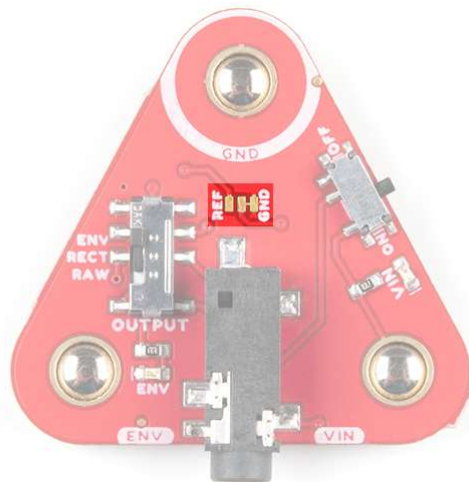
There are three pogo pins on the bottom of the board. These have a much lower profile than the other pogo pins that you may have seen in our storefront. These connect to the MyoWare 2.0 Muscle Sensor's PTH pads. Note that the MyoWare 2.0 Link Shield does not have any labels for the pins. The pogo pin closest to the GND snap connector is the reference pin. The pogo pin just next to the switch is the rectified output pin. The pogo pin just above the VIN snap connector is the raw output.

- **REF** - Reference pin.
  - **RECT** - Rectified output pin.
  - **RAW** - Raw output pin.
-



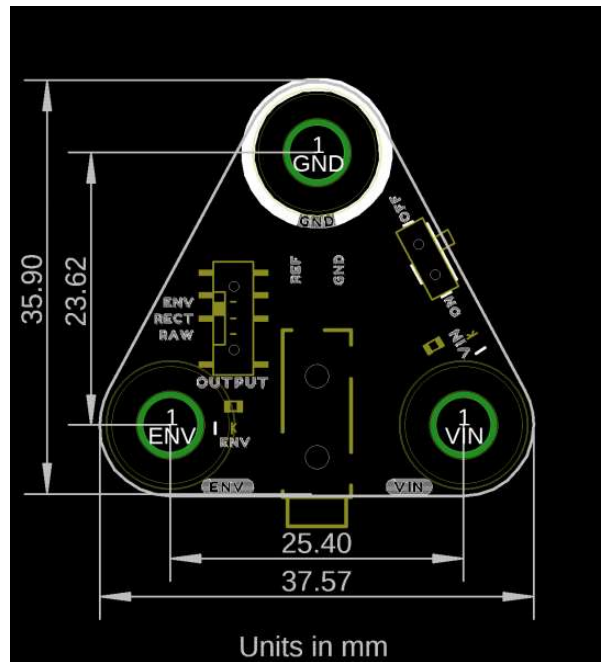
## Jumper

The board includes a 3-way jumper on the top of the board. The point of this jumper is to allow users to change the relative ground point for the output and Arduino. By default, the RAW output will be centered around  $VIN/2$ , which is what REF is. Some users might want to change this so that it is centered around 0V, which would require changing the relative ground point to REF. Cutting the trace connected to GND and adding a solder jumper between the REF and center pad will connect the REF pogo pin to the muscle sensor's REF. For more information on modifying the jumpers, check out our tutorial on working with jumper pads and PCB traces.



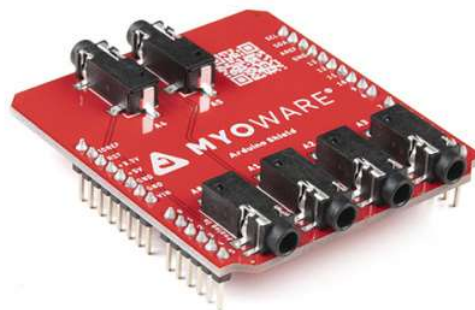
## Board Dimensions

The shield uses the MyoWare 2.0 Muscle Sensor form factor and has board dimensions of 37.57mm x 35.90mm (1.48" x 1.41").



## MyoWare 2.0 Arduino Shield

The MyoWare 2.0 Arduino Shield is designed to easily connect up to six MyoWare 2.0 Muscle Sensors and Link Shields using 3.5mm TRS-to-TRS audio cables. The board comes pre-populated with male headers so all you need to do is stack it on your Arduino and connect the sensors to read up to six muscle groups! The Arduino shield form-factor mates directly with development boards that have the Arduino Uno R3 standard footprint. You'll need a MyoWare 2.0 Link Shield and TRS-to-TRS cable for each MyoWare 2.0 Muscle Sensor.



## MyoWare 2.0 Arduino Shield

© DEV-18426

You'll need a TRS-to-TRS cable to link the boards together. Of course, you could solder wire to a TRS connector as well to make your own. Just make sure to solder wires to the correct pins.



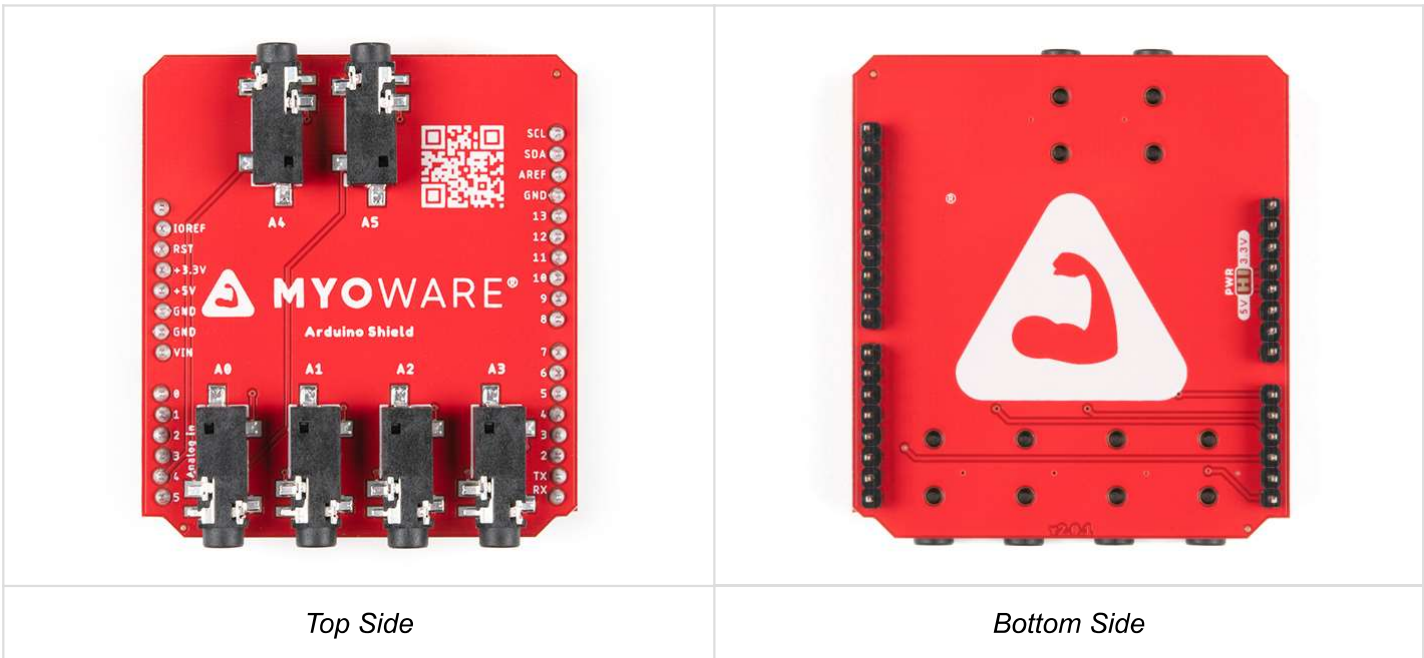
Audio Cable TRS - 1m  
● CAB-18983



TRS Audio Plug - 3.5mm (Metal)  
● COM-15438

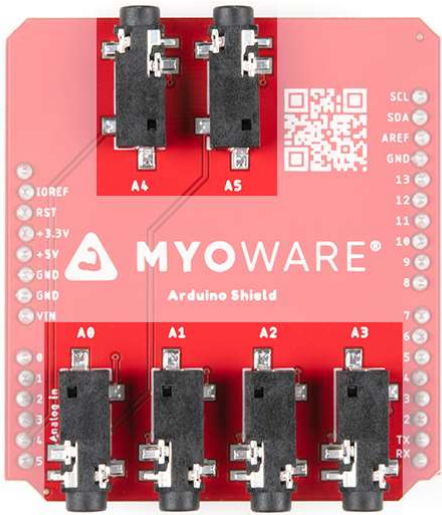
## Hardware Overview

The top side of the board includes the TRS connectors and the QR code. Feel free to take out your smartphone and scan the QR code to head on over MyoWare's official site! The bottom side has the male headers to mate with female headers on Arduino development boards with an R3 footprint.



### 3.5mm TRS Connectors

On the top side of the board you will find six 3.5mm TRS jacks. Each connector is connected to power and your Arduino's analog pins ranging from A0 to A5 .



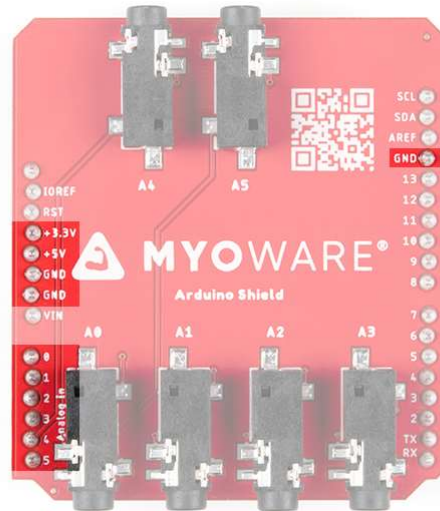
The 3.5mm TRS Connector is connected to the following electrode pins.

Electrode Pin	TRS Pin
Ground [GND]	Sleeve
Voltage [VIN]	Ring
ENV/RECT/RAW [OUTPUT]	Tip

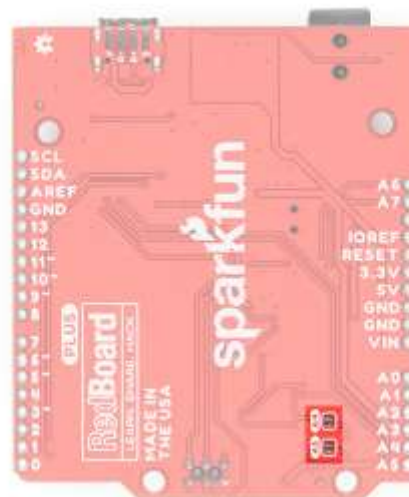
**⚠ Warning:** We recommend removing power from your Arduino before inserting/removing the TRS cable from its connector.

## Reserved Pins

The 5V, 3.3V, GND net, and analog pins ( A0 - A5 ) are reserved for the shield.



**Note:** SparkFun RedBoards with ATmega328P's that have Qwiic enabled connectors use analog pins A4 and A5 for I<sup>2</sup>C. To take analog readings on pins A4 and A5, the jumpers for the Qwiic connector need to be cut. Otherwise, the pull-up resistors for the logic level converters will act as voltage dividers and the pins will read improper values at lower voltages. For more information on modifying the jumpers, check out our tutorial on working with jumper pads and PCB traces.



**Note:** You will notice the PTH pins have an alternating pattern. Don't worry, this was part of the Eagle design with some sneaky footprints. The staggered PTH holes allow our production department to easily solder the headers to the board!

## Jumpers

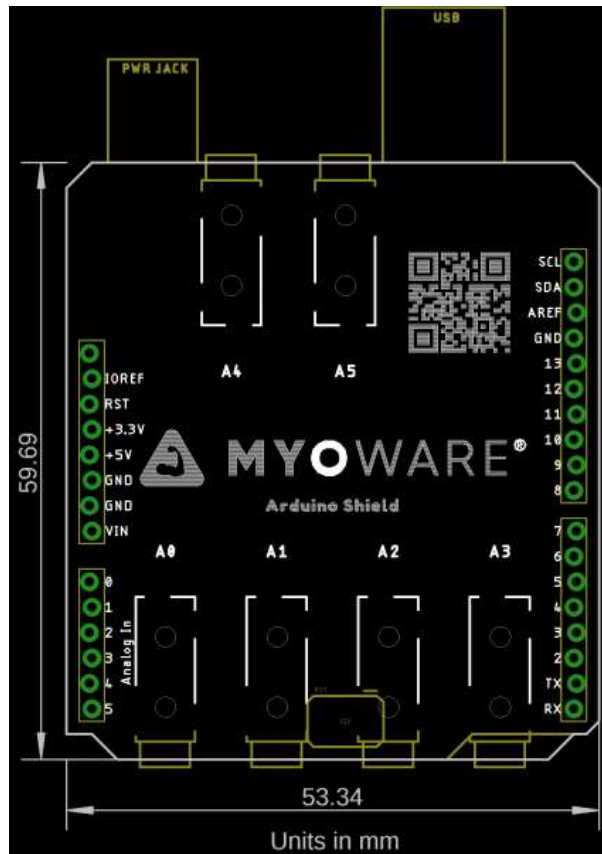
There's not too much going on the bottom of the board. On the back, there is a jumper labeled as PWR . By default, the 3-way jumper connects the 5V pin and the 5V plane on the back of the board. The center pad is connected to the TRS connectors to power any MyoWare 2.0 Muscle Sensors through the Link Shield. Cut the trace and add a

solder blob between the center pad and 3.3V for an alternative voltage. For more information on modifying the jumpers, check out our tutorial on working with jumper pads and PCB traces.



## Board Dimensions

The shield uses the Arduino Uno R3 form factor and has board dimensions of 53.34mm x 59.69mm (2.10" x 2.35").



## Hardware Hookup

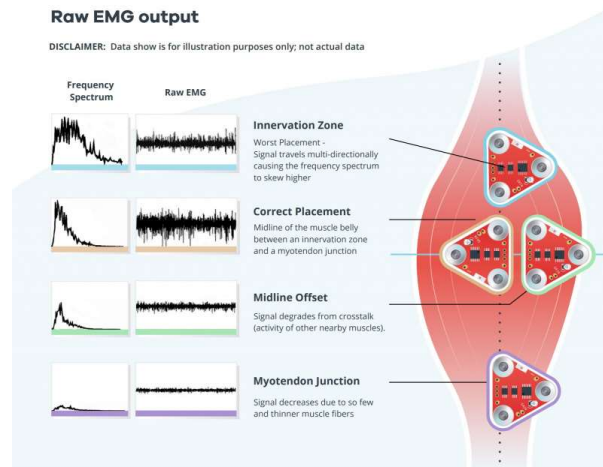
Advancer Technologies has provided a Quick Start and Advanced Guide for the MyoWare 2.0 Muscle Sensor. Feel free to check them out with this guide.

- MyoWare 2.0 Quick Start Guide

- MyoWare 2.0 Advanced Guide

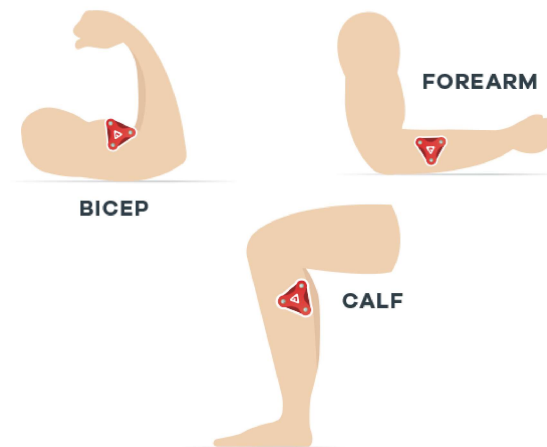
## Selecting a Muscle Group

Select a muscle group to place the MyoWare 2.0 Muscle Sensor. Placement of the sensor is critical when measuring the muscle activity. You will need to connect the MID electrode to the middle of the muscle body with the END electrode lined up in the direction of the muscle length. The REF electrode will be adjacent to the muscle body.



*Image Courtesy of Advancer Technologies taken from the MyoWare 2.0 Advanced Guide*

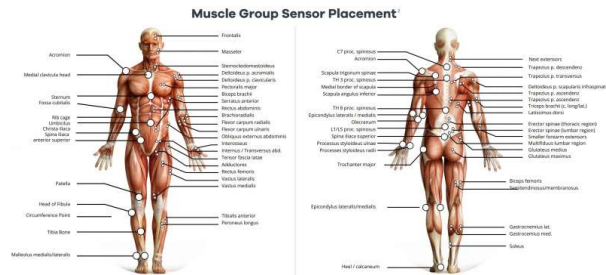
We'll be using the forearm as an example. You can connect the sensor to any muscle group as long as orient the sensor's electrodes with respect to the target muscle as explained above. Note that the muscle group shown in the image below are not to scale.



*Image Courtesy of Advancer Technologies taken from the MyoWare 2.0 Quick Start Guide*

For more other muscle groups, check out the MyoWare 2.0 Advanced Guide. Make sure to zoom in on the image for a closer look at the muscle group. The positions shown in the image are approximate.





*Image Courtesy of Advancer Technologies taken from the MyoWare 2.0 Advanced Guide*

## Preparing the Skin

**Note:** Ensure that there is sparse or no hair on the skin where the EMG pads will stick. If necessary, remove any excess hair. Body hair can cause a poor connection and should be avoided.

Grab an alcohol swab to clean the skin where the MyoWare 2.0 Muscle Sensor will be placed. Cleaning the skin with soap can leave a residue on the skin and should be avoided.



Clean the skin where the EMG pads will stick using an alcohol swab. A cotton ball soaked in isopropyl alcohol is also sufficient to remove any oils or surface contaminants on the skin. Allow the isopropyl alcohol to evaporate before sticking the EMG pads on the skin



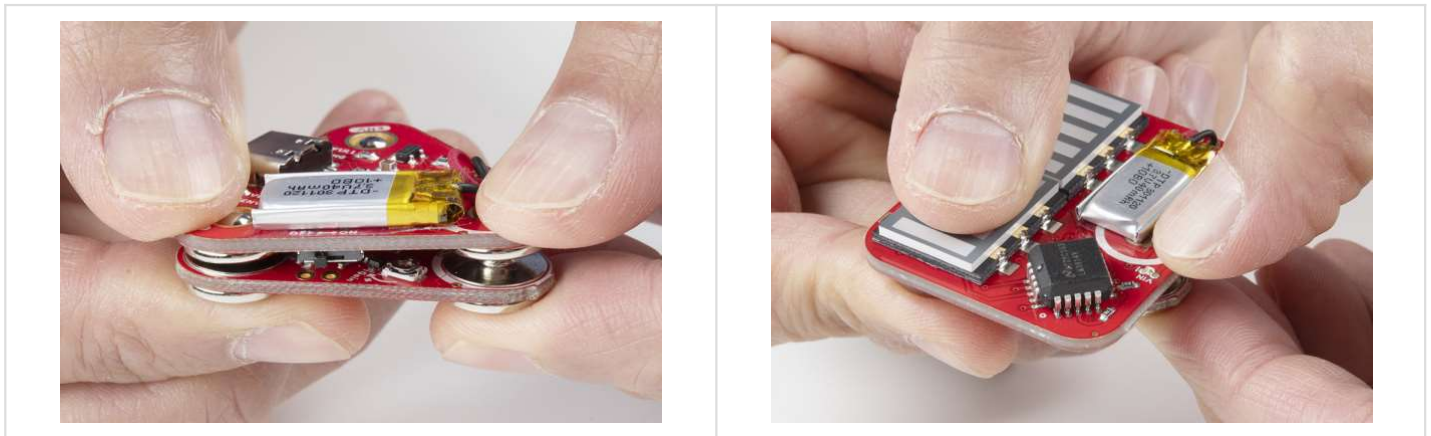
**Note:** We recommend against using the alcohol swab on the face. When cleaning the skin, a brisk wipe is sufficient. Using excessive rubbing alcohol will cause the skin to dry which may cause intermittent readings due to poor contact with the skin.

## Stacking Shields

Since the shields are keyed, there's only one way to stack the boards together! Just look for the snap connector labeled as GND or REF. Align the snap connectors. Make sure the power switch is flipped to the OFF position for shields.

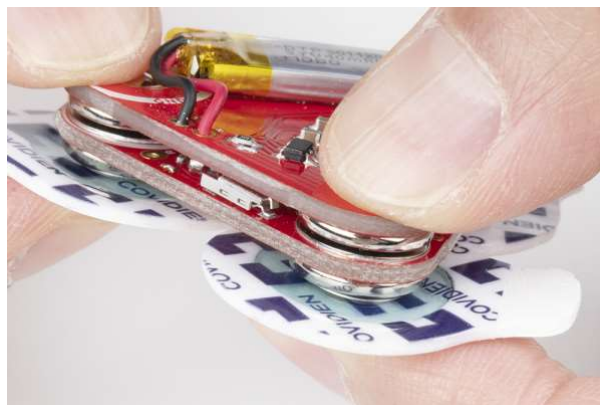


Push down on the snap connectors using your thumb and index finger. Make sure to avoid pressing down on the built-in LiPo battery.



### Connecting to a Muscle Group

Snap the EMG pads to the bottom of the MyoWare 2.0 Muscle Sensor. For users using the MyoWare 2.0 Cable Shield, you'll be snapping it to the sensor cable's snap connectors.



Peel and carefully remove the backs of the electrodes to expose the adhesive.



Attach the pads to the target muscle. In this case, we are using our right forearm.



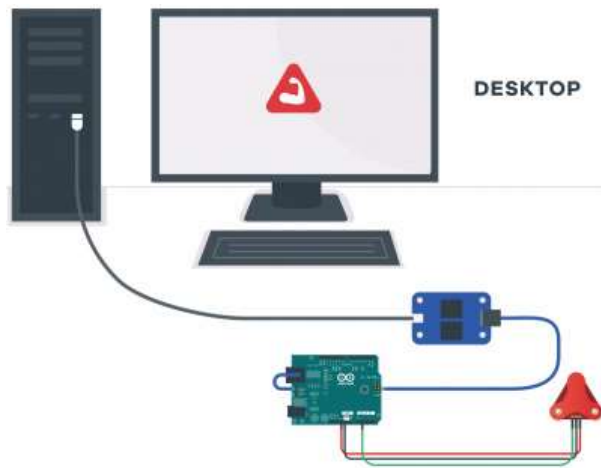
**Note:** As stated in the MyoWare 2.0 Quick Start Guide, we do not recommend snapping the MyoWare 2.0 Muscle Sensor to the electrodes after they've been placed on the skin, doing so can cause bruising!

**Note:** The default gain should be sufficient for most applications. However, advanced users can adjust the trim pot for the ENV's output to the desired gain setting using a screwdriver. Turn the trim pot clockwise to increase the gain. Turn the trim pot counterclockwise to reduce the gain. Take care when rotating the trim pot since it can be damaged by over rotating and overuse.

Ensure that power is disconnected when stacking shields on the top side of the MyoWare 2.0 Muscle Sensor. Once everything is connected, flip the power switch to the ON position on the Power Shield or LED Shield. For users connecting the sensor to an Arduino and computer, check to make sure that your computer is not connected to the wall outlet. Then flip the power switch to the ON position on the Link Shield.

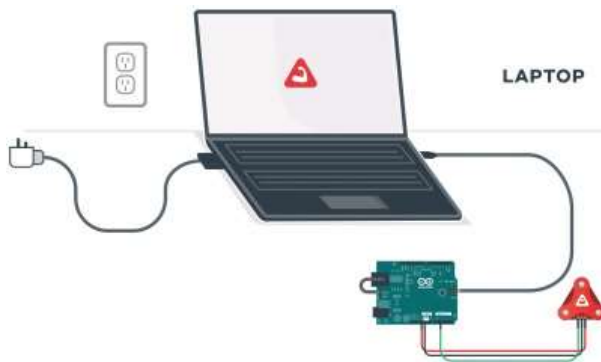
## Connecting Shields to Arduino

**⚡ Danger:** When connecting your MyoWare 2.0 Muscle Sensor to a computer for Serial debugging, it is suggested to have a **USB isolator** with the MyoWare 2.0 Muscle Sensor to prevent noise in the inputs and safeguard against electrical shock when connected to a power grid (such as a computer power supply, computer docks, and external monitors connected to a wall outlet). The diagram below shows USB isolator (as shown as a blue board) between a desktop and the Arduino's USB port.



*Image Courtesy of Advancer Technologies taken from the MyoWare 2.0 Quick Start Guide*

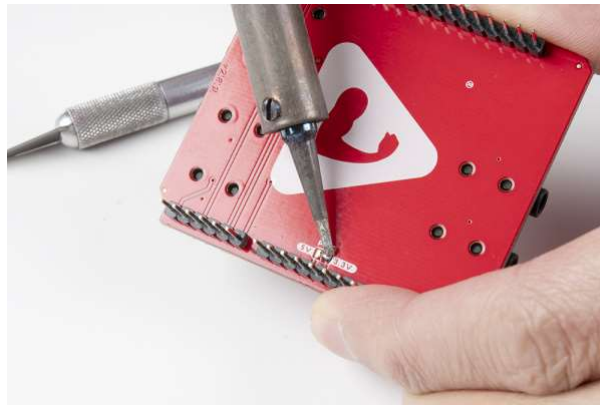
If you do not have a USB isolator, try using a laptop running on battery power and ensure that you are not connecting the laptop to a wall outlet.



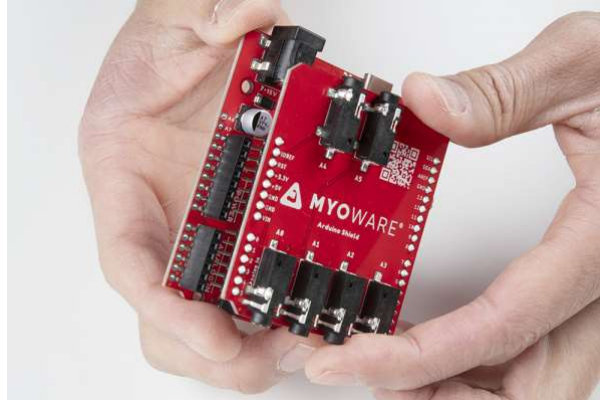
*Image Courtesy of Advancer Technologies taken from the MyoWare 2.0 Quick Start Guide*

We also recommend using a wireless keyboard and mouse when programming in Arduino. The muscle sensor is sensitive enough that it can also pick up a signal from the trackpad or keyboard on your laptop when running on power. As a result, the muscle sensor will have false readings and the Arduino's analog pin will read high.

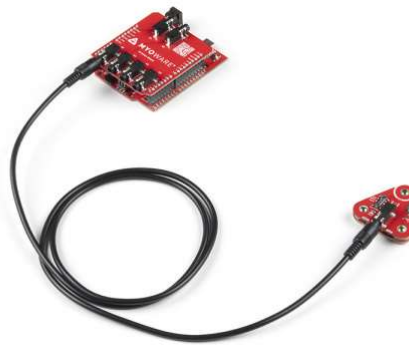
For users using a 3.3V Arduino, we recommend adjusting the PWR jumper by cutting the default trace and adding a solder jumper between the center pad and the 3.3V side. For users using a 5V Arduino, you can leave the jumper connected to the 5V side. For more information on modifying the jumpers, check out our tutorial on working with jumper pads and PCB traces.



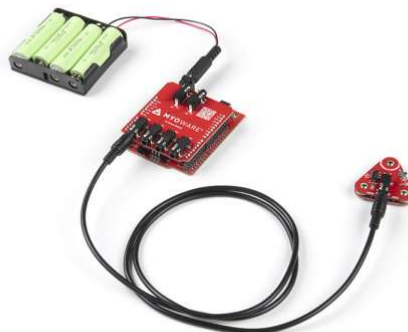
Stack the Arduino Shield on your chosen development board with the Arduino Uno R3 footprint.



Insert the TRS cable between the MyWare 2.0 Link Shield and Arduino Shield's TRS connector. In this case, we will be using the connector labeled as A0 . Make sure to adjust the code when using the other channels.



For remote applications, we recommend using a battery pack. In this case, we used AA batteries (NiMH) and a 4xAA battery pack to power the system.



Make sure to remove the power supply from your computer before connecting your Arduino with the MyoWare 2.0 Muscle Sensor to your USB port. This is to prevent noise and safeguard against electrical shock when connected to the power grid. This includes any computer docks and external monitors.



*Removing power from a laptop before connecting the MyoWare.*

When ready, insert the USB cable between your Arduino and computer's USB port.



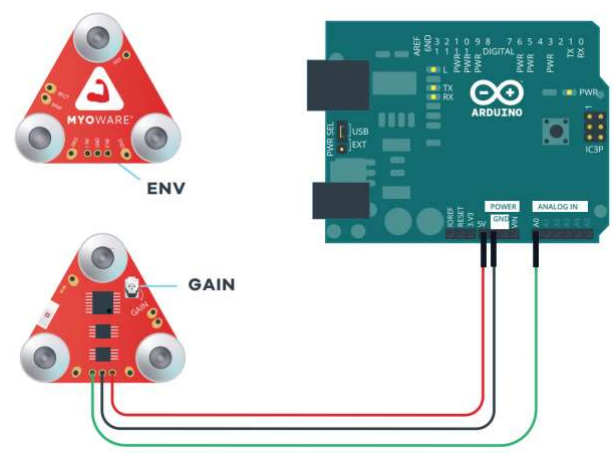
For users transmitting sensor data via Bluetooth, attach a battery pack to the RedBoard Artemis acting as the Bluetooth peripheral device. This is assuming that it has the Bluetooth peripheral example code uploaded. Connect the RedBoard Artemis acting as the Bluetooth central device to your computer's USB port. Since the Bluetooth peripheral device is disconnected, you could connect your laptop to a power supply since the MyoWare 2.0 Muscle Sensor is not connected to the main outlet. You just need to make sure to unplug the power supply every time you need to reprogram the RedBoard Artemis acting as the peripheral device.



**Note:** Confused about which board you are uploading code to? Try getting a sticky note and labeling each RedBoard Artemis board with the word peripheral and central.



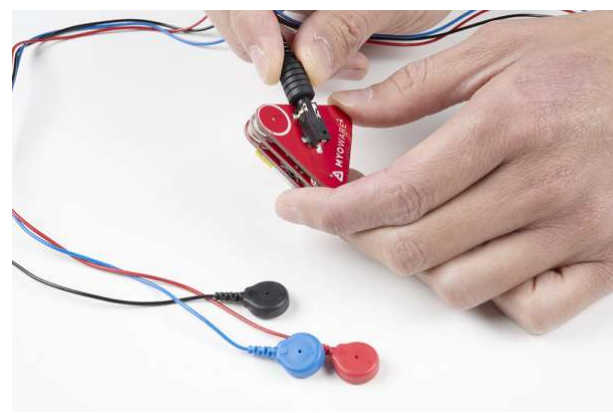
**Note:** For users that are soldering, you could also solder wires to the PTH pads on the muscle sensor and connect to a microcontroller's analog input. The following image shows the MyoWare 2.0 Muscle Sensor connected to an Arduino Uno R3. However, you could connect it to the Raspberry Pi's RP2040 or any other microcontroller that has an analog input. Just make sure to connect 3.3V for the muscle sensor's VIN if your microcontroller's system voltage is 3.3V.



*Image Courtesy of Advancer Technologies taken from the MyoWare 2.0 Advanced Guide*

### External Cable

For muscles that require you to mount the sensor pads away from the sensor, snap the MyoWare 2.0 Cable Shield on the bottom of the MyoWare 2.0 Muscle Sensor. Then insert the sensor cable into the 3.5mm TRS connector.



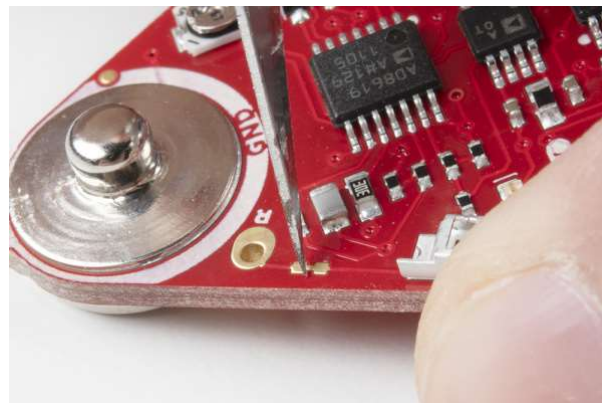
Attach the EMG pads to the snap connectors. After cleaning the skin and selecting the muscle, peel and remove the backs of the electrodes to expose the adhesive. Then attach the pads to the target muscle group based on the MID, END, and REF.

Snap Connector	TRS Pin	Electrode Pin [CAB-12970]
Reference [REF]	Sleeve	Black
End [END]	Ring	Blue
Middle [MID]	Tip	Red

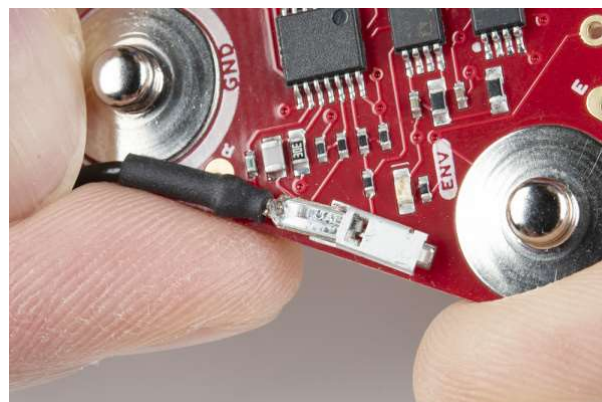
## Reference Cable

**Note:** The built-in reference snap connector should be sufficient for most applications. However, for users having a difficult time connecting to a muscle group, we recommend getting the reference cable. This is mainly intended for large muscle groups (such as the quad, hamstring, etc.).

With a hobby knife, slice the reference jumper pad closest to the GND pin. There are traces near the jumper so you will need to make sure to avoid cutting traces that are adjacent to jumper pad. For more information on modifying the jumpers, check out our tutorial on working with jumper pads and PCB traces.

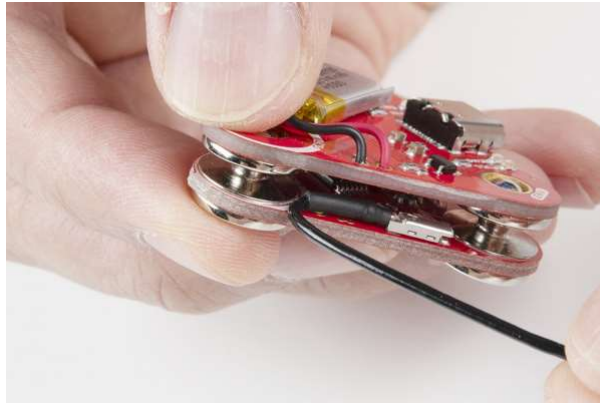


Then insert the reference cable into the MyoWare 2.0 Muscle Sensor, slide the pin into the housing with the tab facing away from the board.

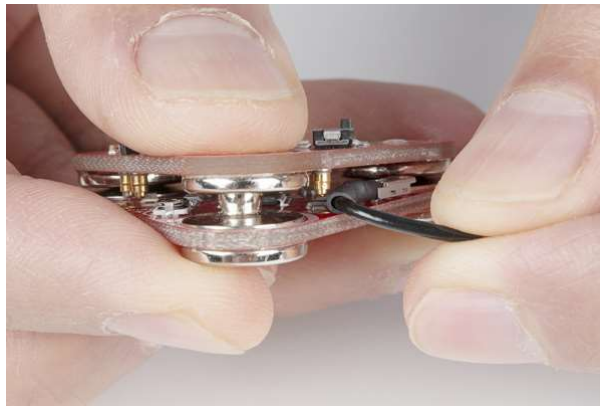


When stacking a shield on top of the MyoWare 2.0 Muscle Sensor, make sure carefully pull the cable away from the shield's snap pin.





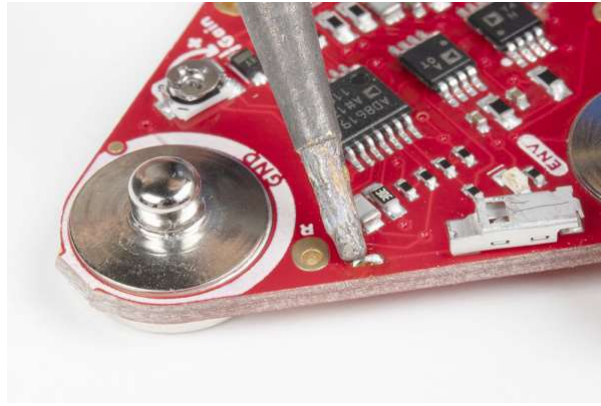
**Note:** You'll notice that the reference cable's wire is over the MyoWare 2.0 Link Shield's reference pogo pin. The tolerance is small so you will need to make sure to pull the cable away gently when stacking a shield on top.



To remove the reference cable, push the tab into the housing with the end of a flathead screwdriver and gently pull the pin out of the housing.



To close the jumper pad, add a small amount of solder on the jumper pad.



## Disconnecting the MyoWare Shield

When prototyping, you will want to test out different shields or you may need to adjust the trim pot. To remove the boards, you will need a flathead screwdriver. With power off, insert the flathead between the snap connectors and ensure that there are no components in the way of the flathead. Gently slide the flathead between the snap connectors.



Angle the flathead against the connectors until the boards disconnect.



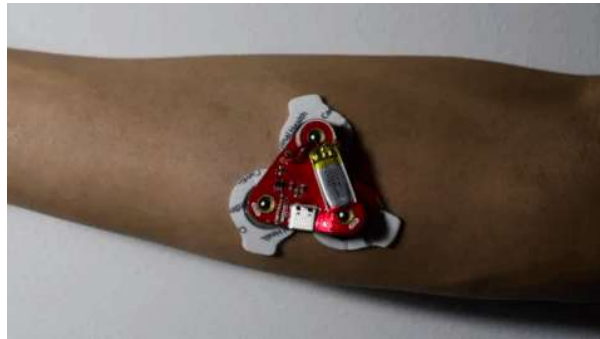
With one side disconnected, pull the boards away from each other so that the other two snap connectors disconnect.



## Standalone Example 0: LED(s)

To test the MyoWare 2.0 Muscle Sensor without an Arduino, you could use the MyoWare 2.0 Power Shield or the LED Shield. Both shields have a built-in battery removing the need to solder a power source to the sensor. After stacking the boards together and sticking it to a muscle group, flip the power switch to the ON position. Start flexing the muscle to see if you can light up the ENV LED on the Power Shield or the LED segments on the LED Shield!

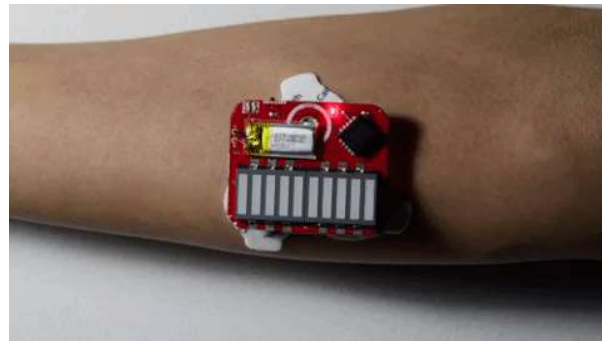
If you are using the MyoWare 2.0 Power Shield, observe the ENV LED. The LED will be off. As you flex, the voltage from the ENV pin increases. As a result, the ENV LED will start to light up and increase in intensity as you flex the muscle. If you look closely between the stacked boards, you will also notice the ENV LED on the MyoWare 2.0 Muscle Sensor light up.



If you are using the MyoWare 2.0 LED Shield, observe the LED segments. The LED segments will be off. As you flex, the voltage from the ENV pin increases. As a result, the LED segments will start to light up. The more you flex the muscle, the more LED segments will light up. When fully flexed, all 10x LED segments will be on.

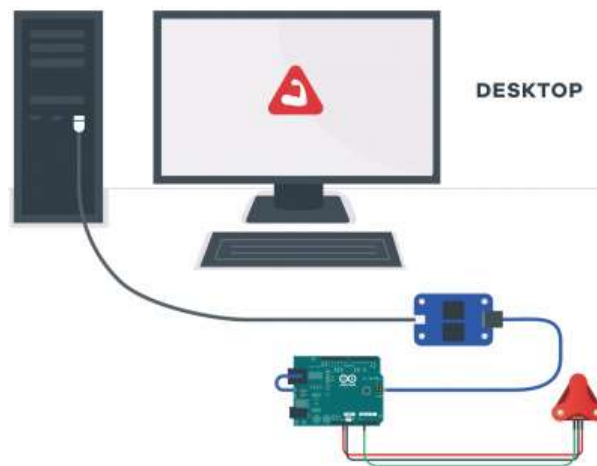


For users that cut the jumper labeled as MODE (i.e. enabling dot mode) on the MyoWare 2.0 LED Shield, you will see a different effect. A leading LED bar (i.e. "dot") will light up and traverse the segments. Any trailing LEDs behind it will be turned off instead of filling in with light. When fully flexed, one LED segment should be on.



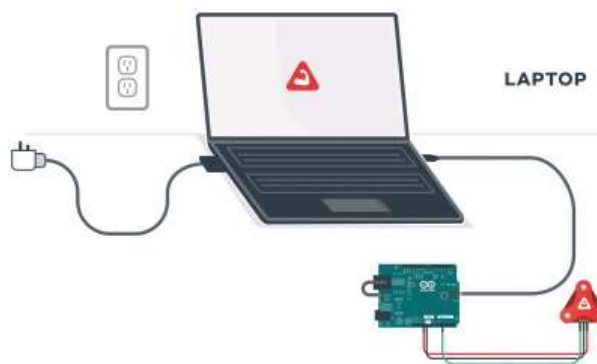
## Arduino Example 1: Analog Read - Single Sensor

⚡ **Danger:** When connecting your MyoWare 2.0 Muscle Sensor to a computer for Serial debugging, it is suggested to have a **USB isolator** with the MyoWare 2.0 Muscle Sensor to prevent noise in the inputs and safeguard against electrical shock when connected to a power grid (such as a computer power supply, computer docks, and external monitors connected to a wall outlet). The diagram below shows USB isolator (as shown as a blue board) between a desktop and the Arduino's USB port.



*Image Courtesy of Advancer Technologies taken from the MyoWare 2.0 Quick Start Guide*

If you do not have a USB isolator, try using a laptop running on battery power and ensure that you are not connecting the laptop to a wall outlet.



*Image Courtesy of Advancer Technologies taken from the MyoWare 2.0 Quick Start Guide*

We also recommend using a wireless keyboard and mouse when programming in Arduino. The muscle

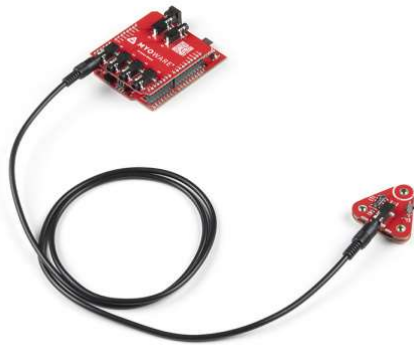
sensor is sensitive enough that it can also pick up a signal from the trackpad or keyboard on your laptop when running on power. As a result, the muscle sensor will have false readings and the Arduino's analog pin will read high.

**Note:** This example assumes you are using the latest version of the Arduino IDE on your desktop. If this is your first time using Arduino, please review our tutorial on installing the Arduino IDE. If you are using the RedBoard Artemis, please check out our guide on installing board definitions.

### Example 1: Analog Read - Single

⚠ **Warning:** Do not change the output (between RAW, RECT, ENV) for the MyoWare 2.0 Link Shield while the board is powered. Adjusting the output switch can cause voltage spikes and damage the muscle sensor.

The following example uses an analog input on the RedBoard Plus with an Atmega328P to read the muscle sensor's analog signal. However, you could also use any microcontroller with the Arduino Uno R3 footprint.



If you have not already, select your Board (in this case the **Arduino Uno**), and associated COM port. Upload the code to the board.

```

/*
MyoWare Example_01_analogRead_SINGLE
SparkFun Electronics
Pete Lewis
3/24/2022
License: This code is public domain but you buy me a beverage if you use this and we meet some
day.
This code was adapted from the MyoWare analogReadValue.ino example found here:
https://github.com/AdvancerTechnologies/MyoWare\_MuscleSensor

This example streams the data from a single MyoWare sensor attached to ADC A0.
Graphical representation is available using Serial Plotter (Tools > Serial Plotter menu).

*Only run on a laptop using its battery. Do not plug in laptop charger/dock/monitor.

*Do not touch your laptop trackpad or keyboard while the MyoWare sensor is powered.

Hardware:
SparkFun RedBoard Artemis (or Arduino of choice)
USB from Artemis to Computer.
Output from sensor connected to your Arduino pin A0

This example code is in the public domain.
*/

void setup()
{
  Serial.begin(115200);
  while (!Serial); // optionally wait for serial terminal to open
  Serial.println("MyoWare Example_01_analogRead_SINGLE");
}

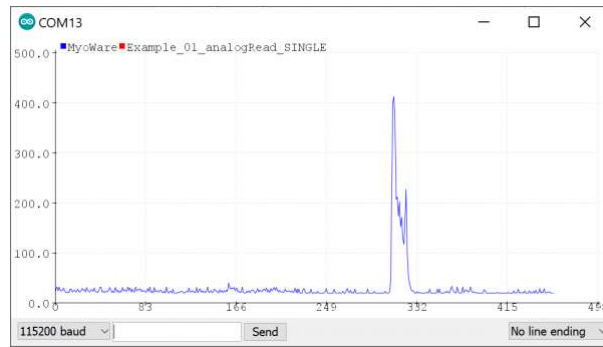
void loop()
{
  int sensorValue = analogRead(A0); // read the input on analog pin A0

  Serial.println(sensorValue); // print out the value you read

  delay(50); // to avoid overloading the serial terminal
}

```

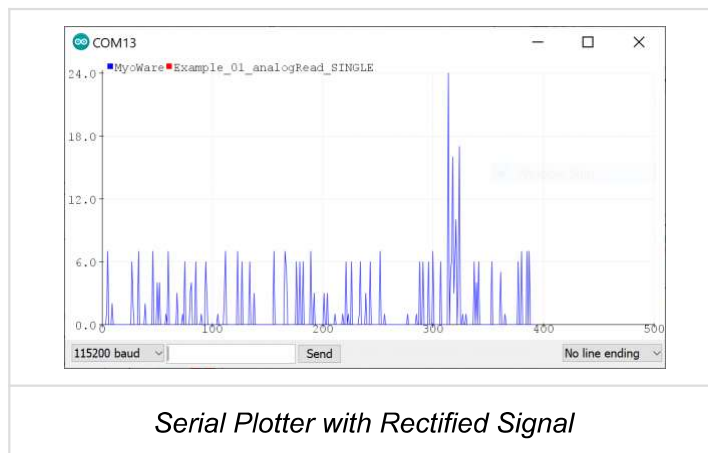
Once uploaded, open the Arduino Serial Plotter (**Tools > Serial Plotter**) to **115200** baud. Flip the power switch on the MyoWare 2.0 Link Shield to provide power and start flexing! You should see something similar to the output below with the envelope signal. In this case, the MyoWare 2.0 Muscle Sensor was attached to my right forearm. For those that are interested in viewing the serial output, you can also open the Arduino Serial Monitor to see the values.



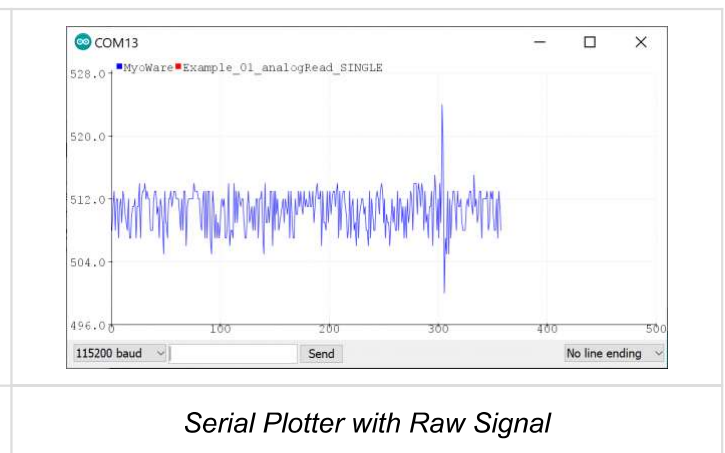
*Serial Plotter with Envelope Signal*

To view the Rectified or Raw signals, flip the power switch to the OFF position on the Link Shield, adjust the OUTPUT switch to either the Rectified or Raw position and then turn the power switch to the ON position. Reopen the Serial Plotter and you will see a smaller signal output when flexing your forearm.

The images below show the forearm being flexed at around 300ms when the signal increases by a value of about 22 for the rectified output and about 20 for the raw output. Note that the raw signal will be centered around a voltage offset of about  $+VIN/2$ . Make sure that the  $+VIN$  is the maximum voltage of your microcontroller's ADC to ensure that you can see the positive and negative portions of the waveform. The output shown in the graphs are values and have not been converted to voltages.



*Serial Plotter with Rectified Signal*



*Serial Plotter with Raw Signal*

## Arduino Example 2: Transmitting Sensor Data via Bluetooth® - Single Sensor

**Note:** When uploading code, make sure to select the correct COM port since you will be using two RedBoard Artemis boards on your computer. You may want to disconnect the board that you are not uploading code to avoid confusion or label the boards with some tape. To save time, you can also have two instances of the Arduino IDE open and select the COM port for each respective board.

**Note:** This example assumes you are using the latest version of the Arduino IDE on your desktop. If this is your first time using Arduino, please review our tutorial on installing the Arduino IDE. Since this example uses the RedBoard Artemis for Bluetooth, please check out our guide on installing board definitions.

The following example requires two RedBoard Artemis boards to send and receive sensor data from one muscle sensor. This is useful for users that need a clean signal or want to transmit the sensor data wirelessly.



For an explanation of how the Bluetooth peripheral and central devices works, we recommend looking at the Arduino reference language under the ArduinoBLE Library for a quick introduction.

### Example 2a: Bluetooth Peripheral "Send"

The following example uses a RedBoard Artemis to transmit the analog reading wirelessly via Bluetooth. If you have not already, select your Board (in this case the **RedBoard Artemis**), and associated COM port. Copy the code below and paste into your Arduino IDE. Then upload the code to the board.



```
/*
MyoWare Sensor BLE Peripheral SINGLE SENSOR Example
SparkFun Electronics
Pete Lewis
3/17/2022

This example reads a single MyoWare Muscle Sensor, and then gets that data from this RedBoard
Artemis
(the peripheral) to a second RedBoard Artemis (the central) over BLE.

This Artemis, aka the "BLE Peripheral", will read the sensor on A0.
It will then update the value to the "bluetooth bulletin board."

Note, in BLE, you have services, characteristics and values.
Read more about it here:

https://www.arduino.cc/reference/en/libraries/arduinoBLE/

Note, before it begins reading the ADC and updating the data,
It first sets up some BLE stuff:
  1. sets up as a peripheral
  2. sets up a service and characteristic (the data)
     -Note, Services and characteristics have custom 128-bit UUID,
     -These must match the UUIDs in the code on the central device.
  3. advertises itself

In order for this example to work, you will need a second Artemis, and it will
need to be programmed with the provided code specific to being a central device,
looking for this specific peripheral/service/characteristic.

Note, both the service and the characteristic get unique UUIDs
(even though they are extremely close to being the same thing in this example)

The second Artemis, aka the "BLE Central," will subscribe to the first board's
characteristic, and check to see if the value has been updated. When it has been
updated, it will print the value to the serial terminal.

Hardware:
MyoWare Sensor with Link Shield snapped on top.
TRS cable from Link Shield to A0 port of Arduino Shield.
Arduino Shield pressed into RedBoard Artemis.
USB from Artemis to Computer.

** For consistent BT connection follow these steps:
** 1. Reset Peripheral
** 2. Wait 5 seconds
** 3. Reset Central
** 4. Enjoy BT connection
**
** ArduinoBLE does not support RE-connecting two devices.
** If you loose connection, you must follow this hardware reset sequence again.
```

```
This example code is in the public domain.
*/

#include <ArduinoBLE.h>

BLEService sensorDataService("19b10000-e8f2-537e-4f6c-d104768a1214"); // BLE Service named "sensorDataService"

// BLE Data Characteristic - custom 128-bit UUID, readable, writable and subscribable by central
// Note, "BLENotify" is what makes it subscribable
BLEByteCharacteristic dataCharacteristic("19b10001-e8f2-537e-4f6c-d104768a1214", BLERead | BLEWrite | BLENotify);

const int ledPin = LED_BUILTIN; // pin to use for the LED

void setup()
{
  Serial.begin(115200);
  while (!Serial); // optionally wait for serial terminal to open, useful with micros that have native USB
  Serial.println("MyoWare Single Sensor Example - BLE Peripheral");

  pinMode(ledPin, OUTPUT); // set LED pin to output mode

  if (!BLE.begin()) { // begin initialization
    Serial.println("starting BLE failed!");
    while (1);
  }
  Serial.println("BLE initialized successfully");

  BLE.setLocalName("MYOWARE1"); // set advertised local name
  BLE.setAdvertisedService(sensorDataService); // set advertised service UUID
  sensorDataService.addCharacteristic(dataCharacteristic); // add the characteristic to the service
  BLE.addService(sensorDataService); // add service
  dataCharacteristic.writeValue(0); // set the initial value for the characteristic
  BLE.advertise(); // start advertising
}

void loop()
{
  BLEDevice central = BLE.central(); // listen for BLE peripherals to connect

  if (central) // if a central is connected to peripheral
  {
    Serial.print("Connected to central: ");
    Serial.println(central.address()); // print the central's MAC address

    Serial.println("Reading Sensor and writing BLE characteristic values now...");

    // while the central is still connected to peripheral:
    while (central.connected())
```

```

{
  int val_int = analogRead(A0); // Read the sensor attached to Analog Pin A0
  byte val_byte = map(val_int, 0, 1023, 0, 255); // map the int to a byte
  delay(10);
  dataCharacteristic.writeValue(val_byte); // "post" to "BLE bulletin board"
  // Note, because our second Artemis in this example (the central) is subscribed to this ch
aracteristic,
  // it can simply call Characteristic.valueUpdated() to see if it has been updated.
  // valueUpdated() will return True if updated, or false if no update has happened.
  // If it has been updated, the central Artemis can read the latest value using Characteris
tic.readValue();
}
Serial.print(F("Disconnected from central: ")); // when the central disconnects, print it ou
t
Serial.println(central.address());
}
}

```

### Example 2b: Bluetooth Central "Receive"

The following example uses a RedBoard Artemis to receive the analog reading from the peripheral RedBoard Artemis wirelessly via Bluetooth. If you have not already, select your Board (in this case the **RedBoard Artemis**), and associated COM port. Copy the code below and paste into your Arduino IDE. Then upload the code to the board.

```
/*
MyoWare Receiver BLE Central SINGLE SENSOR Example
SparkFun Electronics
Pete Lewis
3/17/2022

This example sets up a SparkFun RedBoard Artemis as a BLE central device,
Then, it connects to a second Artemis peripheral device that is reading a single MyoWare
Muscle sensor. It then streams the data on the Serial Terminal.

Note, in BLE, you have services, characteristics and values.
Read more about it here:

https://www.arduino.cc/reference/en/libraries/arduinoBLE/

Note, before it begins checking the data and printing it,
It first sets up some BLE stuff:
  1. sets up as a central
  2. scans for any peripherals
  3. Connects to the device named "MYOWARE1"
  4. Subscribes MYOWARE1's data characteristic

In order for this example to work, you will need a second Artemis, and it will
need to be programmed with the provided code specific to being a peripheral device,
and advertising as MYOWARE1 with the specific characteristic UUID.

Note, both the service and the characteristic get unique UUIDs
(even though they are extremely close to being the same thing in this example)

This Artemis, aka the "BLE Central," will subscribe to the peripheral board's
characteristic, and check to see if the value has been updated. When it has been
updated, it will print the value to the serial terminal.

Hardware:
SparkFun RedBoard Artemis
USB from Artemis to Computer.

** For consistent BT connection follow these steps:
** 1. Reset Peripheral
** 2. Wait 5 seconds
** 3. Reset Central
** 4. Enjoy BT connection
**
** ArduinoBLE does not support RE-connecting two devices.
** If you loose connection, you must follow this hardware reset sequence again.

This example code is in the public domain.
*/

#include <ArduinoBLE.h>

void setup()
```

```

{
  Serial.begin(115200);
  while (!Serial); // optionally wait for serial terminal to open, useful with micros that have
  native USB
  Serial.println("MyoWare Single Sensor Example - BLE Central");

  if (!BLE.begin()) // initialize the BLE hardware
  {
    Serial.println("starting BLE failed!");
    while (1);
  }
  Serial.println("BLE initiallized successfully");

  BLE.scanForUuid("19b10000-e8f2-537e-4f6c-d104768a1214"); // start scanning for peripherals
}

void loop()
{
  BLEDevice peripheral = BLE.available(); // check if a peripheral has been discovered

  if (peripheral) // discovered a peripheral, print out its info
  {
    Serial.print("Found ");
    Serial.print(peripheral.address());
    Serial.print(" ");
    Serial.print(peripheral.localName());
    Serial.print(" ");
    Serial.print(peripheral.advertisedServiceUuid());
    Serial.println();

    if (peripheral.localName() != "MYOWARE1")
    {
      return;
    }

    BLE.stopScan();

    checkUpdate(peripheral);

    Serial.println("Starting to scan for new peripherals again...");
    BLE.scanForUuid("19b10000-e8f2-537e-4f6c-d104768a1214"); // peripheral disconnected, scan ag
ain
    Serial.println("Scan has begun...");
  }
}

// Connect to peripheral
// Then continue to check if the data has been updated,
// If so, print it to terminal
void checkUpdate(BLEDevice peripheral)
{
  Serial.println("Connecting ..."); // connect to the peripheral

```

```

if (peripheral.connect())
{
    Serial.println("Connected");
} else {
    Serial.println("Failed to connect!");
    return;
}

Serial.println("Discovering attributes ..."); // discover peripheral attributes
if (peripheral.discoverAttributes())
{
    Serial.println("Attributes discovered");
} else {
    Serial.println("Attribute discovery failed!");
    peripheral.disconnect();
    return;
}

// retrieve the data characteristic
BLECharacteristic dataCharacteristic = peripheral.characteristic("19b10001-e8f2-537e-4f6c-d104
768a1214");

if (!dataCharacteristic)
{
    Serial.println("Peripheral does not have that characteristic!");
    peripheral.disconnect();
    return;
} else if (!dataCharacteristic.canWrite())
{
    Serial.println("Peripheral does not have a writable characteristic!");
    peripheral.disconnect();
    return;
} else if (!dataCharacteristic.canRead())
{
    Serial.println("Peripheral does not have a readable characteristic!");
    peripheral.disconnect();
    return;
} else if (!dataCharacteristic.canSubscribe())
{
    Serial.println("Characteristic is not subscribable!");
    peripheral.disconnect();
    return;
} else if (!dataCharacteristic.subscribe())
{
    Serial.println("subscription failed!");
    peripheral.disconnect();
    return;
}

while (peripheral.connected()) // while the peripheral is connected
{

```

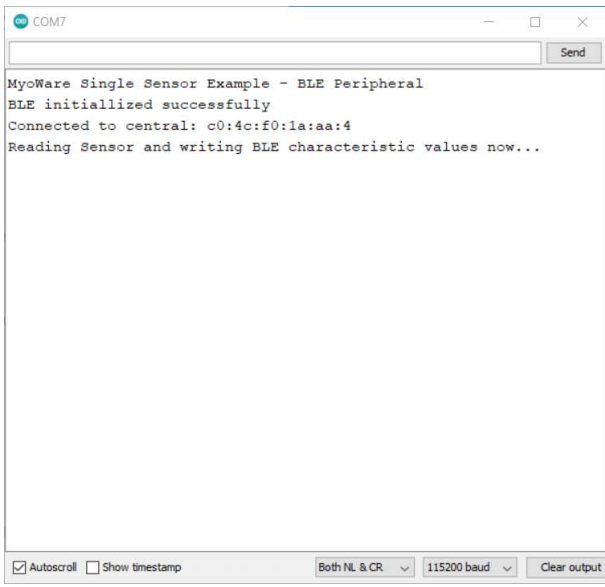
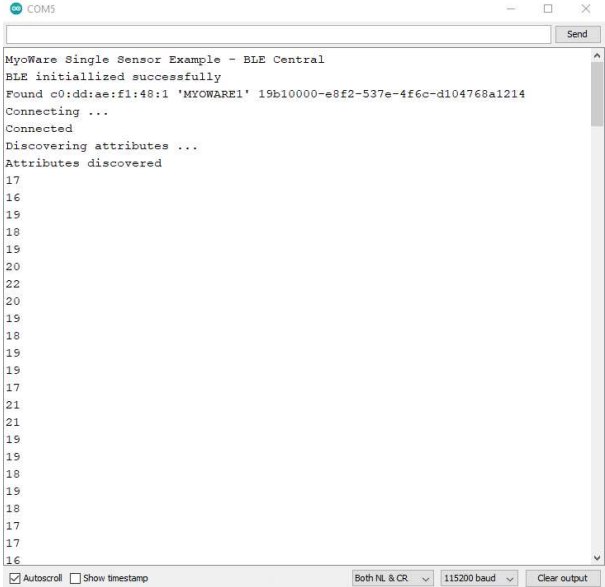
```

if (dataCharacteristic.valueUpdated()) // Check to see if the value of the characteristic has
s been updated
{
    byte received_val = 0;
    dataCharacteristic.readValue(received_val); // note, readValue returns nothing, and needs
the variable to be passed by reference
    Serial.println(received_val);
}
delay(1);
}
Serial.println("Peripheral disconnected");
}

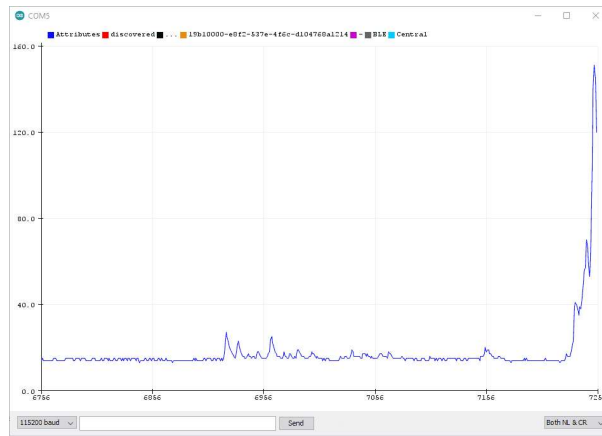
```

## What You Should See

With two instances of the Arduino IDE open with a COM port selected for each respective board, you should see the following output in the Serial Monitor set to **115200** baud. If the Bluetooth fails to connect, try pushing the reset button on the Peripheral first, wait about 5 seconds, then hit the reset button on the Central. Once connected, the muscle sensor readings will be transmitted wirelessly to the central. Try flexing your arm to see if you can see the values increase.

 <p>COM7</p> <p>MyoWare Single Sensor Example - BLE Peripheral  BLE initialized successfully  Connected to central: c0:4c:f0:1a:aa:4  Reading Sensor and writing BLE characteristic values now...</p> <p><input checked="" type="checkbox"/> Autoscroll <input type="checkbox"/> Show timestamp Both NL &amp; CR 115200 baud Clear output</p>	 <p>COM5</p> <p>MyoWare Single Sensor Example - BLE Central  BLE initialized successfully  Found c0:dd:ae:f1:48:1 'MYOWARE!' 19b10000-e8f2-537e-4f6c-d104768a1214  Connecting ...  Connected  Discovering attributes ...  Attributes discovered  17  16  19  18  19  20  22  20  19  18  19  19  17  21  21  19  19  18  19  18  17  17  16</p> <p><input checked="" type="checkbox"/> Autoscroll <input type="checkbox"/> Show timestamp Both NL &amp; CR 115200 baud Clear output</p>
<p>Debug Output from Peripheral Device</p>	<p>Debug Output from Central Device</p>

For users looking for a graphical view of the output, close the Arduino Serial Monitor for the Bluetooth set as the Central device (i.e. the RedBoard Artemis receiving the sensor data). Then open the Arduino Serial Plotter. You should see something similar to the output below. In this case, we flexed our forearm muscle causing the values to increase. Relaxing the muscle caused the values to decrease.



## More Examples!

Looking for more examples? Try checking out the GitHub Repository to use multiple analog channels via BLE! There's also a project example that connects to the Tsunami Super WAV Trigger (Qwiic) and speaker. Or check out the MicroPython example with the RP2040 located in the Python\_Examples folder.

**GITHUB REPO: SPARKFUN MYOWARE CODE EXAMPLES**

## Troubleshooting

### 🔗 Not working as expected and need help?

If you need technical assistance and more information on a product that is not working as you expected, we recommend heading on over to the SparkFun Technical Assistance page for some initial troubleshooting.

**SPARKFUN TECHNICAL ASSISTANCE PAGE**

If you don't find what you need there, the Reddit Forum under MyoWare or our SparkFun Forums under Biometrics are a great place to find and ask for help.

**REDDIT FORUM: MYOWARE**

**SPARKFUN FORUMS: BIOMETRICS**

For your convenience, the following troubleshooting tips are provided from the MyoWare 2.0 Quick Start Guide.

- The Vin LED should remain on constantly as long as the sensor is powered.
- The ENV LED might immediately turn on when power is turned on but it should turn off after a second or two; afterwards, it should only turn on when the sensor detects muscle flexion.
- The ENV LED might sporadically turn on and off or simply remain on if the sensor (or external cables) is not connected to electrodes on the skin. This is normal behavior.
- The ENV LED lighting up seemingly at random while connected to electrodes on the skin usually indicates poor contact between the electrodes and the skin. Try cleaning the area and applying new electrodes.



- The ENV LED not lighting up during flexion could be caused by various issues with the sensor setup. Double check the steps and tips of this guide to ensure proper setup.
- Most of the time the sensor gain does not need to be adjusted at all.
- Recommended that you do not adjust the sensor's gain until you get a consistent, quality signal from the sensor.

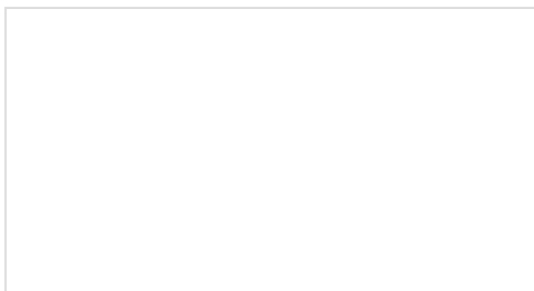
## Resources and Going Further

Now that you've successfully got your MyoWare 2.0 Muscle Sensor up and running, it's time to incorporate it into your own project! Try embedding the sensor in a prosthetic arm, controlling a servo to grip something, Halloween costume, video game, or turning on LEDs in a dance performance! For more information, check out the resources below:

- Board Dimensions
  - Muscle Sensor (PNG)
  - Sensor Cover (PNG)
  - Cable Shield (PNG)
  - Power Shield (PNG)
  - LED Shield (PNG)
  - Link Shield (PNG)
  - Arduino Shield (PNG)
- Advancer Technologies: MyoWare 2.0
  - Quickstart Guide (PDF) (4.37MB)
  - Advanced Guide (PDF) (9.00MB)
  - Patents<sup>[1]</sup>
- MCP73831 Datasheet (PDF)
- Polymer Lithium Ion Battery (40mAh)
  - Datasheet (PDF)
  - MSDS (PDF)
- Arduino Reference Language: ArduinoBLE Library
- GitHub: SparkFun MyoWare Code Examples - Repository for example code using Arduino, MicroPython, and MakeCode
- MyoWare 2.0 Ecosystem Page
- SFE Product Showcase

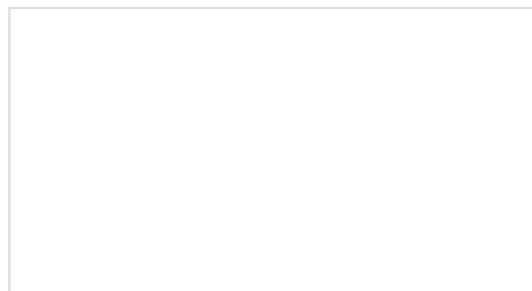
[1] Note: This product is patent protected. To prevent counterfeit boards, the Eagle design files and GitHub hardware repository are not shared for boards in the MyoWare 2.0 ecosystem.

Need some inspiration for your next project? Check out some of these related tutorials tagged with biometrics.



### Hackers in Residence - Hacking MindWave Mobile

Review, teardown, and hacking tutorial for the MindWave Mobile, a \$99 commercial grade EEG

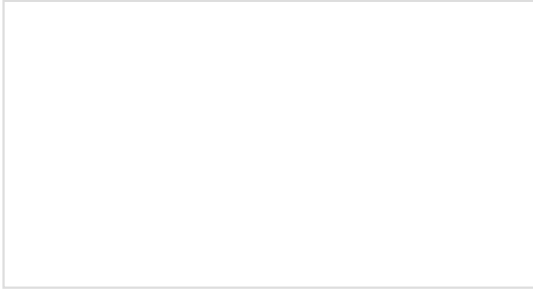


### MAX30105 Particle and Pulse Ox Sensor Hookup Guide

The SparkFun MAX30105 Particle Sensor is a flexible and powerful sensor enabling sensing of distance,

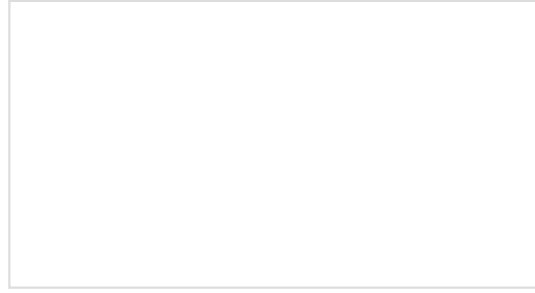
sensor.

heart rate, particle detection, even the blinking of an eye. Get ready. Set. Shine!



### Fingerprint Scanner (GT-521Fxx) Hookup Guide

This tutorial provides information about how to connect to ADH-Tech's fingerprint scanner (GT-521F32) and how to use it with Hawley's FPS\_GT511C3 library for Arduino.



### SparkFun Pulse Oximeter and Heart Rate Monitor Hookup Guide

Find out your oxygen saturation level or check out your heart rate using the MAX30101 biometric sensor and MAX32664 Biometric Hub via I2C!